M. ZEGLAM

COMPARATIVE ASSESSMENT OF SIMPLE AND ACCURATE LOCALIZATION METHODS BASED ON TDOA-AOA MEASUREMENTS

THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
ATILIM UNIVERSITY

MOHAMED ZEGLAM

A MASTER OF SCIENCE THESIS
IN
THE DEPARTMENT OF ELECTRICAL AND ELECTRONICS ENGINEERING

OCTOBER 2021

ATILIM UNIVERSITY 2021

COMPARATIVE ASSESSMENT OF SIMPLE AND ACCURATE LOCALIZATION METHODS BASED ON TDOA-AOA MEASUREMENTS

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
ATILIM UNIVERSITY

BY

MOHAMED ZEGLAM

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF MASTER OF SCIENCE
IN
THE DEPARTMENT OF ELECTRICAL AND ELECTRONICS ENGINEERING

OCTOBER 2021

Approval of the Graduate School of Natural and Applied Sciences, Atilim University.

_____

Prof. Dr. Ender KESKİNKILIÇ
Director

I certify that this thesis satisfies all the requirements as a thesis for the degree of **Master of Science in Electrical and Electronics Engineering Department, Atılım University**.

_____

Assoc. Prof. Kemal Efe ESELLER
Head of Department

This is to certify that we have read the thesis COMPARATIVE ASSESSMENT OF SIMPLE AND ACCURATE LOCALIZATION METHODS BASED ON TDOA-AOA MEASUREMENTS submitted by MOHAMED ZEGLAM and that in our opinion it is fully adequate, in scope and quality, as a thesis for the degree of Master of Science.

_____

Asst. Prof. Dr. Yaser DALVEREN
Supervisor

Examining Committee Members:

Prof. Dr. Ali Kara
Software Eng. Department, Gazi University _____

Asst. Prof. Dr. Yaser Dalveren
Electrical and Electronics Dept., Atılım University _____

Asst. Prof. Dr. Mehmet Efe Özbek
Electrical and Electronics Dept., Atılım University _____

Date: 05.10.2021

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Last Name : Mohamed Zeglam

Signature :

ABSTRACT

COMPARATIVE ASSESSMENT OF SIMPLE AND ACCURATE LOCALIZATION METHODS BASED ON TDOA-AOA MEASUREMENTS

Zeglam, Mohamed
MSc., Department of Electrical and Electronics Engineering
Supervisor: Asst. Prof. Dr. Yaser Dalveren

October 2021, 51 pages

This thesis is aimed to determine a simple but effective localization method for estimating the position of a stationary emitter by utilizing both Time Difference of Arrival (TDOA) and Angle of Arrival (AOA) measurements to be used an ongoing project. For this purpose, firstly, literature survey is conducted for the methods used to estimate the position of a stationary emitter through the TDOA measurements. Then, hybrid methods that utilize both TDOA and AOA measurements together for emitter localization are reviewed. Among these methods, the simplest but accurate methods are determined, namely Least Squared (LS) method and Weighted Least Squared (WLS) method. Next, simulations are conducted to examine the accuracy of these methods under various simulation scenarios considering the different number of sensors and TDOA measurement errors. With the help of simulation results, the performance of the methods are comparatively assessed in terms of estimation accuracy. From the results, it can be concluded that WLS method provides better accuracy under high measurement error and high number of sensors.

Keywords: Localization, Time-difference-of-arrival, Angle-of-arrival, Electronic warfare.

ÖZ

TDOA-AOA ÖLÇÜMLERİNE DAYALI BASİT VE ETKİN YER TESPİT YÖNTEMLERİNİN KARŞILAŞTIRMALI DEĞERLENDİRİLMESİ

Zeglam, Mohamed
Yüksek Lisans, Elektrik ve Elektronik Mühendisliği Bölümü
Tez Yöneticisi : Dr. Öğr. Üy. Yaser Dalveren

Ekim 2021, 51 sayfa

Bu tez, devam eden bir projede kullanılmak üzere hem Varış Zaman Farkı (VZF) hem de Varış Açısı (VA) ölçümlerini kullanarak sabit bir emitörün (emisyon kaynağının) konumunu tahmin etmek için basit ama etkili bir konumlama yöntemi belirlemeyi amaçlamaktadır. Bu amaçla, ilk olarak, VZF ölçümleri yoluyla sabit bir emitörün konumunu tahmin etmek adına kullanılan yöntemler için literatür taraması yapılmıştır. Ardından, emitör konumlaması için hem VZF hem de VA ölçümlerini birlikte kullanan hibrit yöntemler gözden geçirilmiştir. Bu yöntemler arasından bilinen adlarıyla En Küçük Kareler (EKK) yöntemi ve Ağırlıklı En Küçük Kareler (AEKK) yöntemi konumlamada kullanılabilecek basit fakat etkin yöntemler olarak belirlenmiştir. Daha sonra, farklı sensör sayıları ve VZF ölçüm hataları dikkate alınarak çeşitli senaryolar altında bu yöntemlerin doğruluğunu incelemek için benzetimler yapılmıştır. Benzetim sonuçları yardımıyla, tahmin doğruluğu açısından yöntemlerin performansı karşılaştırmalı olarak anazli edilmiştir. Sonuçlar ele alındığında, WLS yönteminin yüksek ölçüm hatası ve çok sayıda sensör altında daha iyi kestirim doğruluğu sağladığı sonucuna varılmıştır.

Anahtar Kelimeler: Konumlama, Varış zaman farkı, Varış açısı, Elektronik harp.

To my mother

ACKNOWLEDGMENTS

I would start by expressing my sincere gratefulness to my supervisor Asst. Prof. Dr. Yaser DALVEREN, it is just very simple to say this work would not have seen the light without his support, guidance, and extreme help.

To my mother, the greatest support in everything. May Allah give me the ability to make you proud my whole life.

I would also like to thank my professors at Atilim University, I can confidently say that you have been the light in my educational journey, and I want to specify Prof. Dr. Ali KARA, my former supervisor, with whom I learned a lot in a long period of time. I should not forget to thank the Atilim University administration for the way I have been treated as a student.

Thanks to my colleagues and the friends I made in this journey and the ones before it.

Finally, I am grateful to my family, for their continued support and for being there whenever I needed them.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ACRONYMS

TOA	Time of Arrival
TDOA	Time Difference of Arrival
RSSI	Received Signal Strength Indicator
RSS	Received Signal Strength
CDF	Cumulative Distributing Function
CRLB	Cramer Rao Lower Bound
GPS	Global Positioning System
LOS	Line of Sight
NLOS	Non-Line of Sight
MSE	Mean Square Error
RMSE	Root Mean Square Error
UWB	Ultra-Wide Band
RD	Range Difference

CHAPTER 1

INTRODUCTION

In recent years, the need to determine the position of an emitter has become extremely important for many applications that rely on technological advances, such as wireless sensor networks, radar, sonar, communications, and so on. Many positioning techniques are used, such as time of arrival, time difference of arrival, angle of arrival, received signal strength, etc. [1].

This study thoroughly covers localization techniques that are widely used, more specifically, the self-localization techniques, and focuses on the range-based techniques. Several localization algorithms have been presented. The extraction of position information from a network has been extensively researched in the literature. It is often believed in this methodology that there are a few sensor nodes with known positions, referred to as reference nodes and that various types of measurements are performed between them. The localization techniques are divided into range-based techniques and range-free techniques, time of arrival (TOA), time difference of arrival (TDOA), received signal strength indicator (RSSI), and angle of arrival (AOA) methods are all included in the range-based techniques [2].

Several papers were surveyed in this study, including [1], in which a well-put comparison between TOA and TDOA is made. On the one hand, the TOA method compares the time of arrival at each node, giving that we already know the transmitting time at each node and simply evaluate the distance from that specific node by knowing the time difference between sending the signal and receiving it. TDOA, on the other hand, does not require transmitting time from the unknown node because it only considers the time difference at the receiving end. However, the paper only mentioned the two previously mentioned

methods, without mentioning any of the categories in which they fall, or the statistical algorithms used in both approaches.

Because of its low cost, lack of additional hardware support, and ease of understanding, the RSSI-based trilateral localization algorithm has become the standard localization algorithm in wireless sensor networks. However, because indoor localization requires higher accuracy. With the advancement of wireless sensor networks and smart devices, the number of WIFI access points in these buildings is growing. If a mobile smart device can identify the positions of three or three more known WIFI hotspots, self-localization would be reasonably simple to achieve (Usually WIFI access points locations are fixed) [3].

These singular approaches have their application-specific advantages. For example, the time of arrival technique has the advantage of accurate range determination. However, it requires a specific emitting time from an unknown source, which may not be available in most applications. The time difference of arrival does not require that, but to have an accurate location of the unknown signal, it needs at least 4 properly located sensors to detect a signal from an unknown location. The angle of arrival technique has the advantage of needing a small number of sensors to determine the location, nonetheless, the sensors must be highly equipped to achieve accurate positioning. The main issue is that the RSSI value is highly susceptible to environmental influences, resulting in substantial calculation errors in RSSI-based localization algorithms. offers an enhanced RSSI-based method; experimental findings suggest that the technique enhances localization accuracy and decreases deviation when compared to original RSSI-based localization algorithms [3].

In [4], an overall look at the different positioning methods, the Neuro-Fuzzy Method, which is a range-free technique, was presented thoroughly, in which the focus shifted from our main study.

1.1. Range-based Self-localization Algorithms

Range-based and range-free localization algorithms are the two main types of wireless localization techniques. Range-based approaches assume that sensor nodes can determine the distances between them and their neighbors, as well as their relative directions, using Time of Arrival (TOA), Time Difference of Arrival (TDOA), Radio Signal Strength

Indicator (RSSI), or Angle of Arrival (AOA), among other methods. Range-based schemes have higher location accuracy than range-free methods, but they require additional gear to obtain distances or angles, and they are susceptible to noise [5].

AOA requires antenna arrays or microphone arrays, while TOA requires correct clock synchronization. TDOA nodes are fitted with ultrasonic transmitters and receivers. These three techniques have excellent localization precision; however, they have significant hardware requirements [6].

The most prevalent range-free localization algorithms are the centroid algorithm, DV-hop algorithm, MDS-MAP algorithm, and convex programming. Range-free approaches rely on the geometric relationship between neighboring nodes to determine localization. They have low-cost hardware requirements, but their localization precision is insufficient for an indoor context [7].

It has long been known that using TDOA techniques in a multiple sensor system can result in highly accurate localization of radar emitters. In TDOA localization, TDOA measurements between two sensors are traditionally translated to Range Differences (RDs) by multiplying them with the speed of light. The sensors at the foci are defined by a hyperbola defined by the constant RD's. The sensors' generated hyperbolas (in two dimensions) with constant RD are quadratic curves known as lines of positions (LOPs). The estimated position of the emitter is determined by the intersection of the set of hyperbolas [8].

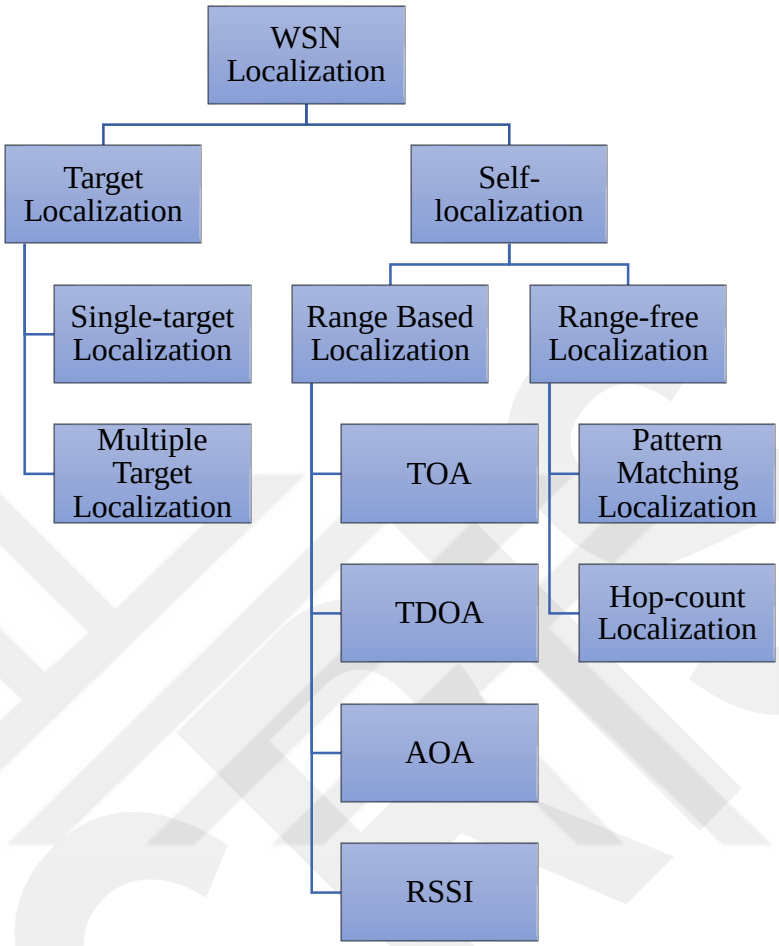

Figure1.1: WSN localization methods

1.2. Measurement Models

The type and quality of measurements collected in WSNs has a significant impact on how well a positioning algorithm performs. Various types of measurements have been investigated for the positioning problem, including received signal strength (RSS), angle-of-arrival (AOA), time-of-arrival (TOA), and time-difference-of-arrival (TDOA) [9]. Because the model of measurements has such a large impact on the design of an estimator for the positioning problem, it is vital to select the right model. We will presume that a sensor node delivers signals, receives them or both. The measurement between two nodes at S1 and S2 generally follows the paradigm provided in [2]. Sensor nodes can be either stationary or movable. They could be able to do a variety of measurements as well. In this

chapter, we will look at three different measurement methodologies based on RSSI, AOA, and TOA [10].

