Distributive Time Division Multiplexed Localization Technique for WLANs

ADNAN UMAR KHAN

Faculty of Technology

De Montfort University, United Kingdom

This thesis is submitted in partial fulfillment of the requirements of De Montfort University for the award of Doctor of Philosophy

JULY, 2012

Acknowledgments

First and foremost, I would like to express my sincere gratitude to my supervisors Professor Dr. Marwan Al-Akaidi and Mr. Jonathan Ivins for their kind supervision, suggestions and advice throughout the preparation of this thesis. Their patience over a long duration has been a constant source of inspiration throughout this work.

Also, I would like to thank my parents for their endless support, prayers and guidance since the day I was born.

I would like to thank Pakistan Telecommunication Company Limited for support and providing environment to conduct this research work.

I would also like to thank the following:
My friends in the Institute of Communication Technologies, Pakistan , in particular Dr. Muhammad Shahid Khalil for his constant motivation, and Mr Shees Ashraf for his support.

Last but not least, I would like to thank Dr. Muhammad Asim Farooqi from Department of Mechanical Engineering, Ghulam Ishaq Khan Institute, Pakistan for his valuable comments.

Abstract

This thesis presents the research work regarding the solution of a localization problem in indoor WLANs by introducing a distributive time division multiplexed localization technique based on the convex semidefinite programming.

Convex optimizations have proven to give promising results but have limitations of computational complexity for a larger problem size. In the case of localization problem the size is determined depending on the number of nodes to be localized. Thus a convex localization technique could not be applied to real time tracking of mobile nodes within the WLANs that are already providing computationally intensive real time multimedia services. Here we have developed a distributive technique to circumvent this problem such that we divide a larger network into computationally manageable smaller subnets. The division of a larger network is based on the mobility levels of the nodes. There are two types of nodes in a network; mobile, and stationery. We have placed the mobile nodes into separate subnets which are tagged as mobile whereas the stationary nodes are placed into subnets tagged as stationary. The purpose of this classification of networks into subnets is to achieve a priority-based localization with a higher priority given to mobile subnets. Then the classified subnets are localized by scheduling them in a time division multiplexed way. For this purpose a time-frame is defined consisting of finite number of fixed duration time-slots such that within the slot duration a subnet could be localized. The subnets are scheduled within the frames with a 1:*n* ratio pattern that is within *n* number of frames each mobile subnet is localized *n* times while each stationary subnet consisting of stationary nodes is localized once. By using this priority-based scheduling we have achieved a real time tracking of mobile node positions by using the computationally intensive convex optimization technique. In addition, we present that the resultant distributive technique can be applied to a network having diverse node density that is a network with its nodes varying from very few to large numbers can be localized by increasing frame duration. This results in a scalable technique. In addition to

computational complexity, another problem that arises while formulating the distance based localization as a convex optimization problem is the high-rank solution. We have also developed the solution based on virtual nodes to circumvent this problem. Virtual nodes are not real nodes but these are nodes that are only added within the network to achieve low rank realization.

Finally, we developed a distributive 3D real-time localization technique that exploited the mobile user behaviour within the multi-storey indoor environments. The estimates of heights by using this technique were found to be coarse. Therefore, it can only be used to identify floors in which a node is located.

Contents

Chapter 1

Introduction

4G wireless networks are beginning to be deployed with an intention of providing broadband services comprising of voice, video and data. The services provided by 4G networks include truly broadband, low-latency applications comprising of mixture of voice, video, data and location-based services.

There are number of applications and requirements of location based services such as health care [1], public safety [2-3], performance enhancement of the network where location of the nodes, for example, could be used for efficient routing, location sensitive browsing, and many more.

Localization is defined as a process to determine the physical coordinates of mobile terminals, simply called as nodes, within a predefined space. In the past, different terms have been used within literature for this process such as radiolocation, position location, geolocation, location sensing, realization [4] or localization [5]. We have adopted the terms of localization and realization within our dissertation. The term realization is normally used when localization is studied in the context of distance geometry.

There are number of techniques by which localization can be achieved; the most popular and well developed techniques are based on GPS. Although GPS based localization are promising, they have one constraint, that is, the mobile device should be under the open sky to receive LOS signals from at least four GPS satellites for localization. Therefore, GPS based localizations are not suitable for indoor and urban environments where signal strength weakens and LOS is not always available [6].

This research develops an alternative technique for indoor localization which is based on distance measurements taken from Time of Arrival. Although time based measurements are highly accurate but the resultant distance based optimization becomes non-convex. Therefore, non-convex optimization techniques can result in large errors due to estimation from local optima. To overcome this limitation, we developed a WLAN localization technique based on convex optimization.

1.1 Localization and Challenges

Localization process can be divided into two sub-processes. In first sub-process measurements are obtained, these could be distance measurements obtained from signal strength, time of arrival (TOA), time difference of arrival (TDOA), carrier signal phase of arrival (POA) or angle measurements based on angle of arrival (AOA). The indoor radio channel suffers from multipath propagation and shadow fading due to which AOA and signal strength based measurements provides less accurate results as compared to TOA measurements [91]. Therefore TOA based distance measurements are recommended for indoor environments. Hence in this thesis we have developed a localization technique based on distance measurements.

Mobile terminals or nodes localization is the second sub-process. Localization based on the distance measurements is such that within a network some nodes positions are known a priori, called hereafter as *anchors*, and others called hereafter as *unknown nodes* or simply *nodes*. Nodes determine their positions by measuring distances with respect to these anchors and the distance measurements among unknown nodes [8-10]. Since the measured distances are never accurate, therefore, some kind of estimation technique is needed in the second sub-process to determine the actual position i.e. coordinates of the nodes.

The analytical and theoretical study of localization problem based on the distances measurements is adopted from the theory of distance geometry, rigidity, and Euclidean Distance Geometry. Wherein the distance geometry, the problem of finding positions

from distances is known as a *graph realization* problem and is considered to be a difficult problem [10, 11]. Furthermore problem of finding a unique position from distances is known as a *unique realization* [8].

Localization based on the distance information is a non-convex optimization problem [8]. The non-convex optimization techniques may result in large estimation errors due to the possibility of finding the local minima instead of global minima. Fortunately, there exist some techniques where a non-convex problem could be formulated in a convex form with promising results.

In [8] Anthony has analyzed the use of SemiDefinite Programming (SDP): a convex optimization technique to study localization problems based on distance measurements; where, some relaxations are applied to the original non-convex problem to formulate it into a convex form. Relaxation affects the dimension of the solution, that is, an optimized node position can end up in higher dimension than required. However, in [8] Anthony has found that if a network to be localized is uniquely realizable then even with a relaxed convex optimization a unique position estimations resulting in required dimension can be found; however, his findings were based on ideal noiseless distances. Apart from Anthony, a lot of research works have been done in applying the relaxed SDP for localization, see for example [11-17,4], where highly accurate results are demonstrated even for noisy measurements.

Although convex SDP localization generate highly accurate results yet there are two drawbacks associated with it: one, it is computationally demanding, secondly, for the relaxed SDP approximations unique realizations could not always be guaranteed for noisy distance measurements. However it has been shown that if a network is universally rigid then unique realization can be found by applying rank reduction techniques as discussed in [16, 50] at the cost of computational complexity. For the case of random networks, large number of distance measurements or larger coverage range normally results in a rigid network for which a unique realization could be achieved [15]; however, large

number of measurements means increase in computation complexity. Therefore, a novel technique is needed which is not only accurate but is also computationally efficient.

Recently various techniques such as [4,9-14] have been proposed to reduce the computational complexity but most of these techniques are proposed for sensor networks which are densely populated and are not an optimum solution for WLAN; where, number of nodes and their mobility level is highly diverse as compared to sensor networks.

In our developed solution we have divided large networks into smaller subnets consisting of computationally manageable number of nodes. Then these subnets are localized iteratively in a time division multiplexed way, that is, one subnet is localized at a time then in next time slot another subnet is localized. The proposed method is similar to 4G networks serving multiple users with data packets in a time division multiplexed access technique. With proposed method only a portion of nodes within a network are localized and the entire network is localized incrementally in time. Hence resultant network is highly scalable, that is, if number of nodes within a network increases it will result in more subnets which are then localized one at a time.

