

PATH PLANNING AND LOCALIZATION FOR MOBILE ANCHOR BASED
WIRELESS SENSOR NETWORKS

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
MIDDLE EAST TECHNICAL UNIVERSITY

BY

ECENAZ ERDEMIR

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

SEPTEMBER 2017

Approval of the thesis:

**PATH PLANNING AND LOCALIZATION FOR MOBILE ANCHOR BASED
WIRELESS SENSOR NETWORKS**

submitted by **ECENAZ ERDEMİR** in partial fulfillment of the requirements for the degree of **Master of Science in Electrical and Electronics Engineering Department, Middle East Technical University** by,

Prof. Dr. Gülbin Dural Ünver
Dean, Graduate School of **Natural and Applied Sciences** _____

Prof. Dr. Tolga Çiloğlu
Head of Department, **Electrical and Electronics Engineering** _____

Prof. Dr. T. Engin Tuncer
Supervisor, **Electrical and Electronics Eng. Dept., METU** _____

Examining Committee Members:

Assoc. Prof. Dr. Afşar Saranlı
Electrical and Electronics Engineering Department, METU _____

Prof. Dr. T. Engin Tuncer
Electrical and Electronics Engineering Department, METU _____

Assist. Prof. Dr. Elif Vural
Electrical and Electronics Engineering Department, METU _____

Assoc. Prof. Dr. Fatih Kamlı
Electrical and Electronics Engineering Department, METU _____

Assoc. Prof. Dr. Sinan Gezici
Electrical and Electronics Engineering Department, IDBU _____

Date: _____

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: ECENAZ ERDEMIR

Signature :

ABSTRACT

PATH PLANNING AND LOCALIZATION FOR MOBILE ANCHOR BASED WIRELESS SENSOR NETWORKS

Erdemir, Ecenaz

M.S., Department of Electrical and Electronics Engineering

Supervisor : Prof. Dr. T. Engin Tuncer

September 2017, 74 pages

In wireless sensor networks, sensors with limited resources are distributed in a wide area. Localizing the sensors is an important problem. Anchor nodes with known positions are used for sensor localization. A simple and efficient way of generating anchor nodes is to use mobile anchors which have built-in GPS units. In this thesis, a single mobile anchor is used to traverse the region of interest to communicate with the sensor nodes and identify their positions. Therefore planning the best trajectory for the mobile anchor is an important problem in this context. The mobile anchor stops on the trajectory to generate anchor nodes which are used in the position estimation of the unknown sensors. Various path planning methods for mobile anchors are proposed to localize as many sensors as possible by following the shortest path with minimum number of anchors. In this thesis, path planning and localization for mobile anchor based wireless sensor networks are investigated. Two novel path planning algorithms are proposed for static and dynamic schemes. These approaches use mobile anchors to cover the monitoring area with minimum path length and by stopping at minimum number of nodes. Moreover, alternating minimization algorithm is proposed for localizing the unknown sensor nodes non-cooperatively. The non-convex, NP-hard node localization problem is converted into a biconvex form and solved iteratively. The performances of the proposed path planning algorithms are compared with alternative approaches through simulations. The results show that more sensors are localized with less anchors in a shorter path and time for both schemes. Further-

more, alternating minimization algorithm provides an effective solution for the sensor localization problem. The simulation results show that the proposed localization approach is less prone to error accumulation than the alternative methods.

Keywords: Wireless Sensor Network Localization, Mobile Anchors based Sensor Networks, Path Planning, Alternating Minimization, Nonconvex Optimization

ÖZ

HAREKETLİ ÇAPAYA DAYALI KABLOSUZ SENSÖR AĞLAR İÇİN ROTA PLANLAMA VE KONUMLANDIRMA

Erdemir, Ecenaz

Yüksek Lisans, Elektrik ve Elektronik Mühendisliği Bölümü

Tez Yöneticisi : Prof. Dr. T. Engin Tuncer

Eylül 2017 , 74 sayfa

Kablosuz sensör ağlarda, güç ve işlem kabiliyeti sınırlı sensörler geniş bir alana dağıtılmış bulunmaktadır. Sensörleri konumlandırma problemi büyük önem arz etmektedir. Genellikle, sensör konumlandırmada bilinen çapa pozisyonları kullanılmaktadır. Çapa düğümü oluşturmanın basit ve etkin bir yolu da GPS birimi bulunan hareketli çapaların kullanılmasıdır. Bu tezde, gözetim alanını tarayarak sensörlerle iletişime girmek ve bu sensörleri konumlandırmak için tek bir hareketli çapa kullanılmaktadır. Bu sebeple hareketli çapa için en iyi rotayı planlamak büyük önem taşımaktadır. Rota üzerindeki hareketli çapa durakları, sensör konum kestiriminde kullanılan çapa düğümlerini oluşturmaya yaramaktadır. En az mesafe katedip en az sayıda çapa düğümü kullanarak, mümkün olan en çok sensörü konumlandırmak için çeşitli rota planlama yöntemleri önerilmektedir. Bu tezde, hareketli çapaya dayalı kablosuz sensör ağlar için rota planlama ve konumlandırma konusu işlenmektedir. Sabit ve dinamik durumlar için iki yeni rota planlama algoritması önerilmektedir. Bu yaklaşımlar, gözetim alanını en kısa yolu katedip en az sayıda düğümde duraklayarak kapsamak için hareketli çapa kullanmaktadırlar. Bununla birlikte, sensör ağını imcesiz olarak konumlandırmak için almaşık enküçültme algoritması önerilmektedir. Dış bükey olmayan, NP zor konumlandırma problemi yarı dışbükey bir forma getirilerek özyineli bir şekilde çözülmektedir. Önerilen rota planlama algoritmaları, simülasyonlar yardımıyla alternatif yaklaşımlarla karşılaştırılmaktadır. Sonuçlar göstermektedir ki, iki

yaklaşım da daha çok sensörü daha kısa yol katedererek ve daha az çapa düğümü oluşturarak konumlandırmaktadır. Öte yandan, almasıık enküçültme algoritması sensör konumlandırma problemi için etkin bir çözüm sağlamaktadır. Simulasyon sonuçlarına göre önerilen konumlandırma yöntemi hata birikmesine alternatif yöntemlerden daha az eğilimlidir.

Anahtar Kelimeler: Kablosuz Algılayıcı Ağ Konumlandırma, Hareketli Çapaya Dayalı Algılayıcı Ağlar, Güzergah Planlama, Almasıık Enküçültme, Dışbükey Olmayan Eniyileme.

To my family and Ali Can

ACKNOWLEDGMENTS

I would like to express my deep gratitude to my supervisor Prof. Dr. T. Engin Tuncer for his great guidance, encouragement and support. I was very lucky to have a supervisor who has always been patient to answer my questions and supported me to develop academic skills. I am thankful for his useful critiques and advices for my studies.

I would like to thank Assoc. Prof. İlkey Ulusoy for her valuable support during my Bachelor's and Master's degree. Her faith in me restored my self-esteem in academic life.

I would like to thank Berk Ünlü, Baver Özceylan and Hasan İhsan Turhan for their valuable encouragement and support during my assistantship, coordinatorship and thesis work. I am thankful for such great hosts providing me with the best studying environment during this summer.

I would like to thank Damla Alptekin, Işinsu Turan and Feza Mutlu for their enjoying friendship and encouragement. Throughout my thesis study, they have always been a great support and sources of motivation for me.

I would like to express my special thanks to each member of my family for their support, unconditional love and faith in me throughout my life. I am grateful to my mother, Neşe Erdemir, for being such a strong woman despite what she has been through. I am very lucky to have my sisters, Miraç Erdemir and Kıvanç Çakıcı, who have been mothers to me and I am thankful for their unconditional support. I would like to thank my father, Rıdvan Erdemir, for his patience and support since my first days at school. I also want to thank my lovely niece, Elif Deniz Çakıcı, for her making my life more joyful during the most stressful times.

Finally, I would like to thank Ali Can Doğan for all his love, support and patience during the past ten years. He has never lost his faith in me even in the hardest times. I am grateful for his encouragement which has helped me to shape my future academic plans.

TABLE OF CONTENTS

ABSTRACT	v
ÖZ	vii
ACKNOWLEDGMENTS	x
TABLE OF CONTENTS	xi
LIST OF TABLES	xiv
LIST OF FIGURES	xv
LIST OF ABBREVIATIONS	xviii
CHAPTERS	
1 INTRODUCTION	1
1.1 Contributions in This Thesis	3
2 MOBILE ANCHOR BASED WIRELESS SENSOR NETWORK LO- CALIZATION	7
2.1 Range-Free and Range-Based Localization	7
2.2 Cooperative and Non-Cooperative Localization	8
2.3 Node Localization	9
2.3.1 Lateration Algorithms	10
2.3.2 Triangulation Algorithms	12

2.3.3	Maximum Likelihood Algorithms	14
2.4	Mobile-Anchor Based Trajectory Planning	15
2.4.1	Random Models	16
2.4.2	Models with Temporal Dependency	17
2.4.3	Models with Spatial Dependency	17
2.4.4	Models with Geographic Restriction	18
2.4.5	Static Path Planning	18
2.4.5.1	SCAN and DOUBLE-SCAN Algorithms	19
2.4.5.2	LMAT Algorithm	19
2.4.5.3	CIRCLES and SPIRAL Algorithms . .	20
2.4.6	Dynamic Path Planning	21
3	A NOVEL STATIC PATH PLANNING ALGORITHM FOR MOBILE-ANCHOR BASED WIRELESS SENSOR NETWORK LOCALIZATION	25
3.1	SSD Search Phase	26
3.2	SSD Decision Phase	29
3.3	Simulation Results for SSD	32
4	A NOVEL DYNAMIC PATH PLANNING ALGORITHM FOR MOBILE-ANCHOR BASED WIRELESS SENSOR NETWORK LOCALIZATION	41
4.1	DSD Search Phase	42
4.2	DSD Decision Phase	42
4.2.1	Perpendicular Bisector Strategy	43

4.3	Simulation Results for DSD	46
5	LOCALIZATION USING ALTERNATING MINIMIZATION AL- GORITHM	57
5.1	Alternating Minimization	59
5.2	Simulation Results for Alternating Minimization	62
6	CONCLUSIONS	67
	REFERENCES	71

LIST OF TABLES

TABLES

Table 3.1 Simulation Parameters of the Static Scenario. 34

Table 4.1 Simulation Parameters of the Dynamic Scenario. 47

LIST OF FIGURES

FIGURES

Figure 2.1	(a) Non-cooperative and (b) Cooperative Wireless Sensor Networks	9
Figure 2.2	Localization classification in terms of mobility taken from [13]. . .	10
Figure 2.3	Node localization using trilateration [29].	11
Figure 2.4	Node localization using triangulation [5].	13
Figure 2.5	Classification of MANAL algorithms [5].	16
Figure 2.6	Movement trajectories of (a) SCAN, (b) DOUBLE-SCAN and (c) LMAT algorithms taken from [5]	20
Figure 2.7	Movement trajectories of (a) CIRCLES and (b) SPIRAL algorithms taken from [5]	22
Figure 3.1	Configuration of (a) anchor nodes in the form of equilateral triangles and (b) circular communication ranges over the surveillance region. .	26
Figure 3.2	Search-phase (a) mobile anchor trajectory and (b) anchor positions for the proposed static path planning algorithm SSD.	27
Figure 3.3	(a) The potential decision-phase trajectory and (b) the potential total mobile anchor trajectory for the proposed static path planning algorithm.	30
Figure 3.4	Examples of the mobile anchor trajectories for $m_2 = 3$, when m_1 is (a) odd and (b) even.	32
Figure 3.5	Example for the positions of sensors deployed in $50m \times 50m$ surveillance region in the form of (a) uniform, (b) Gaussian, (c) exponential and (d) Gaussian mixture distributions.	33
Figure 3.6	Static path planning results for (a) the number of residue unknowns, (b) the number of anchors and (c) distance traversed by the MA versus the number of sensors for the uniform sensor distribution in the noise free case.	36

Figure 3.7 Static path planning results for (a) the number of residue unknowns, (b) the number of anchors and (c) distance traversed by the MA versus the number of sensors for the Gaussian sensor distribution in the noise free case.	37
Figure 3.8 Static path planning results for (a) the number of residue unknowns, (b) the number of anchors and (c) distance traversed by the MA versus the number of sensors for the exponential sensor distribution in the noise free case.	38
Figure 3.9 Static path planning results for (a) the number of residue unknowns, (b) the number of anchors and (c) distance traversed by the MA versus the number of sensors for the Gaussian mixture sensor distribution in the noise free case.	39
Figure 4.1 Perpendicular bisector strategy with (a) search-phase anchors and (b) decision-phase anchors for the decision-phase of DSD.	44
Figure 4.2 An example for the proposed dynamic path planning algorithm trajectory.	46
Figure 4.3 Dynamic path planning results for (a) the number of residue unknowns, (b) the number of anchors and (c) distance traversed by the MA versus the number of sensors for the uniform sensor distribution in the noise free case.	49
Figure 4.4 Dynamic path planning results for (a) the number of residue unknowns, (b) the number of anchors and (c) distance traversed by the MA versus the number of sensors for the Gaussian sensor distribution in the noise free case.	50
Figure 4.5 Dynamic path planning results for (a) the number of residue unknowns, (b) the number of anchors and (c) distance traversed by the MA versus the number of sensors for the exponential sensor distribution in the noise free case.	51
Figure 4.6 Dynamic path planning results for (a) the number of residue unknowns, (b) the number of anchors and (c) distance traversed by the MA versus the number of sensors for the Gaussian mixture sensor distribution in the noise free case.	52
Figure 4.7 Number of anchors versus number of sensors for L-Shaped distribution.	53
Figure 4.8 Localized sensor ratio versus number of sensors for L-Shaped distribution.	53

Figure 4.9 Number of anchors versus number of sensors for U-Shaped distribution.	54
Figure 4.10 Localized sensor ratio versus number of sensors for U-Shaped distribution.	54
Figure 5.1 RMSE versus noise standard deviation (σ) for different algorithms.	63
Figure 5.2 Performance comparison for AMA linear and AMA approaches. . .	63
Figure 5.3 RMSE versus σ , when AMA is applied to DSD, GM and GM-VAF.	64
Figure 5.4 RMSE versus σ , when AMA and linear method are applied to DSD.	64
Figure 5.5 Performances of SOCP, PSO and AMA.	65

LIST OF ABBREVIATIONS

AMA	Alternating minimization algorithm
AOA	Angle of arrival
DSD	Dynamic search-and-decide
GM	Gauss-Markov
GPS	Global positioning system
LMAT	Localization algorithm with a mobile anchor node based on trilateration
LS	Least squares
MA	Mobile anchor
MANAL	Mobile anchor node assisted localization
MANET	Mobile ad-hoc networks
ML	Maximum likelihood
PBS	Perpendicular bisector strategy
RSS	Received signal strength
RW	Random walk
RWP	Random waypoint
SDP	Semidefinite programming
SOCp	Second order cone programming
SSD	Static search-and-decide
TDOA	Time difference of arrival
TOA	Time of arrival
WSN	Wireless sensor network

CHAPTER 1

INTRODUCTION

Wireless sensor network localization has become essential for various application areas, such as military surveillance, habitant monitoring, environmental monitoring, target tracking and wireless security [1]. Wireless sensor networks (WSNs) contain large number of inexpensive, randomly distributed sensors having limited battery and processing capability [2], [3]. In a WSN, sensors are able to collect physical data from the environment for target tracking, signal routing, surveillance and emergency responding purposes [1]. The sensor positions are unknown and critical for mapping the network area.

Estimating the coordinates of a sensor node or the special relationships among objects is known as localization [4]. Coordinates of sensors can be determined by Global Positioning System (GPS) measurements. However, it is not efficient to use GPS devices on hundreds of sensors due to high cost resulting from the battery consumption of GPS. Therefore, various strategies are developed to localize sensors using only some known positions in the network. These special nodes are called anchors (or “beacons”) and their locations are determined by GPS measurements [5]. The unknown sensor coordinates are estimated by using anchor positions and additional data coming from the sensors for various WSN schemes, such as connectivity, node-to-node distances which is used in this thesis. Having links to known anchor positions increases the chance of a sensor being localized. Therefore, anchor position quality is critically important for full coverage of the monitoring area and accurate sensor localization. In order to achieve good anchor configurations, a mobile vehicle (a robot) can be used to generate anchor nodes on a predefined path. Instead of using large

number of stationary anchors with costly GPS devices, using a single mobile anchor which generates anchor nodes at each stop is more cost efficient [6]. While this approach is more practical, mobile anchor requires a path planning and efficient routing. Path planning is the determination of the optimal mobile anchor route based on the information about the environment and the relative anchor location. Hence, effective algorithms are required to generate anchor nodes on the optimum path length and time.

In order to plan optimum trajectory for a mobile anchor, various algorithms are proposed [7, 8, 9, 10, 11, 12, 13, 14]. The main purpose in trajectory planning is to achieve maximum coverage of the surveillance area and high localization accuracy with minimum processing cost. Depending on the system requirements, path planning and mobility algorithms are proposed for the determination of mobile anchor trajectory. In this context, a mobile anchor moves along the monitoring area and broadcasts its coordinate signal at the time when it stops at certain positions. Neighboring sensors receive the position information as an anchor node coordinate and send back an acknowledgment signal to the mobile anchor. Path planning algorithms can be considered as static and dynamic approaches. In static path planning, the trajectory is planned before the mobile anchor starts its movement. Moreover, the path is designed for specific localization methods and it cannot be changed during the procedure. While the static path is not adjustable for environmental changes, it does not require a mobile robot with high processing capability to decide the trajectory. On the other hand, in the dynamic approach, the mobile anchor can adjust its path during the localization and it is flexible for the changes in localization methods. In the dynamic case, the mobile anchor requires high processing capability to make decisions considering the environmental factors. For both cases, when the anchor nodes are obtained, their positions are used together with distance measurements for sensor localization. The node-to-node distances can be obtained by TOA (Time of Arrival), TDOA (Time Difference of Arrival) and RSS (Received Signal Strength) measurements. Finding the unknown sensor positions by localization is an NP-hard, non-convex problem. Various optimization algorithms are proposed to reach the optimum solution for exact locations. In this thesis, novel static and dynamic path planning algorithms are proposed to determine the mobile anchor trajectory. Furthermore, a node localization

algorithm is proposed to localize sensors in the network.

In Chapter 2, background information about wireless sensor network localization is given in detail. Moreover, the literature on various node localization methods and mobile anchor trajectory planning algorithms are examined. In Chapter 3 and 4, novel static and dynamic path planning algorithms are presented. Two-phased solutions for both methods and the final trajectories are presented and the simulation results are discussed for each method. In Chapter 5, the proposed node localization algorithm is presented and its performance in terms of localization accuracy is discussed through the simulations. In Chapter 6, the conclusion of this thesis is given together with the summary of the proposed path planning and localization algorithms.

1.1 Contributions in This Thesis

In this thesis, two novel path planning algorithms are proposed for static and dynamic schemes in mobile anchor based wireless sensor networks. Besides, a node localization algorithm based on convex optimization is proposed for locating the unknown sensor nodes using the anchor positions generated in the path planning process.

