ABSTRACT

Title of Thesis:	DISTRIBUTED PASSIVE SENSOR NETWORK FOR THE GEOLOCATION OF RF EMITTERS
	Matthew Dillon, Master of Science, 2019
Thesis Directed By:	Professor Anthony Ephremides Department of Electrical and Computer Engineering

The ability to localize an RF emitter has emerged in both commercial and military technology, and is an important capability in modern cognitive radios to achieve spectral awareness. Of importance, is the accuracy of the geolocation of the RF emitter. In this thesis, we address the blind localization problem given a network of software-defined radio receivers that monitor the spectrum to determine the presence of an unknown emitter. We discuss the underlying challenges and various approaches to the geolocation problem that can be utilized. In particular, two algorithms that are used extensively in literature are investigated: time-difference of arrival, and power-difference of arrival. In the first part of the thesis, the algorithms are presented, and the error performance is characterized analytically, and then conducted through simulation. A more robust method which implements the fusion of both algorithms for an improved estimation. In the second part, we conduct a small-scale laboratory emulation of the geolocation algorithms on a network of radios to

contrast the simulation results of the algorithms from the emulation results. The results provided insight to the shortcomings of each algorithm, and potential extensions for further accuracy improvement.

DISTRIBUTED PASSIVE SENSOR NETWORK FOR THE GEOLOCATION OF RF EMITTERS

by

Matthew Dillon

Thesis submitted to the Faculty of the Graduate School of the
University of Maryland, College Park, in partial fulfillment
of the requirements for the degree of
Master of Science
2019

Advisory Committee:
Professor Anthony Ephremides, Chair/Advisor
Professor Richard La
Professor Min Wu

Acknowledgements

I would like to thank my family, friends, and my advisor Dr. Anthony Ephremides for all the support and guidance I have received. I would also like to thank my committee members for agreeing to be part of my committee. Special thanks to my colleagues at US Naval Research Laboratory for all of their support

Table of Contents

List of Tables

List of Figures

Chapter 1:   Introduction

1.1 Overview

The geolocation of RF emitters is an important capability in modern spectrum situational awareness. Geolocation systems are seen in radar, where a radio wave is transmitted. When the wave hits a target, it is reflected back to the transmitter, where its time of return is determined. This is one example of time-of arrival being used as a way to track a target and estimate its position. Another example is the Global positioning system (GPS), which relies on satellites in space to triangulate the current position. One GPS receiver will receive a signal from multiple satellites to estimate its precise position. Being able to identify a target emitter has become useful in military and law-enforcement applications and even civilian use (automated museum guides or GPS). In this thesis research, we first explore the different types of localization methods and their applications in Chapter 1 and discuss which geolocation methods that are most practical to design a geolocation sensor network in an effort to track a target emitter. In Chapter 2, we detail the proposed geolocation methods to be used in this paper. In this case, we apply geolocation methods that rely on received signal strength indicator (RSSI) differences between sensors in the network, also known as Power-difference of arrival (PDOA) and difference between time arrival between sensors, or time-difference of arrival (TDOA). After the methods are detailed, they are analyzed with their performance both analytically and through iterative simulations in software. The PDOA analysis and TDOA conducted in Chapter 3. After the individual analysis, their observed performance motivates a proposed hybrid method that utilizes both TDOA and PDOA measurements, which is also presented in Chapter 4. With the methods laid out, Chapters 5-6 detail a testbed to emulate the geolocation sensor network on software-defined radio (SDR) platforms through a channel emulator environment, vs. the logistical and time-consuming challenge of an actual field test. Chapter 7 details the actual emulation tests performed and details the results, also comparing them to the simulated results of the individual algorithms. Chapter 8 details conclusions and future extensions based on the results of this research.

1.2 The Geolocation Problem

The way to solve the geolocation problem depends entirely on the devices available within the geolocation system, and the a priori information available. Two main types of target geolocation exist: active and passive geolocation. Active geolocation requires the use of nodes configured as both transmitters and receivers. One main example of active geolocation is in radar, where the geolocation radio is sending out its own pulse, and determining the time of arrival of the reflected pulse. Radar applications are readily known; however, not all other positioning systems have the ability to transmit pulses to determine the reflected signal, particularly involving low-power sensor networks, such as GPS. GPS is an example of passive sensing in that the GPS receiver inside a phone will receive a signal from satellites to estimate its own position. This research focuses exclusively on the passive geolocation problem. There are three main passive geolocation methods: triangulation, trilateration, and multilateration. All of these methods involve one common property of the receivers being passive. That is, the receivers do not transmit any RF signal in an effort to locate a target.

The triangulation technique relies on the method of angles of arrival at the receiver from the incoming transmitter [5]. Typically, an array of directional antennas is needed to determine the angle of arrival of a signal. Fig. (1.1) shows how knowing the angle of arrival, the distance can be calculated as a law-of-sines and cosines problem. This method is explored in Angle of arrival (AoA) scenarios. [11] [12]. For example, in Fig. (1.1), we do not know a priori information about node 1 want to find the distance A and B. we can measure the angle of the received signal at node 3 and node 1 transmitted by 1. We then have knowledge of the angles. $\propto$, β, and γ. The side lengths B and A can be solved for with the law of sines and cosines. This technique will not be applied in our research, since it requires the use of directional antennas. This research focuses on a more minimalist approach to the hardware requirements, so we assume a single omnidirectional receiver antenna on the sensors.

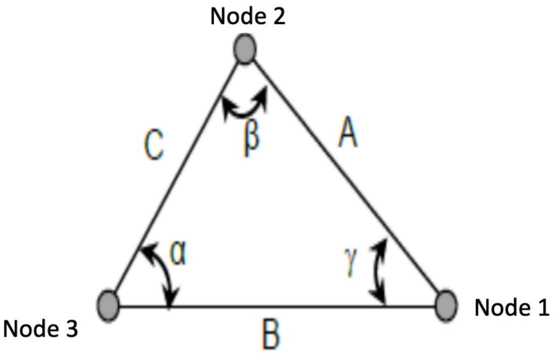


Figure 1.1 - The triangulation technique, used to estimate the position when the general direction of the transmitter is known by at least three nodes

The trilateration technique uses either the RSSI of the transmitter or the time-of-arrival of (ToA). For ToA, the receiver and transmitter must be synchronized. GPS is a good example of this, as the GPS receiver and satellite are well synchronized by atomic clocks with good resolution, where the synchronization drift is corrected regularly. In knowing the ToA or RSSI, a distance to a node and any number of other nodes with known locations (anchors) can be estimated. From this, the node can either locate itself, if its position is not known using the locations of the anchor (GPS receiver utilizing satellites as anchors), or a node can use its ToA or RSSI measurements with the other anchor nodes measurements to estimate the position of an unknown emitter. In 2D space, the distance between a node and its anchor results in a circle (in 3D a sphere) [5][13]. A node must exist along the circle, where the radius of the circle is equal to the distance between the node and the anchor. When there are multiple distance measurements given, such as three or more measurements in 2D space, and four or more in 3D space, the circles will intersect, and the desired location can be estimated. This method is illustrated in Fig. (1.2). This methodology is seen in the PDOA technique, which utilizes the RSSI between sensor node pairs to calculate the power-difference between these pairs. This power-difference will be shown to geometrically represent the trilateration technique needed to solve the geolocation problem. ToA is not explored as the geolocation problem applied in this research involves a transmitter that is not assumed to be synchronized with the nodes in the sensor network. The trilateration technique consists of a minimum three sensors *A*, *B*, and *C*. The anchor node *S* is at a point (x_0, y_0). The distance between a particular sensor i and *S* is given in (1.1). By squaring this, the circles in Fig. (1.2) are generated and their intersections are determined.

$$d_i = \sqrt{(x_i - x_s)^2 + (y_i - y_s)^2} \quad (1.1)$$

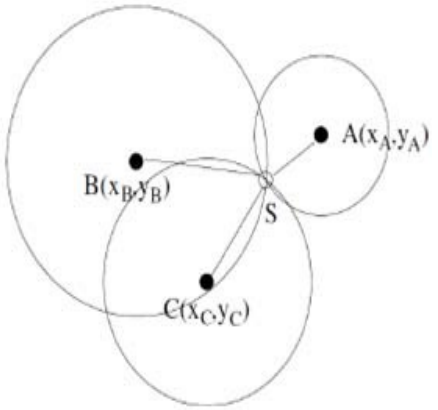


Figure 1.2: The trilateration technique, which the distances between a node and its anchors results in overlapping circles with common intersections

The value d_i can be related to an RSSI using a particular path loss model. The distance can also be related to the time-of-arrival, assuming the anchor node and base stations are synchronized. Further use of this technique applied too this research is outlined in Chapter 2, when we utilize the power-difference between sensor measurements.

The multilateration technique makes use of multiple receivers that are synchronized, with known position. With synchronized receivers, it is possible to determine the time-difference of arrival between each pair of anchor nodes. With this, the geometry generated by the TDOA measurements become hyperbolae instead of circles. Maximum likelihood can be used to solve for the solution to this set of equations. This method is explored further in Chapter 2 when we introduce TDOA in conjunction with the PDOA trilateration technique in this research. The illustration of multilateration is shown in Fig. (1.3). The general method is using a total of N receivers, and computing the time-difference between receivers 2 to N and receiver 1 as follows:

$$ct_{ij} = \sqrt{(x_i - x_0)^2 + (y_i - y_0)^2} - \sqrt{(x_j - x_0)^2 + (y_j - y_0)^2} \quad (1.2)$$

where t_{ij} is the time difference between nodes i and j and (x_0, y_0) is the location of the emitter

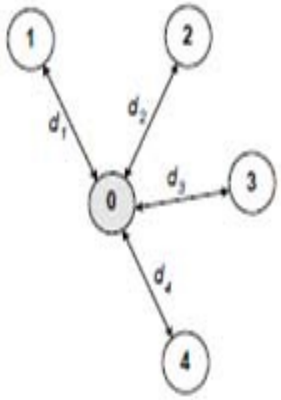


Figure 1.3: The multilateration technique, which the time-difference between nodes is used to estimate the position of an emitter

For the application in this research, we assume a network of *N* nodes and an unknown emitter located in an arbitrary location as shown in Fig. (1.4) with unknown distances between a particular sensor i and the emitter as d_i. A-priori information is important in designing the network. Some known information may be the general direction of the transmitter; For example, in a coastal monitoring system in which we track ship radar, multilateration based TDOA will have two differing locations, one on one side of the network and another on the other side, where it can be assumed that the direction of the transmitter is off the coast and not inland. Other a-priori information known will be the type of waveform being sent, we know which channel and modulation the waveform is operating on. We know the position of each of the nodes in the network, as they have embedded GPS within their sensors. In addition, the sensors will also be able to measure RSSI, and each sensor has similar noise power-spectral density (PSD). The timing between the sensors is also synchronized, which will allow for the computation of accurate time differences between nodes.

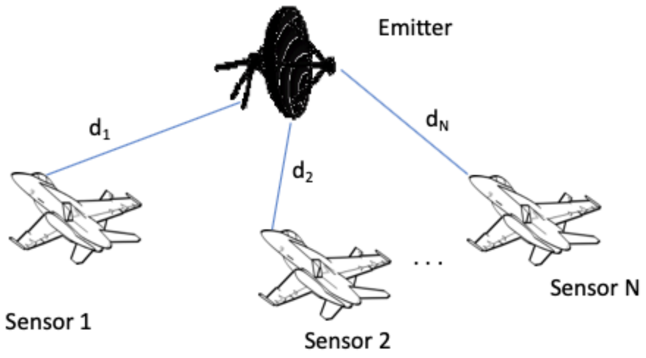


Figure 1.4: An abstraction of a geolocation sensor network. Quantities shown in red are not known and must be estimated

Chapter 2: Geolocation Algorithms

2.1 Power Difference of Arrival

In a network scenario where the DSA assets are limited, it is helpful to optimize the limited information available to obtain a good estimate. For example, a network which only has radios that can detect a received signal's power level or RSS (received signal strength), but require the ability to geolocate the source of the signal will find the power difference of arrival (PDOA) algorithm to be applicable.

We consider a number of transceivers distributed spatially over some geographic area as shown in Fig. (2.1), referred to as the area of operation (AO). The measurements gathered from the sensors include (x_i, y_i) and received power P_i. There exists a fusion node that pulls data from the RF sensors and triangulates the estimate position (x_T, y_T) of the emitter that transmits at power level P_T.

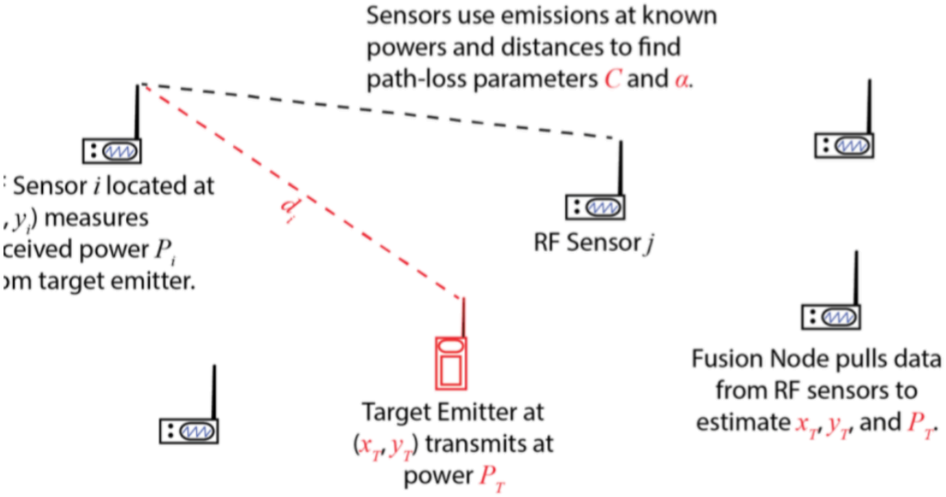

Sensors use emissions at known powers and distances to find path-loss parameters C and α.
: Sensor i located at
$, y_i)$ measures
ceived power P_i
om target emitter.
d_i
RF Sensor j
Fusion Node pulls data from RF sensors to estimate x_T, y_T, and P_T.
Target Emitter at (x_T, y_T) transmits at power P_T

Figure 2.1: An abstraction of a geolocation sensor network. Quantities shown in red are not known and must be estimated

