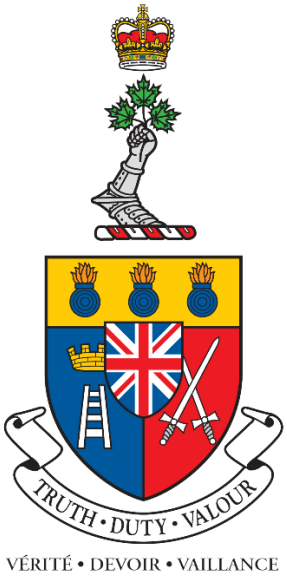
Numerical Methods of AOA Geolocation Using a Single Moving Sensor

Méthodes numériques de géolocalisation AOA à l'aide d'un seul capteur mobile



A Thesis Submitted to the Division of Graduate Studies
Of the Royal Military College of Canada
by

James Helmar Bayes, CD
Captain

In Partial Fulfillment of the Requirements for the Degree of
Masters of Applied Science in Electrical Engineering

May 2023

Abstract

There are several applications that require an estimation of the location of an electromagnetic emitter. Search and rescue, as well as other civilian and military applications require the determination of an emitter's location. This is also known as geolocation. Several methods have been developed for geolocation including measuring the received power from the emitter, the angle that the electromagnetic wave arrives at a sensor and calculating the difference in time of arrival of the emitter signal at spatially separated sensors.

Recent research has focused on increasing the accuracy of the geolocation by using multiple sensors and complex algorithms. Although this is very useful, there are some geolocation applications that may benefit from simpler, low-cost systems. This thesis employs Angle of Arrival measurements and a simple algorithm of linear equations. Although this method is less accurate than other methods, it is computationally efficient and enables geolocation by a single moving sensor. This numerical method is therefore a viable option for systems where simplicity or lower costs are a more limiting factor than accuracy.

Résumé

Il existe plusieurs applications qui nécessitent une estimation de la localisation d'un émetteur électromagnétique. La recherche et le sauvetage, ainsi que d'autres applications civiles et militaires, nécessitent la détermination de l'emplacement d'un émetteur, également connu sous le nom de géolocalisation. Plusieurs méthodes ont été développées pour la géolocalisation, notamment la mesure de la puissance reçue de l'émetteur, l'angle d'arrivée de l'onde électromagnétique sur un capteur et le calcul de la différence de temps d'arrivée du signal de l'émetteur sur des capteurs spatialement séparés.

Des recherches récentes se sont concentrées sur l'augmentation de la précision de la géolocalisation en utilisant plusieurs capteurs et des algorithmes complexes. Bien que cela soit très utile, certaines applications de géolocalisation peuvent bénéficier de systèmes plus simples et peu coûteux. Cette thèse utilise des mesures d'angle d'arrivée et un algorithme simple d'équations linéaires. Bien que cette méthode soit moins précise que d'autres méthodes, elle est efficace en termes de calcul et permet la géolocalisation par un seul capteur mobile. Cette méthode numérique est donc une option viable pour les systèmes où la simplicité ou les faibles coûts sont un facteur plus limitatif que la précision.

Table of Contents

List of Figures

List of Tables

List of Abbreviations

AOA	Angle of Arrival
CRLB	Cramer-Rao Lower Bound
EW	Electronic Warfare
ML	Maximum Likelihood
RMSE	Root Mean Squared Error
RSS	Received Signal Strength
TDOA	Time Difference of Arrival
WGN	White Gaussian Noise
WLS	Weighted Least Squares

1. Introduction

Determining the location of an energy emitting source is a requirement for cellular telephone service providers and is also important for radar and sonar applications, as well as in Electronic Warfare (EW). Several methods have been developed for geolocation. The most common methods used for geolocation of unknown emitters are Received Signal Strength (RSS); Angle of Arrival (AOA); and Time Difference of Arrival (TDOA). Each method has different physical requirements; advantages and disadvantages. AOA balances simplicity of system design with accuracy of measurements and can enable a single moving sensor to localize a stationary non-cooperative emitter.

Many of the AOA papers use Weighted Least Squares (WLS) algorithms [1] [2] [3] [4] [5] [6]. The standard algorithm uses the AOA measurements to locate the emitter. This thesis uses the law of sines to calculate the distances between the measurement locations and the emitter, then a linear system of equations to locate a non-cooperative emitter. The linear system of equations makes the computation of the estimated location easier at the expense of a small loss of accuracy.

1.1 Motivation

Whether for search and rescue or military operations, an emitter must first be detected within the search area before the localization process can begin. Although any single sensor can detect an emitter, not all localization methods can be conducted with a single sensor. For example, if the emitter is also moving then doppler effects make the use of moving sensors more complicated. In circumstances where localization of the emitter is time sensitive, there are two advantages to using localization methods that require only a single sensor. First, localization can begin as soon as the emitter is detected. Second, systems that use single sensor localization can cover larger areas with the a given number of sensors.

For any given power output of the emitter, there is a maximum detection distance. This means that each sensor has a maximum range to the emitter and in turn a maximum scanning area for the sensor. Systems that require multiple sensors to receive the signal from the emitter simultaneously, also require the sensors to overlap their areas of localization (Figure 1). This reduces the total area that a given number of sensors can cover at any time. In contrast, if the

localization can be performed by a single moving sensor, the area of localization can be larger, as seen in Figure 2.

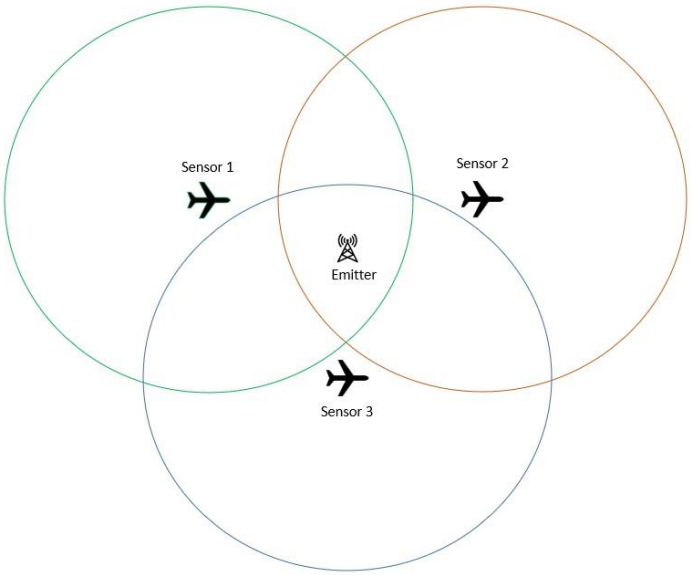


Figure 1 Overlapping Scan Areas

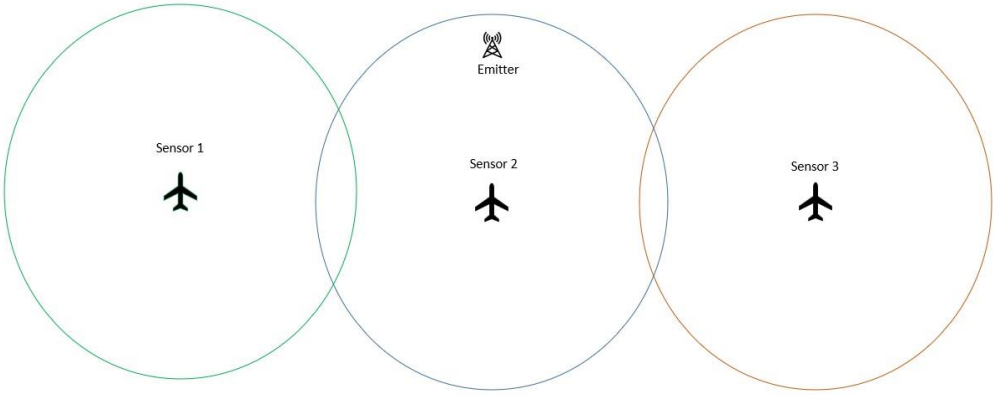


Figure 2 Minimally Overlapping Scan Areas

Methods like TDOA require overlapping localization aeras and therefore reduce the total areas covered at one time [7]. When minimally overlapping localization areas are employed the area of operations is maximised. The accuracy of a single sensor measurement will be less than if multiple sensors make simultaneous

measurements. However, additional sensors could be moved towards the emitter once the initial sensor has an estimated location.

AOA methods can be used by a single moving sensor to localize non-cooperative emitters. It has been shown that systems with fewer sensors are less accurate [5] [8] [9]. Increasing the number of measurements taken by a single sensor can increase the accuracy of the estimation in a manner similar to increasing the number of sensors. Increasing the number of measurements will require a proportional increase in the number of calculations required.

In order to enable a single sensor to accurately geolocate a non-cooperative emitter, a computationally simple algorithm is required. Such a system would provide a low complexity and low-cost solution that could maximise the location area for a given set of sensors.

1.2 Statement of Deficiency

The basic geometry of AOA measurements is depicted in Figure 3. This thesis assumes that the first measurement location is used as a reference for all calculations.

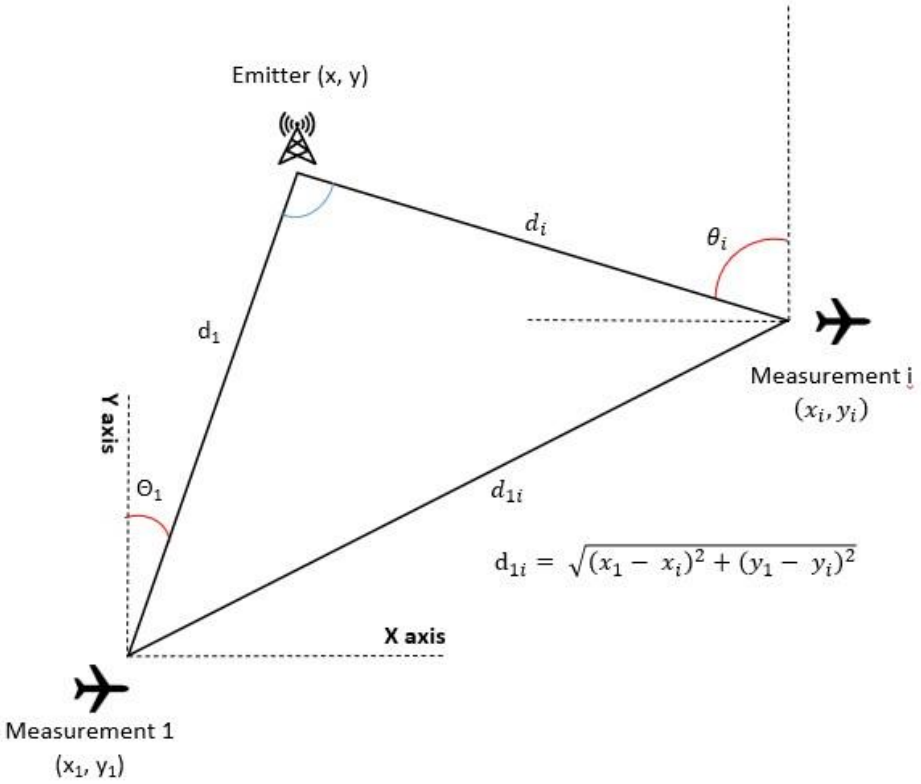


Figure 3 Basic AOA Geometry

$$\theta_i = \arctan\left(\frac{x - x_i}{y - y_i}\right) \tag{1}$$

The reliance on WLS algorithms for AOA geolocation assumes that the problem is linear. From the relationship in equation (1), the general estimation equation used in [1] [5] and [9] is:

$$\underbrace{\begin{bmatrix} \cos\theta_1 & -\sin\theta_1 \\ \cos\theta_2 & -\sin\theta_2 \\ \vdots & \vdots \\ \cos\theta_i & -\sin\theta_i \end{bmatrix}}_{A} \underbrace{\begin{bmatrix} x \\ y \end{bmatrix}}_{\mu} = \underbrace{\begin{bmatrix} x_1\cos\theta_1 - y_1\sin\theta_1 \\ x_2\cos\theta_2 - y_2\sin\theta_2 \\ \vdots \\ x_i\cos\theta_i - y_i\sin\theta_i \end{bmatrix}}_{B} \tag{2}$$

where θ_i is the AOA measurement taken at point (x_i, y_i) and μ is the location of the Target Emitter.

When a WLS algorithm is used estimate of μ becomes [1] [3] [5] [9]:

$$\hat{\mu} = (A^T W A)^{-1} A^T W B \tag{3}$$

where W is the weighting matrix.

The measurements of θ_i are noisy. These noisy measurements in A and B are correlated. This creates a bias in the system. By reducing the correlation, it may be possible to reduce the bias in the estimation of the target location.

1.3 Thesis Statement

This thesis uses a linear system of equations of AOA measurements taken by a single moving sensor to estimate the geolocation of a stationary non-cooperative emitter. The accuracy and computational time of this algorithm will be compared to a commonly used AOA WLS algorithm.

1.4 Research Activities and Scope

Matlab is used to simulate AOA measurements in order to localize a non-cooperative emitter. The simulation assumes that the sensor knows its own location (without error) at the moment each measurement is taken and the AOA measurement (Angle between the sensor and emitter) has random errors. Two metrics are used to evaluate a given method of estimation: root means squared error (RMSE) of the estimation; and the time required for the estimation. Five Thousand independent trials of the estimation process are conducted to compute the metrics.

1.5 Originality and Contribution

The standard AOA localization equation, given in equation (2) uses the measurements θ_i directly in both the A and B matrices. This thesis uses the angular measurements and manipulation of equation (1) to determine the distance between the measurement locations and the emitter. This allows the localization problem to be described as in equation (4) which will be derived in Chapter 3.

$$\underbrace{\begin{bmatrix} -2x_1 & -2y_1 & 1 \\ -2x_2 & -2y_2 & 1 \\ \vdots & \vdots & \vdots \\ -2x_M & -2y_M & 1 \end{bmatrix}}_{\text{A}} \underbrace{\begin{bmatrix} x \\ y \\ c \end{bmatrix}}_{\mu} = \underbrace{\begin{bmatrix} d_1^2 - k_1 \\ d_2^2 - k_2 \\ \vdots \\ d_M^2 - k_M \end{bmatrix}}_{\text{B}} \tag{4}$$

The A matrix in equation (4) contains only known true values. However, the A matrix in equation (2) contains measurements that include noise. The hypothesis that this thesis investigates is that the estimations calculated from equation (4) will have less bias than the standard AOA localization using equation (2).

Although this thesis focuses only on AOA localization, the use of distance instead of angular measurements permits this linear system of equations to be more easily combined with other localization methods. Both RSS and TDOA methods use calculations of distance rather than angular measurements to locate the emitter's position. The linear system of equations used in this thesis therefore enables hybrid methods using AOA and other methods to use a single formula whereby the A matrix in (4) contains only known values of measurement location coordinates. Existing hybrid formulation requires a separate A matrix for each method. Given that multiple studies have demonstrated the benefits of hybrid system, this thesis can be applied to future hybrid localization research.

Another contribution of this thesis is the use of a single moving sensor instead of M fixed sensors. Using a single sensor reduces significantly the hardware requirements (by a factor of M). Furthermore, it reduces the transmission requirements or even completely eliminates them: when we use M fixed sensors, they must send their measurements to a central processing unit, which uses these measurements to calculate the localization of the emitter. If we are using a single sensor, a single transmission link to the processing unit is required instead of M links, which significantly reduces the bandwidth requirements. If the moving sensor can process the data and compute the localization of the emitter, a transmission to a processing unit is no longer required. Hence, in situations where the transmission bandwidth is limited, using a single moving sensor is more beneficial than M fixed sensors.

It will be shown that a moving sensor can improve the localization accuracy compared to M fixed sensors by taking more than M measurements. Performance is further improved if the sensor moves and tries to encircle the emitter, which fixed sensors cannot do.

It is accepted that the benefits of moving sensors listed above apply to the geolocation of stationary emitters. If both the emitter and the sensor are in motion, then the estimation of the emitter location will be less accurate.

1.6 Organization

The remainder of this thesis is organized as follows:

Section 2 presents a review of previous research examining the types of measurements and algorithms most commonly used for passive geolocations.

The physical requirements of the measurements are examined for complexity and applicability to this thesis. This will provide the reader with an understanding of the techniques considered in this thesis.

Section 3 provides details of the research activates for this thesis. This includes the reasoning for use of simulation.

Section 4 will include a summary of the experimental results.

Section 5 will give the conclusions.

2. Literature Review

2.1 Introduction

Electromagnetic wave propagation and antenna design are well known engineering concepts. Detection and measurement of a received electromagnetic signal can therefore be considered a solved problem for the purposes of this thesis. Determining the point of origin of an incoming electromagnetic wave is a significantly different problem. In order to maintain generality, this thesis does not assume any a prior knowledge of the signal or the emitter. All measurements are made at the sensor (receiving antenna), which will be an array of antennas or a directional antenna in most cases.

2.2 Received Signal Strength (RSS)

RSS is the simplest measurement method in terms of physical requirements. Any antenna can be used as all that is required is a measurement of the received power. When the emitter's signal is received, a range calculation is made using a propagation loss formula for the transmission medium [10] [11]. Even if the power at the emitter is unknown, comparing the power received at different locations can be used to produce an estimation of the emitter's location.

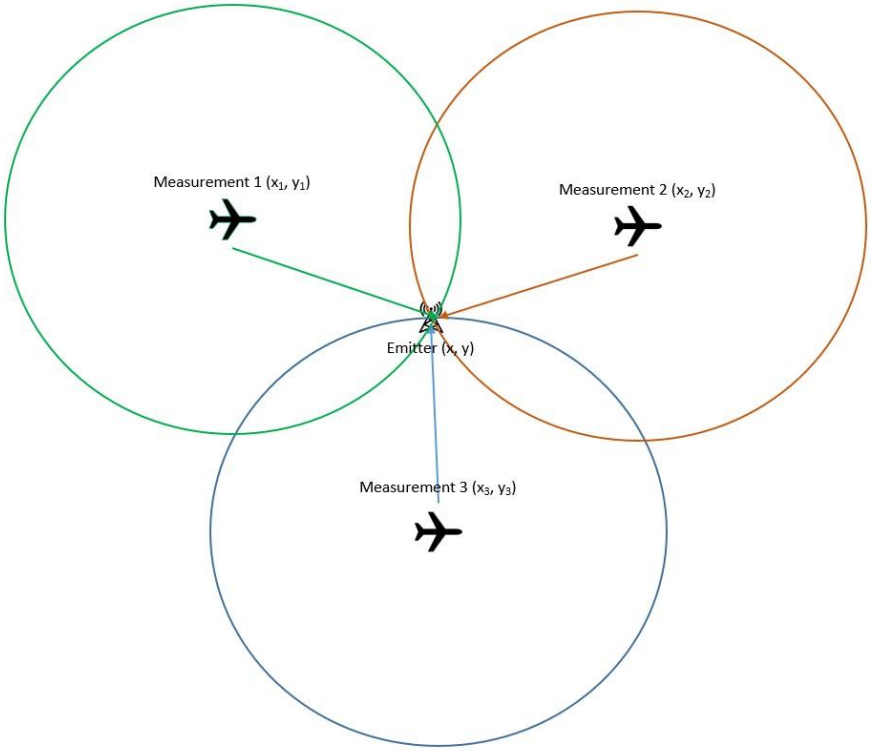


Figure 4 Two-dimensional example of RSS

A single RSS measurement provides a range to the target. If the sensing antenna is omnidirectional, the single range calculation will place the emitter on a sphere surrounding the sensor. For directional antennas, the target will be on a surface within the beam of the antenna. Figure 4 displays a two-dimensional example of RSS localization. Comparing two measurements will reduce the possible location of emitter to an arc at the intersection of the two surfaces. More measurements will further reduce the possible location of the emitter. It should be noted that when measurements are taken from multiple sensors, the individual sensors do not need to be synchronized with each other provided that the measurements can reasonably be assumed to be from the same transmission. This maintains the simplicity of RSS geolocation systems using multiple sensors.

The largest drawback to RSS is the impact of multipath fading [12]. Simple antennas at ranges close to the emitter will experience significant errors depending on whether the received transmission is in a null or a peak. Despite the

low accuracy when compared to other measurement methods, particularly for unknown emitter transmission power, the economy of antenna requirements and the simplicity of the multi-sensor network make RSS an option in applications where the transmitted power can be reasonably estimated, such as in cellular networks, navigation, and search-and-rescue.

2.3 Angle of Arrival (AOA)

Measuring the AOA of an emitter's signal requires either a directional antenna or an antenna array used with beamforming techniques to calculate the angle of the incoming signal. Although this requirement is more complex than RSS, it uses established technologies and well-known techniques. AOA requires no a priori knowledge of the emitter and can be used on any form of transmitted signal. These factors make AOA a preferred option for many applications.