1.2.1. Received signal strength Indicator-RSSI

Because of its simplicity, ease of understanding, and low cost, the RSSI-based trilateral localization method offers a wide range of applications.

The RSS in WSNs is commonly modeled using the following model for received power in wireless channels. The equation below can be used to express the average received power from transmitter I at receiver j in decibels [11].

$$P_{ij} = P_{0i} - 10\beta \log\left(\frac{d(x_i, x_j)}{d_0}\right) + n_{ij} \tag{1.1}$$

where P_{0i} is the power at distance d_0, $d(x_i, x_j)$is the second norm or the range distance between, x_i and, x_j, β is a path-loss exponent that is commonly between 2 and 6, and, n_{ij} is modeled as a zero-mean Gaussian random variable with variance σ_{ij}^2.

So, basically, n_{ij} is the normal distribution with $N(0, \sigma_{ij}^2)$, and it is not dependent on the location of the transmitting node.

$$f(x_i, x_j) = P_{0i} - 10\beta \log\left(\frac{d\|x_i, x_j\|}{d_0}\right) \tag{1.2}$$

P_{0i} and β, are the maximum likelihood estimate of the distance between node i and j can be obtained as

$$\hat{d}_{ij} = d_0 10^{\frac{P_{0i} - P_{ij}}{10\beta}} \tag{1.3}$$

It may be demonstrated that the distance estimation in (1.3) is biased. The following formula can be used to calculate an unbiased distance estimate.

$$\hat{d}_{ij}^{u} = d_0 10^{\frac{P_{0i} - P_{ij}}{10\beta}} e^{-\frac{10\beta}{\sigma_{ij} \ln 10}} \tag{1.4}$$

The Cramer-Rao lower-bound for the variance of any unbiased distance estimator based on RSS measurements can be obtained as

$$E\left(\hat{d}_{ij} - E(\hat{d}_{ij})\right)^2 \geq \left(\frac{\sigma_{ij} d(x_i, x_j) \ln 10}{10\beta}\right)^2 \tag{1.5}$$

The distance estimation accuracy degrades as the distance between the two nodes increases, as does the standard deviation σ_{ij} of measurement noise (1.1). It also illustrates that the bigger the path-loss exponent β, the more accurate the distance estimate, because the average power is more sensitive to distance when the path loss is larger.

When calculating distance using RSS, the path-loss exponent or reference power P_{0i} must be treated as nuisance parameters if they are unknown [12].

1.2.2. Angle of Arrival-AOA

We can measure the angle-of-arrival for a signal received from a sensor i using the 2-D angle of arrival approach, which uses an array of antennas in a sensor node. In radians, the measured angle is written as:

$$\hat{\theta}_{ij} = \tan^{-1}\left(\frac{x_{j2} - x_{i2}}{x_{j1} - x_{i1}}\right) + n_{ij} \tag{1.6}$$

where $\tan^{-1}$ denotes a four-quadrant inverse tangent and n_{ij} is often modeled by a zero-mean Gaussian random variable. Then,

$$f(x_i, x_j) = \tan^{-1}\left(\frac{x_{j2} - x_{i2}}{x_{j1} - x_{i1}}\right) \tag{1.7}$$

Consider a uniform linear array with Na elements and r elements between them. The Cramer-Rao lower bound on the variance of any unbiased estimator of AOA, also known as direction-of-arrival or bearing, is straightforward: Assuming the same fading coefficient for all signals arriving at the array elements, the Cramer-Rao lower bound on the variance of any unbiased estimator of AOA, also known as direction-of-arrival or bearing, is given by

$$E\left(\hat{\theta}_{ij} - E(\hat{\theta}_{ij})\right)^2 \geq \frac{\sqrt{3}c}{\sqrt{2}\pi\sqrt{SNR}B_e r\sqrt{N_a(N_a^2-1)}\sin(\theta_{ij})} \tag{1.8}$$

Now, consider the 3-D scenario shown in Figure 1.2.

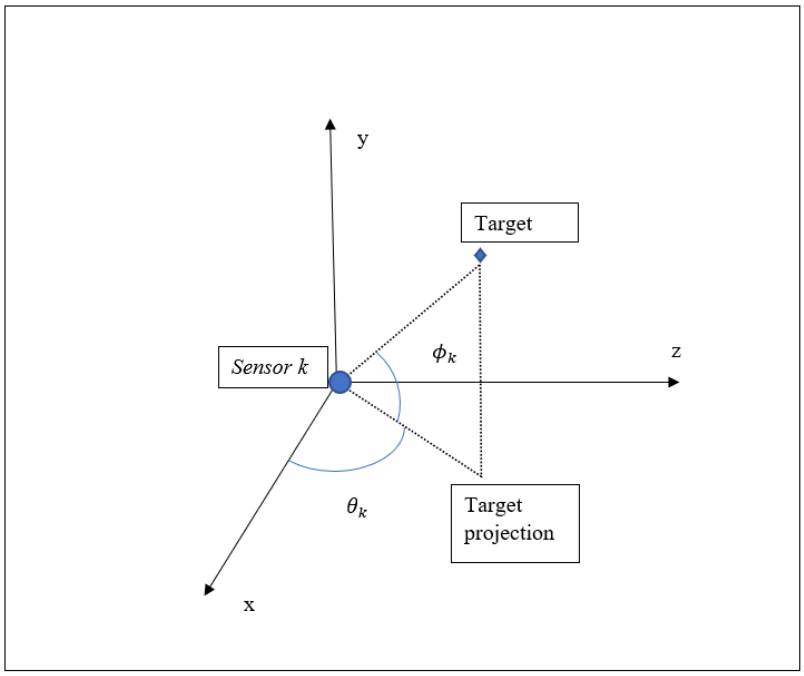

Figure1.2: A 3-D AOA geometrical projection

The ideal measurement without noise

$$\theta_k = \tan^{-1}\frac{y_e - y_k}{x_e - x_k}, \qquad -\pi < \theta_k \leq \pi \tag{1.9}$$

and

$$\phi_k = \tan^{-1}\frac{z_e - z_k}{\|[x_e, y_e] - [x_k, y_k]\|}, \qquad -\frac{\pi}{2} < \phi_k \leq \frac{\pi}{2} \tag{1.10}$$

where the real target location is $[x_e,\ y_e,\ z_e]$ and the kth sensor is located at $[x_k,\ y_k,\ z_k]$ in 3D Cartesian coordinates. The distance between the target and the kth sensor is dk and the projected distance in the xy plane is $d_{xyk} = \|[x_e,\ y_e] - [x_k,\ y_k]\| = d_k cos\ \phi_k$ where $\|\cdot\|$ is the second norm [13].

The primary objective is to use various angle measurements obtained by sensors to pinpoint the target in 3D [14]. The placement of sensors has a considerable impact on the accuracy of localization. As a result, we must first evaluate how sensor placement influences target localization accuracy [15].

Then, noisy angle measurements of the k^{th} sensor may be written as

$$z_k = [\theta_k,\ \phi_k]^T + n_k \tag{1.11}$$

where, z_k is the sensor measurement at sensor k, and n_k is the additive zero-mean independent Gaussian noise vector. The noise variances for $\theta_k\ and\ \phi_k$ measurements are $\sigma_{\theta_k} and\ \sigma_{\phi_k}$, respectively. If there are N sensors in the target localization system, the sensor measurement covariance is

$$\Sigma = \begin{bmatrix} \sigma_\theta^2 I_{N\times N} & 0 \\ 0 & \sigma_\phi^2 I_{N\times N} \end{bmatrix}_{2N\times 2N} \tag{1.12}$$

In [15], it is assumed that the i.i.d. noise, in other words, ${\sigma_{\theta_k}}^2 = {\sigma^2}_{\phi_k} = \sigma^2$ Then we can write the Jacobian of measurement errors evaluated at the true azimuth and elevation angles as

$$J_k = \begin{bmatrix} \frac{\sin\theta_1}{d_{xy1}} & \frac{-\cos\theta_1}{d_{xy1}} & 0 \\ \vdots & \vdots & \vdots \\ \frac{\sin\theta_N}{d_{xyN}} & \frac{-\cos\theta_N}{d_{xyN}} & 0 \\ \frac{\sin\phi_1\cos\theta_1}{d_1} & \frac{\sin\phi_1\sin\theta_1}{d_1} & \frac{-\cos^2\phi_1}{d_{xy1}} \\ \vdots & \vdots & \vdots \\ \frac{\sin\phi_N\cos\theta_N}{d_N} & \frac{\sin\phi_N\sin\theta_N}{d_N} & \frac{-\cos^2\phi_N}{d_{xyN}} \end{bmatrix}_{2N\times 3} \tag{1.13}$$

The Fisher information matrix Φ follows:

$$\Phi = J_k^T \Sigma^{-1} J_k = \frac{1}{\sigma^2} J_k^T J_k \tag{1.14}$$

1.2.3. Time of Arrival-TOA

One of the most common methods for resolving the positioning problem is to employ time-of-arrival (TOA) measurements. It requires a time-synchronized network to calculate the distance between two sensor nodes based on the time the signal spends traveling from one node to another node, which may be accomplished using numerous ways. Correlator or matching filter receivers are widely used to estimate the TOA. Three methodologies are used in this thesis to compute TOA measurements: one-way TOA, two-way TOA (TW-TOA), and time-difference-of-arrival (TDOA). The simplest and most popular ranging approach is time of arrival, which is most famously utilized in the Global Positioning System (GPS) [2]. This method relies on knowing the precise time a signal was sent from the target, the precise time the signal arrives at a reference point, and the signal's speed (usually the speed of light). Once you have these, you can use the simple equation to compute the distance from the reference point [5], [7] .

$$d = c \times (t_{arrival} - t_{sent}) \tag{1.15}$$

The speed of light is denoted by *c*. The set of probable target locations can be identified using this distance. This produces a circle in two dimensions with the equation:

$$d = \sqrt{(x_{ref} - x)^2 + (y_{ref} - y)^2} \tag{1.16}$$

The known position of the reference point is (x_{ref}, y_{ref}). The exact position of the target can be calculated by locating the intersection once this set has been calculated for enough reference points (at least three for two-dimensional and at least four for three-dimensional). A 2-D TOA example is shown in the following.

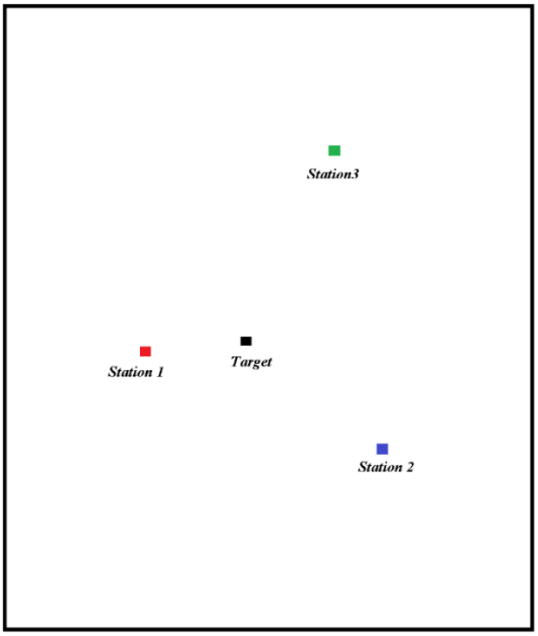

Figure1.3: Example target and station locations

In this case, three well-known stations are attempting to pinpoint the source. At time t_1, a signal is delivered from station 1 to the Target, which is received at time t_2. After calculating the distance (d_1) between Target and station 1, a circle of probable locations is generated, as shown in Figure 1.4 [16].

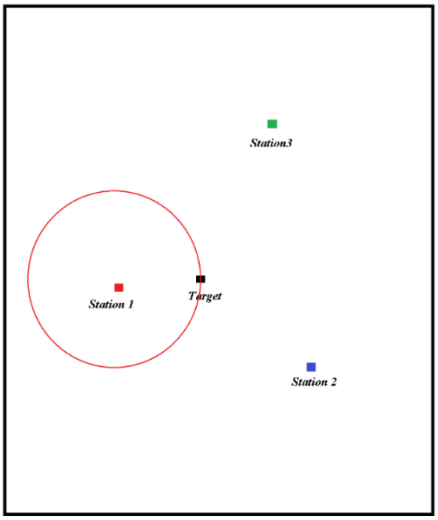

Figure1.4: Possible locations concerning station 1

The same procedure is followed for stations 2 and 3, which result in two more circles, as illustrated in Figure 1.5.