Furthermore nodes present within the WLAN are classified as mobile and stationary based on their mobility levels, then all mobile nodes are placed in subnets marked as mobile while stationary nodes are placed in stationary subnets. The reason for this classification is that for a real time tracking, only mobile node positions will be changing compared to stationary nodes. We can reduce the computational complexity by localizing mobile subnets more frequently compared to stationary nodes. With this priority based localization, nodes can be localized in a real time. Computational complexity is reduced further by sparsifying the network. It is achieved by localizing nodes using distance measurements between at least three anchors for the case of a 2-dimensional (2D) localization and at most two additional distances between neighbouring nodes for more accuracy with a condition that node-node distance measurements are measureable. This

constraint will generate a trilateral network: a network which can be universally rigid and unique position estimations can be found for this kind of network.

We have also developed a 3D distributive localization technique exploiting the fact that the rate of change of a mobile WLAN node user in the horizontal plane will be greater than the rate of change in vertical direction, that is, change in floors for the case of a multi-storey building; therefore, for real time tracking in 3D, the computational complexity can be reduced by estimating positions of nodes iteratively in the 2D and 3D fashion. 2D positions being more frequently estimated compared to estimations in 3D. During 2D position estimations, only the horizontal positions along the ground are estimated. While the height estimates which are the 3rd dimension of positions are retained from previous 3D estimate with an assumption that no change was occurred in vertical direction of the nodes resulting in the overall 3D localization.

We have further found in the 3D localization that errors in estimating the heights of the nodes are much greater compared to planar position estimations. However, it is possible to estimate the floor in which a node exists for the case of a multi-storey building. For many localization applications node's floor estimates will be sufficient instead of node's actual height from the ground.

1.2 Performance Metrics in Localization

Kaemarungsi in [5] listed the most critical performance metric in localization as accuracy which is defined as the difference between the actual node position and its estimated location. Another related parameter to accuracy is called location precision which is a confidence interval with successful location estimations. Apart from these, other performance metrics are delay, capacity, coverage and scalability of the localization technique. Delay refers to the time required in localizing a node. The capacity metric refers to number of node estimations a system can process per unit time. Coverage metric

refers to a boundary of an area within which a system can localize nodes. The scalability metric concerns with how well the system responses to large number of nodes and larger coverage. All of these performance metrics depend on type of measurement adopted, channel characteristics, bandwidth of the measured signal, and the complexity of the localization algorithm used.

In our dissertation we have proposed a distributive localization algorithm for WLANs which is highly scalable and have a traceable capacity.

1.3 Achievements

First of all we gave overview and review of the localization problem and presented analytical and theoretical formulation of the localization problem adopted from distance geometry, rigidity theory and Euclidean distance geometry. Then convex optimization techniques used in the localization problem were reviewed by considering convex formulation of localization based on the Euclidean Distance Matrix and convex formulation of localization based on minimizing *distance square errors*. The major achievements and our contribution in WLAN localization includes:

Derivation of performance benchmarks which are based on Cramer-Rao Lower Bounds (CRLB) and Circular Error Probability (CEP). Then two types of estimators i.e. Weighted Least Square (WLS) and convex semidefinite programming based estimations are analyzed by deriving expressions for their performance bounds. Based on these bounds and simulations, both types of estimators are compared. This work is presented in chapter 3.

Development of a novel distributive localization technique for WLAN is presented in chapter 4. The proposed technique is highly scalable, and can be used for real time tracking. It is achieved by dividing the large number of WLAN nodes into computationally manageable subnets. Then these subnets are localized in a time division multiplexed way by scheduling subnets in time domain with higher priority given to subnets containing

mobile nodes. Apart from being highly scalable, the proposed algorithm is always traceable. The rate with which nodes are localized can always be computed which helps in applying the proposed distributive localization algorithm for real time tracking of mobile nodes. In addition to this, problem of a high rank position estimations arising in convex optimization due to relaxations is mitigated by introducing virtual nodes which are not real nodes to be localized but system inserted nodes. Simulation results show improvements in performance by adding virtual nodes within a network. All this work is presented in Chapters 4 and is published in [18].

3D localization is considered to be a challenging task with very few contributions available in literature. 3D localization becomes necessary in multi-storey indoor environments where nodes are required to be identified within multi-floors. Research in this thesis also presents development and simulation of a computationally efficient distributive 3D localization technique that exploits mobility patterns across floors. The rate with which nodes change floors is always less compared to the rate of change in planar positions. This mobility pattern is used to develop an efficient 3D localization technique which localizes mobile nodes in planar space more frequently compared to vertical floor changes. This work is presented in chapter 5.

1.4 Outline of the Thesis

Research work presented in this thesis focuses on WLAN localization. In chapter 2, we present fundamental concepts of localization, reviewed localization techniques, and formulate the localization problem. Chapter starts with brief review of terms used in distance geometry, rigidity theory and convex optimization which are used to formulate a localization problem for convex optimization. Then least square and convex semidefinite programming (SDP) optimization techniques are listed where localization problem is formulated to be solved using SDP. Finally, the signal model used in the analysis is

presented. In addition, we present the known results from different references in a connected manner to help scientists and researchers to understand and perform further research in localization.

Benchmarks to measure the performance of the estimations are listed in chapter 3. Where two types of bounds are derived: one based on Cramer-Rao Lower Bound (CRLB) for accuracy calculations, and other to quantify the accuracy based on the Circular Error Probability (CEP). Then using simulations, nodes are localized using least square and SDP based localization techniques. Performance of these techniques is evaluated against these bounds. We have found that convex based optimizations outperform the non-convex least square optimizations and this work is presented in chapter 3.

After evaluating the performance of convex based localizations in chapter 3, the novel distributive time division multiplexed WLAN heuristic is described in chapter 4. Computational complexity of the convex optimization is evaluated and then performance of the newly developed distributive localization technique is evaluated based on the scalability, capacity and accuracy. The developed technique is extended to 3D distributive localization which is presented in chapter 5.

Finally chapter 6 concludes the dissertation with future directions.

Chapter 2

Localization

This chapter first of all introduces notations and defines terms used within this dissertation to formulate a localization problem. Then a localization problem is mathematically formulated based on inter node distances using theory from distance geometry, and rigidity. The resultant formulation (consisting systems of equations), generally, does not give a unique solution. Additional constraints are, therefore, needed for guaranteeing a unique solution. These constraints are generated by using theory of rigidity for achieving a unique solution in 2D and 3D environments which is explained in this chapter. Literature review of localization techniques is also presented in this chapter. Least square and convex optimization techniques used in localization have also been discussed. Furthermore, SDP based convex approximation to an original non-convex problem is also presented (section 2.5.1). Finally, the signal model used to generate measured distances for localization is also explained within this chapter.

2.1 Localization Problem and Measurement Metrics

Localization as a process can be divided into two sub-processes: metric measurement sub-process and a localization algorithm sub-process which processes these metrics to determine locations. Metric measurement sub-process measures either approximate Angle Of Arrival (AOA) or approximate distance between some unknown nodes and known nodes (nodes whose positions are known). The Time Of Arrival (TOA), Time

Difference Of Arrival (TDOA) or Received Signal strength (RSS) are commonly used metrics to estimate distances between the nodes. TOA based measurements are proven to be more accurate for indoor environments and are commonly used [19].

Work in this thesis focuses on localization algorithm based on distance measurements between some nodes whose locations are known (termed hereafter as anchors) and nodes whose positions need to be determined (termed hereafter as nodes).

The localization based on distance measurements is formulated using the field of distance geometry, therefore, in next section we introduce notations and terminologies from the distance geometry, rigidity theory and Euclidean distance matrix which are used within this thesis to mathematically formulate and analyze a localization problem.

2.2 Definitions and Notations

In this section, selective scattered terms from theory of distance geometry, Euclidean distance matrix, convex optimization and rigidity are defined. These terms are used to formulate a problem for localization. Purpose of this effort is to familiarize the reader with scattered terms from different aforementioned fields and make thesis easily readable and understandable with minimum consultancy to references. For example, definitions 2.1 to 2.6, and 2.12 to 2.14 are related to formulate a localization problem for convex optimization, definitions 2.7 to 2.11 taken from rigidity theory are used to derive conditions under which a unique solution to a location problem given in equations 2.3 can be guaranteed. Furthermore some simple examples are added to further elaborate the concepts.