Assuming the emitter's transmission is within the beamwidth of the sensor, a line of bearing to the emitter can be projected. Another measurement from another location will produce a second line of bearing. The intersection of these lines is the approximate location of the emitter. Figure 5. shows three aircraft using AOA to locate a target emitter. In ideal conditions, all three lines of bearing would intersect exactly at the location of the target. In practical applications, noise added to the transmission, measurement errors, system noise and other factors will prevent an exact line of bearing to the target. It has been shown that increasing the number of measurements will increase the accuracy of the location estimation [9].

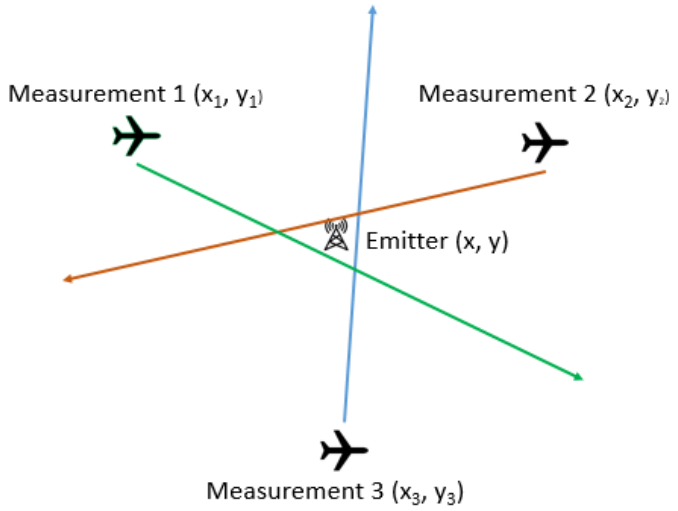


Figure 5 Two-dimensional example of AOA

It is essential that measurements be taken from different sensor locations. If the emitter is stationary, multiple measurements from a stationary sensor will continually produce the same line of bearing and not provide additional information about the emitter location. Therefore, if a single sensor is used, the sensor must be mobile and take multiple measurements at different locations during the time the emitter is transmitting. Multiple sensors taking measurements of the same transmission from different locations is a preferred method. When multiple sensors are used, the ideal placement would be the sensors surrounding the emitter at the largest intervals possible [8]. The specifics of the application may limit the possibility of surrounding the emitter. For example, base stations for cellular networks are in fixed locations, while platforms conducting search-and-rescue will likely approach from one side of the emitter.

At the small cost of more complex sensors, AOA provides greater accuracy than RSS. Requiring no a priori knowledge of the emitter, AOA can be employed to geolocate any electromagnetic emitter. The requirement for either mobility for a single sensor or the coordination of multiple sensors are additional considerations. AOA has been proven to be an effective geolocation method for cellular networks, and military applications [8] [13].

2.4 Time Difference of Arrival (TDOA)

In principle, measuring TDOA is relatively simple. If two sensors both receive a signal from the emitter, any difference in distance between the individual sensors and the emitter will mean that one sensor receives the signal earlier than the other. The difference in time of arrival divided by the speed of propagation of the signal is the range difference between the two sensors and the emitter. In practical applications, TDOA does not require a sophisticated antenna system, but it does require precise time synchronization between three or more sensors [14] [15].

Any antenna that can receive the emitter's signal can be used for TDOA. The measurement of interest is the exact time the transmission was received. Therefore, a precise clock is a requirement on all sensors. Furthermore, the clocks of all sensors must be synchronized to determine the exact difference in arrival time. When sensors make a measurement of the emitter's signal; one sensor is designated as the reference sensor and the arrival time of every other sensor are compared to the reference sensor to determine the difference of arrival time. The relative arrival time allows the computation of a relative distance to the emitter. For each pair of sensors, a hyperbola is produced with the two sensors as the foci as seen in Figure 6. Hence for 2-D geolocation a minimum of 3 sensors are required. Single senor geolocation using TDOA is possible if the emitted signal is periodic as in [16] and [17], but this in not a general case. This thesis places no conditions on the emitter other than a stationary location.

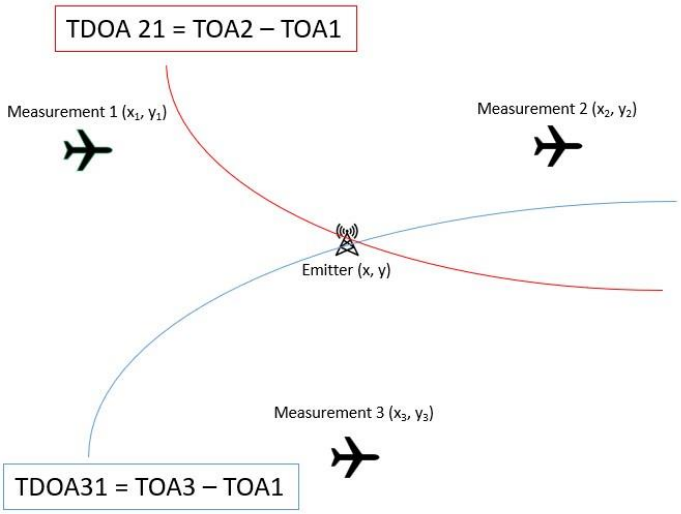


Figure 6 Two-dimensional example of TDOA

TDOA has been shown to provide accurate estimation of emitter locations. However, this method requires a minimum of three sensors to receive the emitter's signal, precise time measurements, and synchronization of all sensors. These factors make TDOA a more complex system to employ than other methods [18].

2.5 Hybrid Localization

Much of the recent work in geolocation has been focused on combining measurement methods into hybrid systems. AOA and RSS were used in [1] and [19]. TDOA and AOA have been used in a number of studies [3] [4] [5]. TDOA has received greater attention because of its accuracy and application for cellular telephone networks. TDOA does require a minimum of three sensors to locate the emitter and all sensors must be precisely synchronized. This increases both the cost and complexity of the sensor system.

2.6 Selection of AOA Measurement Technique

As discussed above, the motivation for this thesis was to develop a system of geolocation that balanced the competing requirements of simplicity and accuracy. The specific requirement is that the system is to be employable by a single moving sensor. This requirement eliminated TDOA as an option, because TDOA requires a

minimum of three sensors. Although RSS is the simplest, it is the least accurate. AOA was therefore selected as the preferred measurement technique.

2.7 Estimation Algorithms

No matter what measurement method, or combination of methods used, the system must have an algorithm that uses the measurements to estimate the location of the target emitter. Maximum Likelihood (ML) estimators have been used in [4] [9] [20]. Weighted Least Squares (WLS) were used in [1] [2] [3] [21] [20]. Both the ML and the WLS estimators assume the localization problem to be non-linear, and therefore use relatively complex algorithms to solve for the emitter's location.

3. Research

The algorithm in this thesis uses the geometry shown in Figure 3 to estimate the distance between the known location of each measurement (x_i, y_i) and the unknown location of the emitter (x, y). This thesis was conducted in a two-dimensional (x, y) plane. It is theoretically possible to extend this research to a three-dimensional space by using the both an (x, y) plane for an azimuth and either an (x, z) or (y, z) plane for an elevation and then combining the results for a three-dimensional geolocation. The general distance between the emitter and the sensor I is:

$$d_i = \sqrt{(x - x_i)^2 + (y - y_i)^2} \tag{5}$$

Squaring the distance and expanding yields:

$$d_i^2 = x^2 + y^2 - 2xx_i - 2yy_i + x_i^2 + y_i^2 \tag{6}$$

Let

$$c = x^2 + y^2 \tag{7}$$

and

$$k_i = x_i^2 + y_i^2 \tag{8}$$

Then equation (5) becomes

$$d_i^2 = c - 2xx_i - 2yy_i + k_i \tag{9}$$

Moving k_i to the other side and expanding for multiple measurement locations, the matrix form of equation (6) becomes:

$$\underset{A}{\begin{bmatrix} -2x_1 & -2y_1 & 1 \\ -2x_2 & -2y_2 & 1 \\ \vdots & \vdots & \vdots \\ -2x_M & -2y_M & 1 \end{bmatrix}} \underset{\mu}{\begin{bmatrix} x \\ y \\ c \end{bmatrix}} = \underset{B}{\begin{bmatrix} d_1^2 - k_1 \\ d_2^2 - k_2 \\ \vdots \\ d_M^2 - k_M \end{bmatrix}} \tag{10}$$

The solution for equation (10) is:

$$\hat{\mu} = (A^T A)^{-1} A^T B \tag{11}$$

3.1 Initial Research

The vector B in equation (10) contains unknown values. Therefore, before an estimate of μ can be made, an estimation of d_i^2 is required. This distance can be determined using the Law of Sines and the geometry in Figure 7.

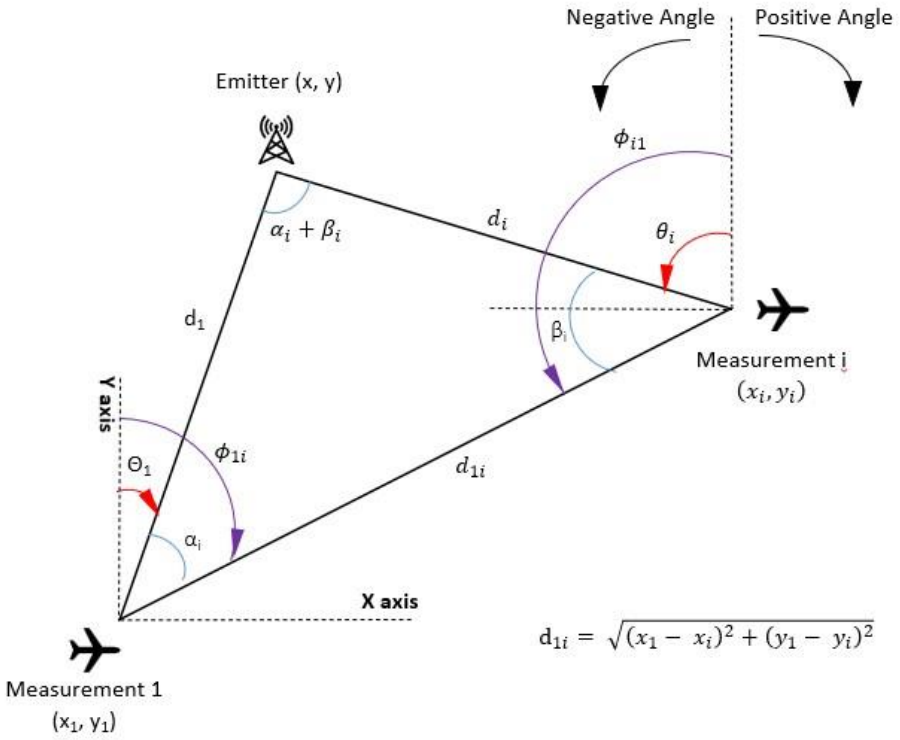


Figure 7 Angles used to determine distance to emitter

This thesis uses the AOA measurements θ_i. It is assumed that the sensor platform has been equipped to make these measurements. The physical requirements and methods on making these measurements is beyond the scope of this thesis. It is also assumed that the sensor locations at: (x_1, y_1), (x_i, y_i) are known without errors. For M measurements, a total of (M-1) triangles are possible. Using the first measurement location as a reference the following calculations are made:

The angle from the first measurement location to the i^{th} measurement location is calculated using the intermediate angle ϕ:

$$\phi_{1i} = \arctan\left(\frac{x_i - x_1}{y_i - y_1}\right) \tag{12}$$

As noted in Figure 7, the ϕ are calculated to be between 0^0 and 180^o but a positive or negative sign is assigned to the angle to ensure that later equations account for the true geometry. The inside angles of the triangle in Figure 7 are:

$$\alpha_i = \phi_{1i} - \theta_1 \tag{13}$$

and

$$\beta_i = \theta_i - \phi_{i1} \tag{14}$$

The law of sines states that

$$\frac{d_{1i}}{\sin(\alpha_i + \beta_i)} = \frac{d_i}{\sin \alpha_i} = \frac{d_1}{\sin \beta_i} \tag{15}$$

Therefore

$$d_1 = \frac{d_{1i} \sin \beta_i}{\sin(\alpha_i + \beta_i)} \tag{16}$$

Similarly:

$$d_i = \frac{d_{1i} \sin \alpha_i}{\sin(\alpha_i + \beta_i)} \tag{17}$$

Equation (15) allows for the calculation of a unique d_1 value for each pair of measurements. For the sake of efficiency d_1 is only calculated once.

Any practical system will contain some noise and errors. This thesis attributes all system errors (including system noise, antenna imperfections, etc) to measurement errors and represents these error as zero mean Gaussian White Noise (WGN). It is also assumed that the error at different measurement locations are uncorrelated. Thus, a noisy AOA measurement is:

$$\hat{\theta}_i = \theta_i + \varepsilon_i \tag{18}$$

where $\hat{\theta}_i$ is the measured value; θ_i is the true values and ε_i is the error (noise). All values calculated with the noisy θ measurements will have the same error. Appling this to equations (16):

$$\hat{d}_1 = \frac{d_{1i} \sin \hat{\beta}_i}{\sin\left(\hat{\alpha}_i + \hat{\beta}_i\right)} = d_1 + \delta_1 \tag{19}$$

where $\hat{d}_1$ is the calculated value; d_1 is the true values and δ_1 is the error.

A Taylor Series expansion is used to express error as a function of the AOA measurement noise.

$$\hat{d}_1 \cong d_1 + \left(\frac{\partial d_1}{\partial \theta_1}\Big|_{\theta_1,\theta_i}\right)\varepsilon_1 + \left(\frac{\partial d_1}{\partial \theta_i}\Big|_{\theta_1,\theta_i}\right)\varepsilon_i \tag{20}$$

where

$$\frac{\partial d_1}{\partial \theta_1} = +\frac{d_{1i}\sin\hat{\beta}_i}{\sin^2(\hat{\alpha}_i+\hat{\beta}_i)}\cos(\hat{\alpha}_i+\hat{\beta}_i) = F_1(i) \tag{21}$$

because $\frac{\partial \alpha_i}{\partial \theta_1} = -1$ (see (13)) and

$$\frac{\partial d_1}{\partial \theta_i} = \frac{d_{1i}\cos\hat{\beta}_i}{\sin(\hat{\alpha}_i+\hat{\beta}_i)} - \frac{d_{1i}\sin(\hat{\beta}_i)}{\sin^2(\hat{\alpha}_i+\hat{\beta}_i)}\cos(\hat{\alpha}_i+\hat{\beta}_i) = F_2(i) \tag{22}$$

It is important to note that both $F_1(i)$ and $F_2(i)$ are both functions of the AOA measurements and must be calculated for each pair of measurements. The first measurement location is used as a refence, and the system has M measurements; i = 2, …, M

Substituting equations (21) and (22) into equation (19) gives:

$$\hat{d}_1 \cong d_1 + F_1(i)\varepsilon_1 + F_2(i)\varepsilon_i \tag{23}$$

Comparing equation (23) to equation (19) implies:

$$\delta_1 = F_1(i)\varepsilon_1 + F_2(i)\varepsilon_i \tag{24}$$

A similar Taylor Series expansion for the values of d_i:

$$\hat{d}_i = \frac{d_{1i}\sin\hat{\alpha}_i}{\sin(\hat{\alpha}_i+\hat{\beta}_i)} = d_i + \delta_i \tag{25}$$

and

$$\hat{d}_i = d_i + \left(\frac{\partial d_i}{\partial \theta_1}\Big|_{\theta_1,\theta_i}\right)\varepsilon_1 + \left(\frac{\partial d_i}{\partial \theta_i}\Big|_{\theta_1,\theta_i}\right)\varepsilon_i \tag{26}$$

where

$$\frac{\partial d_i}{\partial \theta_1} = -\frac{d_{1i}\cos\hat{\alpha}_i}{\sin(\hat{\alpha}_i+\hat{\beta}_i)} + \frac{d_{1i}\sin\hat{\alpha}_i}{\sin^2(\hat{\alpha}_i+\hat{\beta}_i)}\cos(\hat{\alpha}_i+\hat{\beta}_i) = G_1(i) \tag{27}$$

because $\frac{\partial \alpha_i}{\partial \theta_1} = -1$ (see (13)) and

$$\frac{\partial d_i}{\partial \theta_i} = -\frac{d_{1i} \sin \hat{\alpha}_i}{\sin^2(\hat{\alpha}_i + \hat{\beta}_i)} \cos(\hat{\alpha}_i + \hat{\beta}_i) = G_2(i) \tag{28}$$

As above i = 2, …, M and

$$\hat{d}_i \cong d_i + G(i)\varepsilon_1 + G_2(i)\varepsilon_i \tag{29}$$

Also

$$\delta_i = G_1(i)\varepsilon_1 + G_2(i)\varepsilon_i \tag{30}$$

3.2 Derivation of the 1st Stage Weighting Matrix

Using the estimated values in equation (19) and (25) in the original matrix form of equation (6):

$$\underset{A}{\begin{bmatrix} -2x_1 & -2y_1 & 1 \\ -2x_2 & -2y_2 & 1 \\ \vdots & \vdots & \vdots \\ -2x_i & -2y_i & 1 \end{bmatrix}} \underset{\mu}{\begin{bmatrix} x \\ y \\ c \end{bmatrix}} = \underset{B}{\begin{bmatrix} (\hat{d}_1 - \delta_1)^2 - k_1 \\ (\hat{d}_2 - \delta_2)^2 - k_2 \\ \vdots \\ (\hat{d}_i - \delta_i)^2 - k_i \end{bmatrix}} \tag{31}$$

Expanding the squares in B gives:

$$d_i^2 = (\hat{d}_i - \delta_i)^2 = \hat{d}_i^2 - 2\hat{d}_i\delta_i + \delta_i^2 \tag{32}$$

Assuming that δ_i is small so that δ_i^2 can be neglected, substituting (32) into (31) yields:

$$\underset{A}{\begin{bmatrix} -2x_1 & -2y_1 & 1 \\ -2x_2 & -2y_2 & 1 \\ \vdots & \vdots & \vdots \\ -2x_i & -2y_i & 1 \end{bmatrix}} \underset{\mu}{\begin{bmatrix} x \\ y \\ c \end{bmatrix}} = \underset{B'}{\begin{bmatrix} \hat{d}_1^2 - k_1 \\ \hat{d}_2^2 - k_2 \\ \vdots \\ \hat{d}_i^2 - k_i \end{bmatrix}} - \underset{\delta}{\begin{bmatrix} 2d_1\delta_1 \\ 2d_2\delta_2 \\ \vdots \\ 2d_i\delta_i \end{bmatrix}} \tag{33}$$

We let the weight matrix be the expected value of $\delta\delta^T$, i.e.,

$$E\{\delta\delta^T\} = E\left\{\begin{bmatrix}2d_1\delta_1\\2d_2\delta_2\\\vdots\\2d_i\delta_i\end{bmatrix}[2d_1\delta_1 \quad 2d_2\delta_2 \quad \dots \quad 2d_i\delta_i]\right\} \tag{34}$$

Expanding equation (34) for a 5-measurement system:

$$E\{\delta\delta^T\} = 4E\left\{\begin{bmatrix} d_1^2\delta_1^2 & d_1d_2\delta_1\delta_2 & d_1d_3\delta_1\delta_3 & d_1d_4\delta_1\delta_4 & d_1d_5\delta_1\delta_5\\ d_1d_2\delta_1\delta_2 & d_2^2\delta_2^2 & d_2d_3\delta_2\delta_3 & d_2d_4\delta_2\delta_4 & d_2d_5\delta_2\delta_5\\ d_1d_3\delta_1\delta_3 & d_2d_3\delta_2\delta_3 & d_3^2\delta_3^2 & d_3d_4\delta_3\delta_4 & d_3d_5\delta_3\delta_5\\ d_1d_4\delta_1\delta_4 & d_2d_4\delta_2\delta_4 & d_3d_4\delta_3\delta_4 & d_4^2\delta_5^2 & d_4d_5\delta_4\delta_5\\ d_1d_5\delta_1\delta_5 & d_2d_5\delta_2\delta_5 & d_3d_5\delta_3\delta_5 & d_4d_5\delta_4\delta_5 & d_5^2\delta_5^2\end{bmatrix}\right\} \tag{35}$$