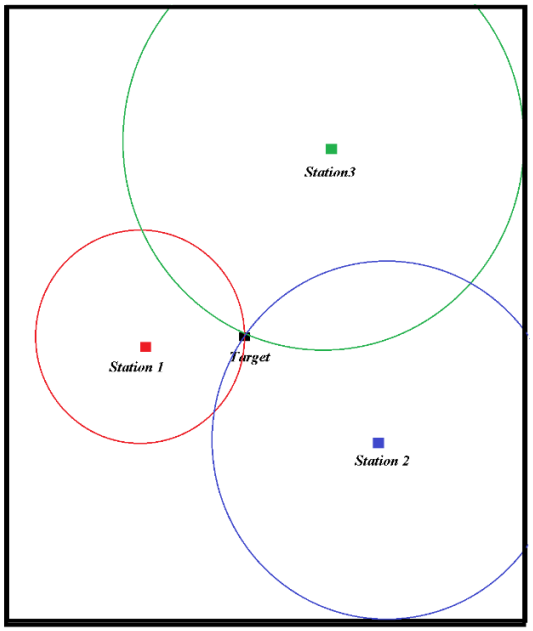

Figure1.5: Possible locations concerning all stations

One Way Time of Arrival:

Assume that the sensor nodes are synced with a common clock and that the transmission is line-of-sight. The TOA estimate for the signal transferred from sensor [3] i to the j_{th} sensor can be calculated using the following equation:

$$\hat{t}_{ij} = \frac{\|x_i - x_j\|}{c} + n_{ij} \tag{1.17}$$

where, n_{ij} is commonly presumed to be a zero-mean Gaussian random variable., i.e., $n_{ij} \sim N(0, \sigma_{ij}^2)$. Then the estimation of the distance is obtained by:

$$\hat{d}_{ij} = c\hat{t}_{ij}\|x_i - x_j\| + cn_{ij} \tag{1.18}$$

Considering the effective bandwidth B_e defined in [17] , the CRLB for the TOA estimate is computed as

$$E\left(\hat{t}_{ij} - E(\hat{t}_{ij})\right)^2 \geq \frac{1}{2\sqrt{2}\pi\sqrt{SNR}B_e} \tag{1.19}$$

It has been discovered that boosting the SNR or effective bandwidth enhances TOA estimation performance [18].

Two Way Time of Arrival:

The distance between two nodes is calculated using round-trip delay estimation in two-way TOA (TW-TOA), which eliminates the necessity for a common time reference. A sensor node i sends a signal to a node j and waits for it to respond in this way.[19] After a certain amount of time has passed, Node j responds with an acknowledgment. As a result, the estimated distance using TW-TOA can be calculated as follows:

$$\hat{d}_{ij} = \|x_i - x_j\| + cT_j^{ar} + c\left(\frac{n_{ij}}{2} + \frac{n_{ji}}{2}\right) \tag{1.20}$$

The TOA estimate errors at node j and node i for the signals broadcast from node *i* and *j*, respectively, are $n_{ij}\ and\ n_{ji}$. T_j^{ar} appears to be known or at the very least accurately estimated. Although TW-TOA eliminates the error caused by a reference node's and a target's faulty synchronization, it still suffers from clock drift in the reference node when measuring TW-TOA. The disadvantage of this method is that, in comparison to the TOA strategy, we must send two signals for each range measurement, which increases the complexity.

Distances calculated via RSSI, one-way TOA, or TW-TOA define numerous circles around the reference nodes for error-free measurements, and the target node is determined at the intersection of them [20]:

1.2.4. Time Difference of Arrival TDOA

The second most common range determining technique is Time Difference of Arrival, which is a little more adaptable than TOA. This approach simply requires the time the signal was received and the speed at which it travels, not the time the signal was transmitted from the target. The difference in arrival time between the target and the two

reference points can be utilized to compute the distance between the target and the two reference points once the signal is received at two reference stations. The following equation can be used to compute the difference:

$$\Delta d = c \times (\Delta t) \tag{1.21}$$

where c denotes the speed of light and t denotes the time difference between each reference point. This gives rise to the following equation in two dimensions:

$$\Delta d = \sqrt{(x_2 - x)^2 - (y_2 - y)^2} - \sqrt{(x_1 - x)^2 - (y_1 - y)^2} \tag{1.22}$$

The known positions of the stations are (x_1, y_1) and (x_2, y_2). This equation can be changed to the form of a hyperbola via nonlinear regression. After calculating enough hyperbolas, the intersection can be used to determine the target's location [21].

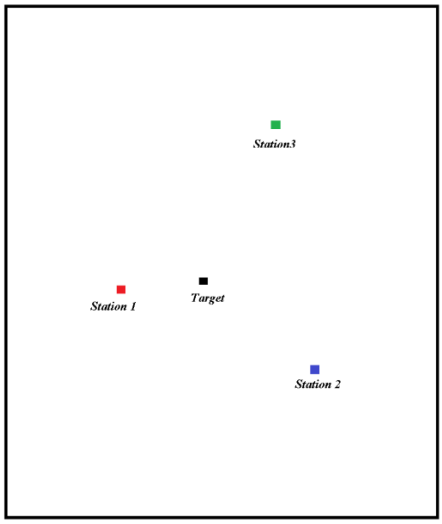

Figure1.6: Example Target and station locations.

We have the same configuration of a target (black) surrounded by three stations in this scenario (red, green, and blue). At an unknown moment, the Target sends a signal, which is received by station 2 at t1 and station 3 at t2. As illustrated in Figure 1.6, the difference in distance (Δd) is calculated, and the hyperbola of probable places is displayed.

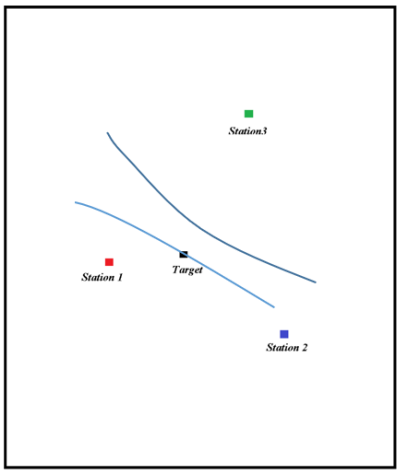

Figure1.7: Possible locations concerning stations 2 and 3

Because this hyperbola has two branches, it will be more difficult to determine the intersection. One of the branches can be deleted if the target's approximate location is known (e.g., through a previously measured location). The top branch is discarded in this scenario. Figure 1.8 shows the result of repeating this technique with the remaining Signal pairings.

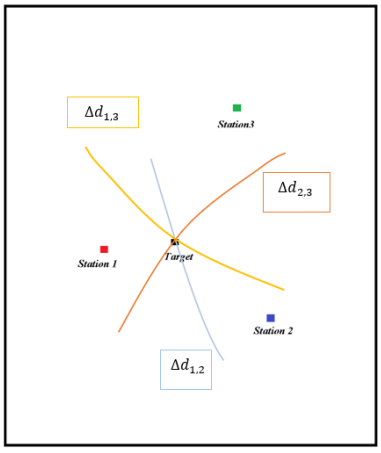

Figure1.8: Possible locations concerning all stations

The Target lies at the signal owing to walls, furniture, or people, as shown in Figure 1.8. Modeling the Non-Line of Sight path and employing low-interference transmissions like Ultra-Wideband are two methods that have been offered for this challenge.

TDOA measurements:

Time-difference-of-arrival (TDOA) instead of measuring the absolute distance between two nodes, measures the distance difference between an unknown node and two synchronized reference nodes. The GPS system employs this method, which involves a receiver in an unknown location measuring the TDOA of received signals from two synchronized satellites. The TDOA between a target node x_i and synchronized sensor nodes at coordinates x_i and x_k, for instance, can be represented as

$$\hat{t}_{jk}^{i} = \hat{t}_{ij} - \hat{t}_{ik} = \frac{\|x_i - x_j\|}{c} - \frac{\|x_i - x_k\|}{c} + n_{ij} - n_{ik} \tag{1.23}$$

As a result, a measurement of the distances between nodes j and k and node i can be represented as

$$\hat{d}_{jk}^{i} = c\hat{t}_{jk}^{i} = \|x_i - x_j\| - \|x_i - x_k\| + c(n_{ij} - n_{ik}) \tag{1.24}$$

$\hat{d}_{jk}^{i}$ and $\hat{t}_{jk}^{i}$, for example, are linked by n_{ik}. Each TDOA measurement defines a hyperbola with a constant distance difference between nodes j and k at all points on the hyperbola.

Propagation effects, clock flaws, and interference are the most common sources of inaccuracy in time-based range. Multipath fading, direct-path delay, and direct-path blocking are some of the propagation effects. Range estimations have substantial inaccuracies due to poor synchronization between nodes. Finally, interference from other signals in the same (or surrounding) frequency band will degrade the range estimation [22].

The received signals are

$$y_i(t) = a_i s(t - \tau_i) + e_i(t), i = 1,2, \dots n \tag{1.25}$$

When the receiver i is at x_i, y_i and the transmitter is at i is at x_i, y_i both of which are unknown.

We can directly estimate i (TOA) and estimate (x; y) using a non-linear least-squares framework, such as GPS, if we have a known reference $s(t)$ and perfect synchronization [23].

When dealing with an unknown reference, the most straightforward approach is to compare the received signals pairwise. Assume you have a correlation function that computes a pairwise estimate of

$$\Delta d_{i,j} = v(\tau_i - \tau_j), \quad 1 \leq i \leq n \tag{1.26}$$

where *v* is the speed of sound, light, or water vibrations.

Here, n is the number of receivers, and $(i{:}\,j)$ is an enumeration of all K pairs of receivers, where,

$$K = \binom{n}{2} \tag{1.27}$$

Each $d_{i:j}$ corresponds to positions $(x;\ y)$ along a hyperbola. The hyperbolic function can be written as

$$d_2 = \sqrt{y^2 + \left(x + {}^{D}/_{2}\right)^2},$$

$$d_1 = -\sqrt{y^2 + (x - {}^{D}/_{2})^2}$$

$$\Delta d = d_2 - d_1 = h(x, y, D) = \sqrt{y^2 + \left(x + {}^{D}/_{2}\right)^2} - \sqrt{y^2 + (x - {}^{D}/_{2})^2} \tag{1.28}$$

This equation can be rewritten in a more compact form with some simplifications as

$$\frac{x^2}{a} - \frac{y^2}{b} = \frac{x^2}{\Delta d^2/4} - \frac{y^2}{D^2/4 - \Delta d^2/4} = 1 \tag{1.29}$$

Asymptotes can be found along the lines of this equation's solution:

$$y = \pm \frac{b}{a} x = \sqrt{\frac{D^2/4 - \Delta d^2/4}{\Delta d^2/4}}\, x \tag{1.30}$$

For far-away transmitters, this specifies the angle of arrival.

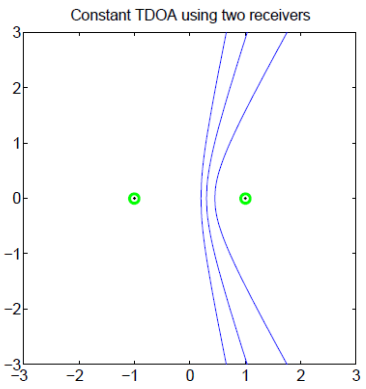

Figure1.9:Illustration of the hyperbolic function in the local coordinate system [12]

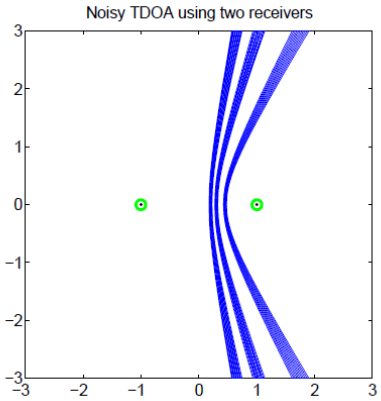

Figure1.10: Similar approach as in Figure 1.9, but with measurement uncertainty in d [12]

We simply convert the hyperbolic function in local coordinates (x; y) to global coordinates (X; Y) for a general receiver position using:

$$\begin{pmatrix} X \\ Y \end{pmatrix} = \begin{pmatrix} X_0 \\ Y_0 \end{pmatrix} + \begin{pmatrix} \cos(\alpha) & -\sin(\alpha) \\ \sin(\alpha) & \cos(\alpha) \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} \tag{1.28}$$

where

$$X_0 = (X_i + X_j)/2\,, and\ Y_0 = (Y_i + Y_j)/2, \tag{1.31}$$

locates the receiver pair's center point. As a result, the hyperbolic function in global coordinates is:

$$\Delta d_{i,j} = h(x, y, D) \tag{1.32}$$

with

$$\begin{gathered} D = \sqrt{\left(Y_i - Y_j\right)^2 + (X_i - X_j)^2} \\ \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} \cos(\alpha) & \sin(\alpha) \\ -\sin(\alpha) & \cos(\alpha) \end{pmatrix} \begin{pmatrix} X & X_0 \\ Y & Y_0 \end{pmatrix} \\ \alpha = \tan^{-1} \frac{Y_i - Y_j}{X_i - X_j} \end{gathered} \tag{1.33}$$

We now have a functional form that may be used to express measurement uncertainty in TDOA, which is a hyperbolic area rather than a line. This can be seen in the illustration in [17]. The greater the absolute uncertainty in the position as one moves further away from the asymptotes. In general, the different techniques can be summarized as:

• RSSI is easy to use and is not time sensitive. It necessitates a precise RSSI-distance dependence model. However, when compared to, perhaps, the TOA-based approach, RSSI estimation is not accurate enough.