The path loss model can be modeled as an exponential function of the distance d_i between the target emitter and the receiving sensor, where the received signal power is proportional to $d_i^{-\alpha}$, where α is path loss exponent that depends on the RF environment. The parameter α ranges between 2 and 4, where 2 represents free-space and 4 is for more lossy environments. This exponential model is known as the log-distance path loss model which is defined in (2.1). The constant $\hat{C}$ is known as the normalization constant. This constant accounts for system losses, such as transmitter and receiver gains. The constant $\hat{C}$ is also unknown and generally varies in a particular RF environment. The path loss exponent and the normalization constant can be estimated by the sensors performing an initial test by measuring the received and transmitted powers between a known emitter, such as one of the radios, in order to accurately estimate these parameters.

$$P_i = \hat{C} - 10\alpha \log(d_i) + P_T \quad (2.1)$$

Given the metrics and the chosen path-loss model, the power-difference between sensors *i* and *j* is defined in (2.2).

$$P_i - P_j = 10\alpha \log\left(\frac{d_j}{d_i}\right) \quad (2.2)$$

From this result, the power difference of arrival (PDOA) algorithm from [1] is described as follows. The equation in (2.2) is then rearranged to obtain the distance ratio in (2.3)

$$q_{ij} = \frac{d_i}{d_j} = 10^{-\frac{P_i - P_j}{10\alpha}} \quad (2.3)$$

Since the transmitter of interest is at a point (x, y), the distance of a particular sensor to the emitter is given by (2.4)

$$d_i^2 = (x - x_i)^2 + (y - y_i)^2 \quad (2.4)$$

The relation in (4) represents a circle with a center (x_i y_i). Using (2.3) and (2.4), we obtain the ratio of two circle equations as shown in (2.5)

$$\frac{(x-x_i)^2+(y-y_i)^2}{(x-x_j)^2+(y-y_j)^2} = q_{ij}^2 \qquad (2.5)$$

The equation (2.5) is expanded by completing the square to obtain another equation of a circle with the center and radius defined in (2.6) and (2.7) where $\boldsymbol{C}_{ij}$ is the center and R_{ij} is the radius. These equations are written in vector form for error analysis later in Chapter 3.

$$\boldsymbol{C}_{ij} = \frac{q_{ij}^2\boldsymbol{x}_j-\boldsymbol{x}_i}{q_{ij}^2-1} \qquad (2.6)$$

$$R_{ij} = q_{ij}\frac{|\boldsymbol{x}_j-\boldsymbol{x}_i|}{|q_{ij}^2-1|} \qquad (2.7)$$

The power-difference between *N* sensors represents a series of circles. This results in a maximum of $N(N-1)$ circle intersections. In the event there is no exact intersection between a particular pair of circles, the closest midpoint between the two circles is determined as an intersection. Once all of the intersections have been computed, a proposed grid-density search algorithm is applied as described in [1].

For the grid density algorithm, all of the intersection points are used to create an area of operation (AO) by using the minimum and maximum x and y values from the set of intersection points. That is the geolocation dataspace is denoted by: $L = \{(x_e, y_e)\}$ e = 1…max(# of intersection points). The grid is partitioned into m x n grid cells of equal size. From there, the grid with the most intersections is chosen, and all intersection points within that grid cell are averaged and the result is the geolocation. This algorithm is visualized in Fig. (2.2). In that case, three sensors result in three different power difference measurements and three circles. The grid algorithm in [1] is recursive and further divides each grid into 4x4 cells until a specific grid resolution is obtained. This is more computationally tedious. A modified less computationally complex method involves dividing the grid into equally spaced 4x4 grid cells where the cell with the most intersections is applied. In the event of a tie between two or more cells, the intersections in the adjacent cells are counted to break the tie. The cell with more intersection points in the adjacent cell will win the tie. In the unlikely even there is still a tie, the average of all intersections is chosen as the estimated position as a crude approximation.

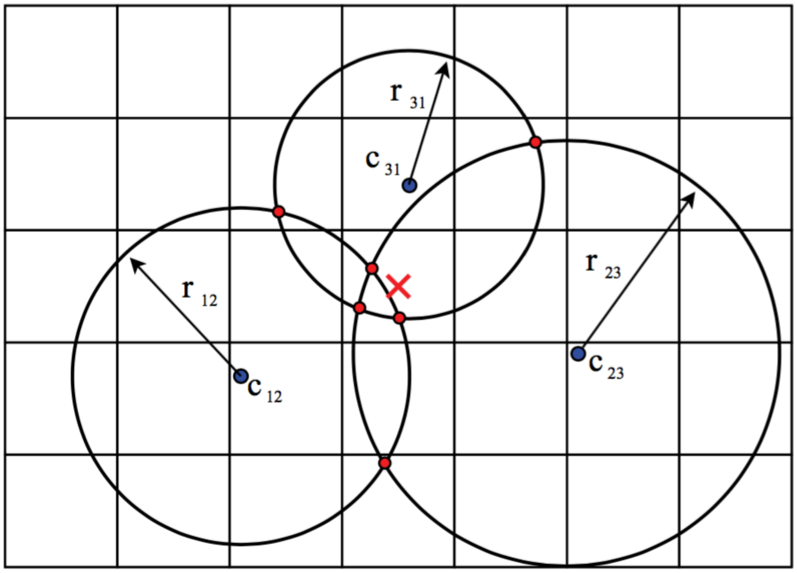


Figure 2.2: A visualization of the intersection grid density method

The grid intersection method allows for ruling out intersections which are outliers, that would otherwise affect the accuracy of emitter location estimation if we simply find the midpoint of the intersection. As Fig. (1.2) shows, most of the intersection points tend to be centered around the true emitter position. This method of geolocation is referred to as triangulation, as the power-difference measurements between receivers form a constraint along the set of points in a circle. This circle geometry of power-difference measurements is referred to as the circle of Apollonius [2].

In chapter 3, further analysis is done on the PDOA algorithm to determine the error of the estimation when subject to perturbations, such as noise and instrumentation errors.

2.2 Time Difference of Arrival

The TDOA algorithm locates an emitter source using the intersection of hyperbolic curves generated by cross-correlating IQ data from sensors. Unlike PDOA, which uses RSSI, the TDOA algorithm collects IQ samples from sensors and cross-correlates the IQ data for each pair of sensors to determine the time-difference between the arrival of the emitter signal at each sensor pair. The technique used in this research is an approximation of the maximum likelihood (ML) estimator described in [3]. Applications of the TDOA algorithm are beneficial for environments with high-noise and high-bandwidth emitters, such as radar.

The time-difference between the emitter and two sensors will generate a hyperbola and a third sensor will generate another hyperbola. The intersection between the hyperbolae is used as the estimated emitter position [3]. Using a network architecture similar to the one depicted in Fig. (3.1), the distance between the sensor and actual emitter is $r_i^2 = (x_i – x)^2 + (y_i – y)^2 = K_i – 2x_ix – 2yyi + x^2 + y^2$, for all i = 1, 2, … M, where

$$K_i = x_i^2 + y_i^2 \qquad (2.8)$$

If c is the signal propagation speed (assumed to be equal to the speed of light) and one of the sensors is selected as a reference sensor (sensor 1) with coordinates (x_1, y_1) and $d_{i,1}$ is the time-difference between sensor i and the reference sensor, then

$$r_{i,1} = cd_{i,1} = r_i - r_1 \qquad (2.9)$$

For the case of three sensors, a closed-form solution exists. With three sensors, x and y can be solved in terms of r_1 in (2.10) as follows:

$$\begin{bmatrix} x \\ y \end{bmatrix} = -\begin{bmatrix} x_{2,1} & y_{2,1} \\ x_{3,1} & y_{3,1} \end{bmatrix}^{-1} \times \left\{ \begin{bmatrix} r_{2,1} \\ r_{3,1} \end{bmatrix} r_1 + \frac{1}{2} \begin{bmatrix} r_{2,1}^2 - K_2 + K_1 \\ r_{3,1}^2 - K_3 + K_1 \end{bmatrix} \right\} \qquad (2.10)$$

Inserting this intermediate result into (2.8) at $i = 1$ gives a quadratic in r_1. Substitution of the positive root back into (2.10) produces the solution, which is used as the emitter estimate. In the event that there is more than one positive root, the ambiguity is resolved by restricting the emitter to a specific area of interest, such as a coastal monitoring system, where the general direction of the emitter is known.

For the case of four or more sensors, the system is over-determined as the number of measurements is greater than the number of unknowns. In the presence of noise, similar to the PDOA case, set of equations will not intersect at the same point. Let $z_a = \left[z_p^T, r_1\right]^T$ be the unknown vector, where $z_p = \left[x, y\right]^T$. The solution to the system involves imposing the known relationship (2.8) to the computed result via another LS computation, which is a two-step procedure and is an approximation of the true ML estimator for emitter localization. The ML estimate of z_a is as follows,

$$z_a \approx \arg\min\left\{\left(h - G_a z_a\right)^T \varphi^{-1}\left(h - G_a z_a\right)\right\} = \left(G_a^T \varphi^{-1} G_a\right)^{-1} G_a^T \varphi^{-1} h \qquad (2.11)$$

where G, h and φ are defined as follows.

$$h = \frac{1}{2}\begin{bmatrix} r_{2,1}^2 - K_2 + K_1 \\ r_{3,1}^2 - K_3 + K_1 \\ . \\ r_{M,1}^2 - K_M + K_1 \end{bmatrix} \qquad G_a = -\begin{bmatrix} x_{2,1} & y_{2,1} & r_{2,1} \\ x_{3,1} & y_{3,1} & r_{3,1} \\ . & . & . \\ x_{M,1} & y_{M,1} & r_{M,1} \end{bmatrix} \qquad \varphi = h - G_a z_a^0 \qquad (2.12)$$

The expression in (2.11) is the generalized least-squares solution of (2.12). For this research, the source is assumed to be far-away so an approximation of (2.11) is found and expressed in (2.13), the explanation is described in [3].

$$z_a = (G_a^T Q^{-1} G_a)^{-1} G_a^T Q^{-1} h \qquad (2.13)$$

The elements of z_a can be expressed as follows, where e_1, e_2, and e_3 are the estimation errors of z_a

$$z_{a,1} = x^0 + e_1, z_{a,2} = y^0 + e_2, z_{a,3} = r_1^0 + e_3 \qquad (2.14)$$

Subtracting the first two components by x_1 and y_1 and then squaring the elements leads to another set of equations.

$$\varphi' = h' \text{-} G_a' z_a' \qquad (2.15)$$

where h', Ga' and za' are defined as follows.

$$h' = \begin{bmatrix} \left(z_{a,1} \text{-} x_1\right)^2 \\ \left(z_{a,2} \text{-} y_1\right)^2 \\ z_{a,3}^2 \end{bmatrix}, G_a' = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 1 & 1 \end{bmatrix}, z_a' = \begin{bmatrix} (x \text{-} x_1)^2 \\ \left(y \text{-} y_1\right)^2 \end{bmatrix} \qquad (2.16)$$

The overall solution and position estimate is obtained from $z_a{}'$ and is defined as follows. The correct solution is the solution that lies within the particular area of interest.

$$z_p = -\sqrt{z_a^{'}} + \begin{bmatrix} x_1 \\ y_1 \end{bmatrix} \text{ or } z_p = \sqrt{z_a^{'}} + \begin{bmatrix} x_1 \\ y_1 \end{bmatrix} \qquad (2.17)$$

where,

$$z_a' \approx (G_a'^T B'^{-1} G_a^T Q^{-1} G_a B'^{-1} G_a')^{-1} (G_a'^T B'^{-1} G_a^T Q^{-1} G_a B'^{-1}) h' \quad (2.18)$$

where,

$$B' = diag(x^0 - x_1,\ y^0 - y_1, r_1^0) \qquad (2.19)$$

For B’, x^0 and y^0 can be approximated by the values found in (2.13).

2.3 Other Techniques in Literature

We will explore other techniques that use PDOA and TDOA estimates in literature, and why they were not used in this research. For PDOA, other techniques are outlined in [14] such as Non-linear least-squares and linear least squares method. For non-linear least squares, a function Q(x, y) is determined as follows:

$$Q(x, y) = \sum_{k<l} \left[P_{kl} - 5\alpha \log \left(\frac{(x - x_l)^2 + (y - y_l)^2}{(x - x_k)^2 + (y - y_k)^2} \right) \right]^2 \quad (2.20)$$

A grid is defined and each point along the grid is plugged into this function until a minimum is determined. This method can be very computationally expensive and there is a tradeoff in the resolution between each point to plug into Q(x, y), and there may be multiple minimum values, which may lead to ambiguity.

The non-linear least squares method can be linearized as follows: Given the equation for power-difference between sensor k and l:

$$P_{kl} = 5\alpha \log\left(\frac{(x-x_l)^2+(y-y_l)^2}{(x-x_k)^2+(y-y_k)^2}\right), 1 \leq k < l \leq N \quad (2.21)$$

A constant is defined as follows:

$$\beta_{kl} = 10^{\frac{P_{kl}}{5\alpha}} \quad (2.22)$$

Therefore, (2.21) can be rewritten as:

$$(x - x_l)^2 + (y - y_l)^2 = \beta_{kl}[(x - x_k)^2 + (y - y_k)^2] \quad (2.23)$$

or,

$$(1 - \beta_{kl})c - 2(x_l - \beta_{kl}x_k)x - 2(y_l - \beta_{kl}y_k)y = w_{kl} \quad 1 \leq k < l \leq N \quad (2.24)$$

where $w_{kl} = \beta_{kl}r_k^2 - r_1^2$ and $c = (x^2+y^2)$ is introduced and is treated as independent of x and y and $r_l^2 = x_l^2+y_l^2$ and $r_k^2=x_k^2+y_k^2$. Thus x and y can be solved as:

$$(c, x, y)^T = (A^T A)^{-1}A^T b \quad (2.25)$$

This method wasn't utilized due to its limited accuracy with a small number of sensors.

For TDOA, other methods to solve the time-difference equations also exist. One common technique is the Taylor-Series method [15]. The equations of time-difference are linearized and the following equation (2.26) is solved iteratively.

$$\begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} = (G_t^T Q^{-1} G_t)^{-1} G_1^T Q^{-1} \boldsymbol{h_t} \quad (2.26)$$

To solve this, we choose an initial guess (x_0, y_0) and add the result of (2.26) to the initial estimate until Δx and Δy converge. While this algorithm is accurate in even with high noise variance, the drawback here is the computational complexity, and the dependency on the initial guess. If the initial guess is far away from the actual solution, it may take a while to converge. Also, there may be cases where it doesn't even converge.

Chapter 3: Geolocation Error Analysis

3.1 Overview and Model Geometry

In this chapter, we further evaluate the performance of the PDOA and TDOA algorithm by using mathematical analysis to determine the constraints on the accuracy under particular RF environment conditions. From Chapter 2, the PDOA algorithm uses the RSSI of sensors and takes the difference between pairs of sensors, where TDOA utilizes difference between the time-of-arrival of a signal at two different receivers.