Or:

$$4E\left\{\Lambda\underbrace{\begin{bmatrix}\delta_1^2 & \delta_1\delta_2 & \delta_1\delta_3 & \delta_1\delta_4 & \delta_1\delta_5\\ \delta_1\delta_2 & \delta_2^2 & \delta_2\delta_3 & \delta_2\delta_4 & \delta_2\delta_5\\ \delta_1\delta_3 & \delta_2\delta_3 & \delta_3^2 & \delta_3\delta_4 & \delta_3\delta_5\\ \delta_1\delta_4 & \delta_2\delta_4 & \delta_3\delta_4 & \delta_5^2 & \delta_4\delta_5\\ \delta_1\delta_5 & \delta_2\delta_5 & \delta_3\delta_5 & \delta_4\delta_5 & \delta_5^2\end{bmatrix}}_{\Omega}\Lambda\right\} \tag{36}$$

Where:

$$\Lambda = \begin{bmatrix} d_1 & 0 & 0 & 0 & 0\\ 0 & d_2 & 0 & 0 & 0\\ 0 & 0 & d_3 & 0 & 0\\ 0 & 0 & 0 & d_4 & 0\\ 0 & 0 & 0 & 0 & d_5\end{bmatrix} \tag{37}$$

Hence:

$$\Phi = E\{\delta\delta^T\} = 4\Lambda\mathrm{E}\{\Omega\}\Lambda \tag{38}$$

Noting that Ω is a symmetrical matrix means that only one half of the elements need to be calculated. The estimated value of the first element of Ω is:

$$E\{\delta_1^2\} = E\{(F_1(i)\varepsilon_1 + F_2(i)\varepsilon_2)^2\} \tag{39}$$

$$E\{\delta_1^2\} = E\{[F_1(i)]^2\varepsilon_1^2 + 2F_1(i)F_2(i)\varepsilon_1\varepsilon_2 + [F_2(i)]^2\varepsilon_2^2\} \tag{40}$$

Because the noise is assumed to be uncorrelated WGN and $E\{\varepsilon_i^2\}$ is taken to be the variance of the noise σ_e^2, equation (40) becomes:

$$E\{\delta_1^2\} = E\{(F_1(i)\varepsilon_1 + F_2(i)\varepsilon_2)^2\} = ([F_1(i)]^2 + [F_2(i)]^2)\sigma_e^2 \quad (41)$$

The second element on the diagonal of Ω is:

$$E\{\delta_2^2\} = E\{(G_1(2)\varepsilon_1 + G_2(2)\varepsilon_2)^2\} \quad (42)$$

This expands to:

$$E\{\delta_2^2\} = E\{G_1^2(2)\varepsilon_1^2 + 2G_1(2)F_2(2)\varepsilon_1\varepsilon_2 + G_2^2(2)\varepsilon_2^2\} \quad (43)$$

Again, noting the properties of WGN, equation (43) simplifies to:

$$E\{\delta_2^2\} = (G_1(2)^2 + G_2(2)^2)\sigma_e^2 \quad (44)$$

Similarly, the remaining elements on the diagonal are:

$$E\{\delta_3^2\} = E\{(G_1(3)\varepsilon_1 + G_2(3)\varepsilon_3)^2\} = (G_1(3)^2 + G_2(3)^2)\sigma_\varepsilon^2 \quad (45)$$

$$E\{\delta_4^2\} = E\{(G_1(4)\varepsilon_1 + G_2(4)\varepsilon_4)^2\} = (G_1(4)^2 + G_2(4)^2)\sigma_\varepsilon^2 \quad (46)$$

$$E\{\delta_5^2\} = E\{(G_1(5)\varepsilon_1 + G_2(5)\varepsilon_5)^2\} = (G_1(5)^2 + G_2(5)^2)\sigma_\varepsilon^2 \quad (47)$$

The elements in the first row of the off-diagonal of Ω are:

$$\begin{gathered} E\{\delta_1\delta_2\} = E\{(F_1\varepsilon_1 + F_2\varepsilon_2)(G_1(2)\varepsilon_1 + G_2(2)\epsilon_2)\} \\ = (F_1G_1(2) + F_2G_2(2))\sigma_\varepsilon^2 \end{gathered} \quad (48)$$

$$E\{\delta_1\delta_3\} = E\{(F_1\varepsilon_1 + F_2\varepsilon_2)(G_1(3)\varepsilon_1 + G_2(3)\varepsilon_3)\} = \big(F_1G_1(3)\big)\sigma_\varepsilon^2 \quad (49)$$

$$E\{\delta_1\delta_4\} = E\{(F_1\varepsilon_1 + F_2\epsilon_2)(G_1(4)\varepsilon_1 + G_2(4)\varepsilon_4)\} = \big(F_1G_1(4)\big)\sigma_\varepsilon^2 \quad (50)$$

$$E\{\delta_1\delta_5\} = E\{(F_1\varepsilon_1 + F_2\varepsilon_2)(G_1(5)\varepsilon_1 + G_2(5)\varepsilon_5)\} = \big(F_1G_1(5)\big)\sigma_\varepsilon^2 \quad (51)$$

The remaining elements are:

$$E\{\delta_2\delta_3\} = E\{[G_1(2)\varepsilon_1 + G_2(2)\varepsilon_2][G_1(3)\varepsilon_1 + G_2(3)\varepsilon_3]\} \quad (52)$$

Which expands to

$$\begin{gathered} E\{G_1(2)G_1(3)\epsilon_1^2 + G_1(2)\varepsilon_1G_2(3)\varepsilon_3 + G_2(2)\varepsilon_2G_1(3)\varepsilon_1 \\ + G_2(2)\varepsilon_2G_2(3)\varepsilon_3\} \end{gathered} \quad (53)$$

After applying the properties of WGN

$$E\{\delta_2\delta_3\} = G_1(2)G_1(3)\sigma_\varepsilon^2 \tag{54}$$

Similarly:

$$E\{\delta_2\delta_4\} = G_1(2)G_1(4)\sigma_e^2 \tag{55}$$

$$E\{\delta_2\delta_5\} = G_1(2)G_1(5)\sigma_e^2 \tag{56}$$

$$E\{\delta_3\delta_4\} = G_1(3)G_1(4)\sigma_e^2 \tag{57}$$

$$E\{\delta_3\delta_5\} = G_1(3)G(5)\sigma_e^2 \tag{58}$$

$$E\{\delta_4\delta_5\} = G_1(4)G_1(5)\sigma_e^2 \tag{59}$$

The general form of Ω is:

$$\begin{bmatrix} (F_1^2+F_2^2)\sigma_e^2 & (G_1F_1(2)+G_2F_2(2))\sigma_e^2 & \big(G_1F_1(3)\big)\sigma_e^2 & \dots & \big(G_1F_1(i)\big)\sigma_e^2 \\ (G_1F_1(2)+G_2F_2(2))\sigma_e^2 & (G_1(2)^2+G_2(2)^2)\sigma_e^2 & G_1(2)G_1(3)\sigma_e^2 & \dots & G_1(2)G_1(i)\sigma_e^2 \\ \big(G_1F_1(3)\big)\sigma_e^2 & G_1(2)G_1(3)\sigma_e^2 & \ddots & & \vdots \\ \vdots & \vdots & G_1(3)G_1(4)\sigma_e^2 & \ddots & G_1(4)G_1(i)\sigma_e^2 \\ \big(G_1F_1(M)\big)\sigma_e^2 & G_1(2)G_1(M)\sigma_e^2 & G_1(3)G_1(M)\sigma_e^2 & & (G_1(M)^2+G_2(M)^2)\sigma \end{bmatrix} \tag{60}$$

Where $F_1(i)$, $F_2(i)$, $G_1(i)$ and $G_2(i)$ are defined in equations (21), (22), (27), and (28) respectively. The solution of μ using the 1st stage least squares weight matrix is:

$$\hat{\mu} = (A^T\Phi^{-1}A)^{-1}A^T\Phi^{-1}B' \tag{61}$$

3.3 Derivation of 2nd Stage Weight Matrix

In the 1st stage least squares, it was assumed that the unknown in $\mu = [x\ y\ c]^T$ are independent. They are actually related by (7). This relationship should be exploited to improve on the 1st stage least squares, leading to the 2nd stage least squares as follows:

The 1st stage estimate is:

$$\hat{\mu} = \begin{bmatrix} \hat{x} \\ \hat{y} \\ \hat{x}^2+\hat{y}^2 \end{bmatrix} \tag{62}$$

Equation (62) assumes that an initial estimation of the emitter location has been made. The true emitter location is (x, y) and the estimated emitter, location is $(\hat{x}, \hat{y})$ where:

$$\hat{x} = x + \xi_x \qquad where\ \xi_x\ is\ the\ error\ in\ \hat{x} \tag{63}$$

$$\hat{y} = y + \xi_y \qquad where\ \xi_y\ is\ the\ error\ in\ \hat{y} \tag{64}$$

$$\hat{x}^2 + \hat{y}^2 = \hat{c} = (x^2 + y^2) + \xi_c \qquad where\ \xi_c\ is\ the\ error\ in\ \hat{c} \tag{65}$$

Squaring $\hat{x}$ and $\hat{y}$ results in:

$$\hat{x}^2 = x^2 + 2x\xi_x + \xi_x^2 \tag{66}$$

and

$$\hat{y}^2 = y^2 + 2x\xi_y + \xi_y^2 \tag{67}$$

If the errors are small then the square of the error can be neglected. The 2nd stage can be expressed as:

$$\underset{H}{\begin{bmatrix}1 & 0\\ 0 & 1\\ 1 & 1\end{bmatrix}}\begin{bmatrix}x^2\\ y^2\end{bmatrix} = \underset{Q}{\begin{bmatrix}\hat{x}^2\\ \hat{y}^2\\ \hat{c}\end{bmatrix}} - \underset{\epsilon}{\begin{bmatrix}2x\xi_x\\ 2x\xi_y\\ \xi_c\end{bmatrix}} \tag{68}$$

We let the second stage weight matrix be the expected value of $\epsilon\epsilon^T$ which is again a symmetric matrix.

$$\Phi_2 = E\left\{\begin{bmatrix}4x^2\xi_x^2 & 4xy\xi_x\xi_y & 2x\xi_x\xi_c\\ 4xy\xi_x\xi_y & 4y^2\xi_y^2 & 2y\xi_y\xi_c\\ 2x\xi_x\xi_c & 2y\xi_y\xi_c & \xi_c^2\end{bmatrix}\right\} \tag{69}$$

Moving the constant terms outside the expected value:

$$\Phi_2 = \begin{bmatrix}2x & 0 & 0\\ 0 & 2y & 0\\ 0 & 0 & 1\end{bmatrix} E\left\{\begin{bmatrix}\xi_x^2 & \xi_x\xi_y & \xi_x\xi_c\\ \xi_x\xi_y & \xi_y^2 & \xi_y\xi_c\\ \xi_x\xi_c & \xi_y\xi_c & \xi_c^2\end{bmatrix}\right\}\begin{bmatrix}2x & 0 & 0\\ 0 & 2y & 0\\ 0 & 0 & 1\end{bmatrix} \tag{70}$$

The similarities between equation (36) and equation (70) can be seen. Because the true values of x and y are unknown, their estimated values from the 1st stage are used. Let:

$$D = \begin{bmatrix} 2\hat{x} & 0 & 0 \\ 0 & 2\hat{y} & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{71}$$

then

$$\Phi_2 = D(A^T \Phi^{-1} A)^{-1} D \tag{72}$$

The solution for the 2nd stage least squares is:

$$\begin{bmatrix} \hat{x}^2 \\ \hat{y}^2 \end{bmatrix} = (H^T \Phi_2^{-1} H)^{-1} H^T \Phi_2^{-1} Q \tag{73}$$

Equation (73) provides the square of the estimate. The 2nd stage estimate is the square root of the solution to equation (73); however, two additional implications must be accounted for. First the square root can be either positive or negative. To resolve this ambiguity, the sign of the 1st stage estimation is used for the 2nd stage solution. Secondly it is possible that the solution equation (73) will be negative number. In these cases, the 2nd stage solution is disregarded and the 1st stage solution is used.

3.4 Development of Simulation

A Matlab simulation was created that allows for the location of an unknown emitter by several measurements at known locations. The user must provide the real location of the emitter and the location of all measurements. The program calculates the AOA measurement using the true values using equation (1). Noise in the system is simulated by adding a random value to the AOA measurements. The random value has a zero mean gaussian distribution with variances specified by the use. To minimize the impact of random outlier, each algorithm was run 5000 times and the mean of the estimates were calculated.

For comparison purposes four algorithms are used: the standard AOA algorithm and three versions of the linear AOA algorithm. One using only the unweighted solution in equation (11); the second using the first stage WLS solution in equation (61); and the third using the 2nd stage WLS solution in equation (73). Each of these algorithms is passed the Data in Table 1 and must calculate all other required values.

Table 1 Data given to Algorithms

Location of Sensor (x_i, y_i)	Known without error
AOA at each measurement location (θ_i)	Measured with noise
Variance of the noise	For calculating the Expected Value

3.5 Validation

For validation purposes the RMSE of the estimation is compared to the Cramer-Rao Lower Bound (CRLB) for the same variance. The CRLB is the lowest mean squared error for an unbiased estimator. The derivation for the CRLB for AOA algorithms was adapted from [9] and is detailed in Appendix 2. If the RMSE of the algorithm approaches the root of the CRLB, then the algorithm is likely to be unbiased and approaches the best possible MSE for the noise variance given. The standard AOA algorithm has been shown to approach the CRLB [1] [9] [21]. Therefore, a comparison of the linear algorithm to the CRLB is at least as valid as a direct comparison to the Standard AOA method.

The second metric of evaluation is the time required for calculation of the estimation. The program has been set to record the time between passing the given information to each algorithm and the algorithm returning an estimate of the emitter location. In this case the linear algorithm is compared directly to the standard AOA algorithm. If less time is required by the program for computation, then the algorithm must be computationally simpler. In a practical system a computationally simpler system would require a smaller or less powerful central processing unit or allow the central processor to be used for other tasks such as communication of maneuvering control.

To simulate the movement of the platform the measurements are taken in four concentric arcs with radii of: 500 meters; 1000 meters 2000 meters and 4000 meters. These measurement configurations provide a balance between the ideal configuration of a surrounded emitter [8] and a realist arc length that a platform could travel while the emitter is transmitting. Each arc is tested separately with a single emitter in one of three positions: at the center of the arc (Figure 8); off center of arc (Figure 9); and behind the cure of the arc (Figure 10). This provides a total of 12 tests for the algorithms.

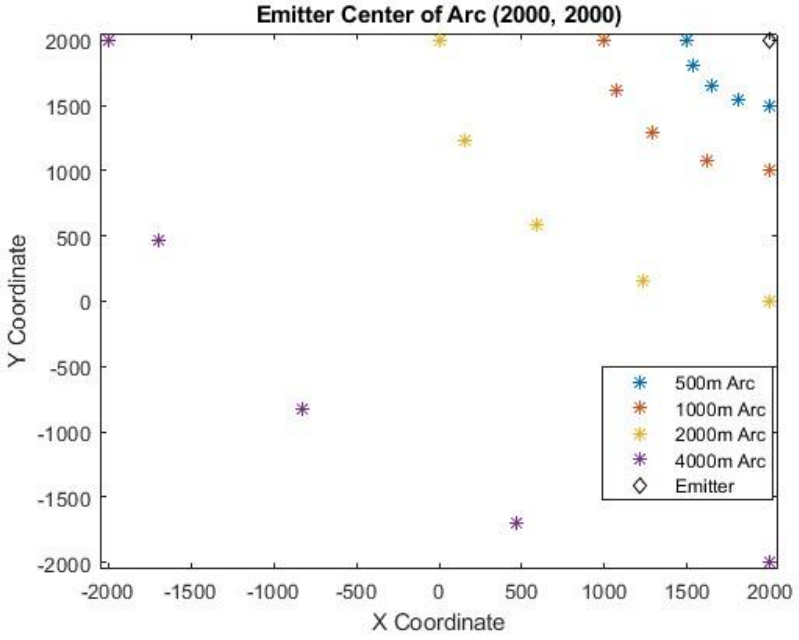


Figure 8 Senor Configuration with Emitter at Center of Arc

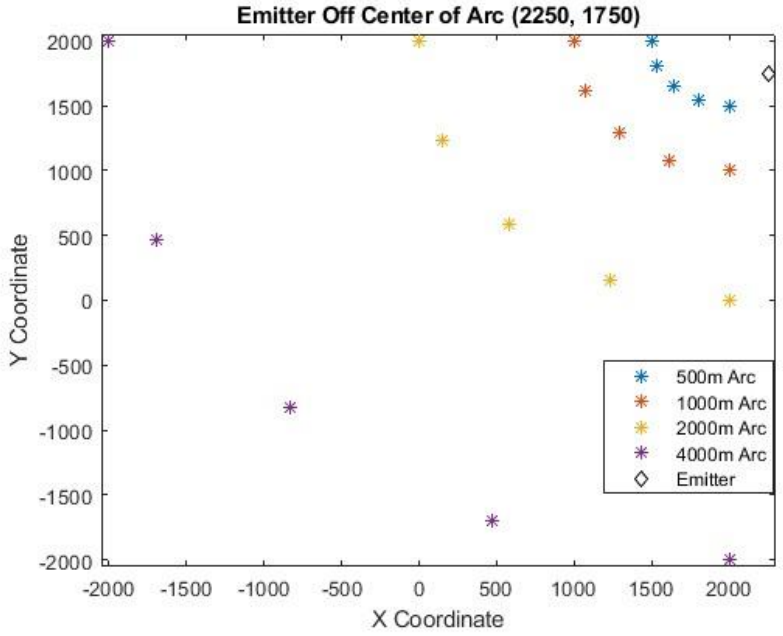


Figure 9 Senor Configuration with Emitter off Center of Arc

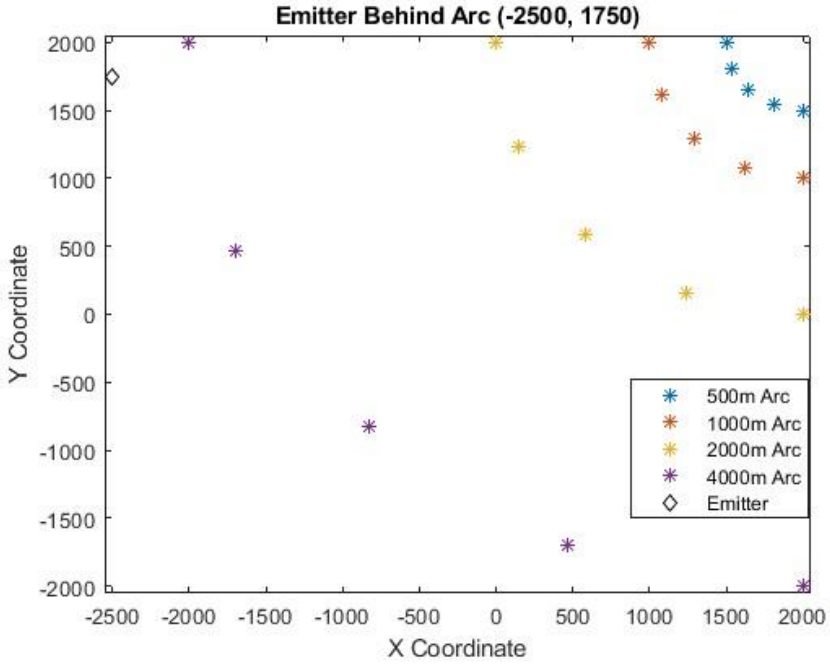


Figure 10 Sensor Configuration with Emitter Behind the Curve of the Arc

4. Experimental Results

The configurations described is Section 3.5 were designed to test the overall effectiveness of the linear algorithm. Additional experiments were conducted to optimise the system and test the effectiveness of the linear algorithm in the specific application of a moving sensor.

4.1 Optimising Sensor Order

The algorithm uses measurement pairs and the law of sines to determine the distance from the measurement locations to the emitter. The first measurement location is used as the refence point. This results in M-1 pairs for a system of M measurements. Theoretically the measurement pairs could be used in any order. However, to reduce the number of computations the distance from the first measurement to the emitter (d_1) is only calculated once and used in all other pairings. Therefore, the selection of the first pair of measurement location will impact the accuracy of the algorithm.

Noting the geometry in Figure 7 and equations (16) and (17) we see that as the distance between the two measurements locations decreases, so does the angle at the emitter ($\alpha_i + \beta_i$). The distances are calculated by dividing by Sin($\alpha_i + \beta_i$). Smaller distances between measurement locations brings the denominator closer to zero. Therefore, the best measurement pair to use for the initial calculation of d_1 is the first measurement location and the measurement location farthest from the first measurement location. This was tested at multiple distances and in multiple configurations. The results of a comparison of sensor orders with an arc radius of 2000 meters is provided in Figure 11.

