Localization Algorithms for Wireless Sensor Networks

by

Vijayanth Vivekanandan

B.A.Sc., Electrical Engineering, University of British Columbia, 2003

A THESIS SUBMITTED IN PARTIAL FULFILMENT OF
THE REQUIREMENTS FOR THE DEGREE OF

MASTER OF APPLIED SCIENCE

in

THE FACULTY OF GRADUATE STUDIES

(�Electrical and Computer Engineering)

..............................................................

..............................................................

..............................................................

THE UNIVERSITY OF BRITISH COLUMBIA

December 2005

© Vijayanth Vivekanandan, 2005
In presenting this thesis in partial fulfilment of the requirements for an advanced degree at the University of British Columbia, I agree that the Library shall make it freely available for reference and study. I further agree that permission for extensive copying of this thesis for scholarly purposes may be granted by the head of my department or by his or her representatives. It is understood that copying or publication of this thesis for financial gain shall not be allowed without my written permission.

(Signature) __________________________

Electrical and Computer Engineering

The University Of British Columbia
Vancouver, Canada

Date __________________________
Abstract

Many applications in wireless sensor networks require sensor nodes to obtain their absolute or relative geographical positions. Due to the size, cost and energy restrictions imposed by sensor nodes, only a few nodes can be equipped with the Global Positioning System (GPS) capability and act as anchors for the rest of the network. The algorithms based on classical Multidimensional Scaling (MDS) [1][2] only require three or four anchor nodes and can provide higher accuracy than some other schemes.

In the first part of this thesis, we propose and analyze the use of ordinal MDS for localization in wireless sensor networks. Ordinal MDS differs from classical MDS by that it only requires a monotonicity constraint between the shortest path distance and the Euclidean distance for each pair of nodes. Simulation studies are conducted under square and C-shaped topologies with different connectivity levels and number of anchors. Results show that ordinal MDS provides a lower position estimation error than classical MDS in both hop-based and range-based scenarios.

In the second part of this thesis, we propose a concentric anchor-beacons (CAB) localization algorithm for wireless sensor networks. CAB is a range-free approach and uses a small number of anchor nodes. Each anchor emits beacons at different power levels
periodically. From the information received by each beacon heard, nodes determine which annular ring they are located within each anchor. Each node uses the approximated center of intersection of the rings as its position estimate. Simulation results show that the estimation error reduces by half when anchors transmit beacons at two different power levels periodically instead of at a single level. CAB also gives a lower estimation error than other range-free localization schemes (e.g., Centroid, APIT) when the anchor-to-node range ratio is less than four.
Contents

Abstract ................................................................. ii

Contents ................................................................. iv

List of Figures ......................................................... viii

List of Tables ........................................................... xiv

List of Abbreviations ................................................... xv

Acknowledgements ....................................................... xvii

1 Introduction ............................................................. 1
   1.1 Motivations and Objectives ..................................... 2
   1.2 Structure of the Thesis ......................................... 5

2 Related Work on Localization ....................................... 6
   2.1 Connectivity-Based Algorithms ................................. 7
      2.1.1 Convex Optimization .................................... 7
      2.1.2 MDS-MAP .................................................. 9
Contents

<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.1.3</td>
<td>MDS-MAP with Patches and Refinement (P, R)</td>
<td>11</td>
</tr>
<tr>
<td>2.1.4</td>
<td>Anchor-Initiated MDS-MAP, On-Demand MDS</td>
<td>12</td>
</tr>
<tr>
<td>2.2</td>
<td>Range-Based Techniques</td>
<td>14</td>
</tr>
<tr>
<td>2.2.1</td>
<td>TERRAIN using ABC</td>
<td>15</td>
</tr>
<tr>
<td>2.2.2</td>
<td>APS</td>
<td>17</td>
</tr>
<tr>
<td>2.2.3</td>
<td>Self-Positioning Algorithm</td>
<td>19</td>
</tr>
<tr>
<td>2.2.4</td>
<td>Ad-hoc Location System (AHLoS)</td>
<td>20</td>
</tr>
<tr>
<td>2.2.5</td>
<td>N-Hop Multilateration Primitive</td>
<td>22</td>
</tr>
<tr>
<td>2.2.6</td>
<td>Hop-TERRAIN with Refinement</td>
<td>23</td>
</tr>
<tr>
<td>2.2.7</td>
<td>APS/Hop-TERRAIN/N-hop Multilateration Primitive</td>
<td>25</td>
</tr>
<tr>
<td>2.3</td>
<td>Angle-Based Techniques</td>
<td>26</td>
</tr>
<tr>
<td>2.3.1</td>
<td>APS using AOA</td>
<td>27</td>
</tr>
<tr>
<td>2.3.2</td>
<td>DV-position</td>
<td>29</td>
</tr>
<tr>
<td>2.3.3</td>
<td>Directionality Localization</td>
<td>31</td>
</tr>
<tr>
<td>2.3.4</td>
<td>Ad-Hoc Ranging and Sectoring</td>
<td>32</td>
</tr>
<tr>
<td>2.4</td>
<td>Range-Free Techniques</td>
<td>33</td>
</tr>
<tr>
<td>2.4.1</td>
<td>Centroid</td>
<td>33</td>
</tr>
<tr>
<td>2.4.2</td>
<td>Approximated Point-in-Triangulation (APIT)</td>
<td>34</td>
</tr>
<tr>
<td>2.5</td>
<td>Other Novel Techniques</td>
<td>37</td>
</tr>
<tr>
<td>2.5.1</td>
<td>Time-based Positioning Scheme TPS</td>
<td>37</td>
</tr>
<tr>
<td>2.5.2</td>
<td>Secure Positioning</td>
<td>39</td>
</tr>
</tbody>
</table>
## Contents

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.5.3 Anchor-Free Localization (AFL)</td>
<td>41</td>
</tr>
<tr>
<td>2.5.4 Sequential Monte Carlo Localization (MCL)</td>
<td>44</td>
</tr>
<tr>
<td><strong>3 Ordinal Multidimensional Scaling</strong></td>
<td>46</td>
</tr>
<tr>
<td>3.1 Motivations and Assumptions</td>
<td>46</td>
</tr>
<tr>
<td>3.2 MDS-MAP(P, O) Algorithm</td>
<td>48</td>
</tr>
<tr>
<td>3.3 Performance Evaluation and Comparison</td>
<td>52</td>
</tr>
<tr>
<td>3.3.1 Random Uniform Network Topology</td>
<td>53</td>
</tr>
<tr>
<td>Hop-Based Performance</td>
<td>53</td>
</tr>
<tr>
<td>Range-Based Performance</td>
<td>56</td>
</tr>
<tr>
<td>Sensitivity Analysis of Range Error on Positioning Error</td>
<td>58</td>
</tr>
<tr>
<td>Sensitivity Analysis of Anchors’ Location on Positioning Error</td>
<td>60</td>
</tr>
<tr>
<td>Further Improvement via (Optional) Global Relative Map Refinement</td>
<td>62</td>
</tr>
<tr>
<td>3.3.2 Random Irregular Network Topology</td>
<td>64</td>
</tr>
<tr>
<td>Hop-Based Performance</td>
<td>64</td>
</tr>
<tr>
<td>Range-Based Performance</td>
<td>66</td>
</tr>
<tr>
<td>Further Improvement via (Optional) Global Relative Map Refinement</td>
<td>67</td>
</tr>
<tr>
<td>3.4 Summary</td>
<td>69</td>
</tr>
<tr>
<td><strong>4 Concentric Anchor-Beacons (CAB) Localization Algorithm</strong></td>
<td>71</td>
</tr>
<tr>
<td>4.1 Motivations and Assumptions</td>
<td>71</td>
</tr>
<tr>
<td>4.2 CAB Algorithm</td>
<td>74</td>
</tr>
</tbody>
</table>
List of Figures

2.1 Connectivity Constraints, [3]. ........................................ 8
2.2 Sequential Localization from starting anchor (A) to ending anchor (D), [4]. 13
2.3 On-Demand positioning from node A to anchors D, H, N, [4] ........ 15
2.4 DV Euclidean model, [5] .................................................. 18
2.5 Collaborative Multilateration model, [6]. .............................. 21
2.6 Nodes obtaining relative angle measurements to other nodes, [7]. ... 28
2.7 Localization using Angulations, [7]. .................................. 29
2.8 DV-position model, [8] ...................................................... 30
2.9 Localization using synchronized, offset, rotating beacon signals, [9]. 31
2.10 Local grid maintaining overlapped triangle regions for localization, [10]. 35
2.11 APIT test using node neighbors, [10]. ............................... 36
2.13 Sensor coverage model, [12]. ......................................... 40
2.14 Different configuration with degrees of freedom, [13]. ............ 42
2.15 Topology construction for fold-freedom, [13]. ..................... 43
3.1 Topology results of hop-based (a) MDS-MAP(P, C) and (b) MDS-MAP(P, O) in a $10r \times 10r$ square network region employing uniform random placement of 200 nodes with connectivity level of 12 and four anchors. Anchors are denoted by shaded circles. Estimation error is represented by lines. 54

3.2 Performance of hop-based MDS-MAP(P, C) and MDS-MAP(P, O) in a $10r \times 10r$ square network region employing uniform random placement of 200 nodes, for varying levels of connectivity and anchors deployed. 55

3.3 Topology results of range-based (a) MDS-MAP(P, C) and (b) MDS-MAP(P, O) in a $10r \times 10r$ square network region employing uniform random placement of 200 nodes with connectivity level of 12 and four anchors. Anchors are denoted by shaded circles. Estimation error is represented by lines. 57

3.4 Performance of range-based MDS-MAP(P, C) and MDS-MAP(P, O) in a $10r \times 10r$ square network region employing uniform random placement of 200 nodes, for varying levels of connectivity and anchors deployed. 58

3.5 Effects of node ranging error on performance of range-based MDS-MAP(P, O) in a $10r \times 10r$ square network region employing uniform random placement of 200 nodes, and four deployed anchors. 59

3.6 Performance comparison between range-based MDS-MAP(P, C) and MDS-MAP(P, O) with respect to varying node range error. 60
3.7 Topology results of range-based MDS-MAP(P, O) when anchors are being placed (a) linearly, (b) in a rectangular manner, (c) close to each other, and (d) randomly. The topology consists of a $10r \times 10r$ square network region employing uniform random placement of 200 nodes with connectivity level of 12 and four anchors. Anchors are denoted by shaded circles. Estimation error is represented by lines. .................................................. 61

3.8 Performance of hop-based MDS-MAP(P, O) and MDS-MAP(P, O, R) in a $10r \times 10r$ square network region employing uniform random placement of 200 nodes, for varying levels of connectivity and anchors deployed. .......................................................... 62

3.9 Performance between range-based MDS-MAP(P, O) and MDS-MAP(P, O, R) in a $10r \times 10r$ square network region employing uniform random placement of 200 nodes, for varying levels of connectivity and anchors deployed. .......................................................... 63

3.10 Topology results of hop-based (a) MDS-MAP(P, C) and (b) MDS-MAP(P, O) in a $10r \times 10r$ irregular (C-shaped) network region employing uniform random placement of 160 nodes with connectivity level of 12 and four anchors. Anchors are denoted by shaded circles. Estimation error is represented by lines. .......................................................... 65
3.11 Performance between hop-based MDS-MAP(P, C) and MDS-MAP(P, O) in a $10r \times 10r$ irregular (C-shaped) network region employing uniform random placement of 160 nodes, for varying levels of connectivity and anchors deployed. ................................................................. 66

3.12 Topology results of range-based (a) MDS-MAP(P, C) and (b) MDS-MAP(P, O) in a $10r \times 10r$ irregular (C-shaped) network region employing uniform random placement of 160 nodes with connectivity level of 12 and four anchors. Anchors are denoted by shaded circles. Estimation error is represented by lines. ................................................................. 67

3.13 Performance of range-based MDS-MAP(P, C) and MDS-MAP(P, O) in a $10r \times 10r$ irregular (C-shaped) network region employing uniform random placement of 160 nodes, for varying levels of connectivity and anchors deployed. ................................................................. 68

3.14 Performance between hop-based MDS-MAP(P, O) and MDS-MAP(P, O, R) in a $10r \times 10r$ irregular (C-shaped) network region employing uniform random placement of 160 nodes, for varying levels of connectivity and anchors deployed. ................................................................. 69

3.15 Performance between range-based MDS-MAP(P, O) and MDS-MAP(P, O, R) in a $10r \times 10r$ irregular (C-shaped) network region employing uniform random placement of 160 nodes, for varying levels of connectivity and anchors deployed. ................................................................. 70
List of Figures

4.1 Anchor beacon transmission ranges with (a) Equal area beacon signals with 3 power levels; and (b) Equal width beacon signals with 3 power levels. \( A_i (i = 1, 2, 3) \) denotes the area of the \( i \)th ring/circle; \( w_i (i = 1, 2, 3) \) denotes the width of the \( i \)th ring/circle. .......................... 73

4.2 An example of localization using CAB .................................................. 76

4.3 Irregular radio patterns for different values of DOI ................................... 80

4.4 Comparison of percentage of nodes localizable versus percentage of anchors deployed for varying levels of ANR. (DOI = 0.05) ......................... 81

4.5 Average estimation error under different number of power levels of the beacons for (a) CAB-EA and (b) CAB-EW. (ANR = 3, DOI = 0.05) .......... 82

4.6 Comparison of estimation error between CAB-EW and CAB-EA. \((m = 2, ANR = 3, DOI = 0.05)\) .................................................. 84

4.7 Comparison of estimation error between CAB-EA and CAB-EW for different DOI values: (a) DOI = 0, (b) DOI = 0.05, and (c) DOI = 0.10. \((m = 3, ANR = 3)\) .................................................. 85

4.8 Comparison of estimation error using randomly heard anchors versus optimally chosen anchors. \((m = 2, ANR = 3, DOI = 0.05)\). .......................... 86

4.9 Comparison between Centroid, APIT, CAB-EA \((m = 2)\), and CAB-EW \((m = 3)\) by increasing the percentage of anchors deployed. \((ANR = 3, DOI = 0.05)\) .................................................. 88
List of Figures

4.10 Comparison between Centroid, APIT, CAB-EA ($m = 2$), and CAB-EW

$(m = 3)$ for varying levels of ANR. (DOI = 0.05) . . . . . . . . . . . . . 89

4.11 Comparison between Centroid, APIT, CAB-EA ($m = 2$), and CAB-EW

$(m = 3)$ under different DOI values. (ANR = 3) . . . . . . . . . . . . . . . . . . . 90
List of Tables