The power-difference is used to generate Apollonian circles and then determine the total number of intersections and perform a grid-search to determine the most likely location where the emitter lies. This process is non-linear, and has significant short-comings regarding accuracy and noise amplification. Environmental effects such as multipath and fading break the assumption of an invertible function relating the received power to the distance between the emitter and sensor.

We consider the network sensor situation shown in Fig. (3.1). A group of sensors are used to determine the location of a transmitter of unknown power at an unknown location $\mathbf{x}_{\text{target}}$. Sensor *I* is located at x_{i} and there exists a point in space x_{cm} which is much like the center of mass (average of sensor locations), but this point need not be a center of mass exactly, but the following relation should hold:

$$|\boldsymbol{D}| \gg |\boldsymbol{d}_i|, \forall i \qquad (3.1)$$

This relation states the condition that the target emitter is located far away from the sensor area of operation (AO). The distance between the emitter and the reference point x_{cm} is much larger than the distance between a sensor and x_{cm}.

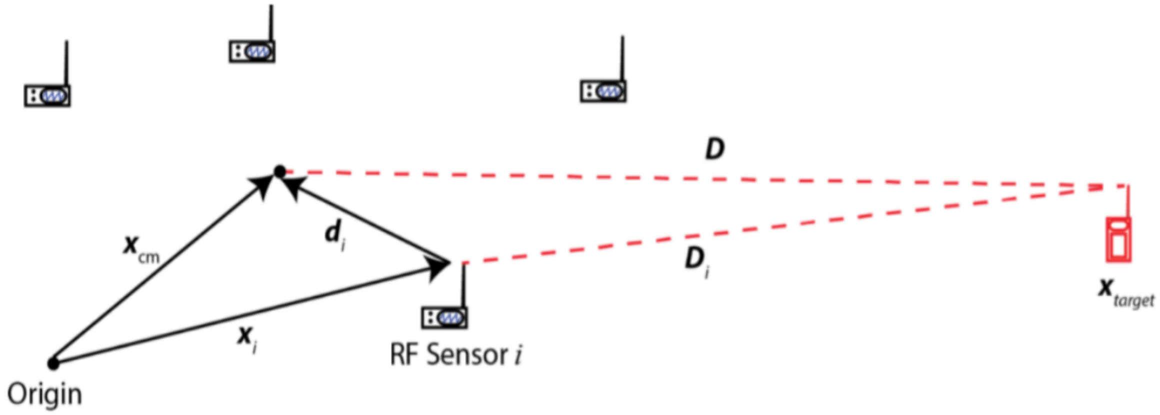


Figure 3.1: A diagram of model geometry

3.2 PDOA Error Analysis

For the error analysis, we apply a Gaussian white noise term to the vector represented by the location of sensor I or x_i as in (3.2), where $\boldsymbol{\varepsilon_i}$ is the Gaussian noise term added to the vector x_i. This noise term represents measurement error accumulated through PDOA. Placing the noise term here allows for more feasible theoretical error analysis of important PDOA parameters.

$$\widetilde{\boldsymbol{x}}_i = \boldsymbol{x}_i + \varepsilon_i \qquad (3.2)$$

We define a perturbation parameter in (3.3) to quantify how large these errors become.

$$\gamma = \frac{max_i|\boldsymbol{d}_i|}{|\boldsymbol{D}|} \ll 1 \qquad (3.3)$$

The PDOA algorithm involves the determination of Apollonian circles for each pair of sensors. On a 2D plane, the target lies somewhere on the circle (with center $\mathbf{C}_{ij}$ and R_{ij}) of every pair of sensors. If there is no measurement error ($|\boldsymbol{\varepsilon}_\mathbf{i}| = 0$ for all i), then the fusion center’s estimates of the circle radii and center would be correct; however, the errors yield estimated parameters $\widetilde{\boldsymbol{C}}_{ij}$ and $\tilde{R}_{ij}$ that are corrupted by noise. The equation of the circle centers and radii defined by the algorithm were shown in (2.6) and (2.7). We now plug into these equations $\widetilde{\boldsymbol{x}}_i$ and $\widetilde{\boldsymbol{x}}_j$, the location of sensors *I* and *J* into the equations to obtain the resulting equations in (3.4) and (3.5)

$$\tilde{\mathbf{C}}_{ij} = \frac{q_{ij}^2 \tilde{\mathbf{x}}_j - \tilde{\mathbf{x}}_i}{q_{ij}^2 - 1} = \mathbf{C}_{ij} + \frac{q_{ij}^2 \boldsymbol{\varepsilon}_j - \boldsymbol{\varepsilon}_i}{q_{ij}^2 - 1} \tag{3.4}$$

$$\begin{gathered} \tilde{R}_{ij}^2 = \left| \frac{q_{ij}}{q_{ij}^2 - 1} \right|^2 \left| \tilde{\mathbf{x}}_j - \tilde{\mathbf{x}}_i \right|^2 = \\ R_{ij}^2 + \left| \frac{q_{ij}}{q_{ij}^2 - 1} \right|^2 \left(\left| \boldsymbol{\varepsilon}_j - \boldsymbol{\varepsilon}_i \right|^2 + 2 \left(\mathbf{x}_j - \mathbf{x}_i \right) \cdot \left(\boldsymbol{\varepsilon}_j - \boldsymbol{\varepsilon}_i \right) \right) \end{gathered} \tag{3.5}$$

Let us express the error terms on the right-hand side of (3.4) and (3.5) in terms of the perturbation parameter. We do this by expressing the distance ratio for a pair of sensors by explicitly separating out the part that is of order $\boldsymbol{\gamma^2}$. Letting $\boldsymbol{d_i} = d_i \widehat{\boldsymbol{d}}_i$ and $\boldsymbol{D} = D\widehat{\boldsymbol{D}}$ for unit-vectors $\widehat{\boldsymbol{d}}_{\boldsymbol{i}}$ and $\widehat{\boldsymbol{D}}$, we have,

$$\frac{|\mathbf{D}_i|}{|\mathbf{D}_j|} = \frac{|\mathbf{D} + \mathbf{d}_i|}{|\mathbf{D} + \mathbf{d}_j|} = 1 + \frac{d_i}{D} \hat{\mathbf{D}} \cdot \hat{\mathbf{d}}_i - \frac{d_j}{D} \hat{\mathbf{D}} \cdot \hat{\mathbf{d}}_j + O[\gamma^2] \tag{3.6}$$

Putting this simplification into our expression for the distance ratio q_{ij} from (2.3), we have,

$$\begin{aligned} q_{ij} &= 1 + \frac{d_i}{D} \hat{\mathbf{D}} \cdot \hat{\mathbf{d}}_i - \frac{d_j}{D} \hat{\mathbf{D}} \cdot \hat{\mathbf{d}}_j + O[\gamma^2] \\ q_{ij}^2 &= 1 + 2 \left(\frac{d_i}{D} \hat{\mathbf{D}} \cdot \hat{\mathbf{d}}_i - \frac{d_j}{D} \hat{\mathbf{D}} \cdot \hat{\mathbf{d}}_j \right) + O[\gamma^2] \\ &= 1 + H + O[\gamma^2] \end{aligned} \tag{3.7}$$

where $H = H(\boldsymbol{D}, d_i, d_j)$ is a function of the distances relative to the reference point x_{cm}. Note that $|H| << 1$ as a consequence of (3.1). In terms of these new variables, the circle parameter estimates in (3.4) and (3.5) can be rewritten as follows.

$$\tilde{\mathbf{C}}_{ij} = \mathbf{C}_{ij} + \boldsymbol{\varepsilon}_j + \frac{1}{H}(\boldsymbol{\varepsilon}_j - \boldsymbol{\varepsilon}_i) \tag{3.8}$$

$$\tilde{R}_{ij}^2 = R_{ij}^2 + \left(\frac{1}{H^2} + \frac{1}{H}\right)\left(\left|\boldsymbol{\varepsilon}_j - \boldsymbol{\varepsilon}_i\right|^2 + 2(\mathbf{x}_j - \mathbf{x}_i) \cdot (\boldsymbol{\varepsilon}_j - \boldsymbol{\varepsilon}_i)\right) \tag{3.9}$$

From (3.8) and (3.9), the terms such as $1/H$ and $1/H^2$ dominate. So what should be expected is for the error for most of the circle terms to increase with respect to the change in H.

Given the expression for the circle center in (3.8), we can rearrange the equation by taking the magnitude of both sides, since the error is associated with the magnitude (distance from the actual circle center to the estimated circle center). We then take the expected value of both sides so that we obtain a linear equation shown in (3.10), which is dependent on the expected value of ε_i and ε_j with a slope that is dependent on H, which varies with the distance of the emitter from the sensor network. Thus, we can show that the variance of the error will increase by a factor inversely proportional to H^2.

$$E[|\tilde{\mathbf{C}}_{ij} - \mathbf{C}_{ij}|] = \frac{1}{H}E[(|(H+1)\boldsymbol{\varepsilon}_j - \boldsymbol{\varepsilon}_i|] \tag{3.10}$$

Similarly, the expected value of the circle radius error is shown in (3.11).

$$E[\tilde{R}_{ij}^2 - R_{ij}^2] = \left(\frac{1}{H^2} + \frac{1}{H}\right) E\left[\left|\boldsymbol{\varepsilon}_j - \boldsymbol{\varepsilon}_i\right|^2 + 2\left(\mathbf{x}_j - \mathbf{x}_i\right) \cdot \left(\boldsymbol{\varepsilon}_j - \boldsymbol{\varepsilon}_i\right)\right] \tag{3.11}$$

Rearranging the equations into this form allows us to verify the relations using a sufficiently large sample size for a simulation to calculate the expected value and variance of error.

To find the intersection between circles, we use a coordinate independent system by letting two circles be represented by $|\boldsymbol{X}\text{-}C_i|^2 = R_i^2$ for i = 0, 1 where i represents a particular circle. We define $U = C_1\text{-}C_0$ and $V = U^{\perp}$. The intersection points can be written in the form as shown in (3.12).

$$\boldsymbol{X} = \boldsymbol{C}_0 + s\boldsymbol{U} + t\boldsymbol{V} \tag{3.12}$$

The equation for s and t are shown in (3.13) and (3.14).

$$s = \frac{1}{2}\left(\frac{R_0^2 - R_1^2}{|\boldsymbol{U}|^2} + 1\right) \tag{3.13}$$

$$t^2 = \frac{R_0^2}{|\boldsymbol{U}|^2} - s^2 \tag{3.14}$$

Expanding these equations to obtain the first-order term plus the error is difficult, so rather than using the same method for the center and radii, we inspect the terms according to their orders of magnitude. Looking at s, both the numerator and denominator are on the order of $1/\mathrm{H}^2$, which means s is close to unity. Similarly, t is also close to unity. From (3.12) it is clear that the error in **X** is on the order of 1/H, so we should expect to see the expected value of error increase linearly.

3.3 TDOA Error Analysis

The error analysis for TDOA is conducted along a similar line to the PDOA error analysis, by adding white Gaussian noise to the parameters needed in the TDOA algorithm. In this case, we add the noise to the TDOA measurements. We then determine the effects on how the size of the variance in noise increases the error of the TDOA measurement, and for the overall estimator. We analyze the error for both the three-sensor case and cases with four or more sensors. The error performance is then compared with the theoretical MSE of the TDOA algorithm to compare the empirical results to the expected algorithm performance.

The perturbation analysis used to estimate the theoretical error in PDOA is not necessary with this TDOA algorithm, as this is an approximation of the maximum likelihood estimator. To calculate this theoretical MSE, we first determine the covariance of the position estimate, z_p calculated in (2.17). The method for determining this is laid out in [3]. First, the solution is expressed in the form $x = x^0 + e_x$ and $y = y^0 + e_y$. as a result of the definition of z_a' in (2.16), it follows that:

$$z'_{a,1} - (x^0 - x_1)^2 = 2(x - x_1)e_x + e_x^2 \qquad (3.15)$$

$$z'_{a,2} - (y^0 - y_1)^2 = 2(y - y)e_y + e_y^2 \qquad (3.16)$$

From (4.1) and (4.2), the errors e_x and e_y are relatively small compared to x^0 and y^0. Therefore, these can be ignored.

Thus, the covariance matrix of $\mathbf{z}_p$ is defined as follows:

$$\mathbf{\Phi} = cov(\boldsymbol{z_p}) = \frac{1}{4}B''^{-1}cov(\boldsymbol{z'_a})B''^{-1} \quad (3.17)$$

where,

$$B'' = \begin{bmatrix} (x^0 - x_1) & 0 \\ 0 & (y^0 - y_1) \end{bmatrix} \qquad (3.18)$$

And the covariance matrix of **z_a’** is found by taking the expectation of **z_a’** and **z_a’z_a**’^T as follows:

$$cov(\boldsymbol{z}_{\boldsymbol{a}}') = (G_a'^T \Psi'^{-1} G_a')^{-1} \quad (3.19)$$

The theoretical MSE is found by summing the elements in the diagonal of **Φ**. This theoretical MSE is compared with the actual algorithm result.

Chapter 4: Error Performance Simulation and Hybrid Geolocation Method

4.1 PDOA Error Simulation Results

To verify the analytical results of the PDOA error analysis, MATLAB™ was used to simulate the error for the circle centers, radii, intersections, and overall error in target estimation as the emitter is moved further outside the area of sensor operation. The locations of the sensors were arbitrarily chosen and the center of mas of the sensors was chosen as the reference point for the model geometry. At each emitter location, the ratio of the distance of the emitter and the furthest sensor from the center of mas was taken and at each of these points, the PDOA algorithm was performed at a given emitter location for *n* = 1000 iterations with ε_i and ε_j changing each time. Of the data (center and radii) generated by each iteration, the expected value and variance of the result is taken. The variance in terms of ε_i and ε_j , ranges from 1 to 10. Three locations of the emitter were chosen at 10, 20 and 30 kilometers from the network reference point respectively.

The corresponding figures show the results of the expected value of error of circle radii and centers as well as intersections and overall location estimation error over the entire emitter path moving further from the sensor area of operation. In addition to the plot of experimental error, the slope of each line was calculated according to the formula for H. The equation for H can be rewritten as follows.

$$H = 2\left(\frac{d_i}{D}\hat{\mathbf{D}}\cdot\hat{\mathbf{d}}_i - \frac{d_j}{D}\hat{\mathbf{D}}\cdot\hat{\mathbf{d}}_j\right) \tag{4.1}$$

From (4.1), it follows that H is dependent on the distance of the emitter from the sensor network, where the sensors distance from the reference point are fixed. The values of H were calculated for each of the three distances and the slopes were calculated from (3.10) and (3.11) to give the theoretical slope.