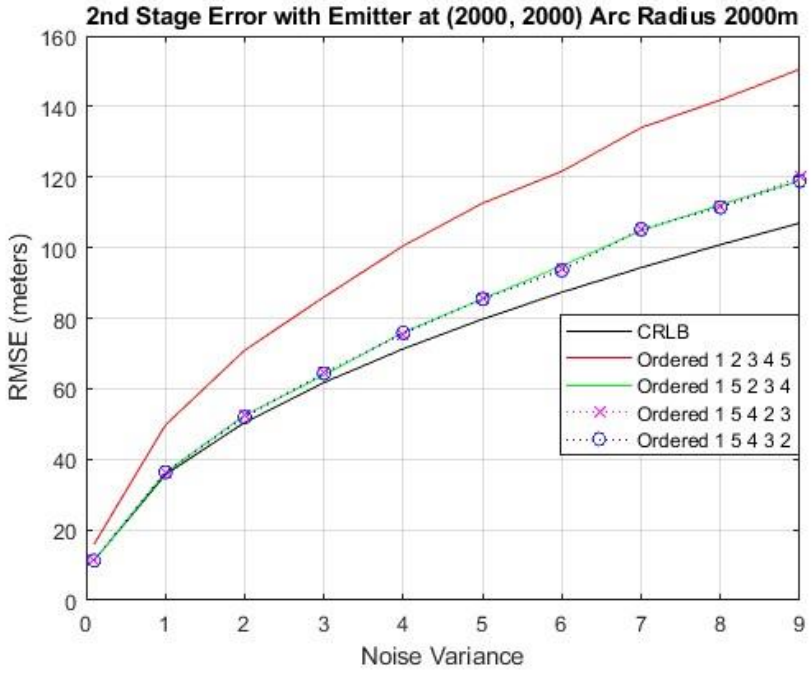


Figure 11 Impact of Sensor Orders on Accuracy

Selecting the farthest measurement from the first measurement in the first pairing increases the accuracy of estimation significantly. There is also a marginal improvement in accuracy by selecting measurement pairs from farthest to closest however, only the first pair has a significant impact. All remaining diagrams will use the sensor order 1 5 2 3 4.

4.2 Testing Emitter at Center of Arc

For validation purposes the RMSE of the linear system of equations is compared to the Standard AOA algorithm and the CRLB. The most favorable measurement configuration used has the emitter at the center of the platform's arc. The results of these test were very consistent. Figures 12 through 15 show the RMSE vise the Noise variance with an arc radius of 500 meters, 1000 meters, 2000 meters and 4000 meters respectively.

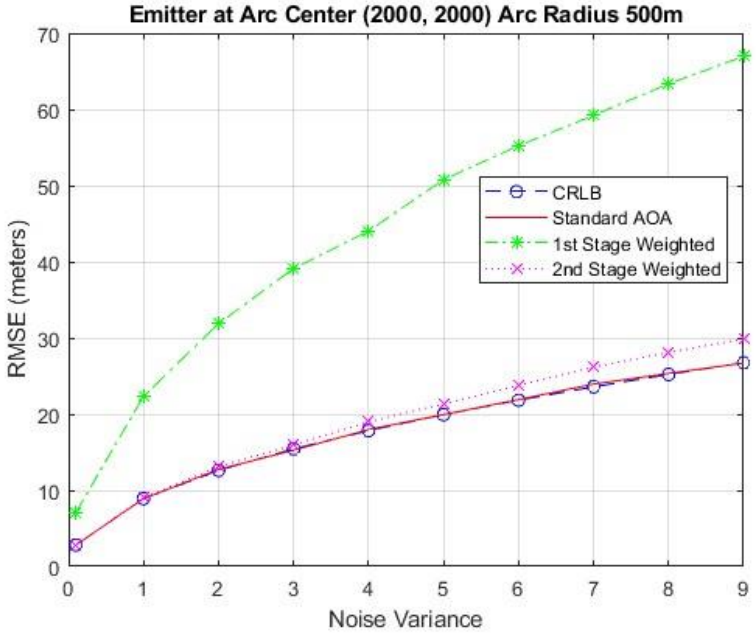


Figure 12 RMSE vs Noise Variance at Radius 500 meters

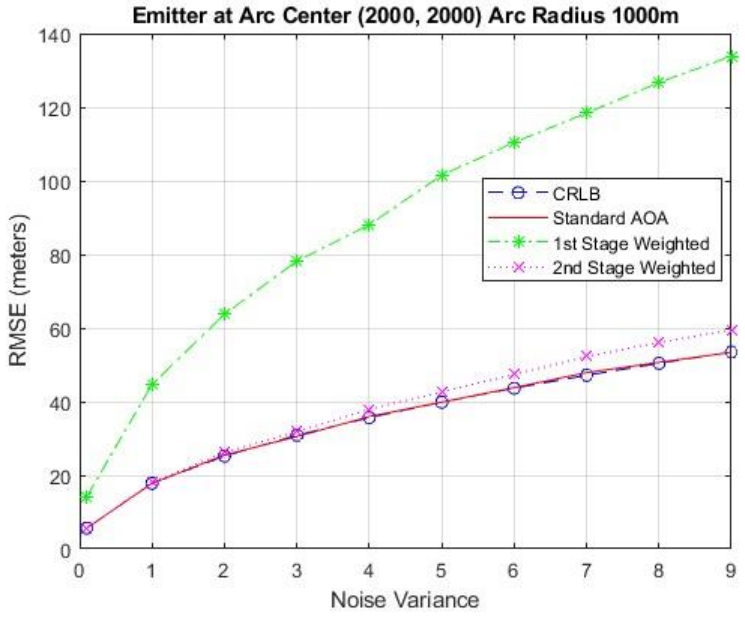


Figure 13 RMSE vs Noise Variance at Radius 1000 meters

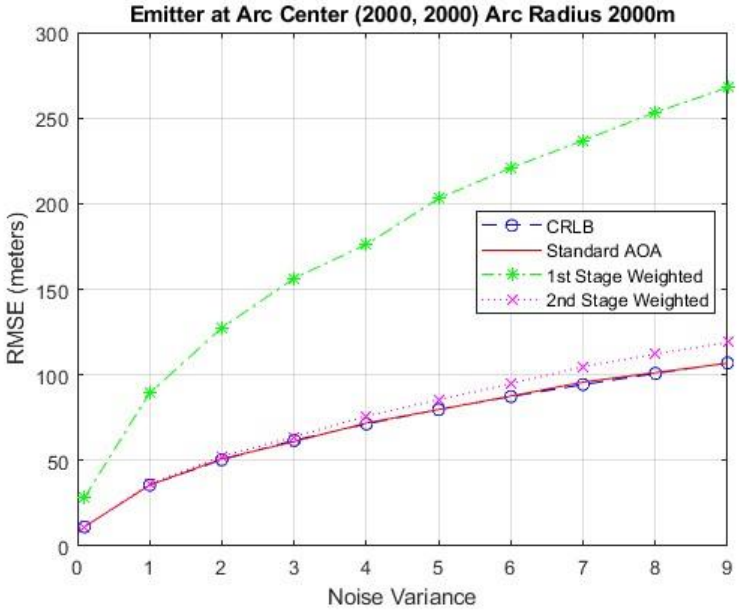


Figure 14 RMSE vs Noise Variance at Radius 2000 meters

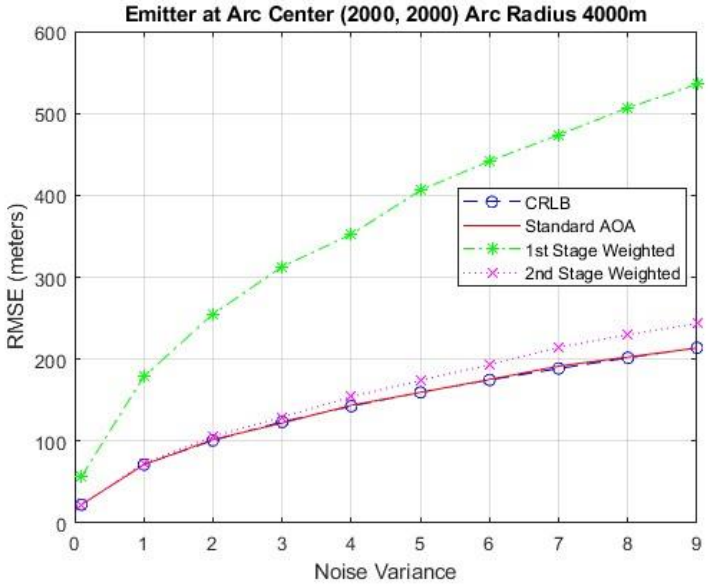


Figure 15 RMSE vs Noise Variance at Radius 4000 meters

The shape of the graph is nearly identical and the RMSE increases proportionately with the distance from the emitter as expected. Although the 1st stage estimation differs significantly from the CRLB the 2nd stage estimation is much closer.

Given that the CRLB is the theoretical limit of estimation, a relative formula for comparison has been used to determine the decibel ratio of each algorithm to the CRLB. The formula used is:

$$dB = 10 \times log_{10}\left(\frac{CRLB}{RMSE\ of\ Algorithm}\right) \tag{72}$$

A negative dB value from equation (72) indicates less accuracy than the CRLB. Figures 16, 17, 18 and 19 display the relative accuracy vs noise variance comparisons for the emitter at the centre of the arc.

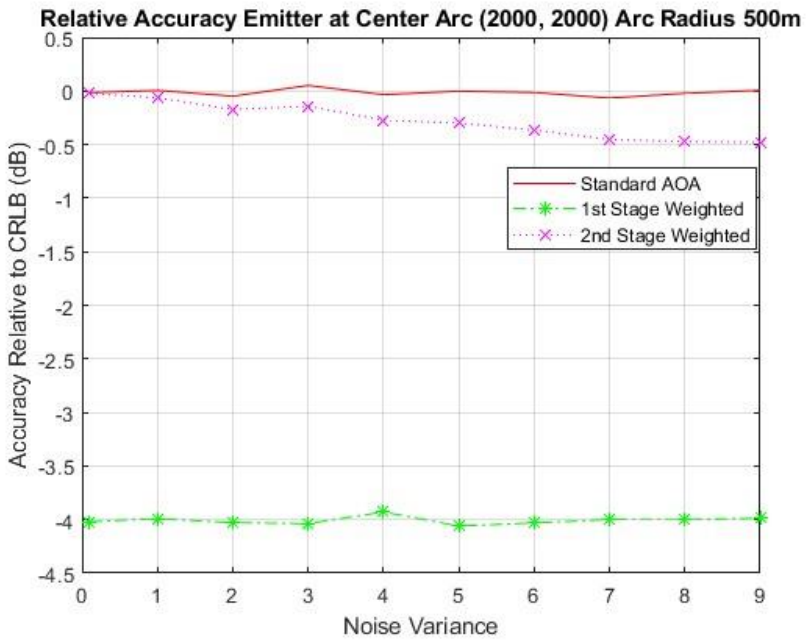


Figure 16 Relative Accuracy vs Noise Variance at Radius 500 meters

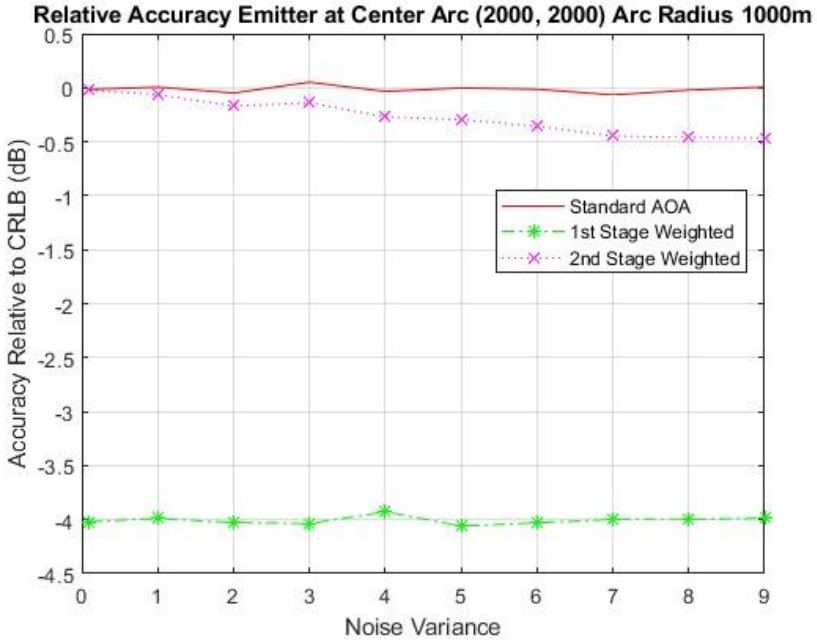


Figure 17 Relative Accuracy vs Noise Variance at Radius 1000 meters

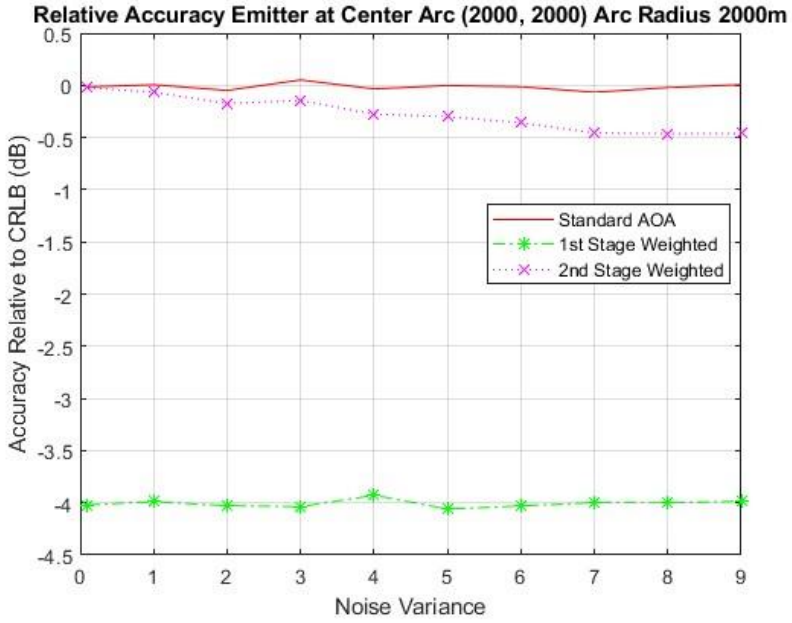


Figure 18 Relative Accuracy vs Noise Variance at Radius 2000 meters

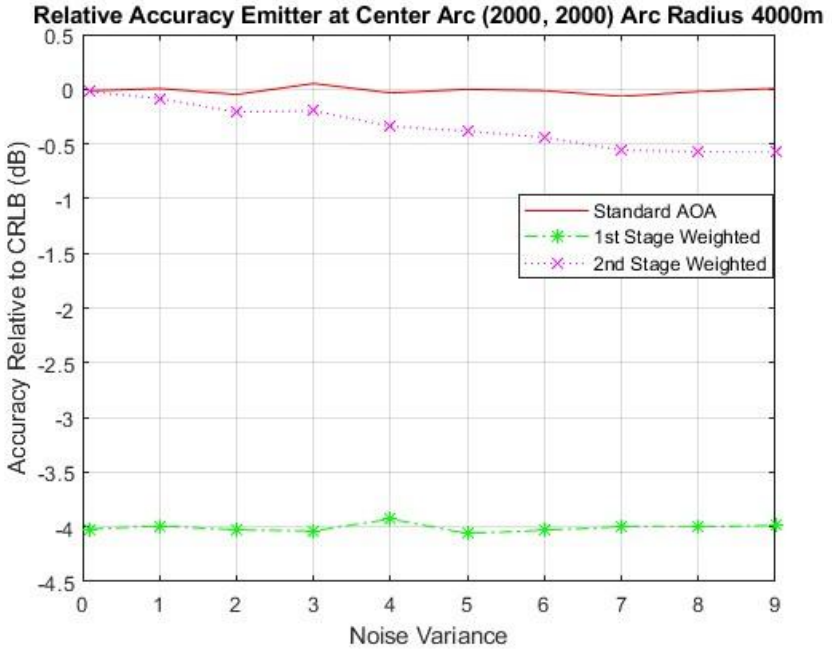


Figure 19 Relative Accuracy vs Noise Variance at Radius 4000 meters

The decibel graph again indicates that the linear system is performing consistently. The 2nd stage estimation is less accurate than the Standard AOA algorithm. However, a 0.5 dB loss of accuracy when with noise variance of 9 is likely to be acceptable in many practical applications.

To determine the computational efficiency of the linear algorithm, the time required to complete an estimation was recorded. The mean of 5000 independent trials for the measurement configuration with the emitter at the center of the arc is given in Figures 20, 21, 22, and 23. The processor used for all tests was a 2.10 GHz AMD Ryzen 5 with 8 GB of RAM.