Measured distances always have noises due to which localization problem has to be solved using optimization. Problem formulated given in equations 2.3(a-c) is a non-convex optimization problem; therefore, it has to be transformed into convex form for accurate estimates. Within convex optimization, a cost function and constraints must be convex.

Sections 2.5 to 2.6 explain how a non-convex localization problem can be approximated into convex form. It is achieved using a double centring transformation given in equation 2.9. Convex constraints are derived using equation 2.9 and are given in equations 2.10a-c. Finally, localization problem is formulated for convex optimizations using two approaches: one based on Euclidean distance matrix given in equation 2.12 and other based on minimizing distance square errors given in equation 2.13. This has been explained in section 2.6.

The real number system is denoted by R, and R^d is the vector space of real *d*-tuples **x**=(ξ_1, ξ_2, …, ξ_d). Consider a network in *d*-dimensional Euclidean space (R^d) with *N* number of nodes including: N_a the number of nodes whose positions $\mathbf{a}_i \in R^d$, i=1,2,…,N_a are known (termed as anchors) and N_n the number of nodes whose positions $\mathbf{x}_i \in R^d$, i=1,2,…,N_n need to be determine. Let the distance between a node '*i*' and node '*j*' is denoted as d_{ij}.

Matrices within this thesis are denoted by capital bold letters, while vectors are denoted by small bold letters. Estimate of any variable 'θ' is denoted by $\bar{\theta}$. **I** and **O** denote the identity and all zero matrices respectively.

Let **X** be the matrix containing the position coordinates of the total number of nodes (*N*) in a *d*-dimensional network given by:

$$\mathbf{X} = [\mathbf{x}_1 \cdots \boldsymbol{x}_N] \in R^{d \times N}$$

Where $\mathbf{x}_i$ is the vector containing position coordinates of node '*i*' in R^d, for d=2 it is written as:

$$\mathbf{x}_i = (x_i, y_i)$$

Next we have defined some terms used hereafter.

Definition 2.1 *Trace or inner product*: Trace of the matrices **A**, **B**$\in R^{n\times n}$ denoted by tr(**A**) is the standard trace inner product between **A** and **B** given as:

$$\langle \boldsymbol{A}, \boldsymbol{B} \rangle = \sum_{i,j=1}^{n} \boldsymbol{a_{ij}}\, \boldsymbol{b_{ij}} = \boldsymbol{tr}(\boldsymbol{A^T B})$$

It is the sum of the elements on the diagonal of $\mathbf{A^T B}$ where $\mathbf{A^T}$ is the transpose of a matrix.

Definition 2.2 *Euclidean norm or l_2-norm*: Euclidean norm of a vector $\mathbf{x}\in R^n$ is defined as

$$\|\mathbf{x}\|_2 = \left(\mathbf{x}^{\mathrm{T}}\mathbf{x}\right)^{1/2} = (x_1^2 + \cdots + x_n^2)^{1/2}$$

Definition 2.3 Frobenius *norm:* Frobenius norm of a matrix $\mathbf{A} \in R^{n\times n}$ is given by

$$\|\mathbf{A}\|_F = \left(tr(\mathbf{A}^{\mathrm{T}}\mathbf{A})\right)^{1/2} = \left(\sum_{i,j=1}^{n} a_{ij}^2\right)^{1/2}$$

Definition 2.4 *Positive definitive matrix*: An *n* × *n* symmetric matrix **A** is said to be positive definite if:

$$\mathbf{x}^{\mathrm{T}}\mathbf{A}\mathbf{x} > 0, \quad where\, \mathbf{x} \in R^n \; and\, \mathbf{x} \neq 0$$

If the inequality is replaced as:

$$\mathbf{x}^{\mathrm{T}}\mathbf{A}\mathbf{x} \geq 0, \quad where\, \mathbf{x} \in R^n \; and\, \mathbf{x} \neq 0$$

then **A** is said to be positive semi-definite matrix and is denoted as **A** $\succcurlyeq$ 0.

In addition, according to [46, Theorem 15.2] for positive semidefinite matrices, following are equivalent statements:

1. **A** $\succcurlyeq$ 0
2. $\lambda_i \geq 0$ for i =1, 2, …, n, where λ_i are the Eigen values of **A**
3. **A** = $\mathbf{X}^{\mathbf{T}}\mathbf{X}$ where $\mathbf{X} \in R^{n \times n}$

The third statement implies that **A** is positive semidefinite if it has a matrix square root, similar to the fact that a real number is said to be positive if its square root is real.

Furthermore, for two symmetric matrices **A** and **B**, **A**$\succcurlyeq$**B** means **A**-**B**$\succcurlyeq$0, i.e. **A**-**B** is a positive semidefinite matrix. The importance of semidefinite matrix comes from the fact that the set of all positive semidefinite matrices forms a convex cone with well defined boundaries.

Definition 2.5 *Euclidean Distance Matrix* (EDM): EDM also known as pre-distance matrix **D** is the $n \times n$ symmetric and nonnegative matrix of distance squared whose entries are given by:

$$D_{ij} = \left\|\mathbf{x_i} - \mathbf{x_j}\right\|_2^2, \quad i, j = 1, 2, \ldots, N \tag{2.1}$$

Definition 2.6 *Gram matrix* (**G**)**:** Positive semidefinite matrix defined as the inner product of **X** is known as a Gram matrix i.e.

$$\mathbf{G} \triangleq < X, \mathbf{X} >= \mathbf{X}^T \boldsymbol{X} \tag{2.2}$$

Definition 2.7 *Graph*($\mathcal{G}$): A graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ is a set of vertices $\mathcal{V}$ together with a nonempty set $\mathcal{E}$ of edges along with a real number associated with each edge.

Definition 2.8 *Trilateral graphs:* Let l be an integer such as $v \geq l + 1$, for a graph $\mathcal{G}$=($\mathcal{V}$, $\mathcal{E}$) with v vertices is said to be trilateral graph if there exists an ordering {1, 2, …, v} of the vertices in $\mathcal{V}$ such that:

a) The first $l + 1$ vertices form a complete graph, and

b) Every vertex $j \geq l + 1$ is connected to at least $l + 1$ of the vertices 1, 2 , …, j-1

Definition 2.9 *Framework*: A framework $\mathcal{G}(\mathcal{X})$ in R^d is a pair ($\mathcal{G}$, $\mathcal{X}$), where $\mathcal{G}$ is a graph and $\mathcal{X}$ is a point such that:

$\mathcal{X} = (x_1, x_2, \ldots, x_v) \in R^{dxv}$

$\mathcal{X}$ is also referred as in [20] a *configuration* of *v* points in R^d, such that for every *i*=1,2,…,*v*, vertex i of $\mathcal{G}$ is located at x_i.

Definition 2.10 *Graph realization*: [8]: In distance geometry, graph realization is a problem in which a graph $\mathcal{G}$ is given along with a real number associated with each edge {w_{ij}, *i*,

$j \in \mathbb{V}\}$, and the goal is to assign coordinates to each vertex so that Euclidean distance between any two adjacent vertices is equal to the real number associated with that edge.

Graph realization problem of distance geometry can be related to localization such as vertices of $\mathcal{G}$ correspond to nodes, vertices can further be partitioned into two sets: one whose positions are known i.e. anchors, and second whose positions needs to be determined i.e. nodes. Edges correspond to communication links between nodes and a real number associated with edges as distances between nodes. The goal in localization is then to determine positions of the nodes (graph realization) given the anchors and distances between some nodes and anchors (d_{ij} is equivalent to w_{ij} in a graph realization). With the presence of anchors, the positions of some vertices in the graph realization problem are constrained to be fixed.

Mathematically distances between nodes-anchors and between nodes-nodes and between anchors-anchors are written as:

$$\left\|\mathbf{x}_i - \mathbf{x}_j\right\|_2^2 = d_{ij}^2 \quad \forall\ (i,j) \in I_{an} \tag{2.3a}$$

$$\left\|\mathbf{x}_i - \mathbf{x}_j\right\|_2^2 = d_{ij}^2 \quad \forall (i,j) \in I_{nn} \tag{2.3b}$$

$$\left\|\mathbf{x}_i - \mathbf{x}_j\right\|_2^2 = d_{ij}^2 \quad \forall (i,j) \in I_{aa} \tag{2.3c}$$

where $\mathbf{x}_i \in \mathrm{R}^d$ and I_{an}, I_{nn} and I_{aa} are a set of indices of anchors-nodes, nodes-nodes and anchors-anchors with measurable distances respectively.