4.1 Information collected by a sensor from its anchors . . . . . . . . . . . . . 77
## List of Abbreviations

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ABC</td>
<td>Assumption Based Coordinates</td>
</tr>
<tr>
<td>AFL</td>
<td>Anchor-Free Localization</td>
</tr>
<tr>
<td>AHLoS</td>
<td>Ad-hoc Location System</td>
</tr>
<tr>
<td>ANR</td>
<td>Anchor-to-Node Range</td>
</tr>
<tr>
<td>AOA</td>
<td>Angle-of-Arrival</td>
</tr>
<tr>
<td>APIT</td>
<td>Approximated Point-In-Triangle</td>
</tr>
<tr>
<td>APS</td>
<td>Ad-hoc Positioning System</td>
</tr>
<tr>
<td>CAB</td>
<td>Concentric Anchor-Beacons</td>
</tr>
<tr>
<td>CAB-EA</td>
<td>Concentric Anchor-Beacons with Equal Area</td>
</tr>
<tr>
<td>CAB-EW</td>
<td>Concentric Anchor-Beacons with Equal Width</td>
</tr>
<tr>
<td>DOI</td>
<td>Degree of Irregularity</td>
</tr>
<tr>
<td>DV</td>
<td>Distance-Vector</td>
</tr>
<tr>
<td>Abbreviation</td>
<td>Description</td>
</tr>
<tr>
<td>--------------</td>
<td>-------------</td>
</tr>
<tr>
<td>EKF</td>
<td>Extended Kalman Filter</td>
</tr>
<tr>
<td>GPS</td>
<td>Global Positioning System</td>
</tr>
<tr>
<td>LMI</td>
<td>Linear Matrix Inequality</td>
</tr>
<tr>
<td>LRG</td>
<td>Location Reference Group</td>
</tr>
<tr>
<td>MCL</td>
<td>Monte Carlo Localization</td>
</tr>
<tr>
<td>MDS</td>
<td>Multidimensional Scaling</td>
</tr>
<tr>
<td>PAV</td>
<td>Pool-Adjacent Violators</td>
</tr>
<tr>
<td>RSS</td>
<td>Received Signal Strength</td>
</tr>
<tr>
<td>SDP</td>
<td>Semi-Definite Program</td>
</tr>
<tr>
<td>SMC</td>
<td>Sequential Monte Carlo</td>
</tr>
<tr>
<td>TDOA</td>
<td>Time-Difference-of-Arrival</td>
</tr>
<tr>
<td>TERRAIN</td>
<td>Triangulation via Extended Range and Redundant Association of Intermediate Nodes</td>
</tr>
<tr>
<td>TOA</td>
<td>Time-of-Arrival</td>
</tr>
<tr>
<td>TPS</td>
<td>Time-based Positioning Scheme</td>
</tr>
</tbody>
</table>
Acknowledgements

I would like to express my sincere gratitude to my graduate supervisor Dr. Vincent Wong for his guidance and support during the course of my graduate studies. I appreciate the considerable amount of time and effort he invested in helping me with my research and thesis. Special thanks to Yi Shang, Wheeler Ruml, and Ying Zhang for providing us with their software code for testing and modification. I would also like to thank my fellow colleagues who have provided helpful suggestions along the way. This thesis could not have been completed without the endless encouragement and love from my family, I am deeply indebted to them.

This work is supported by the Natural Sciences and Engineering Research Council of Canada under grant number 261604-03.
Chapter 1

Introduction

The miniaturization of small devices capable of sensing and communicating with each other has made the possibility of deploying large-scale wireless sensor networks a reality [14]. The purpose of these networks are to remotely sense an area and communicate the results back to a central authority. In order for these sensory data to be valuable, the location from which they were obtained must be known. The functions presented in literature range from military applications to wildlife monitoring. These applications propose of hundreds to thousands of sensors being dropped by airplane over a certain coverage area. This type of deployment would require the nodes to configure their network by themselves. Such vast deployments would restrict the individual placement or programming of all nodes with their unique locations, and hence some system must be in place to accurately and efficiently localize these sensors. Thus, these sensor nodes have limited processing power and battery supply, and need to configure themselves autonomously.

For applications such as position-based routing, event discovery and target tracking, the geographic location of the sensor nodes need to be known. Consider the example where a sensor network is used to detect a fire event in a forest. Once a sensor node has detected that the temperature is higher than a certain threshold, it sends a message to the
central authority by relaying through other nodes in a multi-hop manner. The message needs to indicate the location of the node which detected the event. When an event occurs (or a stimulus is detected), the sensor nodes can forward the data information along with their coordinates. Thus, localization of sensor nodes is important in certain applications.

1.1 Motivations and Objectives

The current location technology such as global positioning system (GPS) does not meet the sensor nodes’ requirements for low cost, low energy consumption, and small size [15]. In addition, GPS requires line-of-sight to global satellites that may not be available for some applications in sensor networks.

In order to solve this dilemma, nodes in the network must collaboratively execute an algorithm in which the objective is to provide each node with an estimate of its position in the network. Position estimates may be based on a relative coordinate system within the network, or an absolute geographical reference. Although, much research has been conducted in the area of localization for cellular networks and mobile ad-hoc networks, sensor networks provide unique challenges that cannot be solved by using traditional techniques from these other areas.

Various centralized [1][3] and distributed [5][16] localization algorithms have been proposed recently. With respect to robustness and energy efficiency, distributed algorithms are preferred over centralized schemes. The localization algorithms can further be divided
into range-based [2][5], angle-based [7][17], and range-free [10][18] approaches.

Range-based schemes assume that sensor nodes have the ability to obtain distance estimates to other nodes. This distance information can be obtained through various techniques involving specialized hardware. Common methods of measuring distance between wireless nodes include the use of received signal strength (RSS), time-of-arrival (TOA), and time-difference-of-arrival (TDOA). In the RSS method, the measured RSS is converted to a distance estimate by using a predetermined path loss model and a reference distance of which the signal strength is known. The TOA method relies on either synchronized nodes being able to determine the time required for a signal to reach the other receiver and the propagation speed, or unsynchronized nodes using round-trip times to estimate distance based on propagation speed. In the TDOA technique, reference nodes determine the time difference between received signals from a common node to form hyperbolic regions that constrain the possible location of the source node, from which distance estimates can be obtained. In angle-based schemes, the relative angular information between nodes is required. This is achieved through the use of antenna arrays or directional antennas. Range-free approaches assume that no specialized angle or range-determining hardware is necessary for the sensor nodes.

To determine the absolute geographical location, most of the localization algorithms also assume the use of special anchor nodes. Each anchor may be equipped with a GPS receiver to obtain its absolute position information. It is generally true that distributed algorithms are more robust and energy efficient than centralized algorithms. In each
Chapter 1. Introduction

Since different localization algorithms perform well under different assumptions, it is difficult to determine which one is the optimal scheme. For most of the performance comparisons reported to date, the common performance metric is the *localization error normalized with respect to the radio range*. This metric depends on the node connectivity (density), anchor population, and the distance measurement error models. Thus, the main objective of our work is to design an accurate localization algorithm which requires a small number of anchor nodes. The second objective is to design a localization algorithm which is accurate, scalable and energy-efficient.

These objectives are met by two different schemes we developed. First, with the objective of accuracy, a range-based *ordinal* multidimensional scaling (MDS) distributed algorithm called MDS-MAP(P, O) is proposed and analyzed [19]. Second, a range-free scheme called Concentric Anchor-Beacons (CAB) [20] is proposed and analyzed. This CAB scheme is proposed in two different implementations using Equal Area (CAB-EA) and Equal Width (CAB-EW). The MDS-MAP(P, O) and CAB algorithms encompass the two different extremes in the sensor localization problem and provide more accurate positioning of nodes than some other schemes proposed in the literature.
1.2 Structure of the Thesis

This thesis is structured as follows. In Chapter 2, a literature survey of the various localization schemes previously proposed is presented. These previous schemes are arranged in a taxonomy that identifies the key features that distinguish certain schemes from others. Additional schemes are included that provide insight into future research directions for localization in sensor network, for completeness. Chapter 3 describes our proposed ordinal multidimensional scaling (MDS) scheme, complete with: motivations and assumptions, the MDS-MAP(P, O) algorithm, the performance evaluation of the scheme followed by a summary of its contributions. Chapter 4 describes our proposed concentric anchor-beacons (CAB) scheme. The motivations and assumptions are stated, followed by a detailed explanation of the scheme. This is then followed by a performance evaluation of the algorithm, a discussion of possible extensions and lastly a summary. The conclusion in Chapter 5 outlines the main contributions of the thesis, summarizes the localization area factors to be optimized and explains future trends in the area.
Chapter 2

Related Work on Localization

In recent years, the problem of localization in wireless sensor networks have been attempted from many different points of view. Most of the previous works have made some assumptions on the environment of the networks that differ amongst others. This has led to diverse techniques being used in localization, such as the availability of careful placement of anchor nodes in a network, the possibility of performing operations centrally and then propagating results back to the nodes. The use of range-based techniques by measuring the received signal strength, time of flight and time difference of flight, versus angle-based techniques using antenna arrays and beam finders, have also been explored [15]. This chapter introduces the key papers in sensor localization in the following order: connectivity-based algorithms, range-based techniques, angle-based techniques, range-free techniques, and other novel techniques. One thing is clear however, no single system is the best in all cases. Currently, each application may have to determine the best technique to be used. In this chapter, we provide an overview of several localization algorithms. Survey papers can also be found in [14][15][21][22][23].
2.1 Connectivity-Based Algorithms

The connectivity-based algorithms allow sensor nodes to obtain position estimates without the requirement of any range or angle measurement hardware. These algorithms rely on the information of which nodes are within communication range of each other, and solve for positions using two similar mathematical approaches. The first proposed scheme is based on convex optimization, which is a centralized method. The second scheme that has been proposed as a centralized method and recently extended to a decentralized case, is based on the psychoanalysis mathematical tool of multidimensional scaling.

2.1.1 Convex Optimization

In [3], Doherty et al. considered a sensor network made up of anchor and normal nodes. The anchor nodes are required to impose absolute position constraints on nodes, whereas normal nodes impose relative position constraints. The constraints are based on the maximum range of communication between nodes, or measured range. If two nodes are able to communicate then, they each must lie within each other’s transmission range, hence defining a circular area of possible positions with reference to each other. Additional connectivity with other neighboring nodes reduces the possible position of a node to the intersection of the individual constraint sets, as seen in Figure 2.1.

These communication constraints imposed are convex and can be mathematically interpreted as linear matrix inequalities (LMI). The LMI are combined to form a semi-definite program (SDP) [3]. Semi-definite programs are generalizations of linear programs...
where the objective function is optimized subject to LMIs as opposed to linear inequalities. The nature of the solution set being circular regions, the constraints cannot be written as linear inequalities, but rather two-dimensional quadratic inequalities that can be represented as an LMI. Thus, constructing the connectivity constraints in the entire network, and characterizing them as an SDP, allows the solution to be computed since many numerical techniques are available to solve SDPs. The drawback of this approach is the centralized computation required to assemble the constraints and solve them before propagating the solutions back to the nodes. Final position estimates of the nodes are obtained from placing a bounding box around the intersection of the constraints on the node’s positions and then choosing the center of the box as the estimate. The nature of the model, requires that anchors be placed near the corners and edges of the network for optimal sufficient performance. It was found that using measured range as opposed to fixed node range improved accuracy of positions, as well, imposing additional angular constraints improved the algorithm at the cost of more information being gathered and computed. In addition, higher connectivities result in more accurate estimates; however networks with greater than 2000 nodes are deemed to be too computationally intensive.
to solve.

2.1.2 MDS-MAP

In [1], Shang et al. proposed a localization scheme based on multidimensional scaling (MDS), a psychoanalysis tool used to place objects in space in order to visualize their relationship based on similarity or dissimilarity measures. These measures are treated as distance-like data and used to construct the model, where \( m \) objects can be placed in \( n \)-dimensional space in order to satisfy the data. However in order to visualize and interpret the data, usually a 2D or 3D embedding of the objects is desired. Several different types of MDS techniques exist. If distance-like measures are available, then Metric MDS is used. If only ordered relationships exist (i.e. distance between objects \( A \) and \( B \) is greater than distance between objects \( A \) and \( C \)), then Non-Metric (Ordinal) MDS is applicable. Other types of MDS include Probabilistic MDS where objects are placed in space according to their probability distribution; Replicated MDS where measurements are obtained from several different objects’ point of view; and Weighted MDS where different dimensions have different weights [1]. Shang et al. applied the Classical MDS, (i.e. the original Metric scaling technique), to solve for the positions of the sensor nodes in the 2D case. The MDS result is the optimal least squares configuration to satisfy the distance constraints between the nodes. The nature of MDS is similar to that of convex optimization, where constraints between nodes are used to obtain a satisfactory solution. The difference between metric MDS and convex optimization lies in the fact that in
convex optimization the constraints limit the maximum distance between two nodes, whereas MDS does not. The main drawback of MDS is the fact that it is a centralized scheme. The MDS-MAP algorithm described in [1] is as follows. First, the shortest paths between all nodes in the network are computed. These distances are used to build the matrix used by classical scaling to perform MDS, in obtaining the relative positions between all nodes. Lastly, in order to obtain absolute coordinates for the nodes, an alignment transformation is performed using anchor nodes (which know their positions, through GPS or manual configuration).

As opposed to the previous scheme in Section 2.1.1, anchors as well as nodes were randomly placed in the square coverage area. Results showed that with fixed range information (i.e. no distance information between nodes) the error was 46% of the range, whereas with range information the error reduced to 24% [1]. This was obtained with a connectivity of 12, and only 2% anchors, including 5% range estimation error. Also observed was that at low connectivities (less than 9), the performance of connectivity and range estimation was approximately the same. In addition, with connectivities greater than 6, more than 93% of all nodes are localized, though at lower connectivities the position errors are much greater. Lastly, MDS obtained much more accurate results where networks were placed in a grid-like fashion with various placement error models.
2.1.3 MDS-MAP with Patches and Refinement (P, R)

In [2], Shang et al. proposed a decentralized scheme using MDS scaling, and included an additional step in the algorithm by incorporating a refinement scheme to further improve position estimates of nodes [2]. Whereas the work in [1] introduced a centralized MDS-MAP scheme which did not perform well in irregular shaped networks (i.e. non-square coverage areas), the new distributed method in [2] is aimed at improving the MDS performance in order to make it more robust under different network scenarios.