The plots shown in Fig. (4.1) illustrate the slope of the line with theoretical data (solid line) compared to that of the experimental data (box-plot). From Fig. (4.1a) and Fig. (4.1b), the experimental mean or expected value of error for both the center and radii of an Apollonian circle generated by a pair of sensors is close to that of the theoretical expected value using the equation derived from the previous section. This shows that the simulated graphical results and the analytical equations derived are consistent with each other. Similarly, it is shown from the box-plot that an increase in expected value of error will also correspond to an increase in the variance of the error of the Apollonian circle terms.

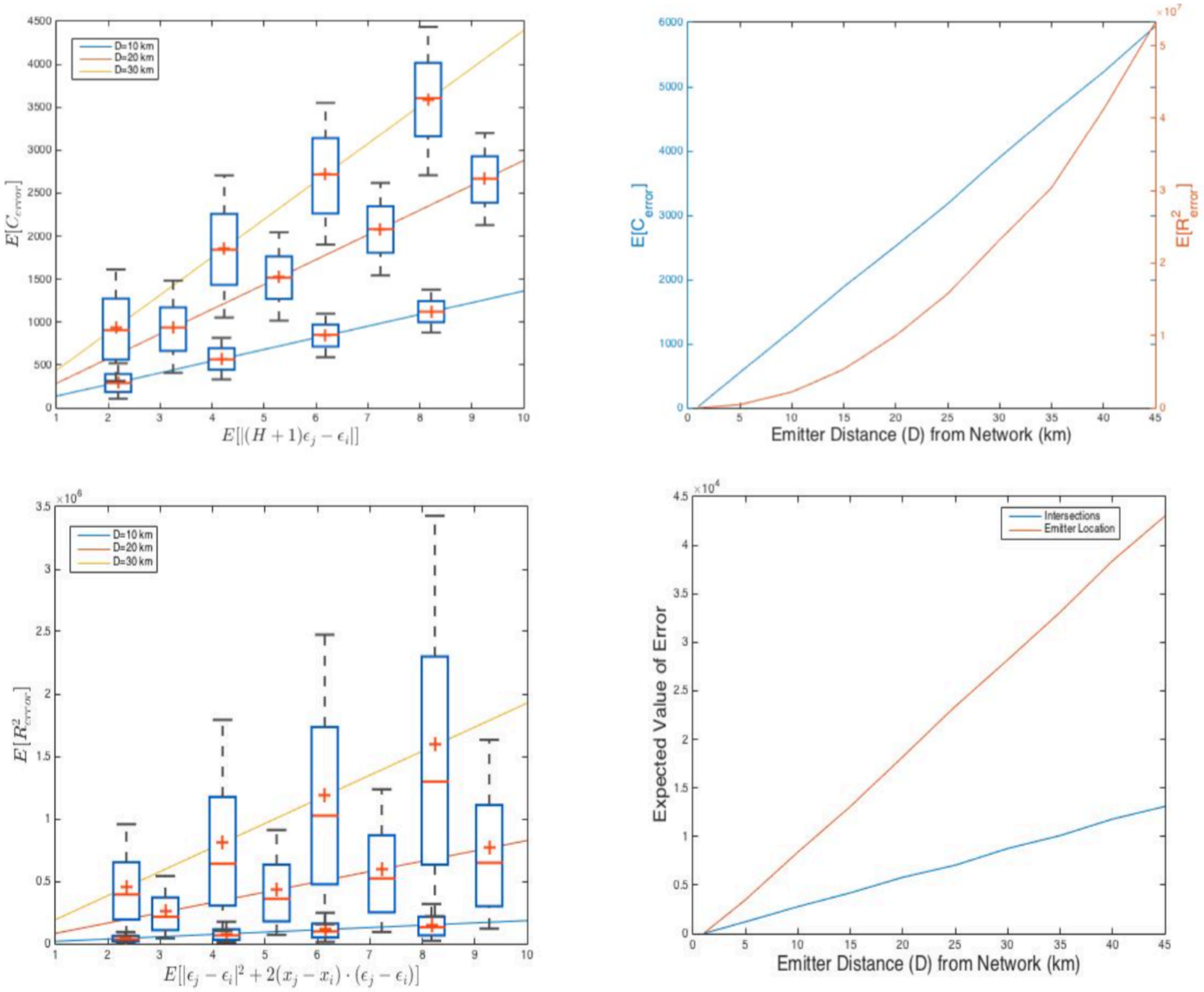


Fig. 4.1 — (a) Expected value of circle center error as a function of expected value of sensor noise. (b) Expected value of circle radii error as a function of expected value of sensor noise. (c) Expected value of error in Apollonian circle centers and radii as a function of distance of the emitter from the sensor network (d) Expected value of overall location error as a function of distance from the network

In addition to verifying the analytical Apollonian circle perturbation equations, the effect of distance of the emitter from the sensor network on the accuracy of the algorithm was also simulated. A range of emitter distances was chosen ranging from 1 kilometer to 45 kilometers away from the reference point. At each distance, the average error over *n* = 1000 iterations of PDOA algorithm was calculated. At each distance for circle centers, radii, and intersections. The overall expected error of the estimated emitter location was also computed to demonstrate that errors in the

Apollonian circle parameters accumulate, leading to an overall effect on the accuracy of the algorithm. Fig. (4.1c) shows the expected value of error for circle radii and centers. As expected, since the error in the center increases on the order of 1/H, the relationship should be linear and for the radii, the relationship should be quadratic as the error increases on the order of $1/H^2$.

The plot in Fig. (4.1d) shows the expected error for circle intersections and the overall location estimation. While it is difficult to mathematically analyze the error in overall location like what was done for the center and radii, it is expected to be linear since it should correspond to error in the intersections, which also increases on the order of 1/H.

From these results, it is clear that the variance and expected value of error is amplified for all of the terms (centers, radii, and intersections) and the overall estimation error, the further the emitter is moved away from the area with respect to the center of mass of the sensors. These results support the finding of the analytical expressions of the linearity of noise amplification in all of the terms with respect to H. An important finding is that the analytical results do not support intuition that in a noisy environment, the noise should help to improve accuracy in some instances. The reason for the significant dominance of error in the algorithm arises from the noise terms being collected in the non-linear expressions for the circle centers and radii.

4.2 TDOA Error Simulation Results

The error analysis test is conducted similar to PDOA, by running the algorithm with n=1000 iterations. The MSE is computed by taking the expected value of the distance between the estimated emitter position vs. the actual emitter position and then squaring it to obtain the MSE. This MSE is averaged over all of the iterations. The experiment utilizes 3 to 5 nodes designated as sensors and the MSE is calculated as a function of three different control variables: noise variance, distance from the network AOI, and the number of active sensors. For the TDOA algorithm, in determining the TDOA covariance matrix, the assumption that each receiver has similar noise power spectral densities will result in diagonal elements with σ_d^2 and the rest of the elements being $0.5\sigma_d^2$, where σ_d^2 is the noise power. This noise is zero mean white Gaussian noise that is added to the TDOA measurements as follows, where r_{ij}^0 is the noise-free time-difference measurement between sensors i and j:

$$r_{ij} = r_{ij}^0 + N(0, \sigma_d^2) \qquad (4.2)$$

The first error analysis test is sweeping the noise variance while holding the other control parameters (distance, number of sensors) constant. For this test, the emitter located 5km from the sensor network AOI, and 4 sensor nodes are active. The MSE as a function of the noise variance is plotted in Fig. (4.2) below. As is expected, the error does increase with the noise variance. In the presence of no variance, the error would be zero. For comparison, the dashed line shows the theoretical MSE. Also expected is the deviation in experimental MSE from the theoretical MSE since the noise variance increases, this is to be expected.

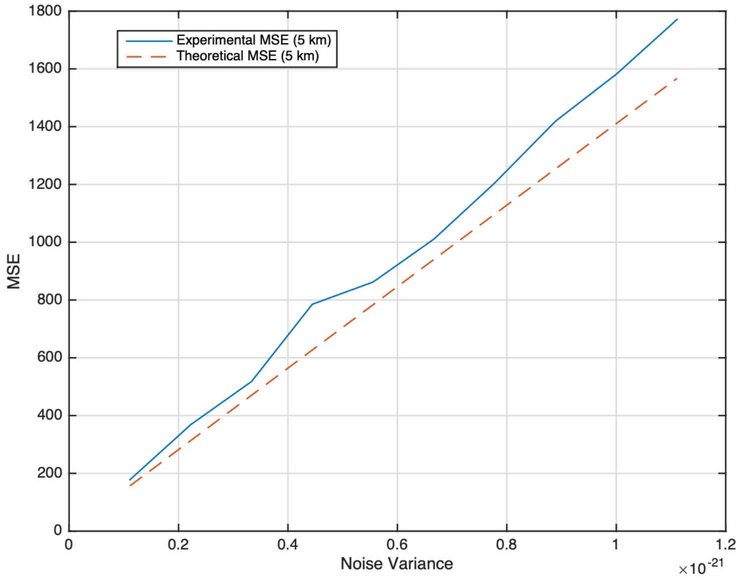


Figure 4.2: Geolocation MSE vs. noise variance for TDOA scenario with four nodes, and an emitter 5 km away from the network

The next error analysis involves sweeping through different emitter-network separation distances. The emitter starting location is inside the sensor AOI, and is then moved up to 10 km away from the sensor area. The noise variance is held constant at a value of $\sigma_d = \sqrt{0.00001/c^2}$. As expected in the plot shown in Fig. (4.3), the error does increase exponentially. This is to be expected as in any path loss environment, there is an exponential decay in the signal power due to distance alone. This results in a poor SNR, and even the small noise variance has a greater effect on the MSE of the geolocation estimation. This MSE performance demonstrates this particular version of TDOA algorithm's strength as a far-source estimator than that of PDOA. While both perform better within the sensor AOI, this algorithm clearly performs better than PDOA at larger distances from the emitter in a low-noise environment. However, when the TDOA estimates are inaccurate, this error can increase significantly. Without proper hardware or multipath environments, this low-noise is difficult to achieve, and the estimation may break down with far-source scenarios.

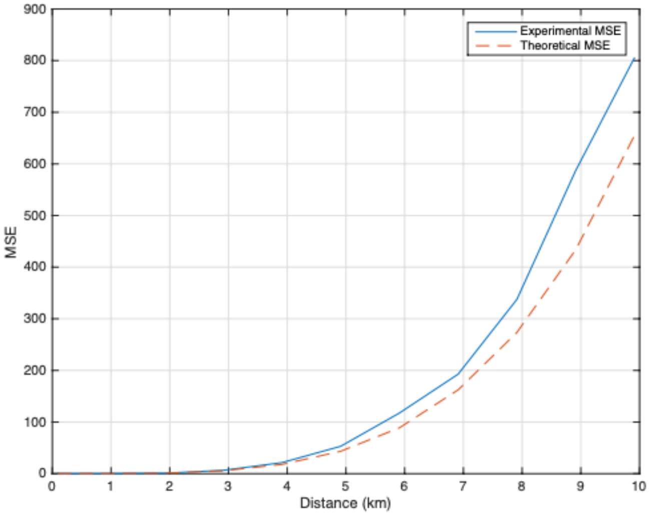
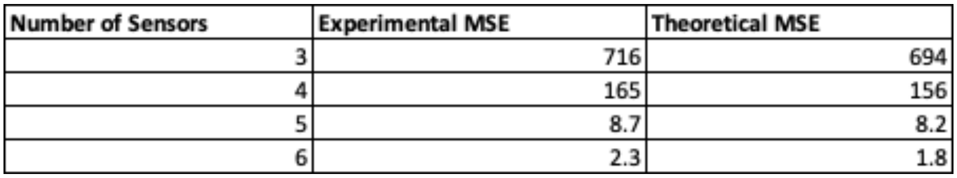


Figure 4.3: Geolocation MSE vs. distance from sensor network for TDOA scenario with four nodes, and noise variance of $\sigma_d = \sqrt{0.00001/c^2}$.

The final analysis conducted is varying the number of sensors, while keeping the variance and distance parameters constant. The noise variance is again held at $\sigma_d = \sqrt{0.00001/c^2}$, with an emitter placed 5 km away from the network. This data is recorded in tabular form in Table 4.1, and the trend, as expected, is an improvement in the MSE accuracy depending on the number of sensors, with three sensors performing the worst. The computation involving three sensors is not a good method to use, since the amount of noise variance can result in some causes of ambiguity, where there are no positive roots to the quadratic equation that results. This is not seen in the computation involving 4 or more sensors, since this method actually uses the noise variance in its calculation.

Table 4.1 – MSE for different number of sensor nodes



Number of Sensors	Experimental MSE	Theoretical MSE
3	716	694
4	165	156
5	8.7	8.2
6	2.3	1.8

4.3 Motivation for a Hybrid Method

From the previous chapters, both the PDOA and the TDOA algorithms were presented and their error performance analyzed. Each algorithm has its advantages and disadvantages. The power-difference of arrival algorithm has the advantage of hardware simplicity. The accurate synchronization between receivers is not necessary to obtain accurate RSSI values from the receivers. The algorithm also works well when the target emitter is close to the sensors, since the SNR of the emitter is higher, and the Apollonian circle intersections will be more exact. However, the PDOA algorithm has significant shortcomings. When the SNR decreases and noise variance increases, this results in greater ambiguity in the geometric intersections of the circles, resulting in error increasing linearly with distance from the sensor network's perimeter.

The TDOA algorithm has shown similar trends when the distance between the sensor network and the emitter is increased; however, in most cases the TDOA algorithm generally outperforms PDOA algorithm. With well-synchronized clocks and relatively accurate time differences, even for far-away sources, the TDOA algorithm will often outperform PDOA as seen from the simulation results. However, having well-synchronized sensors with good resolution is difficult to achieve. The software-defined radios that will be used to verify the geolocation algorithm are limited in sampling resolution and often cannot attain single nanosecond resolution, even with well-synchronized clocks. At greater distances, the noisy time-difference measurements will exacerbate the error even more than noisy PDOA measurements. In addition, multipath can have an impact on the time-difference measurements, such that if one sensor is obstructed compared to the other sensors, taking the time-difference will not cancel the effect of multipath on the time-of-arrival for the sensor. This kind of multipath is common in NLOS applications. Another necessity for accurate geolocation is the characteristics of the transmitter waveform. To properly compute a time-difference, signals with good correlation properties are necessary. Ideal characteristics of the emitter signal are outlined in [6]. Unmodulated carriers and narrowband signals are more difficult to compute TDOA due to ambiguity in the cross-correlation of the signal and its delayed version.

