Distance Measurement-Based Cooperative Source Localization: A Convex Range-Free Approach

by

Fatma Kiraz

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Master of Applied Science
in
Electrical and Computer Engineering

Waterloo, Ontario, Canada, 2013

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Abstract

One of the most essential objectives in WSNs is to determine the spatial coordinates of a source or a sensor node having information. In this study, the problem of range measurement-based localization of a signal source or a sensor is revisited. The main challenge of the problem results from the non-convexity associated with range measurements calculated using the distances from the set of nodes with known positions to a fixed sensor node. Such measurements corresponding to certain distances are non-convex in two and three dimensions. Attempts recently proposed in the literature to eliminate the non-convexity approach the problem as a non-convex geometric minimization problem, using techniques to handle the non-convexity.

This study proposes a new fuzzy range-free sensor localization method. The method suggests using some notions of Euclidean geometry to convert the problem into a convex geometric problem. The convex equivalent problem is built using convex fuzzy sets, thus avoiding multiple stable local minima issues, then a gradient based localization algorithm is chosen to solve the problem.

Next, the proposed algorithm is simulated considering various scenarios, including the number of available source nodes, fuzzification level, and area coverage. The results are compared with an algorithm having similar fuzzy logic settings. Also, the behaviour of both algorithms with noisy measurements are discussed. Finally, future extensions of the algorithm are suggested, along with some guidelines.

Acknowledgements

I would like to express my sincere gratitude to my supervisors, Prof. Karray and Prof. Fidan for their guidance and support throughout my thesis. Without their supervision, this work would not have been succeeded.

I am indebted to Prof. Fidan for his remarkable advice and time commitment during my studies. I am also very grateful to my readers, Prof. Melek and Prof. Ponnambalam for their time to review this thesis. Special thanks to Prof. Melek for his greatest vision and helpful discussions during his Computational Intelligence course.

Furthermore, I would like to thank all my friends who brought me joy and enriched my life intellectually. Finally, my deepest appreciation goes to my family for their support, patience, understanding and unconditional love.

To my lovely family

Contents

List of Acronyms

WSN	Wireless Sensor Network
GPS	Global Positioning System
ToA	Time of Arrival
TDoA	Time Difference of Arrival
AoA	Angle of Arrival
RSS	Received Signal Strength
ToF	Time of Flight
WLS	Weighted Least Square
ML	Maximum Likelihood
POCS	Projection onto Convex Sets
CI	Computational Intelligence
ROCRSSI	Ring Overlapping Based on Comparison of Received Signal Strength Indicator
TSK	Takagi-Sugeno-Kang
FRORF	Fuzzy Ring Overlapping Range Free
CoG	Center of Gravity
RAFRF	Radical Axis Based Fuzzy Range Free

List of Tables

List of Figures

Chapter 1

Introduction

Recent improvements in radio and embedded systems have resulted in the development of sensor nodes with relatively low power and low cost. Those sensor nodes with wireless interfaces originated the real-life wireless sensor networks (WSNs),which led to change in the means of gathering information using the networks. That change has enabled tremendous opportunities for the networks in many applications. Such applications as detecting hazardous chemicals, monitoring seismic activity [2], healthcare and medical research [3], homeland security and defence-related applications [4], [5], as well as personnel tracking in large complexes, intelligent audio players in self-guided museum tours, and intelligent maps for large malls and/or offices [6].

1.1 Problem Domain

A WSN, in general, consists of a large number of sensor nodes with ad-hoc deployment. These sensors will have varying features, such as infrared, ultrasonic, visible light, sonar,

radar, and orientation detection [7], based on the needs of the specific application. These sensor nodes are used to sense the environment, do some computations, and accomplish several common objectives using sensor intercommunications. After deployment, there will be nodes whose location is unknown, as well as the location-known, i.e., *anchors* or *beacons*, because the capacity to have only location-known sensors is a costly one (in terms of time, computational effort, cost of hardware). However, the information collected from sensors is meaningless in most cases if the physical positions of the sensor nodes which gathered the information are unknown. Such cases include animal habitat monitoring, bush fire surveillance, road traffic monitoring, and inventory management [6]. Also, missing coordinate information of a sensor node can often cause an incorrect interpretation of data. Therefore, it is important for a WSN to have the capability of determining the estimate of spatial coordinates of sensor nodes in the network [8].

WSN localization algorithms utilize the location information of *anchors* and inter-sensor measurements to estimate the locations of sensors with initially unknown position information. The position information of anchors are obtained either by employing a global positioning system (GPS), or by installing anchors manually. Excluding the applications requiring a GPS, a local coordinate system is established by these anchors with reference to all sensors in the system, since GPS receivers are expensive in both financial and power-usage terms [6].

Location discovery techniques employing anchors include trilateration, triangulation, and maximum likelihood multilateration [9]. Trilateration employs the intersection of three circles of three anchor nodes, built using distance measurements between the sensor node to be located and the three anchors as the radii of the corresponding circles. However, the number of anchors needs to be decreased, since distance measurements are usually noisy in real applications. To handle this problem, distance measurements from multiple neighbour

nodes can be used by applying maximum likelihood estimation, which aims to minimize the differences between the measured ones and the estimated ones. In triangulation, location of a sensor node is estimated by utilizing the angle information of anchors to the sensor node and their coordinate information.

Another component of WSN localization algorithms is inter-sensor measurements. Such measurements are computed using time of arrival (ToA), time difference of arrival (TDoA), angle of arrival (AoA), and techniques based on received signal strength (RSS) [6, 10, 11]. ToA, also known as time of flight (ToF), utilizes the travelling time of a specific signal from a transmitter to a receiver during propagation along with the propagation speed, to calculate the distance between the transmitter and the receiver. That this requires the sending and receiving clocks to be synchronized is the primary restriction of ToA. However, TDoA eliminates the restriction by using the combination of two different signals with different propagation speeds. AoA employs the coordinate-based angle of the received signal, using an antenna array, to estimate the position of a sensor. Due to the need for array antennas which are high-priced, use of the AoA technique must be carefully considered.

Among the inter-sensor measurement techniques, RSS-based ones are widely appreciated because they depend on a standard feature found in most wireless devices, thus not requiring any extra hardware cost. On the other hand, calculations of estimates based on RSS can arrive at inaccurate results because of the side effects of the radio propagation environment [10, 12]. Overall, there are several inter-sensor measurement techniques for WSNs, and deciding which technique to use is an application-dependent choice.

1.2 Formal Problem Definition

Formally speaking, accurate position approximation of a source by applying relative data of a group of sensors to the source is called *source localization*, and self-location estimation of a sensor applying similar data relative to several *anchors*, is called *sensor localization*.

The generic definition of a sensor/source localization problem is given by:

Problem 1.2.1 *Given known 2 or 3- dimensional anchors $A_1, \ldots, A_N$ ($N > 2$ and $N > 3$ in 2 and 3 dimensions respectively) and a sensor/source node S with unknown position y^*, estimate the coordinates of S, from the distance estimates/measurements $d_i \approx \|y^* - A_i\|$.*

Here, either S symbolizes the position of the unknown source and the A_i are the positions of sensors seeking to approximate the location of S, or S represents the position of the sensor approximating its position, and the A_i are the positions of the *anchors*, i.e., position-known nodes.

There are many ways to obtain distance estimates, such as using signal intensity and features of the vehicle for a signal sending from a source, calculating time to emit a signal to a known point to return for a self-transmitting sensor, and cooperatively using the TDOA information from a set of sensors. However, they are outside the scope of this study, with the exception of the one expressed in the following for simulation purposes.

This approach is known as received signal strength (RSS). It employs source signal strength A, the distance d from the source, and a power loss coefficient η represented by

$$s = A/d^{\eta} \tag{1.2.1}$$

where s is the source emitting a signal. The distance d can be derived from (1.2) using given values of A, η and s distance d.

1.3 Motivation, Focus and Contributions

A common task in many applications involving WSN problems requiring accurate position information of sensor nodes, is to achieve correlation between sensor readings and coordinates, position-based routing and data aggregation, and thus ease the network's self-configuration and -organization utilizing an *ad-hoc* deployment of large number of sensors. Moreover, getting the position information itself is the goal in several applications such as inventory management and target tracking [1].

One can encounter numerous methods for solving localization problems in literature in terms of conventional and soft computing ones, including: using signal intensity and features of the vehicle for a signal emitting from a source; calculating the time needed to emit a signal to a known point and back for a self-transmitting sensor; cooperatively using the TDOA information of a set of sensors; or by applying fuzzy logic, genetic algorithms, neural networks, and hybrids of these involving multiple methods.

The data used for position approximation ranges from bearing to power level, or from time difference of arrival (TDOA) to received signal strength (RSS). However, distance measurements with or without noise are-the only focus of this study.

To have a better understanding of the difference between conventional and soft computational aspects, one method from each perspective, using a similar distance measurement technique, is applied. One proposed in [13] aims to classify sequential nodes into pairs to utilize the radical axis property of pairs of sensor nodes. Thus, the cost function can be rewritten with respect to that property and minimized by applying a gradient descent algorithm. Such classification ensures highly desirable performance in noisy cases, which can have practical advantages in costs, by reducing the number of sensors/ anchors to deal with, and eliminating the necessity of re-initialization. The other utilizes a fuzzy logic

area-based inference system in [1], using area-codes of a sensor as an input, and estimated location as an output, by applying adequate If-Then rules. The algorithm proposed in [1] follows the weighted centroid principle that all of the weighted estimated locations of a sensor are averaged to find an overall mean.

The contribution in this study is to construct such a fuzzy-based inference system using some geometric notions, namely the radical axis of pairs of sensor nodes, to convert non-convex regions into convex ones. Thus, the potential problems in the position of the center of gravity for the corresponding region can be eliminated, and the execution time can be reduced.

The rest of the study is organized as follows. Chapter 2 presents a historical background and literature review of the most-used conventional and soft computing methods. Chapter 3 explains the proposed algorithm in detail. Chapter 4 evaluates the algorithm proposed by comparing it with the algorithm proposed in [1], and highlights the strengths and weaknesses of each algorithm. Chapter 5 summarizes this work and discusses extension of the proposed algorithm for five or more anchors, followed by proposing some suggestions for potential improvements that can be examined in future studies.

Chapter 2

Background and Literature Review

Source/signal localization problems involving wireless networks have had an increasing significance as a result of advances in radio and embedded systems, and studies on them have become more popular. There have been various approaches to tackle localization problems, which can be categorized, from the algorithmic aspect, into two classes; conventional convex optimization, and soft computing-based approaches. This thesis will aim to bring together a set of tools from these two categories. In this chapter, we provide a review of the literature on both approaches.

2.1 Convex Optimization Based Approaches

Convex optimization [14–17] is one of the most popular frameworks used for developing mathematical models and defining problems for real-life applications. Such applications include modelling, system optimization, and estimation and control tasks as applied to various branches of engineering. In a typical model-forming and problem definition proce-

dure based on convex optimization, the optimization/estimation/control task is formulated using a convex (mostly quadratic) objective function, and a set of linear equality and inequality constraints. The system model is constructed in a parametric form suitable for this formulation. The sensor network localization problem of interest in this paper, Problem 1.2.1, is also defined in the aforementioned framework.

Next, we provide a literature review of studies on range measurement estimation-based localization problems of the form of Problem 1.2.1, or similar to it. In such a problem, the ability of each sensor to measure its distance using an adequate number of anchors is critical in anchor deployment. It is also inferred that there is a non-convexity issue resulting from range measurements, in both two and three dimensions.

A number of algorithms to address sensor or source localization problems have been proposed [6,11,18–23]. In fact, S in Problem 1.2.1 can be estimated using linear algorithms if the signal model is linear in the unknown parameters [18], [24]. Thus, the d_i uniquely specify S. However, these range measurements are noisy in real life applications, causing the localization algorithm to arrive at an inaccurate result.

Another approach in the literature is the use of weighted least squares (WLS)-based estimation [22]. This approach functions by utilizing a non-convex cost function, requiring some conditions to prevent local minima. Another similar approach is maximum likelihood (ML) [25] estimator design, assuming the noise measurements are Gaussian. The cost function used in the ML design [25] also has stable local minima. Hence, this design has issues with convergence to a global minimum.

There are a number of attempts in the literature to eliminate the local minimum issue [6, 11, 18–23, 26]. One such recent attempt is the projection onto convex sets (POCS) algorithm proposed in [19]. The POCS algorithm is effective when the sensor node to be

localized resides within the convex hull of the anchors used for localizing it. If the sensor to be localized is outside the convex hull, the algorithm is observed to result in a limit cycle, not reaching the desired target sensor location. POCS was improved by Rydstorm *et al.*, [20], where the improved algorithm was called hyperbolic POCS. However, hyperbolic POCS does not always converge as well, and has unspecified convergence conditions [20].