The algorithm performs distributed MDS by first forming local maps at each node of the network and then combining the maps to form a global map. The size of the local maps is controlled by the hop count parameter, where nodes only consider other nodes that are within the specified number of hops from their location. First, each node determines which nodes are within its local map. Then, as in centralized MDS, the shortest path distances between all nodes in the local map are computed, and MDS is performed on the constructed distance matrix. Then, a refinement procedure is used to perform the least-squares minimization to improve errors due to inaccurate shortest path measurements, by weighting the distances between one-hop neighbors more than those of two hop neighbors.

Next, the local maps are sequentially merged, starting from a randomly picked local map, to form a global map. An optional least squares minimization refinement is also performed on the global map. Lastly the map is converted to an absolute map by transformation using three or more anchor nodes in the entire network.

The results show that the refinement steps are very computationally intensive and
take the majority of computation time during local map refining. The size of the local maps is determined by the complexity of the MDS calculation. For random networks with connectivity greater than 12, a two-hop neighborhood was determined to be optimal. For regular and irregular randomly and uniformly placed topologies, the MDS-MAP(Patch, Refinement) scheme consistently achieves errors less than 30% of radio range for connectivity only information and less than 20% of radio range for distance information [2]. The drawback of this approach is the significant refinement computation time, the propagation of errors from local maps, and sequential merging of local maps. In addition, the classical MDS algorithm performs well under high connectivities, but at low levels other MDS techniques such as ordinal MDS may be more accurate. Compared to other schemes, this algorithm is more robust under irregular topologies and more accurate for the low percentage of anchors present in the system.

2.1.4 Anchor-Initiated MDS-MAP, On-Demand MDS

In [3], Ji and Zha proposed a distributed MDS scheme in order to localize nodes in a sensor network, using the classical MDS algorithm. Also, they proposed an on-demand algorithm for a sensor node to obtain a position estimate of itself without having to localize the entire network first. In contrast to the previous distributed scheme, the absolute positions of the nodes in the network are obtained sequentially as the algorithm starts. One anchor named starting anchor node in the network will flood its position to the network for other anchors named ending anchors to receive and reply back with
their positions and the path of nodes traversed to reach them. Average hop counts between starting and ending anchors can be calculated in order to determine anisotropic environments in different directions of the network. Thus, the average radio ranges are computed from the hop counts and distances between the anchors. These ranges are used as distance estimates between nodes along certain paths. Then, the starting anchor estimates the positions of the nodes that are on the paths to the ending anchors. Using these estimates, the local maps around each node on these paths are computed using MDS of the pairwise distances in the local map and aligned relative to the previously estimated positions of the nodes on the path. This sequential localization proceeds down the path to the ending anchor, resulting in all positions on the path and one hop away from the path being localized as in Figure 2.2.

Nodes further away are localized using pairwise distances to 3 or more nodes that are already localized, and nodes with only two localized neighbors perform iterative MDS to compute position estimates. The results show that for 10% anchor population and 12.6
connectivity level, the error rate is less than 30% of radio range for up to 25% distance measurement error, in a randomly deployed square coverage area. In a square region with different radio ranges in different areas, the accuracy degrades to less than 35%, which proves that this algorithm is robust in dealing with anisotropic environments. The on-demand algorithm allows any sensor node that needs to be localized, to flood the network to at least the 3 nearest anchors and obtain the paths of nodes traversed to reach the anchors. The initial position of the node is obtained by trilateration using the anchors’ positions. MDS is then used to compute relative positions of the nodes neighbors that are on the path to the anchors. Then, MDS is sequentially applied to the neighborhoods of the path towards the anchors. Thus, once the relative positions of the anchors are calculated, an alignment procedure between actual and relative positions is applied and propagated back to the initial node for an accurate position estimate as in Figure 2.3.

With distance measurement error of 25%, 8% anchors, the localization error is less than 40% in anisotropic environments [4].

2.2 Range-Based Techniques

Range-based techniques make extensive use of the distance information to obtain position estimates of the sensor nodes. The algorithms presented in this section show many different approaches to the node localization problem, but have inherently similar architectures in design. One of the predominant architectures is the 3-step node-positioning scheme identified by Langendoen et al. [16]. In the first step, nodes obtain distance estimates
to at least 3 anchors according to algorithm specific details. The second step consists of a rough estimation of the node based on the distance information to the anchors (e.g., lateration, centroid computation, or bounding box). Lastly, a refinement procedure is executed in order to more accurately position all of the nodes from additional information such as inter-node distances and positions. Though quite a few exhibit this architecture, others exhibit other novel schemes that in certain applications may be useful.

2.2.1 TERRAIN using ABC

These two schemes, developed by Savarese et al. [24], were one of the first proposals for distributed sensor positioning algorithms. Their model assumed a dense distribu-
tion of sensor nodes accompanied by a small number of anchors, (e.g., four were used in simulation). Upon random sensor deployment, the sensor nodes have the additional capability to vary their transmission power in order to remain well connected with the network. Hence, the sensor nodes can themselves control their level of connectivity with the network, a unique assumption. The first function which TERRAIN (Triangulation via Extended Range and Redundant Association of Intermediate Nodes) employs is the Assumption Based Coordinates (ABC) algorithm. In this algorithm, a node establishes itself as the origin of the coordinates. The first node that sends its ranging estimate to this node is placed on the x-axis at the range estimate value. The second node’s position in the coordinate system is explicitly solved using the range estimates to the original two nodes and placed in the positive y-axis coordinates. Further nodes are then placed in the relative system from their range estimates to existing nodes. This function thus iteratively places all nodes on a relative coordinate system. TERRAIN, first uses the ABC algorithm only at the four anchor nodes in the system. Once sensor nodes receive the propagated distance measurements from the four anchors, the least-square lateration is performed to obtain initial position estimates. Then, a refinement scheme called Iterative Local Triangulation is performed locally between every node and its immediate one-hop neighbors. Each node uses the neighbors’ position estimates and range measurements to the neighbors to locally laterate its position. This procedure, after many iterations, should result in nodes positions remaining fixed.
2.2.2 APS

In the Ad Hoc Positioning System (APS) developed by Niculescu and Nath [5], a similar approach to TERRAIN is used without a refinement procedure. Their approach to the position problem uses an aspect of distance vector (DV) routing to propagate range estimates of the anchor positions to the nodes. In the DV-Hop case, once the network is deployed, the anchors broadcast their positions with a hop count value of one. Each one-hop neighbor of the anchor receives the packet, stores it in memory and creates a new packet with the anchor position and a hop count value of 2. It then forwards it to all of its one-hop neighbors. Thus all nodes should be able to obtain at least three anchors’ positions and the hop counts to them. In the case where nodes obtain packets containing the same anchor but different hop counts, the lowest hop value is kept. When anchors receive information about other anchors, an average hop size is calculated, by computing the distance between the two anchors, and dividing it by the respective hop count. This average hop size is then propagated to the other nodes, so as to provide the nodes with range estimates to the anchors by multiplying their hop counts by the average hop value for each anchor. At this point, each sensor performs a lateration computation to localize itself using the positions of at least 3 anchors and the estimated distances to them.

The DV-Distance algorithm is exactly the same as DV-Hop, but it is assumed that sensor nodes are capable to obtain range estimates to other nodes [5]. Thus, instead of forwarding hop count values, the sum of the distances originating from the anchors is propagated to all nodes, providing range estimates directly as opposed to multiplying
hop counts by average hop distances. The third APS scheme is the DV-Euclidean which uses geometric quadrilateral properties to position nodes, one at a time. As shown in Figure 2.4 if node A has range values to B and C, both of which have range values to an anchor L, as well as a range value between themselves, then A can directly compute the Euclidean distance to the anchor L, and hence position itself between two possible locations A and A’.

Considering relations between A and its other neighbors solves for this ambiguity. Comparison of the schemes showed that for anisotropic environments, DV-Euclidean performed the best with errors of less than 50% of radio range, with 20% anchors and 10% range estimation error. The DV-Hop/Distance schemes performed well at relatively low connectivity (7), and isotropic environments with error less than 30% of radio range, 20% range estimation error, 20% anchors. The main benefit of this scheme is its low signaling complexity.
2.2.3 Self-Positioning Algorithm

In the self-positioning algorithm [25], a relative coordinate system is constructed, without the use of anchor nodes that traditionally use GPS positioning. Similar to the ABC algorithm employed in the TERRAIN scheme, first every node identifies its one-hop neighbors and obtains distance estimates to them preferably by some time-of-arrival (TOA) techniques suggested by the authors. TOA techniques have been determined to be more accurate than received signal strength (RSS) techniques. Then, this information is sent to all of the one-hop neighbors, essentially providing every node with information of up to a two-hop neighborhood. Each node then assumes itself as the center of the system, and sequentially places all nodes it has information for, into its coordinate system, as in ABC. At this point, all nodes have placed their one and two-hop neighbors in Cartesian coordinates centered upon themselves. As opposed to aligning all coordinate systems to a single node’s system that may require network wide updates if the node is moved, the alignment is performed relative to a group of nodes. A Location Reference Group (LRG) of nodes are chosen, of which a coordinate system is developed as the main system to which all network nodes must align to. This is used to enable the presence of mobile nodes in the system. However it comes at the cost of extensive communication and data exchange.
2.2.4 Ad-hoc Location System (AHLoS)

This distributed algorithm [6] uses three functions to produce an accurate fine-grained approach to sensor node localization. These functions combine to produce absolute node positions important for noting origins of events, routing, group querying and network coverage. The design goals included the ability for the network to be deployed indoors or outdoors, decentralized and use maximum-likelihood estimation of node positions. The network consists of both anchor and regular sensor nodes. Both of which have additional hardware to use time-based range measurements as opposed to RSS. The first function, atomic multilateration, can be used for nodes that have 3 or more anchors as neighbors. These nodes obtain position estimates by lateration, essentially solving a system of linear equations. If there are more than 3 anchors, then the position solution is the linear least squares solution of the overdetermined set of equations. For nodes that do not have at least 3 anchor neighbors, the iterative multilateration approach is used. In this approach, nodes that are unable to perform atomic multilateration wait until other nodes have obtained their positions. Thus, those nodes that are successful with atomic multilateration behave as anchor nodes. The remaining nodes then use the presence of traditional anchors with these newly positioned anchors to perform multilateration. This iterative scheme, will thus gradually resolve the positions of the majority of nodes in the network. The drawback in this scheme is the potential error propagation by using approximated anchors in iteratively solving node positions.

Lastly, collaborative multilateration is used in cases where unpositioned neighbor
nodes cannot laterate estimates by themselves since not enough resolved neighboring
nodes are present, but the combination of their relations to other nodes can be used to
solve for each of their positions, as in Figure 2.5. Thus, these schemes, involve the use
of information greater than one-hop away, and is useful when the percentage of anchors
nodes is low locally.

In Figure 2.5 node A and B cannot resolve positions since each only has two resolved
neighbors, but combining their information a system of four equations and two unknowns
is used to solve each node’s position. Of course, some nodes may not be able to position
themselves at all if they have less than three neighbors, as the position will have ambi-
guity, the algorithm includes a test of participating neighbors to identify if collaborative
multilateration can be used or not [6]. The main drawback of this approach is the rel-
atively high percentage of anchors needed in the system to provide sufficient accuracy.
This requirement can only be avoided by increasing the node density in the network that
will increase the iterations required to localize.
2.2.5 N-Hop Multilateration Primitive

In this scheme [26], the positioning problem is generalized as a collaborative multilateration approach as introduced in [6]. Anchor nodes are either placed in the network area or are capable of automatically establishing a coordinates system for the other nodes. The algorithm consists of three phases. In the first phase, collaborative subtrees are constructed and the unique solvability of each unknown node is determined. Collaborative subtrees are explicit combinations of anchors and nodes that yield an overdetermined or exact number of non-linear equations to solve for the positions of the unknown nodes [26]. In addition, before computing solutions to these equations, tests are performed to determine if each unknown node satisfies the conditions necessary for unique positioning. The search for the possible uniqueness of the node extends from one-hop to two-hops to n-hops, in order to identify enough equations to solve for the nodes as well. Thus, this phase is an extension of the previous algorithm’s collaborative multilateration approach to n-hops. Nodes that do not satisfy the conditions do not proceed to the next phase. The conditions for this uniqueness are summarized in the following rules.

1. A unique solution exists if the node has three neighbors with uniquely possible solutions.

2. The neighbors must not be co-linear, since this would result in multiple solutions after lateration.

3. For two nodes using each other as constraints, each node must also have another
neighbor to which the other node is not directly constrained.

The second phase, obtaining initial position estimates, uses bounding boxes to set the extreme possible locations of the node. The bounding box is constructed by taking the strictest constraints on node positions by the reference neighbor nodes determined in the collaborative subtrees algorithm. This approach is used as opposed to direct lateration due to its computational simplicity. The position of the node is then determined as the center of the bounding box. Phase three then performs refinement using the Kalman filter that is the least squares equivalent in the static network case. This refinement is performed distributively using approximated methods that control the global solution of each subtrees in order to reach global minima and obtain accurate positions. After this procedure, nodes previously excluded in phase one are solved for using atomic multilateration. The drawbacks of this scheme are the extensive computations in the refinement scheme, and its rapid increase in error with increased network size, in addition to requiring greater than 20 percent of nodes to be anchors for sufficient performance [26].

2.2.6 Hop-TERRAIN with Refinement

An algorithm similar to that of APS, and an extension to the TERRAIN approach, Hop TERRAIN with Refinement [27] is an amalgamation of both schemes. As the name suggests, two phases are present in the scheme, and initial start-up phase followed by a refinement step. The design of this system assumes a very low percentage of anchors, with
at least four being required in the network. The other main goal is to avoid the inaccurate ranging errors typical of RSS based approaches in obtaining positions estimates. The result is the use of anchors propagating their positions to all nodes, with receiving nodes noting the hop counts to the anchors and forwarding anchor positions to neighbors. When an anchor receives information from another anchor, an average hop distance is calculated as in APS. This average distance is propagated throughout the network as in APS. The difference in this implementation, however, is that once a node receives information about average hop distance, it uses the information for all hop measurements to the anchors it knows, not only for the anchor that sent the information. This reduces the information being sent throughout the network. After obtaining at least 3 distance estimates to anchors, nodes use lateration to compute the initial positions estimates. As new information from other anchors not previously used or closer estimates to known anchors are received by the node, additional laterations are performed to obtain better initial estimates. After this initial startup phase, all nodes will have position estimates. The refinement procedure consists of nodes exchanging initial positions with neighbors and obtaining range estimates to neighbors through a ranging technique. Then nodes perform local least squares lateration to improve initial position estimates. In addition, all nodes contain confidence weights indicating the certainty of their position estimate \[27\]. Anchor nodes have weights of 1.0 while other nodes have weights equal to the average of the weights of the surrounding neighbors. Thus nodes closer to anchors have higher confidences in their positions than others. Lastly, nodes that have confidence
levels less than 0.1 are regarded as unpositionable. Other constraints are used to make sure refined positions meet the initial communication constraints imposed by the anchors estimated distances, and hence have not diverged during refinement. In general, the refinement scheme improves the initial estimates by at least 3 times the error rates. For an anchor population of 20%, range error of 5%, connectivity of 7, the position error is 30% of the range \[\text{[27]}\]. As range error increases, the improvement provided by refinement decreases linearly, and at 40% range error the improvement is non-existent. This scheme has proved to be more accurate in lower connectivities compared to other schemes, and achieves performance comparably at high connectivities with others.