With these disadvantages in both algorithms, it is useful to evaluate the effectiveness of using both measurements to improve the accuracy of geolocation. The proposed hybrid geolocation method will involve obtaining measurements needed for both PDOA and TDOA and to use the estimated location of PDOA to determine the validity of the time-difference measurements and vice-versa. The following section details the proposed hybrid PDOA/TDOA geolocation algorithm.

4.4 PDOA/TDOA Hybrid Method

In other literature, utilizing both TDOA and PDOA involves setting a power threshold. In one experiment detailed in [7], the PDOA and TDOA location is computed and only one of them is used depending on whether the power is below or above a set threshold. In the mentioned experiment, if the RSSI of the furthest sensors is below a threshold, then the TDOA estimate is used. If the RSSI is above a threshold then the PDOA estimate is used. The drawback of this method is that it does not consider how noisy TDOA estimates at larger distances will significantly affect the accuracy to an even larger margin than TDOA. In this research, we apply the following algorithm as detailed s follows:

Algorithm:

Input: PDOA Estimate, Pwr Threshold, TDOA measurements, RSSI measurements

Output: Estimated emitter location

```
If RSSI > threshold
   - Apply PDOA to obtain general direction of signal
   - Apply TDOA

If RSSI < threshold
     - Apply PDOA and use location estimate to determine TDOA measurements
      - If TDOA obtained from PDOA is within +/- 40 ns of actual TDOA
         - Compute TDOA with improved TDOA estimate
      - ELSE
         - Use original TDOA estimate
   - Repeat for each TDOA measurement
   - Update TDOA noise variance vector
```

Within the sensor AOI, TDOA generally has a high noise tolerance and can estimate the emitter more accurately than the PDOA measurements subject to similar noise.

Outside the AOI, a large error in the TDOA measurement (> 10 ns error) will exacerbate the emitter position estimate error, sometimes more than PDOA. PDOA results are computed to obtain the general direction of the signal. Recall from Chapter 2 that there are two possible solutions for the final estimate of TDOA. This ambiguity is resolved in standalone TDOA by having a priori information about the general direction of the signal. The PDOA algorithm will not have this ambiguity, and comparing the two TDOA solutions to the PDOA solution can be used to filter out the extraneous TDOA solution. This hybrid method is proposed to both solve the ambiguous TDOA solution problem and to correct for the exacerbation in TDOA error estimates at larger distances due to insufficient time-resolution.

Since we sometimes use the PDOA location estimate to determine the time-differences, the resulting TDOA noise variance is not understood without evaluating it over time. To properly address the TDOA variance from the PDOA estimate, we need to determine the variance of the TDOA values associated with it. After performing each location estimate, the measured time-difference is added to an observation a vector, this vector is 10 samples long and contains the time-difference measurements used in the algorithm for a given sensor. The oldest time-difference measurements are deleted for a particular sensor after the vector exceeds 10 values. With these 10 values, we measure the variance of those values to update the time-difference noise matrix Q used in the TDOA algorithm.

The hybrid method will inevitably have some drawbacks at these larger distances in a multipath environment. In such an environment, the RSSI is not immune to multipath effects such as shadowing and fading. In this case, both estimates would be unreliable, and there is no guarantee that the range estimates from the PDOA location result would improve the location accuracy in the TDOA algorithm at a far distance.

In the following chapters, we outline the method for creating an emulation environment to verify the performance of the PDOA and TDOA simulations, and to evaluate the effectiveness of the hybrid TDOA/PDOA method. We will discuss the RF laboratory testbed layout, devices used in the testbed, and how the geolocation algorithms were implemented on the radios.

Chapter 5: Emulation Testbed Environment

5.1 Overview

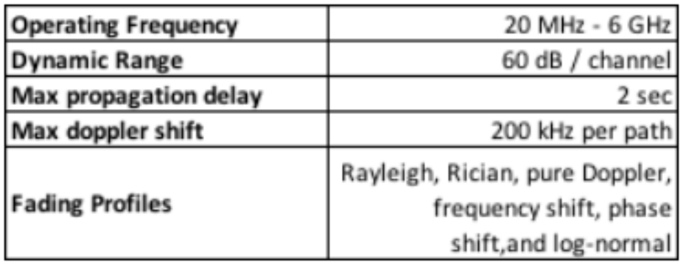
As previously shown in Chapter 2, the overall network architecture consists of at least three sensors plus a central fusion node where the sensor data collected from the radios are sent to for processing and triangulation. This was shown in Fig. (2.1). In this chapter, the laboratory testbed environment in which the proposed network will be emulated is detailed, as well as each of the components of the testbed. In any emulation environment, it is useful to observe environmental effects on the RF network in a controlled laboratory environment. Being able to control the channel properties, with realism similar to a field test, is one useful aspect of emulation over field-testing. Advantages to emulation include repeatability, realism, and cost saving. The RF wireless network environment emulator (RFWNEE) at US Naval Research Laboratory has the capability to emulate various RF radio networks, and the environment is used toward emulating and integrating the geolocation sensor system in this thesis as an extension of the MATLAB simulations shown in earlier sections. Previous applications of the RFWNEE at NRL are documented in [4].

5.2 RF Channel Emulator

Central to the RFWNEE is the channel emulator device used. The testbed is integrated with a RFNest D512 series channel emulator [18], produced by Intelligent Automation Inc (IAI). Some key specifications of the RFNest are shown in Table 5.1. The controllable channel effects in the RFNest consist of channel channel attenuation, propagation delay, and doppler shift. The emulator also supports a variety of network configurations such as SISO, MIMO, MISO, SIMO, and full mesh. The RFNest is interfaced with the RFWNEE as shown in Fig. (5.1), where the radio under test (RUT) is interfaced with the D512 through the RF daughterboard (RFDB). The RFDB converts the RF to IF bands for transmitted and received signals between the RF front end and the internal circuitry of the RFNest. The digital daughterboard (DDB) interfaces with the RFDB and the main FPGA board, and consists of high speed ADCs and DACs. Each DDB has a mid-size FPGA, which is used for signal multiplexing in order to maximize the transmission capacity between the DDB and main FPGA. The main FPGA contains the RF channel emulation engine.

The RFNest is controlled through a software module which controls scenario generation, channel modeling, and node operation among the radios. Because of the geolocation aspects of this thesis research, GPS simulators are also integrated, as depicted in Fig. (5.1). USB realtime spectrum analyzers (RSA) are also provided for spectrum monitoring and management purposes. A network of computers is integrated to execute emulations for scenarios. As shown in Fig. (5.1), a controller computer contains the software for controlling the RFNest, it is also capable of supporting virtual radios through the EMANE environment, but that capability is beyond the scope of this research. Slave servers are integrated for controlling external peripherals such as the radios or RSAs. Everything is interfaced through an ethernet switch for communication.

Table 5.1 – RFNest D512 series key specifications



Operating Frequency	20 MHz - 6 GHz
Dynamic Range	60 dB / channel
Max propagation delay	2 sec
Max doppler shift	200 kHz per path
Fading Profiles	Rayleigh, Rician, pure Doppler, frequency shift, phase shift,and log-normal

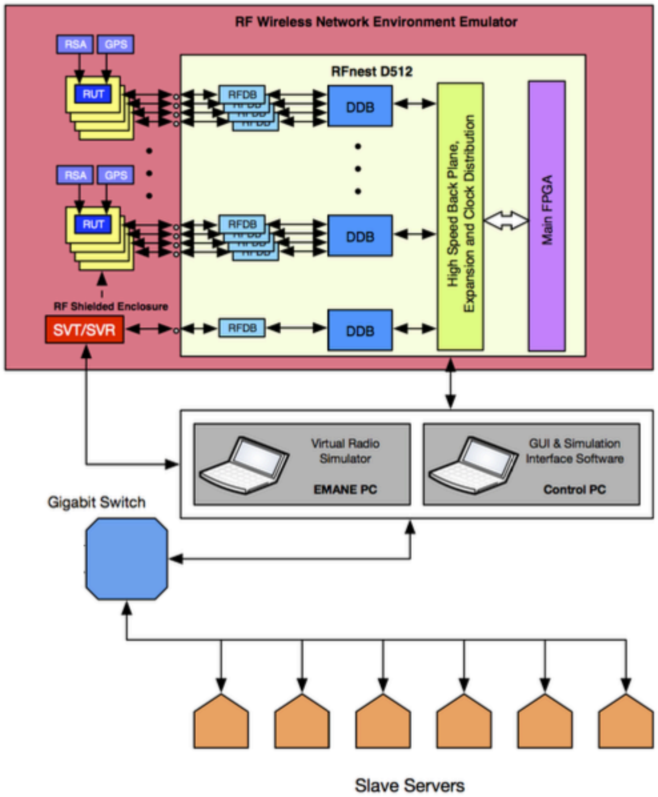


Fig. 5.1 — RFWNEE testbed architecture

5.3 Software Defined Radios

The radios used as the sensors are universal software radio peripherals (USRP), an software-defined radio, with embedded programmable FPGAs that interface with GNU radio, an open-source software development toolkit that provides the signal processing blocks to be implemented on the USRPs. The USRP model used in this test is the USRP N210, where the specs are detailed in Table 5.2 [19]. Some of the key features of the N210 include two RF ports. One port configured for both RX/TX (RF1) and the other port exclusively for RX (RF2). The USRP contains a GPS disciplined oscillator (GPSDO), which can be used for synchronization purposes. In a lab setting, a PPS (pulse per second) and 10 MHz reference input also exists to obtain time synchronization necessary to obtain accurate TDOA measurements. The internal architecture of the USRP N210 is shown in Fig. (5.2). The

USRPs are interfaced with a host PC via ethernet cable, where the main FPGA can programmed using GNU radio or python scripts. The USRP uses the CBX daughtercard, where the key specs are included in Table 5.2.

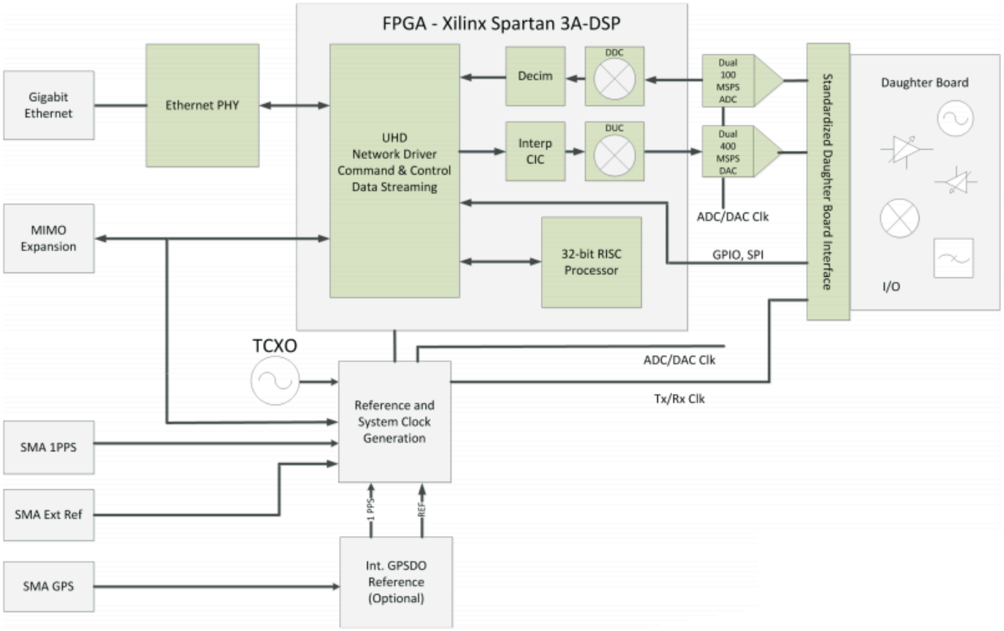
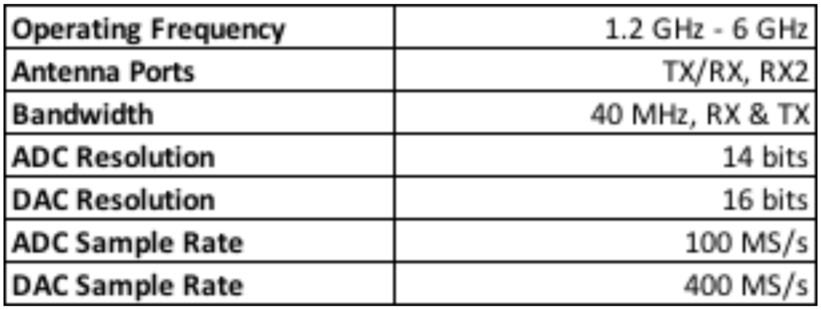


Fig. 5.2 — USRP N210 block architecture

Table 5.2 – USRP N210 w/CBX daughtercard key specifications

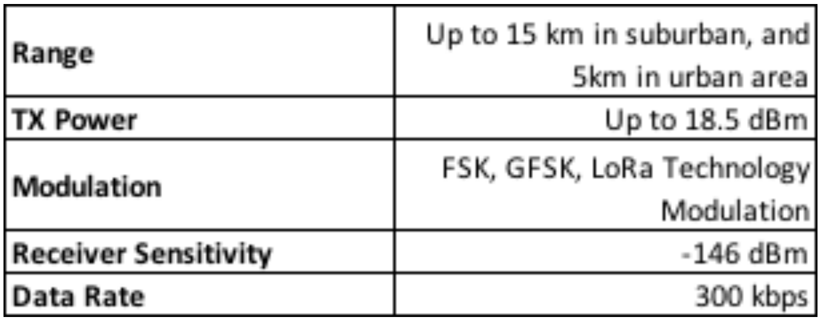


Operating Frequency	1.2 GHz - 6 GHz
Antenna Ports	TX/RX, RX2
Bandwidth	40 MHz, RX & TX
ADC Resolution	14 bits
DAC Resolution	16 bits
ADC Sample Rate	100 MS/s
DAC Sample Rate	400 MS/s

5.4 LoRaWAN Modem

In order to isolate the need for the USRPs to switch between transmit and receive modes, the objective is to use the USRPs as sensors, so we use another communication backhaul that allows for the transmission of sensor data acquisitions from the USRP to the central fusion processor. The priority when choosing this backhaul was longer range. Where WiFi has a higher data rate, it does not cover the long distances needed for a radar coastal monitoring system that could extend multiple miles. LoRaWAN technology offers this longer range at the expense of lower data rate and maximum packet sizes. For each sensor, a LoRaWAN modem is used, that is embedded with a RN2903 produced by Microchip. The advantage offers low-power and long range communications between each modem and its key specs are shown in Table 5.3 [20]. The antenna is configured to both transmit and receive from other LoRaWAN modems, where its figure is depicted in Fig. (5.3).