• NLOS conditions have a significant impact on AOA. The accuracy is determined by the RF bandwidth and signal-to-noise ratio (SNR).

• TOA/TDOA is a precise method that suffers from NLOS issues. The precision of precisely synchronized networks is determined by the RF bandwidth and SNR [23].

1.2.5. Hybrid Measurements

In the literature there are several hybrid localization methods for determining the emitter location. In the following chapter, these methods are briefly discussed.

CHAPTER 2

HYBRID LOCALIZATION METHODS

To increase positioning performance, the hybrid algorithm combines the benefits of many positioning methods. The TOA / AOA hybrid positioning algorithm can make full use of signal characteristics, overcome time synchronization errors, and reduce the environmental impact on positioning accuracy [24]. The Gauss-Newton method, the Taylor series expansion method, the two-step maximum likelihood function estimation approach, and the PSO algorithm are all examples of indoor positioning algorithms. For placement, hybrid measures can also be used. TOA/AOA, TDOA/AOA, TDOA/TWTOA, and TOA(TDOA)/RSS are some of the hybrid schemes like in that have been examined in the literature. When the measurement follows a Gaussian distribution, they perform exceptionally well. With only two base stations, it investigated the accuracy of the TOA/ AOA hybrid positioning algorithm. In the presence of the LOS path, Taponecco et al. [9] investigated the TOA-AOA joint estimator for indoor location [25].

The use of semidefinite programming to hybrid localization based on the combination of RSSI and TOA was presented in [26]. For sole RSSI, sole TOA, and the combination of RSSI and TOA, semidefinite programs have been proposed.

In [26], using the hyperbolic equations built from the RD data, the categories are two for source localization. The first method is based on the nonlinear least-squares (NLS) framework, which uses Taylor-series expansion for linearization and solves the problem iteratively. The global minimum of the multimodal NLS cost function corresponds to the maximum-likelihood (ML) position estimate under the conventional assumption that the RD measurements are Gaussian distributed.

An algebraic solution for the position and velocity of a moving source using time differences of arrival (TDOAs) and frequency differences of arrival (FDOAs) of a signal received at several receivers is presented in [27]. To obtain a position estimate, the method uses only multiple weighted least-squares minimizations and does not require initial solution estimations. It avoids the problems of initialization and local convergence that plague the traditional linear iterative technique. At moderate noise levels, the predicted accuracy of the source position and velocity achieves the Cramér–Rao lower bound for Gaussian TDOA and FDOA noise before the thresholding effect occurs.

An alternative solution is presented for hyperbolic position fix in [28], when the TDOA estimate errors are small, the solution is in closed-form, valid for both distant and close sources, and an approximation of the maximum likelihood (ML) estimator [29]. It presents a 2-D localization problem with any array manifold. A precise solution is obtained using three sensors to yield two TDOA estimates. The original set of TDOA equations is changed into another set of equations that are linear in source position coordinates and add an extra variable when four or more sensors are used. An initial solution is provided by the weighted linear LS. A second weighted LS improves the location estimation by taking advantage of the known constraint between source coordinates and the extra variable. The location variance expression is derived. The precision of the estimator's localization is compared to the CRLB results are compared to those found in the literature [24], [30].

The technique provided in [31] simulates a situation where multiple sensors with known locations are detecting an emitter within their range. The aim is to localize the unknown sensor using the input data from both sensors. The input data are time difference of arrival TDOA and both azimuth and elevation angle θ and ϕ.

Another technique based on LS approach is provided in [32]. This method also uses AOA and TDOA measurement in order to estimate the location of the emitter. In fact, the methods provided in [31] and [32] are the most simplest methods in terms of computational complexity. However, their performances have not been scrutinized so far. In order to compare their performances, this thesis aims to conduct simulation

experiments. But before that, the algorithms for the mentioned methods are described below.

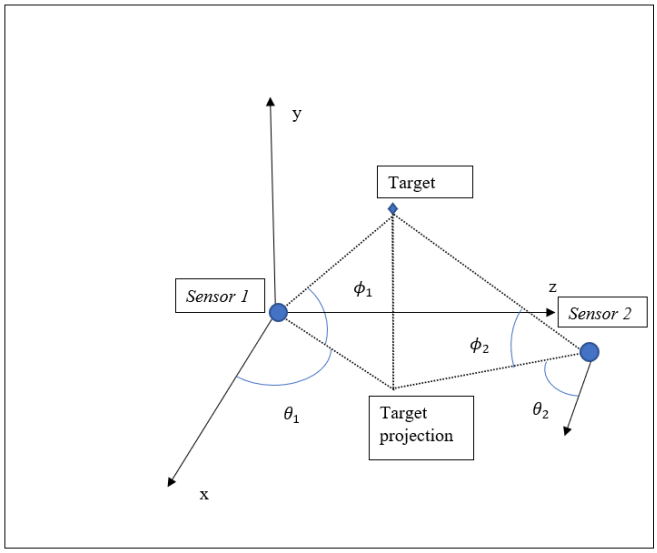

Figure2.11: 3-D Geometry of TDOA-AOA scenario

1.1. The Algorithm of WLS-based Hybrid Method [31]

1.1.1. Measuring the Real Values

The algorithm starts by setting 3-D cartesian values for known sensors, while the other sensors are assumed to be at 300 meters from each other and the reference location, and therefore S_m is set to be at $300 * (cos(\varphi_S), sin(\varphi_S), 0)(-1^{Sensor's\ number})$

$$s_m = [s_{x,m} s_{y,m} s_{z,m}]^T \tag{2.1}$$

where *(T)* is the transpose function needed for the matrices' mathematical equations, therefore $s_m \in \mathbb{R}^{m \times 3}$.

To show that the algorithm can determine u under different practical circumstances, the value of the unknown sensor is evaluated with different values.

$$u = [u_x u_y u_z]^T \tag{2.2}$$

The next step is to calculate the real value of the range difference (RD's) between the known sensors s_1, s_2, …, s_m, and the unknown sensor u.

$$r_m = \|u - s_m\| \; and \; r_1 = \|u - s_1\| \tag{2.3}$$

where $r_1 \; and \; r_m$ are the RD's between U and $s_{1,2,..,m}$, and || . || is the second norm.

Normally, RD's are different from the time of arrival TOA, the difference is the speed of light, which is 3 * 10^8; however, for simplicity, in this thesis, we are assuming that RD's end TOA's, are the same thing, and therefore, T21 in this next equation will be referred to as the time difference of arrival but its value is in meters.

$$T21 = r_2 - r_1 \tag{2.4}$$
$$Tm1 = r_m - r_1$$

Other than T21, the two known sensors S1 and S2, are assumed to have the ability to measure the azimuth angle (θ) and the elevation angle (ϕ), and for that, $\theta_m \; and \; \phi_m$ real-values calculation are in order as follows:

$$\theta_m = atan2(u_y - s_{y,m}, u_x - s_{x,m}) \tag{2.5}$$

and

$$\phi_m = atan2\left(u_z - s_{z,m}, \sqrt{(u_x - s_{x,m})^2 + (u_y - s_{y,m})^2}\right) \tag{2.6}$$

Having the real values of T21, $\theta_m, and \; \phi_m$ a vector K is constructed to represent all these values and to allow us to apply gaussian randomization in the second step.

$$K = [T_{21}, K_1^T, K_2^T]^T \qquad \text{K} \in \mathbb{R}^{3m-1} \tag{2.7}$$

where

$$K_m = \begin{bmatrix} atan2(u_y - s_{y,m}, u_x - s_{x,m}) \\ atan2\left(u_z - s_{z,m}, \sqrt{(u_x - s_{x,m})^2 + (u_y - s_{y,m})^2}\right) \end{bmatrix} = \begin{bmatrix} \theta_m \\ \phi_m \end{bmatrix}, m = 1,2. \ \& \ K_m \in \mathbb{R}^{2 \times m} \quad (2.8)$$

1.1.2. Randomizing the Input

Vector K in equation (2.7) represents all the exact values of the input data, so to speak. Having evaluated that, the actual value vector K will include an error and therefore, $\widehat{K}$ here well represent the input with every possible error according to Gaussian distribution.

$$\widehat{K} = \left[\hat{T}_{21}, \widehat{K}_1^T, \widehat{K}_2^T\right]^T \quad (2.9)$$

where,

$$\widehat{K} = \square + \square \quad (2.10)$$

To evaluate $\widehat{K}$, a covariance matrix Q is calculated based on the variances σ_{RD} and σ_{AOA}, whose values will be simulated differently, to evaluate the efficiency of the algorithm. Their effect on the other hand will be throughout vector $\widehat{K}$ on the range difference, calculated in (2.4) and the angles of arrival AOAs, calculated in (2.7) and (2.8), respectively.

$$Q = \text{diag}[\ \sigma_{RD}^2, \sigma_{AOA}^2, \sigma_{AOA}^2, \sigma_{AOA}^2, \sigma_{AOA}^2] \quad (2.11)$$

Having the matrix Q, we apply a multivariant normal distribution function on the vector K to calculate the approximate value of K, ($\widehat{K}$). the process of choosing the approximate values of vector K is done in the algorithm repeatedly and randomly for L number of times.

The probability density function (pdf) of the d-dimensional multivariate normal distribution is:

$$y = f(x,\mu,\mathrm{Q}) = \frac{1}{\sqrt{|\mathrm{Q}|} * (2\pi)^d} exp^{-\frac{1}{2}(x-\mu)* \mathrm{Q}^{-1}(x-\mu)'} \quad (2.12)$$

where x and μ are 1-by-d vectors and Q is a d-by-d symmetric, positive definite matrix. Only the mvnrnd function allows positive semi-definite Q matrices, which can be singular. The pdf cannot have the same form when Q is singular.

1.1.3. Applying the Algorithm on the Randomized Input

First, we calculate a vector bm for every sensor using the randomized values, as they represent the values from the sensors within the possibility of their error in the real-life appliance.

$$b_m = [\cos\phi_m \cos\theta_m , \cos\phi_m \sin\theta_m , \sin\phi_m]^T \quad (2.13)$$

$$b_m \in \mathbb{R}^{3 \times m}$$

where bm is the unit vector of the source location. It follows the equation.

$$u - s_m = r_m b_m, \qquad m = 1,2, \quad (2.14)$$

Then we compute a matrix G_m through $\theta_m \; and \; \phi_m$, being taken from the randomized values vector ($\widehat{\mathrm{K}}$) as well, assuming at this point the algorithm is only using these values as they represent the real-life values that the algorithm will later be applied on.

$$G_m = \begin{bmatrix} \sin\theta_m & \sin\phi_m \cos\theta_m \\ -\cos\theta_m & \sin\phi_m \sin\theta_m \\ 0 & \cos\phi_m \end{bmatrix}, m = 1,2. \quad (2.15)$$

Matrix Gm has special characteristics, the columns of Gm are in orthonormal basis of the plane orthogonal to b_m, in other words, it follows:

$$G_m^T G_m = I_m \; and \; G_m^T b_m = 0_m \quad (2.16)$$

where $I_m \in \mathbb{R}^{m \times m}$ is the unit vector, and $0_m \in \mathbb{R}^m$ is the zero vector.