\subsection*{2.2.7 APS/Hop-TERRAIN/N-hop Multilateration Primitive}

In the paper by Langendoen and Reijers \[\text{[16]}\], a comparison of three distributed schemes that have been deemed to be the most robust distributed schemes of the ones proposed to that point, are compared. The authors identify the key similarities in the schemes as the 3 phases being incorporated in each. First, distances between nodes and anchors are obtained. Then, the initial estimates are computed. Lastly, a refinement is performed to achieve more accurate results. In the initial phase, three possibilities are available as proposed by APS: DV-Hop, DV-distance and DV-Euclidean. The second phase has two options, lateration previously used by APS and Hop-TERRAIN, and min-max the bounding box scheme used by the N-hop primitive. Refinement was used as the third phase. The results from the simulations of the different combinations of the three phases
provided several conclusions. In the first phase, DV-distance always overestimated distances to anchors, DV-Hop is immune to range error but does not perform well under irregular topologies, and lastly DV-Euclidean rapidly decays as range errors increase. In the second phase, lateration is better than min-max for precise range measurements but more computationally intensive than min-max. In addition, min-max requires anchor nodes to be located at edges for better performance. The combinations of DV-distance and min-max performed comparatively to DV-Hop/Lateration both achieving accuracy less than 25% of range for 10% range errors, connectivity of 12, and 5% anchors [16]. However, in both cases the number of nodes localized drops by approximately half due to refinement divergence and nodes not uniquely localizable. At low connectivities (less than 9) DV-Hop/Lateration is best, while medium connectivities (less than 15) DV-distance/Min-Max is superior. At high connectivities DV-distance/Lateration performed well. DV-Euclidean was only worthwhile at zero range measurement errors and connectivities larger than 14. The main drawback of this family of schemes is the percentage of nodes achieving accurate positions, since refinement results in almost half the network’s nodes not achieving acceptable positions, at the cost of improving the accuracy of other nodes’ positions by more than half.

2.3 Angle-Based Techniques

As opposed to range-based schemes, angle-based schemes provide several advantages. These advantages depend on the type of angle or bearing techniques used. For example,
whereas lateration using three anchors is needed for range-based techniques, two anchors are needed for angulation using bearing information. If angles between anchors can be determined instead, then three anchors are sufficient in triangulation, the counterpart of trilateration. Though range-based techniques have been extensively proposed, several algorithms using some angle measurements have been proposed.

2.3.1 APS using AOA

The extension of the APS scheme to involve the use of angle-of-arrival (AOA) measurements at the nodes has been implemented in [7]. All sensor nodes are assumed to have additional hardware capable of determining the AOA of signals from other nodes. These nodes can determine the direction of the signal relative to their own reference direction known as their heading. If these headings are absolute directions relative to the North direction, and hence behave as compasses for the nodes, the localization is even easier, as shown in Figure 2.6.

The hardware to achieve this AOA capability arises from the use of antenna arrays at each node and computations relating to signal strength at the antennas of the array and received signal time difference between the antennas. If a node can obtain a bearing to three anchors, then it can triangulate its positions using the angles between the anchors and the anchor positions as in Figure 2.7.

The mathematical solution of triangulation is transformed into a trilateration problem and solved using the least-squares algorithm. If nodes’ headings are relative to North,
Figure 2.6: Nodes obtaining relative angle measurements to other nodes. \[7\].

then only two measurements are needed, each providing the equation of a line and the intersection determining the position of the node.

The two schemes proposed are DV-Bearing and DV-Radial \[7\]. Whereas bearing refers to the direction of a node from the originating node’s point of view, the radial is the reverse bearing (from the other node’s point of view). This distinction is made when the headings of the nodes are not absolute (i.e. relative to North). If they were absolute, then the information would be redundant. The DV-Radial case uses more communication as two packets of information are exchanged, but the resulting computations are less. In both schemes, nodes with anchor neighbors obtain bearings or radials to the anchors. This information is propagated to other nodes, which are then able to obtain bearings/radials to the anchors as well. However, in order for a two-hop neighbor to
obtain its bearing to an anchor, it must have two neighbors that have bearings to the same anchor. Finally, once nodes have bearings/radials to three anchors, they can compute their positions. The increased need of neighbors with measurements to the same anchor, increases the node connectivity to 9 or greater for AOA to produce sufficient results. The errors present in this system are inherent in the degree of accuracy in small angles as opposed to large ones. Of the two schemes, DV-Radial is less dependent on percentage of anchors needed, and obtains better accuracy than DV-Bearing. However, the accuracy is comparable to the radio range between nodes, not as precise as most range-based schemes.

2.3.2 DV-position

Another scheme proposed by Niculescu and Nath includes both range and AOA measurements, and allows nodes to position themselves in one step if anchors have headings
relative to North [8]. As shown in Figure 2.8 if node A has the bearing to anchor B and the anchor knows its bearing to node A, as well as its heading from North, then the equation of the line connecting A and B can be computed. In addition, range information between A and B will select one point on the line, corresponding to node A’s position.

This is an example of multimodal sensing, that combines two techniques of measurement to provide less requirement in terms of cooperation with other nodes, at the cost of additional sensor hardware. This scheme is optimal at anchor percentage of 10%, node connectivity higher than 6, with error rates less than 40% radio range. However, this is not practical due to the size restrictions imposed by sensor nodes and the hardware requirements needed by the nodes. Also, AOA measurements have been scrutinized over possible complications over reflections from objects, thus introducing considerable error into the system.
Chapter 2. Related Work on Localization

2.3.3 Directionality Localization

The directionality localization algorithm \cite{9} employs the use of at least three fixed high-powered directional anchors placed at the corners of a sensor network. It is assumed that all nodes in the network can receive signals from every anchor when the direction of transmission of the anchor is towards the sensor node. In addition, the anchors are synchronized to be rotating their directional beacon signals 360 degrees at the same constant speed but with a fixed angle offset from each other, as shown in Figure 2.9.

Nodes must be able to distinguish between the beacons received from the different anchors. The algorithm begins with each sensor node noting the times at which it receives each signal from the specific anchors. The time differences between the received signals can be translated to the angles between the anchors, given that the nodes know the

![Figure 2.9: Localization using synchronized, offset, rotating beacon signals, \cite{9}](image-url)
offset and speed at which the anchors are rotating the signals, which is predetermined to sensor network deployment. Then the location of the node can be determined using two of the angles and the positions of the anchors using trigonometry. Simplistic in nature, the complexity in the anchors is offset by the simplicity of sensor nodes not requiring any special hardware. However, this scheme is not ad-hoc, though it is distributed and suffers from the possibility of multipath reflections, no line of sight between all nodes and anchors, and the precision (width) of the directional beam.

2.3.4 Ad-Hoc Ranging and Sectoring

Another scheme that promotes the use of multimodal sensing in order to solve the node-positioning problem is proposed by Chintalapudi et al. [17]. The authors proposed this scheme by pointing out that range-based schemes by themselves require much higher connectivities than needed in general for the sensor network to perform its actual sensing function. From the schemes presented in section 2.2, one can see that position errors less than half the radio range typically require connectivities greater than 9. Chintalapudi et al. state it is unacceptable to deploy so many extra nodes just to obtain the nodes’ positions. They aim to reduce the connectivity required as well as keeping the percentage of special nodes (anchors) at a minimum. By using range and sectoring this requirement can be alleviated, to needing only one neighbor that has positioned itself, as explained in DV-position. Thus the scheme proposed is similar to iterative multilateration, but in this case neighbors to anchors position themselves first followed by other neighbors. To
combat error propagation, refinement schemes are proposed such as least-mean-squares. The results show that for sectors as large as 60 degrees, with node connectivity of 5 and 20% anchors, more than 90% of network nodes are localized with 6% localization error [17]. These results are the most accurate of all ad-hoc systems, though the physical feasibility of such a system is questioned. In addition, a large number of iterations are required to achieve these results during the refinement procedure, requiring a high computational capacity for such sensor nodes.

2.4 Range-Free Techniques

Range-free approaches assume that no specialized angle or range-determining hardware is necessary for the sensor nodes.

2.4.1 Centroid

Bulusu et al., proposed a simple, cost effective, RF-based positioning method called Centroid in [18]. By design, the technique is classified as coarse grained since it computes localization using proximity methods and simple averaging to compute the centroid location. Fine-grained localization, in contrast, uses measurements or fixed ranges obtained from the sensor node to compute position estimates. In this technique, anchors are placed in a coverage area in a symmetric grid pattern. The transmission range by the anchors is assumed to be greater than those by normal sensor nodes, similar to the APIT [10] algorithm. The coverage areas by each beacon overlap, providing sensors with the ability
to hear several beacons from different anchors at any location in the coverage area. The anchors are synchronized to transmit one beacon during a specified time slot of a period $T$ of system operation. Thus, the period depends on the number of anchors deployed. Sensor nodes determine their connectivity by listening to beacons for several consecutive time periods. An anchor is determined to be within range, if the node receives more than a threshold number of beacons during the listening period from the anchor. The position of the node is then calculated as the average of the coordinates of each anchor within range, and hence is noted as the centroid or center of the anchors’ positions. The scheme enables sensors to operate with minimum power consumption as no communication is needed with other nodes, and very little computation is required. However, the density of the anchors in the coverage area determines the accuracy of the scheme. In addition, the deployment of anchors in an ad-hoc manner, severely affects the positioning accuracy as well. The synchronization of the anchors may be unreasonable for large-scale deployments.

2.4.2 Approximated Point-in-Triangulation (APIT)

APIT [10] is range-free and does not require any additional hardware for sensor nodes. The design goals included performing under irregular radio patterns and having low communication overhead. The scheme assumes the availability of high powered anchors that are able to transmit much greater distances than normal sensor nodes. The main algorithm employs a unique test for localization, the Point-in-Triangulation (PIT) com-
Figure 2.10: Local grid maintaining overlapped triangle regions for localization, [10].

Each node is assumed to be able to hear many anchors (much greater than 3) and the connectivity is assumed to be higher than 6. The PIT comparison tests if a node is located within each triangle formed by combinations of 3 anchors heard. A grid is maintained with positions in the grid satisfying the PIT test being incremented while those outside being decremented, as shown in Figure 2.10.

After considering all combinations of anchors heard, the grid points with the highest values determine the position of the node. A node is located within a triangle of anchors if there is no neighboring position where it can be moved, that is not simultaneously closer or farther from all three anchors. This movement is avoided by just considering the position of the node’s neighbors as the new position of the node and by comparing if they are closer or farther from all anchors at the same time. This is the Approximated PIT (APIT) test as shown in Figure 2.11, which does have several degenerate cases that
only account for 14% of the deployments actually being affected by the error.

Though distance measurements are not used, each node in a table maintains the received signal levels of the anchors. Each node maintains its table and exchanges it with the neighbors so that all nodes have information of the one-hop neighborhood. Then APIT looks at a column to determine if all the nodes with the same three anchors lie within the triangle, (i.e. the signal levels at all three anchors are not simultaneously greater or smaller than the nodes’ levels). All combinations of anchors are examined and then the grid array is constructed with 0.10 radio range blocks. The position is then determined to be the center of the highest valued points area. The required reduction in anchor percentages is inversely proportional to the anchor to node range ratio (ANR). The scheme achieves errors of 40% range, for node density of 8, anchors heard of 16 and ANR of 10 \([10]\). All of this comes at moderate communication cost and robust performance to irregular radio patterns and anchor positioning error.
The work in [28] considered the possibility of quantized received signal strength (RSS) and presented mathematical analysis of the accuracy with varying levels of quantization. The performance comparisons between ROCRSSI and APIT schemes are given in [29].

2.5 Other Novel Techniques

Other schemes exist that cannot be classified solely as one of the type of schemes mentioned earlier. These localization algorithms provide unique ways of obtaining node positions and assume quite different network characteristics than previously mentioned schemes.

2.5.1 Time-based Positioning Scheme TPS

A fixed anchor deployment scheme proposed by Cheng et al. is the Time-based Positioning Scheme (TPS) [11]. In this scheme no time-synchronization is required between anchors. The anchors are deployed outside of the coverage area, enclosing the area in a triangular geometry, as shown in Figure [2.12].

The anchors’ transmission range is large enough to cover the entire area, hence any node within the coverage area will be able to hear all three beacons from the anchors. First, anchor A sends a beacon to the area. When anchor B hears the beacon, it sends its own beacon containing the information of the time between receiving A’s beacon and sending its own beacon. Then anchor C also hears A’s beacon and sends its own beacon containing similar information like B, but using C’s reception and broadcast time.
difference. All three of these beacons are received by the nodes in the area, and the nodes will maintain local time information of when the packets from the anchors are received. This constitutes one beaconing period for the system. This information is enough for nodes to triangulate their position based on geometric equations relating to the time differences between beacons and reception by the nodes. The accuracy of the position depends on the accuracy of the timing measurements, and can be improved merely by nodes averaging their results over several beaconing intervals. Errors presented in the system include system delay in reception and transmission of beacons, non line-of-sight transmission and multipath fading. In addition, if a node is much closer to one anchor than the other two, significant error is present as well as the case where the node is much farther to all three anchors \[11\]. Similar to Centroid, the nodes are able to position
themselves without communication to other sensors, but more computations are required compared to Centroid. This is compensated by the lack of synchronization required by the anchors and nodes, since only time-difference of arrivals (TDOA) are used to compute distance estimates to anchors and the following trilateration. Thus, this is a fine-grained scheme as distance measurements are used as well as the anchor positions. The increased transmission ranges required by anchors may be troublesome for large-scale deployment and may require more anchors and coordination among subtriangular regions.