Table 5.3 – LoRaWAN modem embedded with RN2903 microchip key specs



Range	Up to 15 km in suburban, and 5km in urban area
TX Power	Up to 18.5 dBm
Modulation	FSK, GFSK, LoRa Technology Modulation
Receiver Sensitivity	-146 dBm
Data Rate	300 kbps

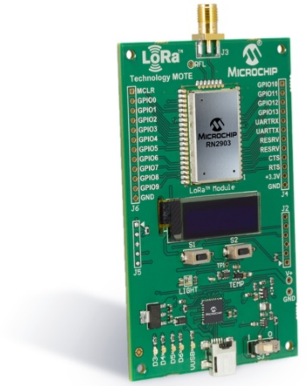


Fig. 5.3 — LoRaWAN RN2903 modem

Chapter 6: Emulation Implementation of Geolocation Algorithms

6.1 PDOA Implementation on SDRs

From Chapter 5, the USRP is being used as the sensors in the network used for geolocation. The effort is to configure the USRP to sense across a specific part of the spectrum in order to obtain an RSSI measurement on the USRP. The GNU Radio software was used to create the signal processing blocks to implement the software radios. We first create a USRP source, which is the first block of any receiver configuration on radios. The source detects any incoming signal. The source is configured with a sampling rate of 25 MHz. This is actually the maximum sampling rate of the radios due to limitation of the analog to digital conversion as shown in Table 5.2. The center frequency was set to 903 MHz, so that it operates in the ISM band. With this configuration, we start the top block of the USRP. The connections are detailed in Fig. (6.1). The skip-head block will skip the recording of initial samples due to a tune-up delay in the radio. This is approximately the first tenth of a second after the top block is created. We create a header block that does the sampling necessary to perform a 400 point FFT. Because of the sampling rate being limited to 25 MHz, we perform a total of four sweeps to obtain a total range of 100 MHz in frequency spectrum. With each sweep, a frequency bin is generated with the hertz per bin equal to the sampling rate divided by the FFT length, which in this case is 62500 Hz. With this, we have a sufficient set of the frequency spectrum to analyze.

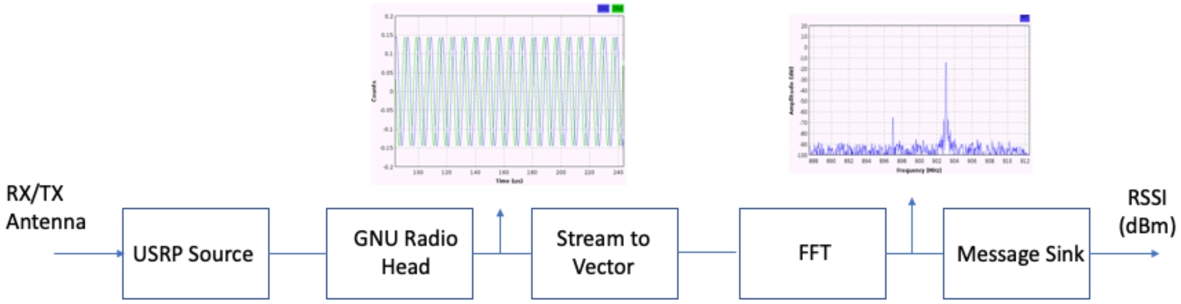


Fig. 6.1 — GNU Radio blocks used to measure RSSI on the USRPs

After collecting the samples, we stream the data to vector format for FFT computation. After the FFT is computed, which results in a complex vector that is converted to a magnitude. This magnitude is then sent to a message sink, where it can then be processed by the host PC connected to the radio. Because we want to view 100 MHz of the spectrum, the top block shown in Fig. (6.1) is created and then deleted at the end of each 25 MHz sweep. Before using the recorded RSSI, the FFT magnitudes are assessed compared to existing calibration tables to the USRP. Insertion losses due to cables are also accounted for in adjusting the measured power output. Since each USRP is different, they each had their own calibration tables with lists of received powers at certain gain levels. Interpolation is then applied by comparing the raw RSSI with the calibration table RSSIs. Adjusting the raw data based off the calibration tables results in smoother RSSI data, which is necessary to optimize the RSSI stability for PDOA.

The timing between the receivers are synchronized by the internal GPSDO (GPS disciplined oscillator) clock, that is driven by an external GPS signal. This allows synchronization within the nanosecond range, which is more than sufficient for PDOA. More detail on the synchronization of the USRPs is described in 6.2.

6.2 TDOA Implementation on SDRs

6.2.1 Synchronization of radios

Measuring the time-difference of a particular signal at two different receivers requires the two receivers to share a common clock. The USRPs have the ability to be synchronized by a common clock. Because of the USRP crystal oscillator, the clocks will drift overtime. The USRP contains a GPS disciplined oscillator (GPSDO) that allows for an external GPS connection, and also contains ports for other external references of 10 MHz and 1 PPS pulses if a GPS signal is not available [8]. Using GPS as a synchronization technique will deliver more accuracy over other common synchronization schemes [9]. The clock in each satellite is continuously calibrated based off a uniform world time standard, this can deliver synchronization performance to the nanosecond range, required for accuracy in TDOA. In a laboratory environment, obtaining a signal directly from GPS is not feasible, so we rely on supplying an external 10 MHz reference signal and a 1 PPS (Pulse per second) signal to achieve synchronization similar to GPS. Obtaining synchronization in a laboratory environment is an easier task vs over-the-air (OTA) because of the ability to connect devices via cable, which gives the ability to use an external 10 MHz reference and PPS over cable. Both are generated using the output from GPS emulator and the time standard used to configure the USRP time is Unix time.

When using the reference for synchronization, we wait for a lock into the reference signal. Once the reference is locked, we then wait for the pulse per second to change, and then wait to determine the Unix time where the PPS edge occurs, and will then set the USRP time to the Unix time. This Unix time is determined on the host PC connected to each radio via an NTP server.

Even with synchronized clocks, a certain amount of offset and clock skew is expected. The accuracy of the GPS drift was evaluated in [9]. Given the reference time set to *C(t) = t*, the clock function of a radio *i* is given in (6.1).

$$C_i(t) = \Delta f_i t + \Delta\theta_i \qquad (6.1)$$

where Δf_i and $\Delta\theta_i$ are the clock skew and the initial offset respectively. The relative offset ad relative skew between two radios *i* and *j* are $\Delta C_{ij} = C_i(t) - C_j(t)$ and $\Delta f_{ij} = \Delta f_i - \Delta f_j$ respectively. IF perfectly synchronized, these offsets are zero. For GPS time, if was found in [9], that the offset was on average 200 ns of drift. Using the NTP server time with the 10 MHz reference and 1 PPS signal connected to the radios by cable, this drift can be corrected for over time, and on each sampling iteration, we correct for this by resetting the USRP time to the UNIX time when the last at the PPS edge. This synchronization setup resulted in synchronization accuracy, well within the sampling resolution of the USRPs.

6.2.2 Estimating the time-difference

We now outline general method of computing TDOA requires the streaming of time-domain samples at each receiver. After a sufficient number of samples has been collected by the sensors, they are processed to determine the time-difference between them. Since we are tracking one particular emitter, ideally the two receivers will see the same signal but shifted in time relative to each other. We outline the generalized cross-correlation problem used to compute the time difference given in [10]. The block diagram for the problem is shown in Fig. (6.2).

Given an emitter defined as $s(t)$, the signals seen at two receivers is as follows:

$$x_1(t) = A_1 s(t-\tau_1) + \eta_1(t) \qquad (6.2)$$

$$x_2(t) = A_2 s(t-\tau_2) + \eta_2(t) \qquad (6.3)$$

where $\tau = \tau_2 - \tau_1$ is the desired time-difference between the two sensors. From (6.2) and (6.3), the cross-correlation between the two received signals x_1 and x_2 is as follows:

$$R_{x1x2} = R_{ss}(t-\tau) + R_{n1n2}(t) \qquad (6.4)$$

If we assume that the noise is white-Gaussian and uncorrelated, then $R_{n1n2}(t)$ is zero. Then the it becomes:

$$R_{x1x2}(\tau) = \int_t^T s(t)s(t-\tau)dt \qquad (6.5)$$

The discrete formula for autocorrelation is obtained as follows:

$$\hat{R}_{x1x2}(m) = \frac{1}{N}\sum_0^{N-1} s(m)s(n+m) \qquad (6.6)$$

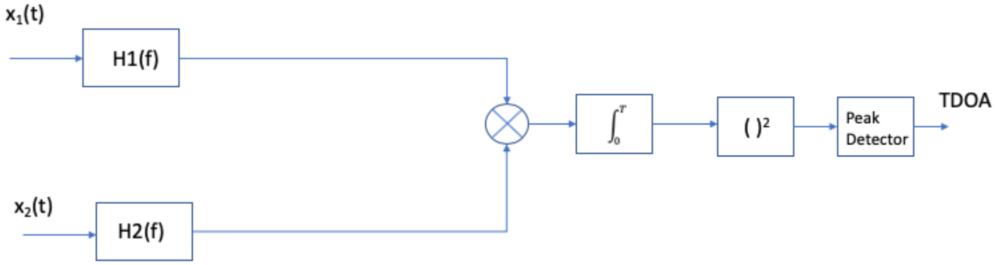


Fig. 6.2 — Block diagram for generalized cross-correlation of two receiver signals

Taking the Fourier transform, we obtain:

$$G_{y1y2} = H_1(f)H_2^*(f)G_{x1x2}(f) \qquad (6.7)$$

where G_{x1x2}(f) is the Fourier transform of the cross-correlation between x_1 and x_2.

The multiplication of H1(f) and H2*(f) comprises of the general frequency weighting. For generalized cross-correlation this is simply equal to 1. Other weighting functions are outlined in [10] and are useful when there exists multipath and other noise outside of white Gaussian noise. Choosing filters allows for more efficient separation of delays due to multipath that could affect the ambiguity of the peak detector of the correlation of the output of the filters. Since we are not exploring multipath mitigation techniques in this research, we do not apply these filters in our time-difference computation.

After the cross-correlation is found, the result is squared to get rid of negative values, and the maximum value of the square cross-correlation is found to be the estimated time-difference between the sensors. The resulting time-difference will have the peak at a particular sample number *n*. When sampling with the USRP N210, the limit of 25 MHz sampling rate restricts the time-resolution to 40 nanoseconds per sample. The implications of this limitation is seen in the emulation results in Chapter 7.

6.3 Emulation Integration

6.3.1 Waveform Selection

The selection of the waveform is important for the accuracy of TDOA estimation. Because the cross-correlation is taken, the signal must have good correlation properties. If a continuous sine-wave is used, the time-difference cannot be calculated because there will be phase ambiguity. This means the signal must be a modulated waveform. To meet this requirement, we use an FM signal with a modulation index larger than 0.5 for the emitter, which is an implementation of wideband FM. We use a sine-wave as the modulating signal. We use metrics similar to a broadcast FM system with a frequency deviation of 75 kHz and a maximum modulation frequency of 15 kHz. This signal will have sufficient bandwidth to minimize phase ambiguity.

6.3.2 Hardware Configuration

For the emulation, we had a total of four radio assets on that fit the specifications of the experiment. Each USRP is connected to a host PC via ethernet cable. The GNU Radio blocks created to collect spectral and raw IQ data is sent by the radio to the host PC where the host PC then processes that info accordingly. All of the data needed for the geolocation is then sent over-the-air by the LoRa modems connected to each PC. As detailed in Chapter 5, this communications protocol is used as a backhaul so the radios would only receive sensing data, and not have their sensing interrupted by a transmit period. The LoRa modems send the sensor data to a LoRa router connected to a host PC allocated specifically for the fusion of the sensor data. This PC will organize the sensor data, perform the cross-correlation on the TDOA data, and then perform the necessary geolocation algorithm. The signal generator used as the emitter is applied on one of the RFNest input ports. The hardware connections for the emulation of the radios over the RFWNEE are shown in Fig. (6.3). Next, we will describe how the geolocation scenarios are emulated in software.

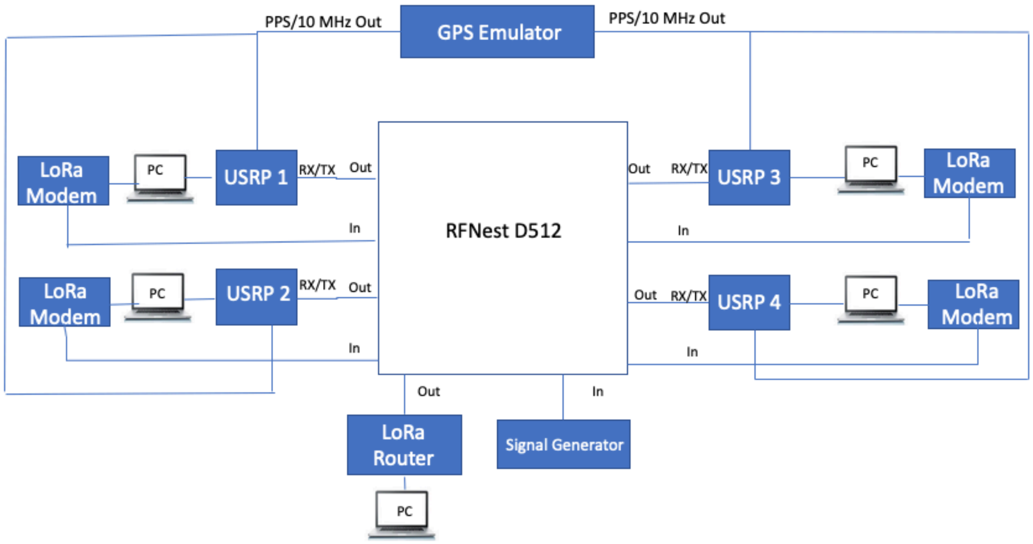
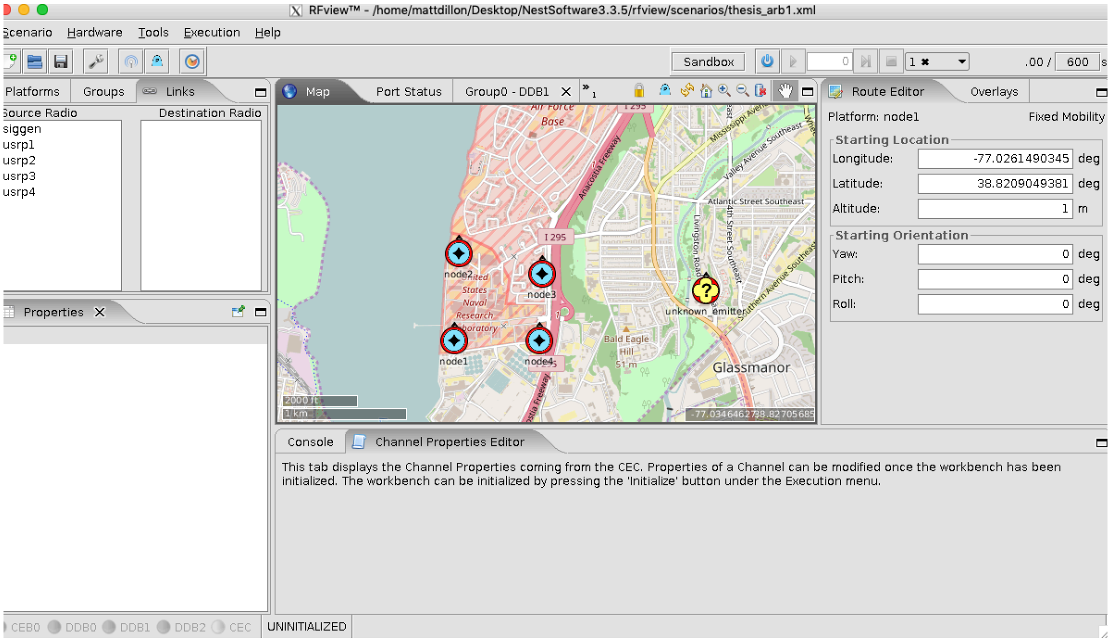


Fig. 6.3 — Hardware schematic for the laboratory emulation of the geolocation network

6.3.3 Software Emulation Tools

The RFView emulator is a software GUI tool that is used to configure the RFnest for a given emulation scenario. A screenshot of the GUI is shown in Fig. (6.4). In the figure, a platform with a set location is set up. This platform can either be airborne, or ground-based. For this research, everything was ground-based. The host-PC has access to online maps repository and when placing the platforms into RFview, the real-world latitudes and longitudes can be adjusted accordingly. When the platforms have been created, the individual radios are configured and assigned to an associated port on the RFNest. As mentioned in Chapter 5, there are a total of twelve IO ports allowing for a total of 12 devices in half-duplex mode [4].