After that, a matrix T for the two known sensors situation $T \in \mathbb{R}^{3m-1 \times 3m-1}$, and follows

$$T = \begin{bmatrix} -(b_m - b_1)^T b_1 & r_1 b_m^T L_1 & r_2 b_1^T L_2 & \dots & r_m b_1^T L_m \\ 0_2 & T_1 & 0_{2\times 2} & \dots & 0_{2\times 2} \\ \vdots & 0_{2\times 2} & T_2 & \dots & \vdots \\ \vdots & \vdots & \ddots & \ddots & 0_{2\times 2} \\ 0_2 & 0_{2\times 2} & 0_{2\times 2} & 0_{2\times 2} & T_m \end{bmatrix} \tag{2.17}$$

as

$$L_m = \begin{bmatrix} -\cos\phi_m \sin\theta_m & -\sin\phi_m \cos\theta_m \\ \cos\phi_m \cos\theta_m & -\sin\phi_m \sin\theta_m \\ 0 & \cos\phi_m \end{bmatrix} \tag{2.18}$$

and

$$T_m = -r_m \begin{bmatrix} \cos\phi_m & 0 \\ 0 & 1 \end{bmatrix} \tag{2.19}$$

From *T* in (2.17) and Q in (2.13), a matrix W is calculated as follows:

$$W = TQT^T \tag{2.20}$$

From G_m and b_m in (2.13) and (2.12) respectively, the matrices *G* and *h* are calculated as follows:

$$G = [2(b_m - b_1), G_1, G_2, \dots, G_m], \qquad G \in \mathbb{R}^{3\times 3m-1} \tag{2.21}$$

and

$$h = [(b_m - b_1)^T (s_1 + s_m - T_{m1} b_1), s_1^T G_1, s_2^T G_2, \dots, s_m^T G_m], \tag{2.22}$$

The unknown sensor's location $\hat{u}$, is calculated from all the previously randomized measurements follows the next equation, this will allow us to determine the source location and know with whatever available sensors or data, to evaluate the efficiency of

$$\begin{aligned} \hat{u} &= \arg\min\left(\hat{h} - \hat{G}^T u\right)^T W^{-1} \left(\hat{h} - \hat{G}^T u\right) \\ &= \left(\hat{G} W^{-1} \hat{G}^T\right)^{-1} \hat{G} W^{-1} \hat{h} \end{aligned} \tag{2.23}$$

with the least square method we use the same equation except we ignore the weight presented in the matrix W therefore the equation becomes as follows:

$$\hat{u} = \arg\min\left(\hat{h} - \hat{G}^T u\right)^T \left(\hat{h} - \hat{G}^T u\right) = \left(\hat{G}\hat{G}^T\right)^{-1} \hat{G}\hat{h} \tag{2.24}$$

CHAPTER 3

SIMULATIONS

This chapter covers the results, discussion, and conclusion of the implemented method, comparing the results with the original implementation of the algorithm. Simulation parameters are listed in the following table.

Table 2.1: Simulation parameters

Parameters	Values/Considerations
S	$300 * (cos\,\varphi_S, sin\,\varphi_S, 0) * (Sensor's\ number) * (-1^{SN})$
u	$[10^3\ 10^3\ 10^3]$
σ_{RD}	$[1 :\ 10^3]$
σ_{AOA}	[0.25: 0.25: 2.5]
NoS	2,3,4,5,6,7

In the simulations, the final implementation is comparing σ_{RD}in meters with the RMSE associated with the σ_{RD} used. After finding $\hat{u}$, the RMSE is found for each σ_{RD} from the equation (2.24) in the previous chapter. The loop *N* is set to take a 'σ_{AOA}' from the set every round 'n' and run it 5000 times again the number of trails.

To evaluate the source location's measurement, the root means squared error between the estimated location $\hat{u}$ and the real value of its location is measured as follows:

$$RMSE(u) = \sqrt{\sum_{l=1}^{L} \|\hat{u} - u\|^2 / L} \tag{2.25}$$

where L is the number of rounds the algorithm is run. In other words, for each " l " round there is an approximate evaluation of $\hat{u}$, the source location, $\hat{u}_l$.

Six simulations are conducted to show the RMSE which stands for the root means squared error between the simulation result of the unknown sensor's location and the real coordinate of its location. In the first implementation, the RMSE is shown in comparison with the standard deviation of the error in range difference (σ_{RD}) between the known sensors and the emitter. Besides, in the simulations, the number of sensors is varied between 2 and 7. Results are shown in the following figures.

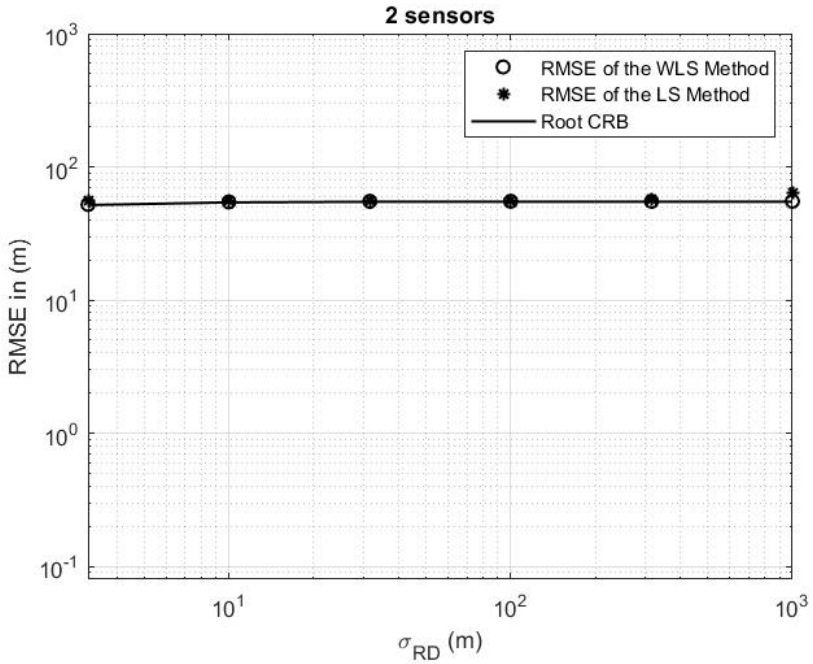

Figure 3.1: Simulation results for 2 sensors

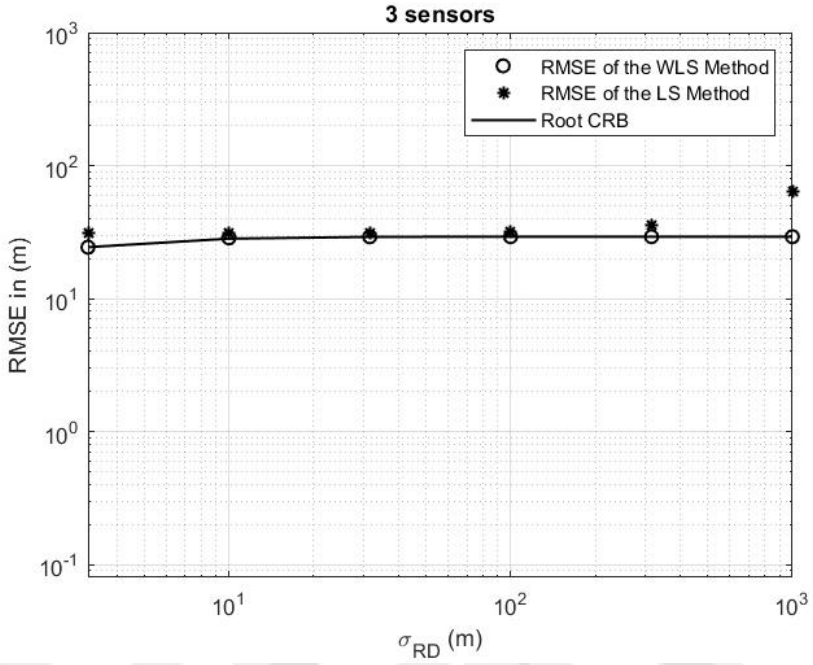

Figure 3.2: Simulation results for 3 sensors

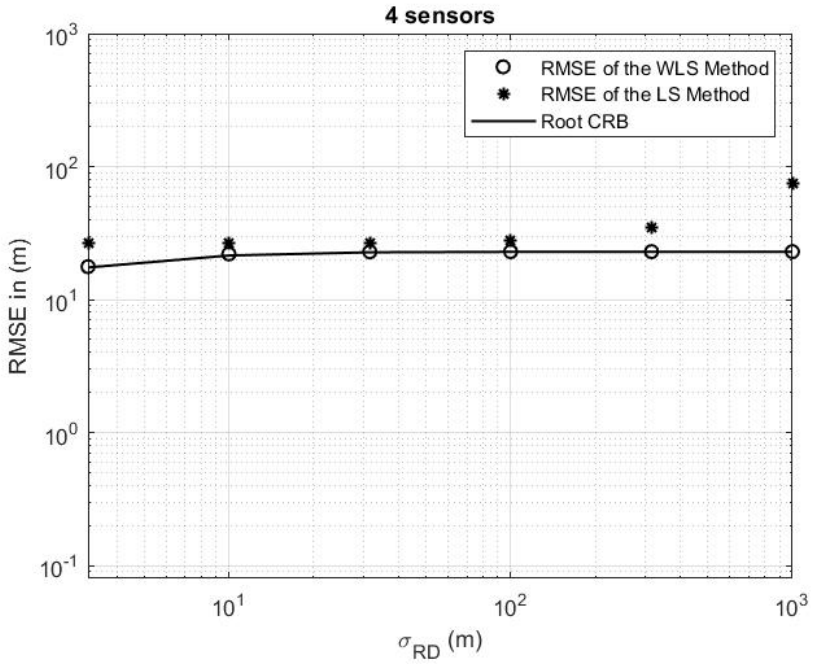

Figure 3.3:Simulation results for 4 sensors

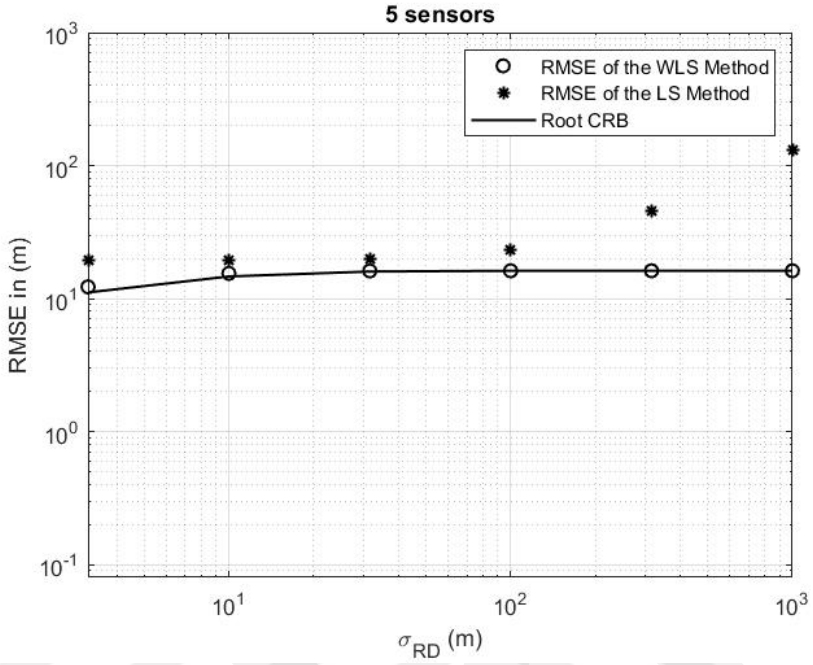

Figure 3.4: Simulation results for 5 sensors

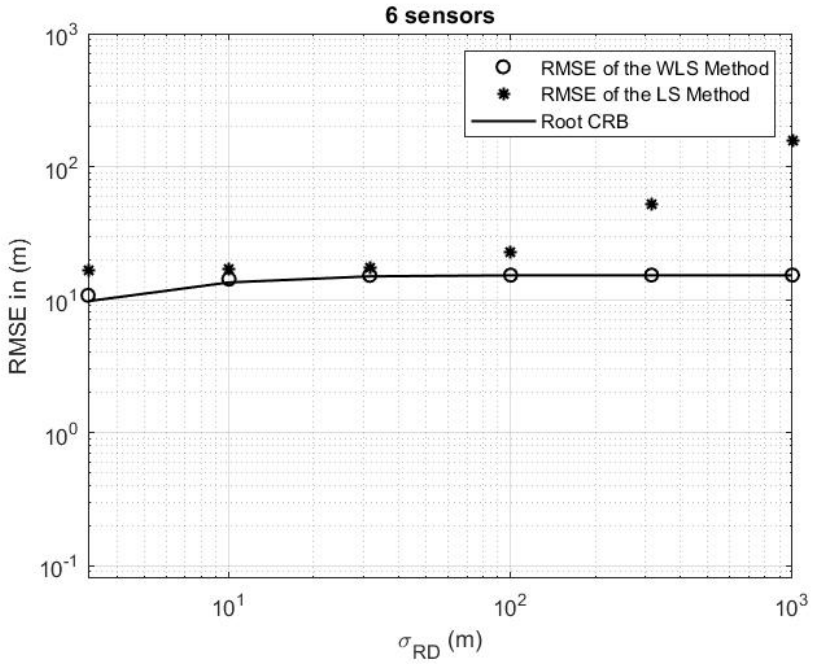

Figure 3.5: Simulation results for 6 sensors

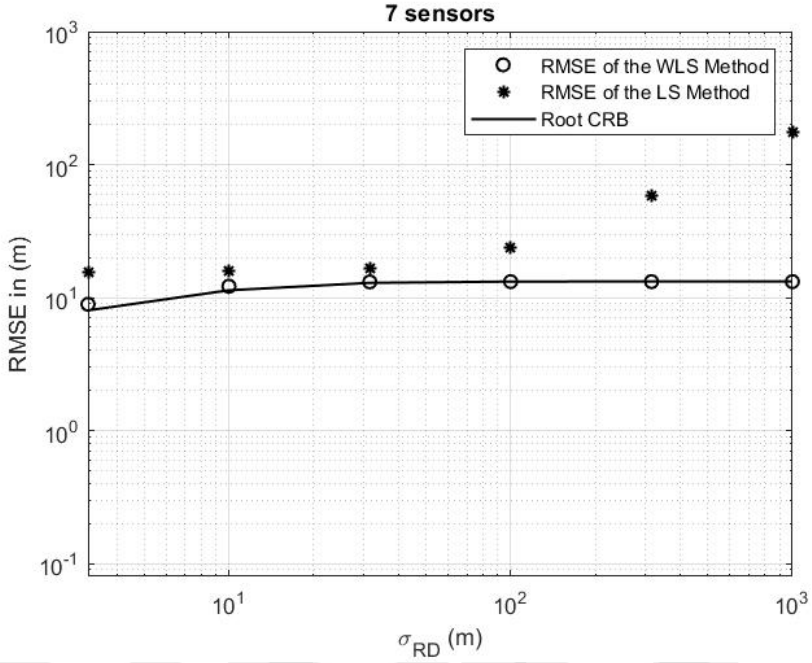

Figure 3.6: Simulation results for 7 sensors

From the results, it is seen that the first result of the simulation is similar to those in [31], [32]. However, the algorithm is examined in multiple sensors scheme and compared with the least squared method of determining the location of an unknown emitter.