In the static approach, the trajectory is designed based on trilateration which is the method of finding the position of a point by using distance information. The unknown location is found as the intersection point of the circles centered at known positions. The radii of these circles are the distances between the corresponding known position and the unknown position. In the proposed static path planning algorithm, the anchor positions are determined such that they form equilateral triangles and all the sensors are in the communication range of at least three anchors. The proposed static approach achieves the same localization performance as other trilateration based trajectories using less anchors and a shorter path. This is because, anchor nodes are not generated in sensor free areas. Two-phased proposed algorithm first divides the localization area into grids and detects the sensors in those grids. Then the mobile anchor revisits the occupied grids to generate anchor nodes in the form of triangles for localization. Hence, the proposed approach generates more anchors in the grids where there exist acknowledgment signals and less anchors in sensor free areas. This

shortens the path length and the total number of anchors generated. The proposed static path planning algorithm is compared with CIRCLES [9], LMAT (Localization algorithm with a Mobile Anchor node based on Trilateration) [11] and SLMAT [12] algorithms. It is presented in the simulation results that the proposed static approach performs better than the alternative methods in terms of path length, number of anchors and localization success. Broadcasting the current position fewer times for anchor generation and following a shorter distance requires less power. This provides an energy efficient localization. In terms of this criteria, the proposed static approach requires less power than the alternative approaches.

The dynamic path planning algorithm is proposed such that it can be adjustable to different localization techniques. Similar to the static approach, first the sensors in the grids are detected and the occupied grids are revisited for localization. In order to generate the required number of anchors for localization, perpendicular bisector strategy [6] is used. This strategy is based on generating non-collinear anchor nodes, since collinearity increases the position estimation error. While all sensors in the area are covered and detected, the redundant areas are not revisited for localization. This decreases the path length and the required number of anchor nodes. The proposed dynamic approach is compared with the known techniques in the literature [6], [15], [16]. It is presented in the simulation results that, the proposed dynamic algorithm performs better than the alternative methods in terms of localization success, the number of anchors and the path length.

The difference between the static and dynamic scenario is that the static path planning is done prior to the localization procedure and cannot be changed during the localization [17]. In the proposed static approach, the path is determined after the sensor detection in the surveillance area. Then, the mobile anchor starts its movement along the predefined path. Moreover, the static path is designed for a specific localization technique which is trilateration in this thesis. However, in the dynamic path planning, the trajectory can be modified with the demands of user or the unknown nodes [17]. For instance, the number of anchors required for localization, the mobile anchor trajectory and path length can be changed in real time. In the proposed dynamic approach, after the detection of sensors, the mobile anchor determines the position and the number of anchor nodes to be generated according to the demand of

the user. Since the localization of the sensors is performed during the second phase of the dynamic approach, modification of the path is possible for parameter changes in this phase.

The sensor network is localized using the generated anchor positions at the end of path planning process. Alternating minimization algorithm (AMA) is proposed for non-convex sensor localization problem. The original form of the problem is converted into a biconvex form and the relations between the distances and node positions are satisfied implicitly. The modified problem is solved iteratively by alternating minimization. AMA is compared with linear localization [18], second order cone programming (SOCP)[19] and particle swarm optimization (PSO) [20] algorithms and the effectiveness of AMA is analyzed in terms of sensor position accuracy.

CHAPTER 2

MOBILE ANCHOR BASED WIRELESS SENSOR NETWORK LOCALIZATION

Wireless sensor networks consist of a vast number of randomly distributed sensors (termed as “unknown nodes” or “sensor nodes”) with unknown positions. The sensors are used to gather physical information from the environment. The positions of these sensors are important for mapping the network. Therefore, sensor network localization is defined as estimating the coordinates of a sensor node or relationships among objects [4]. In order to localize the sensors, special nodes with known coordinates and their relationship with sensors are used in various WSN schemes.

Wireless sensor networks can be mainly classified as range-free and range-based, cooperative and non-cooperative, mobile-anchor based and static-anchor based schemes [21]. Mobile anchor based sensor network localization contains the mobile anchor trajectory planning and node localization. On the other hand, static anchor based schemes contains only node localization. The details of these classifications are explained in the following sections.

2.1 Range-Free and Range-Based Localization

In range-based localization, the existence of anchor nodes is required. The remaining nodes in the network use node-to-node distances or inter-node angles to estimate their positions [22]. The distance measurements for WSN localization are obtained by the methods such as TOA (Time of Arrival) [23], TDOA (Time Difference of Arrival) [24] and RSS (Received Signal Strength) [25]. Besides, it is possible to use

angle measurements obtained by AOA (Angle of Arrival) [4] method for locating the sensors. Although the range-based localization is costly due to expensive ranging measurements, it provides high localization accuracy for the cases in which the exact locations are critical.

In range-free localization, connectivity information among sensor nodes is used to localize sensors. While the existence of anchor nodes is still necessary for anchor-based solutions, there are also anchor-free and hybrid techniques for the range-free schemes [26]. For example, controlled events (such as light) can be generated in the surveillance region for the anchor-free case. The unknown position of a sensor can be found by using the event properties and the time that the event is sensed [26]. In some WSN applications, very high localization accuracy is not required. On the other hand, the low power consumption may be more critical. The range-free techniques provide a low accuracy-low cost localization for WSNs [22]. The most common methods in the range-free localization are APIT (Approximate Point-In-Triangle) [22] and DV-Hop (Distance-Vector Hop) [27]. In DV-Hop algorithm, only anchor positions and node-to-node connectivities are known. For sensor localization, approximate distances are estimated from the connectivities and the anchor coordinates. In APIT algorithm, triangles are formed using any three neighboring anchors of the unknown node. Determining whether the unknown node is within these triangles, the node localizes itself up to a certain accuracy [22]. The range-free localization is a low cost technique for large scale and dense wireless sensor networks. On the other hand, it may not be preferable for sparse networks due to its low accuracy resulted from insufficient connectivity information.

2.2 Cooperative and Non-Cooperative Localization

In non-cooperative localization, sensors can only communicate with anchor nodes. In Figure 2.1a, filled and empty circles represent anchor and sensor nodes, respectively. Moreover, the straight lines between the circles represent their connection. As in Figure 2.1a, the sensors only have connections with the anchor nodes in a non-cooperative WSN. For instance, in non-cooperative range-based localization, node-to-node distances between a sensor and a certain number of anchors are required to

be known for locating the sensor. Similarly, the connectivity information between a sensor and a number of anchors are used for non-cooperative range-free sensor localization.

In cooperative localization, sensors can communicate with each other and they may still have links to anchor nodes as expressed in Figure 2.1b. The dashed lines in Figure 2.1b represent the lines connecting the sensors only. That is, in cooperative WSNs, sensor nodes can communicate with both anchors and sensors. This leads to an increased connectivity in the network. While sensor-to-sensor distances are used for the range-based scenarios, range-free methods require sensor connectivity information. Cooperation among unknown nodes provides an improved coverage and higher localization accuracy than non-cooperative techniques [28]. However, connectivity among the unknown nodes increases the computational complexity in cooperative localization.

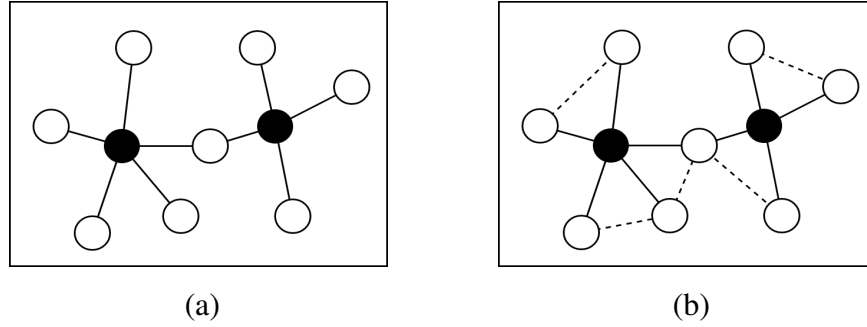


Figure 2.1: (a) Non-cooperative and (b) Cooperative Wireless Sensor Networks

2.3 Node Localization

In wireless sensor networks, anchor positions play an important role for improving the localization accuracy. In a network, increasing the number of anchor nodes provides more reference points. This leads to less position errors.

Sensor networks may consist of stationary anchor nodes with fixed coordinates or mobile anchors following a specific path as shown in Figure 2.2. In both mobile and static anchor schemes, the sensors can be either stationary or mobile. In static-anchor based static-sensor networks, only the localization procedure is taken into account.

On the other hand, in mobile-anchor schemes, path planning is carried out before localization. In this section, common range-based localization algorithms for sensor networks are explained. They can be classified as lateration, maximum likelihood and triangulation algorithms [5]. There are also ad-hoc node localization algorithms which are designed for specific purposes in ad-hoc wireless sensor networks.

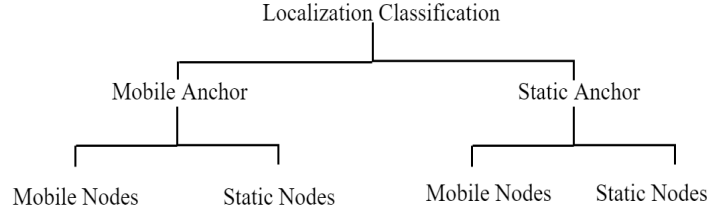


Figure 2.2: Localization classification in terms of mobility taken from [13].

2.3.1 Lateration Algorithms

Lateration algorithms are range-based methods that are used to estimate unknown node positions from measurements of the difference in distance. Bi-lateration, tri-lateration and multi-lateration methods are used for localizing the sensors in one-dimension (1D), two-dimensions (2D) and three or higher dimensions (3D+), respectively [29]. Sensor network problem is generally solved for 2D in general. Hence, in this section, trilateration algorithm is described in detail.

Trilateration algorithm is defined as finding the position of an unknown node using the intersections of three circles geometrically [29]. Each circle is centered at an anchor as in Figure 2.3. The radius of each circle is determined by the distance measurements between the unknown node and the corresponding anchor. Sensor coordinates are found to be the intersection point of the three circles. When there are distance measurement errors in the medium, the intersection may be a region rather than a single point. This results in erroneous solution.

Localization of a sensor is geometrically described in Figure 2.3. The mathematical solution for this problem consists of the distances between the sensor and three anchors. The distance from the k^{th} anchor to the i^{th} sensor is expressed by

$$d_{ki} = \sqrt{(a_{kx} - x_{ix})^2 + (a_{ky} - x_{iy})^2} \quad (2.1)$$

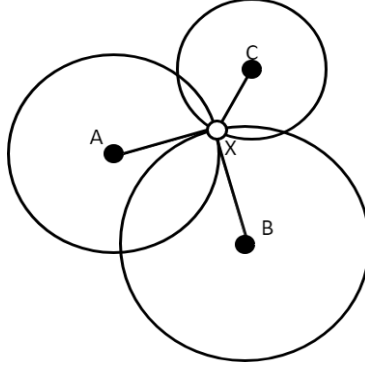


Figure 2.3: Node localization using trilateration [29].

In the expression (2.1), a_{kx} and a_{ky} are x and y components of the vector \mathbf{a}_k which is the k^{th} anchor coordinate. Similarly, x_{ix} and x_{iy} are the components of \mathbf{x}_i which represents the i^{th} sensor.

Solution for the localization problem is nonlinear due to the quadratic form of distance formulations. To linearize the nonlinear system, at least three distance measurements are required in 2D. Then, taking the square distances and subtracting the first distance equation from the rest of the equations, the problem is linearized. The relation for the distances between k number of anchor nodes and the i^{th} sensor node are given as,

$$\begin{aligned}
 (x_{ix} - a_{1x})^2 + (x_{iy} - a_{1y})^2 &= d_{1i}^2 \\
 (x_{ix} - a_{2x})^2 + (x_{iy} - a_{2y})^2 &= d_{2i}^2 \\
 (x_{ix} - a_{3x})^2 + (x_{iy} - a_{3y})^2 &= d_{3i}^2 \\
 &\vdots \\
 (x_{ix} - a_{kx})^2 + (x_{iy} - a_{ky})^2 &= d_{ki}^2
 \end{aligned} \tag{2.2}$$

where x_{ix} and x_{iy} are x and y coordinates of the unknown position of the i^{th} sensor; a_{kx} and a_{ky} are x and y coordinates of the k^{th} anchor and d_{ki} is the Euclidean distance between the k^{th} anchor and the i^{th} sensor [30]. When the last equation is subtracted from the first one and the resulting $n - 1$ equations are arranged, a relation in the form $\mathbf{Ax} = \mathbf{b}$ is obtained, where

$$\mathbf{x}_i = [x_{ix} \ x_{iy}]^T \tag{2.3}$$

$$\mathbf{A} = 2 \begin{pmatrix} a_{1x} - a_{kx} & a_{1y} - a_{ky} \\ a_{(k-1)x} - a_{kx} & a_{(k-1)y} - a_{ky} \end{pmatrix} \quad (2.4)$$

$$\mathbf{b} = \begin{pmatrix} (a_{1x}^2 + a_{1y}^2) - (a_{kx}^2 + a_{ky}^2) - d_{1i}^2 + d_{ki}^2 \\ (a_{(k-1)x}^2 + a_{(k-1)y}^2) - (a_{kx}^2 + a_{ky}^2) - d_{(k-1)i}^2 + d_{ki}^2 \end{pmatrix} \quad (2.5)$$

The unknown node position is found by solving a linear problem in the form of $\mathbf{Ax} = \mathbf{b}$, where \mathbf{A} and \mathbf{b} are defined as (2.4) and (2.5) [5]. Trilateration method has computational complexity of $O(k^2)$ where k is the number of neighboring anchors [31]. Although trilateration is a commonly used algorithm, its localization accuracy is directly affected by the geometry of anchors.

In [32], the quality of trilateration is investigated in detail. Poor localization accuracy can result from; non-intersecting circles in noisy environment, flip ambiguity, anchor non-consistency due to high number of neighboring anchors and error propagation in multi-hop networks. Yang and Liu quantifies the quality of anchor geometries and assign a confidence value to each anchor in [32]. The localization procedure starts in the order from high to low confidence anchors iteratively. Hence, the localization accuracy of sensor nodes is increased.

In [33], Sweeps algorithm is proposed to iteratively solve the flip ambiguity in sparse networks. Firstly, the sensors with at least three neighboring anchors are localized and assigned as reference points. Then the remaining sensors are localized according to their consistency with the reference points. Since the connectivity among sensors are used iteratively in [33], the flip ambiguity is solved without additional anchors.

2.3.2 Triangulation Algorithms

Unlike trilateration which uses sensor-to-anchor distances for localization, triangulation is the algorithm of locating an unknown node in 2D using the angular distances between the node and three separate pairs of anchors [34].

In Figure 2.4, x is the unknown coordinate of a sensor and \mathbf{A} , \mathbf{B} and \mathbf{D} are anchor coordinates. Moreover, c_i 's are the center point of the circles. If the angles between

lines connecting the sensor with each anchor are known, \mathbf{x} position is found by triangulation algorithm [34]. Applying the law of sines to triangles $(\mathbf{A}, \mathbf{x}, \mathbf{D})$, $(\mathbf{D}, \mathbf{x}, \mathbf{B})$ and $(\mathbf{B}, \mathbf{x}, \mathbf{A})$, radii of the corresponding circles are found as in,

$$r_{c1} = \frac{1}{2} \frac{\|\mathbf{D} - \mathbf{B}\|}{\sin(\widehat{\mathbf{BxD}})} \quad (2.6)$$

The midpoint of the line segment (\mathbf{D}, \mathbf{B}) is given as,

$$\mathbf{m}_{DB} = \mathbf{B} + \frac{1}{2}(\mathbf{D} - \mathbf{B}) \quad (2.7)$$

\mathbf{c}_1 can be determined by solving (2.8).

$$\frac{(\mathbf{D} - \mathbf{B})}{\|\mathbf{D} - \mathbf{B}\|} R(\pi/2) = \frac{(\mathbf{c}_1 - \mathbf{m}_{DB})}{\|\mathbf{c}_1 - \mathbf{m}_{DB}\|} \text{ for } R(\pi/2) = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} \quad (2.8)$$

where the matrix $R(\pi/2)$ is a counterclockwise rotation of $\pi/2$ rad. (2.8) can be rewritten as,

$$\mathbf{c}_1 = \mathbf{m}_{DB} + r_{c1} \cos(\pi - \sin(\widehat{\mathbf{BxD}})) \frac{(\mathbf{D} - \mathbf{B})}{\|\mathbf{D} - \mathbf{B}\|} R(\pi/2) \quad (2.9)$$

where r_{ci} is the radius of the i^{th} circle. \mathbf{x} is the intersection point of these circles and is calculated by,

$$\|\mathbf{x} - \mathbf{c}_i\|^2 = r_{ci}^2 \quad (2.10)$$

Hence, it is stated in [34] that the unknown sensor position \mathbf{x} can be calculated using the relations (2.6), (2.7) and (2.8).

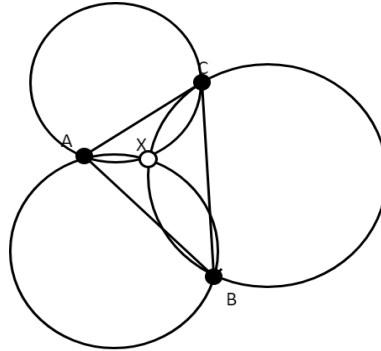


Figure 2.4: Node localization using triangulation [5].

2.3.3 Maximum Likelihood Algorithms

Sensor coordinates can be found using maximum likelihood (ML) estimation method. An unknown node can be localized with the help of sensor-to-anchor and sensor-to-sensor distances obtained from TOA measurements. The maximum likelihood formulation of a non-cooperative sensor localization problem can be expressed as,

$$\underset{\mathbf{x}_i}{\text{Minimize}} \quad \sum_{(i) \in N, (j) \in M} \frac{1}{\sigma_{ij}^2} (\|\mathbf{x}_i - \mathbf{a}_j\|_2 - d_{ij})^2 \quad (2.11)$$

where \mathbf{x}_i is the position of i^{th} sensor, \mathbf{a}_j is the position of j^{th} anchor, σ_{ij}^2 is the variance of additive white Gaussian noise, d_{ij} is the distance between i^{th} sensor and j^{th} anchor. In (2.11), the squares of the errors between real and estimated distances are summed and minimized. In this way, the equality relation between $\|\mathbf{x}_i - \mathbf{a}_j\|_2$ and d_{ij} is satisfied. Since it is a non-cooperative localization problem, only the links between the sensors and the anchors are considered.

For a cooperative sensor localization problem, the ML formulation can be expressed as,

$$\underset{\mathbf{x}_1, \dots, \mathbf{x}_N}{\text{Minimize}} \quad \sum_{(i,j) \in N} \frac{1}{\bar{\sigma}_{ij}^2} (\|\mathbf{x}_i - \mathbf{x}_j\|_2 - \bar{d}_{ij})^2 + \sum_{(i) \in N, (j) \in M} \frac{1}{\sigma_{ij}^2} (\|\mathbf{x}_i - \mathbf{a}_j\|_2 - d_{ij})^2 \quad (2.12)$$

where \bar{d}_{ij} is the distance between i^{th} and j^{th} sensors, $\bar{\sigma}_{ij}^2$ is the noise variance between the i^{th} and j^{th} sensors. Since it is a cooperative localization problem, both sensor-to-sensor and sensor-to-anchor distances are considered.