Localization will then be the solution in R^d space to the set of quadratic equations given in (2.3a-c) with the conditions that there exist at least d+1 non collinear anchors and a unique solution exists (see example 2.1 for further details). Generally, there could be

more than one solution to these set of equations, meaning multiple positions for a node or nodes which is not possible. The conditions under which a unique solution is guaranteed are studied in rigidity theory for frameworks [4, 21-22,65]. The same theoretical work is adopted and extended for a localization problem see for example [4, 23-25].

In rigidity theory, it is stated that if a network is *globally rigid* then there will be a unique solution to the set of quadratic equations 2.3 a-c [4]. Global rigidity states that **X** is the only realization in R^d up to rigid motion or stated as [21]:

Definition 2.11 *Globally rigid:* Two frameworks one $\mathcal{G}(\mathbf{X})$ with a realization **X**, and for any other realization **Y**, framework $\mathcal{G}(\mathbf{Y})$ in R^d are said to be *equivalent* or congruent, when

$\|\mathbf{x_i} - \mathbf{x_j}\| = \|\mathbf{y_i} - \mathbf{y_j}\|, \ \forall i, j \in N$ (where N is total number of vertices and x_i, $y_i \in R^d$)

written as:

$$\mathcal{G}(X) \equiv \mathcal{G}(Y)$$

Framework obeying these conditions is said to be globally rigid [21,24,26,27]. Figure 2.1 to 2.3 depict non-rigid, only rigid and globally rigid graphs in R^2. The condition of only rigidity framework must be such that there does not exist continuous deformations to reach to other possible realization, as shown in figure 2.2, there exists two realization of the graph but graph cannot be continuously deformed by satisfying all the distance constraints i.e. by keeping distances fixed to reach for other realization.

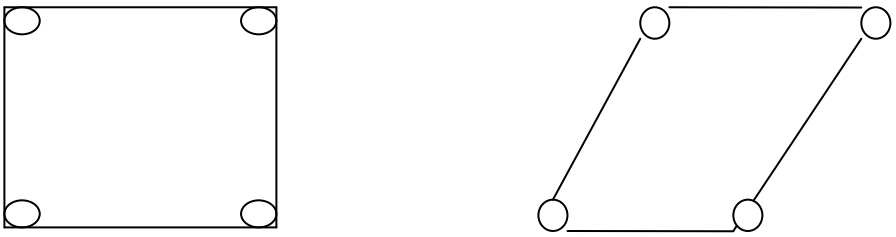


Figure2.1: Non rigid graph having many realizations satisfying distance constraints

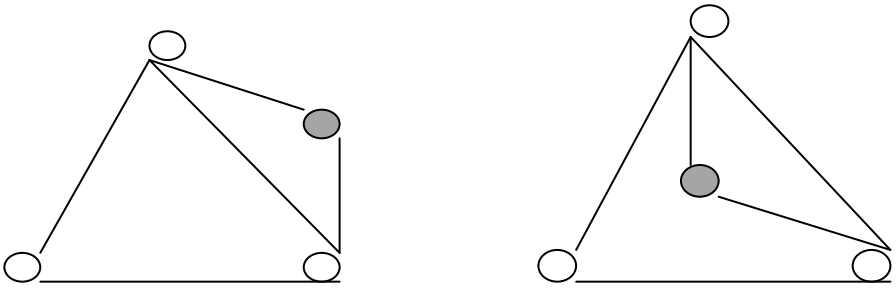


Figure 2.2: A rigid graph having two realizations but it is not possible to move a shaded vertex continuously to approach to other realization by keeping all distances fixed as is the case for the graph of figure 2.1.

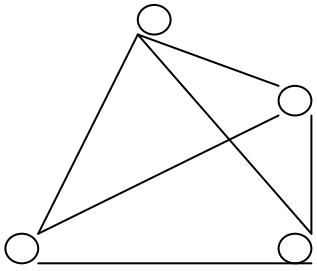


Figure2.3: A globally rigid graph having unique realization.

In conclusion, if a network with measured distances is globally rigid and anchors are not collinear and at least *d*+1 anchors exist then by solving set of quadratic equations (2.3a-c) unique positions of nodes could be determined.

Then next step is to solve quadratic equations (2.3 a-c). These equations can be solved directly; example 2.1 demonstrates graphical solution to equations 2.3.

Example 2.1

Consider an Euclidean 2-D space (R^2) consisting of two anchors with anchor-node measurements. Let us look at the graphical solution of this problem. The plot of equations 2.3 will be two circles centred at anchors having radius as anchor-node measurements, depicted in figure 2.4. The only case in which we can find a unique solution to this problem is when an unknown node exists at the straight line joining two anchors as shown in figure 2.4 (a), otherwise there will always be two solutions as depicted in figure 2.4 (b). Line drawn from anchors to point of intersection represents measured distances between anchors and node. If we add another collinear node to this problem, a unique solution cannot be found as shown in figure 2.4 (c). Hence for a unique solution in R^d, we need at least *d*+1 non–collinear anchors as shown in figure 2.4(d).

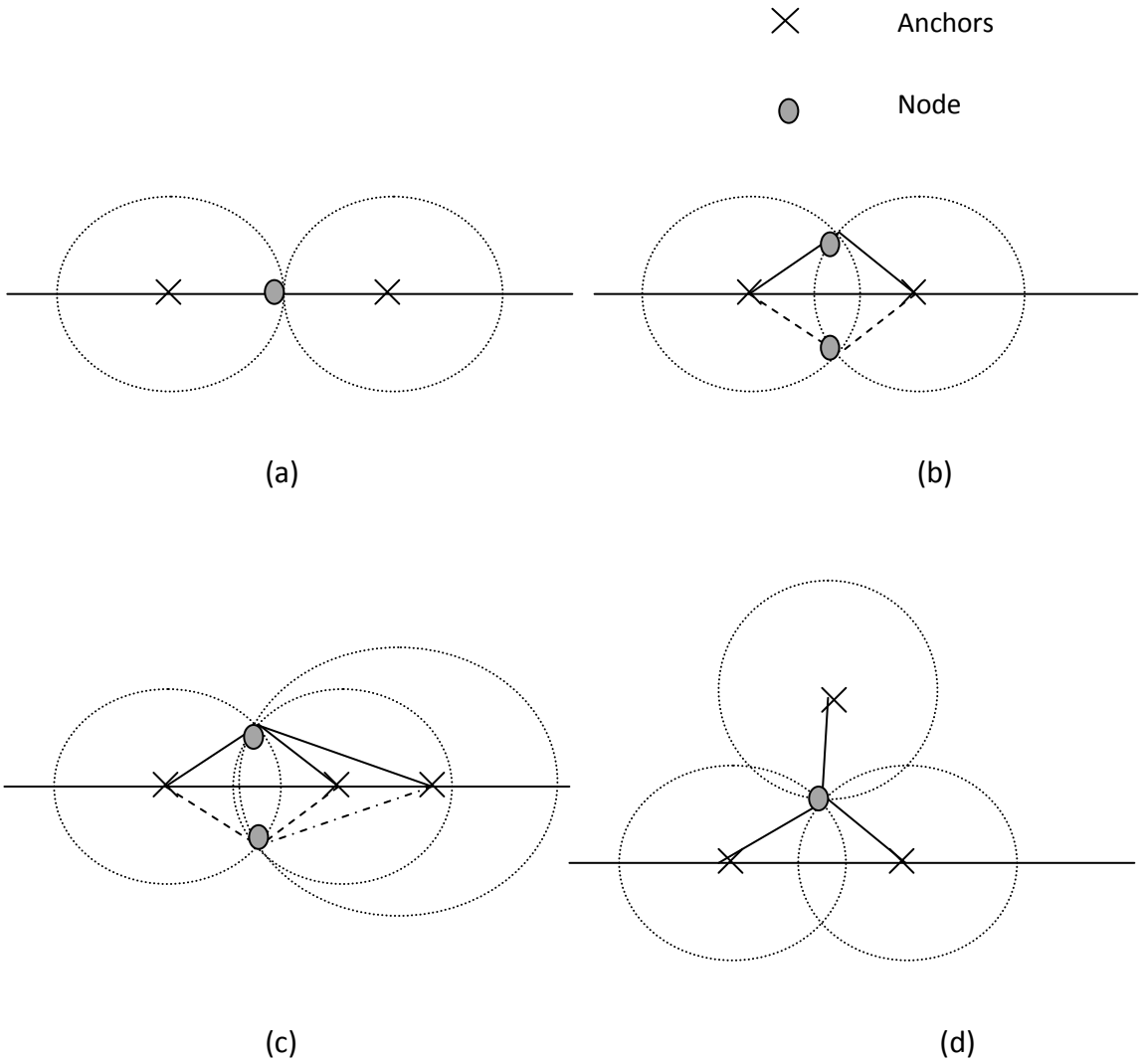


Figure 2.4: Graphical solution of equations 2.3, a) a unique solution, b) two solutions i.e. intersection of two circles, c) two solutions i.e. intersection of three circles d) a unique solution

Let us now add noise to distance measurements. With noise added, we can have a number of possibilities to our graphical solution as depicted in figure 2.5 a-c.