As observed in the studies mentioned above, the cost functions used in formulation of range measurement-based localization problems are naturally non-convex, and have a high chance to be influenced by local minima. Even the presence of a large number of distance measurements in sensor/anchor deployment may not be sufficient for obtaining accurate results. Further, the convergence characteristics of any non-linear algorithm applied must be carefully considered. From another aspect, it is also essential to build regions enclosing fewer numbers of sensors/anchors to prevent wasting energy and to minimize delays. If a source/sensor is known to reside in one such region, a cost function that is convex in this region can be minimized by applying a gradient descent algorithm, resulting in exponential convergence in the noise-free case for the specific region. Such a regional partitioning eases practical localization in three aspects: ensuring accuracy in noisy cases, providing practical advantages such as lower computational and communication cost, and preventing re-initialization [21].

In Section 3.3, we revisit Problem 1.2.1 and the approach of [21], proposing a technique that employs a geometric feature to build a convex formulation of the localization problem. Once the convex formulation is built, numerous convex minimization methods can be used, including the gradient-based localization algorithm chosen for this study.

2.2 Soft Computing-Based Approaches

Computational intelligence (CI) is an area of study about mechanisms having the ability to learn, to adapt to changes in environments, and to make human-like inferences. Many soft computing methods such as neural networks, reinforcement learning, swarm intelligence, evolutionary algorithms, fuzzy logic, and genetic algorithms, have surpassed conventional ones under conditions of uncertainty. Furthermore, compared to conventional approaches, soft computing-based approaches are better able to handle situations with multiple restrictions, such as in power supply, communication bandwidth, and computational capabilities. [27]

Classical set theory, preferred in the conventional approaches, allows elements to be either in a set or not, which prevents the system from using human-like reasoning, such as dealing with uncertainty and imprecision. This reasoning can be modelled using fuzzy logic, allowing elements to have partial membership degrees in a set [27]. The membership degree of an input in a certain range is the degree of truth for the input ranging between 0 and 1.

Fuzzy logic-based system involves the three main steps demonstrated in Figure 2.1: fuzzification, rule-based inferencing, and defuzzification. In the fuzzification step, the sytem takes crisp, i.e. non-fuzzy, input values, and converts them into fuzzy values using membership functions. These functions can be in different forms, such as triangular, trapezoidal, Gaussian, etc. The membership functions result in membership degrees of the input. These membership degrees are combined based on some IF-THEN rules in the fuzzy inference engine, and then are sent into the defuzzifier. The inference methods include the max-min method, the averaging method, the root sum squared method, and the clipped center of gravity method [28]. In the defuzzification step, opposite to fuzzification, the

fuzzy values obtained in the inference engine are converted into crisp values.

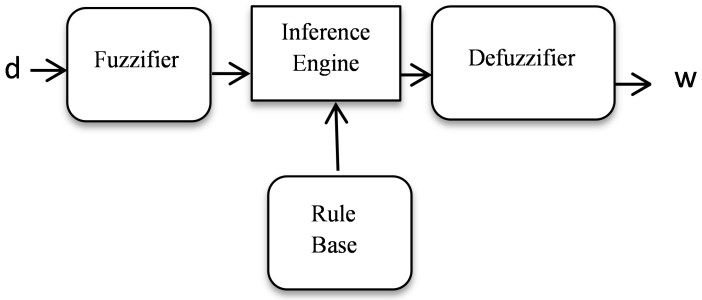

Figure 2.1: *A typical fuzzy inference system with crisp input d and crisp output w.*

Since fuzzy logic-based methods mimic human decision-making by giving degrees of truth, such methods are outstanding preferences in several applications. These applications include air conditioning systems, heating systems, home appliances, power systems, industrial automation, digital image processing, and pattern recognition [27].

Fuzzy logic-based approaches are also chosen in many localization problems. Ability to give detailed location data is one of the main reasons to use fuzzy logic in such problems. Many studies, such as [29], use fuzzy sets to express different kinds of uncertainty in location data.

Furthermore, fuzzy logic is complimented by Dharne et al., [30] for avoiding the use of strict probabilistic approximation methods or special hardware in the localization of mobile robots. They also mention that a fuzzified system helps to lessen computational resources, to be more robust to inaccurate sensor location estimates, and to ease heuristic rule determination. Fuzzy logic also assists in accomplishing multiple simultaneous tasks for target localization applications. One of the studies regarding multiple simultaneous tasks, developed by Rowaihy et al., [31], uses exact position information and fuzzy logic in order to employ various types of different directed sensor nodes.

To have more optimized and self-learning systems, hybrid fuzzy techniques consisting of fuzzy logic and another soft computing method have been employed by several researchers [32, 33]. One of these studies employs genetic-fuzzy and neuro-fuzzy algorithms to handle the shortcomings of the centroid localization [32]. The study aims to achieve more accurate results and error resilience by using several kinds of membership functions. [33]

On the other hand, there have been various studies [34–37] involving fuzzy logic and area-based localization methods. Area-based range-free localization methods are preferred, because they can handle larger environments by dividing the environment into a number of overlapping sub-regions. The absence or presence of a sensor node in such regions guides the algorithm to find the potential sub-region where the sensor node is located. He *et al.* [37], introduce a range-free algorithm in which a sensor node is tested as to being inside or outside the triangle defined by three anchors, and the diameter of the approximate area can be narrowed using the locations of anchors to obtain better position approximation.

Liu *et al.* [34] describe the ROCRSSI (Ring-overlapping based on comparison of received signal strength indicator) algorithm, in which intersecting overlapping rings constrain the location of a sensor node. Such rings take all anchors as centre points and use signal strength between anchors to form the radii of these centres. The signal strength which the sensor node receives from anchors are also employed to localize the sensor. However, there are some issues in the proposed method which prevent the algorithm functioning properly [36]. The first is choosing a false ring, which is unfolded by selecting the valid region with the maximum number of intersecting rings. Another is large localization errors caused by small radio irregularities in the vicinity of ring boundaries.

In order to eliminate the side effects of inaccurate RSSI measurements which resulted in high cost, a method based on the Takagi-Sugeno-Kang (TSK) fuzzy model is proposed in [35]. This method employs fuzzy logic to transform RSSI into edge weights of adjacent

anchors used for localization by a weighted centroid localization approach.

As a remedy for ROCRSSI, the fuzzy ring overlapping range-free (FRORF) localization technique was proposed by Velimirovic et. al, [1]. The FRORF algorithm represents localization areas with overlapping rings, similar to ROCRSSI. Rings are formed with anchors as ring centres, based on RSS measurements between the anchor at the centre and other anchors. Furthermore, FRORF utilizes fuzzy logic to handle uncertain RSSI measurements which can result in significant localization errors. Overlapping rings are represented as fuzzy sets, and membership degrees of a sensor node are obtained from membership functions based on RSSI measurements. To find the estimate of the sensor, the weighted average center of gravity of localization areas with these membership degrees is calculated. (see Figure 2.2 for general working scheme of FRORF)

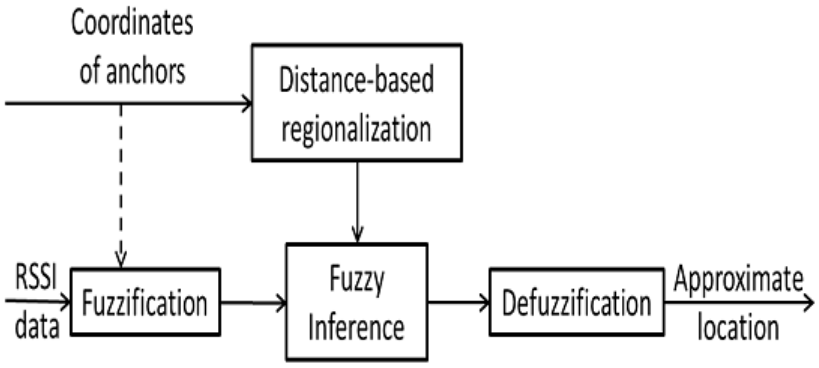

Figure 2.2: *Working scheme overview of FRORF [1].*

Although FRORF overcomes the uncertainty in RSSI measurements, there is another issue: non-convexity. This issue is a result of local minima at the intersection of overlapping rings. To avoid non-convexity, we propose a new range-free method, using overlapping lines instead of rings to form localization areas. Absolute distance measurements between anchor pairs, and a geometric feature, called radical axis, are used to build line intervals.

Membership degrees and location estimation of a sensor node are determined similar to FRORF, and will be explained in details in Chapter 3.

2.3 Absolute Range-Based vs Range-Free Algorithms

Wireless sensor network localization algorithms are classified based on various criteria [10]. One such categorization, which has motivated this study is absolute range-based vs. range-free localization.

Absolute range-based techniques employ absolute distance or angle measurements between anchors and sensor nodes. These measurements are used to determine the location of the sensor nodes. Range-based algorithms utilize inter-sensor measurement techniques such as received signal strength (RSS), time of arrival (TOA), time difference of arrival (TDoA), or angle of arrival (AoA) [6,10,11]. Information on coordinates of anchors is populated by using a GPS (for outdoor application), or an accurate indoor positioning system (for indoor applications), or via direct assigments during set-up, before the localization process.

Range-based methods can output more accurate location estimates with the help of additional hardware equipment. However, they bear higher cost and power consumption, and generally require higher traffic in network communication [10], as well as being more sensitive to measurement noises. Furthermore, certain WSNs might not have hardware devices needed to implement absolute range-based algorithms. Hence, range-free methods are preferred in many cases, because they use only standard features found in most radio modules, and are thus more economical and simpler than range-based ones. Although most range-free methods yield low system costs, their localization results are not as accurate as

those of range-based ones [1]. Also, the proximity calculation can be easily affected by the environment and thus fail.

Range-free methods utilize approximate distances, in contrast to range-based techniques which use absolute estimates of distances. Such distance estimates include means of anchor proximity, near-far information, and comparison of less accurate distance estimation, to determine which sensor node is in the transmission range of which anchor. This determination guides the system to produce approximate distance information for the sensor nodes. There is no single perfect algorithm which works very well for all localization problems. Thus both absolute range-based and range-free algorithms are used in practice, a choice depending on the specific application and requirements [38]. Nevertheless, the focus of this work will be settings with inaccurate measurements, where range-free algorithms are more appropriate to use.

In this study, to avoid non-convexity, we propose a new range-free method using certain overlapping rectangles instead of rings. Absolute distance measurements between anchor pairs, and a geometric feature called radical axis, are used to build these regions. We develop our methodology combining the fuzzy framework of (FRORF) and a new convex geometric methodology proposed in [13].

2.4 Summary

This chapter gives insights into the field of study of convex optimization and soft computing-based approaches. In order for the reader to help assess the two approaches, several algorithms in both fields of study are mentioned, with pros and cons. Finally, absolute range-based and range-free algorithms are compared, to understand further the motivation behind the algorithm proposed in the following chapter (in Section 3.4).

The next chapter introduces a new fuzzy logic based range-free localization algorithm: the general structure and main components of our algorithm are introduced in Section 3.1, the main components are further elaborated in Sections 3.2 and 3.3, and our algorithm design based on these main components is presented in Section 3.4.

Chapter 3

Proposed Fuzzy Logic Based Range-Free Localization Algorithm

In this chapter, we propose a new fuzzy logic-based range-free localization algorithm, combining the fuzzy approach of ring-overlapping proposed in [1], and a new convex geometric methodology proposed by us in [13]. The general structure and main components of our algorithm are introduced in Section 3.1. The main components are further elaborated in Sections 3.2 and 3.3. Our algorithm design based on these main components is presented in Section 3.4. Concluding remarks of the chapter are given in Section 3.5.

3.1 General Structure and Main Components

Our motivation for designing a new fuzzy range-free localization algorithm is the potential non-convexity issue (multiple stable local minima) in similar algorithms in the literature, particularly a recently proposed algorithm, FRORF [1]. This issue stems from

the non-convex nature of range measurement: the set of points having a fixed distance x from a sensor S is non-convex, and is a circle if we consider a planar setting.

The fuzzy logic-based studies in the literature, including FRORF [1], consider fuzzy sets whose definition is based on such natural non-convex sets. Instead of fixed distances, ranges at distances are used, leading to rings, which are also non-convex.

To overcome the convexity issue we propose using the notions of radical axes and radical axis powers. As detailed in Section 3.3, these notions help to compose convex fuzzy regions, still based on distance information. Our proposed algorithm consists of two main components:

(i) A fuzzy logic setting similar to that of FRORF

(ii) A new methodology for generating convex fuzzy sets based on the radical axis property.

In the following two sections, these components are explained in details.

3.2 The Fuzzy Ring-Overlapping Range-Free Localization Algorithm

The idea in FRORF as proposed by Velimirovic et al. [1] is built using an area-based range-free localization approach. The authors aim to partition the region containing a number of nodes into a number of fuzzy regions. The fuzzy regions are in the form of intersections of overlapping rings that are represented as fuzzy sets with ambiguous boundaries. The working scheme of the proposed algorithm for Problem 1.2.1 in [1] is as follows.

Consider an anchor sensor A_i and the estimate distance measurement x of the sensor node S from the anchor A_i. Order the other anchors A_{k_j} $(k_j \neq i, j = 1, \ldots, N-1)$ according to their distances $d_{ij} = |A_{k_j} A_i|$ from A_i in the Euclidean domain, such that $d_{ij} \leq d_{i(j+1)}$ for all j. Define also $d_{i,0} = 0$ and $d_{i,N} = \infty$. Based on this ordering, define the ring intervals

$$RI_j^i = \{S | \alpha_j \leq |A_i S| \leq \beta_j\} := RI(A_i)[\alpha_j, \beta_j] \tag{3.2.1}$$

where $\alpha_j^i = d_{ij}$ and $\beta_j^i = d_{i(j+1)}$ for $j = 1, \ldots, N-1$. As illustrated in Figure 3.1, ring intervals for A_i are derived from (3.2.1) according to ordering of the distances from A_i.