The performance of the algorithms tested in this thesis is showing the efficiency of weighted least squared method that is because the result of the mentioned method seems to improve as the number of sensors increases.

All findings show better performance of WLS method over LS method, nonetheless, the LS method shows similar results when the number of sensors is low or when σ_{RD} is low as well.

CHAPTER 4

CONCLUSION

In this thesis, a comparison between the performance of the least squared method with the weighted least squared method for the localization of an unknown emitter is proposed, the algorithms implemented above are an improved version or a different evaluation of the algorithms presented in [31], [32]. In which the methods are examined under different range difference standard deviation σ_{RD} and a multisensory scenario of the stations detecting the unknown source.

The performance of the weighted least squared method shows that it is barely affected by σ_{RD}. In other words, the effect of a higher σ_{RD} does not affect the RMSE that much. Which says that this algorithm is designed to tolerate a noisy environment more than the least squared method.

In the overall look at the results, we can see that the higher the number of sensors used the lower the RMSE for both WLS and LS methods, which makes sense, as the ability to predict the emitter location is easier using a higher number of sensors.

The simulation of a simple method for the 3D passive source localization problem using multiple observation stations by exploiting the hybrid TDOA and AOA measurements matches the results in the studied paper [31] for the two stations, the evaluation of the proposed method, led us to validate the results shown in the paper.

For the LS method, the results show its susceptibility to the higher σ_{RD}, which means it is not the perfect solution for a noisy environment or noisy sensors.

The studied methods are closed form, with performance above the CRB for Gaussian noise over the small error area, where the bias to variance ratio is modest enough to

disregard. In comparison to gradient based MLE, it needs less calculations and does not rely on a reasonable starting guess, and it may have a higher noise tolerance of the threshold effect. The theoretical analysis and simulation results back up the effectiveness of the proposed strategy. It can theoretically be extended to more than two stations while maintaining CRB performance [31].

For further studies, other TDOA-AOA localization methods can be implemented improved and compared with the aim of developing an algorithm that chooses witch method fits the best for witch situation.

REFERENCES

[1] A. Grambozov, V. Atanasovski, and L. Gavrilovska, "Practical evaluation of TDoA, AoA and hybrid methods for geolocation of wireless transmitters," *IEEE Int. Symp. Broadband Multimed. Syst. Broadcast. BMSB*, vol. 2015-Augus, pp. 4–8, 2015.

[2] S. M. Nekooei and M. T. Manzuri-Shalmani, "Location finding in wireless sensor network based on soft computing methods," *2011 Int. Conf. Control. Autom. Syst. Eng. CASE 2011*, no. October, pp. 1–6, 2011.

[3] X. Zhu and Y. Feng, "RSSI-based Algorithm for Indoor Localization," *Commun. Netw.*, vol. 05, no. 02, pp. 37–42, 2013.

[4] C. Shen, C. Wang, K. Zhang, X. Wang, and J. Liu, "A time difference of arrival/angle of arrival fusion algorithm with steepest descent algorithm for indoor non-line-of-sight locationing," *Int. J. Distrib. Sens. Networks*, vol. 15, no. 6, 2019.

[5] H. Khan, M. N. Hayat, and Z. Ur Rehman, "Wireless sensor networks free-range base localization schemes: A comprehensive survey," *Proc. 2017 Int. Conf. Commun. Comput. Digit. Syst. C-CODE 2017*, pp. 144–147, 2017.

[6] S. F. Chuang, W. R. Wu, and Y. T. Liu, "High-Resolution AoA Estimation for Hybrid Antenna Arrays," in *IEEE Transactions on Antennas and Propagation*, vol. 63, no. 7, IEEE, 2015, pp. 2955–2968.

[7] A. Kaur, P. Kumar, and G. P. Gupta, "A weighted centroid localization algorithm for randomly deployed wireless sensor networks," *J. King Saud Univ. - Comput. Inf. Sci.*, vol. 31, no. 1, pp. 82–91, 2019.

[8] Y. Dalveren, "Use Of Multipaths In Passive Localization Of Emitters," Ph.D.

thesis, Atilim University, Turkey, 2016, pp. 1–69.

[9] L. Taponecco, A. A. D'Amico, and U. Mengali, "Joint TOA and AOA estimation for UWB localization applications," *IEEE Trans. Wirel. Commun.*, vol. 10, no. 7, pp. 2207–2217, 2011.

[10] Y. Sharma and V. Gulhane, "Hybrid mechanism for multiple user indoor localization using smart antenna," in *International Conference on Advanced Computing and Communication Technologies, ACCT*, vol. 2015-April, IEEE, 2015, pp. 602–607.

[11] R. S. Rosli, M. H. Habaebi, and M. R. Islam, "Characteristic Analysis of Received Signal Strength Indicator from ESP8266 WiFi Transceiver Module," *Proc. 2018 7th Int. Conf. Comput. Commun. Eng. ICCCE 2018*, pp. 504–507, 2018.

[12] Fredrik Gustafssonk, "POSITIONING USING TIME-DIFFERENCE OF ARRIVAL MEASUREMENTS Fredrik Gustafsson and Fredrik Gunnarsson Department of Electrical Engineering," *Proc. IEEE Int. Conf. Acoust Speech Signal Process*, pp. 8–11, 2003.

[13] P. Sinha, Y. Yapici, and I. Guvenc, "Impact of 3D Antenna Radiation Patterns on TDOA-Based Wireless Localization of UAVs," *INFOCOM 2019 - IEEE Conf. Comput. Commun. Work. INFOCOM WKSHPS 2019*, pp. 614–619, 2019.

[14] J. Liang, D. Wang, L. Su, B. Chen, H. Chen, and H. C. So, "Robust MIMO radar target localization via nonconvex optimization," *Signal Processing*, vol. 122, pp. 33–38, 2016.

[15] S. Xu and K. Doğançay, "Optimal sensor deployment for 3D AOA target localization," *ICASSP, IEEE Int. Conf. Acoust. Speech Signal Process. - Proc.*, vol. 2015-Augus, no. April 2015, pp. 2544–2548, 2015.

[16] Z. Zheng, J. Hua, Y. Wu, H. Wen, and L. Meng, "Time of arrival and Time Sum of arrival based NLOS identification and localization," *Int. Conf. Commun. Technol. Proceedings, ICCT*, pp. 1129–1133, 2012.

[17] F. O. Akgul, M. Heidari, N. Alsindi, and K. Pahlavan, *Localization algorithms and strategies for wireless sensor networks: Monitoring and surveillance techniques for target tracking*. USA: Information Science Reference (an imprint of IGI Global) , 2009, pp. 54–95.

[18] S. G. Nagarajan, P. Zhang, and I. Nevat, "Geo-Spatial Location Estimation for Internet of Things (IoT) Networks with One-Way Time-of-Arrival via Stochastic Censoring," in *IEEE Internet of Things Journal*, vol. 4, no. 1, IEEE, 2017, pp. 205–214.

[19] S. Zhao, X. P. Zhang, X. Cui, and M. Lu, "Optimal Two-Way TOA Localization and Synchronization for Moving User Devices with Clock Drift," *IEEE Trans. Veh. Technol.*, vol. 70, no. 8, pp. 7778–7789, 2021.

[20] K. W. K. Lui, F. K. W. Chan, and H. C. So, "Semidefinite programming approach for range-difference based source localization," *IEEE Trans. Signal Process.*, vol. 57, no. 4, pp. 1630–1633, 2009.

[21] H. C. Schau and A. Z. Robinson, "Passive Source Localization Employing Intersecting Spherical Surfaces from Time-of-Arrival Differences," *IEEE Trans. Acoust.*, vol. 35, no. 8, pp. 1223–1225, 1987.

[22] B. O 'keefe, "ECE Senior Capstone Project Finding Location with Time of Arrival and Time Difference of Arrival Techniques," *Tech Notes, ,* vol. 1, no. 1, pp. 1-3, 2017.

[23] C. Shi, Y. Zheng, H. Qiu, and J. Wang, "Nonlinear Gossip Algorithms for Wireless Sensor Networks," *J. Appl. Math.*, vol. 2014, pp. 1-25, 2014.

[24] A. Beck, P. Stoica, and J. Li, "Exact and approximate solutions of source localization problems," *IEEE Trans. Signal Process.*, vol. 56, no. 5, pp. 1770–1778, 2008.

[25] Z. Zhang, S. Kang, and X. Zhang, "A Bayesian Probabilistic TOA/AOA Hybrid Localization Algorithm in Multipath Environments," *2020 IEEE 3rd Int. Conf. Comput. Commun. Eng. Technol. CCET 2020*, pp. 303–307, 2020.

[26] M. Laaraiedh, S. Avrillon, N. Amiot, and B. Uguen, “A semidefinite programming approach to hybrid localization using RSSI and TOA,” *Proc. 8th Work. Position. Navig. Commun. 2011, WPNC 2011*, pp. 106–110, 2011.

[27] K. C. Ho and W. Xu, “An accurate algebraic solution for moving source location using TDOA and FDOA measurements,” *IEEE Trans. Signal Process.*, vol. 52, no. 9, pp. 2453–2463, 2004, doi: 10.1109/TSP.2004.831921.

[28] Y. T. Chan and K. C. Ho, “A Simple and Efficient Estimator for Hyperbolic Location,” *IEEE Trans. Signal Process.*, vol. 42, no. 8, pp. 1905–1915, 1994.

[29] M. P. and J. R. G. Mellen, “Closed-form solution for determining emitter location using time difference of arrival measurements,” *IEEE Transactions On Aerospace And Electronic Systems*,vol. 39, no. 3, 2003, pp. 1056–1058.

[30] K. Agrawal, A. Vempaty, H. Chen, and P. K. Varshney, “Target localization in Wireless Sensor Networks with quantized data in the presence of Byzantine attacks,” *Conf. Rec. - Asilomar Conf. Signals, Syst. Comput.*, vol. 0, no. 1, pp. 1669–1673, 2011.

[31] J. Yin, Q. Wan, S. Yang, and K. C. Ho, “A simple and accurate TDOA-AOA localization method using two stations,” *IEEE Signal Process. Lett.*, vol. 23, no. 1, pp. 144–148, 2016.

[32] T. Jia, H. Wang, X. Shen, J. Gao, and X. Liu, “An accurate TDOA-AOA localization method using structured total least squares,” *Ocean. 2017 - Aberdeen*, vol. 2017-Octob, pp. 1–6, 2017.