2.5.2 Secure Positioning

A different area not previously considered in the node localization problem, is the security of the network and its communications, in the event of it being deployed in a hostile environment as suggested in many papers. In [12], several methods of obtaining sensor locations in a secure manner are employed. The threats inherent in a sensor network include the removal of sensors, manipulation of sensors and the compromise of the system. In order to counteract these issues, the anchors are assumed to be tamper resistant and physically immovable. A central authority where all network data is sent to is also assumed to be secure. In addition, the central system is able to maintain the location of all unique sensors in the system. Two notions considered are the introduction of malicious and compromised nodes in the system. Malicious nodes are controlled by attackers, but the central authority is aware of the malicious nodes. Compromised nodes, however, are controlled by the attackers and are thought to be normal nodes by the system. The
attacks considered to be possible include: displacement of a node, introduction of a worm-hole, malicious distance enlargements and dissemination of false position information. In order to prevent such instances from affecting a sensor network, the secure methods are meant to be used in conjunction with node localization algorithms previously proposed in the literature. In the authenticated distance estimation method, nodes communicate using cryptographic keys in order to obtain distance estimates to each other. Another method, authenticated distance bounding, enables each node to obtain distance bounds to other nodes independently. These methods are used to obtain verifiable trilateration for position estimates of nodes and verifiable TDOA for trilateration.

These methods are employed in the sensor system using anchors deployed in a grid-like fashion consisting of triangles, as shown in figure 2.13.

Sensor nodes obtain position estimates by the methods previously mentioned. Since
anchors are tamper proof, they will be able to obtain true distances to all nodes enclosed in their coverage triangle and thus provide true positions of nodes through authenticated communications, and therefore verify the locations claimed by the nodes. In the distributed sensor network, the use of Basic Distance Verification (BDV) will also provide for reliable positions of nodes. The algorithm determines the position of a node by observing its placement within a series of triangles enclosed by other nodes. However, in the case where more than one node is malicious or compromised, the system is subject to unreliable information. Results show that in order for a network to be more robust in terms of security, the node connectivity (density) must be higher than required for localization. In addition, the security of the system increases the communication and computations required by each node to perform localization.

2.5.3 Anchor-Free Localization (AFL)

The Anchor-Free Localization (AFL) scheme is another distributed algorithm for sensor networks [13], which omits the use of specialized anchor nodes always required by other schemes. The algorithm, implemented without the use of anchors results in the network discovering its topology and connectivity, and hence relative node coordinates to one another. If absolute coordinates are required, three anchors can be used to translate or orientate the AFL’s relative coordinates to achieve this. This scheme has two phases: fold-free embedding, and mass-spring optimization [13]. The objective of the first phase is to produce a map of the network that looks similar to the actual network topology.
This approach is emphasized as important in order for the second phase to reach a globally optimum solution that matches the actual topology. Previous methods of building local coordinate systems and aligning them, sometimes fall into obtaining local minima solutions and result in topologies that match distances between nodes but fail to produce topologies matching the actual case. The authors identify this as graphs that are not globally rigid, meaning that given observed distance measurements between connected nodes, several different topologies are possible to satisfy the given constraints, as shown in Figure 2.14.

The first phase of the scheme builds a reference coordinate system randomly, paying special attention to form a globally rigid system. A node is picked randomly as the starting point from which four extrema of the system are identified and the center node of the reference system is determined. Then all other nodes use estimated distances to these 5 points (excluding starting node), to place them in the coordinate system using
Figure 2.15: Topology construction for fold-freedom, [13].

The resulting embedding is usually larger than the actual topology, which aids in the avoidance of local minima. The second phase consists of nodes broadcasting their estimated positions to neighbors, and calculating estimated distances to neighbors using the aforementioned data. Comparing estimated distances with actual measured distances, the second phase optimizes the estimates of the first phase iteratively by estimating node positions that reduce the overall network error, until a solution is reached analogous to a mass-spring system [22]. The robustness of the scheme, lies in the fact that even at low connectivities AFL is successful in obtaining a coordinate system, whereas other incremental schemes fail.
2.5.4 Sequential Monte Carlo Localization (MCL)

In contrast to most schemes presented, Hu and Evans proposed a localization scheme for mobile sensor networks [30], in which all nodes are mobile. Their approach is based on the assumptions that no distance measurements can be made by nodes, anchor densities are low and nodes are deployed in an ad-hoc manner. The algorithm itself is based on the sequential monte carlo or particle filter technique, a statistical approach to the mobile node-positioning problem. In this technique, there are two steps, the prediction and filtering stage. The position of a node is maintained by a set of samples of the posterior distribution of the possible location of the node. In the first step, a prediction of the possible location of the node is made using the transition distribution probability and previous set of samples. The second step, involves the node using new information gathered after movement to eliminate previous prediction that do not correspond with the new data. To counteract the reduction of valid samples, resampling is performed to maintain a discrete number of samples for the posterior distribution of node location. The restrictions imposed in the first step, is due to the speed of the node, assuming constant speed \( v \), the location of a node given a initial starting point will be within a circle of radius \( v*t \), where \( t \) is the time from initial start [30]. Thus, the location is randomly picked with uniform distribution within the circle. Nodes, using information of which anchors were within range in the previous time, and which anchors are now within range after movement, will refine the samples generated from before. In addition, nodes share with neighbors, the anchors that are within communication range, to provide
further identification data. From the results, assuming a *random walk* node movement model, estimated errors as good as 50% of radio range are achieved for node density of 10, and anchor ratio of 10% \[30]. As more samples are kept, the greater the accuracy of the scheme and surprisingly, the faster the anchors and nodes move, the faster the algorithm converges in obtaining accurate position estimates. The last result is due to the fact that faster speed result in much more information on which anchors are within communication range of which nodes at certain times, resulting in more information being available to filter bad position samples.

Several schemes have explored the use of mobile anchors and nodes. In \[31\], a single mobile anchor traverses the network and allows stationary sensor nodes to compute their location estimates based on at least three neighboring nodes’ locations. Multiple mobile anchors are used in \[32\]. In \[30\], both anchors and sensor nodes are mobile. The Monte Carlo algorithm is used for localization. In \[33\], an extended Kalman filter (EKF)-based state estimator is used in tandem with mobile robots for localization.
Chapter 3

Ordinal Multidimensional Scaling

In this chapter, we propose the implementation of ordinal MDS for localization in wireless sensor networks. We first state the motivations and assumptions. It is followed by the description of our MDS-MAP(P, O) localization algorithm. We then present the simulation results for the performance comparisons between MDS-MAP(P, O) and MDS-MAP(P, C) \[2\].

3.1 Motivations and Assumptions

The advantage of MDS localization algorithms is the relative low percentage of estimation error while using a small number of anchor nodes. The classical (or metric) MDS algorithm assumes that there exists a linear transformation which relates the shortest path distance and the Euclidean distance between each pair of nodes. For each pair of nodes \((i, j)\), if the shortest path distance is denoted by \(p_{ij}\) and the Euclidean distance is denoted by \(d_{ij}\), then \(d_{ij} = mp_{ij} + c\) for some constants \(m\) and \(c\). Classical MDS uses *singular value decomposition* [34] to determine the relative coordinates of the sensor nodes.
As mentioned in Chapter 2, previously proposed localization algorithms based on classical Multidimensional Scaling (MDS) \cite{1,2,4} have proven to be robust with respect to both hop-based and range-based implementations. Only three or four anchor nodes are necessary to determine the absolute locations, in two or three dimensions, respectively. These MDS algorithms achieve a higher accuracy than some other schemes.

By using the similar terminology in \cite{1}, the term MDS-MAP(C) is used for the classical MDS localization algorithm. The original MDS-MAP(C) is a centralized algorithm. In \cite{2}, a distributed MDS algorithm called MDS-MAP(P, C) was proposed where P denotes the use of patching of local maps and C denotes the use of classical MDS. As mentioned in the previous papers, further work is required to study the application of other MDS techniques (e.g., probabilistic MDS, ordinal MDS) on localization in sensor networks. Our proposed scheme in this chapter is called MDS-MAP(P, O) where P again denotes the use of patching of local maps and O denotes the use of ordinal MDS. MDS-MAP(P, O) is also a distributed algorithm.

The main difference between classical MDS and ordinal MDS is that the former assumes there is a linear equation which relates the shortest path distance and the Euclidean distance between each pair of nodes, the latter simply assumes a monotonicity constraint. That is, for ordinal MDS, given two pairs of nodes \((i, j)\) and \((k, l)\), if the shortest path distance of \((i, j)\) is greater than that of \((k, l)\), then the Euclidean distance of \((i, j)\) is also greater than that of \((k, l)\), and vice versa.
3.2 MDS-MAP(P, O) Algorithm

The major steps of the MDS-MAP(P, O) algorithm are as follows:

1. Each node first gathers either the distance (for range-based) or hop count (for hop-based) information within its two-hop neighborhood.

2. In each node, the Dijkstra’s algorithm is invoked to determine the shortest path between each pair of nodes within the two-hop neighborhood. We use the notation $p_{ij}$ to denote the shortest path distance between nodes $i$ and $j$.

3. The *ordinal MDS algorithm* is applied to create the relative local map for each node.

4. Each local map is refined by using the least-squares minimization between the calculated Euclidean distance and the measured distance (or hop) between each pair of neighboring nodes.

5. The local maps are then patched (or merged) into a global map by using a predetermined initial starting node’s local map and sequentially adding each neighbor that has the largest number of common nodes to the starting node. This map then grows until all nodes have been included.

6. The global absolute map is created by using the anchors’ positions and the global relative map.
Assume that the average number of sensor nodes in each two-hop neighborhood is $M$, the average number of neighbors is $K$, the total number of sensor nodes is $N$, and total number of anchors is $A$. In the above MDS-MAP(P, O) algorithm, steps (2) and (4) have a complexity of $O(M^3)$. Step (3) has a complexity of $O(M^4)$. Steps (5) and (6) have a complexity of $O(K^3N)$ and $O(A^3 + N)$, respectively.

The ordinal MDS algorithm (step (3) above) is now described in detail. The major steps of the ordinal MDS algorithm are as follows [35]:

1. Assign arbitrary initial location estimation $(x^0_i, y^0_i)$ for $i \in M$, where $M$ includes all the nodes within the two-hop neighborhood. Specify $\epsilon > 0$ and set $n = 0$.

2. For each $i, j \in M$, compute the Euclidean distance by

$$d_{ij}^n = \sqrt{(x^n_i - x^n_j)^2 + (y^n_i - y^n_j)^2} \quad (3.1)$$

3. By using the matrices $[p_{ij}]$ and $[d_{ij}^n]$, apply monotone regression by using the pool-adjacent violators (PAV) algorithm [35] to determine $[\hat{d}_{ij}^n]$. For example, once the $p_{ij}$’s are ordered from the least to the highest, if $(p_{ij} < p_{kl})$ and $(d_{ij}^n > d_{kl}^n)$, then

$$\hat{d}_{ij}^n = \hat{d}_{kl}^n = (d_{ij}^n + d_{kl}^n) / 2.$$ 

Otherwise, $\hat{d}_{ij}^n = d_{ij}^n$ and $\hat{d}_{kl}^n = d_{kl}^n$. If there are consecutive violators, they are all
grouped and averaged together, as per the PAV algorithm, to maintain monotonicity throughout the entire set.

4. Increment $n$ by 1. For $i \in M$, compute the new relative coordinate $(x^n_i, y^n_i)$ for node $i$ by

$$x^n_i = x^{n-1}_i + \frac{\alpha}{|M| - 1} \sum_{j \in M, j \neq i} \left( 1 - \frac{\hat{d}^{n-1}_{ij}}{d_{ij}} \right) \left( x^{n-1}_j - x^{n-1}_i \right)$$

$$y^n_i = y^{n-1}_i + \frac{\alpha}{|M| - 1} \sum_{j \in M, j \neq i} \left( 1 - \frac{\hat{d}^{n-1}_{ij}}{d_{ij}} \right) \left( y^{n-1}_j - y^{n-1}_i \right)$$

where $|M|$ denotes the number of sensor nodes within the two-hop neighborhood.

5. For each $i, j \in M$, update the Euclidean distance $d^n_{ij}$ by using equation (1).

6. Use Kruskal’s Stress1 test to determine the goodness fit $36, 37$:

$$Stress1 = \sqrt{\frac{\sum_{i<j} \left( d^n_{ij} - \hat{d}^{n-1}_{ij} \right)^2}{\sum_{i<j} (d^n_{ij})^2}}$$

(3.2)

7. If $Stress1 < \epsilon$, stop. Otherwise, go to Step (3).

In the above algorithm, the first two steps calculate the Euclidean distance from an arbitrary initial configuration. Step (3) determines the disparities $\hat{d}^n_{ij}$ by constructing a monotone regression relationship $38$ between $p_{ij}$’s and $d^n_{ij}$’s. Step (4) updates the
relative positions. The parameter $\alpha$ is the step width. We use $\alpha = 0.2$ as suggested by Kruskal [39]. Step (5) updates the Euclidean distance. The Stress1 measure in step (6) determines whether or not the updated values $d^n_{ij}$ fit the given dissimilarities $\tilde{d}^{n-1}_{ij}$. Note that other goodness fit tests (e.g., Kruskal’s Stress2, normalized raw stress, S-Stress) can also be used; however we choose the Stress1 measure since it is the most common measure used for ordinal MDS. Step (7) determines if the derived configuration’s goodness fits are close enough such that the procedure can be terminated.

The MDS-MAP(P, O) algorithm assumes that there is a monotonic relationship between the shortest path distances and the actual Euclidean distances. This assumption may not be valid if the network being considered is sparse and large. However, most of the applications in wireless sensor networks require the networks to be dense (i.e., with a high connectivity or average node degree) in order to provide redundancy and robustness in case of a node’s failure. In addition, in our distributed approach, only the nodes within the 2-hop neighborhood are being considered. In this case, the assumption of the monotonic relationship between the shortest path distances and the actual Euclidean distances is valid.

By the iterative nature of the ordinal MDS algorithm in minimizing stress in equation (3.2), the final solution may not guarantee to be the global minimum [40]. In fact, the ordinal MDS algorithm can have several local minima. However, the use of the anchors in our application of the ordinal MDS algorithm increases the likelihood of reaching the global minimum. This is due to the imposed transformation required to obtain the
absolute coordinates for all of the nodes. Another way to further increase the chance of reaching the global minimum is by using the multiple starting configurations approach and retaining the configuration which results in the lowest stress value. However, this approach is inefficient due to the additional computation effort required.

### 3.3 Performance Evaluation and Comparison

To implement MDS-MAP(P, O) algorithm, the source codes for MDS-MAP(P, C) that were written in Matlab by [2] were modified to implement ordinal MDS. Two different topologies are considered as the sensor network’s coverage area. The first one is a uniformly distributed square region. The second is an irregular C-shaped topology. In both topologies, the average connectivity levels (i.e., average number of neighboring sensors) are varied along with the number of anchors in the area. The average connectivity level is varied between 9 and 21 by modifying the radio range $R$, within the fixed coverage area. In each set of simulation run, 50 trials were performed and 95% confidence intervals were plotted.