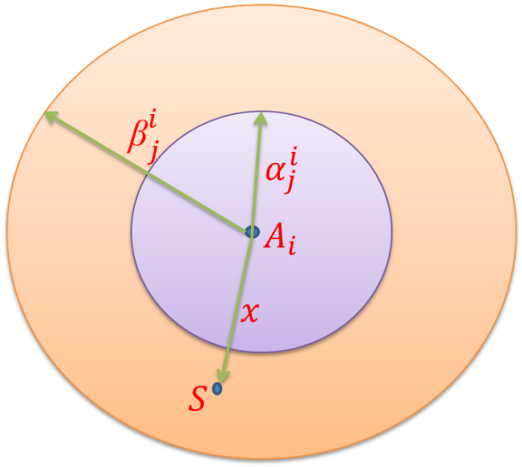

Figure 3.1: *Representation of a ring interval formed according to an anchor A_i proposed in FRORF [1].*

Each such ring interval RI_j (dropping the superscript i for brevity) can be considered as the intersection

$$RI_j = LT_j \cap GT_j \tag{3.2.2}$$

of the region $LT_j = RI(A_i)[0, \beta_j]$, which is the area inside the circle $C(A_i, \beta_j)$ [1] and the region $GT_j = RI(A_i)[\alpha_j, \infty)$, which is the area outside the circle $C(A_i, \alpha_j)$.

[1]Given $A \in \Re^2$ and $r \in \Re^+$, $C(A, r)$ denotes the circle with center A and radius r.

For the sensor S with crisp value x, representing the distance estimate to A_i, LT_j and GT_j crisp intervals are represented by fuzzy sets $\widetilde{LT_j} = \{(x, \mu_{LT_j}(x))|x \in R^+\}$ and $\widetilde{GT_j} = \{(x, \mu_{GT_j}(x))|x \in R^+\}$. Here, $\mu_{LT_j}(x)$ and $\mu_{GT_j}(x)$ are the membership functions illustrated in Figure 3.2, and are formally defined as follows for a pre-defined fuzzification parameter $p \in [0, 1]$:

$$\mu_{LT_j}(x) = \begin{cases} 1 & \text{if} \quad x \leq \beta_j(1-p) \\ \frac{(1+p)\beta_j - x}{2p\beta_j} & \text{if} \quad \beta_j(1-p) < x < \beta_j(1+p) \\ 0 & \text{if} \quad x \geq \beta_j(1+p) \end{cases} \tag{3.2.3}$$

$$\mu_{GT_j}(x) = \begin{cases} 0 & \text{if} \quad x \leq \alpha_j(1-p) \\ \frac{x-(1-p)\alpha_j}{2p\alpha_j} & \text{if} \quad \alpha_j(1-p) < x < \alpha_j(1+p) \\ 1 & \text{if} \quad x \geq \alpha_j(1+p) \end{cases} \tag{3.2.4}$$

$$\mu_{RI_j}(x) = \mu_{LT_j}(x) + \mu_{GT_j}(x) - 1 \tag{3.2.5}$$

The membership functions (3.2.3) - (3.2.4) follow fuzzy intersection rules in fuzzy set theory [28, Chapter 2]. Here, the fuzzification parameter $p \in [0, 1]$ enables the system or the user to decide on the width of the region in the neighbourhood of ring boundaries corresponding to α_j and β_j.

For fuzzification, two different approaches were proposed in [1]: direct-RSS fuzzification, and indirect-RSS fuzzification. In this study, indirect-RSS fuzzification is preferred for comparison purposes, since the algorithm we propose uses known distances between anchors. In the indirect-RSS fuzzification, after defining fuzzy rings in the Euclidean domain,

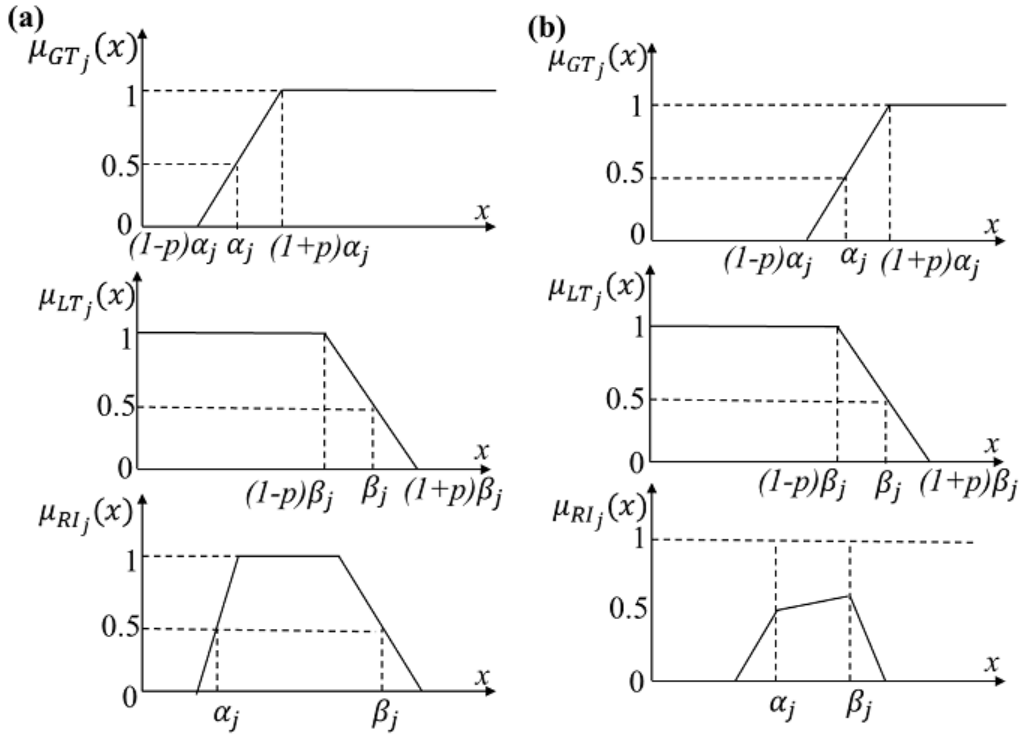

Figure 3.2: *Membership functions for a fuzzy ring: (a) non-overlapping fuzzy regions of μ_{LT_j} and μ_{GT_j} ; (b) overlapping fuzzy regions of μ_{LT_j} and μ_{GT_j} [1].*

RSS measurements of sensor nodes are converted into distances, and final membership degree of a sensor node S is taken as the average of membership degrees of S for distance estimation by each anchor A_i. Details of the RSS measurement mechanism and distance estimate generation are beyond the scope of this thesis. Hence, we assume availability of the estimated distance x.

Next step is to build fuzzy inference and fuzzy sets based on the fuzzy intervals and membership functions as defined for each anchor A_i. The fuzzy ring set corresponding to

anchor A_i is defined as

$$\widetilde{RS_i} = \{(k_j, \mu_{RI_j^i}) | \mu_{RI_j^i} > 0, k_j \neq i, k_j \in \{1, ..., N\}\}. \tag{3.2.6}$$

The overall fuzzy regional map $\widetilde{RM}$ is constructed by taking the Cartesian product over the fuzzy ring sets of the N anchors:

$$\widetilde{RM} = \widetilde{RS_1} \times \widetilde{RS_2} \times \ldots \times \widetilde{RS_N} = \{(c, \mu_{RM}(c))\} \tag{3.2.7}$$

where

$$\mu_{RM}(c) = \prod_{m=1}^{N} \mu_{RI_{jm}^m} \tag{3.2.8}$$

is the membership degree of the region with area code

$$c = (k_{j_1}, k_{j_2}, ..., k_{j_N}) \tag{3.2.9}$$

denoting the intersection region $\cap_{m=1}^{N} RI_{jm}^m$. The feasible area code set is defined as

$$\widetilde{R} = \{c | (c, \mu_R(c)) \in \widetilde{RM}\} \tag{3.2.10}$$

i.e., the set corresponding to all the non-empty fuzzy intersection regions.

As an example of the region area coding, examine the system of three anchors and one sensor node, depicted in Figure 3.3. Let the sensor node be located in the shaded area. Considering the anchor A_1, this region is inside the fuzzy ring RI_2^1. From the aspects of A_2 and A_3 anchors, respectively, it is inside RI_2^2 and RI_3^3. Hence the corresponding area code is $c = (2, 2, 3)$.

The last step, defuzzification, determines the location estimate according to the membership degrees of sensor node to regions $\widetilde{RM}$. Each area code in $\widetilde{R}$ is associated with a center of gravity (CoG) of the corresponding region and a membership degree in $\widetilde{R}$.

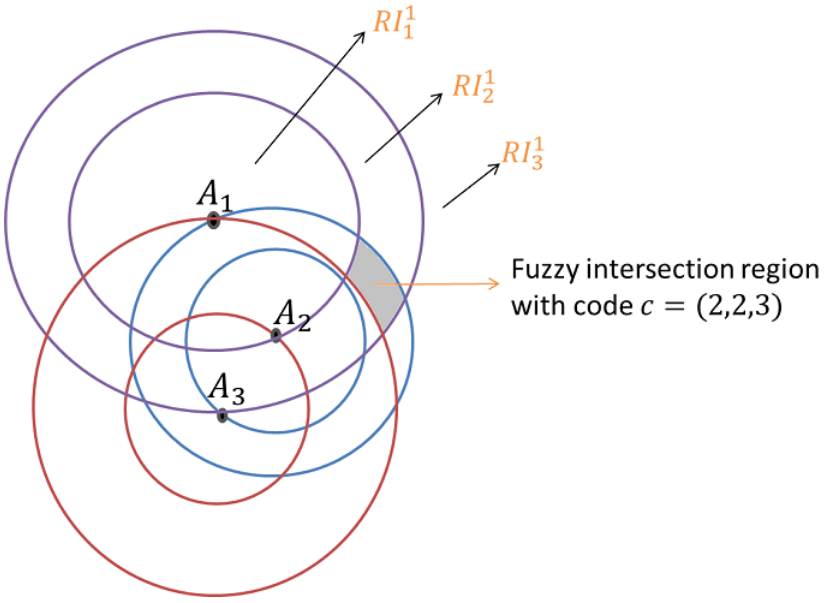

Figure 3.3: *An instance of finding area code for* 3*-Anchors in FRORF [1].*

The final location estimation formula is as follows:

$$\hat{S} = \textit{Estimated location} = \frac{\sum_{c\in\widetilde{R}} g_c(c)\mu_{RM}(c)}{\sum_{c\in\widetilde{R}} \mu_{RM}(c)} \tag{3.2.11}$$

where $g_c(c)$ is the CoG of the region with area code c.

In most localization problems, non-convexity is a common issue. This issue is also encountered in FRORF, since FRORF employs ring intervals to represent fuzzy sets. To work with the fuzzy logic-based framework of [1] more efficiently, there is a need for an alternative approach which can handle the non-convexity problem. Since ring intersections lead to non-convex regions, a different geometric technique is proposed in the next section to construct convex regions based on distance measurements.

3.3 Preliminaries of the Proposed Algorithm: Constructing Convex Fuzzy Regions

In this section, we present the geometric preliminaries of the algorithm we proposed in Section 3.4. The aim is to construct an alternative set of convex regions in place of the non-convex rings used by FRORF. In a slightly different context, in [13], a geometric methodology is proposed to construct such convex regions. The methodology is based on a new cost convex function with the following Euclidean geometric property:

Theorem 3.3.1 *[39, Fact 45] Given two non-concentric circles with centres C_1, C_2 and radii r_1, r_2, respectively, there is a unique line consisting of points S holding equal powers with regard to these circles, i.e., satisfying*

$$\|S - C_1\|^2 - r_1^2 = \|S - C_2\|^2 - r_2^2 \tag{3.3.1}$$

This line is perpendicular to the line connecting C_1 and C_2, and if the two circles intersect, passes through the intersection points.

The unique line mentioned in Theorem 3.3.1 is called the *radical axis* of $C(C_1, r_1)$ and $C(C_2, r_2)$ [39]. In this work, we call the difference

$$\|S - C_1\|^2 - \|S - C_2\|^2 = r_1^2 - r_2^2 \tag{3.3.2}$$

the radical axis power of S with respect to the pair (C_1, C_2) and denoted by $P(S, (C_1, C_2))$.