In [19], RSS based localization problem is solved for non-cooperative and cooperative cases. Since the ML estimator is computationally complex, a non-convex least squares (LS) estimator corresponding to the ML estimator for small noise is introduced. Second order cone programming (SOCP) and mixed semidefinite programming (SDP)/SOCP estimators are proposed for non-cooperative and cooperative localization schemes, respectively. Both cases where source transmit power and path loss exponent are known and unknown are considered. The proposed approaches in

[19] estimate the node positions and unknown parameters effectively.

As an ad-hoc node localization algorithm, a particle swarm optimization (PSO) approach is proposed in [20] for ad-hoc WSN's. Here, particles are generated in the search space randomly. Assuming each particle is the potential unknown position, the errors are calculated. An iterative optimization procedure is performed to find the searching directions of the particles which minimize these errors using the maximum likelihood approach. The unknown sensor position is estimated to be where the particle positions converge. This procedure is applied to the network iteratively.

2.4 Mobile-Anchor Based Trajectory Planning

In static wireless sensor networks, anchor node positions play an important role for improving the localization accuracy. However, increasing the number of costly anchors also increases the cost of the localization process. Moreover, in static anchor scenarios, all anchors become useless after the localization process. Hence, using a mobile anchor with the ability to broadcast its position decreases the cost of the localization [6].

According to the localization procedure based on mobile anchor and stationary sensor networks, a GPS carrying vehicle moves along the monitoring area. The mobile anchor broadcasts a GPS signal carrying its position information at each stop. If the mobile anchor and sensors can communicate, the mobile anchor receives acknowledgments from the neighboring sensors. In order to consider a sensor as the neighbor of a mobile anchor, the mobile anchor must be in the communication range of the corresponding sensor. This range is represented by a circular region of radius R .

Impact of a mobile anchor path on the quality of anchor positions is critical in terms of localization accuracy. Hence, there are various algorithms for the path decision procedure, namely Mobile Anchor Node Assisted Localization (MANAL) algorithms [5]. They can be classified as in Figure 2.5. The MANAL algorithms are divided into two categories as mobility models and path planning. Besides, mobility models are also divided into random models, models with temporal dependency, models with spatial dependency and models with geographic restrictions. In addition, localization

based on path planning is also classified as static and dynamic path planning.

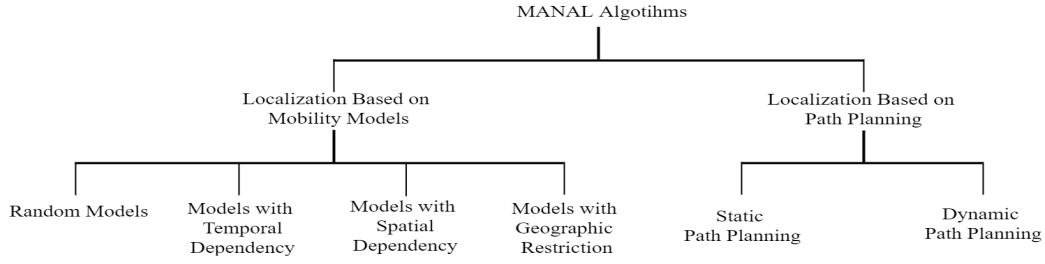


Figure 2.5: Classification of MANAL algorithms [5].

As one of the categories of MANAL algorithms, mobility models describe the pattern of mobile anchor trajectories and the change of the parameters, such as position, speed and acceleration in time. Mobility models are used generally in mobile ad-hoc networks (MANETs) which comprise of wireless mobile nodes forming self-configuring network without using any existing infrastructure [35]. Here, self-configuring means that nodes in a mobile ad-hoc network do not need extensive knowledge of network. Hence, mobility models offer flexibility and freedom for WSN localization.

2.4.1 Random Models

In random models, the mobile anchor velocity and direction is determined randomly. That is, the next position of the mobile anchor at the next time interval is random.

In Random Way-point (RWP) [36], the mobile anchor chooses a target position at each move. The velocity is randomly chosen from the interval $[v_{min}, v_{max}]$, where v_{min} and v_{max} are minimum and maximum allowed velocities for the mobile anchor. After its move, the mobile anchor stops during t_{pause} . If $t_{pause} = 0$, this is a continuous mobility. Similar to RWP, Random Walk (RW) model chooses the velocity and direction of the mobile anchor randomly.

In [36] which is a random walk model, the mobile anchor moves with a random mobility in the localization area. Here, the velocity and angular direction of the mobile anchor is chosen randomly from the intervals $[v_{min}, v_{max}]$ and $[0, 2\pi]$. Inside the localization region, the mobile anchor moves randomly and stops at each constant time interval to create anchor nodes. If the mobile anchor exits the region, it returns to

the area with a predefined angle. Since the trajectory is random, this model does not guarantee full coverage of the localization area. However, due to the randomness in velocity and direction, sharp turns and sudden stops occur in both RWP and RW models.

2.4.2 Models with Temporal Dependency

In localization based on models with temporal dependency, there is a correlation between velocities at different time instants. Unlike random models, velocity and direction have memory and their mobility model is not random. [35]

In [37], Kuang et al. propose Gauss-Markov (GM) mobility for the mobile anchor velocity and direction. GM is a memory model which determines the current velocity and direction from the previous time velocities and directions. Then, the next position of the mobile anchor is found by using the current mobile anchor position, velocity and direction. However, it takes a long time for the mobile anchor to enter the localization region, when it exits the area. Unlike the approach in [36], [37] is not prone to sudden velocity and direction changes because the level of randomness can be adjusted.

2.4.3 Models with Spatial Dependency

In the mobility models with spatial dependency, mobile nodes do not move independently as in random models. The velocity and direction are affected by the neighboring nodes. Therefore, the velocities of different nodes are correlated in space. This leads to a group mobility. [35]

In [38], the effect of a correlated mobility on the data transfer speed and delay performance in MANETs is investigated. Here, the mobile nodes in the network are divided into groups. The center of each group moves according to independent and identically distributed (i.i.d.) mobility models. Hence, correlated mobility provides a better performance than independent node mobility.

2.4.4 Models with Geographic Restriction

Environments with obstacles requires the usage of the mobility models with geographic restrictions. The randomness of the velocity and direction is bounded by the environment restrictions. Real life applications require mostly solutions for obstacle problems. [35]

In [39], a visible obstacle mobility model based on activity area (VOMBAA) is proposed for MANETs. The mobile anchors can only perceive visible obstacles. The anchor finds the first obstacle to bypass. Using the A* [40] function, it calculates the assessed value of vertices that can be seen. Then, it selects a vertex as an intermediate point. The mobile anchor moves to this point to go around the obstacle, and make the distance to destination the current shortest.

As another category of MANAL algorithms, path planning aims at proposing a feasible mobile anchor trajectory in the given surveillance area. The mobile anchors in these schemes are autonomous vehicles which either follows a predefined path or a dynamic trajectory according to the environment. The quality of anchor positions are significant for high localization accuracy. Thus, the movement trajectories of mobile anchors are necessary to satisfy these properties: (i) Created anchor nodes should be close to as many sensors as possible for full localization; (ii) At least three non-collinear anchors should be in the neighborhood of each sensor in 2D to avoid position ambiguities; (iii) The path should be as short as possible for energy consumption [5]. Path planning algorithms are classified as static and dynamic algorithms in the following sections.

2.4.5 Static Path Planning

The trajectory of static path planning is determined prior to the localization and the mobile anchor cannot change its trajectory during the procedure. Besides, static algorithms are designed based on a specific node localization method and they are not flexible for modifications. Commonly used path planning algorithms such as SCAN, DOUBLE-SCAN, CIRCLES, SPIRAL and LMAT are discussed in the following sections.

2.4.5.1 SCAN and DOUBLE-SCAN Algorithms

SCAN trajectory is based on one direction movement of the mobile anchor and aims at full coverage of the surveillance region. The mobile anchor simply traverses the area along the y-axis as in Figure 2.6a. The resolution of the path is decided by distances between the parallel lines [8].

In [41], sensor positions are found by chord selection. The perpendicular bisector of a circle chord passes through the center of this circle. Hence, two chords selected from the same circle provide with the exact position of the circle center. According to this localization scheme in [41], the mobile anchor path consists of parallel lines spaced with $(R - \phi)$ intervals where R is the communication radius of each sensor and ϕ is the resolution parameter. ϕ is set to be in the range $0 < \phi < \frac{R}{3}$ such that the highest possible chord length is achieved for precise localization.

In [8], sensor positions are determined by sensor-to-anchor distances estimated from the probability distribution functions of Received Signal Strength Indicator (RSSI) values. The main disadvantage of SCAN algorithm is stated to be the collinearity of anchor positions. Since the movement trajectory is linear, consecutive anchors which are likely to be in the communication range of the same sensor are collinear.

DOUBLE-SCAN trajectories aim at solving the collinearity problem faced in SCAN algorithms. The mobile anchor first moves along y-axis similar to SCAN and sensor positions are found with ambiguity in x coordinates. Then the mobile anchor moves along the x-axis to eliminate the ambiguities as in Figure 2.6b. However, to achieve the same resolution as in the SCAN trajectory, the distance traveled is doubled compared to the SCAN algorithm. [8]

2.4.5.2 LMAT Algorithm

Localization algorithm with a Mobile Anchor node based on Trilateration (LMAT) is a static path planning algorithm proposed by Han et al. in [11]. Following a path similar to SCAN in x direction, the mobile anchor zigzags in y axis as shown in Figure 2.6c. The anchors are generated such that equilateral triangles are formed for tri-

literation based localization. This method is proposed to eliminate anchor collinearity which results from straight trajectories. Moreover, distances between horizontal segments are chosen such that the coverage of entire localization area is guaranteed. That is, the circular communication ranges of anchors intersect to localize all sensors deployed in the region. On the other hand, LMAT trajectory contains high number of corners which leads to high energy consumption for the mobile anchor.

In [12], Han et al. propose SLMAT trajectory. Here, the purpose is to minimize the energy consumption resulting from the high number of corners in LMAT. SLMAT is a combination of SCAN and LMAT algorithms. That is, the mobile anchor in SLMAT follows the SCAN trajectory, but the anchors are generated in LMAT positions to form equilateral triangles. Although SLMAT preserves energy, the localization performances converge to the one in LMAT for various resolutions.

In [13], Z-curve trajectory is proposed for static path planning. Similar to LMAT, the purpose is to generate as many non-collinear anchors as needed for accurate localization. The mobile anchor traverses the surveillance region in Z shape by determining the Z-curve sizes such that the maximum coverage is achieved. In [13], Z-curve shows a close localization performance to LMAT and CIRCLES.

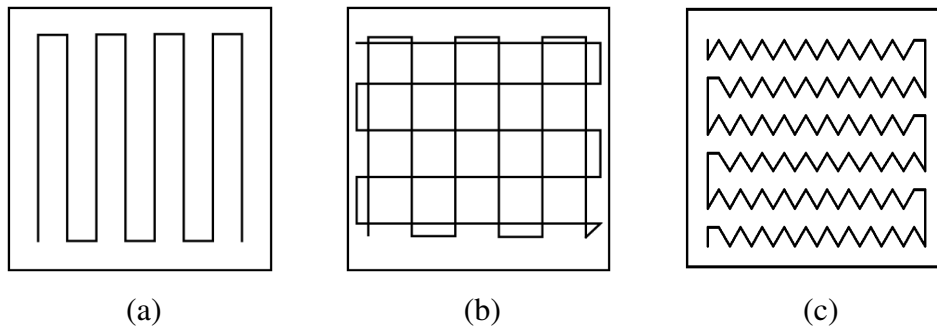


Figure 2.6: Movement trajectories of (a) SCAN, (b) DOUBLE-SCAN and (c) LMAT algorithms taken from [5]

2.4.5.3 CIRCLES and SPIRAL Algorithms

CIRCLES trajectories are composed of sequences of concentric circles which are centered at the surveillance region as in Figure 2.7a. This method is proposed by Huang et al. in [9] to solve the collinearity issue. While the straight lines in SCAN

and DOUBLE-SCAN schemes create collinear anchors, curved path in CIRCLES trajectories provide non-collinear anchor nodes.

In [9], the mobile anchor starts following the innermost circle with radius $2R$. When it finishes its move here, the mobile anchor follows the outer circle trajectory by increasing the radius by R at each outer circle. The resolution for CIRCLES scheme is defined to be half of the innermost circle radius, R . According to [9], distance that the mobile anchor moves is shorter in CIRCLES than SCAN. Besides, the proposed trajectory is scalable when the size of surveillance area changes. However, as the scale increases, curved path becomes more straight and non-collinearity is reduced [9].

In the CIRCLES trajectory, transition between concentric circles can be energy consuming due to sharp direction changes at transition points. SPIRAL trajectory proposed by Hu et al. in [10] has the same non-collinearity advantage as CIRCLES while it provides an energy efficient continuous movement. As in Figure 2.7b, the mobile anchor follows a spiral trajectory by creating anchors at equidistant points. In [10], anchor coordinates are expressed as,

$$\begin{aligned} x &= x_0 + D \times t \times \cos(2\pi t) \\ y &= y_0 + D \times t \times \sin(2\pi t) \end{aligned} \tag{2.13}$$

where x_0 and y_0 are the initial coordinates of the mobile anchor; D is the distance between two consecutive anchors and t is used to control the moving angle and the number of turns inside the surveillance area. Similar to CIRCLES, increasing the scale in SPIRAL reduces the non-collinearity. Furthermore, in both algorithms, the mobile anchor follows a shorter distance than SCAN trajectory [5]. On the other hand, CIRCLES and SPIRAL trajectories are unable to cover the whole surveillance region, if the area is not circular [9]. For full coverage, number of turns is increased which leads to higher path length.

2.4.6 Dynamic Path Planning

Static path planning provides a predetermined path and full coverage of the entire localization area. When the sensors are distributed uniformly and a fixed node local-

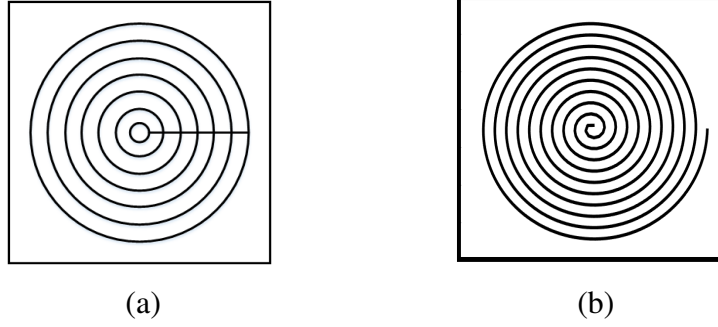


Figure 2.7: Movement trajectories of (a) CIRCLES and (b) SPIRAL algorithms taken from [5]

ization algorithm is used, static path planning algorithms work well. However, for non-uniform sensor distributions and non-ideal environments, static path planning results in longer path, higher number of anchors, longer localization time and higher energy consumption. Thus, dynamic path planning becomes more important for such environments. It provides a dynamic trajectory which is adjustable to the observable environments and sensor distributions during localization. [5]

In [15], a group of mobile anchors are used for dynamic path planning. An anchor is at the center of four anchors forming a '+' shape. The group of mobile anchors are connected to each other and moves together through the network. The center anchor controls the direction of group movement and finds the unknown node positions. The others in the group are called direction-anchors which collect acknowledgments from the unknown nodes to determine the direction. Making use of the received acknowledgment signals by each direction-anchor, the path is determined dynamically. As in the simulation results shown for L-shaped and U-shaped sensor distributions, the algorithm proposed in [15] decides the direction of the mobile anchor group and only the necessary regions are visited. Hence, this approach provides an efficient path for nonuniform unknown node distributions over the network.

In [6], a dynamic path planning algorithm is proposed. It is mainly based on Gauss-Markov (GM) mobility model and virtual forces. Velocity adjustment, perpendicular bisector and virtual repulsive strategies are applied to GM mobility adaptively. The approach proposed in [6] can be abbreviated as GM algorithm. Here, the mobile anchor trajectory is dynamically modified according to the environment. That is, the

velocity and direction of the mobile anchor is adjusted depending on the received acknowledgments from the unknown sensors. In order to decrease the randomness in velocity and direction, the mobile anchor slows down in the communication range of a sensor and becomes faster in sensor-free area. In addition, when the mobile anchor enters the communication range of an already localized sensor, it is repelled out of the range by a virtual force. Similarly, if the mobile anchor goes away from the localization area, a virtual force is applied from the mobile anchor towards the region. Furthermore, the localization is done by creating the anchors inside the range of a sensor according to perpendicular bisection. The adaptive strategies provide the mobile anchor with control over the velocity and direction. This results in short path, less anchors and energy consumption.

In [7], the performance of the GM algorithm proposed in [6] is improved by introducing a virtual attractive force. The algorithm proposed in [7] can be abbreviated as GM-VAF. While GM provides control over the mobile anchor velocity in sparsely populated areas where the sensor population is low, it does not control the direction of the mobile anchor. GM-VAF algorithm provides a control over the direction in both sparsely and densely populated areas. In [7], the surveillance region is divided into grids and represented by a matrix H . When the mobile anchor generates an anchor node at a grid, the corresponding index of the matrix is increased by one. In sparsely populated regions, if the mobile anchor visits a sensor free grid more than a predefined time, it is attracted by the closest grid with minimum $H_{i,j}$. This enables the mobile anchor to visit unvisited areas and provides full coverage. In densely populated regions where large number of sensors are distributed, if the mobile anchor revisits the communication range of a localized sensor, it is attracted by the closest grid with minimum $H_{i,j}$. Hence, instead of spending time around localized sensors, the mobile anchor visits unvisited grids. GM-VAF controls the direction of the mobile anchor in the given cases and it decreases the randomness of the velocity and the direction in GM algorithm.

CHAPTER 3

A NOVEL STATIC PATH PLANNING ALGORITHM FOR MOBILE-ANCHOR BASED WIRELESS SENSOR NETWORK LOCALIZATION

Static path planning is used for mobile anchor based wireless sensor network localization. An autonomous vehicle is considered as a mobile anchor and travels a predefined path. As mentioned in chapter 2.4.5, static path planning trajectories are determined based on specific node positioning techniques. When the localization method changes, the static path planning trajectory requires to be adjusted for the specific method. The number and position of anchors required for high localization accuracy can be determined using the approach proposed in [42]. Optimizing the SNR of range measurements, best anchor configuration can be determined as stated in [42]. In this chapter, a novel static path planning algorithm is proposed. The mobile anchor positions are determined based on trilateration technique which is mentioned in chapter 2.3.1. The trajectory is designed such that all the sensor nodes distributed in the localization area have at least three neighboring anchors in their communication ranges.

In trilateration, an unknown node position, \mathbf{x}_i , can be found as the intersection point of the circles whose centers are the anchor positions, \mathbf{a}_l , and the radii are anchor-to-sensor distances, d_{li} . When there is noise added to d_{li} 's, the solution for the unknown position is not a single point but a region. As Theorem 1 in [11] states, the error region is minimum when the anchor positions are symmetric to the center point of the error region. For precise localization in 2D, anchors are created in the form of equilateral triangles as shown in Figure 3.1a. In Figure 3.1a, the blue dots represent the anchor

nodes generated by the mobile anchor and the green rectangle is the surveillance region where the localization is carried out. The proposed configuration satisfies the optimum anchor placement relations proposed in [42]. The circles in Figure 3.1b represent the communication range of the anchor nodes. According to Theorem 3 in [11], when the side length of the triangles is R , the best sensor coverage and localization accuracy are obtained. In the proposed static path planning algorithm, the mobile anchor trajectory and anchor node positions are determined based on these theorems. The proposed anchor configuration results in high region coverage and localization accuracy.