It is envisioned that there is a large number of sensors within a square coverage area to increase the robustness of the sensor network. In the simulations, either 160 or 200 sensor nodes were used. The number of anchors is between four and ten. Although in theory only three anchors are necessary to obtain the absolute position estimates for MDS-MAP(P, O), the location of the anchors in this minimal case has a significant impact on the estimation error. Thus, it is recommended that a minimum of four anchors be...
used for deployment.

In the hop-based scenarios, hop count is used as the distance metric between a pair of nodes. For each node to have a unique position in MDS-MAP(P, O), the hop count values are blurred with noise so that nodes with identical hop count values to neighbors are not co-located.

In the range-based scenarios, the range is modelled as the actual distance combined with Gaussian noise. Thus, the range is a random value drawn from a normal distribution with actual distance as mean and variance of 5%. The amount of Gaussian error is also varied and its effect on the average positioning error is studied. In the ordinal MDS algorithm, $\epsilon$ is set to be $10^{-4}$.

### 3.3.1 Random Uniform Network Topology

For evaluation of the random uniform deployment, a $10r$ by $10r$ square topology was used, where $r$ represents the reference unit length. Anchor nodes are placed randomly within the coverage area, and have the same communication range (i.e., radio range denoted by $R$) as other nodes.

**Hop-Based Performance**

In Figure 3.1, the topologies estimated by hop-based MDS-MAP(P, C) and MDS-MAP(P, O) are shown. The hollow circles represent the actual positions. The lines indicate the amount of error of the estimated positions. The filled circles represent the location of
the 4 anchors. The position estimation errors by MDS-MAP(P, C) and MDS-MAP(P, O) are 47% and 42% of the radio range, respectively.

Figure 3.1 shows that the estimation error encountered by each sensor node is different. Most of them have small estimation error while a few of them may have higher estimation error. The ordinal MDS algorithm reduces the amount of estimation error for the nodes which would have a higher estimation error if classical MDS were used. This can be attributed to the fact that the ordinal algorithm iteratively improves the initial random topology estimate thereby improving the goodness fit of the ordered distances within the 2-hop neighborhood of each node. In other words, the larger error encountered by classical MDS is due to the inaccurate modeling of the shortest path distances being equal to the actual distances between nodes, which may not always be the case. The
Figure 3.2: Performance of hop-based MDS-MAP(P, C) and MDS-MAP(P, O) in a 10r × 10r square network region employing uniform random placement of 200 nodes, for varying levels of connectivity and anchors deployed.

monotonic constraint in ordinal MDS provides accurate iterative minimization of stress and hence lower position errors for most nodes, especially nodes with large estimation errors.

Figure 3.2 shows the position estimation errors as a function of the average connectivity level by hop-based MDS-MAP(P, C) and MDS-MAP(P, O), respectively, with different numbers of anchors deployed. Results show that MDS-MAP(P, O) outperforms MDS-MAP(P, C) by a 5% lower position estimation error. The performance improvement confirms the conjecture that in sensors’ localization problem, the use of the monotonic constraints in ordinal MDS is more appropriate than the use of linear constraints in
The estimated node positions with error less than 40% of the radio range have been proven to suit the applications of sensor networks [10]. When the number of anchors is greater than 3 and the connectivity level is greater than 12, the position estimation error is always less than 40% of the radio range (our target value).

As the average connectivity level increases, the confidence intervals reduce in size. This shows that dense networks can provide more consistent average error values. This is due to the fact that dense networks have smaller two-hop regions, which in turn lead to more accurate shortest path distances. These distances therefore improve the classical MDS results as well as the ordinal MDS results, since more accurate distances translate into more accurate proximities in the ordinal case.

**Range-Based Performance**

Figure 3.3 shows the topologies estimated by range-based MDS-MAP(P, C) and MDS-MAP(P, O), respectively. The position estimation errors by MDS-MAP(P, C) and MDS-MAP(P, O) are 16% and 13% of the radio range, respectively.

Figure 3.4 shows the position estimation errors as a function of the average connectivity level by range-based MDS-MAP(P, C) and MDS-MAP(P, O), respectively. As the number of anchors deployed increases the additional gain in accuracy is minimal above 6 anchors. Results show that MDS-MAP(P, O) outperforms MDS-MAP(P, C) by a 2% lower position estimation error. We notice that the improvement in the range-based
Figure 3.3: Topology results of range-based (a) MDS-MAP(P, C) and (b) MDS-MAP(P, O) in a $10r \times 10r$ square network region employing uniform random placement of 200 nodes with connectivity level of 12 and four anchors. Anchors are denoted by shaded circles. Estimation error is represented by lines.

The range-based case is only slightly better compared with the hop-based case. However, it is clear that the range-based case outperforms the hop-based scheme by a significant margin which is desirable in order for the tradeoff of additional ranging hardware to be accepted.

Results show that in the range-based case, the position estimation error is more sensitive to the average connectivity level than to the number of anchors. Again, if the target position estimation error is less than 40%, our proposed MDS-MAP(P, O) can achieve this value when the average node connectivity level is above 9.

By comparing Figures 3.2 and 3.4, it shows that the range-based scheme outperforms the hop-based scheme in MDS-MAP(P, O). This is due to the fact that more accurate distance measurement information is available in the range-based scheme. By just using the connectivity information (i.e., hop-count), the hop-based scheme will have some nodes
Figure 3.4: Performance of range-based MDS-MAP(P, C) and MDS-MAP(P, O) in a \(10r \times 10r\) square network region employing uniform random placement of 200 nodes, for varying levels of connectivity and anchors deployed.

having the same distance measurement when in fact their actual Euclidean distances may differ. This directly affects the ordinal scheme when it comes to applying the monotonic constraint and iterative stress reduction.

Sensitivity Analysis of Range Error on Positioning Error

In this section, the effect on the position estimation error when the distance between two neighboring nodes is not determined accurately is studied in the range-based scheme. As mentioned earlier, a Gaussian noise model is used to model the estimated range distances between two nodes. The actual distance is taken as the mean value. The value of the
Figure 3.5: Effects of node ranging error on performance of range-based MDS-MAP(P, O) in a $10r \times 10r$ square network region employing uniform random placement of 200 nodes, and four deployed anchors.

Variance is changed from 0% to 50%. Figure 3.5 shows that the position estimation error increases as the range error (i.e., variance) increases. Results also show that the range-based scheme is more sensitive to the range error than to the average connectivity level.

Figure 3.6 shows that both MDS-MAP(P, O) and MDS-MAP(P, C) exhibit similar behavior to the increase of range error. This is due to the fact that both algorithms need to use the same shortest-path distance matrix constructed from the measured node ranges.
Figure 3.6: Performance comparison between range-based MDS-MAP(P, C) and MDS-MAP(P, O) with respect to varying node range error.

Sensitivity Analysis of Anchors’ Location on Positioning Error

In this section, we study the effect on the position estimation error when anchors are being placed at different locations. Figure 3.7 shows the results when the anchors are being placed (a) linearly, (b) in a rectangular manner, (c) close to each other, and (d) randomly. As we expect, the position estimation error is the highest when anchors are close to each other (see Figure 3.7(c)). On the other hand, the position estimation errors for the other three cases are similar (see Figure 3.7(a), (b), (d)), with the rectangular case performing the best with an error of 9%. Since ordinal MDS algorithm does not require the use of triangulation, the position estimation error is still considered to be
Figure 3.7: Topology results of range-based MDS-MAP(P, O) when anchors are being placed (a) linearly, (b) in a rectangular manner, (c) close to each other, and (d) randomly. The topology consists of a 10r × 10r square network region employing uniform random placement of 200 nodes with connectivity level of 12 and four anchors. Anchors are denoted by shaded circles. Estimation error is represented by lines.

small even when the anchors are being placed linearly (see Figure 3.7 (a)). As long as the anchors are not close to each other, the orientation between the anchors has a small impact on the error as seen from the performance in both linear and rectangular cases.
Chapter 3. Ordinal Multidimensional Scaling

Figure 3.8: Performance of hop-based MDS-MAP(P, O) and MDS-MAP(P, O, R) in a 10$^r \times 10^r$ square network region employing uniform random placement of 200 nodes, for varying levels of connectivity and anchors deployed. (Figure 3.7 (a) - (b)).

Further Improvement via (Optional) Global Relative Map Refinement

The accuracy of the MDS-MAP(P, O) localization algorithm can further be improved by using an optional global relative map refinement [2]. This optional step is invoked after the patching of the local maps. The least-squares minimization is used for the measured and calculated distances between neighboring nodes. This optional refinement step has a complexity of $O(N^3)$ where $N$ is the total number of sensor nodes. We use the notation MDS-MAP(P, O, R) to denote the original MDS-MAP(P, O) algorithm with
Figure 3.9: Performance between range-based MDS-MAP(P, O) and MDS-MAP(P, O, R) in a $10r \times 10r$ square network region employing uniform random placement of 200 nodes, for varying levels of connectivity and anchors deployed.

Figures 3.8 and 3.9 show the performance comparisons between MDS-MAP(P, O) and MDS-MAP(P, O, R) in hop-based and range-based scenarios, respectively. The number of anchors deployed is varied from 4 to 10. In the hop-based case, there is significant reduction on the position estimation error when the average node connectivity level is above 9. The difference between the results is greater than 30% for high average connectivity levels. In the range-based case, there is only a slight improvement when the average connectivity level is greater than 12. Note that the global relative map refinement comes at a cost. A sensor node must process the global map and then propagate the
results to all the sensors in the network (e.g., via flooding). This may cause a higher signaling overhead.

3.3.2 Random Irregular Network Topology

Whereas most papers presented have only considered uniform sensor network deployments, the method in which these networks are meant to be deployed may not guarantee uniform coverage. Wireless sensor networks may exhibit regions of sparseness once deployed. Therefore, localization algorithms must be able to perform well under different conditions. In this section, we evaluate the performance of MDS-MAP(P, O) by using the same topology in \[2\], (i.e., a C-shaped topology). In the simulations, it is noticed that the position estimation errors are changed when the anchors are placed at different positions. For good performance, it is recommended to have at least one anchor on each wing of the C-shaped topology.

Hop-Based Performance

Figure 3.10 shows the topologies estimated by hop-based MDS-MAP(P, C) and MDS-MAP(P, O), respectively. The position estimation errors by MDS-MAP(P, C) and MDS-MAP(P, O) are 74% and 65% of the radio range, respectively. The position estimation error of each individual sensor node varies. There is no correlation for sensors that are closer to the anchors to have better position estimation.

Figure 3.11 shows the position estimation errors as a function of the average connec-
Figure 3.10: Topology results of hop-based (a) MDS-MAP(P, C) and (b) MDS-MAP(P, O) in a $10r \times 10r$ irregular (C-shaped) network region employing uniform random placement of 160 nodes with connectivity level of 12 and four anchors. Anchors are denoted by shaded circles. Estimation error is represented by lines.

Results show that MDS-MAP(P, O) outperforms MDS-MAP(P, C) by a 9% lower position estimation error when the connectivity is 12. This difference is greater than the square topology case; however, the confidence intervals among the two algorithms show considerable overlap. This is to be expected since the estimated shortest path distances are more prone to errors arising from the geometry of nodes that are within the inside corners of the network. When the average connectivity levels are 12 or higher, the position estimation error is less than 65%.
Figure 3.11: Performance between hop-based MDS-MAP(P, C) and MDS-MAP(P, O) in a 10r × 10r irregular (C-shaped) network region employing uniform random placement of 160 nodes, for varying levels of connectivity and anchors deployed.

Range-Based Performance

Figure 3.12 shows the topologies estimated by range-based MDS-MAP(P, C) and MDS-MAP(P, O), respectively. The position estimation errors by MDS-MAP(P, C) and MDS-MAP(P, O) are 52% and 46% of the radio range, respectively.

Figure 3.13 shows the position estimation errors as a function of the average connectivity level by range-based MDS-MAP(P, C) and MDS-MAP(P, O), for a C-shaped network topology. When the average connectivity level is 12, results show that MDS-MAP(P, O) outperforms MDS-MAP(P, C) by a 6% lower position estimation error. This difference
Figure 3.12: Topology results of range-based (a) MDS-MAP(P, C) and (b) MDS-MAP(P, O) in a $10r \times 10r$ irregular (C-shaped) network region employing uniform random placement of 160 nodes with connectivity level of 12 and four anchors. Anchors are denoted by shaded circles. Estimation error is represented by lines.

is greater than the square topology case; however, once again the confidence intervals among the two algorithms show significant overlap. When the average connectivity level is 12 or higher, the estimation error is less than 46%.

Further Improvement via (Optional) Global Relative Map Refinement

Figures 3.14 and 3.15 show the performance comparisons between MDS-MAP(P, O) and MDS-MAP(P, O, R) in hop-based and range-based scenarios, respectively. In the hop-based case, there is significant reduction on the position estimation error when the average node connectivity level is 12 or greater. The additional refinement scheme outperforms ordinal MDS in the hop-based results on average 25%, for connectivity levels 12 and
Figure 3.13: Performance of range-based MDS-MAP(P, C) and MDS-MAP(P, O) in a $10r \times 10r$ irregular (C-shaped) network region employing uniform random placement of 160 nodes, for varying levels of connectivity and anchors deployed.

greater, which is larger than in the square topology case. In the range-based case, the refinement scheme achieves on average 26% better accuracy than the ordinal MDS scheme, which again is a higher gain in accuracy compared to the square topology results. Thus the benefit of the MDS-MAP(P, O, R) is greater in the irregular (C-shaped) topology than in the uniform (square) topology. This is expected, since the iterative refinement of the global map in the irregular case takes into account the topology from the result of the initial MDS-MAP(P, O) result. Thus, the refinement scheme is able to improve the result by iteratively reducing position estimation error from a global viewpoint of the network.
Figure 3.14: Performance between hop-based MDS-MAP(P, O) and MDS-MAP(P, O, R) in a 10r x 10r irregular (C-shaped) network region employing uniform random placement of 160 nodes, for varying levels of connectivity and anchors deployed.

3.4 Summary

In this chapter we proposed and analyzed, the MDS-MAP(P, O) localization algorithm for wireless sensor networks. The MDS-MAP(P, O) algorithm is an extension of the MDS-MAP(P, C) algorithm originally proposed in [2]. The algorithm is extended by using the ordinal MDS algorithm instead of the classical MDS algorithm. The proposed MDS-MAP(P, O) algorithm is essential for future sensor applications which require a high accuracy of nodes’ position by using a small number of anchor nodes. The algorithm can be applied not only to the case where nodes are equipped with distance-estimation hard-
Figure 3.15: Performance between range-based MDS-MAP(P, O) and MDS-MAP(P, O, R) in a $10r \times 10r$ irregular (C-shaped) network region employing uniform random placement of 160 nodes, for varying levels of connectivity and anchors deployed.