Lemma 3.3.1 *In 2 dimensions, if the distance estimates/measurements d_i in Problem 1.2.1 are noise-free, the intersection of the radical axes of any $N-1$ distinct circle pairs $C(x_i, d_i)$, $C(x_j, d_j)$ $(i \neq j)$ is S.*

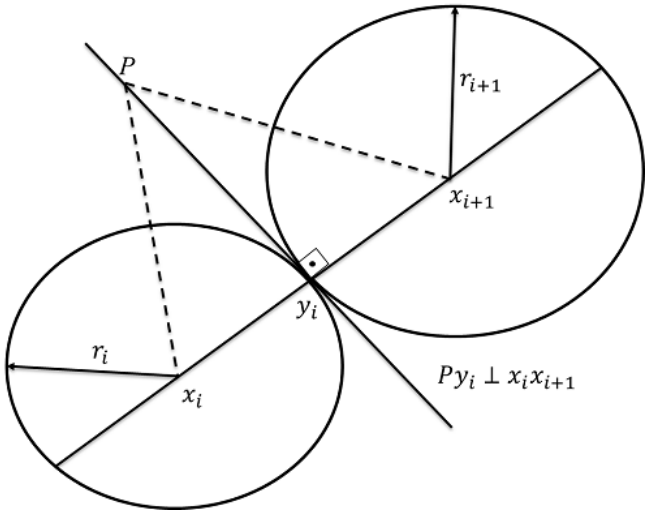

Figure 3.4: *The case of two circles, intersection utilizing the radical axis property.*

Proof 3.3.1 *The result straightforwardly follows from the Problem 1.2.1 definition and the last statement of Theorem 3.3.1.*

See the mathematical representation of the radical axis of a circle pair $C(x_i, d_i)$, $C(x_j, d_j)$ $(i \neq j)$ given the values of x_i, x_j, d_i, and d_j shown in Figure 3.4 for the case of circle intersection. Here, j is taken to be $i + 1$.

Using Theorem 3.3.1 for a sensor network with N nodes located at $x_1, \ldots, x_N$, one can consider the $N - 1$ node-distance measurement pairs $(x_i, d_i), (x_{i+1}, d_{i+1})$ and the corresponding radical axes line l_i perpendicularly intersecting $x_i x_{i+1}$ at y_i (refer to Figure 3.5). Hence any point y on l_i satisfies

$$(y - y_i)^T e_i = 0, \forall i \in \{1, ..., N - 1\} \tag{3.3.3}$$

where

$$e_i = x_{i+1} - x_i$$

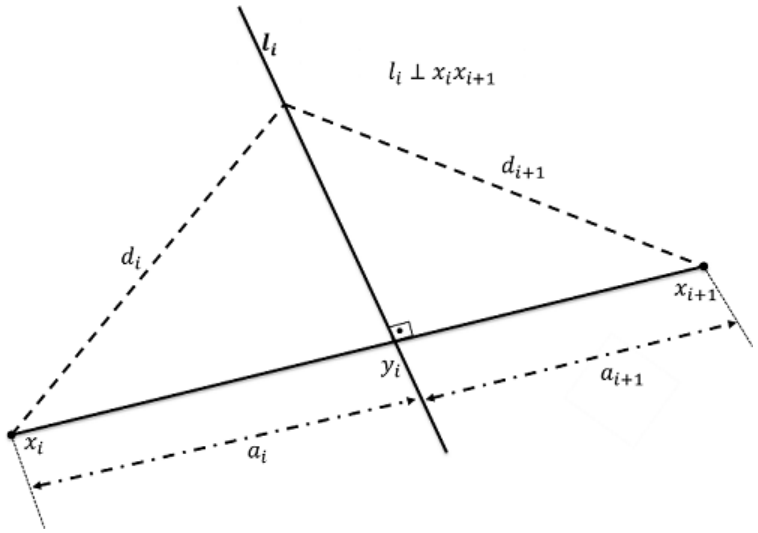

Figure 3.5: *Representation of node-distance measurement pairs* (x_i, d_i), (x_{i+1}, d_{i+1}) *and the corresponding radical axis* l_i.

y_i can be computed as

$$y_i = x_i + a_i \frac{e_i}{\|e_i\|}, \tag{3.3.4}$$

as well as ${d_i}^2 - {a_i}^2 = {d_{i+1}}^2 - {a_{i+1}}^2 = {d_{i+1}}^2 - (\|e_i\| - a_i)^2$, from which a_i can be calculated as

$$a_i = \frac{\|e_i\|^2 + {d_i}^2 - d_{i+1}^2}{2\|e_i\|}. \tag{3.3.5}$$

Note that sequential pairs of nodes are chosen in this study. Next, we use Lemma 3.3.1 to build a convex alternative for the cost function proposed in [13]. Therefore, Lemma 3.3.1 implies that the intersection of $l_1, \ldots, l_{N-1}$ is S, i.e., S is the unique point satisfying (3.3.3). As a result, S is the unique solution of the equation $J(y) = 0$, where

$$J(y) = \frac{1}{2} \sum_{i=1}^{N-1} \left((y - y_i)^T e_i\right)^2. \tag{3.3.6}$$

Note that (3.3.6) is a convex cost function.

The assessments above bring about the following results: in a two-dimensional system, if the distance measurements d_i in Problem 1.2.1 are noise-free, then S satisfies the following:

- S is the intersection point of the radical axis lines $l_1, \dots l_{N-1}$ defined by (3.3.3) for $i = 1, \dots, N-1$, respectively.
- S is the unique local minimum, and consequently the global minimum, of (3.3.6).

As mentioned in Section 2.1, there have been various algorithms to compute S in Problem 1.2.1, which can be formed based on the convex cost function (3.3.6). Here, to enable easy analysis and comparison with [21], a gradient-based algorithm with the standard iterations of estimate updates in the negative gradient direction is proposed.

$$-\nabla J(y) = -\left(\frac{\partial J(y)}{\partial y}\right) = -\sum_{i=1}^{N-1}\left[(y-y_i)^T e_i\right] e_i. \tag{3.3.7}$$

The corresponding gradient-based localization algorithm is given by

$$y\,[k+1] = y\,[k] - \mu \nabla J(y[k]), \tag{3.3.8}$$

where μ is a small positive design coefficient.

Note that $\nabla J(y) = 0$ if and only if $y = S$ as a result of Lemma 3.3.1. Thus, once the algorithm (3.3.8) reaches point $y[k] = S$, it becomes settled at that point. There are several analyses of how to choose the gradient gain μ, and its effects on the convergence of the gradient descent algorithm (3.3.8) in the convex optimization literature [11, 15, 23, 25]. Here, the main convergence result for (3.3.8) with constant gain $\mu > 0$ is summarized as follows:

Theorem 3.3.2 *Consider Problem 1.2.1 in 2 dimensions, with noise-free distance measurements d_i. Assume that the x_i, $i \in \{1, \dots, N\}$ are non-collinear. Then for every $M > 0$,*

there exists a $\mu^*(M)$ *such that*

$$\lim_{k\to\infty} y[k] = S. \tag{3.3.9}$$

whenever

$$J(y[0]) \leq M \tag{3.3.10}$$

and

$$0 < \mu \leq \mu^*(M). \tag{3.3.11}$$

that for any arbitrary choice of $y[0]$*, there exists a sufficiently small* μ *for the algorithm* (3.3.8) *such that* (3.3.9) *is satisfied.*

Proof 3.3.2 *Consider an arbitrary scalar* $M > 0$ *and the corresponding convex set* $S_J(M) = \{y : \ J(y) \leq M\}$*. Since* (3.3.6) *is a quadratic and hence differentiable function of* y*, it is Lipschitz within* $S_J(M)$*, i.e., there exist* $L(M)$ *such that for any* $\bar{y}_1, \bar{y}_2 \in S_J(M)$*,* $\|\nabla J(\bar{y}_1) - \|\nabla J(\bar{y}_2)\| \leq L(M)$*. Hence, Proposition 1.2.3 of [14] (or Theorem 1 on pp. 21 of [16]) implies that, for* $y[0] \in S_M$

$$0 < \mu < \mu^*(M) = 2/L(M), \tag{3.3.12}$$

we have (i) $J(y[k+1]) \leq J(y[k])$*, and hence* $y[k] \in S_J(M)$ *for* $k = 0, 1, \ldots$*; and (ii)* $\lim_{k\to\infty} \nabla J(y[k]) = 0$*, which, by Lemma 3.3.1 is equivalent to* (3.3.9)*.*

An example for the effect of the radical axis on the convergence is given in Figure 3.6.

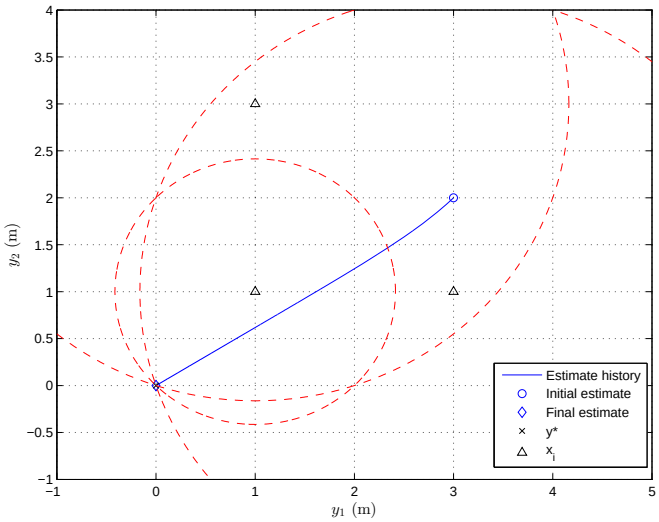

Figure 3.6: *Radical axis convergence test illustration of* 3 *anchors in a* $2-$*dimensional sytem.*

3.4 The Algorithm

RAFRF is built in a similar manner to FRORF; however, it employs the radical axis property mentioned in Theorem 3.3.1 to form the fuzzy sets, and to avoid potential local minima. These fuzzy sets are represented with the intersection of overlapping rectangular regions for each anchor pair in the sequel. The working scheme of RAFRF algorithm for Problem 1.2.1 is as follows.

Consider an anchor pair (A_i, A_j) in the sequel and estimate distance measurement x of the sensor node S with respect to a power value of the anchor pair (A_i, A_j). For a given sensor node S, the radical axis power (defined in (3.3.2)) to anchor pair (A_i, A_j) is denoted by $P(S, (A_i, A_j))$.

Order all anchors $A_1, \ldots, A_N$ according to radical axis power

$$P_{(i,j)}(A_k) = P(A_k, (A_i, A_j)), \forall i, j \in \{1, \ldots, N\}. \tag{3.4.1}$$

Define the permutation set $I_{(i,j)} = \{\rho_k^{(i,j)}\}$ of $\{1, \ldots, N\}$ based on this ordering, so that

$$P_{(i,j)}(A_{\rho_k^{(i,j)}}) \leq P_{(i,j)}(A_{\rho_{k+1}^{(i,j)}}) \text{ for all } k. \tag{3.4.2}$$

With respect to the anchor ordering, describe the *power line intervals* as

$$PI_k^{(i,j)} = \{S | \alpha_k \leq P_{(i,j)}(S) \leq \beta_k\}, \text{ for } k = 0, 1, \ldots, N \tag{3.4.3}$$

where $\alpha_0^{(i,j)} = -\infty$ and $\beta_N^{(i,j)} = \infty$. As demonstrated in Figure 3.7, power line intervals are obtained from 3.4.3 based on the ordering. Also, define

$$\alpha_k^{(i,j)} = \beta_{k-1}^{(i,j)} = P_{(i,j)}(A_{\rho_k^{(i,j)}}). \tag{3.4.4}$$

Each such line interval PI_k (dropping the superscript (i,j) for brevity) can be considered as the intersection

$$PI_k = LT_k \cap GT_k \tag{3.4.5}$$

of the region $LT_k = PI_k(A_k)(-\infty, \beta_k]$ inside the power line interval with maximum radical axis power β_k, and $GT_k = PI_k(A_k)[\alpha_k, \infty)$, outside the power line interval with minimum radical axis power α_k.