APPENDIX A

THE MATLAB CODE FOR THE WLS METHOD

```
clc
clear;
VNTDOA=1;
VNAOA=1-VNTDOA;
m = input ('input No of S = ');
u=[1000;1000;1000];
L1=10;
cr=300;
sigmaTDOA=10^2;
tdoanf=[10^0.5,10,10^1.5,100,10^2.5,1000].^2;
sigmaAOA=(pi/360)^2;
aoanf=([0.5:0.5:2.5]/180*pi).^2;
if (VNTDOA) nf=tdoanf; else nf=aoanf; end;
er=zeros(length(nf),L1);
er1=zeros(length(nf),L1);
ter=zeros(length(nf),1);
er1=zeros(length(nf),L1);
ter1=zeros(length(nf),1);
mser=zeros(length(nf),L1);
tmser=zeros(length(nf),1);
CRBr=zeros(length(nf),L1);
tCRBr=zeros(length(nf),1);
se_bias=zeros(length(nf),L1);
tse_bias=zeros(length(nf),1);
se_bias1=zeros(length(nf),L1);
tse_bias1=zeros(length(nf),1);
L=5000;
for il=1:L1,
    fprintf('.');
    sphai=unifrnd(0,2*pi);
          for im=1:m,
      s0(im,:)=((-1)^im)*im*cr*[cos(sphai),sin(sphai),0];
        S=[s0];
      end
```

```
        [sr sc]=size(S);
    AOA=zeros(2,sr);
        for ir=1:sr,
        AOA(1,ir)=atan2(u(2)-S(ir,2),u(1)-S(ir,1));
        AOA(2,ir)=atan2(u(3)-S(ir,3),sqrt((u(1)-
S(ir,1))^2+(u(2)-S(ir,2))^2));
    end
       TDOA=zeros(sr-1,1);
        r1=norm(u-S(1,:).',2);
        for ir=1:sr-1,
        TDOA(ir)=norm(u-S(ir+1,:).',2)-r1;
    end
    tdoan=randn(L,sr-1);
    for ic=1:sr-1,
        tdoan(:,ic)=tdoan(:,ic)-sum(tdoan(:,ic))/L;
    end
      aoanoise1=randn(L,sr);
    aoanoise2=randn(L,sr);
    for ic=1:sr,
        aoanoise1(:,ic)=aoanoise1(:,ic)-
sum(aoanoise1(:,ic))/L;
        aoanoise2(:,ic)=aoanoise2(:,ic)-
sum(aoanoise2(:,ic))/L;
    end
    for nfnum=1:length(nf),
        if (VNTDOA)
            sigmaTDOA=tdoanf(nfnum);
        else
            sigmaAOA=aoanf(nfnum);
        end;
    Q=zeros(3*sr-1,3*sr-1);
    Q(sr:end,sr:end)=sigmaAOA*eye(2*sr);
    Q(1:sr-1,1:sr-1)=sigmaTDOA*(eye(sr-1)+ones(sr-1))/2;
    G=zeros(3*sr-1,3);
        rowindex1=1;
        rowindex2=1;
        for ir=1:sr-1,
    G(rowindex1,:)=2*[cos(AOA(2,ir+1))*cos(AOA(1,ir+1))-
cos(AOA(2,1))*cos(AOA(1,1)),cos(AOA(2,ir+1))*sin(AOA(1,ir+1
))-cos(AOA(2,1))*sin(AOA(1,1)),sin(AOA(2,ir+1))-
sin(AOA(2,1))];
    rowindex1=rowindex1+1;
        end
    for ir=1:sr,
    G(sr-1+rowindex2,:)=[sin(AOA(1,ir)),-cos(AOA(1,ir)),0];
```

```
    G(sr-
1+rowindex2+1,:)=[sin(AOA(2,ir))*cos(AOA(1,ir)),sin(AOA(2,i
r))*sin(AOA(1,ir)),-cos(AOA(2,ir))];
      rowindex2=rowindex2+2;
        end
    B=zeros(3,sr);
    for ir=1:sr,

B(:,ir)=[cos(AOA(2,ir))*cos(AOA(1,ir));cos(AOA(2,ir))*sin(A
OA(1,ir));sin(AOA(2,ir))];
        end
    Lm=zeros(3,2,sr);
    for ir=1:sr,
    Lm(:,:,ir)=[-cos(AOA(2,ir))*sin(AOA(1,ir)),-
sin(AOA(2,ir))*cos(AOA(1,ir)); ...
    cos(AOA(2,ir))*cos(AOA(1,ir)),-
sin(AOA(2,ir))*sin(AOA(1,ir)); ...
                0,cos(AOA(2,ir))];
        end
        Km=zeros(3,2,sr);
        for ir=1:sr,
            Km(:,:,ir)=[-
sin(AOA(2,ir))*sin(AOA(1,ir)),cos(AOA(2,ir))*cos(AOA(1,ir))
; ...

sin(AOA(2,ir))*cos(AOA(1,ir)),cos(AOA(2,ir))*sin(AOA(1,ir))
; ...
                0,sin(AOA(2,ir))];
        end
        Jm=zeros(3,sr);
        for ir=1:sr,
            Jm(:,ir)=[cos(AOA(1,ir));sin(AOA(1,ir));0];
        end
        T=zeros(3*sr-1,3*sr-1);
        index1=1;
        index2=1;
        for ir=1:sr-1,
            T(index1,index1)=-(B(:,ir+1)-B(:,1)).'*B(:,1);
            T(index1,sr:sr+1)=(2*u-TDOA(ir)*(B(:,ir+1)-
B(:,1))-(S(1+ir,:).'+S(1,:).'-
TDOA(ir)*B(:,1))).'*Lm(:,:,1);
            T(index1,(sr+2+2*(index1-1)):(sr+3+2*(index1-
1)))=(S(1+ir,:).'+S(1,:).'-TDOA(ir)*B(:,1)-
2*u).'*Lm(:,:,1+ir);
            index1=index1+1;
        end
```

```
        for ir=1:sr,
            T(sr-1+index2,sr-1+index2)=(S(ir,:)-
u.')*Jm(:,ir);
            T(sr-1+index2+1,sr-1+index2:sr-
1+index2+1)=(S(ir,:)-u.')*Km(:,:,ir);
            index2=index2+2;
        end
        mse=inv((inv(T)*G)'*inv(Q)*inv(T)*G);
        mser(nfnum,il)=trace(mse);
        D=zeros(3*sr-1,3);
        index2=1;
        for ir=1:sr-1,
            D(ir,:)=[(u(1)-S(ir+1,1))/norm(u-
S(ir+1,:).',2)-(u(1)-S(1,1))/norm(u-S(1,:).',2),(u(2)-
S(ir+1,2))/norm(u-S(ir+1,:).',2)-(u(2)-S(1,2))/norm(u-
S(1,:).',2),(u(3)-S(ir+1,3))/norm(u-S(ir+1,:).',2)-(u(3)-
S(1,3))/norm(u-S(1,:).',2)];
        end
        for ir=1:sr,
            D(sr-1+index2,:)=[ -(u(2)-S(ir,2))/((u(1)-
S(ir,1))^2+(u(2)-S(ir,2))^2),(u(1)-S(ir,1))/((u(1)-
S(ir,1))^2+(u(2)-S(ir,2))^2),0];
            D(sr-1+index2+1,:)=[-(u(1)-S(ir,1))*(u(3)-
S(ir,3))/((u-S(ir,:).')'*(u-S(ir,:).')*sqrt((u(1)-
S(ir,1))^2+(u(2)-S(ir,2))^2)),-(u(2)-S(ir,2))*(u(3)-
S(ir,3))/((u-S(ir,:).')'*(u-S(ir,:).')*sqrt((u(1)-
S(ir,1))^2+(u(2)-S(ir,2))^2)),sqrt((u(1)-S(ir,1))^2+(u(2)-
S(ir,2))^2)/((u-S(ir,:).')'*(u-S(ir,:).'))];
            index2=index2+2;
        end
        FIM=D'*inv(Q)*D;
        CRB=inv(FIM);
        CRBr(nfnum,il)=trace(CRB);
        ebias=zeros(length(u),1);
        ebias1=zeros(length(u),1);
        for ensemnum=1:L,
            eTDOA=zeros(sr-1,1);
            for ir=1:sr-1,

eTDOA(ir,1)=TDOA(ir,1)+sqrt(sigmaTDOA)*tdoan(ensemnum,ir);
            end
            eAOA=zeros(2,sr);
            for ir=1:sr,

eAOA(1,ir)=AOA(1,ir)+sqrt(sigmaAOA)*aoanoise1(ensemnum,ir);
```

```
eAOA(2,ir)=AOA(2,ir)+sqrt(sigmaAOA)*aoanoise2(ensemnum,ir);
            end
            eG=zeros(3*sr-1,3);
            rowindex1=1;
            rowindex2=1;
            for ir=1:sr-1,

eG(rowindex1,:)=2*[cos(eAOA(2,ir+1))*cos(eAOA(1,ir+1))-
cos(eAOA(2,1))*cos(eAOA(1,1)),cos(eAOA(2,ir+1))*sin(eAOA(1,
ir+1))-cos(eAOA(2,1))*sin(eAOA(1,1)),sin(eAOA(2,ir+1))-
sin(eAOA(2,1))];
                rowindex1=rowindex1+1;
            end
            for ir=1:sr,
                eG(sr-1+rowindex2,:)=[sin(eAOA(1,ir)),-
cos(eAOA(1,ir)),0];
                eG(sr-
1+rowindex2+1,:)=[sin(eAOA(2,ir))*cos(eAOA(1,ir)),sin(eAOA(
2,ir))*sin(eAOA(1,ir)),-cos(eAOA(2,ir))];

                rowindex2=rowindex2+2;
            end
            eB=zeros(3,sr);
            for ir=1:sr,

eB(:,ir)=[cos(eAOA(2,ir))*cos(eAOA(1,ir));cos(eAOA(2,ir))*s
in(eAOA(1,ir));sin(eAOA(2,ir))];
            end
            eh=zeros(3*sr-1,1);
            for ir=1:sr-1,
                eh(ir,1)=eG(ir,:)/2*(S(ir+1,:).'+S(1,:).'-
eTDOA(ir)*eB(:,1));
            end
            index1=1;
            index2=1;
            for ir=sr:2:3*sr-1,
                eh(sr-1+index2,1)=eG(ir,:)*S(index1,:).';
                eh(sr-
1+index2+1,1)=eG(ir+1,:)*S(index1,:).';
                index1=index1+1;
                index2=index2+2;
            end
            eW=eye(3*sr-1);
            eu=inv(eG'*inv(eW)*eG)*eG'*inv(eW)*eh;
            eu1=inv(eG'*eG)*eG'*eh;
```

```
            eLm=zeros(3,2,sr);
            for ir=1:sr,
                eLm(:,:,ir)=[-
cos(eAOA(2,ir))*sin(eAOA(1,ir)),-
sin(eAOA(2,ir))*cos(eAOA(1,ir)); ...
                    cos(eAOA(2,ir))*cos(eAOA(1,ir)),-
sin(eAOA(2,ir))*sin(eAOA(1,ir)); ...
                    0,cos(eAOA(2,ir))];
            end
            eT=zeros(3*sr-1,3*sr-1);
            for ir=1:sr-1,
                eT(ir,ir)=-(eB(:,ir+1)-eB(:,1)).'*eB(:,1);
            end
            for inum=1:2,
            index1=1;
                for ir=1:sr-1,
                    eT(index1,sr:sr+1)=norm(eu-
S(1,:).',2)*eB(:,1+ir).'*eLm(:,:,1);
                    eT(index1,(sr+2+2*(index1-
1)):(sr+3+2*(index1-1)))=-norm(eu-
S(1+ir,:).',2)*eB(:,1).'*eLm(:,:,1+ir);
                    index1=index1+1;
                end
                index1=1;
                for ir=sr:2:3*sr-1,
                    eT(ir,ir)=-norm(eu-
S(index1,:).',2)*cos(eAOA(2,index1));
                    eT(ir+1,ir+1)=-norm(eu-
S(index1,:).',2);
                    index1=index1+1;
                end
                eW=eT*Q*eT';
                eu=inv(eG'*inv(eW)*eG)*eG'*inv(eW)*eh;
                eu1=inv(eG'*eG)*eG'*eh;

            end
            er(nfnum,il)=er(nfnum,il)+(eu-u)'*(eu-u);
            er1(nfnum,il)=er1(nfnum,il)+(eu1-u)'*(eu1-u);
            ebias=ebias+(eu-u);
        end
    er(nfnum,il)=er(nfnum,il)/L;
    er1(nfnum,il)=er1(nfnum,il)/L;
    se_bias(nfnum,il)=sum((ebias/L).^2);
    end
   end
for nfnum=1:length(nf),
```

```
    ter(nfnum)=sum(er(nfnum,:))/L1;
    ter1(nfnum)=sum(er1(nfnum,:))/L1;
    tmser(nfnum)=sum(mser(nfnum,:))/L1;
    tCRBr(nfnum)=sum(CRBr(nfnum,:))/L1;
    tse_bias(nfnum)=sum(se_bias(nfnum,:))/L1;
  end
fprintf('\n');
ter=sqrt(ter);
ter1=sqrt(ter1);
tmser=sqrt(tmser);
tCRBr=sqrt(tCRBr);
tse_bias=sqrt(tse_bias);
figure;
if (VNTDOA)
    tdoaErr=sqrt(tdoanf);

loglog(tdoaErr,ter,'ko',tdoaErr,ter1,'k*',tdoaErr,tCRBr,'k-
','LineWidth',1.2)
    xlabel('\sigma_R_D (m)')
    ylabel('RMSE in (m)')
    legend('RMSE of the WLS Method','RMSE of the LS
Method','Root CRB')
    xlim([10^0.5 1000]);
    ylim([0.8e-1 1000]);
    title ([num2str(m),' sensors'])
    grid
else
    aoaErr=sqrt(aoanf)/pi*180;

semilogy(aoaErr,ter,'ko',tdoaErr,ter1,'k*',aoaErr,tCRBr,'k-
','LineWidth',1.2)
    xlabel('\sigma_A_O_A (degrees)')
    ylabel('RMSE in (m)')
    legend('RMSE of the WLS Method','RMSE of the LS
Method','Root CRB')
    xlim([min(aoaErr) max(aoaErr)]);
    ylim([0.8e-1 1000]);
    title ([num2str(m),' sensors'])
    grid
end;
```