Hardware (range-based), but also to the case where only connectivity information (hop-based) is available. Simulation studies were conducted under both regular (square) and irregular (C-shaped) topologies, resulting with MDS-MAP(P, O) providing a lower position estimation error than MDS-MAP(P, C) in both hop-based and range-based scenarios.
Chapter 4

Concentric Anchor-Beacons (CAB) Localization Algorithm

In this chapter, we propose two CAB localization algorithms for wireless sensor networks; CAB with Equal Area (CAB-EA) and CAB with Equal Width (CAB-EW). We first state the motivations and assumptions. It is followed by the description of the CAB algorithms. We then discuss the advantages and limitations of the proposed scheme. It is followed by the performance evaluation and comparison to other schemes.

4.1 Motivations and Assumptions

Although distributed range-based algorithms have a higher accuracy than the distributed range-free approaches in general, the range-free approaches are more cost effective. In this chapter, we focus on the design of a distributed range-free localization algorithm that has a high accuracy and does not require communication between neighboring sensor nodes. In our proposed scheme, each sensor node estimates its position solely based on the information gathered directly from the anchor nodes. Since it does not depend on neigh-
boring sensor node communication, it is independent of network connectivity. Sensor nodes do not require any special range-determining hardware for localization. On the other hand, anchors are equipped with GPS modules. Thus, anchor nodes are more costly, consume more energy, and are larger in size than normal sensor nodes. In addition, as in the case of some other schemes (e.g., \[10\]), anchors are assumed to have larger communication range than normal sensor nodes. The anchor-to-node range (ANR) ratio is equal to the maximum communication range of an anchor divided by the communication range of a sensor node.

The main difference between CAB and other range-free localization approaches is that in CAB, anchors transmit beacon signals at different power levels periodically. This requirement is feasible in current wireless sensor networks. For example, the Mica2 mote sensor nodes have a range of 18 meters for transmission power of -10 dbm, and 50 meters for 0 dbm \[41,42\].

Ideally, the different power levels divide the possible transmission ranges of an anchor into a circle and rings. As shown in Figure 4.1, the lowest power level creates a circular coverage area, and the following higher levels are distinguished by rings emanating from this lowest level.

The maximum range an anchor can transmit a beacon is defined to be the largest distance from an anchor that a node can lie and be able to hear and decode the beacon. Theoretically, the maximum range \(r\) is determined by the maximum power \(P_{\text{max}}\) (textit{Watts}) that an anchor node can transmit, and the minimum received power \(P_{\text{threshold}}\)
Figure 4.1: Anchor beacon transmission ranges with (a) Equal area beacon signals with 3 power levels; and (b) Equal width beacon signals with 3 power levels. $A_i$ ($i = 1, 2, 3$) denotes the area of the $i$th ring/circle; $w_i$ ($i = 1, 2, 3$) denotes the width of the $i$th ring/circle.

that a sensor node requires to accurately decode the received beacon signal [22]. With $n$ denoting the path loss exponent, this relationship is shown in the equation below:

$$r \propto \sqrt[n]{\frac{P_{\text{max}}}{P_{\text{threshold}}}} \quad (4.1)$$

In the free-space propagation environment, the required power to transmit at a distance $d$ is directly proportional to the square of the distance. In general, the transmit power $P_{\text{transmit}}$ is directly proportional to the $n^{th}$ power of the distance $d$, where $n$ denotes the path loss exponent. The relationship is given in the following equation:

$$P_{\text{transmit}} \propto d^n \quad (4.2)$$

Thus, the ability of a node to hear beacons with lower transmitted powers, implies
that the node lies closer to the anchor than the maximum range. Consider an example in Figure 4.1(a). If the sensor node can hear the beacons with power levels $P_1$, $P_2$ and $P_3$, then the distance between the anchor and the node is less than $r_1$. That is, the sensor node lies within the inner-most circle. On the other hand, if the node can only hear the beacon with power level $P_3$, then it lies within the outermost ring.

The CAB algorithm is developed based on the above assumptions. Note that if a beacon is heard by a node, it is assumed that the received signal power at the node is greater than or equal to the minimum threshold power.

### 4.2 CAB Algorithm

In this work, two variations of the CAB localization algorithm are considered, namely: CAB-EA and CAB-EW. For CAB-EA (CAB with Equal Area), it is assumed that the area of the inner-most circle and the rings are all the same. That is, in Figure 4.1(a), the circle with radius $r_1$ has the same area as each of the rings outside that circle. The relationship between the beacon transmission ranges $r_i$ and the maximum transmission range $r$ is given by the following equation:

$$ r_i = \sqrt{\frac{i}{m}} r \quad i = 1, 2, \ldots, m $$

(4.3)

where $i$ denotes the beacon number starting from the lowest power level (or transmission range), $m$ denotes the total number of different beacon power levels, $r_i$ denotes the transmission range for beacon $i$, and $r$ denotes the maximum range that an anchor can
transmit at the corresponding maximum power level $P_{\text{max}}$.

Therefore, in relation to power levels, the corresponding equation is as follows:

$$P_i = \left( \sqrt{\frac{i}{m}} \right)^n P_{\text{max}} \quad i = 1, 2, \cdots, m \quad (4.4)$$

where again $i$ denotes the beacon number starting from the lowest power level, $m$ denotes the total number of power levels, $n$ denotes the path loss exponent, and $P_i$ represents the transmit power for beacon $i$ in terms of the maximum transmit power $P_{\text{max}}$.

For CAB-EW (CAB with Equal Width), it is assumed that the width of the inner-most circle and the rings are all the same. The relationship between the beacon transmission ranges $r_i$ and the maximum transmission range $r$ is given by the following equation:

$$r_i = \frac{i}{m} r \quad i = 1, 2, \cdots, m \quad (4.5)$$

The corresponding relationship between power transmission levels $P_i$ and the maximum transmit power $P_{\text{max}}$ is given below:

$$P_i = \left( \frac{i}{m} \right)^n P_{\text{max}} \quad i = 1, 2, \cdots, m \quad (4.6)$$

Before deployment, measurement is necessary to relate the transmission power $P_i$ and coverage range $r_i$. Thus, the maximum range is empirically determined to be the distance from an anchor node, transmitting at maximum power, at which a sensor node has a received signal power equal to the minimum threshold power. This is important in order to ensure the accuracy of the range-free approach. In Section [4.4] the effects
when the information between \( P_i \) and \( r_i \) is not accurate due to interference from the neighboring environments will be studied.

The CAB localization algorithm is now described in detail. The algorithm described below is applicable to both CAB-EA and CAB-EW. Anchors transmit beacon signals with different power levels periodically. Each beacon signal packet includes the anchor’s ID, the anchor’s location, the transmit power level \( P_i \) information, and the estimated maximum distance that the beacon signal can be heard. Each node listens for beacons and collects the anchor’s information. For each beacon heard, the sensor node determines within which region of the anchor’s concentric transmission circles it lies. Figure 4.2 shows an example with a sensor node surrounded by three anchors. Each anchor transmits beacons periodically at two different power levels. The corresponding information table collected by the sensor is shown in Table 4.1.

Depending on the percentage of anchors deployed, each sensor node can hear multiple
beacons from different anchors. For computational simplicity, information from at most three neighboring anchors is used to estimate a sensor’s location. In order to increase the accuracy of the position estimate, it is necessary to minimize the region of intersection by choosing the three anchors that are farthest. This is accomplished by calculating all the possible triangular areas that are made up of the anchors heard, and by choosing the three anchors that form the largest triangle.

Each sensor node can receive multiple beacons with different power level information from the same anchor. Based on this information, the sensor node can determine which particular ring or inner circle it lies within from that anchor. This is called the constraint region. Mathematically, this region is bound by either two equations of circles (for the ring case) or just one equation of a circle (for the inner-most region of the anchor). The last column in Table 4.1 shows the constraint regions that the sensor node lies within based on the scenario in Figure 4.2.

Given the three chosen anchors, two of them are selected at a time to calculate the intersection points. The valid intersection points satisfy all three anchors’ constraint
regions. The *invalid intersection points* are those that do not lie within the other anchor’s constraint region. Consider the example in Figure 4.2. Let \((x_A, y_A), (x_B, y_B), (x_C, y_C)\) denote the positions of anchors A, B, and C, respectively. Let \(I\) denote the set of intersection points. For each point \((x, y) \in I\), it is a *valid intersection point* if the following constraints are satisfied:

\[
\begin{align*}
    r_1 &\leq \sqrt{(x_A - x)^2 + (y_A - y)^2} \leq r, \\
    r_1 &\leq \sqrt{(x_B - x)^2 + (y_B - y)^2} \leq r, \\
    r &\leq \sqrt{(x_C - x)^2 + (y_C - y)^2} \leq r_1.
\end{align*}
\]

The final position estimate is taken as the average of all the valid intersection points. Figure 4.2 shows the estimated position determined by four valid intersection points.

### 4.3 Discussion

There are three distinct advantages of the CAB localization algorithm. First of all, CAB is distributed and is simple to implement. For the anchors, their only task is to transmit beacon signals with different power levels periodically. For each sensor node, the determination of the intersection points from three chosen anchors as well as the position estimate by averaging are not computationally intensive. Secondly, no information exchange between neighboring sensors is necessary. This reduces the energy requirement for localization. In addition, CAB has a higher accuracy than some other range-free
localization algorithms. Simulation comparisons will be presented in the next section.

For the qualitative comparisons with some other localization algorithms, APIT requires communication between neighboring nodes for the exchange of tabular information of nearby anchors. CAB does not require that procedure and achieves better results under smaller anchor-to-node range (ANR) ratio. In comparison to Centroid which requires a grid-based deployment, CAB is able to perform sufficiently well in ad hoc deployments. Whereas ring sizes are determined from RSS values in ROCRSSI, the rings in CAB are pre-determined according to the number of power levels desired to be used by the anchor. No communication is required between anchors in CAB.

The CAB scheme does have its limitations. Being solely dependent on anchor nodes for position estimation, the accuracy depends on the percentage of anchor nodes deployed. This percentage can be decreased by increasing the maximum radio range of the anchors. However, this results in less accuracy since the intersection areas become larger. Also, since the scheme’s computation relies on a circular radio model, it can be affected by irregular radio propagation, of which other range-free schemes are relatively immune to. In Section 4.4, we will also present the results of the scheme under a different degree of radio pattern irregularity.

### 4.4 Performance Evaluation and Comparison

In this section, the performance evaluation of CAB-EA and CAB-EW as well as the comparisons with APIT and Centroid algorithms are presented. All algorithms
are simulated in Matlab. The wireless sensor network consists of 280 nodes and a varying number of anchors are randomly placed. The network topology is a square of side $10R$ by $10R$, where $R$ is the sensor node communication range. The average connectivity among nodes is set to 8.

The technique in [10] to model the *irregular radio pattern* is used in this simulation model. In this model, all nodes within half of the maximum transmit radio range of anchors are guaranteed to hear from the anchor, whereas nodes between the maximum radio range and half of that range may or may not hear from the anchor depending on the radio pattern in that direction. The degree of irregularity (DOI) parameter is defined as the maximum radio range variation per unit degree change in direction. Examples of different DOI values of this irregular radio pattern model are shown in Figure 4.3.

For the simulation of CAB, it is assumed that the path loss exponent ($n$) is equal to 2. The anchor-to-node range (ANR) ratio is set to 3. The DOI value is set to 0.05. The estimation errors are normalized with respect to the sensor node range ($R$).
Chapter 4. Concentric Anchor-Beacons (CAB) Localization Algorithm

Figure 4.4: Comparison of percentage of nodes localizable versus percentage of anchors deployed for varying levels of ANR. (DOI = 0.05)

4.4.1 Performance of CAB

In Figure 4.4, the percentage of nodes that are able to hear at least three anchors versus the percentage of anchors deployed is shown. In general, it is desirable to deploy a minimal percentage of anchors nodes to localize the system. The results show that for 12% of anchor nodes deployed, ANR values of 3 or higher enable at least 90% of all nodes to obtain position estimates. As the ANR value increases, the percentage of nodes able to hear at least three anchors also increases; however, this results in less accurate position estimation. Considering this tradeoff, it can be reasoned that an ANR of 3 is suitable for anchor percentages as low as 16%. 
Figure 4.5: Average estimation error under different number of power levels of the beacons for (a) CAB-EA and (b) CAB-EW. (ANR = 3, DOI = 0.05)
Figure 4.5 shows the accuracy gain of (a) CAB-EA and (b) CAB-EW by increasing the number of power levels of the beacons (i.e., increase of \( m \)). When beacons are being transmitted at a single power level (\( m = 1 \)), the intersection area is constructed by determining the intersections of three circles centered at their corresponding anchors. It is clear that with two different power levels (\( m = 2 \)), it reduces the intersection area to intersections of rings and circles. Figure 4.5 shows that the estimation error reduces by at least 0.44\( R \) when \( m \) increases from 1 to 2 for CAB-EA, a significant improvement.

Notice that when the number of different power levels increases to 3 (or higher), the performance improvement is marginal. This is due to the fact that when \( m \) is further increased, the anchor coverage area is subdivided into a circle and more concentric rings. The irregular radio pattern model introduces more errors to the rings with smaller ring width. However for CAB-EW, the performance improvement is significantly better for 3 power levels than 2. For anchor percentages greater than 12% CAB-EW with 3 levels outperforms its 2 power level counterpart by 0.25\( R \). Thus for CAB-EW, it is beneficial to use 3 power levels, while for CAB-EA, 2 power levels are sufficient.

Figure 4.6 shows the performance comparison for CAB-EW and CAB-EA for 2 power levels. Clearly, the CAB-EA scheme outperforms CAB-EW in this case, since the estimation error is reduced by 0.22\( R \) on average when using CAB-EA.