For the sensor S with crisp value x, representing the radical axis power estimate with respect to A_iA_j, LT_k and GT_k crisp intervals are represented by fuzzy sets $\widetilde{LT_k} = \{(x, \mu_{LT_k}(x)) | x \in R^+\}$ and $\widetilde{GT_k} = \{(x, \mu_{GT_k}(x)) | x \in R^+\}$. Here, $\mu_{LT_k}(x)$ and $\mu_{GT_k}(x)$ are membership functions formally defined as follows for a pre-defined fuzzification parameter $p \in [0, 1]$:

$$\mu_{LT_k}(x) = \begin{cases} 1 & \text{if} \quad x \leq sign(\beta_k)(1-p)\beta_k \\ \frac{(1+p)\beta_k - x}{2p\beta_k} & \text{if} \quad sign(\beta_k)(1-p)\beta_k < x < sign(\beta_k)(1+p)\beta_k \\ 0 & \text{if} \quad x \geq sign(\beta_k)(1+p)\beta_k \end{cases} \tag{3.4.6}$$

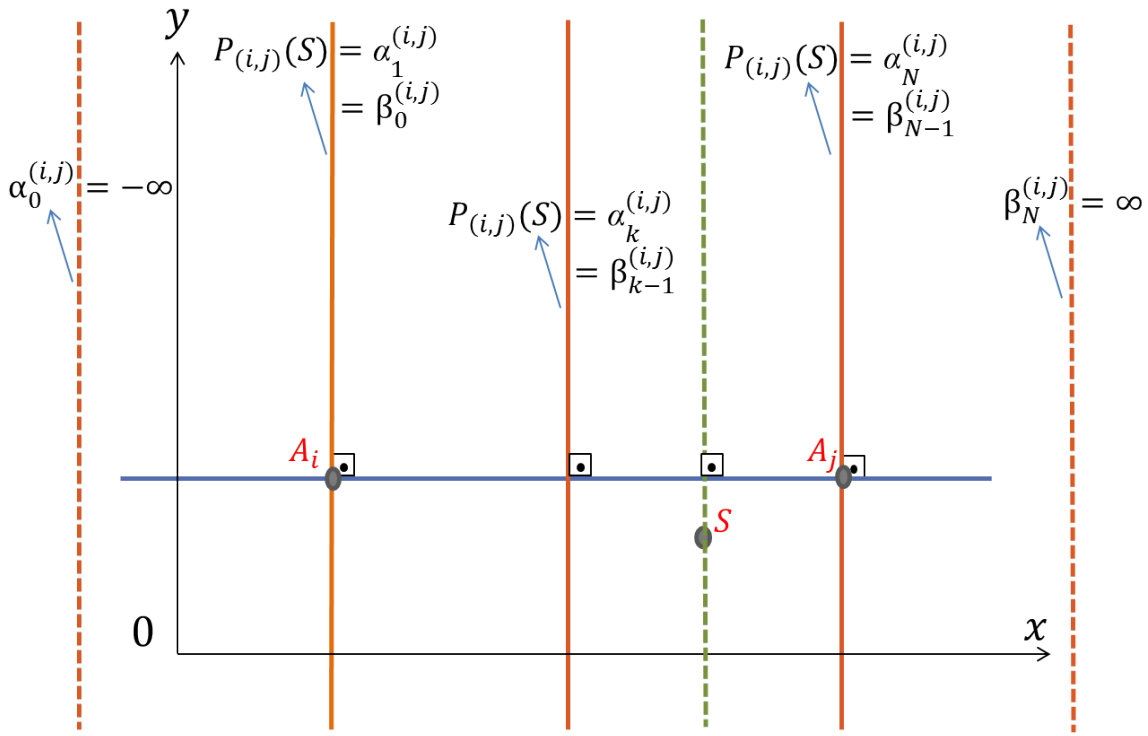

Figure 3.7: *Generic representation of power line intervals formed based on ordering of radical axis power of sensor S to anchor pairs.*

$$\mu_{GT_k}(x) = \begin{cases} 0 & \text{if} \quad x \leq sign(\alpha_k)(1-p)\alpha_k \\ \frac{x-(1-p)\alpha_k}{2p\alpha_k} & \text{if} \quad sign(\alpha_k)(1-p)\alpha_k < x < sign(\alpha_k)(1+p)\alpha_k \\ 1 & \text{if} \quad x \geq sign(\alpha_k)(1+p)\alpha_k \end{cases} \tag{3.4.7}$$

$$\mu_{PI_k}(x) = \mu_{LT_k}(x) + \mu_{GT_k}(x) - 1. \tag{3.4.8}$$

Similar to FRORF, the fuzzification parameter p enables the system or the user to decide on the width of the region in the neighbourhood of the power line boundaries corresponding

to α_k and β_k.

For fuzzification, indirect-RSS fuzzification is also preferred, and we assume availability of distance estimate x. Note that the number of power line intervals corresponding to each anchor pair A_iA_j is equal to $N+1$ for $k=1,\ldots,N$. Also, the number of anchor pairs A_iA_j for $i,j=1,\ldots,N$ in the network is given as

$$M=\binom{N}{2}. \tag{3.4.9}$$

Next step is to build fuzzy inference and fuzzy sets based on the fuzzy intervals and membership functions defined for each anchor pair (A_iA_j). The fuzzy set for (A_iA_j) is defined as

$$\widetilde{PS}_{(i,j)}=\{(k,\mu_{PI_k^{(i,j)}})|\mu_{PI_k^{(i,j)}}>0,k\in\{0,1,\ldots,N\}\}. \tag{3.4.10}$$

Overall fuzzy regional map $\widetilde{PM}$ is constructed by taking the Cartesian product over the fuzzy sets of the M anchor pairs:

$$\widetilde{PM}=\widetilde{PS}_{(1,2)}\times\ldots\widetilde{PS}_{(1,N)}\times\widetilde{PS}_{(1,N)}\times\ldots\times\widetilde{PS}_{(N-1,N)}=\{(c,\mu_{PM}(c))\} \tag{3.4.11}$$

where

$$\mu_{PM}(c)=\prod_{i=1}^{N-1}\prod_{j=i+1}^{N}\mu_{PI_{k_{(i,j)}}^{(i,j)}} \tag{3.4.12}$$

is the membership degree of the region with area code

$$c=(k_{(1,2)},k_{(1,3)},\ldots,k_{(N-1,N)}) \tag{3.4.13}$$

denoting the intersection region of $PI_{k_{(1,2)}}^{(1,2)},PI_{k_{(1,3)}}^{(1,3)},\ldots,PI_{k_{(N-1,N)}}^{(N-1,N)}$. The feasible area code set is defined as

$$\widetilde{P}=\{c|(c,\mu_{PM}(c))\in\widetilde{PM}\} \tag{3.4.14}$$

i.e., the set corresponding to all the non-empty fuzzy intersection regions. Consequently, the total (maximum) number of intersection (fuzzy) regions is equal to

$$n_F = (N+1)^M. \tag{3.4.15}$$

As an example of the region area coding, examine the system of three anchors and one sensor node, depicted in Figure 3.8. Let the sensor node be located in the shaded area. Considering the power line of anchor pair (A_1, A_2), this region is inside the fuzzy line $PI_2^{1,2}$. From the aspects of (A_1, A_3) and (A_2, A_3) anchor pairs, respectively, it is inside $PI_3^{1,3}$ and $PI_3^{2,3}$. Thus, the corresponding area code is $c = (2, 3, 3)$.

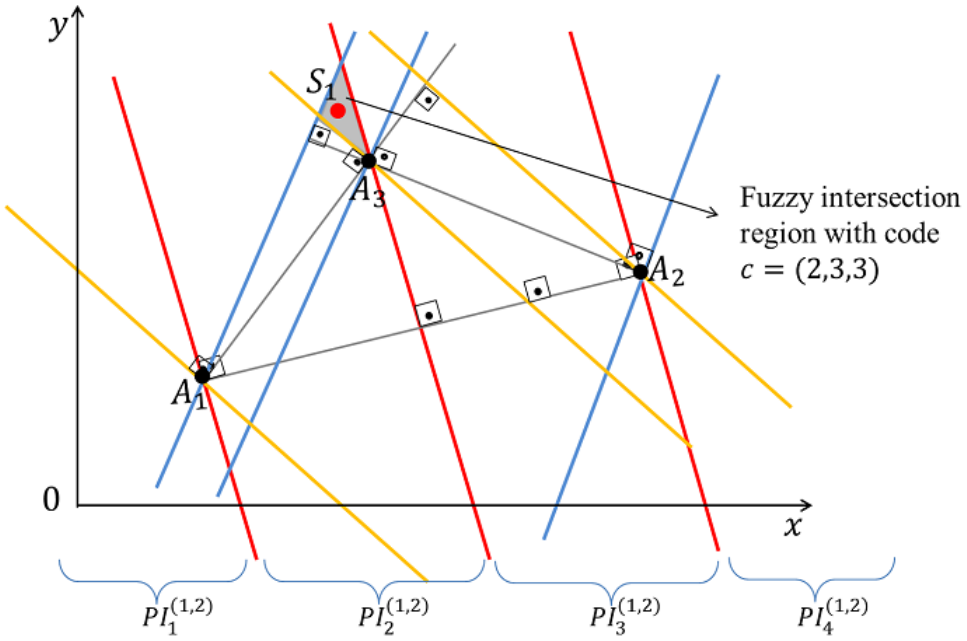

Figure 3.8: *An instance of finding area code for* 3 *anchors in RAFRF.*

In the defuzzification step, similiar to FRORF, location is estimated using membership degrees of the sensor node to regions in $\widetilde{PM}$, and the corresponding center of gravity (CoG)

value associated with those regions. Thus, the final location estimation formula is

$$\hat{S} = \textit{Estimated location} = \frac{\sum_{c\in\widetilde{P}} g_c(c)\mu_{PM}(c)}{\sum_{c\in\widetilde{P}} \mu_{PM}(c)} \tag{3.4.16}$$

where $g_c(c)$ is the COG of the region with area code c.

Initialization of RAFRF

$A = \{A_1, \ldots, A_N\}$ Set of anchors
for $i = 1$ to $N - 1$ **do**
  for $j = i + 1$ to N **do**
    Consider anchor pair (A_i, A_j)
    for $k = 0$ to N **do**
    Define intervals/sets $LT_k^{(i,j)}$, $GT_k^{(i,j)}$ and $PI_k^{(i,j)}$ using (3.4.3) and (3.4.5)
    Define membership functions $\mu_{PI_k^{(i,j)}}$ using (3.4.6), (3.4.7), (3.4.8)
    end for
  end for
end for

Algorithmic Representation of RAFRF

1: x = crisp radical axis estimate of sensor node S with respect to (A_i, A_j)
2: **for** $i = 1$ to $N - 1$ **do**
3: **for** $j = i + 1$ to N **do**
4: **for** $k = 0$ to N **do**
5: Evaluate $\mu_{PI_k^{(i,j)}}(x)$
6: Define $K_{(i,j)} = \{k_{(i,j)} | \mu_{PI_k^{(i,j)}} > 0\}$
7: **end for**
8: **end for**
9: **end for**
8: Evaluate membership degree of the intersection region $\mu_{PM}(x)$ using (3.4.12)
9: Evaluate center of gravity of the corresponding region $g_c(x)$
10: Calculate estimated location of S using $\mu_{PM}(x)$ and $g_c(x)$ via (3.4.16)

3.5 Summary

In this chapter, an absolute distance measurement-based range-free localization method, RAFRF, is proposed. First, the reasons for developing a new algorithm are explained, and the generic form of the algorithm is presented. Then, the bases of the RAFRF are elaborated in Sections 3.2 and 3.3. Section 3.2 expressed the details of the utilized algorithm proposed in [1], and demonstrates the potential problematic parts of FRORF. Section 3.3 introduces a way of handling such issues in FRORF. In this section, a notion of Euclidean geometry called the radical axis is introduced as a solution for non-convexity. Section 3.4 combines fuzzy logic settings in FRORF and the radical axis property to get leverage from the performance of FRORF, which is tested in next chapter.

Chapter 4 illustrates experimental results. A comparative analysis of RAFRF is conducted, using results obtained with FRORF. Although RAFRF gives more accurate estimates, the algorithm needs to scan more sub-regions to obtain the center of gravity. Given a network with N anchors, the number of regions to deal with in RAFRF is computed in 3.4.15. However, the number of regions to calculate in FRORF is N^N for the same network.

Chapter 4

Simulation Tests and Numerical Assessments

In the simulations, the FRORF localization algorithm is implemented in a similar fashion to that proposed in [1] using MATLAB. Also, several experiments are conducted to evaluate the performance of RAFRF, and to compare it with the FRORF. Findings are analyzed based on changes in level of fuzzification (p), number of anchor nodes, range of the localization area, and noise.

It is intended to illustrate approximation errors applying the algorithms separately, and localization areas where the algorithms work well by indicating the differences of those errors values. Localization errors used in evaluations are defined by taking norms of the approximation errors, which are simply norm differences in the Euclidean domain between the actual points obtained in every step size, and their estimates found using the algorithms (see Equation (3.1.7)).

4.1 Localization of a Chosen Sensor Node

In this section, simulations are performed to test performance of the algorithms in the case of localization of one sensor node. Positions of anchor nodes are assigned randomly.

In the first set of simulations, we consider a network of three anchors located at $A_1 = [3, -0.5]^T$, $A_2 = [2, -2]^T$, $A_3 = [0.5, 0.8]^T$. The fuzzification parameter is taken as $p = 0.1$. Figures 4.1 and 4.2 show the estimation errors $e_S = \|\hat{S} - S\|$ vs. the actual sensor position $S = [S_x, S_y]^T$ to be estimated. For easier comparison of performances, the difference of these plots, $e_{S,FRORF} - e_{S,RAFRF}$, is plotted in Figure 4.9.