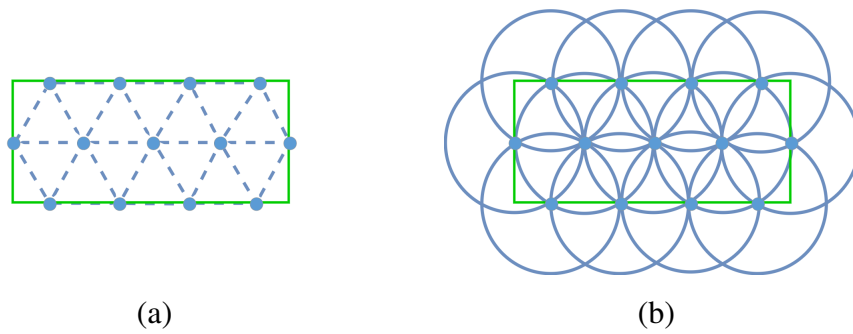


Figure 3.1: Configuration of (a) anchor nodes in the form of equilateral triangles and (b) circular communication ranges over the surveillance region.

The static path planning approach proposed in this thesis is composed of two phases, namely search and decision phases. Therefore the proposed method is called static search-and-decide (SSD) algorithm throughout the remaining chapters. In the first phase, an initial search is performed over the surveillance region to identify the best spots for the mobile anchor in decision-phase. After this search, SSD algorithm is finished and the best anchor positions are identified. Then, the mobile anchor starts its movement along the surveillance region. Since SSD is a static path planning algorithm, the mobile anchor starts the localization procedure after both phases are carried out.

3.1 SSD Search Phase

The role of this part in the path planning procedure is the detection of sensors distributed over the surveillance region. In the search-phase of SSD approach, the mobile

anchor traverses the localization area and collects information about the existence of sensors. The search trajectory is made of straight paths in one direction similar to the SCAN algorithm in [8]. Broadcasting its coordinate at each stop, the mobile anchor creates anchor nodes along the localization region. Then, it collects acknowledgment signals from the neighboring sensors which receive the position information of the mobile anchor.

Let the surveillance region be an $L_1 \times L_2$ rectangle represented with green lines in Figure 3.2a. The dark blue line with arrows is the path followed by the mobile anchor in the search-phase, blue dots are anchor nodes and blue circles represent the communication range of these anchors. The anchor nodes generated in this phase, namely the search-phase anchors, are separated by the radius R of the circular communication range in x direction and by $\sqrt{3}R$ in y direction. As in Figure 3.2b, the position vectors of search-phase anchors are represented by $\mathbf{a}_{i,j}$, where $i = 1, 2, \dots, m_1$ and $j = 1, 2, \dots, m_2$. The starting and ending points of the mobile anchor are $\mathbf{a}_{m_1,1}$ and \mathbf{a}_{1,m_2} , respectively. However, the starting and the ending position of the mobile anchor trajectory may change depending on the size of the surveillance region, the communication range radius or the mobile anchor direction while starting its movement.

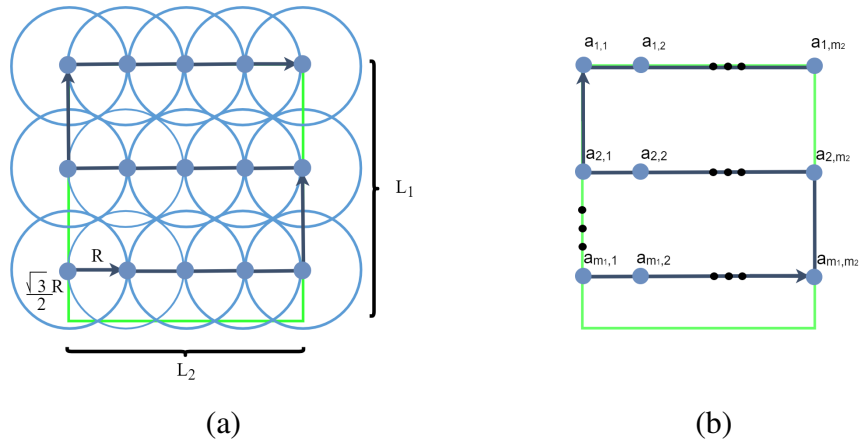


Figure 3.2: Search-phase (a) mobile anchor trajectory and (b) anchor positions for the proposed static path planning algorithm SSD.

The total number of anchor nodes, M_{search}^{SSD} , generated by the mobile anchor in the

search-phase of the proposed SSD algorithm is given as,

$$m_1 = \left\lceil \frac{L_1}{\sqrt{3}R} \right\rceil \text{ and } m_2 = \left\lceil \frac{L_2}{R} \right\rceil \quad (3.1)$$

$$M_{search}^{SSD} = m_1 \times m_2 \quad (3.2)$$

where L_2 and L_1 are the side lengths of surveillance region in x and y directions, respectively; m_1 and m_2 are integers representing the minimum number of anchors by Ceil operators which map an input to the smallest integer greater than or equal to this input.

Consider the case which the mobile anchor starts the search-phase of SSD from the coordinates $(0, \frac{\sqrt{3}}{2}R)$ and follows the path shown in Figure 3.2 for m_1 is odd. The search-phase anchor positions are found as,

$$\mathbf{a}_{i,j} = ((j-1)R, (m_1-i)\sqrt{3}R + \frac{\sqrt{3}R}{2}) \quad (3.3)$$

where $i = m_1, m_1 - 1, \dots, 2, 1$ which is in a decreasing manner due to the direction of movement of the mobile anchor and $j = 1, 2, \dots, m_2$.

When the entire region is traversed, an $m_1 \times m_2$ storage matrix, \mathbf{W} , is introduced. Indexes of \mathbf{W} represent the search-phase anchor positions and the elements at these indexes contain the number of acknowledgment signals received by the corresponding anchor, i.e.,

$$\mathbf{W} = \begin{pmatrix} n_{1,1} & n_{1,2} & \dots & n_{1,m_2} \\ n_{2,1} & n_{2,2} & \dots & n_{2,m_2} \\ \vdots & & & \\ n_{m_1,1} & n_{m_1,2} & \dots & n_{m_1,m_2} \end{pmatrix} \quad (3.4)$$

In the expression (3.4), $n_{i,j}$ is the number of acknowledgment signals received by the corresponding anchor located at $\mathbf{a}_{i,j}$ as in Figure 3.2b. The acknowledgment signals coming from unknown sensors contain the sensor identification numbers (INs).

Hence, in addition to the number of sensors located in the circular grids, the identifications of the sensors occupying each index of \mathbf{W} are known by the user.

The first search movement of the mobile anchor makes use of the intersecting communication ranges of the search-phase anchors and provide detection of the sensors in the surveillance region. However, the number of these anchor nodes are not enough for localization since at least three neighboring anchor positions are required for localization. Hence, the decision-phase is introduced to determine the required anchor positions and mobile anchor trajectory for localization of all sensors by forming equilateral triangles. After determining the total trajectory, the mobile anchor is informed about the path to be followed and the anchor positions to be visited.

3.2 SSD Decision Phase

This part of the static path planning contains the decision of returning trajectory and anchor generation. Decision-phase begins after the mobile anchor searches the surveillance region and \mathbf{W} is formed according to (3.4). In this phase, intersections of the circular communication ranges are checked to determine where the sensors are located. The red lines and dots, $\mathbf{b}_{i,j}$, in Figure 3.3a are the trajectory and potential decision-phase anchors, respectively. The real decision-phase anchors are subsets of the anchor positions shown in Figure 3.3a. The purpose of the decision-phase is to determine the best $\mathbf{b}_{i,j}$ positions to visit. The shaded areas in Figure 3.3b are the intersection zones. Adjacent $n_{i,j}$'s are checked to determine which shaded zones are potentially occupied by which sensors. Then, the equilateral triangles are formed by creating anchors at red dotted intersection points to localize the unknown sensors. If there is at least one sensor in all circles, the overall SSD trajectory is in Figure 3.3b. Otherwise, redundant circles and the corresponding positions with red dots are skipped.

Decision-phase of the proposed SSD algorithm starts with the sensor detection in the intersecting circle zones. Let us call the decision-phase anchors $\mathbf{b}_{i,j}$ whose starting point may vary depending on where the search-phase anchors start and m_1 's parity(evenness) as in Figure 3.4. To decide if an anchor should be created at $\mathbf{b}_{i,j}$, the

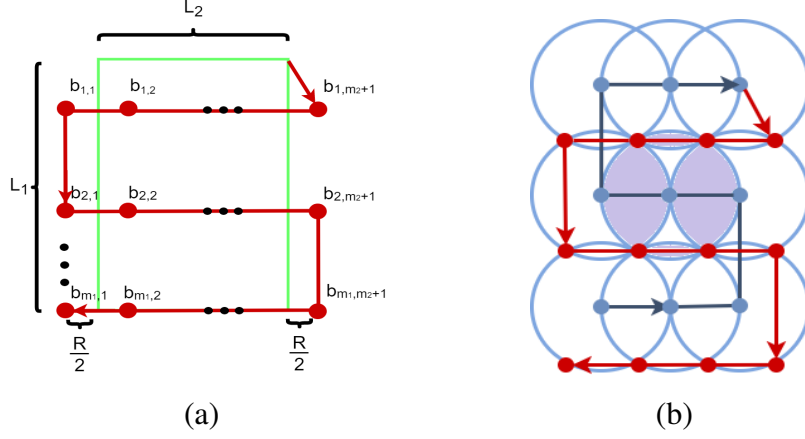


Figure 3.3: (a) The potential decision-phase trajectory and (b) the potential total mobile anchor trajectory for the proposed static path planning algorithm.

corresponding indexes of the upper and lower intersecting circles are checked. If both $n_{i,j}$ and $n_{i+1,j}$ are zero, no anchor is planned to be created here and the position is skipped. When a nonzero index is found, the circles surrounding the occupied circle are checked if they have common sensors with the same INs. In this way, which sensors are located in which shaded zones in Figure 3.3b are determined. The unshaded zones are assumed to be occupied if the number of detected sensor INs are less than the corresponding $n_{i,j}$. After checking all the circular grids and determining the required decision-phase anchor positions, the total trajectory is planned and the mobile anchor starts its move to follow the determined path. The number of generated anchors is $M_{decision}^{SSD}$ in the decision-phase.

The total static search-and-decide procedure for $m_1 = odd$ is given in Algorithm 3.1. For $m_1 = even$ case, the mobile anchor starts its move at $(0, \frac{\sqrt{3}R}{2})$ and finishes at $(0, L_1)$ as in Figure 3.4b. The application of the proposed method for $m_1 = even$ case is straightforward.

Algorithm 3.1: Static Search-and-Decide (SSD)

Let m_1 be odd, and \mathbf{W} be the matrix with elements $n_{i,j}$'s which are the number of sensors in the corresponding grid. The direction of movement (DoM) is shown by the arrows in Figure 3.4. The sensor IN's are known together with $n_{i,j}$'s to know which intersection zones are occupied by which sensors.

Step 1: Traverse the surveillance area as in Figure 3.2a and generate anchors at $a_{i,j}$'s according to (3.3). Create \mathbf{W} from the received acknowledgments in the search-phase.

Step 2: Start the decision-phase at $b_{i,j+1}$ where $j = m_2$. Check if $n_{i,j}$ and $n_{i+1,j}$ are both zero. If so, skip this position. Otherwise, plan to create an anchor at $b_{i,j+1}$.

Step 3: If $j = 1$, go to step 9. Otherwise, check $n_{i,j}$ and $n_{i+1,j}$ in DoM for $m_2 - 1 \geq j \geq 2$, when i is fixed. If both $n_{i,j}$ and $n_{i+1,j}$ are zero, skip $b_{i,j+1}$ position.

Step 4: If $n_{i,j} = 0$ and $n_{i+1,j}$ is nonzero, go to step 5. If $n_{i,j}$ is nonzero and $n_{i+1,j} = 0$, go to step 6. If $n_{i,j}$ and $n_{i+1,j}$ are both nonzero, go to step 7.

Step 5: If $(i + 1, j)^{th}$ circle and the next circle in DoM, which is $(i + 1, j - 1)$ for i is odd, are occupied by the same sensors, skip $b_{i,j+1}$ position and go to step 3. Otherwise go to step 8.

Step 6: If $(i, j)^{th}$ circle and its next circle in DoM, which is $(i, j - 1)$ for i is odd, are occupied by the same sensors, skip $b_{i,j+1}$ position and go to 3. Otherwise go to 8.

Step 7: If both 5 and 6 are valid skip $b_{i,j+1}$ position and go to 3. Otherwise go to 8.

Step 8: Plan to create an anchor at $b_{i,j+1}$ and go to step 3.

Step 9: If $n_{i,j}$ and $n_{i+1,j}$ are both zero, skip $b_{i,j+1}$ and $b_{i,j}$ positions. Otherwise plan to generate anchors at $b_{i,j+1}$ and $b_{i,j}$.

Step 10: Increase i by 1 and go to step 3.

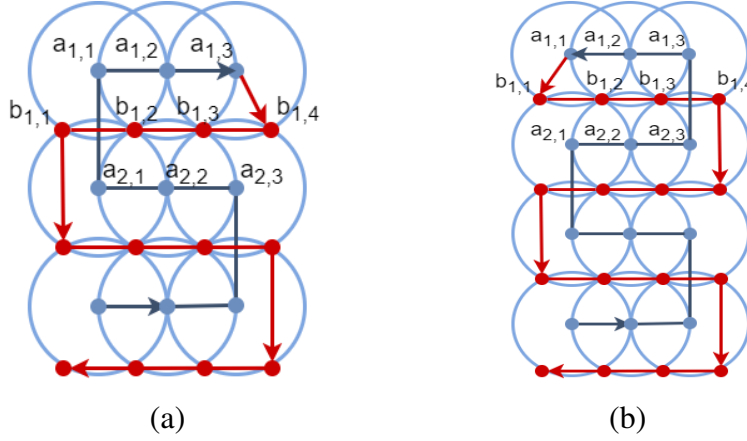


Figure 3.4: Examples of the mobile anchor trajectories for $m_2 = 3$, when m_1 is (a) odd and (b) even.

The proposed SSD algorithm is a static path planning method, since the mobile anchor does not change its position dynamically. When the search-phase is finished, the decision-phase trajectory is determined. Then the mobile anchor starts to move accordingly, without changing its planned trajectory.

3.3 Simulation Results for SSD

In the simulation results for SSD, the residue unknowns stand for the number of remaining unknown sensor positions. N is the number of sensors which are randomly deployed in the localization area with various distributions, namely uniform, Gaussian, exponential and Gaussian mixture distributions as in Figure 3.5. The Gaussian mixture distribution is generated by combining two Gaussian random distributions with different mean and variances. The given distributions can be considered for the scenarios where the sensors are placed on the terrain by a vehicle uniformly, released from a helicopter in the form of Gaussian or exponential distribution and they are released by two different sources in the form of Gaussian mixture distributions.

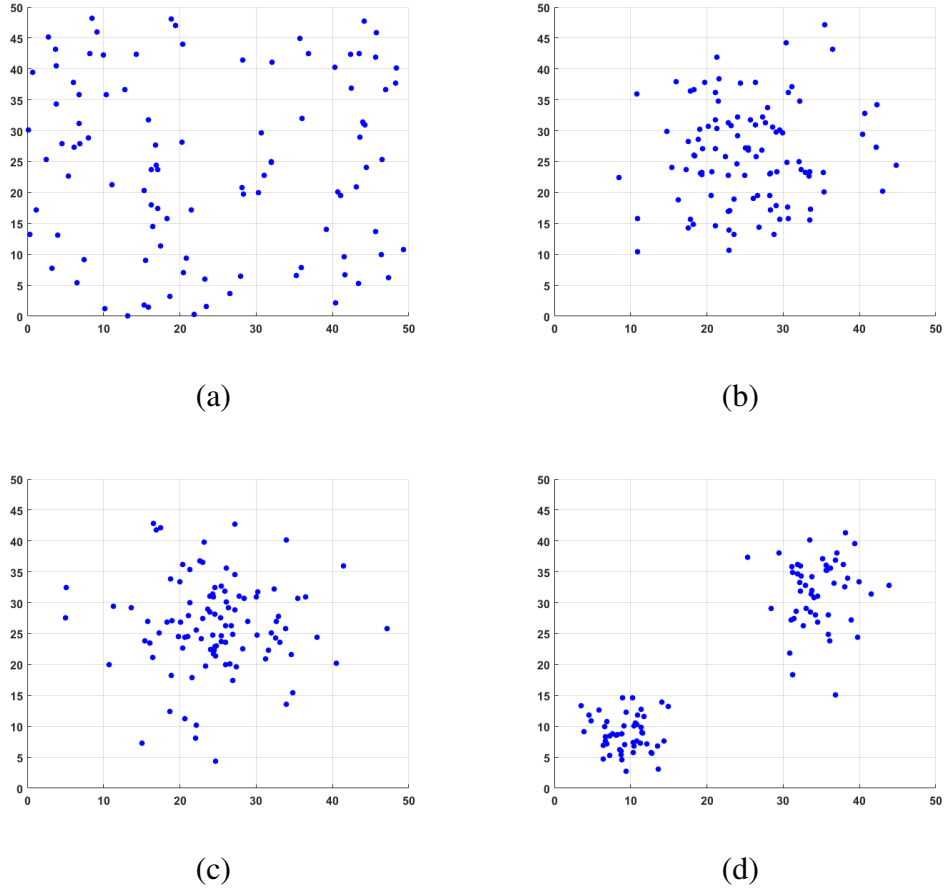


Figure 3.5: Example for the positions of sensors deployed in $50m \times 50m$ surveillance region in the form of (a) uniform, (b) Gaussian, (c) exponential and (d) Gaussian mixture distributions.

In this scenario, SSD is compared with single-mobile anchor based static algorithms, namely CIRCLES [9], LMAT (Localization Algorithm with a Mobile Anchor node based on Trilateration) [11] and SLMAT [12].

In [13], Rezazadeh et. al. propose Z-curve method and evaluate the performances of SCAN, HILBERT, LMAT, CIRCLES and RWP which are considered to be benchmark for the static path planning scenarios. As stated in [13], LMAT and CIRCLES algorithms have better percentage of localization success compared to the rest. Moreover, CIRCLES provides a shorter localization time than the others. On the other hand, LMAT has the same coverage ratio and has a very close localization accuracy as Z-curve trajectory.

In [12], the localization performance of SLMAT is very close to LMAT, since both generate anchors at the same positions. However, SLMAT creates less corners and consumes less power than LMAT trajectory because it follows a different path. That is, LMAT turns a corner at each move in its zigzag trajectory but SLMAT takes corners only at the end of its each horizontal move. Since SSD follows a trajectory similar to SCAN, its corner number is the same as SLMAT in the worst case. Considering the results in [13] and [12], our proposed method SSD is compared with LMAT, CIRCLES and SLMAT algorithms for validation.