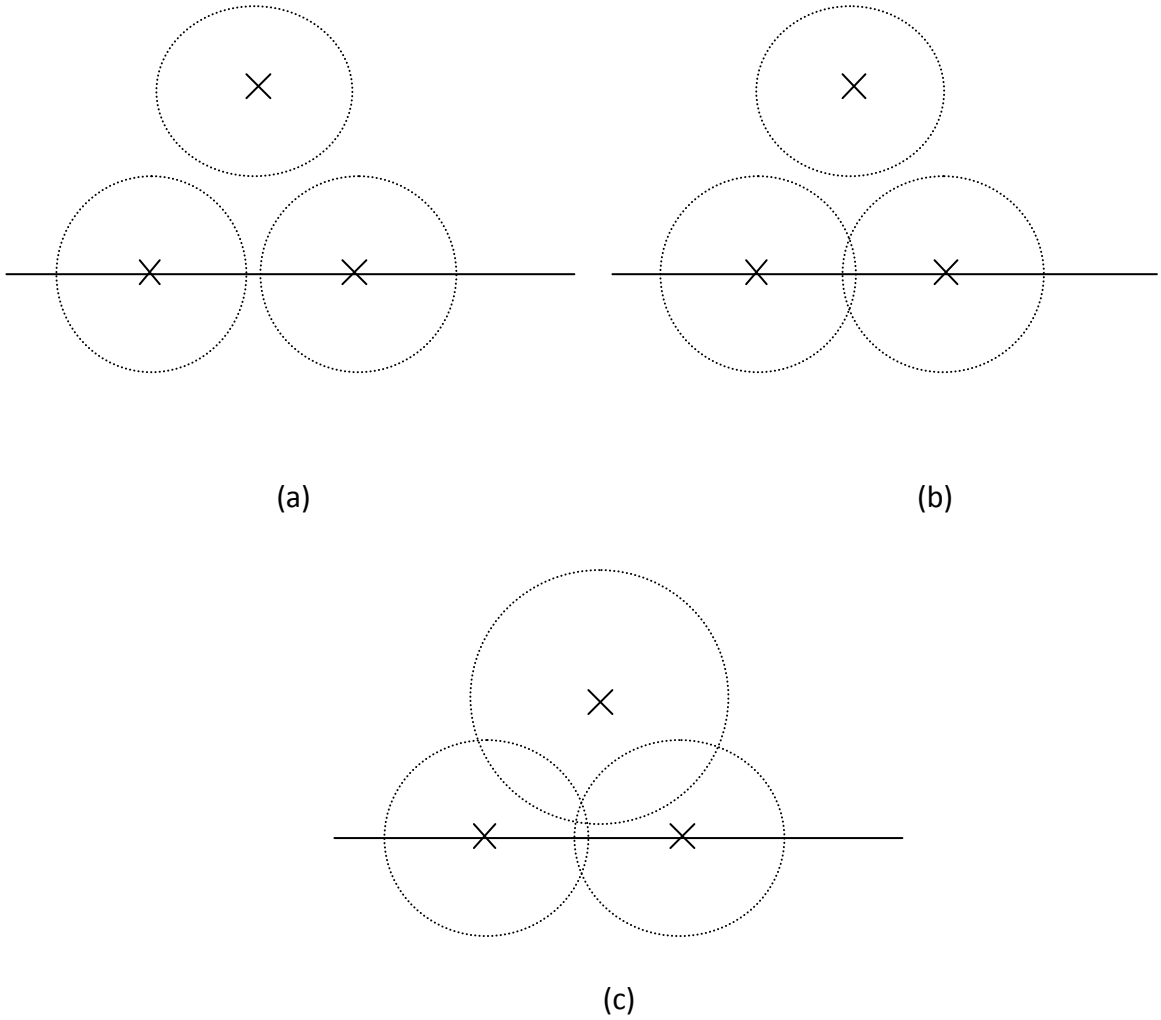


Figure 2.5: Cases (a)-(c) where due to presences of noises within distance measurements a unique solution cannot be found by solving equation 2.3 directly.

Due to noisy distance measurements there might not be a unique solution or any solution at all, see figure 2.5 (a)-(c). Therefore, equations are not generally directly solved, instead optimization techniques are applied to mitigate effects of noise.

Recently novel techniques have been proposed for indoor localization covering sensor networks and WLANs. Example given in [9] summarizes various techniques. In [27-29], localization is performed using multi-dimensional scaling (MDS), while [30, 31] use Cayley-

Menger determinant. Similarly, a technique based on DV hop method suitable for densely populated sensor networks is also proposed in [32].

In [31], authors proposed a technique to mitigate noise in distance measurements using Cayley-Menger determinants. When the equality constraints are derived from Cayley-Menger determinants, these constraints can then be used with any estimation technique to further improve results. As per our knowledge, this technique has not been used within a convex optimization.

Some of the terms used within convex optimizations are defined as:

Definition 2.12 *Affine function or affine transformation [87]:* the affine transformation f from R^n to R^m is the mapping of the form linear plus constant i.e.

$$f(x) = Ax + b$$

If **F** is a matrix valued function i.e. **F** mappings from R^n to $R^{p\times q}$, then **F** is affine if

$$\mathbf{F}(x) = \mathbf{A_o} + x_1\mathbf{A_1} + \ldots + x_n\mathbf{A_n}$$

where $\mathbf{A_i} \in R^{p\times q}$ is a linear transformation.

Definition 2.13 *Convex Set [87]:* A subset C of R^n is a convex set if it contains the line segment joining any of its points, i.e. if

$$(1-\theta)x + \theta y \in C, \qquad 0 < \theta < 1$$

whenever $x \in C$, and $y \in C$

Some graphical examples of convex and non convex sets are given in figure 2.6:

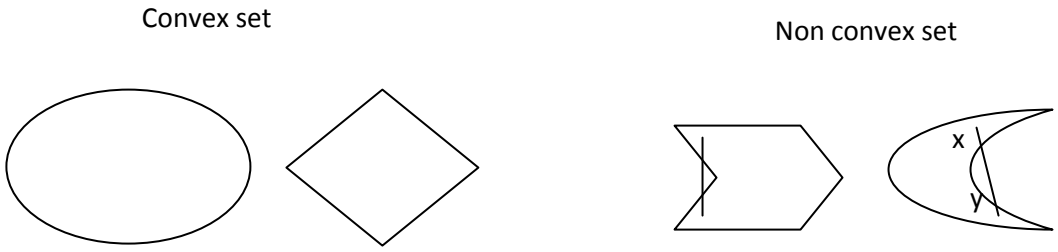


Figure 2.6: Examples of convex and non convex sets

Definition 2.14 *Convex function [87, theorem 4.1]:* A function *f* from *C*=R^n to R, where *C* is a convex set, then *f* is convex on *C* if and only if

$$f(\theta x + (1-\theta)y) \leq \theta f(x) + (1-\theta)f(y), \quad 0 < \theta < 1 \qquad (2.4a)$$

For every $x \in C$ and $y \in C$.