APPENDIX B

THE MATLAB CODE FOR THE LS METHOD

```
clc
clear;
VNTDOA=1;
VNAOA=1-VNTDOA;
m = input ('input No of S = ');
u=[1000;1000;1000];
L1=10;
cr=300;
sigmaTDOA=10^2;
tdoanf=[10^0.5,10,10^1.5,100,10^2.5,1000].^2;
sigmaAOA=(pi/360)^2;
aoanf=([0.5:0.5:2.5]/180*pi).^2;
if (VNTDOA) nf=tdoanf; else nf=aoanf; end;
er1=zeros(length(nf),L1);
er1=zeros(length(nf),L1);
ter1=zeros(length(nf),1);
mser=zeros(length(nf),L1);
tmser=zeros(length(nf),1);
CRBr=zeros(length(nf),L1);
tCRBr=zeros(length(nf),1);
L=5000;
for il=1:L1,
    fprintf('.');
    sphai=unifrnd(0,2*pi);
          for im=1:m,
      s0(im,:)=((-1)^im)*im*cr*[cos(sphai),sin(sphai),0];
        S=[s0];
      end
        [sr sc]=size(S);
    AOA=zeros(2,sr);
        for ir=1:sr,
        AOA(1,ir)=atan2(u(2)-S(ir,2),u(1)-S(ir,1));
        AOA(2,ir)=atan2(u(3)-S(ir,3),sqrt((u(1)-
S(ir,1))^2+(u(2)-S(ir,2))^2));
    end
       TDOA=zeros(sr-1,1);
```

```
        r1=norm(u-S(1,:).',2);
        for ir=1:sr-1,
        TDOA(ir)=norm(u-S(ir+1,:).',2)-r1;
    end
    tdoan=randn(L,sr-1);
    for ic=1:sr-1,
        tdoan(:,ic)=tdoan(:,ic)-sum(tdoan(:,ic))/L;
    end
      aoanoise1=randn(L,sr);
    aoanoise2=randn(L,sr);
    for ic=1:sr,
        aoanoise1(:,ic)=aoanoise1(:,ic)-
sum(aoanoise1(:,ic))/L;
        aoanoise2(:,ic)=aoanoise2(:,ic)-
sum(aoanoise2(:,ic))/L;
    end
    for nfnum=1:length(nf),
        if (VNTDOA)
            sigmaTDOA=tdoanf(nfnum);
        else
            sigmaAOA=aoanf(nfnum);
        end;
    Q=zeros(3*sr-1,3*sr-1);
    Q(sr:end,sr:end)=sigmaAOA*eye(2*sr);
    Q(1:sr-1,1:sr-1)=sigmaTDOA*(eye(sr-1)+ones(sr-1))/2;
    G=zeros(3*sr-1,3);
        rowindex1=1;
        rowindex2=1;
        for ir=1:sr-1,
    G(rowindex1,:)=2*[cos(AOA(2,ir+1))*cos(AOA(1,ir+1))-
cos(AOA(2,1))*cos(AOA(1,1)),cos(AOA(2,ir+1))*sin(AOA(1,ir+1
))-cos(AOA(2,1))*sin(AOA(1,1)),sin(AOA(2,ir+1))-
sin(AOA(2,1))];
    rowindex1=rowindex1+1;
        end
    for ir=1:sr,
    G(sr-1+rowindex2,:)=[sin(AOA(1,ir)),-cos(AOA(1,ir)),0];
    G(sr-
1+rowindex2+1,:)=[sin(AOA(2,ir))*cos(AOA(1,ir)),sin(AOA(2,i
r))*sin(AOA(1,ir)),-cos(AOA(2,ir))];
      rowindex2=rowindex2+2;
        end
    B=zeros(3,sr);
    for ir=1:sr,
```

```
B(:,ir)=[cos(AOA(2,ir))*cos(AOA(1,ir));cos(AOA(2,ir))*sin(A
OA(1,ir));sin(AOA(2,ir))];
        end
    Lm=zeros(3,2,sr);
    for ir=1:sr,
    Lm(:,:,ir)=[-cos(AOA(2,ir))*sin(AOA(1,ir)),-
sin(AOA(2,ir))*cos(AOA(1,ir)); ...
    cos(AOA(2,ir))*cos(AOA(1,ir)),-
sin(AOA(2,ir))*sin(AOA(1,ir)); ...
                0,cos(AOA(2,ir))];
        end
        Km=zeros(3,2,sr);
        for ir=1:sr,
            Km(:,:,ir)=[-
sin(AOA(2,ir))*sin(AOA(1,ir)),cos(AOA(2,ir))*cos(AOA(1,ir))
; ...

sin(AOA(2,ir))*cos(AOA(1,ir)),cos(AOA(2,ir))*sin(AOA(1,ir))
; ...
                0,sin(AOA(2,ir))];
        end
        Jm=zeros(3,sr);
        for ir=1:sr,
            Jm(:,ir)=[cos(AOA(1,ir));sin(AOA(1,ir));0];
        end
        T=zeros(3*sr-1,3*sr-1);
        index1=1;
        index2=1;
        for ir=1:sr-1,
            T(index1,index1)=-(B(:,ir+1)-B(:,1)).'*B(:,1);
            T(index1,sr:sr+1)=(2*u-TDOA(ir)*(B(:,ir+1)-
B(:,1))-(S(1+ir,:).'+S(1,:).'-
TDOA(ir)*B(:,1))).'*Lm(:,:,1);
            T(index1,(sr+2+2*(index1-1)):(sr+3+2*(index1-
1)))=(S(1+ir,:).'+S(1,:).'-TDOA(ir)*B(:,1)-
2*u).'*Lm(:,:,1+ir);
            index1=index1+1;
        end
        for ir=1:sr,
            T(sr-1+index2,sr-1+index2)=(S(ir,:)-
u.')*Jm(:,ir);
            T(sr-1+index2+1,sr-1+index2:sr-
1+index2+1)=(S(ir,:)-u.')*Km(:,:,ir);
            index2=index2+2;
        end
```

```
        mse=inv((inv(T)*G)'*inv(Q)*inv(T)*G);
        mser(nfnum,il)=trace(mse);
        D=zeros(3*sr-1,3);
        index2=1;
        for ir=1:sr-1,
            D(ir,:)=[(u(1)-S(ir+1,1))/norm(u-
S(ir+1,:).',2)-(u(1)-S(1,1))/norm(u-S(1,:).',2),(u(2)-
S(ir+1,2))/norm(u-S(ir+1,:).',2)-(u(2)-S(1,2))/norm(u-
S(1,:).',2),(u(3)-S(ir+1,3))/norm(u-S(ir+1,:).',2)-(u(3)-
S(1,3))/norm(u-S(1,:).',2)];
        end
        for ir=1:sr,
            D(sr-1+index2,:)=[ -(u(2)-S(ir,2))/((u(1)-
S(ir,1))^2+(u(2)-S(ir,2))^2),(u(1)-S(ir,1))/((u(1)-
S(ir,1))^2+(u(2)-S(ir,2))^2),0];
            D(sr-1+index2+1,:)=[-(u(1)-S(ir,1))*(u(3)-
S(ir,3))/((u-S(ir,:).')'*(u-S(ir,:).')*sqrt((u(1)-
S(ir,1))^2+(u(2)-S(ir,2))^2)),-(u(2)-S(ir,2))*(u(3)-
S(ir,3))/((u-S(ir,:).')'*(u-S(ir,:).')*sqrt((u(1)-
S(ir,1))^2+(u(2)-S(ir,2))^2)),sqrt((u(1)-S(ir,1))^2+(u(2)-
S(ir,2))^2)/((u-S(ir,:).')'*(u-S(ir,:).'))];
            index2=index2+2;
        end
        FIM=D'*inv(Q)*D;
        CRB=inv(FIM);
        CRBr(nfnum,il)=trace(CRB);
            for ensemnum=1:L,
            eTDOA=zeros(sr-1,1);
            for ir=1:sr-1,

eTDOA(ir,1)=TDOA(ir,1)+sqrt(sigmaTDOA)*tdoan(ensemnum,ir);
            end
            eAOA=zeros(2,sr);
            for ir=1:sr,

eAOA(1,ir)=AOA(1,ir)+sqrt(sigmaAOA)*aoanoise1(ensemnum,ir);

eAOA(2,ir)=AOA(2,ir)+sqrt(sigmaAOA)*aoanoise2(ensemnum,ir);
            end
            eG=zeros(3*sr-1,3);
            rowindex1=1;
            rowindex2=1;
            for ir=1:sr-1,

eG(rowindex1,:)=2*[cos(eAOA(2,ir+1))*cos(eAOA(1,ir+1))-
cos(eAOA(2,1))*cos(eAOA(1,1)),cos(eAOA(2,ir+1))*sin(eAOA(1,
```

```
ir+1))-cos(eAOA(2,1))*sin(eAOA(1,1)),sin(eAOA(2,ir+1))-
sin(eAOA(2,1))];
                rowindex1=rowindex1+1;
            end
            for ir=1:sr,
                eG(sr-1+rowindex2,:)=[sin(eAOA(1,ir)),-
cos(eAOA(1,ir)),0];
                eG(sr-
1+rowindex2+1,:)=[sin(eAOA(2,ir))*cos(eAOA(1,ir)),sin(eAOA(
2,ir))*sin(eAOA(1,ir)),-cos(eAOA(2,ir))];

                rowindex2=rowindex2+2;
            end
            eB=zeros(3,sr);
            for ir=1:sr,

eB(:,ir)=[cos(eAOA(2,ir))*cos(eAOA(1,ir));cos(eAOA(2,ir))*s
in(eAOA(1,ir));sin(eAOA(2,ir))];
            end
            eh=zeros(3*sr-1,1);
            for ir=1:sr-1,
                eh(ir,1)=eG(ir,:)/2*(S(ir+1,:).'+S(1,:).'-
eTDOA(ir)*eB(:,1));
            end
            index1=1;
            index2=1;
            for ir=sr:2:3*sr-1,
                eh(sr-1+index2,1)=eG(ir,:)*S(index1,:).';
                eh(sr-
1+index2+1,1)=eG(ir+1,:)*S(index1,:).';
                index1=index1+1;
                index2=index2+2;
            end
            eW=eye(3*sr-1);
            eu1=inv(eG'*eG)*eG'*eh;
            eLm=zeros(3,2,sr);
            for ir=1:sr,
                eLm(:,:,ir)=[-
cos(eAOA(2,ir))*sin(eAOA(1,ir)),-
sin(eAOA(2,ir))*cos(eAOA(1,ir)); ...
                    cos(eAOA(2,ir))*cos(eAOA(1,ir)),-
sin(eAOA(2,ir))*sin(eAOA(1,ir)); ...
                    0,cos(eAOA(2,ir))];
            end
            eT=zeros(3*sr-1,3*sr-1);
            for ir=1:sr-1,
```

```
                eT(ir,ir)=-(eB(:,ir+1)-eB(:,1)).'*eB(:,1);
            end
            for inum=1:2,
            index1=1;
                for ir=1:sr-1,
                    eT(index1,sr:sr+1)=norm(eu1-
S(1,:).',2)*eB(:,1+ir).'*eLm(:,:,1);
                    eT(index1,(sr+2+2*(index1-
1)):(sr+3+2*(index1-1)))=-norm(eu1-
S(1+ir,:).',2)*eB(:,1).'*eLm(:,:,1+ir);
                    index1=index1+1;
                end
                index1=1;
                for ir=sr:2:3*sr-1,
                    eT(ir,ir)=-norm(eu1-
S(index1,:).',2)*cos(eAOA(2,index1));
                    eT(ir+1,ir+1)=-norm(eu1-
S(index1,:).',2);
                    index1=index1+1;
                end
                eW=eT*Q*eT';
                eu1=inv(eG'*eG)*eG'*eh;

            end
            er1(nfnum,il)=er1(nfnum,il)+(eu1-u)'*(eu1-u);
        end
    er1(nfnum,il)=er1(nfnum,il)/L;
    end
   end
for nfnum=1:length(nf),
    ter1(nfnum)=sum(er1(nfnum,:))/L1;
    tmser(nfnum)=sum(mser(nfnum,:))/L1;
    tCRBr(nfnum)=sum(CRBr(nfnum,:))/L1;
   end
fprintf('\n');
ter1=sqrt(ter1);
tmser=sqrt(tmser);
tCRBr=sqrt(tCRBr);
figure;
if (VNTDOA)
    tdoaErr=sqrt(tdoanf);
    loglog(tdoaErr,ter1,'k*',tdoaErr,tCRBr,'k-
','LineWidth',1.2)
    xlabel('\sigma_R_D (m)')
    ylabel('RMSE in (m)')
    legend('RMSE of the LS Method','Root CRB')
```

```
    xlim([10^0.5 1000]);
    ylim([0.8e-1 1000]);
    title ([num2str(m),' sensors'])
    grid
else
    aoaErr=sqrt(aoanf)/pi*180;
    semilogy(tdoaErr,ter1,'k*',aoaErr,tCRBr,'k-
','LineWidth',1.2)
    xlabel('\sigma_A_O_A (degrees)')
    ylabel('RMSE in (m)')
    legend('RMSE of the LS Method','Root CRB')
    xlim([min(aoaErr) max(aoaErr)]);
    ylim([0.8e-1 1000]);
    title ([num2str(m),' sensors'])
    grid
end;
```