The simulation parameters used in the path planning algorithms for the static case are given in Table 3.1.

Table 3.1: Simulation Parameters of the Static Scenario.

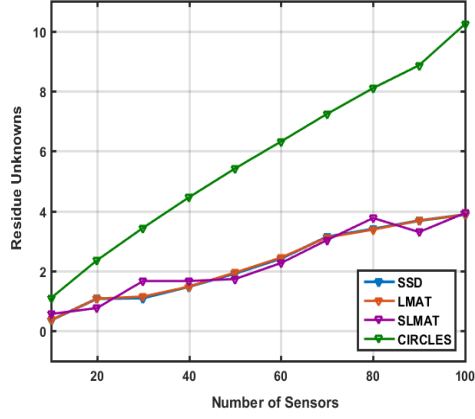
Parameter	Value
Localization Region Size	50m \times 50m
Communication Range (R)	5m
Sensor Distribution	Uniform, Gaussian, Exponential, Gaussian Mixture
Number of Sensors (N)	10,20,30,40,50,60,70,80,90,100
Number of Mobile Anchor	1
Velocity of Mobile Anchor (m/s)	1
Anchor Threshold for Localization	3
Monte Carlo Runs	100

In Figure 3.6, uniform sensor distribution is used to identify residue unknowns, number of anchors and distance with increasing number of sensors (N) for SSD, LMAT, CIRCLES and SLMAT. The proposed static algorithm has less residue unknowns than CIRCLES and the same as LMAT and SLMAT for the uniform distribution in a noise free environment as shown in Figure 3.6a. The reason behind the overlap for SSD, LMAT and SLMAT is the similarity of their region coverage. However, there are certain differences in their techniques as well. SSD does a first pass to the sensor grids for detection. Then only the neighborhoods of the detected sensor regions are revisited in the second pass. On the contrary, LMAT does use a single zigzag pass strategy where a densely populated grid is visited at once. Moreover, SLMAT also use a single pass strategy following straight lines as in SCAN trajectory. Figure 3.6a shows that the CIRCLES algorithm localizes less sensors than SSD, LMAT and SLMAT for the

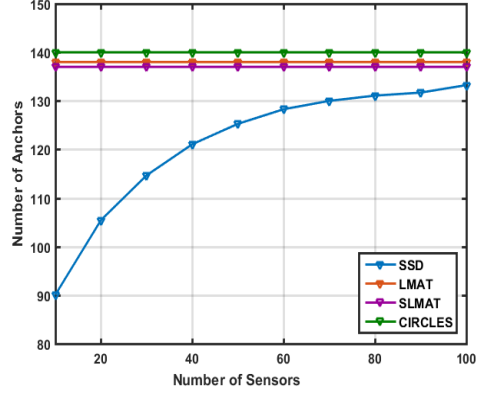
uniform distribution. The performance difference between CIRCLES and the rest is distinct. The reason is that CIRCLES trajectory cannot reach the corners of the localization region when the sensors are uniformly distributed over a rectangular area. On the other hand, this performance gap is insignificant in Figure 3.7a and Figure 3.8a. Since the sensors gather at the center for Gaussian and exponential distributions, almost all algorithms provide full coverage. In addition, CIRCLES performance for the Gaussian mixture distribution in Figure 3.9a is also distinctive. This is because two Gaussian distributions with different mean values are combined and sensors become closer to the corners in this distribution.

As indicated in Figure 3.6b, SSD uses far less anchor nodes to achieve this localization success compared to CIRCLES, LMAT and SLMAT. The number of anchors in SSD stabilizes at a lower level than the rest. If the anchor number is decreased for LMAT, CIRCLES and SLMAT to reach the same performance as SSD, their residue unknowns increase. As a result, SSD uses fewer anchors than LMAT, CIRCLES and SLMAT to localize the same amount of sensors as LMAT and SLMAT and more sensors than CIRCLES. On the other hand, SSD requires less anchors in Gaussian, exponential and Gaussian mixture distributions as in Figures 3.7b, 3.8b and 3.9b, respectively. This is because the sensors are positioned closely in those distributions and less number of anchors are enough for localizing these sensors.

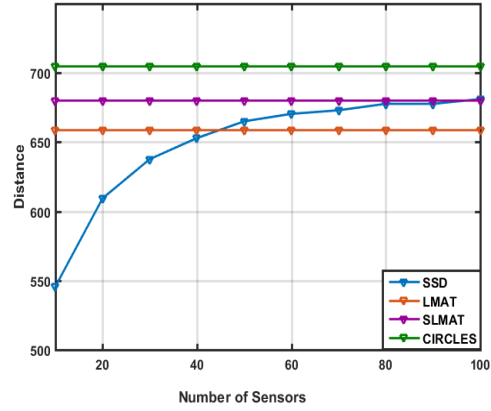
In Figure 3.6c, the distances that should be covered by the mobile anchor is given for the uniform distribution when there is no noise. In the worst case where a sensor is detected at each grid point of the first pass, SSD should populate other grid points to cover the same positions as LMAT which increases the distance traveled by the mobile anchor for SSD. This usually happens for uniform sensor distributions. As in Figure 3.6c, SSD is better for small N but the distance increases as N increases. However, for other sensor distributions in Figures 3.7c, 3.8c and 3.9c, SSD is significantly better than the alternative methods due to non-uniform gathering of sensors.



(a)

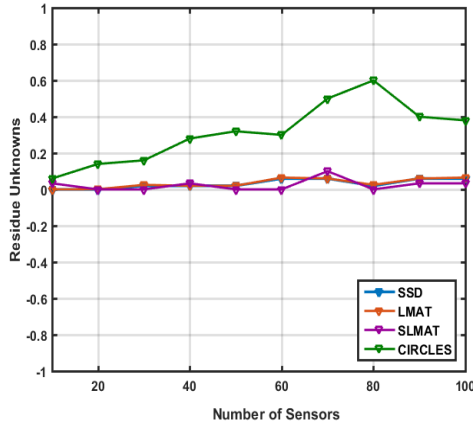


(b)

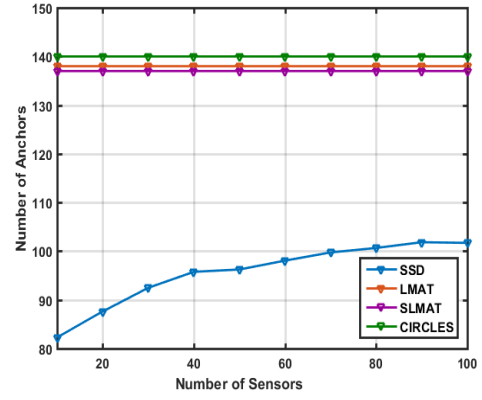


(c)

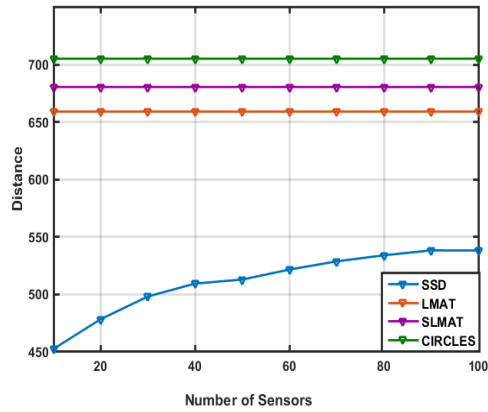
Figure 3.6: Static path planning results for (a) the number of residue unknowns, (b) the number of anchors and (c) distance traversed by the MA versus the number of sensors for the uniform sensor distribution in the noise free case.



(a)

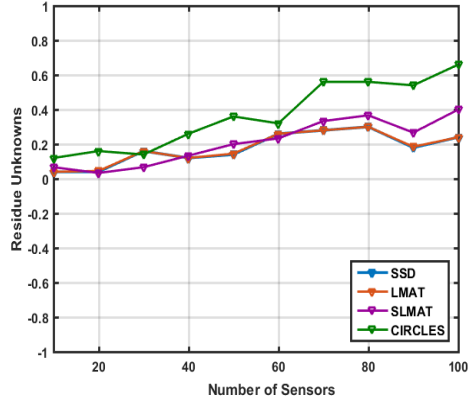


(b)

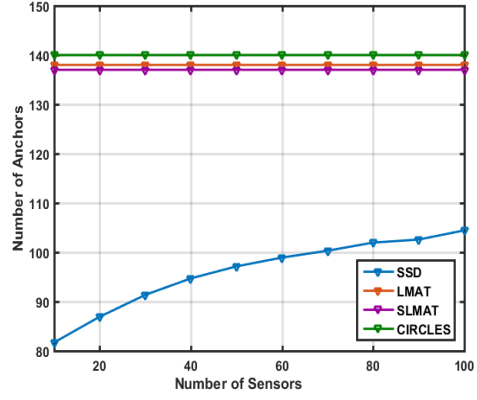


(c)

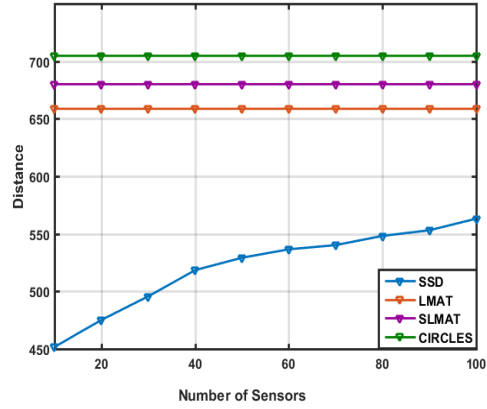
Figure 3.7: Static path planning results for (a) the number of residue unknowns, (b) the number of anchors and (c) distance traversed by the MA versus the number of sensors for the Gaussian sensor distribution in the noise free case.



(a)

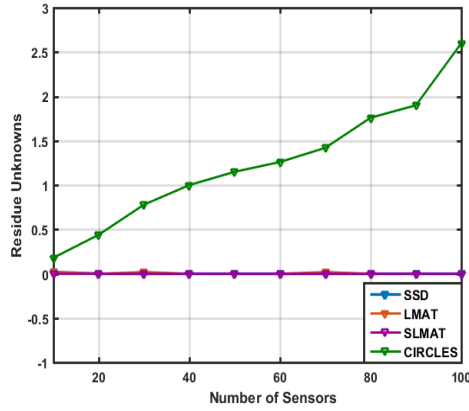


(b)

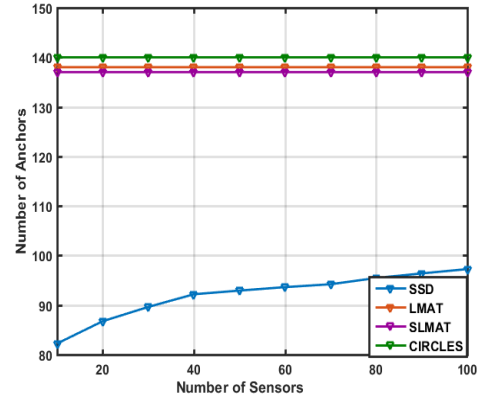


(c)

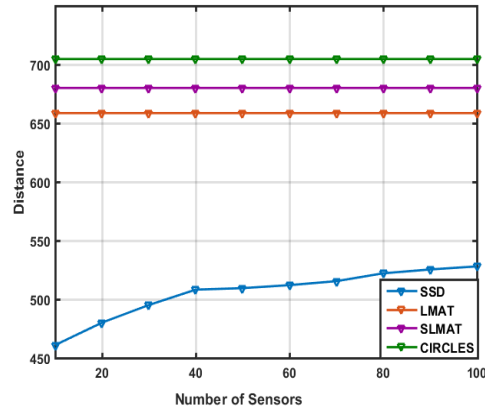
Figure 3.8: Static path planning results for (a) the number of residue unknowns, (b) the number of anchors and (c) distance traversed by the MA versus the number of sensors for the exponential sensor distribution in the noise free case.



(a)



(b)



(c)

Figure 3.9: Static path planning results for (a) the number of residue unknowns, (b) the number of anchors and (c) distance traversed by the MA versus the number of sensors for the Gaussian mixture sensor distribution in the noise free case.

In this chapter, a static path planning trajectory based on trilateration is proposed. The algorithm is composed of two phases. First the surveillance region is searched and the sensors are detected, then the anchor nodes are generated for localization based on trilateration. When it is compared with alternative approaches, the simulation results show that SSD achieves the same performance with trilateration based methods using less anchors and following a shorter path. SSD also performs better than the alternative static approaches.

CHAPTER 4

A NOVEL DYNAMIC PATH PLANNING ALGORITHM FOR MOBILE-ANCHOR BASED WIRELESS SENSOR NETWORK LOCALIZATION

In this section, a dynamic path planning algorithm is proposed for a mobile anchor based WSN, namely dynamic search-and-decide (DSD) algorithm. The aim is to reach the maximum area coverage and localization accuracy, while keeping the path length minimum. Similar to SSD, DSD has search and decision phases. After the mobile anchor traverses the localization area, it detects the sensors in circular grids and passes to the second phase. In this phase, the mobile anchor determines its path dynamically and uses perpendicular bisector strategy (PBS) for high localization accuracy. Since the non-collinear anchor positions are required in localization, PBS is used to create non-collinear anchors.

DSD is a dynamic path planning algorithm, since the mobile anchor trajectory is determined according to the demand of the sensors in each circular communication range of search-phase anchors. That is, unlike SSD, the path can be modified during the second phase of DSD algorithm. While the mobile anchor starts to move after the optimum anchor positions are determined in SSD approach, the mobile anchor of DSD algorithm performs the localization during the second phase of the path planning procedure. This makes DSD flexible for the user demands which can be changed during the process. On the other hand, this flexibility in dynamic case requires higher processing capability for the mobile anchor compared to the static case.

4.1 DSD Search Phase

The mobile anchor trajectory and anchor positions of DSD search-phase are the same as in SSD search-phase. That is, the anchors are generated at $\mathbf{a}_{i,j}$'s according to (3.3) by following the path in Figure 3.2a. Hence, the number of anchors in this phase of DSD is found similar to SSD by,

$$M_{search}^{DSD} = m_1 \times m_2 \quad (4.1)$$

where m_1 and m_2 are calculated as in (3.1). The mobile anchor broadcasts its position $\mathbf{a}_{i,j}$, where $i = 1, 2, \dots, m_1$ and $j = 1, 2, \dots, m_2$ as in Figure 3.2b, at each stop and receives acknowledgment signals from the sensors. The binary data indicating whether a signal is received is recorded in an $m_1 \times m_2$ matrix \mathbf{Z} . Indexes of \mathbf{Z} stand for the search-phase anchors and contain 1s and 0s for received signals by the corresponding anchor. There is a difference between \mathbf{W} of SSD and \mathbf{Z} of DSD. While the elements of \mathbf{W} represent the number of acknowledgments received in the corresponding grids, the elements of \mathbf{Z} indicates if an acknowledgment is received or not.

$$Z_{i,j} = \begin{cases} 1 & \text{if the anchor at } \mathbf{a}_{i,j} \text{ gets acknowledgements} \\ 0 & \text{otherwise} \end{cases} \quad (4.2)$$

The purpose of the search-phase in DSD is to obtain the matrix \mathbf{Z} . It is required to detect the existence of the neighboring sensors to the search-phase anchors. Although, this phase provides full coverage of the surveillance region, the number of anchors are not enough to localize the sensors. Hence, DSD introduces the decision-phase for full localization.

4.2 DSD Decision Phase

In decision-phase of DSD, considering the matrix \mathbf{Z} , only the occupied anchor grids are targeted. The mobile anchor traverses the region first time and stops at \mathbf{a}_{1,m_2} .

It dynamically checks the anchor grids in the reverse direction and moves accordingly. If the corresponding index $Z_{i,j}$ is zero, the mobile anchor skips this grid in a straight path. Otherwise, the localization starts according to the perpendicular bisector strategy. The anchor generation in the decision-phase lasts until all occupied circles are visited and all sensors localized. The number of anchors generated in the decision-phase is $M_{decision}^{DSD}$.

While the perpendicular bisector strategy is known in the literature [6], a summary of this strategy is given below since DSD algorithm uses its steps.

4.2.1 Perpendicular Bisector Strategy

This strategy provides non-collinear anchor generation within the circular communication range of the search-phase anchors. Here, an integer ξ is the minimum number of neighboring anchors required by a sensor for localization. After $b_{i,j}^1$ which is the first created decision-phase anchor in $(i, j)^{th}$ grid is generated in the range of $a_{i,j}$, the next position is determined by the perpendicular bisector strategy as follows;

Step 1: A perpendicular bisector line with red dots in Figure 4.1a is drawn through the dashed line connecting $a_{i,j}$ and $b_{i,j}$.

Step 2: Two potential anchor positions shown by red circles in Figure 4.1a are found on the dotted line according to the distance formula in [6] by,

$$u^{t,t+1} = \delta \times u_{a,b}, \quad t = 1, 2, \dots, \xi \quad (4.3)$$

where $u^{t,t+1}$ is the distance between two consecutive decision-phase anchors within the range; $u_{a,b}$ is the distance between the search-phase anchor $a_{i,j}$ and the first decision-phase anchor $b_{i,j}^1$ in the $(i, j)^{th}$ grid; δ is a tuning parameter [9].

Step 3: Since the perpendicular bisector line is drawn between an anchor and the center point, two potential next anchor positions are symmetric to the dashed line in Figure 4.1a. Therefore, one of the two potential anchor positions are randomly chosen to be the next position $b_{i,j}^2$ inside the $(i, j)^{th}$ grid as indicated with the arrow trajectory in Figure 4.1a.

Step 4: A perpendicular bisector is drawn on the connecting line of two consecutive decision-phase anchors, $b_{i,j}^1$ and $b_{i,j}^2$ as in Figure 4.1b.

Step 5: Two potential anchor positions shown by red circles in Figure 4.1b are found on the dotted line according to the distance formula in [6] by,

$$u^{s,s+1} = \delta \times u^{s,s-1}, \quad s = 2, \dots, \xi \quad (4.4)$$

where $u^{s,s+1}$ and $u^{s,s-1}$ are the distances between consecutive decision-phase anchors within the range.

Step 6: Two potential anchor positions are checked and the one within the communication range of $a_{i,j}$ is chosen to be $b_{i,j}^3$ which is the 3^{rd} decision-phase anchor in $(i, j)^{th}$ grid as in Figure 4.1b. If both potential positions are in the range, one of them is randomly chosen since they both can be used for localization. This continues until the sensors within the range achieves the anchor threshold ξ and the mobile anchor moves to the next occupied grid.

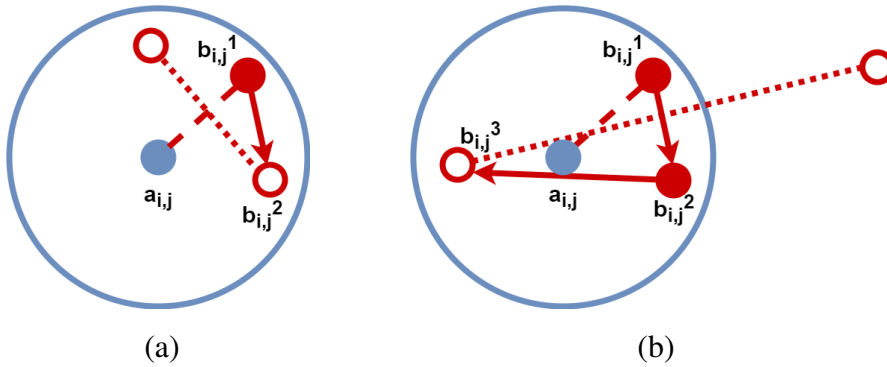


Figure 4.1: Perpendicular bisector strategy with (a) search-phase anchors and (b) decision-phase anchors for the decision-phase of DSD.