Geometrically, equation 2.4a implies that the line segment between (x, $f(x)$) and (y, $f(y)$), which is the chord from x to y lies above the graph of *f*, as shown in figure 2.7.

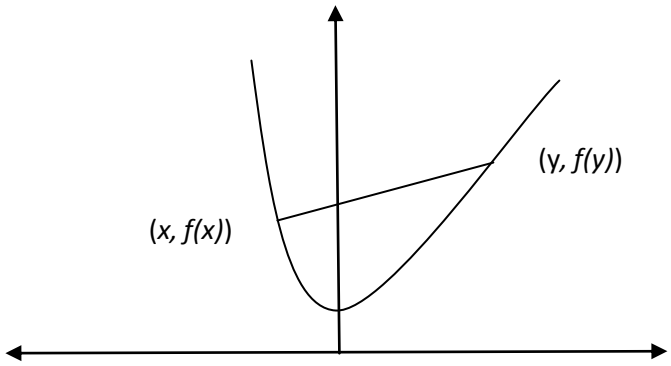


Figure2.7: Graphical representation of a convex function.

Some examples of convex function are:

1. $f(x) = x^2$
2. $f(x)=1/x$
3. $f(x) = ax + b$
4. $f(x) = x^2/y, \quad y>0$

For more than two variables, convex functions are given by:

$$f(\theta_1 x_1 + \cdots + \theta_m x_m) \leq \theta_1 f(x_1) + \ldots + \theta_m f(x_m), \tag{2.4b}$$

whenever

$$\theta_1 \geq 0, \ldots, \theta_m \geq 0, \qquad and\ \theta_1 + \cdots + \theta_m = 1$$

Examples of convex functions of matrices are:

1. $f(\mathbf{X}) = \text{Tr}(\mathbf{A}^T\mathbf{X}) + \mathbf{b}$
2. $f(\mathbf{X}) = \|\mathbf{X}\|_2$

2.3 Literature Review

Research work in an indoor localization was carried out following two paths; one covering localization of nodes in WLANs and other area covering localization of sensors in sensor networks. Separate localization techniques have been developed for these two types of indoor networks and review of these techniques is given in this section.

Sensor networks, which are different to WLANs, are assumed to be small, inexpensive, cooperative and deployed in large numbers [7]. These are typically utilized for monitoring and controlling homes, cities and the environment. Any localization technique proposed for one type of network, although, theoretically can be applied to other type but may not be an optimum solution for the other-one. Hence localization techniques are optimized based on network types.

The reason for including sensor network localization within review is due to the many proposed convex based localization techniques for sensor networks in literature [8,14-17, 41-43,57]. As this research work has developed a convex based localization technique for WLANs, therefore, literature review in this field is used to compare and emphasize our contributions within an existing field.

2.3.1 Senor Network Localization

Different localization techniques are proposed for wireless sensor networks, for example, in order to exploit the smaller range within which network is deployed, DV-hop based techniques are proposed [88-89]. In this technique, instead of measuring actual distances between nodes, the average distances between sensors and anchors are estimated by using the minimum hop information between the two. This technique is promising as well as cost effective only if a network is densely populated and has smaller coverage area. In addition to DV-hop method, cooperative techniques also exist for densely populated sensor networks and are presented in [14, 27, 90]. In these techniques, sensors cooperate with each other in order to localize other unknown sensors. In other words, localized sensors act as anchors and further help to localize unknown sensors.

In addition to connectivity based DV-hop approach, measurement based approaches also exist such as distance measurements based techniques. In these approaches distances between nodes are obtained from signal strength, time of arrival (TOA), time difference of arrival (TDOA), carrier signal phase of arrival (POA) or angle measurements based on angle of arrival (AOA) measurements. The indoor radio channel suffers from multipath propagation and shadow fading, due to which AOA and signal strength based measurements provide less accurate results as compared to TOA measurements [91]. Therefore, TOA based distance measurements are recommended for indoor environments if accuracy is to be achieved.

AOA and signal strength based measurements are less accurate as compared to the TOA based distance measurements [91]. However, there might not be a significant difference in localization performance by using either signal strength or TOA based localization techniques in the case of sensor networks. This is because the sensor networks are deployed in smaller range as compared to WLANs. Measurement errors are a function of distances between nodes [39]. This means that for WLANs having large coverage area, errors in measurements will be significant, if signal strength based techniques are applied. Therefore, signal strength based techniques give coarse estimations [95].

As mentioned earlier in this thesis, distance based localization is a non-convex optimization. This limitation was overcome by relaxing the problem into convex SDP localization form [8], resulting in highly accurate localization. There are two drawbacks associated with this approach; one is computational complexity, secondly unique realizations could not always be guaranteed for the relaxed SDP approximations with distance measurements corrupted with noise. However, it has been shown that if a network is universally rigid, unique realization can be found by applying rank reduction techniques as discussed in [16, 50] at the cost of computational complexity. For the case of random networks, large number of distance measurements or larger coverage range

normally results in a rigid network for which a unique realization could be achieved [15]. However, large number of measurements result in computation complexity. Therefore, a novel technique is needed which should not only be accurate but should also be computationally efficient.

Recently, various techniques such as proposed in [4,9-14, 17] have reduced the computational complexity. However, most of these techniques are proposed for sensor networks which are densely populated and are not an optimum solution for WLAN, where number of nodes and their mobility level is highly diverse as compared to sensor networks. In the following paragraph, we have reviewed some of these techniques.

To make the SDP based localization computationally efficient, a novel edge sparsification technique is proposed in [13], in which the number of measured edges (distances between nodes) is reduced while preserving localization properties. Furthermore, theoretical relationship is derived for a universally rigid network and trilateral graphs are proved to be universally rigid. On the other hand, in [14], an iterative distributive technique is proposed where a larger network is divided into subnets based on geographical positions. In this case, only sensors with the capability of measuring distances between anchors within the subnets, are localized initially. Then, localized sensors are turned into anchors which further help in localizing other unknown nodes Thus overall technique is cooperative, that is, sensors cooperate to further localize unknown nodes. In this technique, any error incurred could propagate. Furthermore, the technique is ad-hoc and there is no measure for a subnet to be uniquely realizable. The technique is proven to generate unique positions for a network which is densely populated. WLANs are not densely populated as compare to sensor networks, therefore, proposed technique is not feasible for localization. Similarly, a distributive heuristic in which a larger network is divided into smaller subnets is proposed by the name of SPASELOC [17], where subnets are formed based on some pre-defined rule sets. Again, it

is proposed for the sensor networks and is not an optimum solution for WLANs based on the diverse node densities and their mobility levels.

In addition, signal strength based co-operative and non co-operative SDP localization techniques have been proposed in [92] for sensor networks where it has been shown that SDP based localization outperforms least square and maximum likelihood-based optimizations.

2.3.2 WLAN Localization

WLANs are different from sensor networks in terms of node densities, that is, nodes in WLANs can vary from few to large numbers while nodes in sensor networks exist in very large numbers [7]. Network size, also known as coverage area, is much greater in WLANs as compared to sensor networks. Additionally, prime services offered by WLANs are time sensitive broadband multimedia services in addition to localization whereas most applications of sensor networks are environmental monitoring, search and rescue services [95]. Hence WLANs are planned and optimized in order to maximize the data rates and coverage. With these characteristics and the existence of a complex multipath rich indoor channel, localization for WLANs has become a real challenge.

A lot of research work has been done for WLAN localization based on Received Signal Strength (RSS) measurements because of simplicity of the technique [95-101]. RSS based methods can be further categorized as trilateration and fingerprinting. In trilateration technique, RSS values are converted into distances by using signal propagation model and then trilateration is applied [101]. Therefore, the accuracy of trilateration depends very much on the surrounding environment, which for the case of indoors is not sound. Accuracy achieved in this case is about 8 meters to 15 meters [97]. Such location estimations are unable to provide context aware services based on user locations.

Fingerprinting, on the other hand, performs localization by using location dependent parameters of measured radio signals in the region deployed. It is divided into two phases, the off-line phase and the on-line phase. In the off-line phase, RSS values received from every access node are collected at different points in the area within WLAN coverage. These measured RSS values at are then stored in the database.

During the on-line phase, the RSS values from every access point is measured by the node, then localization is achieved by comparing measured RSS values with the reference data stored in the fingerprint. Furthermore, in on-line phase, measured signal strength is matched to the stored data either using deterministic approach or probabilistic [97].