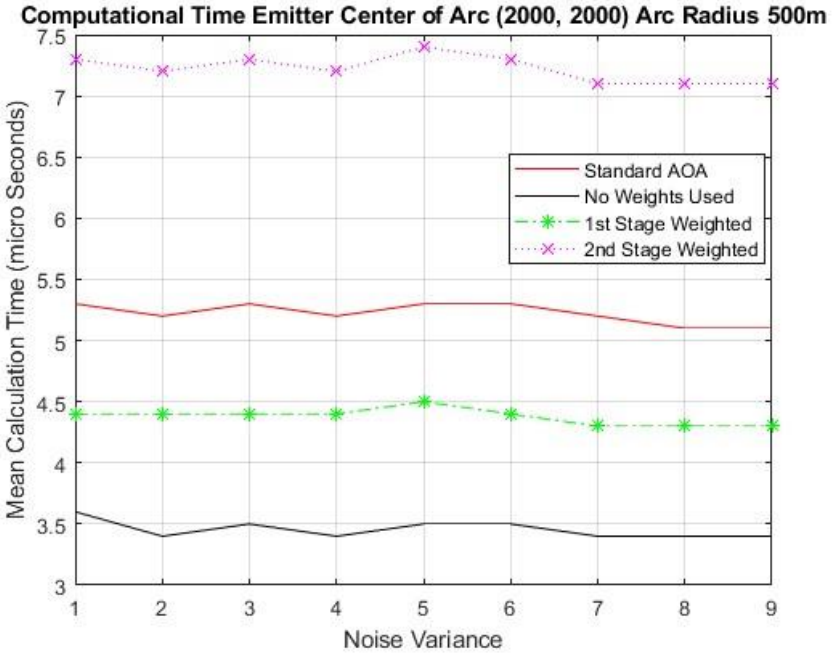


Figure 20 Computational Time with Emitter at Center of Arc and Radius 500m

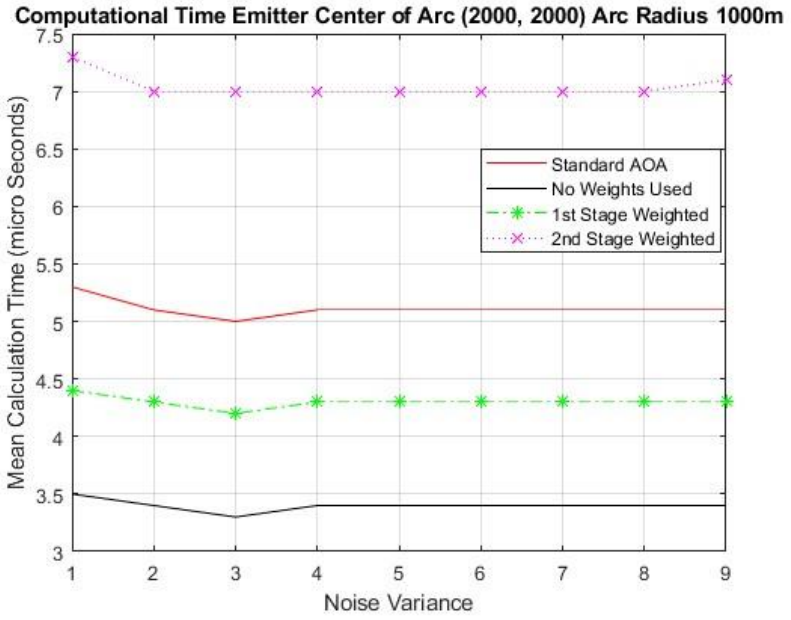


Figure 21 Computational Time with Emitter at Center of Arc and Radius 1000m

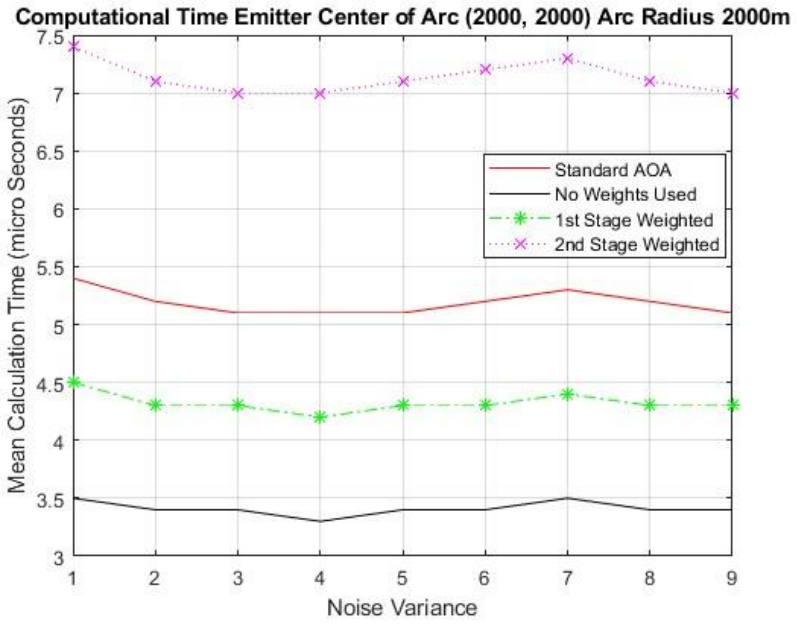


Figure 22Computational Time with Emitter at Center of Arc and Radius 2000m

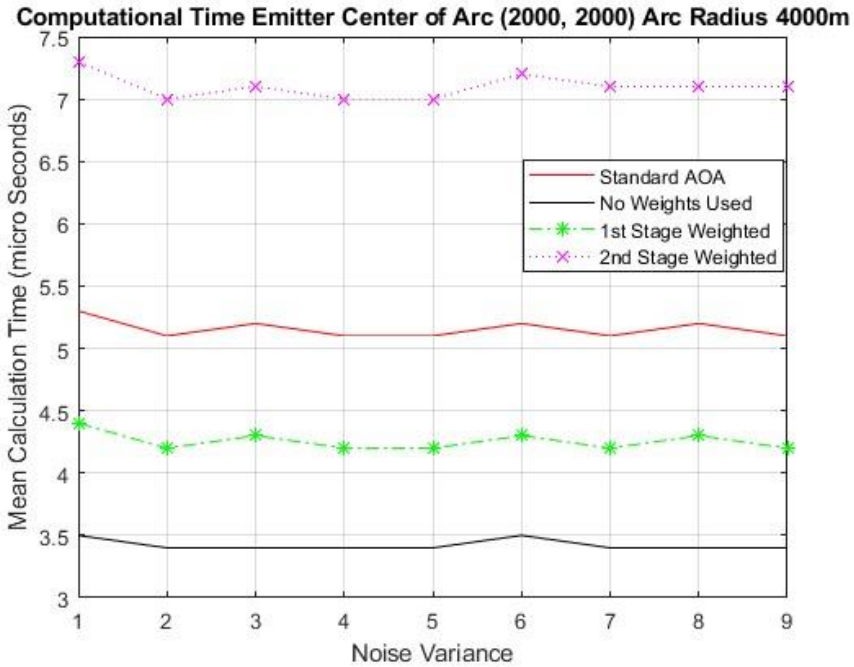


Figure 23 Computational Time with Emitter at Center of Arc and Radius 4000m

There is a consistency with the computational time: 3.5 µs when the linear system is used without a weight matrix; just under 4.5 µs when only the 2st stage is used; about 5.25 for the Standard AOA algorithm and 7 µs when the 2nd stage weight matrix is used in the linear system of equations.

4.3 Emitter Off Center of Arc

In a practical geolocation scenario, it is unlikely that the emitter will be perfectly centered on the arc formed by the sensors. To test the effectiveness of the linear system of equations with an off-centre emitter, the emitter was changed to coordinated (2250, 1750). Figures 24 through 27 provide the relative accuracy of the algorithm, as described by equation72).

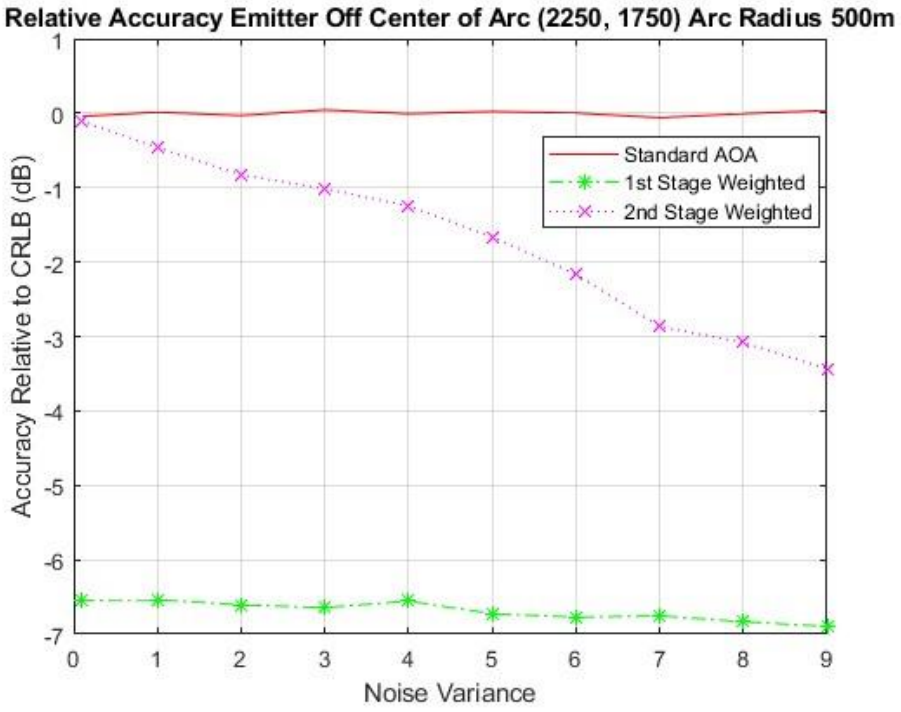


Figure 24 Relative Accuracy for Off Center Emitter Arc Radius 500m

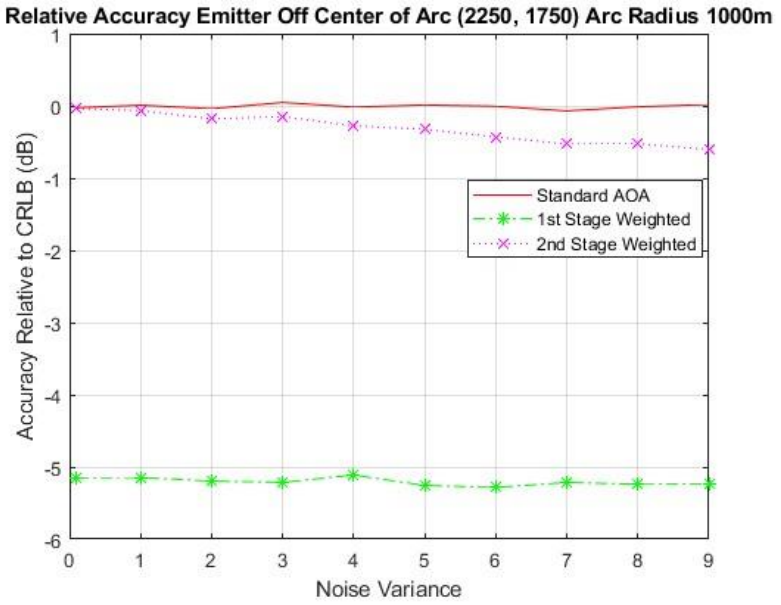


Figure 25 Relative Accuracy for Off Center Emitter Arc Radius 1000m

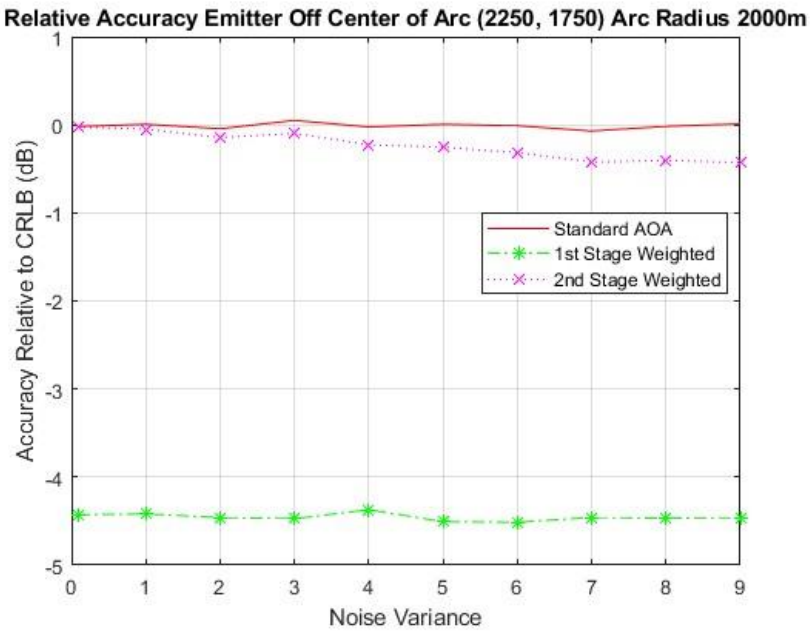


Figure 26 Relative Accuracy for Off Center Emitter Arc Radius 2000m

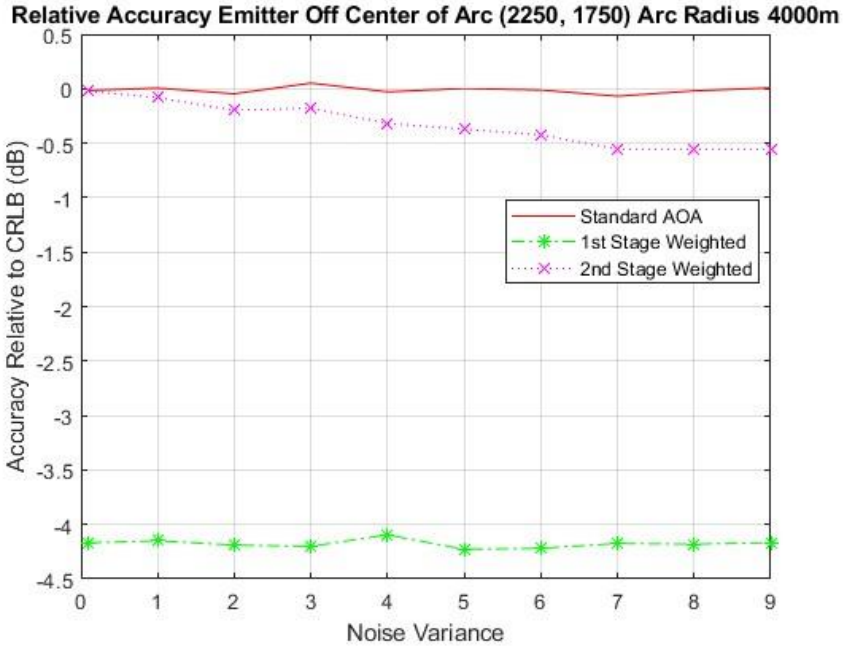


Figure 27 Relative Accuracy for Off Center Emitter Arc Radius 4000m

The poorest accuracy is seen in Figure 24. Of the four sensor arcs used in this test, the emitter is most off centre for the 500m arc and closest to centre for the 4000m arc. It follows that the 500m would have the lowest accuracy and the 4000m arc the highest accuracy for this configuration.

The impact of the off-centre emitter can most clearly be seen in the results for the 1st stage estimation. In the first test, with the emitter centered (figures 16-19) the 1st stage estimation was consistently at -4dB. In the off-centre test the 1st stage estimation moves closer to -4dB as the arc radius increases, and the emitter is more centered.

The computational times for this test are very similar to the time from the first test. Figures 28 and 29 provide the times for the 500m arch and the 1000m respectively.

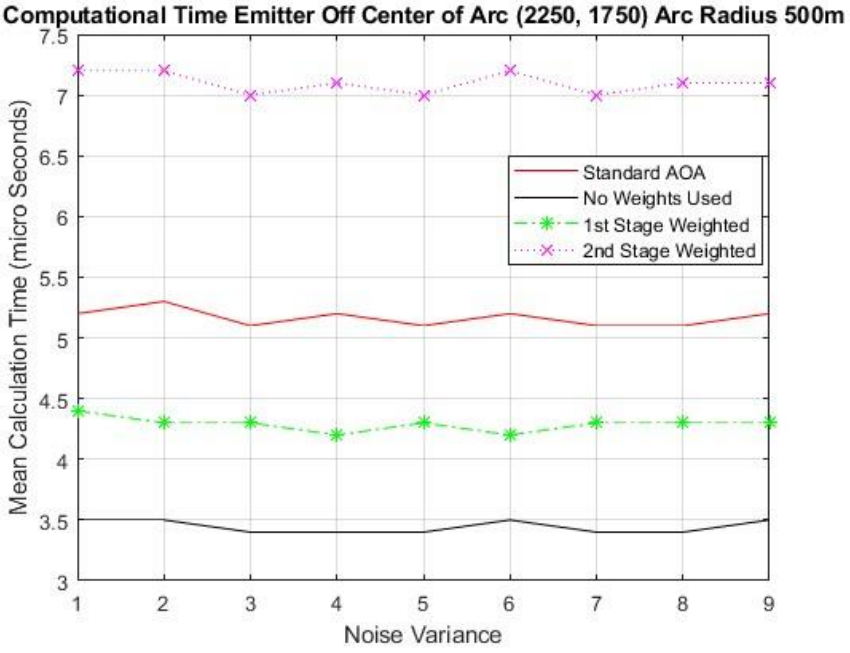


Figure 28 Computational Time with Emitter Off Center of Arc and Range 500m

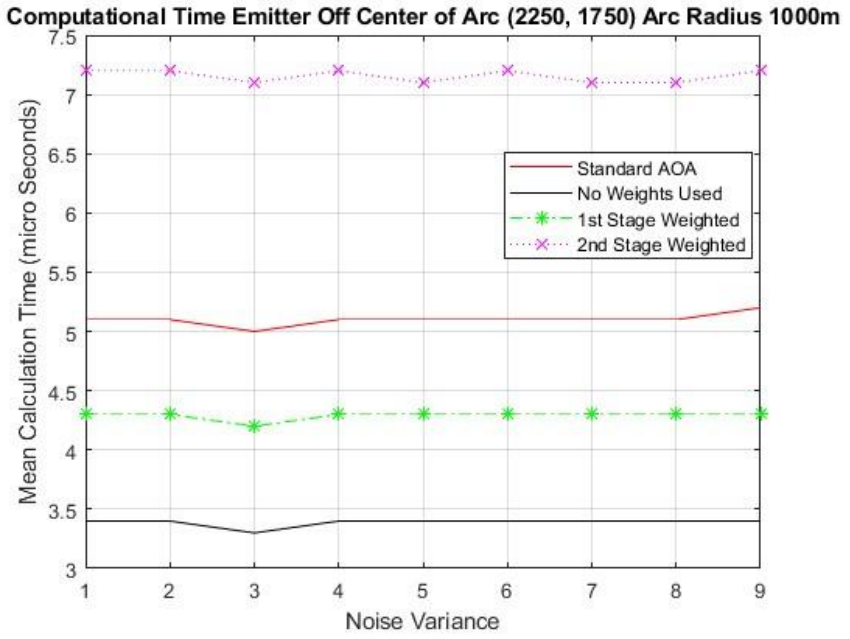


Figure 29 Computational Time with Emitter Off Center of Arc and Range 1000m

These results indicate that the time required to complete an estimation is consistent provided the emitter is within the effective field of view of the sensor configuration.

4.4 Emitter Behind Arc

The third test conducted placed the emitter behind the arc formed by the measurements. This configuration was designed to test the system under unfavourable conditions. The relative accuracy for the 1000 meter, 2000 meter and 4000 meter arcs are presented in Figures 30 through 32 respectively.

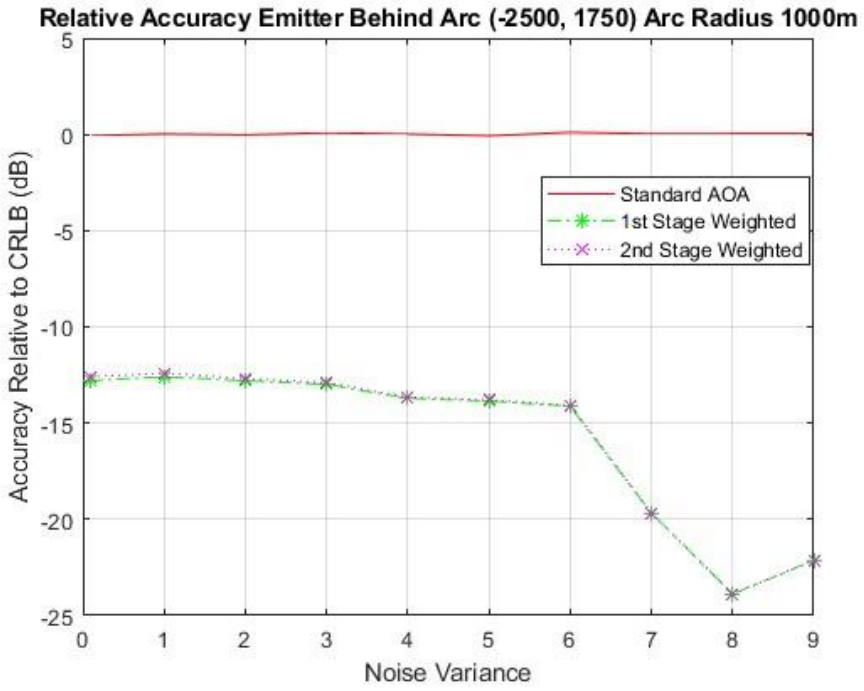


Figure 30 Relative Accuracy for Emitter Behind Arc Radius 1000m

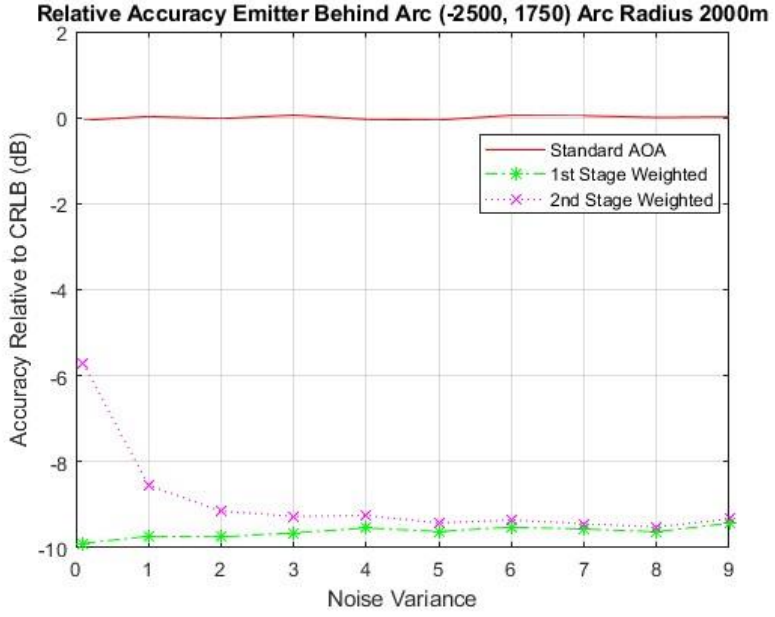


Figure 31 Relative Accuracy for Emitter Behind Arc Radius 2000m

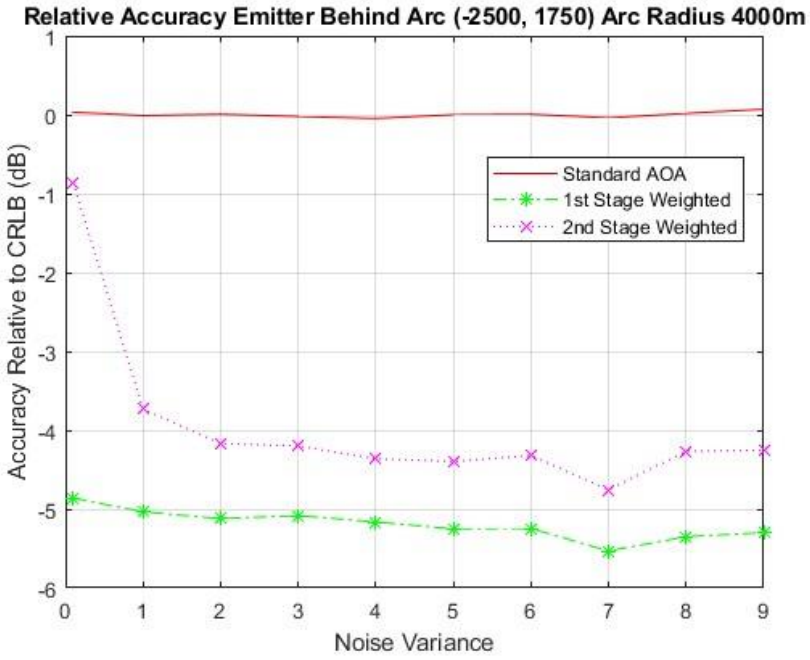


Figure 32 Relative Accuracy for Emitter Behind Arc Radius 4000m

With the emitter behind the arc, it is closest to the largest arc and farthest from the smallest arc. It was therefore expected that the 4000 meter arc would have the best results. In Figures 30 and 21 we see that there is minimal difference between the 1st stage estimation and the 2ns stage estimation. This is an indication that the estimation is inaccurate. A similar pattern can be seen in Figures 24 and 27. As the noise variance increases the 1st stage estimation accuracy remains constant but the relative accuracy of the 2nd stage estimation decreases, approaching the 1st stage estimation.