In the dynamic case, the proposed DSD algorithm searches the surveillance area for sensor detection and uses previously summarized perpendicular bisector strategy for sensor localization. DSD is applied for a sensor network as in Algorithm 4.1.

Algorithm 4.1: Dynamic Search-and-Decide (DSD)

Let \mathbf{Z} be the matrix with binary indexes $Z_{i,j}$ which represents whether the corresponding grid is occupied by a sensor.

Step 1: Traverse the surveillance region with R and $\sqrt{3}R$ intervals in x and y directions, respectively. Create \mathbf{Z} from the received acknowledgments in the search-phase.

Step 2: Check if the last visited search-phase anchor's circular grid is occupied. If Z_{1,m_2} is nonzero go to step 4.

Step 3: Check the search-phase anchors' grids in the backward manner until the first nonzero $Z_{i,j}$ is found and go to step 5. Otherwise, finish the path planning.

Step 4: Applying the steps 1, 2 and 3 of the perpendicular bisector strategy, generate anchors until the corresponding sensor has ξ number of neighboring anchors. Go to step 6.

Step 5: Applying the steps 4, 5 and 6 of the perpendicular bisector strategy, generate anchors until the corresponding sensor has ξ number of neighboring anchors. Go to step 6.

Step 6: Localize the corresponding sensors with ξ number of neighboring anchors using alternating minimization algorithm in Chapter 5. Go to step 3.

Example: A sensor network with a matrix \mathbf{Z} given in (4.5) and the settings of $m_1 = 2$, $m_2 = 3$ and $\xi = 4$ is considered as an example for the dynamic path planning.

$$\mathbf{Z} = \begin{pmatrix} 0 & 1 & 1 \\ 0 & 0 & 1 \end{pmatrix} \quad (4.5)$$

Applying the DSD algorithm, the dynamic trajectory in Figure 4.2 is obtained. The blue and red dots are the search-phase and the decision-phase anchors, respectively. While blue and red straight lines show the search and decision phase trajectories, red dotted lines indicate the perpendicular bisectors.

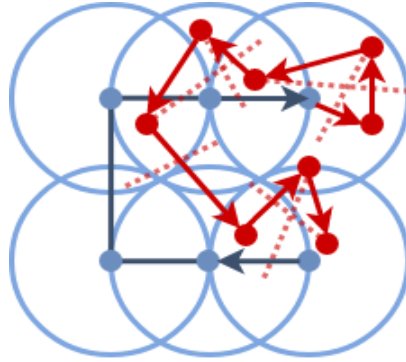


Figure 4.2: An example for the proposed dynamic path planning algorithm trajectory.

4.3 Simulation Results for DSD

In the simulation results for DSD, the residue unknowns stand for the number of remaining unknown sensor positions. N is the number of sensors which are randomly deployed in the localization area with various distributions shown in Figure 3.5. The Gaussian mixture distribution is generated by combining two Gaussian random distributions with different mean and variances.

In this scenario, DSD is compared with four dynamic algorithms. Basically, the performances of an adaptive and virtual repulsive force algorithm based on Gauss-Markov mobility model (GM) [6], a virtual attractive force algorithm based on Gauss-Markov mobility (GM-VAF) [7], maximum multi-hop of nodes (MMN) [15] and

maximum coverage of nodes (MCN) algorithms [16] are evaluated together with DSD.

The simulation parameters used in the path planning algorithms for the dynamic scheme are given in Table 4.1.

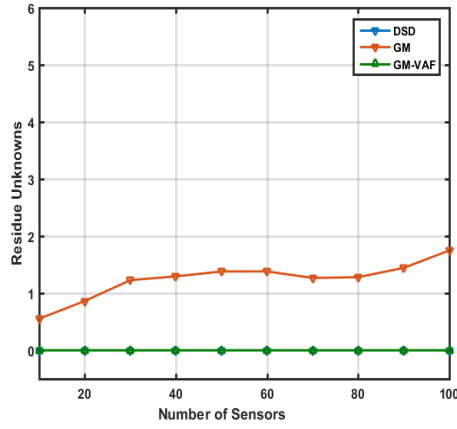
Table 4.1: Simulation Parameters of the Dynamic Scenario.

Parameter	Value
Localization Region Size	50m \times 50m
Communication Range (R)	5m
Sensor Distribution	Uniform, Gaussian, Exponential, Gaussian Mixture, L-Shaped, U-Shaped
Number of Sensors (N)	10,20,30,40,50,60,70,80,90,100 150, 200, 250,300
Number of Mobile Anchor	1, 5(for MMN and MCN)
Velocity of Mobile Anchor (m/s)	1
Anchor Threshold for Localization	3, 5
Monte Carlo Runs	100

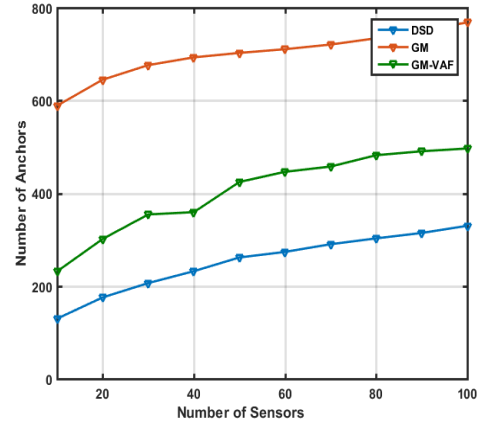
Residue unknowns, number of anchors and distance versus N for DSD, GM and GM-VAF are indicated in Figures 4.3, 4.4, 4.5 and 4.6 for the uniform, Gaussian, exponential and Gaussian mixture sensor distributions, respectively. The anchor threshold is set to 5 for the noise free case. Similar to the first phase of SSD, the mobile anchor searches the localization region to detect the existence of sensors in DSD. Then it revisits only the neighborhood of the detected sensors. Here, the mobile anchor generates anchor nodes exploiting the perpendicular bisector strategy to obtain high quality anchor coordinates. As in all the given sensor distributions in Figures 4.3a, 4.4a, 4.5a and 4.6a, DSD and GM-VAF both localize all the sensors deployed in the area. This is because DSD is designed to revisit all the surveillance region grids occupied by sensors and GM-VAF is designed to visit all the surveillance region grids. This provides full coverage of the sensors deployed in the localization area. Furthermore, less sensor positions remain unknown in DSD than GM for the uniform sensor distribution as in Figure 4.3a. DSD localizes more sensors than GM in general. The difference becomes more significant especially when N increases as in Figures 4.4a, 4.5a and 4.6a.

Due to the randomness in velocity and direction of the mobile anchor in GM algorithm, few of the created anchor nodes are useful. On the other hand, after detecting the existence of sensors, DSD creates anchor nodes only when necessary. Hence, as indicated in Figures 4.3b, 4.4b, 4.5b and 4.6b, DSD generates fewer anchors than GM to achieve the residue unknowns of the corresponding distributions. The effect of randomness in GM is diminished by introducing virtual attractive force in GM-VAF algorithm. Therefore, in most cases, GM-VAF generates less number of anchors than GM. However, the performance of DSD is still better than GM-VAF since the decisions made in DSD are not random.

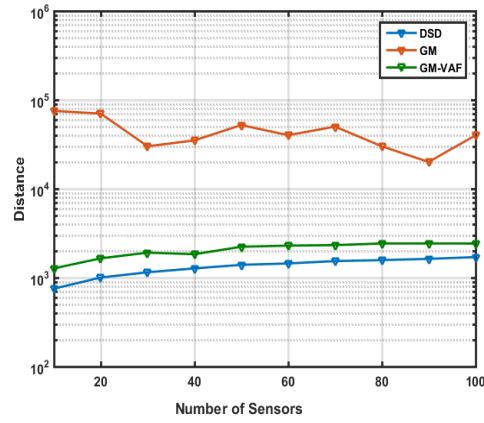
The distances covered by DSD, GM and GM-VAF are shown in Figures 4.3c, 4.4c, 4.5c and 4.6c as the number of sensors increases. Due to the randomness in GM, it takes a longer path to achieve the residue unknowns of the corresponding distributions for GM. Although the decreased randomness in GM-VAF improves the distance performance, it still uses a long path for localization. On the other hand, DSD creates anchors only inside the necessary communication ranges and skips redundant areas. Thus, DSD has a shorter path than GM and GM-VAF.



(a)

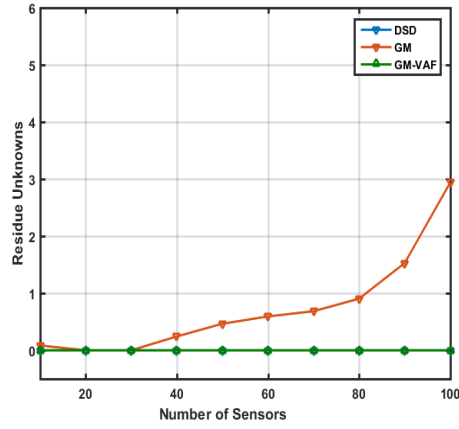


(b)

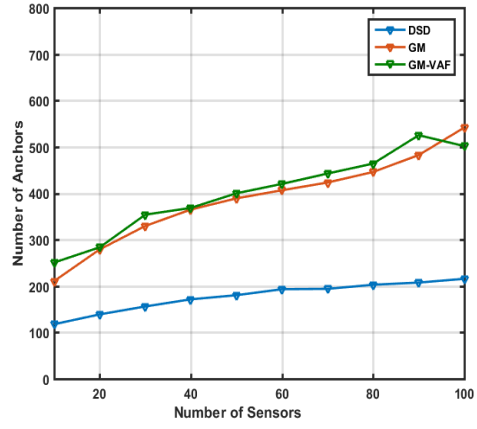


(c)

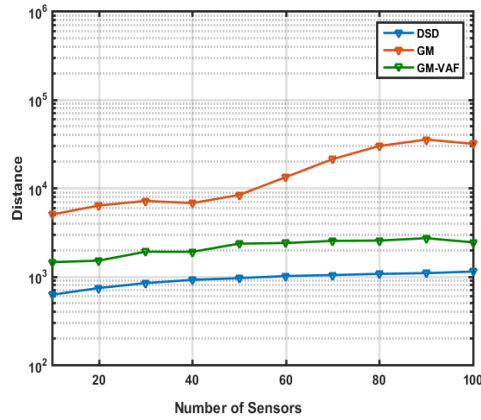
Figure 4.3: Dynamic path planning results for (a) the number of residue unknowns, (b) the number of anchors and (c) distance traversed by the MA versus the number of sensors for the uniform sensor distribution in the noise free case.



(a)

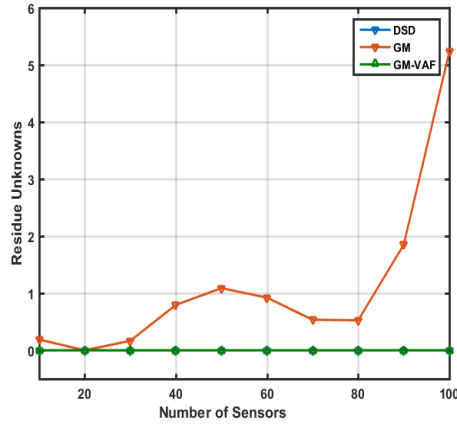


(b)

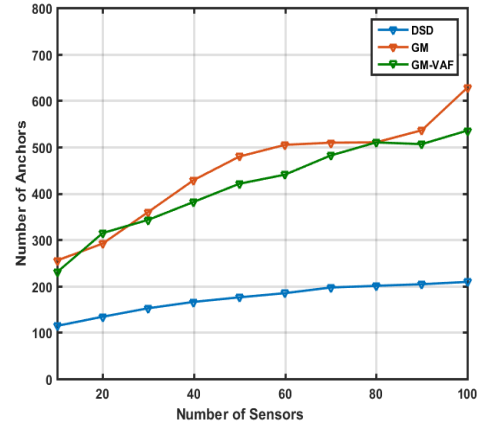


(c)

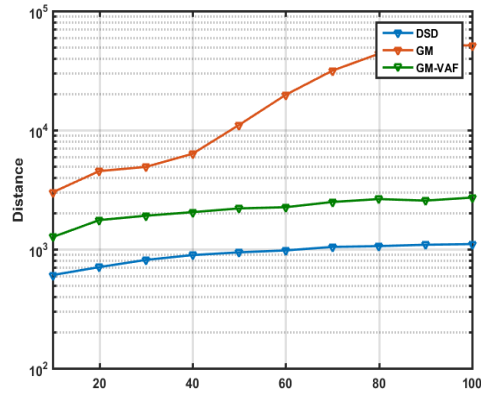
Figure 4.4: Dynamic path planning results for (a) the number of residue unknowns, (b) the number of anchors and (c) distance traversed by the MA versus the number of sensors for the Gaussian sensor distribution in the noise free case.



(a)

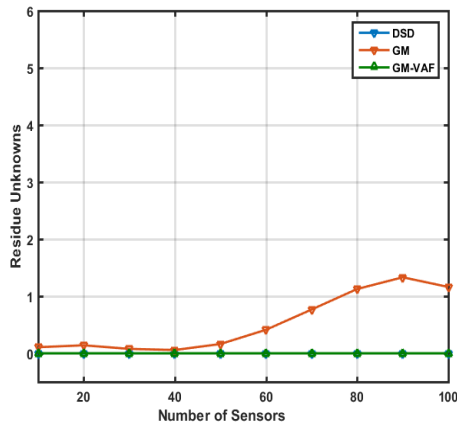


(b)

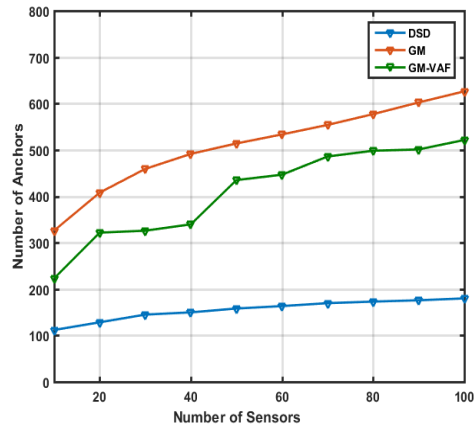


(c)

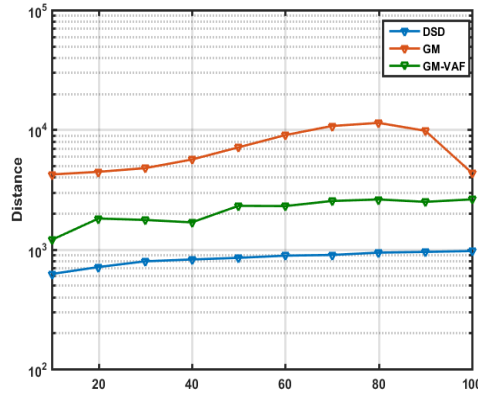
Figure 4.5: Dynamic path planning results for (a) the number of residue unknowns, (b) the number of anchors and (c) distance traversed by the MA versus the number of sensors for the exponential sensor distribution in the noise free case.



(a)



(b)



(c)

Figure 4.6: Dynamic path planning results for (a) the number of residue unknowns, (b) the number of anchors and (c) distance traversed by the MA versus the number of sensors for the Gaussian mixture sensor distribution in the noise free case.

The change in number of anchors and localized sensors rate are indicated with increasing number of sensors in Figures 4.7 and 4.8, respectively. The anchor threshold is set to be 3 and the sensor distribution is L-Shaped. Unlike in DSD, a group of 5 mobile anchors are used in MMN and MCN. That is, the mobile anchor group moves along the localization area and broadcasts its position periodically. Since 5 anchors are created at each hop, 5 times the hop count gives the total number of anchors for MMN and MCN. As a result of the same simulation, Figures 4.7 and 4.8 show that DSD localizes %100 of the sensors generating fewer anchors than MMN and MCN

as the number of sensors increase. Although DSD needs more anchors when there are fewer sensors, localization rate of DSD is greater than MMN and MCN in this case.

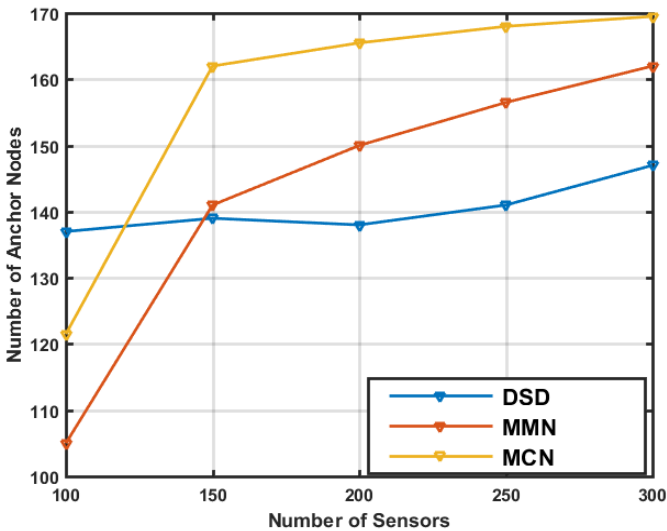


Figure 4.7: Number of anchors versus number of sensors for L-Shaped distribution.

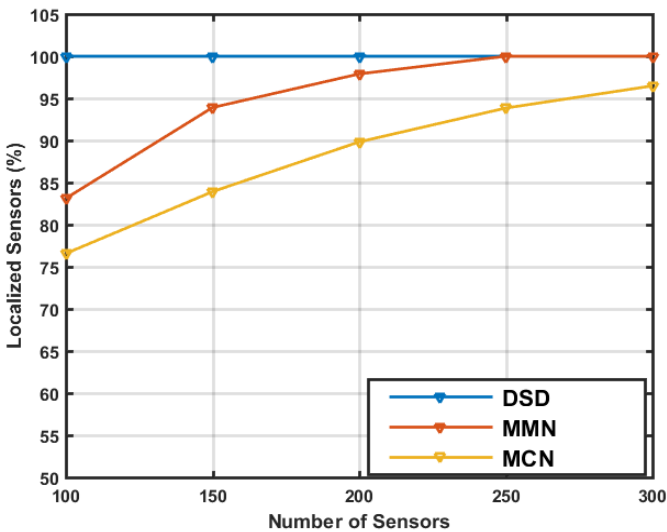


Figure 4.8: Localized sensor ratio versus number of sensors for L-Shaped distribution.

The same analysis with the same settings of L-Shaped distribution is done for U-Shaped sensor deployment. The results are indicated in Figure 4.9 and Figure 4.10. DSD creates fewer anchors to localize %100 of the sensor nodes than MMN and MCN as the number of sensors deployed in the localization region increase. Otherwise, MMN and MCN shows better performance in terms of anchor number. However, in

such cases, MMN and MCN cannot localize all sensors. That is; although DSD uses more anchors than MMN and MCN for low number of sensors as in Figure 4.9, its sensor localization rate is higher for all conditions as shown in Figure 4.10.