RFview™ - /home/mattdillon/Desktop/NestSoftware3.3.5/rfview/scenarios/thesis_arb1.xml

Source Radio: siggen, usrp1, usrp2, usrp3, usrp4

Platform: node1 Fixed Mobility

Starting Location
Longitude: -77.0261490345 deg
Latitude: 38.8209049381 deg
Altitude: 1 m

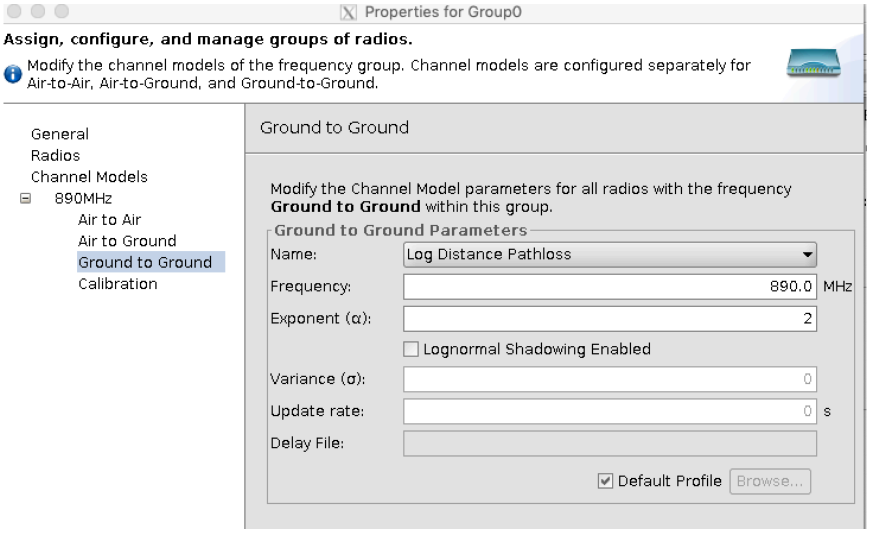
Starting Orientation
Yaw: 0 deg
Pitch: 0 deg
Roll: 0 deg

This tab displays the Channel Properties coming from the CEC. Properties of a Channel can be modified once the workbench has been initialized. The workbench can be initialized by pressing the 'Initialize' button under the Execution menu.

UNINITIALIZED

Fig. 6.4 — A screenshot of the RFView GUI used to configure the RFNest emulator

The radios can be assigned to different groups with different channel parameters. For this research, all radios are operating in the same channel environment. Since the path-loss model used in the PDOA estimation equations was log-normal, this is the channel model used to control the path-loss, delay, and doppler shift. Since we are not concerned with the doppler effect, only the path-loss and delay effects are important. As shown in Fig. (6.5), the channel model is shown. Even though the emitter is set to a carrier frequency of 903 MHz, the actual frequency of the channel is set to 890 MHz. This separation between the waveform frequency and channel frequency is important as the RFNest generates a local oscillator frequency [4], which can interfere with any radios transmitting or receiving at that same frequency.


Properties for Group0

Assign, configure, and manage groups of radios.

Modify the channel models of the frequency group. Channel models are configured separately for Air-to-Air, Air-to-Ground, and Ground-to-Ground.

Ground to Ground

Modify the Channel Model parameters for all radios with the frequency **Ground to Ground** within this group.

Ground to Ground Parameters

Name: Log Distance Pathloss

Frequency: 890.0 MHz

Exponent (α): 2

□ Lognormal Shadowing Enabled

Variance (σ): 0

Update rate: 0 s

Delay File:

☑ Default Profile

Fig. 6.4 — The radios assigned to a group and the channel properties of the group of radios is configured in this window

There is also the capability to control the emulated transmit power of the emitter, using the transmit power offset adjustment. For this test, we transmit at 30 dBm to cover the range of 2 km for RSSIs that are above the noise floor of USRPs. With the emulation environment and all hardware and tools addressed for the evaluation of the radios addressed, we proceed to define the specific emulation scenarios and show the geolocation performance results obtained in the emulation in Chapter 7.

Chapter 7: Geolocation Emulation Results

7.1 Emulation Scenarios

The objective was to emulate the PDOA, TDOA, and hybrid algorithms on the RFNest to demonstrate the differences in error performance between the algorithms under particular environmental conditions, including the distance of the emitter relative to the network, and noisy measurements. It is also useful to explore the effect of the sensor network geometry on the error in location estimation. Two different sensor topologies are explored in these emulations: an arbitrary topology, as shown in Fig. (7.1) where the sensors are placed in any random location within a specific area.

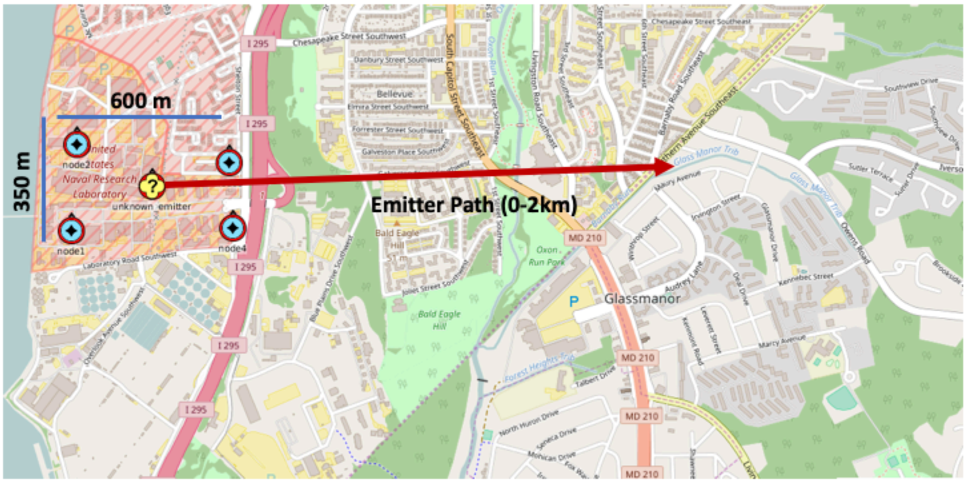


Fig. 7.1 — Arbitrary (non-linear) sensor topology emitter path is indicated by a red arrow, will move 2 km out of the sensor network AO by the end of the run. Dimensions of AO of sensors shown. Sensors are the blue icons. Yellow question mark icon is the emitter starting position

The second topology is linear as shown in Fig. (7.2) where all of the sensors are placed in a straight line. In theory, a linear topology will result in more ambiguity in predicting the direction of the emitter relative to the sensors and it would result in a larger error as a result.

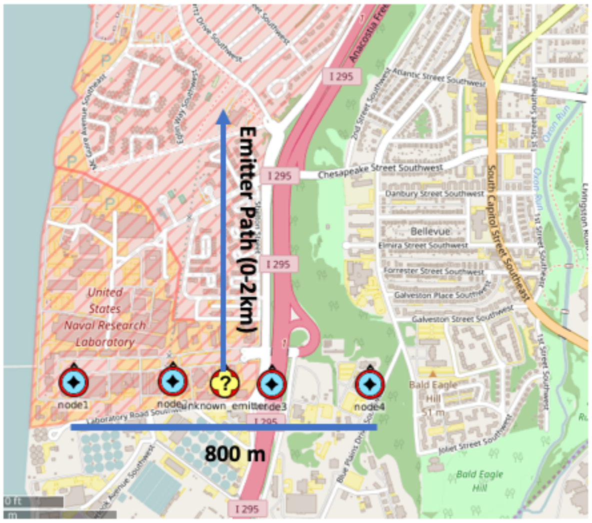

Emitter Path (0-2km)
800 m
node1
node
unknown_emitter
node3
node4
United States Naval Research Laboratory
Chesapeake Street Southwest
South Capitol Street Southwest
Bellevue
Bald Eagle Hill
I 295

Fig. 7.2 — Linear sensor topology emitter path is indicated by a red arrow, will move 2 km out of the sensor network AO by the end of the run. Dimensions of AO of sensors shown. Sensors are the blue icons. Yellow question mark icon is the emitter starting position

For the first set of tests, we will place the emitter in the middle of the sensor network (the common midpoint of all the sensors), and will move the emitter further out of the network with a resolution of 200 m, until it is 2 kilometers away from the network. Given the transmitter power is at 30 dBm, this distance is sufficient such that the RSSIs at 2 km are still above the receiver sensitivity of the USRPs, which is around -85 dBm. In the second set of tests, more noise is added on top of the baseline noise that already exists due to instrumentation errors and thermal noise. We inject additional noise variance to the RSSI measurements observe if the effects are consistent with the simulations when adding variance. No additional variance is added to the TDOA measurements due to the already poor sampling resolution.

7.2 Results

7.2.1 Arbitrary Topology

For each distance, we compute the RMSE of the location estimate, which is the distance in meters from the actual emitter to the estimated position. At each position, the average o 50 runs was computed as the error for that particular distance. The results are shown in Fig. (7.3). On average over all the distances, the PDOA error was 293 m, the TDOA error was 700 m, and the hybrid error was 200 m. Initially, TDOA is the most accurate when the emitter is within the sensor AO. At the initial emitter point, TDOA's average error is 6 meters, and PDOA is 30 meters. Overall, within the network, both algorithms have consistent performance. As the emitter moves outside the network AO, the error increases noticeably similar to the MATLAB error analysis, with a roughly linear trend. The TDOA performance is degraded significantly. While both TDOA and PDOA error have the same trend, due to the poor resolution of TDOA measurement, the TDOA accuracy is degraded more significantly. The hybrid algorithm's overall error average is a significant improvement from the individual algorithms.

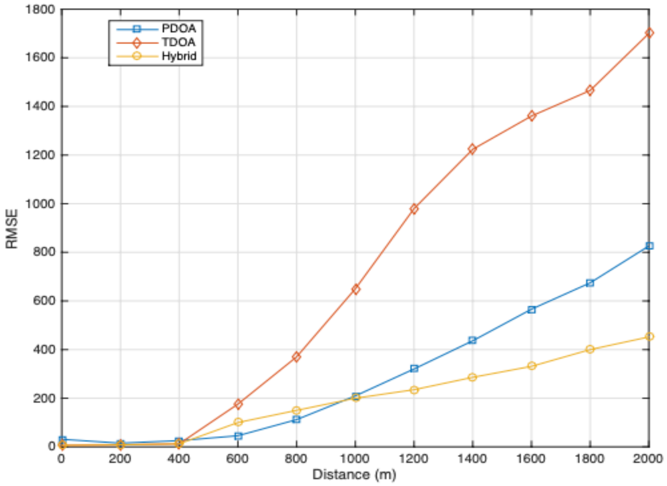


Fig. 7.3 — RMSE for the PDOA (square), TDOA(diamond), and hybrid (circle) algorithms as a function of distance of emitter from center of network for an arbitrary sensor topology

7.2.2 Linear Topology

Fig. (7.5) shows the RMSE results for the linear topology over various emitter/sensor network distances. In this scenario, the hybrid was modified based off the topology to not use the RSSI to determine whether it perform the conventional TDOA or use the time-of-arrivals based off the PDOA estimate, since there is little time the emitter spends within a sensor network AO, due to the linear sensor geometry. Instead, it would always compute the PDOA time-difference estimates and compare them to the actual measured time estimates. The average error for PDOA was 1000 m, for TDOA it was 786 m, and for the hybrid it was 461 m. This is because there is ambiguity in the direction of the emitter for the PDOA estimate. An illustration of how this ambiguity is created by the circle intersections is shown in Fig. (7.4). Without noise, there are two possible locations and only one can be chosen. With noise, the larger amount of intersections will alternate to one side over the other randomly. This means that half the time, the estimate will be on the wrong, side of the sensors where the other half, the estimate will be on the right side. The reason this ambiguity doesn't exist in TDOA results is because the emitter was consistently moved in one particular direction and only one of the solutions in (2.17) was chosen, and it was the correct one. If the emitter was moved in the opposite direction, the error would've been double, due to the wrong solution from (2.17) being used. The hybrid also performs significantly better because even though there is ambiguity among the most optimal intersection from PDOA's grid density algorithm, the relative RSSIs and the erroneous PDOA estimate which lies on the opposite side of the sensor network still has the correct ranges.