The reason the 2nd stage estimation approaches the 1st stage estimation for unfavourable sensor-emitter configuration is a result of equation (71). As noted above, it is possible for the 2nd stage solution to include a negative x^2 or y^2, particularly for noisy measurements or unfavourable configurations. When a negative x^2 or y^2 is calculated, the algorithm rejects the 2nd stage estimation and returns the 1st stage estimation. Therefore, the worst possible result is the 1st stage estimation.

As with the above tests the computation times were recorded. Figures 33 through 35 give the results.

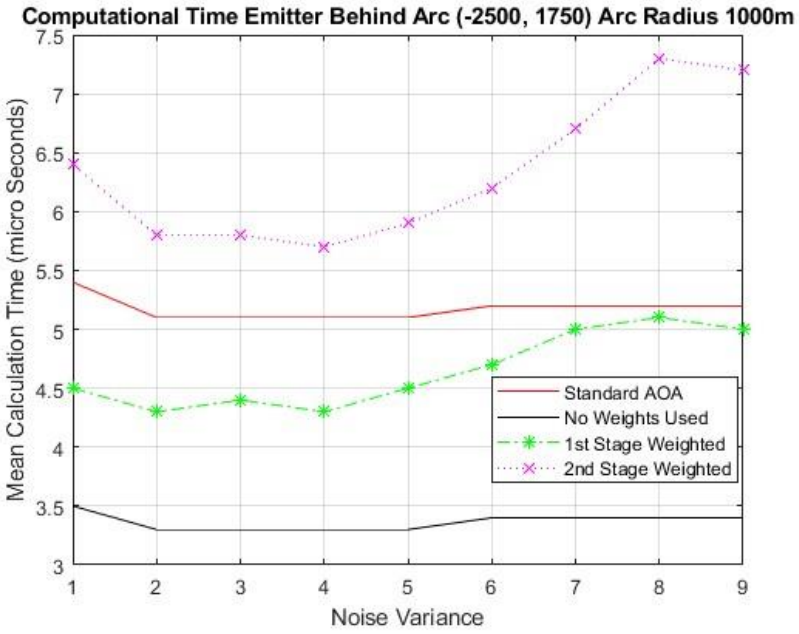


Figure 33 Computational Time with Emitter Behind Arc and Radius 1000m

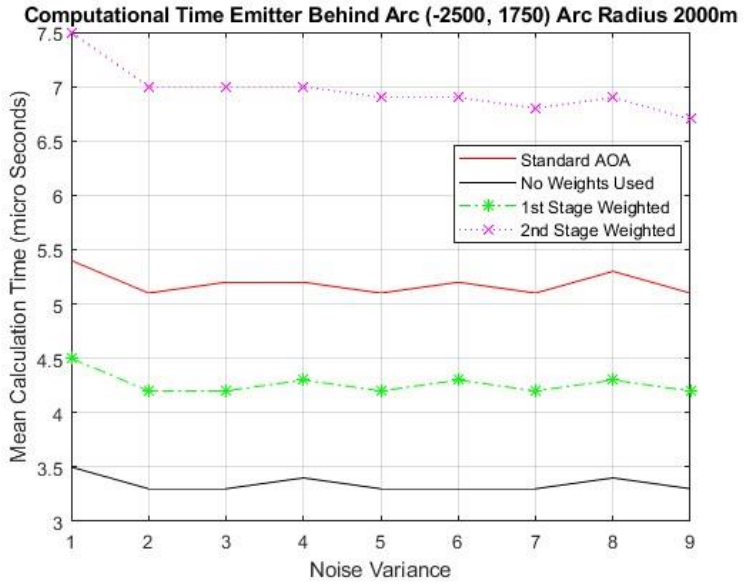


Figure 34 Computational Time with Emitter Behind Arc and Radius 2000m

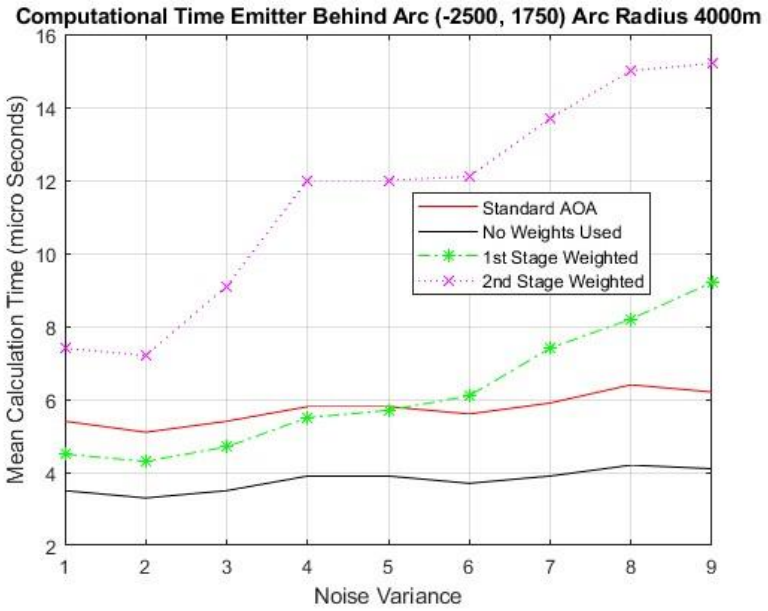


Figure 35 Computational Time with Emitter Behind Arc and Radius 4000m

It was observed with both the 1000 meter arc and the 4000 meter the computational time require increased with the variance of the noise. This is

attributed to the emitter being at an extreme range for the 1000 meter arc and an extreme angle for the 4000 meter arc. The complete failure of the 2nd stage estimation for the 1000 meter arc (Figure 30) confirms that the emitter is beyond an acceptable range. Although Figure 32 shows that the 2nd stage was not completely rejected for the 4000 meter arc, the relatively small difference between the 1st and 2nd stage indicates this configuration is approaching a useable limit.

4.5 Impact of Additional Measurements

Fixed sensors have set locations and set distance between senor locations. Moving sensors have the possibility of increasing the frequency of measurements as the sensor platform moves. If the total distance traveled by the platform remains the same while the frequency of measurements is increased the distances between measurements will decrease. Increasing the number of measurements can increase the accuracy of the estimation, but there is an upper limit. If the noise in the AOA measurements is sufficiently large, or the measurements are taken too close together, then the noise could exceed the angular difference between measurements. This will have the effect of reducing the accuracy of the system. To investigate how the linear algorithms responds to increasing measure, tests were conducted with the emitter at the center of the 2000m and 4000m arcs (similar to Figure 8). The length of the arc and the variance were kept constant while increasing the numbers of equally spaced measurements. Figures 36 and 37 display the results.

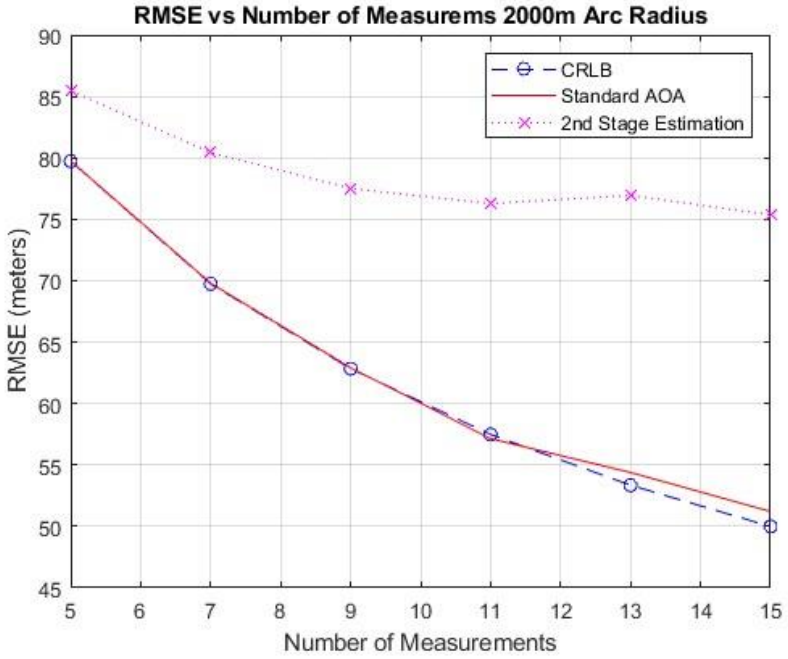


Figure 36 Error for Increasing Measurements 2000m Arc

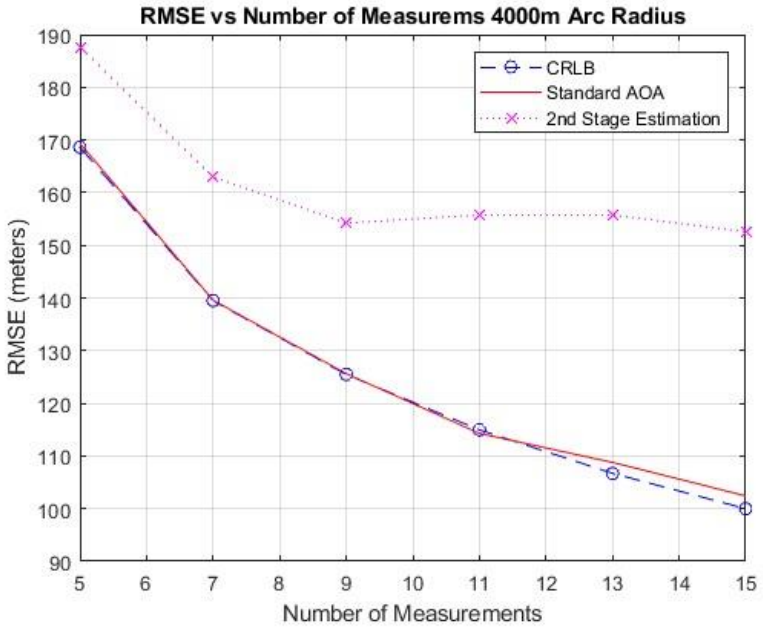


Figure 37 Error for Increasing Measurements 4000m Arc

In both cases the accuracy of the linear system improves with the number of measurements and begins to plateau around 9 measurements in the arc. This shows the advantage of using a moving sensor compared to fixed sensors. With fixed sensors, the number of measurements is limited to the number of sensors, whereas with a moving sensor, the number of measurements can be increased, which will improve the performance.

4.6 Impact of Increasing Arc Length

Another advantages that mobile sensors have over fixed sensors is the ability to change the distance between measurements or to move to locations that come closer to surrounding the emitter. To test the impact of extending the arc length past 90^o tests are conducted with 5 measurements and 9 measurements changing the arc length from 90^o to 130^o in 10^o increments. For all these tests the variance was kept constant and the arc radius was maintained at 2000m. The measurement order for these tests were: first, Last, remaining measurements from second to second last. Figure 38 and 39 show the results.

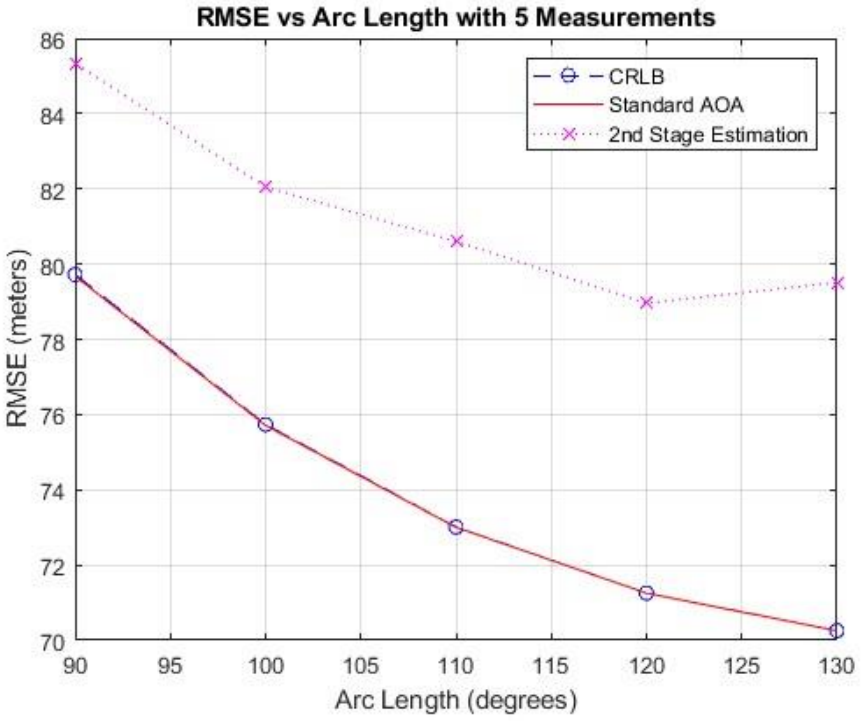


Figure 38 Error for Increasing Arc Length with 5 Measurements

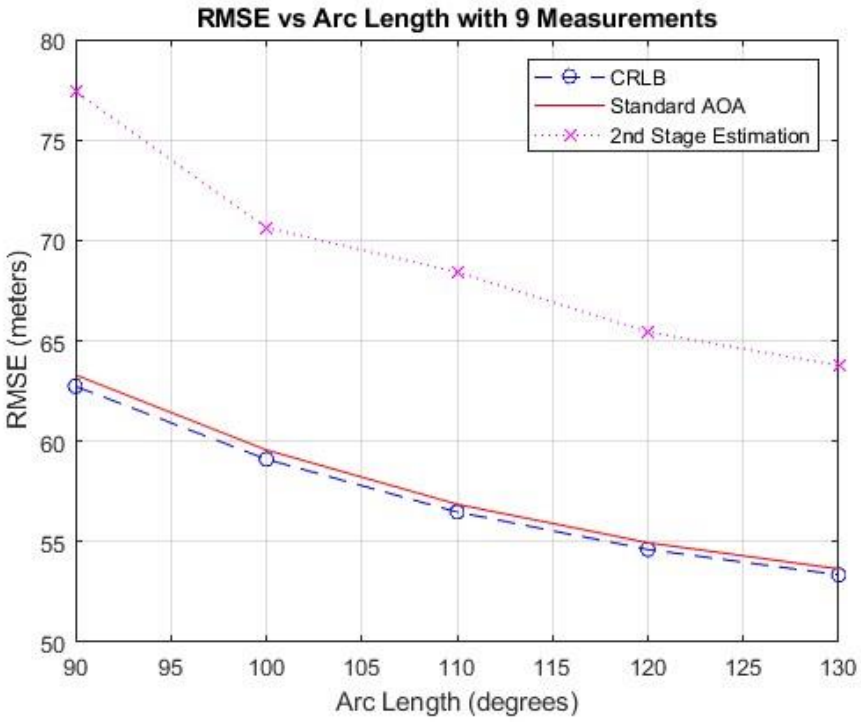


Figure 39 Error for Increasing Arc Length with 9 Measurements

As expected, increasing creasing the distance between measurements improves the accuracy of the estimation. This again shows the advantage of using a moving sensor compared to fixed sensors. With a moving sensor, we can get an estimate of the location of the emitter from the first measurements, and then try to surround the emitter, which of course cannot be done with sensors at fixed locations. In summary, increasing both the number of measurements and the total arclength improved the accuracy even further.

4.7 Discussion of Accuracy

In all the above tests, the accuracy of the linear algorithm was less than the accuracy of the standard AOA algorithm. The premise of this thesis was that using distances instead of angle of arrival measurements would reduce the bias and therefore improve the overall accuracy. The errors and noise in the system were attributed to the angular measurements. It follows that distances calculated from noisy measurements would have errors as well. To determine the cause of the overall system inaccuracy, the errors in the distance calculations are examined.

Knowing the true location of the emitter and true measurement locations, the true distance was calculated and the errors in the distances calculated by the algorithm were determined. For these tests 5000 independent iterations are conducted, then the MatLab function *mean* was used to find the mean of the distance errors and the MatLab function *std* was used to find the standard deviation of the distance errors. Table 2 provides the mean and standard deviation of the distance errors. Note that Table 2 displays the distances in the order 1 2 3 4 5 for ease of reading, however the algorithm calculates the distances in the order 1 5 2 3 4, as it previously discussed that this order improves the performance.

Table 2 Mean and Standard Deviation of Distance Errors (Order 1 5 2 3 4)

	AOA Variance				
	1	2	3	4	5
Mean of d1	0.88	1.44	1.94	2.42	2.88
Mean of d2	7.13	14.79	22.70	30.87	39.28
Mean of d3	1.82	3.62	5.43	7.26	9.09
Mean of d4	0.71	1.40	2.08	2.77	3.46
Mean of d5	0.31	0.63	0.95	1.28	1.60
Std Dev d1	35.46	50.20	61.53	71.11	79.57
Std Dev d2	125.49	180.51	224.97	264.50	301.32
Std Dev d3	60.78	86.26	106.03	122.89	137.91
Std Dev d4	40.52	57.39	70.40	81.42	91.18
Std Dev d5	35.12	49.72	61.53	70.45	78.84

Using the mean as an indication of the bias on the distance calculation, it can be seen that the bias is relatively small compared to the standard deviation. For comparison the same investigation of distance errors was conducted with the measurements in the order 1 2 3 4 5. Table 3 displays the results.

Table 3 Mean and Standard Deviation of Distance Errors (Order 1 2 3 4 5)

	AOA Variance				
	1	2	3	4	5
Mean of d1	9.59	18.60	27.72	37.04	46.59
Mean of d2	9.48	48.44	27.52	36.81	46.33
Mean of d3	1.51	3.18	4.88	6.60	8.34
Mean of d4	0.69	1.37	20.6	2.75	3.44
Mean of d5	0.28	0.57	0.86	1.15	1.45
Std Dev d1	129.51	186.66	233.05	274.50	313.31
Std Dev d2	129.40	186.50	232.86	274.28	313.07
Std Dev d3	60.37	85.73	105.43	122.24	137.23
Std Dev d4	40.66	57.58	70.62	81.67	91.44
Std Dev d5	35.08	49.65	60.85	70.32	78.67

Comparing Table 2 and Table 3, we see that the mean of the errors or bias on the calculation of d3, d4 and d5 is small when the order of the measurements is 1 2 3 4 5. The standard deviations for d2, d3, d4, and d5 are about the same in either order. The bias of d1 and d2 as well as the standard deviation of d1 are significantly higher when the measurement order is 1 2 3 4 5. The impact on the bias and standard deviation of d1 is particularly important as this value is only calculated once, but used in the calculation of all other distance. This confirms what was observed in section 4.1: using the measurements in the order 1 5 2 3 4 instead of 1 2 3 4 5 improves the performance.

The Matlab function *normplot* compares the distribution of a given variable (blue crosses in Figures 40 and 41) to the normal distribution (dashed red line in Figures 40 and 41). Figure 40 is the normplot for d1 with an AOA Variance of 1 and a sensor order of 1 5 2 3 4. Figure 41 is the normplot for d3 under the same conditions. Although d1 closely follows the normal distribution d3 does not.