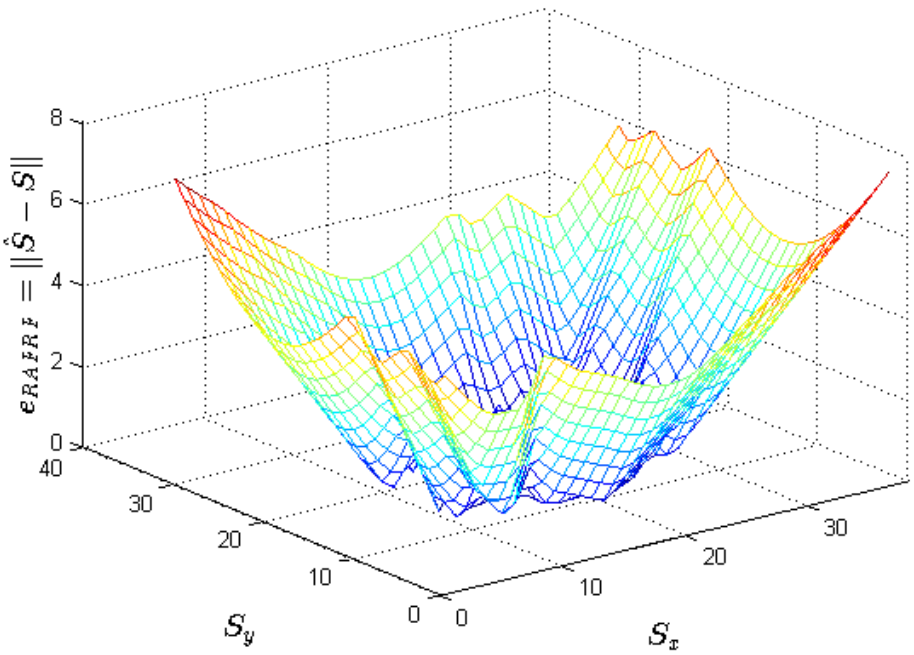

Figure 4.1: *Errors using RAFRF for* 3 *anchors with fuzzification parameter* $p = 0.1$.

As can be seen from Figure 4.2, FRORF does not work properly for points in the edge areas; however, RAFRF gives much better location approximations (see Figure 4.1). The

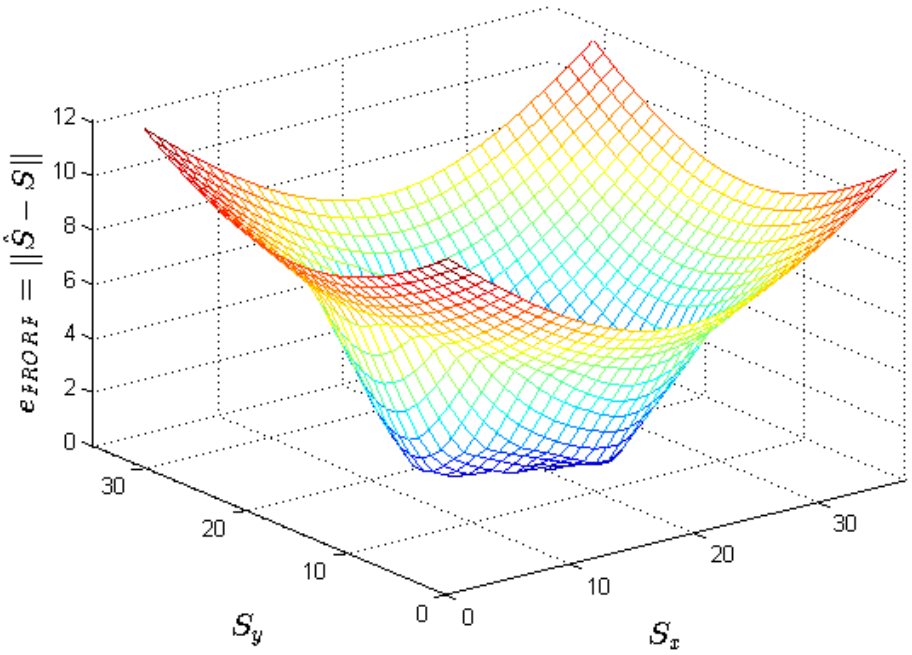

Figure 4.2: *Errors using FRORF for* 3 *anchors with fuzzification parameter* $p = 0.1$.

shortcoming of FRORF for points at the edges occurs because the algorithm is attracted by local optima. The differences between those errors, $e_{S,FRORF} - e_{S,RAFRF}$, represented above, are illustrated in Figure 4.3. Average and maximum values of e_S using RAFRF are 2.7168 and 7.5527, respectively, while the corresponding values for using FRORF are 6.0844 and 12.1709.

It is known that the fuzzification level has an impact on the width of the fuzzy regions [1]. To investigate the effect of change in the fuzzification value p, in the second set of simulations, we choose $p = 0.5$ while maintaining the same anchor positions $\{A_1, A_2, A_3\}$. We examine the algorithms for estimating position of the same $S = [S_x, S_y]^T$ in the same localization area. The results are demonstrated in Figure 4.4 and Figure 4.5.

It can be observed from Figures 4.4 and 4.5 that, as the values of fuzzification level

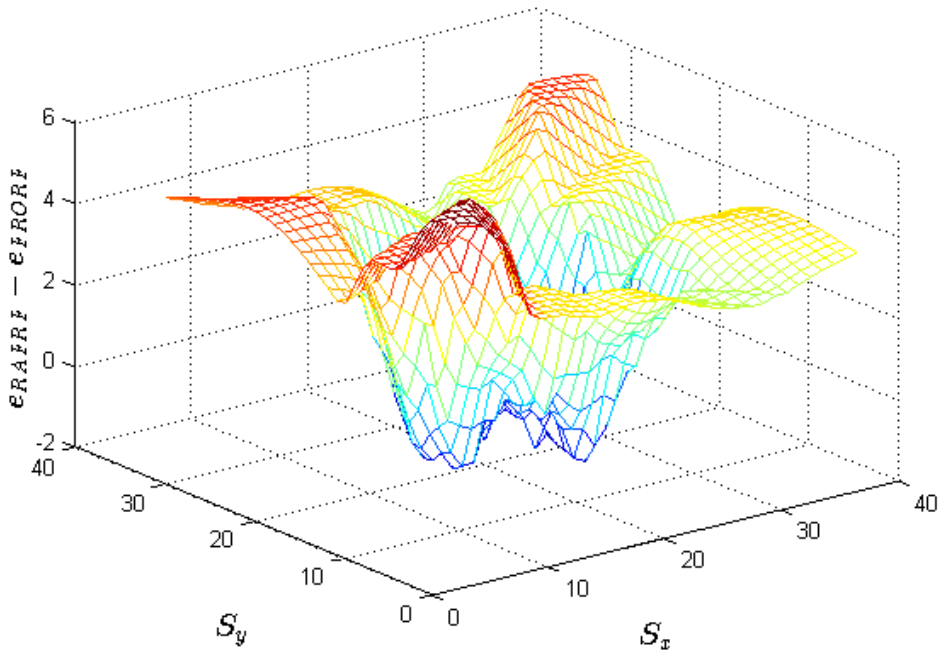

Figure 4.3: *Error differences between RAFRF and FRORF for* 3 *anchors when fuzzification parameter* $p = 0.1$.

increases, better location estimations are obtained, because the higher fuzzification value the inference system has, the higher coverage the fuzzy sets have. Such a high fuzzification level p also helps the fuzzy inference system deal with the larger number of regions. Average and maximum error values of using RAFRF are reduced to 1.0980 and 3.3503, respectively, and while the average error value using FRORF is reduced to 3.3152, the maximum remains as 7.9721.

Another evaluation criterion is the number of anchors (N) in the network. In the third set of simulations, we consider a network of four anchors located at $A_1 = [1, 1]^T$, $A_2 = [3, -0.5]^T$, $A_3 = [2, -2]^T$, $A_4 = [0.5, 0.8]^T$, and the fuzzification parameter remaining as $p = 0.1$. The algorithms try to approximate the coordinates of the same $S = [S_x, S_y]^T$

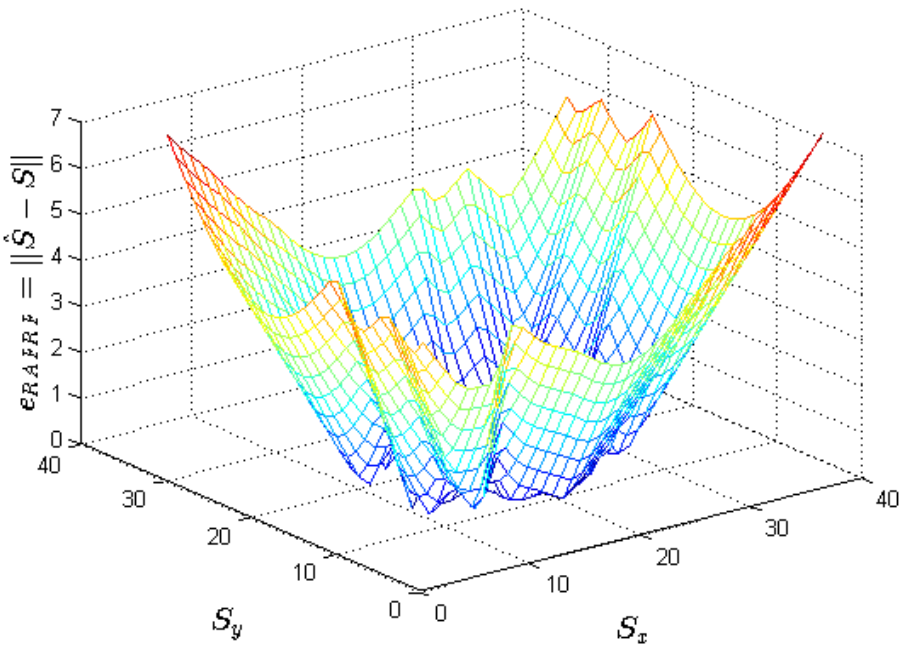

Figure 4.4: *Errors using RAFRF for* 3 *anchors with fuzzification parameter* $p = 0.5$.

in the same localization area.

As can be detected from Figure 4.7, the error values of RAFRF, e_{RAFRF}, show a significant decrease. Hence, RAFRF functions better with the network of four anchors compared to the network of three anchors. Average and maximum error values using RAFRF are 2.4626 and 10.9628, respectively. Although FRORF gives better results with four anchors, there is a very slight decrease in e_{FRORF}, the error values of FRORF, with average 6.0977 and maximum 12.1231.

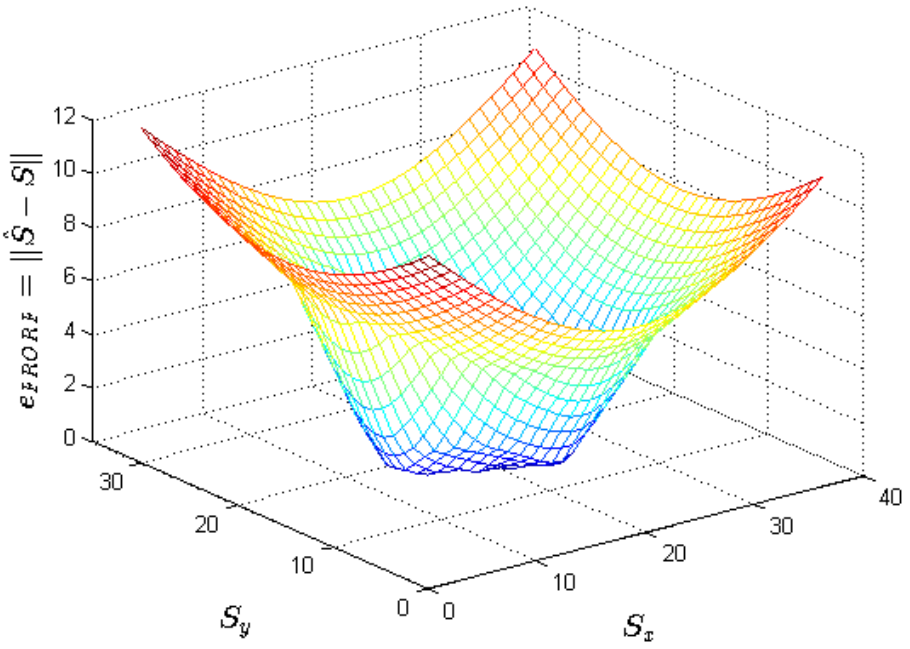

Figure 4.5: *Errors using FRORF for* 3 *anchors with fuzzification parameter* $p = 0.5$.

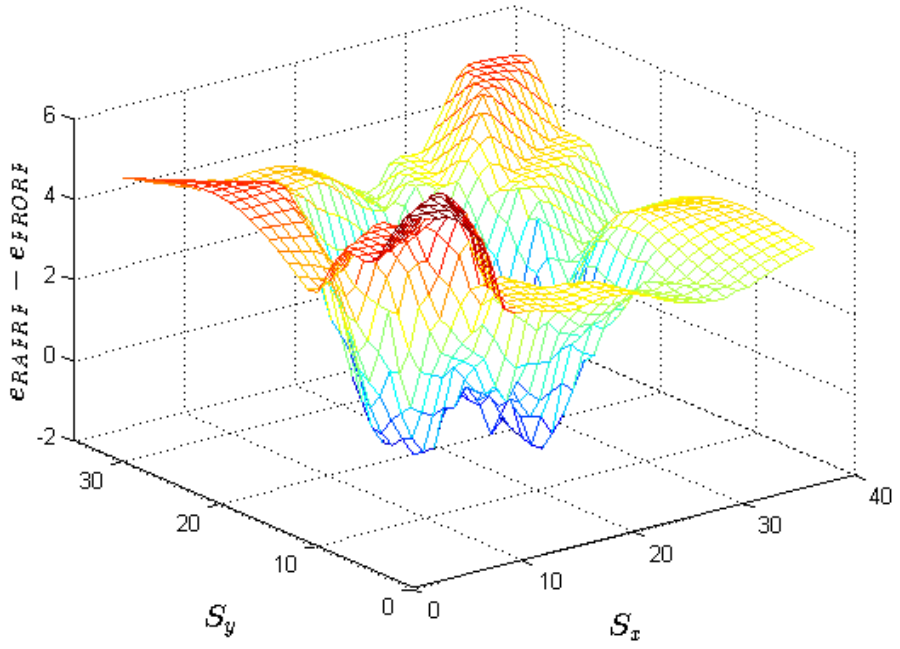

Figure 4.6: *Error differences between RAFRF and FRORF for* 3 *anchors when fuzzification parameter* $p = 0.5$.