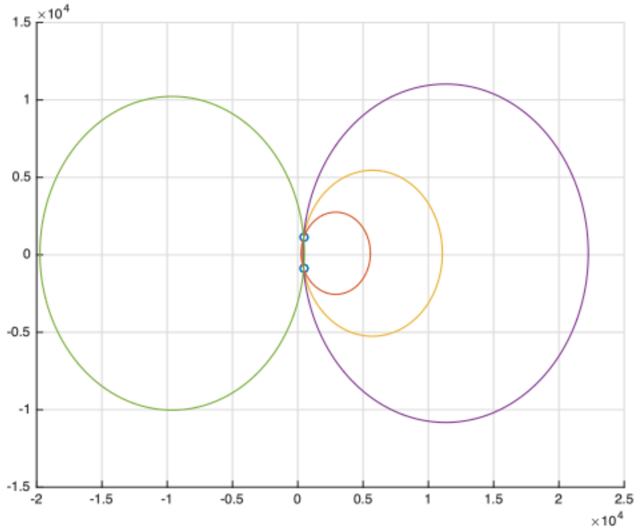


Fig. 7.4 — Circle intersections for a linear topology. When noise-free, there are two possible intersections, resulting in an ambiguous solution

In some practical scenarios, we may know the relative direction of the emitter, and PDOA results will be improved. For example, in the case of a coastal monitoring network tracking ships, we may know the general direction of the emitter. This scenario was run again where the PDOA algorithm was modified to filter out intersections that were found to be on the wrong side were suppressed from the grid-density filter. Fig. (7.6) shows the results, which include PDOA now outperforming TDOA with an average of 547 m. Overall, the error for this topology in every algorithm is worse than the arbitrary topology, as expected due to the shorter duration where the emitter is within a sensor AO.

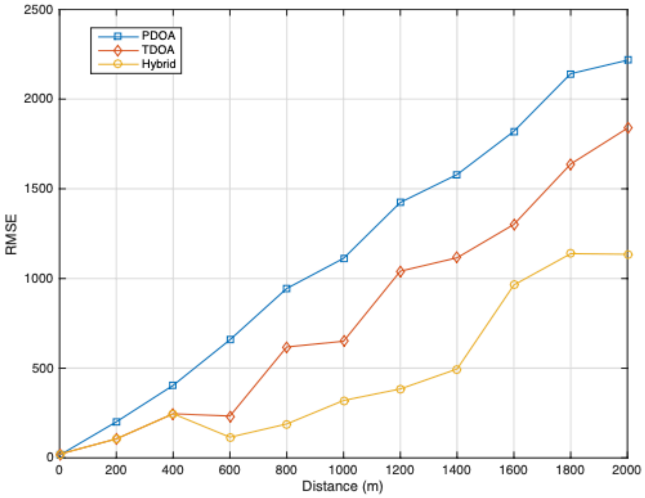


Fig. 7.5 — RMSE for the PDOA (square), TDOA(diamond), and hybrid (circle) algorithms as a function of distance of emitter from center of network for a linear sensor topology

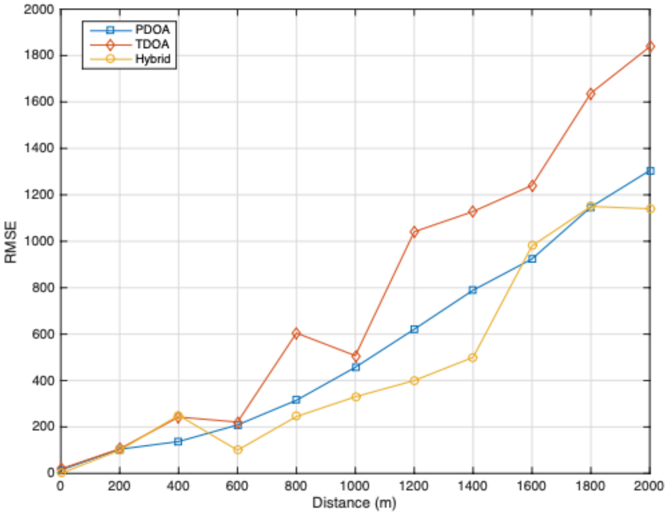


Fig. 7.6 — RMSE for the PDOA (square), TDOA(diamond), and hybrid (circle) algorithms as a function of distance of emitter from center of network for a linear sensor topology with a priori info about the direction of the emitter

7.2.3 Noise Variance Effects

It was useful to observe adding more noise variance to the PDOA in addition to the existing noise already present. The noise variance was swept between – and 4, with increments of 0.5. The average error at each noise variance was taken based off of the results of 50 runs at each noise variance. The results are shown in Fig. (7.7). What was observed was an increase in the PDOA results. Since no additional noise was added to the time-difference, this was held constant. The average error for PDOA was 315 m, the TDOA error was 369 m, and the hybrid error was 295 m. What was observed was that when TDOA's noise error was larger than PDOA, the hybrid algorithm's error converged to TDOA's error. This is because the TDOA calculated from the PDOA location estimate was outside the 40 nanosecond bound, so it was able to recognize that the RSSI by PDOA was not reliable, so the original time-difference measurements were used to estimate the emitter location.

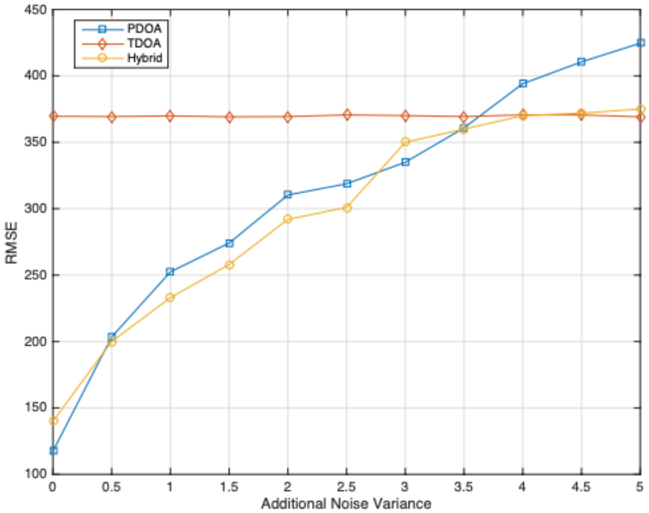


Fig. 7.7 — RMSE for the PDOA (square), TDOA(diamond), and hybrid (circle) algorithms as a function of noise variance in RSSI. Here, we use the arbitrary topology in the first emulation scenario with the emitter held at 1 km from the center of the network

7.3 Discussion

From the emulation scenarios run, the error trend was generally consistent with the MATLAB error performance analysis of the algorithms. The more unintuitive result was that TDOA performed worse on average than PDOA when the emitter was outside the sensor network. This was due to the poor sampling resolution of the radios. However, this kind of TDOA accuracy can be expected due to other factors such as shadowing and other multipath effects, leading to lower accuracy. In these cases, the RSSI measurement was more reliable than TDOA, so the hybrid was applied where the conventional TDOA algorithm was used when the relative RSSI difference between the left sensors were approximately equal to the right sensors. The topology was exploited here and the hybrid used the TDOA measurements entirely over the first 400 meters. Once the emitter moved outside the sensor network, the hybrid recognized the RSSI imbalance between the left and right sensors and started computing time-differences based off the estimate attained from PDOA. These estimates would be used in the TDOA computation if the time-difference was within +/- 40 nanoseconds of the TDOA estimate. If this was not the case, then the RSSI information could not be relied on, and the original TDOA value would be used. This adjustment led to generally better results overall. In the linear topology, we saw that the PDOA result was worse without a priori information given about the direction of the emitter. When given a priori information, its error performance relative to TDOA and the hybrid was generally consistent with the arbitrary topology. In the noise variance analysis, the decision-making process of the hybrid algorithm was tested, as the PDOA error eventually surpassed the TDOA error. The hybrid algorithm found the PDOA time-difference estimates to be out of range of the TDOA result. This analysis can be extended further to further studying the correlation between increasing the noise floor of the radios to determine its overall effect on the time-difference accuracy. In this event, the noise added for the RSSI would not be independent of the noise in TDOA, although TDOA is generally more resilient to the thermal noise found in RSSI. The noise due to multipath effects can lead to more unpredictable measurement results and more error for both TSOA and PDOA.

Chapter 8: Conclusion and Future Work

In this research, the effort was to implement a practical distributed sensor network of software-defined radios with the goal of estimating the position of an unknown emitter with no a priori information about the emitter relative to the group of sensors other than the type of waveform and frequency channel. To effectively geolocate an emitter, two main methods, PDOA and TDOA were investigated. With PDOA, the strategy was to use the difference in RSSI for multiple pairs of sensors. The power-difference results in multiple circles intersecting, with the majority of the points converging around the actual location. A grid-density algorithm was used to determine where this occurs. For TDOA, the time-difference was determined from taking the time-difference between receiver pairs that are synchronized. Due to the hardware limitations, this precision was limited to 40 nanosecond resolution. The algorithm utilized a two-step process where the linear least squares was computed, and then the result was used in an approximate maximum likelihood estimate to determine the location estimate. Error analysis was conducted for both algorithms and simulations run in MATLAB in which noise variance and emitter/network relative distance were varied, and the overall MSE was observed over several iterations, which matched the theoretical analysis of the performance of algorithms. A hybrid method was developed in which the measured time-difference and the PDOA time-difference estimate was computed, and determination was made which measurement to use for a particular sensor pair. The sensor network was implemented in a laboratory environment using the RFnest channel emulator to closely match a field-test environment. The algorithms were tested on the emulator to study the error performance of the geolocation on the actual radios, and there was correspondence in the error performance for both MATLAB and laboratory emulation for the defined scenarios. In addition, the choice of network topology (linear or arbitrary) was investigated. It was also found that the hybrid algorithm outperforms the standalone TDOA and PDOA algorithms on average.

The complexity of the sensor network and channel environment should be explored further, such as adding more radios to the network. It is also worth investigating a large-Muscale network over a vast area. Some new challenges over a large area of coverage would include inconsistencies in the terrain, and as a result, the channel model. Methods of channel equalization should be explored to improve PDOA algorithms. It is also important to analyze shadowing effects and multipath on the algorithm accuracy and appropriate mitigation techniques, such being able to recognize inconsistent measurements. The hybrid algorithm's recognition of poor PDOA data is promising in finding a path forward on various mitigation techniques. Furthermore, another layer of complexity is the effect of mobility of the emitter. Fast-moving emitters will have the doppler-effect, which could affect the time and frequency measurements. In this case, it may be important to implement frequency-difference of arrival (FDOA) to consider these effects [16]. Other combinations of geolocation algorithms such as FDOA and AoA (triangulation), should also be assessed, compared to the two main range- based algorithms of PDOA and TDOA.

Bibliography

[1] S. Guo, B. Jackson, S. Wang, R. Inkol, and W. Arnold, “A novel density-based geolocation algorithm for noncooperative radio emulator using power difference of arrival”, *SPIE Defense, Security and Sensing,* vol. 8061, (May 2011).

[2] J. Cox, M.B. Partensky “Spatial localization problem and the circle of Apollonius”, *arXiv preprint physics/0701146* (2007)

[3] T. Chan, “A simple and efficient estimator for hyperbolic location”, *IEEE Transactions on Signal Processing* Vol. 42, (August 1994).

[4] M. Dillon, E.H. Fu, D. Gdula, T. Mai, J. Molnar, and L. Tran, *Real and Virtual Wireless Radio Network Emulator,* Military Communications Conference (IEEE, 2016). doi:10.1109/MILCOM.2016.7795507

[5] L. Asmaa, K. A. Hatim and M. Abdelaaziz, "Localization algorithms research in wireless sensor network based on Multilateration and Trilateration techniques," *2014 Third IEEE International Colloquium in Information Science and Technology (CIST)*, Tetouan, 2014, pp. 415-419. doi: 10.1109/CIST.2014.7016656

[6] ITU, “Comparison of time-difference-of-arrival and angle-of-arrival methods of signal geolocation”, *SM Series Spectrum management*, pp. 5-6, (2018).

[7] Ghannouchi, F.M. & Wang, Donglin & Tiwari, Smita. (2012). Accurate Wireless Indoor Position Estimation by Using Hybrid TDOA/RSS Algorithm. 2012 IEEE International Conference on Vehicular Electronics and Safety, ICVES 2012. 437-441. 10.1109/ICVES.2012.6294291.

[8] “Common Reference Signals.” *USRP Hardware Driver and USRP Manual: Device Synchronization*, files.ettus.com/manual/page_sync.html.

[9] Zan Li, D. C. Dimitrova, T. Braun and D. Rosário, "Highly accurate evaluation of GPS synchronization for TDOA localization," *2013 IFIP Wireless Days (WD)*, Valencia, 2013, pp. 1-3.
doi: 10.1109/WD.2013.6686489

[10] Ahmed, Hesham Ibrahim et al. “Estimation of Time Difference of Arrival (TDOA) for the Source Radiates BPSK Signal.” (2013)

[11] Karanam, Chitra R., Belal Korany and Yasamin Mostofi. “Magnitude-Based Angle-of-Arrival Estimation, Localization, and Target Tracking.” *2018 17th ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN)*(2018): 254-265.

[12] Yanping Zhu, Daqing Huang and Aimin Jiang, "Network localization using angle of arrival," *2008 IEEE International Conference on Electro/Information Technology*, Ames, IA, 2008, pp. 205-210. doi: 10.1109/EIT.2008.4554297

[13] S. Murphy, William & A. Hereman, Willy. “Determination of a position using approximation distances and trilateration”, (1995)

[14] B. R. Jackson, S. Wang, and R. Inkol, “Emitter Geolocation estimation using power difference of arrival. An algorithm comparison for non-cooperative emitters,” rept., Defence R&D Canada - Ottawa, DRDC Ottawa (May 2011).

[15] W.H. Foy,, “Position-location solutions by Taylor Series estimation,” *IEEE Trans. Aerospace Electronic Systems.,* vol. AES-12, pp. 187-194, (1976)

[16] D. Musicki and W. Koch, "Geolocation using TDOA and FDOA measurements," *2008 11th International Conference on Information Fusion*, Cologne, 2008, pp. 1-8.

[17] S. Ravindra, S.N. Jagadeesha, “Time of Arrival Based Localization in Wireless Sensor Networks: A Linear Approach”, Jawaharlal Nehru National College of Engineering, Visvesvaraya Technological University, Belguam, Karnataka, India. *Signal and Image Processing: An International Journal* Vol. 4, (August 2013).

[18] RFnest Product Specifications, Intelligent Automation Inc, 2017 https://www.i-a-i.com/wp-content/uploads/2017/07/RFnest-Specsheet-2017.pdf

[19] USRPN210 User Manual, Ettus http://files.ettus.com/manual/page_usrp2.html

[20] “RN2903 Low-Power Long Range Technology Transceivr Module”, Microchip, 2018 http://ww1.microchip.com/downloads/en/DeviceDoc/RN2903-LoRa-Technology-Transceiver-Module-Data-Sheet-DS50002390G.pdf