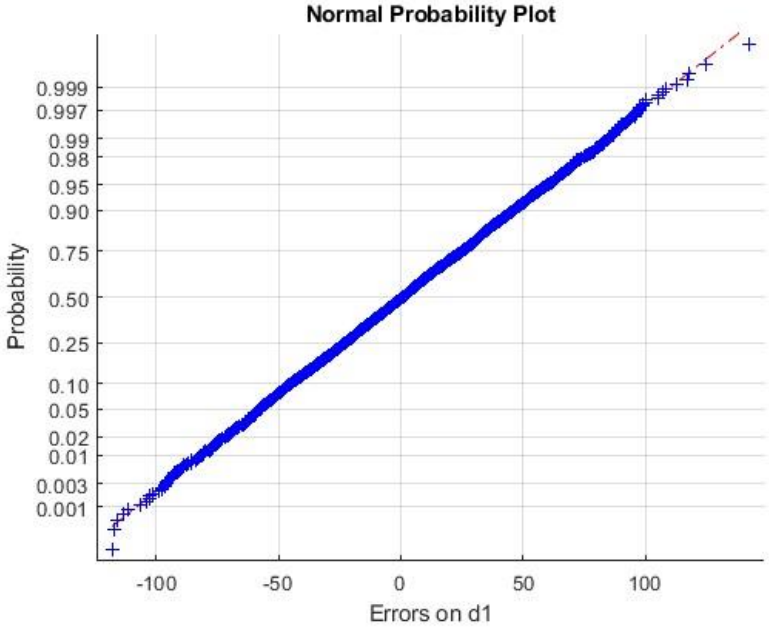


Figure 40 normplot for d1 with AOA Variance of 1

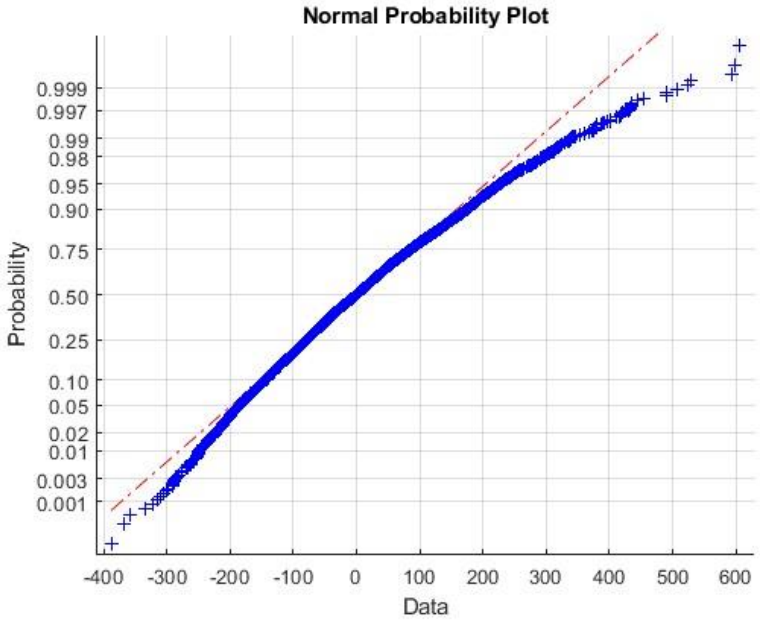


Figure 41 normplot for d3 with AOA Variance of 1

This has two important impacts on the accuracy of the linear algorithm. First the errors are not zero mean. Second the distribution is not strictly Gaussian. The noise in the system was assumed to be zero mean, additive white gaussian noise. The fact that this is not the case creates inaccuracies in the system. However, without a model of the noise, an estimate cannot be calculated.

5. Conclusion

The objective of this thesis was to utilize an algorithm based on a linear system of equations to estimate the location of an emitter from AOA measurements. The metrics to be used were the accuracy compared to the standard AOA algorithm and computational efficiency compared to the standard AOA algorithm. When using a 2-stage weighted least squares with the linear system of equations the algorithm provides an estimated emitter location within -0.5 dB of the CRLB. Although the accuracy of the new algorithm does not match the standard AOA algorithm, it is a reasonably accurate estimation. The 2-stage weighted least squares estimation does require approximately 2 microseconds more computational time than the standard AOA algorithm. It is noted that in most case the 1st stage weighted least squares estimation is fast than the standard AOA algorithm. In fact, the when the noise variance is was below 4 the 1st stage WLS estimation was in all cased faster that the standard AOA algorithm, even in unfavourable conditions. The algorithm used for this thesis is an effective estimation tool when the sensor-emitter configuration is not completely unfavorable and its accuracy is close to that of the standard AOA and to the CRLB.

Using a single moving sensor reduces the hardware complexity compared to several fixed sensors and also reduces the throughput required to transmit the data from the sensors to a central processing unit. If the moving sensor can process the collected data and estimate the location, data transmission can be completely eliminated. Furthermore, a moving sensor can increase the number of measurements and try to encircle the emitter, which will improve the localization accuracy.

5.1 Future Work

The use of distances instead of measured angles in the system matrix may provide an advantage if this algorithm were to be combined with either RSS or TDOA measurements. When the standard AOA algorithm is used in hybrid estimations with RSS and TDOA, two equations are required: the angular equation for AOA and a distance equation for RSS or TDOA. In this thesis the conversion of the angular measurements into distances is included in the time recorded for the estimation. If this algorithm were used in a hybrid estimation, the second equation would not be required and the overall time for a hybrid estimation may be reduced.

As noted in Chapter 4, increasing the frequency of measurements that include noise has an upper limit where the noise overtakes the measurements. The implementation of a filter on the noisy measurements may reduce the impact of the noise and increase the overall accuracy of the algorithm.

References

[1] Y. Chan, F. Chan, W. Read, B. Jackson and B. Lee, "Hybrid Localization of an Emitter by Combining Angle-of-Arrival and Received Signal Strength Measurements," in *IEEE 27th Canadian COnference on Electrical and Computer Engineering*, 2014.

[2] Y. Chan, H. So, B. Lee, F. Chan, B. Jackson and W. Read, "Angle-of-Arrival Localization of an Emitter from Air Platforms," in *26th IEEE Canadian Conference of Electrocal and Computer Engineering*, 2013.

[3] J. Yin, Q. Wan, S. Yang and K. C. Ho, "A Simple and Accurate TDOA-AOA Localization Method Using Tqo Stations," *IEEE Signal Processing Letters,* vol. 1, no. 23, pp. 144 - 148, January 2016.

[4] Y. Wang and K. C. Ho, "Unified Near-Field and Far-Field Localization for AOA and Hybrid AOA-TDOA Positionings," *IEEE Transaction on Wireless Communicaitions,* vol. 17, no. 2, pp. 1242 - 1254, February 2018.

[5] Y. Zhao, Z. Li, B. Hao, P. Wan and L. Wang, "How to Select the Best Sensors for TDOA and TDOA/AOA Localization," *Communications Theories & Systems,* pp. 134 - 145, February 2019.

[6] Y. Wang and K. Ho, "An Asymptotically Efficient Estimator in Closed-Form for 3-D AOA Localization Using a Sensor Network," *IEEE Transactions on Wireless Communication,* vol. 14, no. 12, pp. 6524 - 6535, December 2015.

[7] P. Delima, G. York and D. Pack, "Localization of Ground Targets Using a Flying Sensor Network," in *IEEE International Conference on Sensor Networks, Ubiquitous, and Trustworthy Computing*, Taichung Taiwan, 2006.

[8] G. J. Toussant, P. De Lima and D. J. Pack, "Localizing RF Targets with cooperative Unmanned Aerial Vehicles," in *American Control Conference*, New York, 2007.

[9] M. Gavish and A. J. Weiss, "Performance Analysis of Bearing-Only Target Location Algorithms," *IEEE Transactions on Aerospace and Electronic Systems,* vol. 28, no. 3, pp. 817 - 828, July 1992.

[10] Y. Chan, B. Lee, R. Inkol and F. Chan, "Estimation of Emitter Power, Location, and Path Loss Exponent," in *25th IEEE Canadian Conference of Electrical and Computer Engineering*, Montreal Canada, 2012.

[11] A. Kulaib, R. Shubair, M. Al-Qutayri and J. W. Ng, "Investigation of a Hybrid Localization Technique using Recieved Signal Strength and Direction of Arrival," in *IEEE 20th International Conference on Electronics, Circuits, and Systems*, Abu Dhabi, United Arab Emirates, 2014.

[12] Z. Yilong, X. Shuguo, Y. Meiling and Z. Ming, "Emitter Localization Using a Single Moving Observer Based on UKF," in *IEEE International Conference on Communication Technology*, Changdu, 2017.

[13] L. Cong and W. Zhuang, "Hybrid TDOA/AOA Mobile User Location for Wideband CDMA Celluar Systems," *IEEE Transactions on Wireless Communications,* vol. 1, no. 3, pp. 439 - 447, July 2002.

[14] R. Giacometti, A. Baussard, C. Cornu, A. Khenchaf, D. Jahan and J.-M. Quellec, "Accuracy Studies for TDOA-AOA Localization of Emitters with a Single Sensor," in *IEEE Radar Conderence*, Philidelphia, 2016.

[15] D. J. Torrieri, "Statistical Theory of Passive Location Systems," *IEEE Transactions on Aerospace and Electronic Systems,* vol. 20, no. 2, pp. 183 - 198, March 1984.

[16] Y. Zou and Q. Wan, "Emitter Source Localization Using Time-of-Arrival Measurements from a Single Moving Receiver," in *2017 IEEE International Conference on Acoustic, Speach and Sugnal Processing*, New Orleans USA, 2017.

[17] S. Zhang, Z. Huang and J. He, "A Single Sensor Passive Localization Algorithm Using Second Difference of Time Delay," in *IEEE Global Conference on SIgnal and Information Processing*, Montreal Canada, 2017.

[18] B. Jackson, S. Wang and R. Inkol, "Emitter Geolocation Estimation using Power Difference of Arrival An Algorithm Comparison for Non-Cooperative Emitters," Defence R&D Canada, Ottawa, 2011.

[19] S. Wang, B. R. Jackson and R. Inkot, "Hybrid RSS/AOA Emitter Location Estimation Based on Least Squares and Maximum Likelihood Criteria," in *2012 26th Biennial Symposium on Communicaitons*, Kingston Canada, 2012.

[20] E. Tzoreff, B. Z. Bobrovsky and A. J. Weiss, "Single Reciever Emitter Geolocation Based on Signal Periodicity With Oscillator Instability," *IEEE Transactions on Signal Processing,* vol. 62, no. 6, pp. 1377 - 1385, 15 March 2014.

[21] L. Zhang, T. Zhang, H.-S. Shin and X. Xu, "Efficirnt Underwater Acoustical Location Method Based On Time Difference and Bearing Measurements," *IEEE Transactions on Instrrumentation and Measurement,* vol. 70, December 2021.

Appendix A. MatLab Code

A1.1 Main Program

```
% Linear AOA Method based on  Dr. Y.T Chan's notes Sep 22
%
% Written by Capt. James Bayes
% Partial Fulfillment of the Requirements for the Degree of
% Master of Applied Science in Electrical Engineering
clear all

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%
% Target and measurement locations

TargetX = [2000];% X coordinate of Emitter
TargetY = [2000];% Y coordinate of Emitter

SensorX = [0.0 2000 152.2 585.8 1234.6]; % X coordinate of sensor
measurements
SensorY = [2000.0 0 1234.6 585.8 152.2]; % Y coordinate of sensor
measurements

Nsensors = length(SensorX);

NoiseVar = [0.1 1.0 2.0 3.0 4.0 5.0 6.0 7.0 8.0 9.0]; % used to
set the variance of the noise
% NoiseVar =[5 5 5];
NumVar = length(NoiseVar);

NumTrials = 5000; % used to number of iterations
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%
rng(2647589,"v4"); % Seed for random number generator

% call function to calculate the AOA for each measurement
% This represents the measurements taken at each location
% After this function call calculations will be bases on:
%     The Sensor Locations (known to Sensors) and
%     The AOA 'measured' in this function call
ThetaAngles = AOAmeasure (TargetX, TargetY, SensorX, SensorY);

% initalising Variables for Standard AOA metrics
StandCalMean = zeros(NumVar,1);
StandRSME = zeros(NumVar,1);
StandBiasX= zeros(NumVar,1);
```

```
StandBiasY= zeros(NumVar,1);
% initalising Variables for Linear with stage 1 weight metrics
Stage1CalMean = zeros(NumVar,1);
Stage1RSME = zeros(NumVar,1);
Stage1BiasX= zeros(NumVar,1);
Stage1BiasY= zeros(NumVar,1);
% initalising Variables for Linear with stage 2 weight metrics
Stage2CalMean = zeros(NumVar,1);
Stage2RSME = zeros(NumVar,1);
Stage2BiasX= zeros(NumVar,1);
Stage2BiasY= zeros(NumVar,1);
complexcount = zeros(NumVar,1);

for varit = 1:NumVar
    % Estimate and Calculation time for Standard AOA calculations
    Standx=zeros(NumTrials,1); Standy=zeros(NumTrials,1);
    StandCalcT=zeros(NumTrials,1);
    % Estimate and Calculation time for Linear with no weights
    No_weightx=zeros(NumTrials,1); No_weighty=zeros(NumTrials,1);
    No_weightCalcT=zeros(NumTrials,1);
    % Estimate and Calculation time for 1st stage weights
    Stage1x=zeros(NumTrials,1); Stage1y=zeros(NumTrials,1);
Stage1c=zeros(NumTrials,1);
    Stage1CalcT=zeros(NumTrials,1);
    % Estimate and Calculation time for 2nd stage weights
    Stage2x=zeros(NumTrials,1); Stage2y=zeros(NumTrials,1);
    Stage2CalcT=zeros(NumTrials,1);
    baddata = 0;

    for it = 1:NumTrials % Running multiple iterations of method
        ThetaAnglesafter =
ThetaAngles+sqrt(NoiseVar(varit))*randn(Nsensors,1); % adding
noise

        % Call function to calculate distance from sensor location
to Target
        % These function used the AOA and the distance between
measurements
        % to calculates the distance to target using the Law of
Sines
        tic;
        [Standx(it), Standy(it)] = AOA_2013_2D(SensorX, SensorY,
ThetaAnglesafter,NoiseVar(varit));
        StandCalcT (it,varit) = toc; % Recording the calculation
time for the Standard AOA method
        tic;
```

```
        [No_weightx(it), No_weighty(it),] = AOA_0stage (SensorX,
SensorY, ThetaAnglesafter);
        No_weightCalcT (it,varit) = toc; % Recording the
calculation time for Linear with no weights
        tic;
        [Stage1x(it), Stage1y(it), Stage1c(it)] = AOA_1stage_AB
(SensorX, SensorY, ThetaAnglesafter,NoiseVar(varit));
        Stage1CalcT (it,varit) = toc; % Recording the calculation
time for 1st stage weights
        tic
        [Stage2x(it), Stage2y(it), baddata] = AOA_2stage_AB
(SensorX, SensorY, ThetaAnglesafter,NoiseVar(varit));
        Stage2CalcT (it,varit) = toc; % Recording the calculation
time for 2nd stage weights
        complexcount(varit)=complexcount(varit)+baddata;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%
    end

    % Determine the CRLB
    [RCRLB_AOA(varit)] =
AOA_RCRLB(Nsensors,SensorX,SensorY,1,TargetX,TargetY,NoiseVar(vari
t));

    % Determining the Mean calculation time for each algorithm
    StandCalMean(varit)=mean(StandCalcT(:,varit));
    No_WCAlMean(varit)=mean(No_weightCalcT(:,varit));
    Stage1CalMean(varit)=mean(Stage1CalcT(:,varit));
    Stage2CalMean(varit)=mean(Stage2CalcT(:,varit));

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%
    % CONDUCT ANALYSIS OF THE RESULTS OF THE ESTIMATION
%%%%%%%%%%%%%%%%%%%%

    % Determine the average of the estimated target location
    StandXerror=TargetX -Standx;
    StandYerror=TargetY - Standy;
    StandDisError=sqrt(StandXerror.^2+StandYerror.^2);
    StandRSME(varit)=sqrt(mean(StandDisError.^2));

    No_WXerror=TargetX -No_weightx;
    No_WYerror=TargetY - No_weighty;
    No_WDisError=sqrt(No_WXerror.^2+No_WYerror.^2);
    No_WRSME(varit)=sqrt(mean(No_WDisError.^2));
```

```
    Stage1Xerror=TargetX -Stage1x;
    Stage1Yerror=TargetY - Stage1y;
    Stage1DisError=sqrt(Stage1Xerror.^2+Stage1Yerror.^2);
    Stage1RSME(varit)=sqrt(mean(Stage1DisError.^2));

    Stage2Xerror=TargetX -Stage2x;
    Stage2Yerror=TargetY - Stage2y;
    Stage2DisError=sqrt(Stage2Xerror.^2+Stage2Yerror.^2);
    Stage2RSME(varit)=sqrt(mean(Stage2DisError.^2));

    % Determine the bias in the estimations
    StandBiasX(varit)=(mean(Standx))-TargetX;
    StandBiasY(varit)=(mean(Standy))-TargetY;

    Stage1BiasX(varit)=(mean(Stage1x))-TargetX;
    Stage1BiasY(varit)=(mean(Stage1y))-TargetY;

    Stage2BiasX(varit)=(mean(Stage2x))-TargetX;
    Stage2BiasY(varit)=(mean(Stage2y))-TargetY;
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%
% Saving data to text file
fileid = fopen('14-04-2023.txt','a+');

fprintf(fileid, '\n\nFile written at : %s \n', datetime);
fprintf(fileid,
'***********************************************\n' );
fprintf(fileid, '*****Target at Center of Arc (2000, 2000)
*******\n' );
fprintf(fileid, 'Number of Iteration Per Noise Variance:  %6.0d
\n',NumTrials);
fprintf(fileid, 'True Target X Location is  %3.3d \n',TargetX);
fprintf(fileid, 'True Target Y Location is  %3.3d \n\n',TargetY);
fprintf(fileid, 'Measurement Location');
fprintf(fileid, '\nX coordinate'); fprintf(fileid,'
%5.1f',SensorX);
fprintf(fileid, '\nY coordinate'); fprintf(fileid,'
%5.1f',SensorY);

fprintf(fileid, '\n********* Range 2000m
*************************\n\n' );
fprintf(fileid, 'Noise Variances Used: '); fprintf(fileid,'%8.6f
',NoiseVar);
```

```
fprintf(fileid, '\n-----------------------------------------------
-----------------------------------' );
fprintf(fileid, '\nRCRLB             '); fprintf(fileid,'
%11.6f',RCRLB_AOA);
fprintf(fileid, '\n-----------------------------------------------
-----------------------------------' );

% fprintf(fileid, '\nStandard AOA Method -------------------------
---------------------------------------\n' );
fprintf(fileid, '\nRSME Standard AOA '); fprintf(fileid,'
%11.6f',StandRSME);
% fprintf(fileid, '\nBias on X coordinate is   ');
fprintf(fileid,' %11.6f',StandBiasX);
% fprintf(fileid, '\nBias on Y coordinate is   ');
fprintf(fileid,' %11.6f',StandBiasY);
% fprintf(fileid, '\nMean Time for Calculations');
fprintf(fileid,'%12.6f', StandCalMean);

% fprintf(fileid, '\nStandard Linear without Weights -------------
-----------------------------------------\n' );
fprintf(fileid, '\nRSME No Weights   '); fprintf(fileid,'
%11.6f',No_WRSME);
% fprintf(fileid, '\nBias on X coordinate is   ');
fprintf(fileid,' %11.6f',LineBiasX);
% fprintf(fileid, '\nBias on Y coordinate is   ');
fprintf(fileid,' %11.6f',LineBiasY);
% fprintf(fileid, '\nMean Time for Calculations');
fprintf(fileid,'%12.6f', LineCalMean);

% fprintf(fileid, '\nStandard Linear 1st Stage Weights -----------
-----------------------------------------\n' );
fprintf(fileid, '\nRSME S1 Weighted  '); fprintf(fileid,'
%11.6f',Stage1RSME);
% fprintf(fileid, '\nBias on X coordinate is   ');
fprintf(fileid,' %11.6f',LineBiasX);
% fprintf(fileid, '\nBias on Y coordinate is   ');
fprintf(fileid,' %11.6f',LineBiasY);
% fprintf(fileid, '\nMean Time for Calculations');
fprintf(fileid,'%12.6f', LineCalMean);

% fprintf(fileid, '\nStandard Quadratic 2nd Stage Weights --------
-----------------------------------------\n' );
fprintf(fileid, '\nRSME S2 Weighted  '); fprintf(fileid,'
%11.6f',Stage2RSME);
% fprintf(fileid, '\nBias on X coordinate is   ');
fprintf(fileid,' %11.6f',QuadBiasX);
```

```
% fprintf(fileid, '\nBias on Y coordinate is   ');
fprintf(fileid,' %11.6f',QuadBiasY);
% fprintf(fileid, '\nMean Time for Calculations');
fprintf(fileid,'%12.6f', QuadCalMean);
fprintf(fileid,
'\n************************************************\n' );
fprintf(fileid, '\nMean Time (in seconds) required for
Calculations \n' );
fprintf(fileid, 'Noise Variances used: '); fprintf(fileid,'%8.6f
',NoiseVar);
fprintf(fileid, '\n-----------------------------------------------
-------------------------------------' );
fprintf(fileid, '\nStandard AOA      '); fprintf(fileid,'%12.6f',
StandCalMean);
fprintf(fileid, '\nS1 No Weights     '); fprintf(fileid,'%12.6f',
No_WCAlMean);
fprintf(fileid, '\nS1 Weighted       '); fprintf(fileid,'%12.6f',
Stage1CalMean);
fprintf(fileid, '\nS2 Weighted       '); fprintf(fileid,'%12.6f',
Stage2CalMean);
fclose(fileid);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%
% FUNCTIONS USED IN THE SCRIPT ARE LISTED BELOW
%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%
% Function to calculate the AOA for each measurement
function [AOAAngles] = AOAmeasure(TX,TY,SX,SY)
% Calculate angles between target and measurement
Xtd = TX-SX';  Ytd = TY-SY'; % distance between Target and sensor
AOAAngles = atan2d(Xtd,Ytd);  % returns angle from positive Y-Axis
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%
```