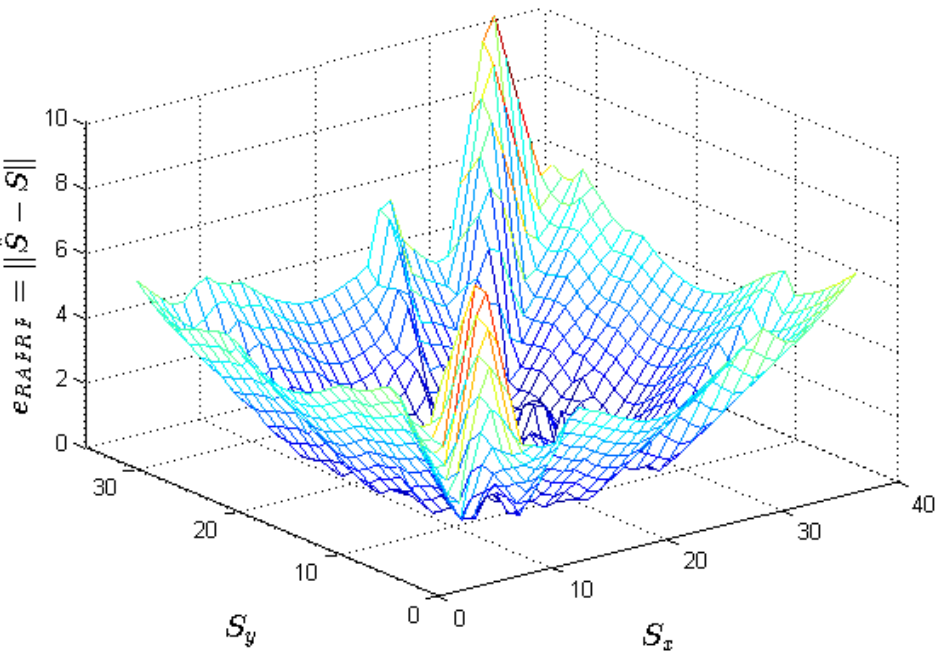

Figure 4.7: *Errors using RAFRF for* 4 *anchors with fuzzification parameter* $p = 0.1$.

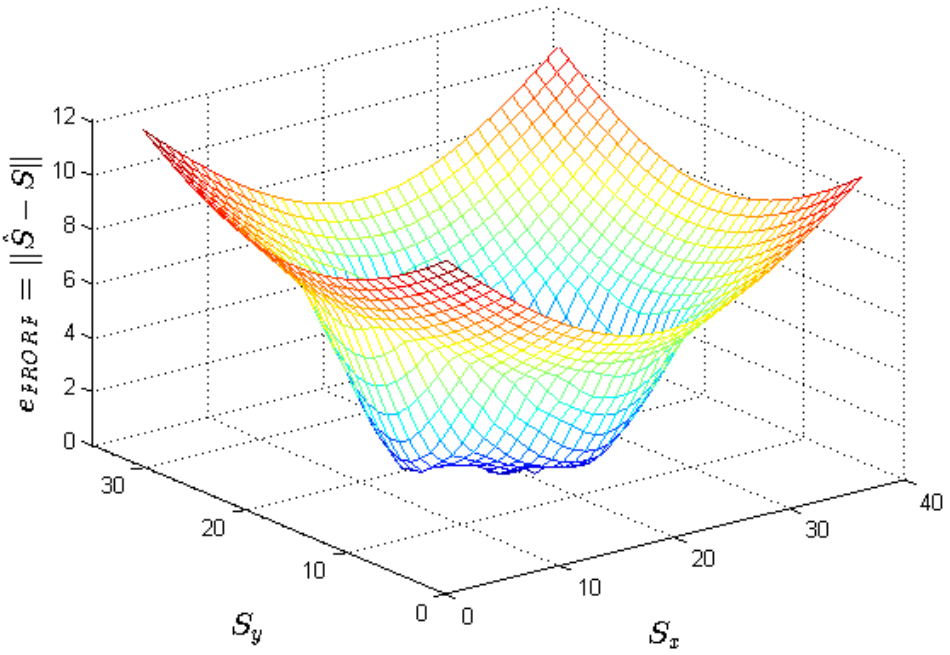

Figure 4.8: *Errors using FRORF for* 4 *anchors with fuzzification parameter* $p = 0.1$.

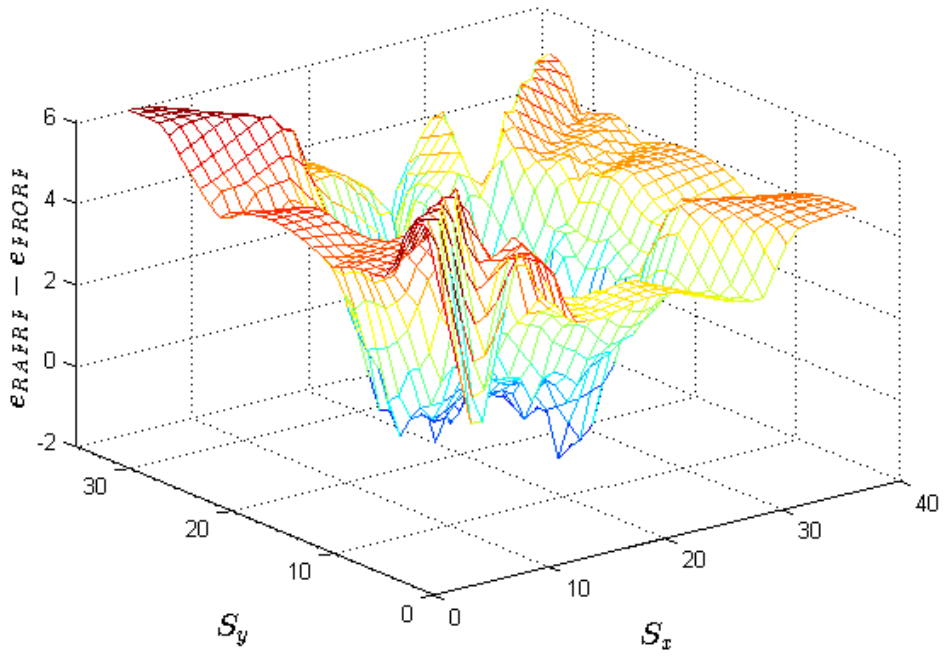

Figure 4.9: *Error differences between RAFRF and FRORF for* 4 *anchors when fuzzification parameter* $p = 0.1$.

4.2 Testing the Effect of Anchor Locations

To acquire general understanding, ten WSNs of four anchors are randomly generated and tested in different scenarios.

First, the anchors have randomly generated positions $(x, y) \in [-3, 6] \times [-6, 6]$. The localization performances for all $S = [S_x, S_y]^T$ locations, $S_x \in \{-6, -5.5, -5, \ldots, +9\}$, $S_y \in \{-9, -8.5, -8, \ldots, +9\}$, using FRORF and RAFRF are summarized in Table 4.2. As can

Table 4.1: *Average and maximum error values using FRORF and RAFRF*

	$ave(e_S)$		$\max(e_S)$	
Set No	FRORF	RAFRF	FRORF	RAFRF
1	2.8301	2.7961	6.7794	5.8563
2	2.2562	2.5856	7.9962	5.8057
3	2.4044	2.1457	5.3470	5.8652
4	2.4456	1.4516	6.9227	5.6235
5	2.2339	2.1740	5.4664	5.8313
6	1.8371	1.1956	5.4104	3.1399
7	1.8883	1.1956	5.4104	3.1399
8	2.7566	2.5186	7.0481	6.6550
9	1.8864	1.9380	6.9928	7.2340
10	2.4908	1.6635	7.7783	5.0856

be inferred from Table 4.1, RAFRF has better results than FRORF, in general. Then, the extension of the localization area range is considered, to test performance of the algorithms for nodes outside of the localization area. Maintaining the same sets of anchors, the range

of the area is extended to $[-3, 6] \times [-6, 6]$, and the results for estimating $S = [S_x, S_y]^T$ are shown in Table 4.2. The inference can be drawn from the results that RAFRF is

Table 4.2: *Average and maximum error values using FRORF and RAFRF in an extended area*

	$ave(e_S)$		$\max(e_S)$	
Set No	FRORF	RAFRF	FRORF	RAFRF
1	5.1400	4.6540	12.7279	12.6989
2	4.4296	4.2332	12.9292	11.4641
3	4.4969	4.0387	12.5084	10.1078
4	5.0208	3.4789	12.3840	12.0416
5	4.5949	3.9338	12.7279	10.8167
6	4.4684	3.4756	12.7279	10.5178
7	4.4228	2.9798	12.7279	10.8167
8	4.9771	3.9922	12.6352	10.6381
9	4.4297	3.9224	15.9784	12.6945
10	5.0646	3.4247	15.8777	10.7072

working more efficiently than FRORF does for nodes outside the region. Previous results were obtained in a noise-free case in general, algorithms which work well without noise will perform less well with noise. Therefore, Gaussian noise with $\sigma = 0.05$ is inserted, to examine the performance of the algorithms in the case with noise. Results are shown in Table 4.3 as follows.

The results demonstrate that the superior performance of RAFRF is valid in measurements with noise $\sigma = 0.05$. Further, higher Gaussian distance noise with $\sigma = 0.1$ is included into the algorithms to evaluate their performance, as summarized in Table 4.4.

Table 4.3: *Average and maximum error values using FRORF and RAFRF with noise ($\sigma = 0.05$)*

	$ave(e_S)$		$\max(e_S)$	
Set No	FRORF	RAFRF	FRORF	RAFRF
1	5.1410	4.6343	12.7279	12.7279
2	4.4288	4.1684	12.9293	11.6745
3	4.4971	4.0106	12.6320	9.9619
4	5.0229	3.4910	12.3840	12.0416
5	4.5943	3.9243	12.7279	10.8167
6	4.4674	3.4684	12.7279	10.5645
7	4.4254	2.9996	12.7279	11.4339
8	4.9785	4.0163	12.6319	10.5520
9	4.4319	3.9233	15.9023	12.6397
10	5.0657	3.4155	15.8777	10.6102

To sum up, FRORF and RAFRF are simulated by taking into consideration the change in number of anchors N, fuzzification level p, and area coverage. RAFRF gives better outcomes in different scenarios compared to FRORF.

Table 4.4: *Average and maximum error values using FRORF and RAFRF with noise ($\sigma = 0.1$)*

	$ave(e_S)$		$\max(e_S)$	
Set No	FRORF	RAFRF	FRORF	RAFRF
1	5.1427	4.6475	12.7279	12.3794
2	4.4310	4.2344	12.9255	11.9586
3	∞	4.0516	12.5588	9.9138
4	5.0233	3.6158	12.3840	12.3794
5	4.5948	3.9434	12.7279	10.8167
6	4.4710	3.4899	12.7279	10.5085
7	4.4242	3.0331	12.7279	11.0114
8	4.9788	4.0992	12.6405	10.5867
9	4.4344	3.9466	15.9201	12.7279
10	5.0707	3.4823	15.8777	10.5649

4.3 Discussion

Many localization algorithms have a common shortcoming: non-convexity. The non-convexity problem is solved in this study using a notion of Euclidean geometry called radical axis. This notion eliminates the problem by redefining fuzzy regions using convex sets. This is one reason for the superior performance of RAFRF.

Another performance failure of FRORF, as can observed from the simulations results above, is that it tends toward inaccuracy in the edges of the localization area. As the sensor node to be localized gets further from the anchors in the network, FRORF degrades. This is because the coordinates of center of gravity of some regions are not in the same region,

due to the shape of the region.

On the other hand, center of gravity calculation in FRORF does not require much computational time, since the number of overlapping regions is not high. Thus, FRORF can still estimate the location of a sensor with four or more anchors, while RAFRF has to handle a larger number of overlapping regions. This handling process takes much more time compared to the time needed by FRORF. Furthermore, there is a feasibility problem caused by non-matching grid points and regions during center of gravity calculation.

Chapter 5

Conclusion and Future Work

5.1 Summary

This study introduced a new fuzzy logic-based range-free localization technique for solution of Problem 1.2.1. Giving some relevant algorithms for source/sensor localization problems enhanced the basis of Problem 1.2.1. Further, the pros and cons of the measurement techniques were discussed, to analyse the algorithm better.

The algorithm the combines soft computing-based method of FRORF and a new convex optimization methodology. Such a combination works with uncertainty in distance measurements employing fuzzy logic, and eliminates multiple local minima using notions of radical axis and power lines.