Deterministic matching uses a simple signal strength map in which each location has a list of access nodes within range and an average value of signal strength for each access point. Matching might be either on one point or on several points, whose coordinates are averaged.

Probabilistic matching requires more data in the signal strength map. Signal strength values must be described by a probability distribution. Matching is achieved by probabilistic methods based on Gaussian models or the kernel method. Fingerprint based techniques have better accuracy of around 2 meters as compared to triangulation. Therefore, Fingerprint based techniques are widely used in the indoor location estimation.

Although fingerprinting technique has been shown to provide promising results but has many limitations, such as applying it in a greater scale is not practical. This is because it is very time consuming to collect the RSS measurements of all locations in a large area. This large amount of measurements will also cause the storage problem. Furthermore, it is very sensitive to the surrounding environments, thus re-calibration or re-collection of data is often required. Another limitation is that most of the on-line phase algorithms have

been developed on the assumption that the devices used for training and position phase perform identically. There are extremely wide arrays of WLAN chipsets in the market, which are built-in on variety of devices such as laptops, iphone, USB-devices etc. This results in variation in RSS measurements. These variations in measurements due to different devices have been studied aggressively in [102]. It is found that different WLAN devices perform significantly differently, even those which have come from the same vendor. It is also found that even two identical models of same vendor did not perform identically. Moreover, some devices are entirely unsuitable for positioning purpose as they report bogus RSS values [102]. In addition to this, RSS values also depend on antenna design, hardware design, drivers and environment. All these factors have effect on accuracy of the localization. Therefore, accuracy reported during experimental setup cannot be achieved in a real environment where different WLAN devices are used. To minimize the aforementioned dependence of RSS values on hardware, heterogeneous signal model and algorithms to circumvent this problem are proposed in [103]. The achievements of the proposed technique is proven using simulations only, an actual experimental verification using off the shelf devices is required for more validation. With these limitations, RSS based localization are only suitable for coarse sensor localization [95].

Apart from RSS, AOA based localization is reported in [104]. Limitation of AOA based techniques is that it requires a directional antenna. It is also difficult to design AOA sensing devices with small form factor and low energy consumption [101]. Furthermore, the indoor radio channel suffers from multipath propagation and shadow fading due to which AOA based measurements provide less accurate results [91].

Time based localization techniques are considered to be highly accurate [101] as compared to RSS and AOA based techniques. Localization using TOA is proposed in [105], where TOA based trilateration algorithms and non convex least square optimizations are

used for localization. Although least square optimizations are computationally efficient with good performance, but there are two drawbacks that algorithms depend on starting point or initial guess and a problem becomes non-convex outside the parameter of anchors which may result in large variations in performance. Furthermore, different time based techniques have been presented that reduces the effects of biased measurements by using prior statistical characteristics of the measurements [106]. Again problem of non-convexity and its effects are not considered.

Next we review work done in solving localizing problem using least square and SDP estimation techniques.

2.4 Least Square Estimations

Least square (LS) estimations are applied to localization problem in [33-35,64], while least square estimations are discussed in detail in [36-38]. In [34], authors linearized the quadratic equations 2.3a-c using Taylor series and then used LS estimations for localization. Similarly [39] used Taylor expansion to linearize quadratic equations before applying LS estimations. Apart from this, A H Sayed in [40] performed localization using LS without linearizing quadratic equations to mitigate the effects of noise. In chapter 3, we have shown that estimators based on convex optimization outperform LS estimation techniques, see also [92].

2.5 Convex Optimization and Semidefinite Programming

The maximizing or minimizing of a given function, possibly subject to some type of constraints, is known as optimization. The given function is known as the *objective function* or the *cost function*. A mathematical optimization problem has the form:

$$minimize\ f_o(x)$$

$$subject\ to\ f_i(x) \leq 0, i = 1, \dots, m \qquad (2.5)$$

$$h_i(x) = 0, i = 1, \dots, p$$

The above equation describes the optimization problem of finding a vector **x** of decision variables that minimizes cost function $f_o(\boldsymbol{x})$, satisfying constraints given by *inequality constraint* functions, $f_i(\boldsymbol{x})$, and *equality constraints,* $h_i(\boldsymbol{x})$.

Convex programming studies the case when the objective function is convex and the constraints, if any, form a convex set. Mathematically, convex optimization problem is written as:

$$minimize\ f_o(x)$$

$$subject\ to\ f_i(x) \leq 0, i = 1, \dots, m \qquad (2.6)$$

$$a_i^T x = b_i, \quad i = 1, \dots, p$$

Where f_o, …, f_m are convex functions. Comparing it with that of standard optimization problem given in equation 2.5, convex optimizations require cost function, inequality constraints to be convex and equality constraint function to be affine. Convex SDP techniques are discussed in [47].

2.5.1 Semidefinite Programming

Semidefinite programming is a subfield of convex optimization where the underlying variables are semidefinite matrices. It is a generalization of linear and convex quadratic programming. The standard mathematical representation of semidefinite programming is given as [87]:

$$minimize\ c^T x$$

$$subject\ to\ x_1\mathbf{F}_1 + \ \ldots + x_n\mathbf{F}_n + \mathbf{G} \geq 0$$

$$\mathbf{A}x = b$$

where $\mathbf{F}_i$, $\mathbf{G}$ are all positive semidefinite matrices.

2.6 Formulating Localization Problem for Convex Optimization

Localization based on distance measurements as given in equations (2.3a-c) is a non-convex problem [8]. Therefore, if non convex optimization techniques are used, it can result in large estimation errors. There exist some techniques where a non-convex problem could be formulated in a convex form. In [8], Anthony has analyzed the use of SDP a convex optimization technique to study localization problems based on distance measurements by applying some relaxations to an original problem. He has shown that if a network is uniquely realizable, then even with relaxations a unique position estimations resulting in required dimension can be found. However, his findings were based on ideal noiseless distances. Apart from Anthony, a lot of research work has been done in applying a relaxed SDP for localization; see references [41-45], where highly accurate results are demonstrated even for noisy measurements. Here we will review two heuristics used to transform a localization problem into a SDP problem. One based on EDM while other on minimizing distance square errors found in [14, 16,44], where an optimization problem minimizes a distance measure.

2.6.1 EDM Based SDP Localization

EDM based localization formulation into SDP are studied in [48-50], where localization is performed using theory adopted from EDM completion problem. Book by Dattorro [50] specifically explores the links between EDM and a convex optimization. We have briefly explained the problem formulation while details on properties of the EDM and formulation can be found in aforementioned references and in a work by Gower presented in [51,66]

Entries of EDM (D_{ij}) and those of Gram matrix (G_{ij}) are related as [52]:

$$D_{ij} = G_{ii} + G_{jj} - 2G_{ij} \tag{2.7}$$

As **G** is a function of **X** (see equation 2.2), **D** can also be written as a function of **X**, i.e. **D**(**X**) [50]. Although **D** is a function of **X** but it is not a convex function nor a positive semidefinite matrix, therefore, it cannot be directly used within optimization to find **X** given partially completed **D** (generally not all the distances between nodes are measurable). Instead, partially completed **G**, which is a positive semidefinite matrix, is used within optimizations. Once all the entries of **G** are known, **X** could be found by decomposing **G**.

On the other hand if all entries of EDM **D** are known then position matrix **X** is found by representing **G** in terms of EDM with an assumption that one of the vertex $\mathbf{x_0}$ is at origin (this corresponds to a simple translation) as:

$$G_{ij} = \frac{1}{2}\left(D_{i0} + D_{j0} - D_{ij}\right) \tag{2.8}$$

G is further decomposed to obtain the position matrix **X**. Problem can also be formulated in which matrix **G** is completed and then **X** is determined; see [50] for details.

Not all the entries of **D** or **G** are measurable or known. In this case, problem is formulated to find unknown entries of **D** or **G** with a condition that resultant matrix is an EDM for former case and for the case of **G,** it must be positive semidefinite (psd). In SDP, cost function and constraints must be positive semi definite.