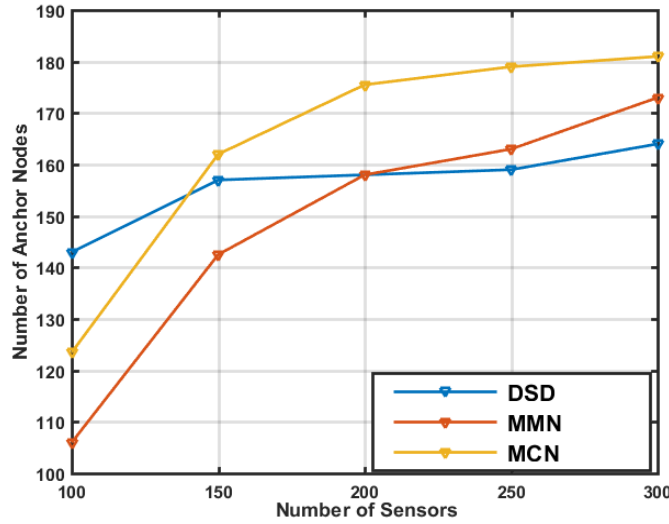


Figure 4.9: Number of anchors versus number of sensors for U-Shaped distribution.

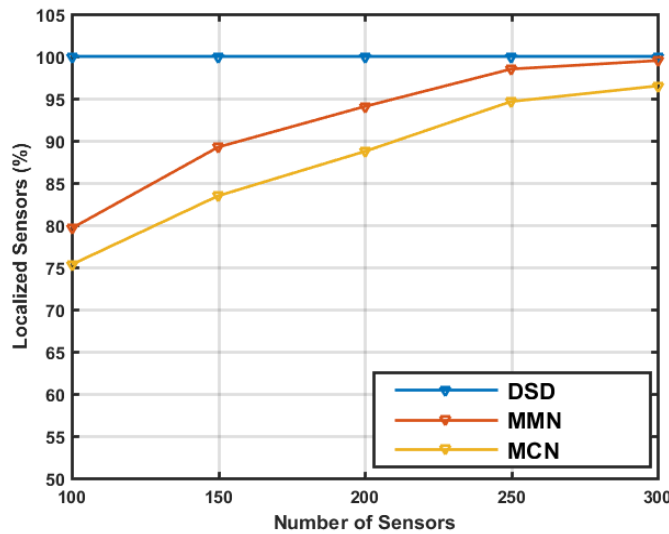


Figure 4.10: Localized sensor ratio versus number of sensors for U-Shaped distribution.

As mentioned before, MMN and MCN algorithms contain a group of mobile anchors and the distance is calculated to be the path length of the group. Since it is not fair to compare the path length of a single mobile anchor with of a group, the distance parameter is not analyzed for MMN and MCN.

In this chapter, a dynamic path planning trajectory is proposed. Similar to SSD, the dynamic algorithm comprises of two phases. First the sensors are detected in the surveillance region, then the anchor nodes are generated for sensor localization by using perpendicular bisector strategy. Since DSD is a dynamic approach, the localization is performed during the trajectory planning process. The simulation results show that DSD outperforms alternative dynamic methods in terms of localization success, number of anchors and path length.

CHAPTER 5

LOCALIZATION USING ALTERNATING MINIMIZATION ALGORITHM

In this thesis, two novel path planning algorithms are proposed for static and dynamic scenarios. The main purpose of these approaches is to obtain high quality anchor positions and high localization accuracy with low energy consumption. Making use of these anchor positions, the proposed node localization algorithm provides an effective solution for the nonlinear and non-convex sensor network localization problem.

The proposed node localization algorithm exploits node-to-node distances. In 2D, an unknown coordinate with at least three neighboring anchors can be found by solving a linear set of equations [3]. Using the generated anchor positions and sensor-to-anchor distances, the non-convex and non-linear sensor localization problem can be solved by using optimization. In this thesis, the original problem is modified to a bi-convex form. The modified problem is solved by using alternating minimization algorithm iteratively. While AMA does not guarantee the optimum solution, it provides an effective sensor localization. Since sensor-to-anchor distances are used together with anchor positions, AMA performs localization non-cooperatively.

Considering the proposed path planning algorithms SSD and DSD, let the total number of anchors deployed in the surveillance region be M where $M = M_{search}^{SSD} + M_{decision}^{SSD}$ for SSD and $M = M_{search}^{DSD} + M_{decision}^{DSD}$ for DSD. The Euclidean distance d_{li} can be obtained by TOA measurements. For the formulation of localization problem,

the distance between l^{th} anchor and i^{th} sensor is denoted by,

$$d_{li} = \sqrt{(a_{l,x} - x_i)^2 + (a_{l,y} - y_i)^2} + v_{li} \quad (5.1)$$

where $a_{l,x}$ and $a_{l,y}$ are the components of l^{th} anchor position \mathbf{a}_l for $l = 1, 2, \dots, M$; x_i and y_i are the components of i^{th} sensor position \mathbf{x}_i for $i = 1, 2, \dots, N$; v_{li} is the additive white Gaussian noise.

Another way of obtaining d_{li} is using RSS measurements when the transmitted power (P_{Tr}) and the path loss exponent (γ) are known. The localization problem can be solved by directly using RSS measurements in an optimization algorithm as in [19]. In the proposed approach, on the other hand, the distance information is obtained from the RSS measurements and used in alternating minimization. The path loss model in [20] is given as,

$$RSS(d) = P_{Tr} - PL(d_0) - 10\gamma \log_{10} \frac{d}{d_0} + X_\sigma \quad (5.2)$$

where $RSS(d)$ is the total path loss for distance d , PL is the path loss for a reference distance d_0 and X_σ is additive, zero-mean white Gaussian noise with the standard deviation σ .

The optimization procedure is based on finding the coordinates of a sensor which satisfy the distance constraints. Making use of the known anchor positions and the sensor-to-anchor distances, localization problem is solved by minimizing the difference between d_{li} and $\|\mathbf{a}_l - \mathbf{x}_i\|$ [3]. That is, non-cooperative localization problem is given as,

$$\underset{\mathbf{x}_i}{\text{Minimize}} \quad \sum_{(l) \in M} (\|\mathbf{a}_l - \mathbf{x}_i\|_2 - d_{li})^2 \quad (5.3)$$

However, the quadratic cost function in (5.3) is non-convex. In this chapter, the nonlinear and non-convex localization problem is modified such that it becomes bi-convex and satisfies (5.3) implicitly. The modified problem is solved by alternating minimization using a convex optimization tool. The proposed alternating minimiza-

tion algorithm (AMA) does not guarantee the global optimum but it provides an effective solution for this problem.

5.1 Alternating Minimization

The least squares expression in (5.3) is changed to a constrained optimization problem in the proposed algorithm. The distance constraint that must be satisfied is given as,

$$\|\mathbf{a}_l - \mathbf{x}_i\|_2 = d_{li} \quad (5.4)$$

However, equality constraints cause the optimization problem to be non-convex. Therefore, the equality constraint in (5.4) is modified as an inequality, i.e.,

$$\|\mathbf{a}_l - \mathbf{x}_i\|_2 \leq d_{li} \quad (5.5)$$

On the other hand, without a lower bound, the inequality (5.5) leads to the trivial solution. To avoid this, the summation of the left term in (5.5) for all $(l) \in M$ can be maximized in the cost function. Using the inequality (5.5) in the constraint and maximizing the summation $\sum_{(l) \in M} \|\mathbf{a}_l - \mathbf{x}_i\|_2^2$ in the cost function implicitly satisfy the relation (5.3). However, $\|\mathbf{a}_l - \mathbf{x}_i\|_2^2 = (\mathbf{a}_l - \mathbf{x}_i)^T (\mathbf{a}_l - \mathbf{x}_i)$ is non-convex. In order to make it convex, it is changed to an affine form as in,

$$\|\mathbf{a}_l - \mathbf{x}_i\|_2^2 \approx (\mathbf{a}_l - \mathbf{x}_i)^T (\mathbf{a}_l - \bar{\mathbf{x}}_i) \quad (5.6)$$

where $\bar{\mathbf{x}}_i$ is the sensor position found in the previous iteration. In addition, slack variables w_l 's are introduced to obtain a feasible inequality constraint. w_l 's which are the elements of vector \mathbf{w} are minimized. Hence, the formulation of non-convex localization problem in (5.3) is converted to a biconvex form and the modified problem is solved iteratively, i.e.,

$$\underset{\mathbf{x}_i, \mathbf{w}, t}{\text{Minimize}} \quad t - \sum_{(l) \in M} (\mathbf{a}_l - \mathbf{x}_i)^T (\mathbf{a}_l - \bar{\mathbf{x}}_i) \quad (5.7a)$$

$$\text{Subject to} \quad \|\mathbf{a}_l - \mathbf{x}_i\|_2 \leq d_{li} + w_l \quad (5.7b)$$

$$\|[2\mathbf{w}; t - 1]\| \leq t + 1 \quad (5.7c)$$

$$\mathbf{x}_i, \mathbf{a}_l \geq \mathbf{0}, \quad i = 1, \dots, N, \quad l = 1, \dots, M \quad (5.7d)$$

Here, the maximizing the summation of the terms in (5.6) is implied in the minimization with minus sign in (5.7a). In alternating minimization, $\|\mathbf{a}_l - \mathbf{x}_i\|_2$ is minimized and the summation of distance squares from the previous iteration are maximized. The slack variables introduced for a feasible inequality at (5.7b) are minimized due to the constraint (5.7c). Hence, the estimated distances converge to d_{li} 's. Having a convex cost function and affine constraints, the biconvex formulation in (5.7) becomes a modified version of the original non-convex localization problem. Therefore, it can be solved by convex optimization tools. The initial value of $\bar{\mathbf{x}}$ is chosen to be the center of the neighboring anchors.

The implementation of the alternating minimization algorithm in wireless sensor networks for the non-cooperative scheme is presented in Algorithm 5.1. Application of the alternating minimization algorithm for the cooperative scenario where the sensor-to-sensor and sensor-to-anchor distances are known is proposed in [43] and [44].

Algorithm 5.1: Alternating Minimization Algorithm

Let \mathbf{x} be the unknown sensor position and $\bar{\mathbf{x}}$ be the sensor position obtained in the previous iteration. ξ is an integer representing the minimum number of neighboring anchors required for localization.

Step 1: Detect the sensors having at least ξ number of neighboring anchors in their circular communication ranges.

Step 2: Choose an unknown sensor for localization. Find the initial sensor position by calculating $\bar{\mathbf{x}}$ as the center point of the neighboring anchor positions.

Step 3: Using the initial position $\bar{\mathbf{x}}$, solve the alternating minimization problem in (5.7) at most 2 iterations and find \mathbf{x} .

Step 4: If all the sensors with at least ξ number of neighboring anchors are localized, finalize the localization process. Otherwise go to step 2.

The Algorithm 5.1 can be applied in a sensor network at once or as in [45]. That is; when the sensors with ξ number of neighboring anchors are detected, Algorithm 5.1 is applied to those sensors and all of them are localized at once. In addition, as in [45], firstly the sensors with ξ number of neighboring anchors are localized and they are assigned as anchors. Then, the remaining sensors are localized by Algorithm 5.1 using the updated anchor positions. This is the iterative application of the Algorithm 5.1.

5.2 Simulation Results for Alternating Minimization

In this section, the performance of the proposed alternating minimization algorithm is considered. The linear localization approach, particle swarm optimization (PSO) and second order cone programming (SOCP) algorithms are compared with AMA in terms of their localization performances. The root mean square error (RMSE) of the localization is,

$$RMSE = \sqrt{\frac{1}{N} \sum_{i=1}^N (x_i - \hat{x}_i)^2} \quad (5.8)$$

where x_i and \hat{x}_i are the true and estimated sensors positions, respectively.

AMA is implemented in SSD, LMAT, CIRCLES and SLMAT for $N = 10$. Figure 5.1, shows the change in $RMSE$ calculated by (5.8) with increasing standard deviation of the additive Gaussian noise in sensor distances given in (5.1). Since SSD, LMAT and SLMAT create anchor nodes at the same coordinates for the same unknown sensor positions, they have close $RMSE$ results as in Figure 5.1. In addition, the $RMSE$ of CIRCLES algorithm is slightly lower than SSD, LMAT and SLMAT for the same number of residue unknowns. This is because CIRCLES uses more anchor nodes to localize the same amount of sensors as SSD and LMAT for the corresponding distributions.

When both AMA and linear localization [18] are implemented in SSD using the same settings of Figure 5.1, AMA has lower $RMSE$ as in Figure 5.2. Thus, it is more effective in terms of localization accuracy compared to the linear method [18]. This is because the localization requires the solution of a nonlinear second order equation. AMA solves this nonlinear equation while the linear approach linearizes the equations and solves them indirectly.

The change in $RMSE$ with increasing noise standard deviation is indicated in Figure 5.3, where the anchor threshold is 5 and 10 sensors are deployed in the region with given distributions. Although GM, GM-VAF and DSD algorithms use perpendicular bisector strategy to create anchors, DSD mostly chooses the positions on the communication circle. This creates higher quality anchor coordinates and increases the localization accuracy of DSD compared to GM and GM-VAF algorithms.

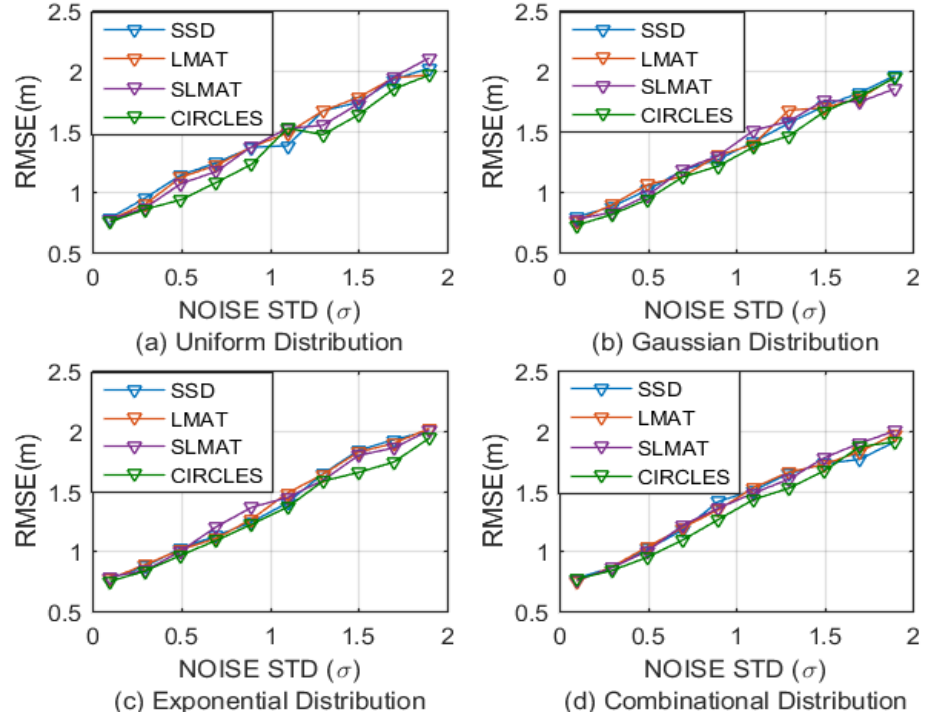


Figure 5.1: RMSE versus noise standard deviation (σ) for different algorithms.

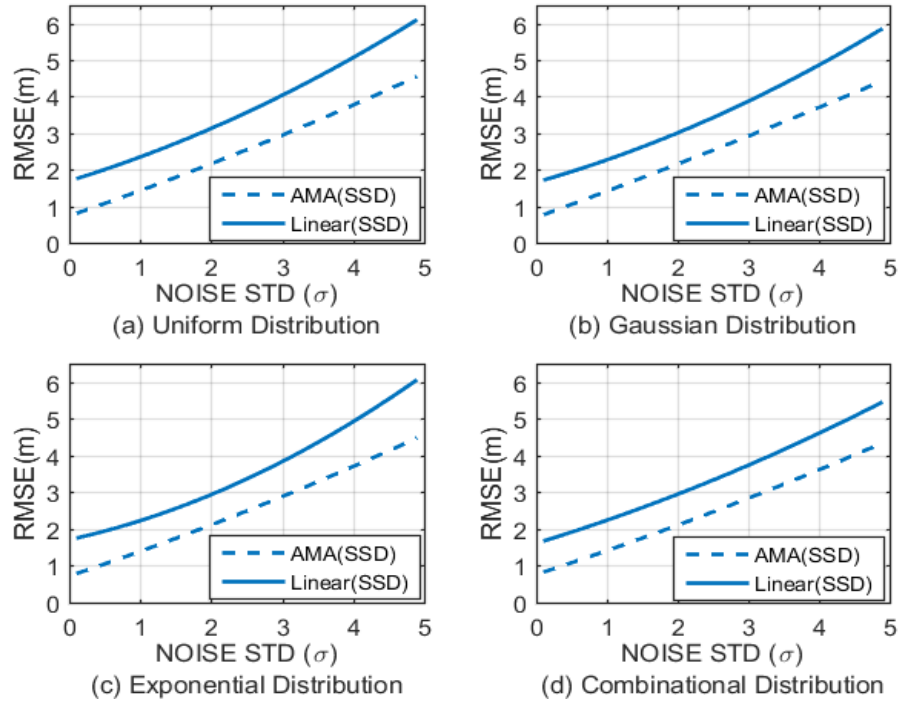


Figure 5.2: Performance comparison for AMA linear and AMA approaches.

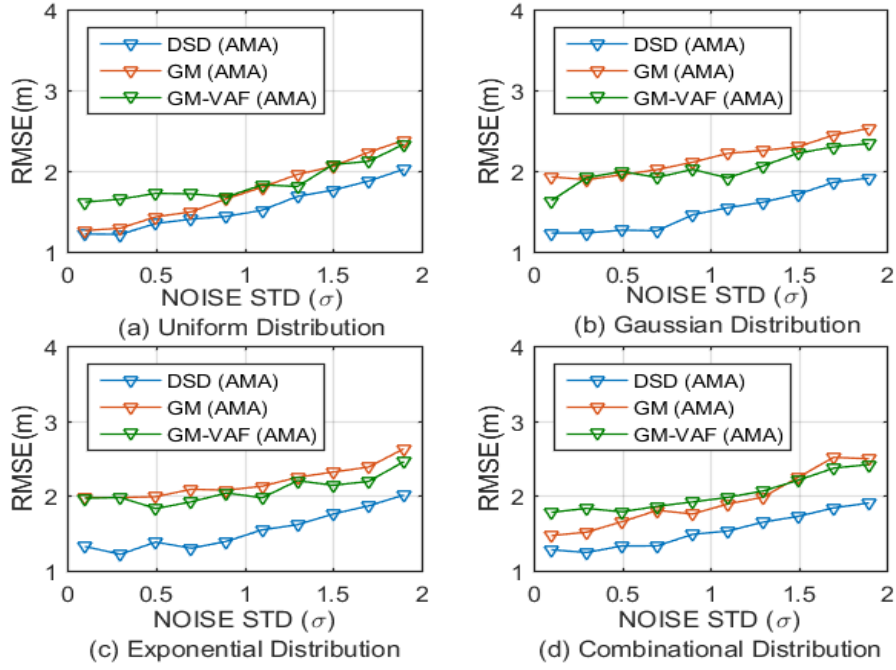


Figure 5.3: RMSE versus σ , when AMA is applied to DSD, GM and GM-VAF.