Next, assessments of the proposed algorithm were demonstrated using comparisons with FRORF. The inferior performance of FRORF is shown in several illustrations for the system of four anchors. The center of gravity calculation method in RAFRF is problematic for five or more anchors due to the number of overlapping regions. This issue will be studied

in the future, and a brief guide is explained in the following section.

5.2 Extension to Cases with Higher Number of Anchors

WSNs are highly preferred because a WSN enables the system to communicate in a large area through the union of several small sensors. Anchors play a significant role in network communication; hence, deploying more anchors increases the network communication, resulting in more accurate location estimates.

In the proposed algorithm, RAFRF, we would like to extend the number of anchors to obtain better localization results. However, there is a computational difficulty resulting from center of gravity(CoG) calculation of fuzzy regions. The CoGs are calculated using a grid point check. Checking whether each point belongs to a region or not causes a feasibility problem, since the number of grid points are quite a bit smaller than the number of regions, meaning that the CoG of most regions cannot be calculated. Thus, there is a need for an alternative method, a closed-form solution, to calculate CoGs.

A new CoG calculation procedure; hence, follows:

Define the symmetry axis of $PI_k^{(i,j)}$, $l_{c_k}^{(i,j)}$ for $k = 0, \ldots, N$. Note that

$$l_{c_k}^{(i,j)} \perp A_i A_j. \tag{5.2.1}$$

Consequently,

$$l_{c_k}^{(i,j)} \perp e^{(i,j)} \tag{5.2.2}$$

where

$$e^{(i,j)} = A_j - A_i. \tag{5.2.3}$$

The symmetric axis $l_{c_k}^{(i,j)}$ passes through

$$\frac{A_{\rho_k}^{(i,j)} + A_{\rho_{k+1}}^{(i,j)}}{2} = y_k^{(i,j)}. \tag{5.2.4}$$

y_0 and y_N are defined separately taking A_0 and A_{N+1} on the border at the intersection point the region of interest. So, the line equation of the symmetric axis can be written as

$$l_{c_k}^{(i,j)} : (y - y_k^{(i,j)})^T e^{(i,j)} = 0. \tag{5.2.5}$$

Find the CoG of region with code $c = (k_{(1,2)}, \ldots, k_{(N-1,N)})$: this region, for ideal conditions, needs to contain an intersection point or any line pair $(l_{c_{k_{(i_1,j_1)}}}^{(i_1,j_1)}, l_{c_{k_{(i_1,j_1)}}}^{(i_1,j_1)})$. Mutual intersection points passing both line pairs satisfy the line equations as follows (dropping the superscript (i,j) for brevity)

$$\begin{aligned} (y_I - y_{k_1}^{(i_1,j_1)})^T e^{(i_1,j_1)} &= 0 \\ (y_I - y_{k_2}^{(i_2,j_2)})^T e^{(i_2,j_2)} &= 0. \end{aligned} \tag{5.2.6}$$

The system of linear equations can be written as

$$\begin{bmatrix} e^{(i_1,j_1)^T} \\ e^{(i_2,j_2)^T} \end{bmatrix} y_I = \begin{bmatrix} y_{k_1}^{(i_1,j_1)^T} e^{(i_1,j_1)} \\ y_{k_2}^{(i_2,j_2)^T} e^{(i_2,j_2)} \end{bmatrix}. \tag{5.2.7}$$

In short, the solution of the system is

$$y_I = E^{-1} B \tag{5.2.8}$$

where

$$E = \begin{bmatrix} e^{(i_1,j_1)^T} \\ e^{(i_2,j_2)^T} \end{bmatrix}, B = \begin{bmatrix} y_{k_1}^{(i_1,j_1)^T} e^{(i_1,j_1)} \\ y_{k_2}^{(i_2,j_2)^T} e^{(i_2,j_2)} \end{bmatrix}. \tag{5.2.9}$$

The CoG can be taken as the average of location coordinates of all these intersection points. Such a CoG calculation helps to eliminate the feasibility problem.

A brief explanation of the number of elements needed for CoG calculation is given in the following: for a line data-base, there are $M = \binom{N}{2}$ anchor pairs (i, j) for $i < j \in \{1, \ldots, N\}$. For each pair, there are $N + 1$ lines $l_{c_k}^{(i,j)}$, $k = 0, \ldots, N$. These lines are defined by (5.2.2) and (5.2.4). The database for $e^{(i,j)}$ and $y_k^{(i,j)}$ are given respectively as

Number of $e^{(i,j)}$: $\binom{N}{2} = \frac{N^2-N}{2}$ elements

Number of $y_k^{(i,j)}$: $\binom{N}{2} \times (N+1) = \frac{(N+1)N(N-1)}{2} = \frac{N^3-N}{2}$ elements.

From (5.2.8), the intersection database is given in the following:

E database : $\binom{\binom{N}{2}}{2} = \frac{N^4-2N^3-N^2+2N}{2}$ elements

B database : $\binom{\frac{N^3-N}{2}}{2} = \frac{N^5-2N^4-2N^3+N^2+2N}{2}$ elements

The numbers calculated above are much smaller than the number calculated in (3.4.15), which is $(N+1)^M$.

5.3 Extension to Three Dimensions

In this study, the design steps and analysis of the algorithm are explained when considering WSNs in two dimensions. This section provides an understanding of the algorithm for networks in three dimensions.

Similar to settings in two dimensions, the proposed algorithm can be designed for three-dimensional networks. The main components of the algorithm for three-dimensional networks differ from those mentioned in Section 3.1 in building convex fuzzy sets based on the radical plane property.

The radical plane is defined as a plane having the circle built as a result of the intersection of two spheres. The rest of the algorithm can be designed with respect to such a

notion. Note that in order to have a location estimate point, two different anchor pairs are needed.

References

[1] S. Velimirovic, G. Djordjevic, M. Velimirovic, and M. Jovanovic. Fuzzy ring-overlapping range-free (FRORF) localization method for wireless sensor network. *Computer Communications*, 35(13):1590–1600, 2012.

[2] F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci. Wireless sensor networks: A survey. *Computer Networks, IEEE*, (38):393–422, 2004.

[3] W. Chung, Y. Lee, and S. Yung. A wireless sensor network compatible wearable u-healthcare monitoring system using integrated ecg, accelerometer and spo2. In *in Proc. Annu. Int. Conf. IEEE Eng. Med. Biol. Soc., EMBC - "Pers. Healthc. Through Technology"*, pages 1526–1532. IEEE, 2008.

[4] T. He, S. Krishnamurthy, J.A. Stankovic, T. Abdelzaher, L. Luo, R. Stoleru, T. Yan, and L. Gu. Energy-efficient surveillance system using wireless sensor networks. In *In Mobisys*, pages 270–283. ACM Press, 2004.

[5] A. Boukerche, H. Oliveria, E. Nakamura, and A. Loureiro. Secure localization algorithms for wireless sensor networks. *IEEE Communications Magazine*, 46(4):96–101, 2008.

[6] G. Mao and B. Fidan. *Localization algorithms and strategies for wireless sensor networks*. Information Science Reference: IGI Publishing, 2009.

[7] M. Abdelhadi and M. Anan. A fuzzy-based collaborative localization algorithm for wireless sensor networks. In *in Collaboration Technologies and Systems (CTS), International Conference on*, pages 152–156. IEEE, 2012.

[8] A.A. Kannan. *Flip ambiguity mitigation for robust wireless sensor network localisation*. PhD thesis, The University of Sydney, 2010.

[9] A.R. Kulaib, R.M. Shubair, M.A. Al-Qutayri, and J. Ng. An overview of localization techniques for wireless sensor networks. In *Innovations in Information Technology (IIT), 2011 International Conference on*, pages 167–172. IEEE, 2011.

[10] G. Han, H. Xu, T.Q. Duong, J. Jiang, and T. Hara. Localization algorithms of wireless sensor networks: a survey. *Telecommunication Systems*, 47(3-4):1–18, 2011.

[11] S. Zekavat and R. Buehrer, editors. *Handbook of Position Location*. IEEE Press/Wiley, 2012.

[12] G. Mao, B. Fidan, and B.D.O. Anderson. Wireless sensor network localization techniques. *Computer Networks*, 51(10):2529 – 2553, 2007.

[13] B. Fidan and F. Kiraz. On convexification of range measurement based sensor and source localization problems. *submitted to IEEE Signal Processing Letters*, 2013.

[14] D.P. Bertsekas. *Nonlinear Programming*. Athena Scientific, 1995.

[15] S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004.

[16] B.T. Polyak. *Introduction to Optimization*. 1987.

[17] Y. Nesterov. *Introductory Lectures on Convex Optimization*. Kluwer, 2004.

[18] A. Tarighat and A.H. Sayed. Network based wireless localization. *IEEE Signal Processing Magazine*, pages 24–40, 2005.

[19] A.O. Hero and D. Blatt. Sensor network source localization via projection onto convex sets (POCS). In *Proc. IEEE Int. Conf. Acoust., Speech, and Sig. Proc*, volume 3, pages 689–692. IEEE, 2005.

[20] M. Rydstrom, E. G. Strom, and A. Svensson. Robust sensor network positioning based on projection onto circular and hyperbolic convex sets (pocs). In *Signal Processing Advances in Wireless Communications, 2006. SPAWC '06. IEEE 7th Workshop on*, pages 1–5. IEEE, 2006.

[21] B. Fidan, S. Dasgupta, and B.D.O Anderson. Guaranteeing practical convergence in algorithms for sensor and source localization. *Signal Processing, IEEE Transactions on*, 56(9):4458–4469, 2008.

[22] C. Meng, Z. Ding, and S. Dasgupta. A semidefinite programming approach to source localization in wireless sensor networks. *Signal Processing Letters, IEEE*, 15:253–256, 2008.

[23] N. Patwari, J.N. Ash, S. Kyperountas, A.O. Hero, R.L. Moses, and N.S. Correal. Locating the nodes: cooperative localization in wireless sensor networks. *Signal Processing Magazine, IEEE*, 22(4):54–69, 2005.

[24] S. Dandach, B. Fidan, S. Dasgupta, and B.D.O. Anderson. Adaptive source localization with mobile agents. In *in Proceedings of CDC*, pages 2045–2050. IEEE, 2006.

[25] H.C. So. *Handbook of Position Location*, chapter 2, pages 25–66. IEEE Press/Wiley, 2012.

[26] D. Blatt and A.O. Hero. APOCS: a rapidly convergent source localization algorithm for sensor networks. In *Statistical Signal Processing, 2005 IEEE/SP 13th Workshop on*, pages 1214–1219, 2005.

[27] R.V. Kulkarni, A. Forster, and G.K. Venayagamoorthy. Computational intelligence in wireless sensor networks: A survey. *Communications Surveys Tutorials, IEEE*, 13(1):68–96, 2011.

[28] F.O. Karray and C.D. Silva. *Soft Computing and Intelligent Systems Design*. Addison Wesley, 2004.

[29] K. LeBlanc and A. Saffiotti. Multirobot object localization: A fuzzy fusion approach. *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, 39(5):1259–1276, 2009.

[30] A.G. Dharne, L. Jaeyong, and S. Jayasuriya. Using fuzzy logic for localization in mobile sensor networks: simulations and experiments. In *American Control Conference, 2006*, pages 6 pp.–, 2006.

[31] H. Rowaihy, M.P. Johnson, and D. Pizzocaro. Detection and localisation sensor assignment with exact and fuzzy locations. In *5th IEEE International Conference*, pages 28–43. IEEE, 2009.

[32] S. Yun, W. Lee, J. andChung, E. Kim, and S. Kim. A soft computing approach to localization in wireless sensor networks. *Expert Systems with Applications*, 36(4):7552 –7561, 2009.

[33] S.M. Nekooei and M.T. Manzuri-Shalmani. Location finding in wireless sensor network based on soft computing methods. In *Control, Automation and Systems Engineering (CASE), 2011 International Conference on*, pages 1–5. IEEE, 2011.

[34] C. Liu, T. Scott, K. Wu, and D. Hoffman. Range-free sensor localization with ring overlapping based on comparison of received signal strength indicator. *International Journal of Sensor Networks (IJSNet)*, 2(5):399–413, 2007.

[35] S. Yun, J. Lee, W. Chung, E. Kim, and S. Kim. Centroid localization method in wireless sensor networks using tsk fuzzy modeling. In *in: Proceedings of International symposium on Advanced Intelligent Systems*, pages 971–974, 2008.

[36] S. Velimirovic, G. Djordjevic, M. Velimirovic, and M. Jovanovic. A fuzzy set-based approach to range-free localization in wireless sensor networks. *Facta Univ. Ser.: Elec. Energ.*, 2010.

[37] Y. Shang, W. Ruml, Y. Zhang, and M. Fromherz. Localisation from mere connectivity. In *in Proceedings of the 9th Annual International Conference on Mobile Computing and Networking, MOBICOM 2003*, pages 81–95. ACM, 2003.

[38] J.F. Sanford, S. Slijepcevic, and M. Potkonjak. *Localization in wireless networks: foundations and applications*. Springer, 2012.

[39] R.A. Johnson. *Advanced Euclidean Geometry*. Dover Publications, 2007.