Comparison of CAB-EW and CAB-EA using three beacons and varying DOI values is shown in Figure 4.7. In Figure 4.7(a), the perfect radio propagation model results in the CAB-EA marginally outperforming CAB-EW. When the DOI value is increased to
Figure 4.6: Comparison of estimation error between CAB-EW and CAB-EA. \((m = 2, \text{ANR} = 3, \text{DOI} = 0.05)\)

0.05 as shown in (b), the CAB-EW achieves lower estimation error for anchors deployed of 7% or greater. As the DOI value is further increased to 0.10 in (c), CAB-EW is clearly more accurate, achieving \(0.27R\) lower error for 16% of anchors deployed. These results show that, CAB-EW is more resilient to DOI irregularity than CAB-EA. An explanation for this may be due to the fact that CAB-EA has rings closer together near the maximal transmission range, and therefore for larger DOI values, the irregularity results in error prone determinations of which ring the sensor node is within. CAB-EW, however, has uniformly separated rings, and is therefore less prone to errors near the edge of the maximal transmission range. Therefore, we can conclude that the use of CAB-EA is suitable for \(m = 2\) and for CAB-EW it is best to use \(m = 3\).
Chapter 4. Concentric Anchor-Beacons (CAB) Localization Algorithm

Figure 4.7: Comparison of estimation error between CAB-EA and CAB-EW for different DOI values: (a) DOI = 0, (b) DOI = 0.05, and (c) DOI = 0.10. (m = 3, ANR = 3)
It is possible that a sensor node may receive beacon signals from more than three anchors. In CAB, only three neighboring anchors are used for localization. Figure 4.8 shows the comparison between two different ways of choosing those neighboring anchors. For the case of random, the three anchors heard with the lowest IDs are chosen. For the case of optimal, the three anchors which form the largest triangle are chosen. Results in Figure 4.8 show that the optimal approach provides a much lower estimation error than the random choice. As an example, when the percentage of anchors deployed is 11%, the optimal choice provides an estimation error $0.95R$ lower than the random choice on average. In addition, for the optimal choice, the estimation error decreases when the percentage of anchors deployed increases. This is expected since there are more anchors
from which to choose.

The choice of using only three anchors for position estimation is to reduce the computational complexity, since considering more anchors results in many more intersection points to be computed. We are aware that choosing the anchors that result in the largest triangular region does not always guarantee the smallest coverage intersection, since the intersection also depends on the size of the circle or ring constraining the position of the node. However, further results show that choosing the 3 anchors that form the largest triangle is sufficient in achieving good accuracy.

### 4.4.2 Comparisons between CAB, APIT, and Centroid

Figure 4.9 shows the position estimation errors as a function of the percentage of anchors deployed. CAB has a better performance than both APIT and Centroid. As an illustration, when the percentage of anchors deployed is 16%, CAB-EW with three power levels achieves 0.78\(R\) accuracy and CAB-EA with two power levels has an average error of 0.81\(R\). The other schemes, APIT and Centroid achieve 0.94\(R\), and 1.31\(R\) accuracy, respectively. Note that the performance of CAB can further be improved by utilizing information from more than three anchors at the expense of a higher computation complexity.

Figure 4.10 shows the results of the estimation error as a function of ANR ratio. The percentage of anchors deployed is 9%. As the ANR value increases, this results in a loss of accuracy in all schemes. In the Centroid scheme, nodes can now hear anchors
Figure 4.9: Comparison between Centroid, APIT, CAB-EA ($m = 2$), and CAB-EW ($m = 3$) by increasing the percentage of anchors deployed. (ANR = 3, DOI = 0.05)

that are farther, and the result is a more coarse grained estimation of position. In the APIT scheme, the ANR actually improves the accuracy until ANR equals 5. The error then increases with higher ANR values. This unique behavior can be attributed to the InToOut error identified in [10] which is more significant at low ANR values and diminishes with increasing ANR. The CAB algorithm only relies on anchor information and thus increases in error as ANR increases, as can be seen by the CAB-EA and CAB-EW plots. Once again, CAB-EW uses three different power levels whereas CAB-EA uses two levels. The higher ANR values result in larger ring areas that in turn create larger intersections within which the node estimate is taken. Figure 4.10 also shows that APIT
outperforms both CAB schemes for ANR greater than 4. Note that in APIT, each sensor node consumes additional energy for the exchange of information between neighboring nodes. In CAB, information exchange between neighboring nodes is not necessary. In other words, the accuracy of CAB does not depend on the average node degree or the connectivity information.

Figure 4.11 shows the effects of irregular radio propagation on the accuracy of the range-free schemes. The percentage of anchors deployed is 9%. Due to the use of the fixed empirical range values for different transmit power levels of the beacons, the CAB schemes are more sensitive to the irregular radio pattern than Centroid and APIT. When DOI values are less than 0.09, CAB-EA outperforms the other two schemes, whereas CAB-EW
Figure 4.11: Comparison between Centroid, APIT, CAB-EA ($m = 2$), and CAB-EW ($m = 3$) under different DOI values. (ANR = 3)

outperforms all schemes since it incorporates 3 power levels.

Based on the above results, it is suggested that the following parameters be used for the implementation of the CAB-EA localization algorithm: $m = 2$, $r_1 = 0.707r$, $ANR \leq 3$, percentage of anchors deployed higher than 9%. For applications requiring greater accuracy, the use of CAB-EW is suitable, with the following parameters: $m = 3$, $r_1 = 0.33r$, $r_2 = 0.66r$, $ANR \leq 3$, and percentage of anchors deployed higher than 9%. 

![Graph showing comparison between different localization algorithms](image-url)
4.5 CAB Extension to other Localization Schemes

The novelty of the CAB algorithm is the different power levels at which the anchor nodes broadcast beacons. Although this property has been used to construct a range-free scheme that estimates positions of sensor nodes to be within the intersections of rings or circles, it is evident that different aspects of CAB can be applied to some other previously proposed localization schemes. The three different aspects of the CAB algorithm are: (1) different power level beacons, (2) circular and ring position constraints, and (3) position estimation based on selected anchors’ information. These techniques can be independently applied to several other schemes to enhance the performance.

In traditional range-based schemes, specialized ranging hardware is required to obtain the distance information between nodes. However, due to channel fading and interference, estimation based on RSS does not always provide a robust means of distance information. By incorporating the beaconing anchor property, the reliance on specialized hardware and/or RSS measurements can be reduced. The corresponding result is that range-based schemes can function in a relatively range-free manner depending on scheme-specific details.

The only assumption in CAB is that there is a reference maximum transmission power and the corresponding transmission range that is empirically derived prior to system deployment to take into account environment propagation characteristics.

The incorporation of circular and ring constraints can be separately included to the existing schemes, in order to provide an overlapping region based localization as opposed
to lateration or triangulation. Lastly, as evidenced in the CAB scheme, instead of using all information gathered from all anchors heard to determine position estimates, three anchors were selectively chosen from which the position estimate was to be computed. In the proposed CAB algorithm, it is advantageous to do so from a computational point of view which is necessary for practical implementations. Other schemes may also be able to benefit from the selectivity of anchor information, whether to reduce computational needs of the scheme or to avoid information that may be prone to errors.

4.5.1 Possible Scheme-Specific Modifications

In the convex positioning scheme \cite{6}, instead of using anchors and nodes with fixed radio range, the use of different power level beaconing can be advantageous. Nodes can listen to beacons from other nodes and determine their communication constraint as the distance corresponding to the lowest power level heard from another node. The accuracy of the approach depends on the number of power levels used. Once the nodes have determined their constraining circles, the linear matrix inequalities (LMI) can be obtained and the solutions can be determined by the corresponding semi-definite programs (SDPs).

For anchor propagation schemes such as APS \cite{5}, in addition to the hop-by-hop transmission of beacons, the anchors can also transmit at different power levels to directly reach further sensor nodes. Nodes can then simultaneously execute both APS and CAB procedures and ensure that the estimated position based on APS falls within the intersection of the rings and circles determined by CAB. Alternatively, when nodes calculate distances
based on hop count, they can use the information from the power levels heard to ensure that the calculated distances are accurate prior to computing a position estimate. In addition, APS can benefit from selecting only the closest anchors for position estimation, since distances calculated from anchors that are farther away can be inaccurate.

Range-free schemes can also benefit from the use of multiple beacons from anchors. The main benefit is the additional information gathered by the node, in terms of how close it is from the anchor by simply monitoring the received power levels. In the Centroid scheme, each node determines its location by averaging the positions based on all of the different anchors heard. By using multiple power levels, more information is gathered by nodes to further enhance the position estimate. Nodes can therefore assess the different power levels in order to estimate their position more accurately. Thus, a node that can hear more than one power level from an anchor will give more weight to that anchor than another where it can only hear a single power level.

The APIT scheme can also be extended since its structure is similar to CAB in the use of anchors with larger transmission ranges than nodes. Instead of using a SCAN-grid algorithm to determine the overlapping of only triangles formed by anchor positions, the overlapping of circular regions can also be included to further optimize the position estimate. Alternatively, the overlapping of rings can be used in the SCAN-grid algorithm to estimate the position of the node. In order to reduce the errors identified as InToOut and OutToIn in [10], sensor nodes can obtain estimates of distance from the three closest anchors by using the corresponding distance related to the lowest power level heard from
an anchor. Using these three estimated distances, lateration can be used to ensure that the APIT results satisfy the circular constraints.

4.6 Summary

In this chapter, the CAB localization algorithm for wireless sensor networks was proposed. CAB is a distributed range-free approach that does not require information exchange between neighboring sensors. It has a low computational overhead that is simple to implement. CAB uses anchors that broadcast beacon signals at varying power levels periodically. This allows each sensor node to identify which annular ring centered at the anchor that the node resides in. The estimated position of the node is taken as the average of all the valid intersection points. Simulation results show that CAB provides a lower position estimation error than APIT and Centroid under a wide range of conditions. It is also evident that the novel method of anchor-beaconing can be applied to previously proposed schemes that may be worthwhile in reducing position estimation error and even outperform CAB itself.
Chapter 5

Conclusions and Future Work

As evidenced from the diverse techniques being employed to solve the sensor node localization problem, this area is under continued refinement and development. Though many different proposals have been made, the assumptions in the models have limited the applicability of the individual schemes in other networks. Hence, for future work, the continued search for a technique or algorithm capable of providing accurate sensor positions, lies in the realm of robustness under different network conditions. To suit this purpose, the scheme must work in a decentralized manner, contain a low percentage of anchor nodes (less than 10%), and be able to cope with low node connectivity. In general, connectivity is considered to be low if a node has less than 9 neighbors, medium if it has up to 15 neighbors and high if the connectivity is greater than 15. Decentralized networks may contain anchors deployed in random fashion or carefully placed in the coverage area. The latter being a more suitable design goal for new schemes. In addition, the aspect of positioning delay has seldom been considered in proposed schemes. In terms of this delay, the better scheme is able to provide the positions of all nodes within a specified amount of time, having much relevancy to the way in which the algorithm is executed in the network.
5.1 Future Work

In terms of future work, several areas are of interest: multimodal and measurement-free techniques, accuracy, energy consumption, mobility, and security.

The use of multimodal techniques employing both range and angle measurements, have outperformed sole range or angle-based techniques in terms of measurement accuracy. Indeed, this will be an area to further explore, as only a few papers have explored this possibility, and have not yet come with an efficient way of combining both measurements in a decentralized and parallel manner. The main problem in this approach has been the actual feasibility of sensor nodes being equipped with multimodal hardware. The complete opposite to this approach is avoiding measurements at all, sighting inaccuracies of measurement approaches, several papers have presented novel schemes \[10,12\] involving reducing the possible position of nodes by constraining regions as has been proposed in this thesis.

In general, all schemes have quoted their accuracy with respect to sensor communication range. In terms of future work, the development of more accurate schemes is clearly possible. According to results so far, the most accurate schemes contain medium or high node connectivity levels, as evidenced by the unsurpassed results obtained by MDS techniques. Further work must be done in order to achieve accuracy at low levels of connectivity, thus having a greater impact by drastically reducing the amount of nodes that need to be deployed. In addition, the development of ranging technology to be used in the sensors can greatly affect the accuracies achieved. The use of UWB technology
has shown to provide incredibly accurate results in the presence of obstacles from indoor deployments. The extension to outdoor sensor networks has yet to be explored.

Another aspect of the localization problem is the energy consumption. Few, if any, schemes have actually explored the impact of their techniques on the sensor nodes’ battery power. This is by far the most under-developed aspect of this area, relative to the significance in an actual sensor network deployment. Since localization of the nodes is only necessary to aid the sensor network in achieving its deployment purpose, the positioning scheme must not deplete the sensor nodes and thus prevent them from achieving their main deployment purpose. This can be avoided by schemes involving the passive localization of sensor nodes, in which nodes position themselves relative to fixed system anchors, and need not communicate with other sensors in order to obtain accurate positions. Of course, this again affects the robustness of the scheme in terms of ad-hoc deployments, and thus further work may have to be done in developing hybrid models where the necessary communication between nodes is limited.

Another important step is in the development of positioning schemes for mobile sensor nodes. A few papers have extended their schemes from fixed positions to enable moderate mobility in the network, but clearly such schemes do not make use of the increased information available to them due to node movement. Whereas, some have presented the case that mobility adversely affects position estimation, others have countered that claim. As nodes become mobile, the neighborhood information is constantly changing with respect to each node. This additional information, unavailable in static networks,
Chapter 5. Conclusions and Future Work

should be used as an advantage in the node localization problem. One promising approach is in the realm of statistical based techniques, extensively used in the target tracking and signal processing communities. Such techniques use information to construct probability density functions of each node’s position in the network to obtain position information. The continuous movement of nodes will eventually result in accurate tracking of all nodes in the network. Many questions still remain to be answered in this area, including the accuracies possible, energy consumptions, computational effort required and the efficacy of decentralized approaches.

Lastly, a facet remaining to be explored is the issue of security in the network communications of the sensors. Many applications have been proposed for such networks in adversarial settings, which impose security threats on the network. Several papers have outlined some of these issues and proposed methods to be employed in existing schemes to make them more secure. However, separate development of security and positioning techniques is inefficient, and a better approach would be to incorporate security measures within the localization scheme. The more robust architectures in terms of security are more centralized, since it is assumed that fixed anchors and central computation authorities are trusted, whereas in distributed systems sensor nodes greatly affect the systems information and thus malicious nodes have greater impact on the system.
Bibliography


[22] N. Patwari, J. Ash, S. Kyperountas, A. Hero III, R. Moses, and N. Correal. Locat-
ing the nodes: Cooperative localization in wireless sensor networks. *IEEE Signal


networks. In *Proc. of Hawaii Int. Conf. on System Sciences (HICSS-34)*, Maui,

[26] A. Savvides, H. Park, and M. Srivastava. The bits and flops of the n-hop multilat-
eration primitive for node localization problems. In *Proc. of ACM WSNA*, pages
112–121, Atlanta, GA, September 2002.

[27] C. Savarese, J. Rabaey, and K. Langendoen. Robust positioning algorithms for
distributed ad-hoc wireless sensor networks. In *Proc. of USENIX Technical Annual

[28] N. Patwari and A.O. Hero III. Using proximity and quantized RSS for sensor local-
September 2003.