When both AMA and linear localization [18] are implemented in DSD using the same settings of Figure 5.3, AMA has lower $RMSE$ as in Figure 5.4. Thus, localization by AMA is more accurate compared to the linear approach.

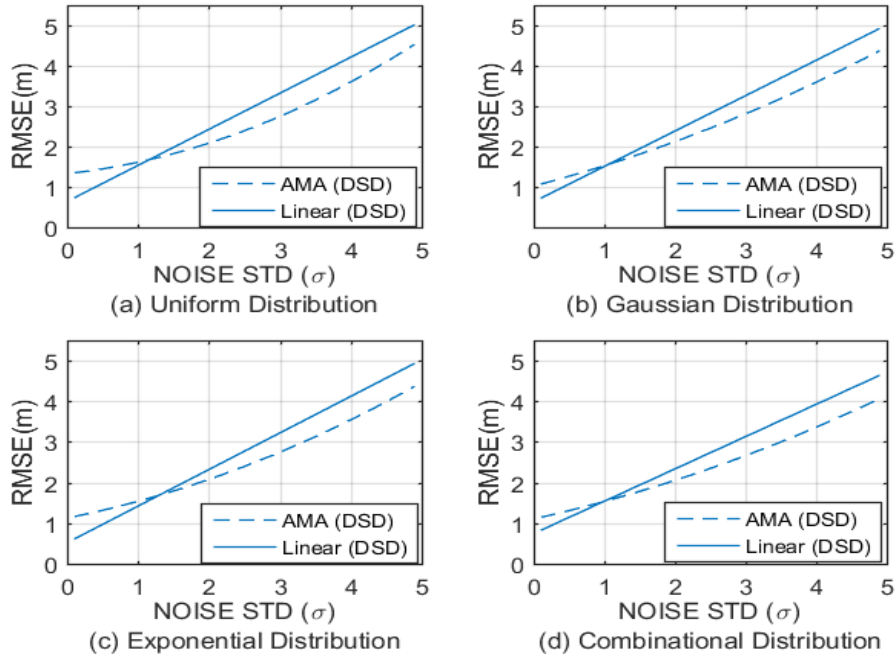


Figure 5.4: RMSE versus σ , when AMA and linear method are applied to DSD.

AMA is compared with PSO [20] and SOCP [19]. In [20], PSO is compared with a neural network algorithm [46], DV-distance [47], ecolocation [48] and iterative multi-lateration algorithms [45]. Since PSO outperforms these techniques, AMA is compared with PSO for validation. AMA, PSO and SOCP are applied iteratively. Once they localize a sensor, it becomes an anchor node for the next iteration. Here, $d_{l,i} = \|\mathbf{a}_l - \mathbf{x}_i\|_2$ is obtained from RSS measurements in (5.2).

The same settings as in [20] are used for fair comparison. That is, L_1 and L_2 are $50m$, $R = 25m$, $P_{Tr} = 0dBm$, $d_0 = 1m$, $PL(d_0) = 55dBm$, $\gamma = 4$ and $\sigma = 2$. The number of anchors is %10 of the number of nodes in the region. Here, the localization error is calculated as in (5.9) for all sensors.

$$\text{Localization Error} = \frac{1}{N} \sum_{i=1}^N \frac{|d_{li} - \|\mathbf{a}_l - \mathbf{x}_i\|_2|}{R} \quad (5.9)$$

The localization error (%) versus the number of nodes is shown in Figure 5.5 for uniform distribution. AMA outperforms SOCP. For fewer nodes, AMA has lower error than PSO. Although the error in AMA is higher than PSO for high number of nodes, AMA and PSO converge closely as the number of nodes increases. Since AMA localizes a sensor in at most 2 iterations, its error accumulation is lower than the rest.

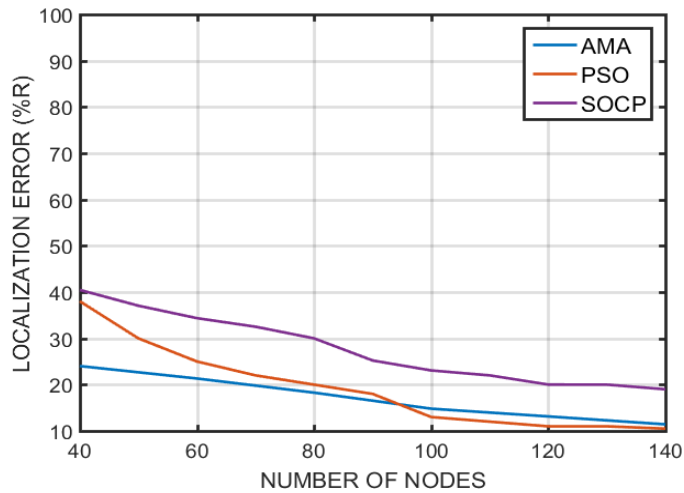


Figure 5.5: Performances of SOCP, PSO and AMA.

In this chapter, alternating minimization algorithm is proposed for wireless sensor network localization problem. In this approach, the non-convex problem is converted to a biconvex form and solved iteratively. Without making rank assumptions, alternating minimization algorithm provides an effective solution for the original localization problem. The simulation results show that, the proposed algorithm performs better than the alternative node localization methods.

CHAPTER 6

CONCLUSIONS

In this thesis, mobile anchor based wireless sensor network localization problem is investigated. Two novel path planning algorithms and a node localization algorithm are proposed for this problem. The proposed algorithms can be outlined as follows;

1. The static search-and-decide (SSD) algorithm for static path planning scenarios.
2. The dynamic search-and-decide (DSD) algorithm for dynamic path planning scenarios.
3. Alternating minimization algorithm (AMA) for node localization in mobile anchor based wireless sensor networks.

In this thesis, SSD and DSD algorithms use a two-phase procedure to search the surveillance region and decide the mobile anchor trajectory. Furthermore, the non-convex, NP-hard node localization problem is solved iteratively using alternating minimization algorithm.

In the first phase of the proposed static path planning algorithm, the mobile anchor traverses the surveillance region by generating anchor nodes. The area is divided into circular grids and the acknowledgment signals coming from the sensors at each grid are recorded. In this way, the mobile anchor detects the sensors deployed in the region. In the second phase, the mobile anchor visits the grids occupied by the sensors and generate anchors in the form of equilateral triangles for localization based on trilateration. In SSD, the mobile anchor does not revisit redundant areas which

results in generating less anchors with shorter path. Since LMAT and CIRCLES outperform SCAN, HILBERT and RWP algorithms, the proposed static path planning algorithm is compared with LMAT, SLMAT and CIRCLES for validation. The simulation results show that SSD performs better than the alternative static approaches in terms of localization success, the number of anchor and the path length. Moreover, the proposed static algorithm achieves the same localization success as other trilateration based methods such as LMAT and SLMAT by generating less number of anchor nodes in a shorter path. SLMAT algorithm provides the same anchor positions with LMAT by using less corners for the mobile anchor movement. Since LMAT follows a zigzag path and SLMAT follows a straight path to generate anchor nodes, SLMAT turns the corners only at the leftmost and the rightmost of the surveillance region. This decreases the energy consumption for the mobile robot. Similar to SLMAT, SSD follows a straight path and turns fewer corners. Therefore, even in the worst case when the sensor population is high, SSD has the same number of corners as SLMAT has. In the sparsely populated cases, SSD has less corners than SLMAT. That is, SSD provides more feasible and energy efficient trajectory.

In the proposed dynamic path planning algorithm, a similar two phased method is used. When the mobile anchor detects the sensors in the surveillance region, it revisits the grids populated by the sensors for localization. However, DSD performs the sensor localization process dynamically. That is, the localization is performed during the path planning process. In the decision-phase of DSD algorithm, the mobile anchor starts to revisit the populated grids and generates anchor nodes using perpendicular bisector strategy. The mobile anchor leaves a grid only if all the sensors in the corresponding grid are localized in this phase. The proposed dynamic path planning algorithm is compared with GM, GM-VAF, MMN and MCN algorithms. The simulation results show that DSD localizes more sensors than the alternative approaches using less number of anchors and a shorter path. The reason is that the redundant areas are not revisited in DSD. Therefore, DSD is an energy efficient approach using less number of anchors and shorter path. Moreover, DSD algorithm provides high localization accuracy since it generates non-collinear anchors using the perpendicular bisector strategy.

In the proposed localization algorithm, an effective solution is brought to the non-convex, NP-hard node localization problem. For the non-cooperative scenario, the original nonlinear and non-convex problem is converted into a biconvex form. The modified version of the problem is solved by using alternating minimization algorithm and second order programming, iteratively. AMA is compared with linear localization, PSO and SOCP approaches in terms of node position accuracy. The simulation results show that the proposed alternating minimization algorithm outperforms the alternative methods, because it is less prone to error accumulation and provides more accurate sensor positioning.

REFERENCES

- [1] D. Ciuonzo, P. S. Rossi, and P. Willett. Generalized rao test for decentralized detection of an uncooperative target. *IEEE Signal Processing Letters*, 24(5):678–682, May 2017.
- [2] 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. *IEEE Signal Processing Magazine*, 22(4):54–69, July 2005.
- [3] Z. q. Luo, W. k. Ma, A. M. c. So, Y. Ye, and S. Zhang. Semidefinite relaxation of quadratic optimization problems. *IEEE Signal Processing Magazine*, 27(3):20–34, May 2010.
- [4] G. Mao, Ba. Fidan, and B. D.O. Anderson. Wireless sensor network localization techniques. *Computer Networks*, 51(10):2529 – 2553, 2007.
- [5] G. Han, J. Jiang, C. Zhang, T. Q. Duong, M. Guizani, and G. K. Karagiannidis. A survey on mobile anchor node assisted localization in wireless sensor networks. *IEEE Communications Surveys Tutorials*, 18(3):2220–2243, thirdquarter 2016.
- [6] Z. Zhong, D. Luo, S. Liu, X. Fan, and Z. Qu. An adaptive localization approach for wireless sensor networks based on gauss-markov mobility model. *Acta Automatica Sinica*, 36(11):1557–1568, 2010.
- [7] E. Erdemir and T. E. Tuncer. Path planning and localization for mobile anchor based wireless sensor networks. In *2017 Proceedings of the 25th European Signal Processing Conference (EUSIPCO)*, Manuscript in preparation, 2017.
- [8] D. Koutsonikolas, S. M. Das, and Y. C. Hu. Path planning of mobile landmarks for localization in wireless sensor networks. *Computer Communications*, 30(13):2577 – 2592, 2007. Sensor-Actuated Networks.
- [9] R. Huang and G. V. Zaruba. Static path planning for mobile beacons to localize sensor networks. In *Pervasive Computing and Communications Workshops, 2007. PerCom Workshops '07. Fifth Annual IEEE International Conference on*, pages 323–330, March 2007.
- [10] Z. Hu, D. Gu, Z. Song, and H. Li. Localization in wireless sensor networks using a mobile anchor node. In *2008 IEEE/ASME International Conference on Advanced Intelligent Mechatronics*, pages 602–607, July 2008.

- [11] G. Han, H. Xu, J. Jiang, L. Shu, T. Hara, and S. Nishio. Path planning using a mobile anchor node based on trilateration in wireless sensor networks. *Wireless Communications and Mobile Computing*, 13(14):1324–1336, 2013.
- [12] G. Han, X. Yang, L. Liu, M. Guizani, and W. Zhang. A disaster management-oriented path planning for mobile anchor node-based localization in wireless sensor networks. *IEEE Transactions on Emerging Topics in Computing*, PP(99):1–1, 2017.
- [13] J. Rezazadeh, M. Moradi, A. S. Ismail, and E. Dutkiewicz. Superior path planning mechanism for mobile beacon-assisted localization in wireless sensor networks. *IEEE Sensors Journal*, 14(9):3052–3064, Sept 2014.
- [14] M. L. Sichitiu and V. Ramadurai. Localization of wireless sensor networks with a mobile beacon. In *2004 IEEE International Conference on Mobile Ad-hoc and Sensor Systems (IEEE Cat. No.04EX975)*, pages 174–183, Oct 2004.
- [15] Y. Yang and L. Zhang. Dynamic path planning of mobile beacon for localization in wireless sensor network. In *2013 International Conference on Wireless Communications and Signal Processing*, pages 1–5, Oct 2013.
- [16] L. Wang and W. Wen. Path planning for mobile beacon nodes in localization for wireless sensor networks. *International Journal of Digital Content Technology and its Applications*, 7(3):16, 2013.
- [17] G. Han, J. Chao, C. Zhang, L. Shu, and Q. Li. The impacts of mobility models on dv-hop based localization in mobile wireless sensor networks. *Journal of Network and Computer Applications*, 42:70–79, 2014.
- [18] F. Liu, H. Zhu, Z. Gu, and Y. Liu. A linear localization algorithm for wireless sensor network based on rssi. In *Advanced Research on Computer Education, Simulation and Modeling*, pages 384–389. Springer, 2011.
- [19] S. Tomic, M. Beko, and R. Dinis. Rss-based localization in wireless sensor networks using convex relaxation: Noncooperative and cooperative schemes. *IEEE Transactions on Vehicular Technology*, 64(5):2037–2050, 2015.
- [20] P. Chuang, C. Wu, et al. Employing pso to enhance rss range-based node localization for wireless sensor networks. *Journal of information science and engineering*, 27(5):1597–1611, 2011.
- [21] U. Nazir, N. Shahid, M. A. Arshad, and S. H. Raza. Classification of localization algorithms for wireless sensor network: A survey. In *2012 International Conference on Open Source Systems and Technologies*, pages 1–5, Dec 2012.
- [22] I. F. Akyildiz and M. C. Vuran. *Wireless sensor networks*, volume 4. John Wiley & Sons, 2010.

- [23] I. Guvenc and C. Chong. A survey on toa based wireless localization and nlos mitigation techniques. *IEEE Communications Surveys & Tutorials*, 11(3), 2009.
- [24] M. Boccadoro, G. De Angelis, and P. Valigi. Tdoa positioning in nlos scenarios by particle filtering. *Wireless Networks*, 18(5):579–589, 2012.
- [25] G. Wang and K. Yang. A new approach to sensor node localization using rss measurements in wireless sensor networks. *IEEE Transactions on Wireless Communications*, 10(5):1389–1395, 2011.
- [26] R. Stoleru, T. He, and J. A. Stankovic. Range-free localization. *Secure Localization and Time Synchronization for Wireless Sensor and Ad Hoc Networks*, pages 3–31, 2007.
- [27] D. Niculescu and B. Nath. Ad hoc positioning system (aps). In *Global Telecommunications Conference, 2001. GLOBECOM'01. IEEE*, volume 5, pages 2926–2931. IEEE, 2001.
- [28] H. Wymeersch, J. Lien, and M. Z. Win. Cooperative localization in wireless networks. *Proceedings of the IEEE*, 97(2):427–450, 2009.
- [29] A. Y. Teymorian, W. Cheng, L. Ma, X. Cheng, X. Lu, and Z. Lu. 3d underwater sensor network localization. *IEEE Transactions on Mobile Computing*, 8(12), 2009.
- [30] X. Li, Y. Zhang, K. Xu, G. Fan, and H. Wu. Research of localization and tracking algorithms based on wireless sensor network. *JOURNAL OF INFORMATION & COMPUTATIONAL SCIENCE*, 8(4):708–715, 2011.
- [31] G. S. Kuruoglu, M. Erol, and S. Oktug. Three dimensional localization in wireless sensor networks using the adapted multi-lateration technique considering range measurement errors. In *Proc. IEEE GLOBECOM Workshops*, pages 1–5, 2009.
- [32] Z. Yang and Y. Liu. Quality of trilateration: Confidence-based iterative localization. *IEEE Transactions on Parallel and Distributed Systems*, 21(5):631–640, 2010.
- [33] D. K. Goldenberg, P. Bihler, M. Cao, J. Fang, B. Anderson, A. S. Morse, and Y. R. Yang. Localization in sparse networks using sweeps. In *Proceedings of the 12th annual international conference on Mobile computing and networking*, pages 110–121. ACM, 2006.
- [34] F. Santos and I. Tecnico. Localization in wireless sensor networks. *ACM Journal Name*, 5:1–19, 2008.
- [35] T. Camp, J. Boleng, and V. Davies. A survey of mobility models for ad hoc network research. *Wireless communications and mobile computing*, 2(5):483–502, 2002.

- [36] D.B. Jhonson, D. A. Maltz, and J. Broch. Dynamic source routing in ad hoc wireless networks. *Mobile Computing*, 353:153–181, 1996.
- [37] X. Kuang, S. Hui-He, and F. Rui. A new distributed localization scheme for wireless sensor networks. *Acta Automatica Sinica*, 34(3):344–348, 2008.
- [38] D. Ciullo, V. Martina, M. Garetto, and E. Leonardi. Impact of correlated mobility on delay-throughput performance in mobile ad hoc networks. *IEEE/ACM Transactions on Networking (TON)*, 19(6):1745–1758, 2011.
- [39] Z. Shasha, W. Dong, and L. Xiaohong. A visible obstacle mobility model based on activity area. *emergency*, 5(4), 2012.
- [40] P. E. Hart, N. J. Nilsson, and B. Raphael. A formal basis for the heuristic determination of minimum cost paths. *IEEE transactions on Systems Science and Cybernetics*, 4(2):100–107, 1968.
- [41] C. Ou and W. He. Path planning algorithm for mobile anchor-based localization in wireless sensor networks. *IEEE Sensors Journal*, 13(2):466–475, 2013.
- [42] Z. Liu, W. Dai, and M. Z. Win. Node placement for localization networks. In *2017 IEEE International Conference on Communications (ICC)*, pages 1–6, May 2017.
- [43] E. Erdemir and T. E. Tuncer. Wireless sensor network localization using alternating minimization algorithm. In *Signal Processing and Communications Applications Conference (SIU), 2015 23th*, pages 2191–2194. IEEE, 2015.
- [44] E. Erdemir and T. E. Tuncer. Cooperative localization in one-hop and multi-hop wireless sensor networks using alternating minimization algorithm. In *Signal Processing and Communication Application Conference (SIU), 2016 24th*, pages 1357–1360. IEEE, 2016.
- [45] A. Savvides, C. Han, and M. B. Strivastava. Dynamic fine-grained localization in ad-hoc networks of sensors. In *Proceedings of the 7th annual international conference on Mobile computing and networking*, pages 166–179. ACM, 2001.
- [46] A. Dana, A. K. Zadeh, and B. Hekmat. Localization in ad-hoc networks. In *Telecommunications and Malaysia International Conference on Communications, 2007. ICT-MICC 2007. IEEE International Conference on*, pages 313–317. IEEE, 2007.
- [47] D. Niculescu and Ba. Nath. Dv based positioning in ad hoc networks. *Telecommunication Systems*, 22(1):267–280, 2003.
- [48] K. Yedavalli, B. Krishnamachari, S. Ravula, and B. Srinivasan. Ecolocation: a sequence based technique for rf localization in wireless sensor networks. In *Proceedings of the 4th international symposium on Information processing in sensor networks*, page 38. IEEE Press, 2005.