A1.2 Standard AOA Function

```
function [x, y] = AOA_2013_2D(SX,SY,Az,VarAzi)
% Created:  Feb 2, 2023 (Groundhog Day)
% Reason:   This is the AOA algorithm from the 2013 paper AOA
localization for an emitter.
% Status:   Active
SX=SX'; SY=SY';
%   Remembered that one must use radians
VarAzi = VarAzi*(pi/180)^2;     % Convert to radians
M = size(SX,1);     % Number of sensors
a = cosd(Az);   b = sind(Az);  %

A = [-a b];
B = [-SX.*a + SY.*b];
% Get initial estimate
u1 = (A'*A)\A'*B;

% Check to see if we want Weight Matrix or not.
for repcnt = 1:2
    % YT's version of the weight matrix for the 2D part
    Xp = SX-u1(1);  Yp = SY-u1(2);
    Di = (Xp).^2 + (Yp).^2;
    W = diag([Di*VarAzi]);
%   Get updaetd estimate
    u1 = (A'/W*A)\A'/W*B;
    % update the weighting matrix with the estimate of the target
location
end
x = u1(1);  y = u1(2); %
end
```

A1.2 Linear Function without Weight Matrix

```
function [x, y,c] = AOA_0stage (SX,SY,AOAAngles)

M=length(SX);
% Creating the A Matrix
A= ones(M,3);  %  A Matrix is used for 1st stage weight [-1xi -
2yi, 1]
A(:,1)=-2*SX'; A(:,2)=-2*SY'; % placing x and y values in A

Ki=SX.^2+SY.^2;
Ki=Ki';

Xp = (SX(2:M)-SX(1))';     Yp = (SY(2:M)-SY(1))';
% Distance between sensor 1 and all other sensors
DELp = sqrt(Xp.^2 + Yp.^2);
% Angle between 1st sensore and other sensors
PHI1m = atan2d(Xp, Yp);     % angle from sensor 1 to p
PHIm1 = atan2d(-Xp,-Yp);    % angle from sensor p to 1

%   You have possible M1 triangles
BetaP = abs(AOAAngles(2:M) - PHIm1);
AlphaP = abs(PHI1m - AOAAngles(1));
BetaP(BetaP>180) = 360 - BetaP(BetaP>180);      % At sensor p,
this is the angle between sensor 1 to the target.
AlphaP(AlphaP>180) = abs(360 - AlphaP(AlphaP>180)); % Angle at
sensor 1 between target and sensor p.

%   We use the law of sines to solve Dp1
d1_p = DELp.*sind(BetaP)./(sind(AlphaP+BetaP));
% Calculate values for AOA
dp = DELp./sind(AlphaP+BetaP).*sind(AlphaP);

B= zeros(M,1);
B(1) = d1_p(1)^2-Ki(1);
B(2:M) = dp.^2-Ki(2:M);

uxy = (A'*A)\A'*B;

x = uxy(1);  y = uxy(2);
end
```

A1.3 Linear Function with 1st Stage Weight Matrix

```
function [x, y, c] = AOA_1stage (SX,SY,AOAAngles,VarAzi)
%   Remembered that one must use radians
VarAzi = VarAzi*(pi/180)^2;
M = size(SX,2); % Number of sensors
M1 = (M-1);

% Creating the A Matrix
A= ones(M,3);  %  A Matrix is used for 1st stage weight [-1xi -
2yi, 1]
A(:,1)=-2*SX'; A(:,2)=-2*SY'; % placing x and y values in A

Ki=SX.^2+SY.^2;
Ki=Ki';

Xp = (SX(2:M)-SX(1))';     Yp = (SY(2:M)-SY(1))';
% Distance between sensor 1 and all other sensors
DELm = sqrt(Xp.^2 + Yp.^2);
% Angle between 1st sensore and other sensors
PHI1m = atan2d(Xp, Yp);     % angle from sensor 1 to p
PHIm1 = atan2d(-Xp,-Yp);    % angle from sensor p to 1

%   You have possible M1 triangles
BetaM = zeros(M1,1);   AlphaM = zeros(M1,1);
% These variables contain the sign of the error
BetaMs = ones(M1,1);   AlphaMs = ones(M1,1);
k = 0;
i = 1;
for j = i+1:M
    k = k  + 1;
    %   The code below tracks the sign of the errors
    BetaM(k) = (AOAAngles(j) - PHIm1(k));
    if BetaM(k) < 0
        BetaM(k) = abs(BetaM(k));   BetaMs(k) = -BetaMs(k);     %
angle is negative, make pos, flip sign of error
    end
    if BetaM(k) > 180
        BetaM(k) = 360 - BetaM(k);  BetaMs(k) = -BetaMs(k);     %
angle greater than 180, sub tract from 360, flip sign of error
    end
    AlphaM(k) = (PHI1m(k) - AOAAngles(i));
    if AlphaM(k) < 0
        AlphaM(k) = abs(AlphaM(k));   AlphaMs(k) = -AlphaMs(k);
    end
    if AlphaM(k) > 180
        AlphaM(k) = 360 - AlphaM(k);  AlphaMs(k) = -AlphaMs(k);
```

```
    end
end

%   We use the law of sines to solve Dp1
dk_m = DELm.*sind(BetaM)./(sind(AlphaM+BetaM)); % d(1)
% Calculate values for AOA
d_m = DELm.*sind(AlphaM)./sind(AlphaM+BetaM);   % d(2:M)

distm= zeros(M,1); % Creating a single matrix for d1 through dm
distm(1,1) = dk_m(1); distm(2:M) = d_m;
% Creating the B Martix [(di)^2-ki]
B= zeros(M,1);
B(1) = dk_m(1)^2-Ki(1);
B(2:M) = d_m.^2-Ki(2:M);

% Build the weighting matrix
% Init vectors
F1=zeros(M,1); F2 = F1; G1 = F1; G2 = F1;
k = 0;
i=1;
for j = i+1:M
    k = k + 1;
    % F1 and F2 are for the d(1)
    F1(k) = -DELm(k).*sind(BetaM(k)).*cosd(AlphaM(k)+BetaM(k))...
        ./(sind(AlphaM(k)+BetaM(k)).^2);
    F2(k) = DELm(k).*cosd(BetaM(k))./(sind(AlphaM(k)+BetaM(k)))
...
        -
DELm(k).*sind(BetaM(k)).*cosd(AlphaM(k)+BetaM(k))./(sind(AlphaM(k)
+BetaM(k)).^2);

    % G1 and G2 are for the d(i)
    % NOTE the G1,G2 elements are shifted one place to align with
the
    % Epsilon Matrix below
    G1(k+1) = DELm(k).*cosd(AlphaM(k))./(sind(AlphaM(k)+BetaM(k)))
...
        -
DELm(k).*sind(AlphaM(k)).*cosd(AlphaM(k)+BetaM(k))./(sind(AlphaM(k
)+BetaM(k)).^2);
    G2(k+1) = -
DELm(k).*sind(AlphaM(k)).*cosd(AlphaM(k)+BetaM(k))./(sind(AlphaM(k
)+BetaM(k)).^2);
end
% Building the Epsilon Matrix
Epsilon = zeros(M,M);
```

```
Epsilondiag = zeros(M,M);
Epsilonoff = zeros (M,M);
for Rowi =1:M
    for Coli = 1:M
        if (Rowi == 1) && (Coli ==1)
            Epsilondiag(Rowi,Coli) =
distm(Rowi)^2*(AlphaMs(1)*AlphaMs(1)*F1(1)^2+BetaMs(1)*BetaMs(1)*F
2(1)^2)*VarAzi;
        elseif (Rowi == 1) && (Coli ==2)
             Epsilonoff(Rowi,Coli) =
distm(Rowi)*distm(Coli)*(AlphaMs(1)*AlphaMs(1)*F1(1)*G1(2)+BetaMs(
1)*BetaMs(1)*F2(1)*G2(2))*VarAzi;
        elseif (Rowi == 1) && (Coli >=3)
            Epsilonoff(Rowi,Coli) =
distm(Rowi)*distm(Coli)*AlphaMs(1)*AlphaMs(Coli-
1)*F1(1)*G1(Coli)*VarAzi;
        elseif (Rowi ~=1) && (Coli == Rowi)
            Epsilondiag(Rowi,Coli) = distm(Rowi)^2*(AlphaMs(Coli-
1)*AlphaMs(Coli-1)*G1(Coli)^2+BetaMs(Coli-1)*BetaMs(Coli-
1)*G2(Coli)^2)*VarAzi;
        elseif (Rowi >=2) && (Coli > Rowi)
            Epsilonoff(Rowi,Coli) =
distm(Rowi)*distm(Coli)*AlphaMs(Rowi-1)*G1(Rowi)*AlphaMs(Coli-
1)*G1(Coli)*VarAzi;
        else
            Epsilonoff(Rowi,Coli) = 0;
        end
    end
end
Epsilon = Epsilondiag+Epsilonoff+Epsilonoff';
% resolve the 1st stage solution
uW = (A'/Epsilon*A)\A'/Epsilon*B;
x=uW(1); y=uW(2); c=uW(3);
end
```

A1.4 Linear Function with 2nd Stage Weight Matrix

```
function [x, y, negc] = AOA_2stage (SX,SY,AOAAngles,VarAzi)
%   Remembered that one must use radians
VarAzi = VarAzi*(pi/180)^2;
M = size(SX,2); % Number of sensors
M1 = (M-1);

% Creating the A Matrix
A= ones(M,3);  %  A Matrix is used for 1st stage weight [-1xi -
2yi, 1]
A(:,1)=-2*SX'; A(:,2)=-2*SY'; % placing x and y values in A

Ki=SX.^2+SY.^2;
Ki=Ki';

Xp = (SX(2:M)-SX(1))';     Yp = (SY(2:M)-SY(1))';
% Distance between sensor 1 and all other sensors
DELm = sqrt(Xp.^2 + Yp.^2);
% Angle between 1st sensore and other sensors
PHI1m = atan2d(Xp, Yp);     % angle from sensor 1 to p
PHIm1 = atan2d(-Xp,-Yp);    % angle from sensor p to 1

%   You have possible M1 triangles
BetaM = zeros(M1,1);   AlphaM = zeros(M1,1);
% These variables contain the sign of the error
BetaMs = ones(M1,1);   AlphaMs = ones(M1,1);
k = 0;
i = 1;
for j = i+1:M
    k = k  + 1;
    %   The code below tracks the sign of the errors
    BetaM(k) = (AOAAngles(j) - PHIm1(k));
    if BetaM(k) < 0
        BetaM(k) = abs(BetaM(k));   BetaMs(k) = -BetaMs(k);     %
angle is negative, make pos, flip sign of error
    end
    if BetaM(k) > 180
        BetaM(k) = 360 - BetaM(k);  BetaMs(k) = -BetaMs(k);     %
angle greater than 180, sub tract from 360, flip sign of error
    end
    AlphaM(k) = (PHI1m(k) - AOAAngles(i));
    if AlphaM(k) < 0
        AlphaM(k) = abs(AlphaM(k));   AlphaMs(k) = -AlphaMs(k);
    end
    if AlphaM(k) > 180
        AlphaM(k) = 360 - AlphaM(k);  AlphaMs(k) = -AlphaMs(k);
```

```
    end
end

%   We use the law of sines to solve Dp1
dk_m = DELm.*sind(BetaM)./(sind(AlphaM+BetaM)); % d(1)
% Calculate values for AOA
d_m = DELm.*sind(AlphaM)./sind(AlphaM+BetaM);   % d(2:M)

distm= zeros(M,1); % Creating a single matrix for d1 through dm
distm(1,1) = dk_m(1);
distm(2:M) = d_m;

% Creating the B Martix [(di)^2-ki]
B= zeros(M,1);
B(1) = dk_m(1)^2-Ki(1);
B(2:M) = d_m.^2-Ki(2:M);

% Build the weighting matrix
% Init vectors
F1=zeros(M,1); F2 = F1; G1 = F1; G2 = F1;
k = 0;
i=1;
for j = i+1:M
    k = k + 1;
    % F1 and F2 are for the d(1)
    F1(k) = -DELm(k).*sind(BetaM(k)).*cosd(AlphaM(k)+BetaM(k))...
        ./(sind(AlphaM(k)+BetaM(k)).^2);
    F2(k) = DELm(k).*cosd(BetaM(k))./(sind(AlphaM(k)+BetaM(k)))
...
        -
DELm(k).*sind(BetaM(k)).*cosd(AlphaM(k)+BetaM(k))./(sind(AlphaM(k)
+BetaM(k)).^2);

    % G1 and G2 are for the d(i)
    % NOTE the G1,G2 elements are shifted one place to align with
the
    % Epsilon Matrix below
    G1(k+1) = DELm(k).*cosd(AlphaM(k))./(sind(AlphaM(k)+BetaM(k)))
...
        -
DELm(k).*sind(AlphaM(k)).*cosd(AlphaM(k)+BetaM(k))./(sind(AlphaM(k
)+BetaM(k)).^2);
    G2(k+1) = -
DELm(k).*sind(AlphaM(k)).*cosd(AlphaM(k)+BetaM(k))./(sind(AlphaM(k
)+BetaM(k)).^2);
end
```

```
% Building the Epsilon Matrix
Epsilon = zeros(M,M);
Epsilondiag = zeros(M,M);
Epsilonoff = zeros (M,M);
for Rowi =1:M
    for Coli = 1:M
        if (Rowi == 1) && (Coli ==1)
            Epsilondiag(Rowi,Coli) =
distm(Rowi)^2*(AlphaMs(1)*AlphaMs(1)*F1(1)^2+BetaMs(1)*BetaMs(1)*F
2(1)^2)*VarAzi;
        elseif (Rowi == 1) && (Coli ==2)
             Epsilonoff(Rowi,Coli) =
distm(Rowi)*distm(Coli)*(AlphaMs(1)*AlphaMs(1)*F1(1)*G1(2)+BetaMs(
1)*BetaMs(1)*F2(1)*G2(2))*VarAzi;
        elseif (Rowi == 1) && (Coli >=3)
            Epsilonoff(Rowi,Coli) =
distm(Rowi)*distm(Coli)*AlphaMs(1)*AlphaMs(Coli-
1)*F1(1)*G1(Coli)*VarAzi;
        elseif (Rowi ~=1) && (Coli == Rowi)
            Epsilondiag(Rowi,Coli) = distm(Rowi)^2*(AlphaMs(Coli-
1)*AlphaMs(Coli-1)*G1(Coli)^2+BetaMs(Coli-1)*BetaMs(Coli-
1)*G2(Coli)^2)*VarAzi;
        elseif (Rowi >=2) && (Coli > Rowi)
            Epsilonoff(Rowi,Coli) =
distm(Rowi)*distm(Coli)*AlphaMs(Rowi-1)*G1(Rowi)*AlphaMs(Coli-
1)*G1(Coli)*VarAzi;
        else
            Epsilonoff(Rowi,Coli) = 0;
        end
    end
end
Epsilon = Epsilondiag+Epsilonoff+Epsilonoff';
% resolve the 1st stage solution
uW = (A'/Epsilon*A)\A'/Epsilon*B;
PhiS1= (A'/Epsilon*A); % storing for using in 2nd stage

x=uW(1); y=uW(2); c=uW(3);

% Record signs for stage 2
xsign =1; ysign =1;
if x<0
    xsign =-1;
end
if y<0
    ysign =-1;
```

```
end
negc=0;
% If c is negative the estimate will be complex
% If c is negative use stage 1 values only
if (x^2 >=0) && (y^2 >=0) && (c >=0)
    %Stage 2
    Dmat = zeros(3,3);
    Dmat (1,1) = 2*x; Dmat (2,2) = 2*y; Dmat(3,3)=1;
    PhiS2 = Dmat/PhiS1*Dmat;

    H = [1,0;0,1;1,1];
    Qmat = [x^2;y^2;c];

    uW2 = (H'/PhiS2*H)\H'/PhiS2*Qmat;
    if (uW2(1)>=0) && (uW2(2) >=0)
        % resolve solution for 2nd stage
        x = sqrt(uW2(1))*xsign; y=sqrt(uW2(2))*ysign;
    else
        negc=1;
    end
else
    negc=1;
end
end
```

Appendix B. Derivation of Cramer Rao Lower Bound

This derivation is adapted from [9] but altered to match the notation used in this thesis.

For AOA geolocation in two dimensions, we let the true emitter location be:

$$E = \begin{bmatrix} x \\ y \end{bmatrix} \tag{B1}$$

The location of M sensors is (x_1, y_1), (x_2, y_2), …, (x_M, y_M) and the AOA measurements between the sensors and the emitter are:

$$\theta = \begin{bmatrix} \theta_1 \\ \theta_2 \\ \vdots \\ \theta_M \end{bmatrix} \tag{B2}$$

Each of these measurements contain the true AOA:

$$g(E) = \begin{bmatrix} \tan^{-1}\left(\frac{x - x_1}{y - y_1}\right) \\ \tan^{-1}\left(\frac{x - x_2}{y - y_2}\right) \\ \vdots \\ \tan^{-1}\left(\frac{x - x_M}{y - y_M}\right) \end{bmatrix} = \begin{bmatrix} \tan^{-1}\left(\frac{\Delta x_1}{\Delta y_1}\right) \\ \tan^{-1}\left(\frac{\Delta x_2}{\Delta y_2}\right) \\ \vdots \\ \tan^{-1}\left(\frac{\Delta x_M}{\Delta y_M}\right) \end{bmatrix} = \begin{bmatrix} g_1(E) \\ g_2(E) \\ \vdots \\ g_M(E) \end{bmatrix} \tag{B3}$$

and zero mean additive white gaussian noise:

$$\delta\theta = \begin{bmatrix} \delta\theta_1 \\ \delta\theta_2 \\ \vdots \\ \delta\theta_M \end{bmatrix} \tag{B4}$$

Therefore equation (B2) can be written as:

$$\theta = \begin{bmatrix} g_1(T) + \delta\theta_1 \\ g_2(T) + \delta\theta_2 \\ \vdots \\ g_M(T) + \delta\theta_M \end{bmatrix} \tag{B5}$$

Noting that the distance between any sensor and the emitter is:

$$d_i = \sqrt{[x - x_i]^2 + (y - y_1)^2} \tag{B6}$$

The derivative of $g(E)$ evaluated to true emitter position is:

$$g_E = \frac{\partial g(T)}{\partial T} = \begin{bmatrix} \frac{\Delta x_1}{r_1^2} & \frac{-\Delta y_1}{r_1^2} \\ \frac{\Delta x_2}{r_2^2} & \frac{-\Delta y_2}{r_2^2} \\ \vdots & \vdots \\ \frac{\Delta x_M}{r_M^2} & \frac{-\Delta y_M}{r_M^2} \end{bmatrix} \tag{B7}$$

Defining a diagonal matrix of the noise variance as:

$$S = \begin{bmatrix} \sigma_1^2 & 0 & 0 & 0 \\ 0 & \sigma_2^2 & 0 & 0 \\ 0 & 0 & \ddots & 0 \\ 0 & 0 & 0 & \sigma_M^2 \end{bmatrix} \tag{B8}$$

The Cramer Rao Lower Bound can be calculated as:

$$C = (g_E^T S^{-1} g_E)^{-1} \tag{B9}$$

and evaluated as:

$$C = \frac{1}{M} \sum_{i=1}^{M} \frac{1}{\sigma_i^2} \begin{bmatrix} \frac{(\Delta y_i)^2}{r_i^4} & \frac{-\Delta x_i \Delta y_i}{r_i^4} \\ \frac{-\Delta x_i \Delta y_i}{r_i^4} & \frac{(\Delta x_i)^2}{r_i^4} \end{bmatrix} \tag{B10}$$