A famous result by Schoenberg, also known as double centring [51-53], is used to formulate and define constraints for SDP formulation of a localization problem. The simple steps for localization using the convex formulation based on EDM similar to MDS scaling [29] are:

Step1: Compute EDM from distance measurements between nodes (not all the distances between nodes are measurable resulting in an incomplete EDM)

Step2: Apply double centring as:

$$\mathbf{F} = -\frac{1}{2}\mathbf{JDJ}, \tag{2.9}$$

where

$\mathbf{J} = \mathbf{I} - \frac{1}{N}\mathbf{1}\mathbf{1}^T$ and **1** is a vector of all ones.

If resultant **F** is a positive semidefinite matrix, then according to Schoenberg the **D** in equation 2.9 will be an EDM. In other words, the EDM **D** will result in a positive semidefinite **F** by applying Schoenberg transformation given in equation 2.9. If **F** can be decomposed as $\mathbf{F}=\mathbf{X}\mathbf{X}^{\mathsf{T}}$, then the rows of **X** give coordinates of points that generated **D**. In addition, **X** will be a matrix with least rank, explained shortly in subsequent paragraphs.

Step 3: Complete the EDM **D** using SDP optimizations and by using equation 2.9

Step 4: Compute Eigen value decomposition as

$$\mathbf{F} = \boldsymbol{Q}\boldsymbol{\Lambda}\boldsymbol{Q}^{T}$$

Step 5: Realization matrix **X** is then given by:

$$\boldsymbol{X} = \mathbf{Q}_{+}\boldsymbol{\Lambda}_{+}^{\mathbf{1/2}}$$

Dattorro in [50] found positive semidefinite matrix based constraints for SDP in terms of double centring given as:

$$tr\left(-\tfrac{1}{2}\mathbf{JDJe_i e_i^T}\right) = \|x_i\|_2^2 \quad (2.10a)$$

$$tr\left(-\tfrac{1}{2}\mathbf{JDJ}\left(\mathbf{e_i} - \mathbf{e_j}\right)\left(\mathbf{e_i} + \mathbf{e_j}\right)^{T}\right) = d_{ij}^2 \quad (2.10b)$$

$$tr\left(-\tfrac{1}{2}\mathbf{JDJ}\left(\mathbf{e_i e_j^T} + \mathbf{e_j e_i^T}\right)\tfrac{1}{2}\right) = x_i^T x_j \quad (2.10c)$$

Where vector $\mathbf{e}_i$ in equations 2.10a-c is a standard basis vector with i^{th} element as '1' and all others elements as '0'.

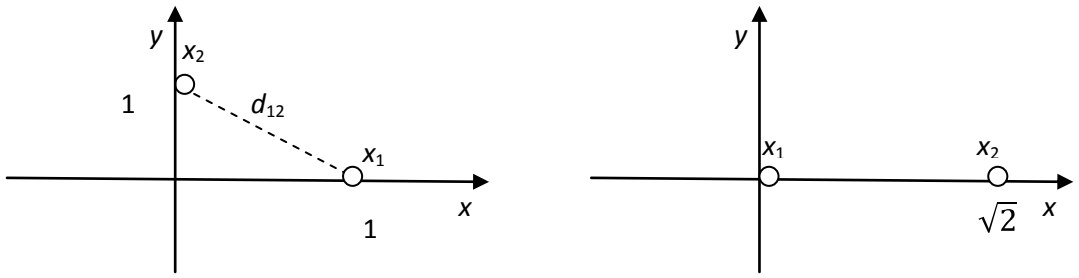
We further discuss rank and dimensionality of EDM in terms of Schoenberg criteria, which is used to formulate localization problem in SDP. The dimensionality of **D** is the minimum space containing points that generate **D**. It is, therefore, the minimum rank of a matrix **X** that generates **D**. We clarify this statement by considering two different configurations $\mathbf{X_1}$ and $\mathbf{X_2}$ with different ranks but both generating same EDM, **D**:

Let

$\mathbf{X_1} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$ be the realization with rank =2, and

$\mathbf{X_2} = \begin{bmatrix} 0 & \sqrt{2} \\ 0 & 0 \end{bmatrix}$ with realization rank =1

Realization $\mathbf{X_1}$ and $\mathbf{X_2}$ are depicted in figure 2.8.



(a) (b)

Figure2.8: (a) Graphical representation of configuration X1, and (b) graphical representation of configuration X2.

Both configurations given in figures 2.8a and b will generate same **D** a distance square EDM as:

$$\mathbf{D} = \begin{bmatrix} 0 & 2 \\ 2 & 0 \end{bmatrix}$$

In this case, dimensionality of **D** will be 1 that is the minimum rank with which **D** can be generated is 1. Furthermore, all the **G** matrices (i.e. gram matrix) with minimum rank will also be **F** matrix given by equation 2.9 [66, theorem 5], i.e.

$\mathbf{F} = \mathbf{G} = \mathbf{X}^T\mathbf{X}$, if **G** is a minimum rank matrix.

Summarizing, if we compute an incomplete matrix **D** or **F** given by equation 2.9, then resultant configuration matrix **X** will always be the one with a minimum rank or the points will be in a lowest possible dimensional space.

By using aforementioned expressions, localization problem can be formulated in terms of **F** matrix given in equation 2.9 as:

$$find\ \mathbf{F}$$

$$subject\ to\ \mathbf{F} \in \mathbf{C},$$

$$(where\ \mathbf{C}\ is\ a\ convex\ set\ assumed\ to\ contain\ positive\ semidefinite\ matrices)$$

$$\mathbf{F} \succcurlyeq 0 \qquad (2.11a)$$

$$A(\mathbf{D}) = \mathbf{b}$$

$$rank\ \mathbf{F} = d$$

where in equation 2.11a, instead of $\mathbf{F}=\mathbf{X}^T\mathbf{X}$, we have written an equivalent expression as rank **F**=*d* [8], whereas A(**D**) = **b** can have constraints as given in equation 2.10. If problem 2.11a can be solved and **F** with rank d is obtained, then by decomposing it or by taking a matrix square root of it [54], configuration matrix **X** with a minimum rank will be obtained.

As the rank constraint problem 2.11a is not a convex optimization problem, authors in [54-56] proposed an equivalent rank constraint convex problem by replacing rank constraint with trace minimization as:

$$minimize\ tr\ (\mathbf{F})$$

$$subject\ to\ \mathbf{F} \in C$$

$$A(\mathbf{D}) = \mathbf{b} \qquad (2.11b)$$

$$\mathbf{F} \succcurlyeq 0$$

Example 2.2 explains how a localization problem can be formulated using equation 2.11b.

Example 2.2

Consider a network in R^2 Euclidean space with three anchors located at $\mathbf{a_1}$= (x_1, y_1), $\mathbf{a_2}$= (x_2, y_2), and $\mathbf{a_3}$= (x_3, y_3) and a node at $\mathbf{x_4}$=(x_4, y_4) in a Cartesian coordinate system as shown in the figure 2.9. Let all the distances are measurable, then localization problem is formulated using SDP as:

$$minimize\ tr(\mathbf{F})$$

$subject\ to$

$$tr\left(\mathbf{F}\left(\mathbf{e_i}-\mathbf{e_j}\right)\left(\mathbf{e_i}+\mathbf{e_j}\right)^T\right)=d_{ij}^2,\ \forall(i,j)\in(I_{aa}\cup\ I_{an}\ \cup\ I_{nn})$$

$$tr\left(\mathbf{F}\mathbf{e_i}\mathbf{e_i^T}\right)=\|x_i\|_2^2,\ \forall\ \{i=1,2,\dots N_a\}$$

$$tr\left(\mathbf{F}\left(\mathbf{e_i}\mathbf{e_j^T}+\mathbf{e_j}\mathbf{e_i^T}\right)\frac{1}{2}\right)=x_i^T x_j,\ \forall(i,j)\in N_a$$

$$\mathbf{X}(1:2,2:4)=[\mathbf{a_1}\ \mathbf{a_2}\ \mathbf{a_3}]$$

$$\begin{bmatrix}\mathbf{I} & \mathbf{X}\\ \mathbf{X^T} & \mathbf{F}\end{bmatrix}\succcurlyeq\ 0$$

where for **F**≽0 an equivalent linear matrix inequality [8] is written as:

$$\begin{bmatrix}\mathbf{I} & \mathbf{X}\\ \mathbf{X^T} & \mathbf{F}\end{bmatrix}\succcurlyeq 0$$

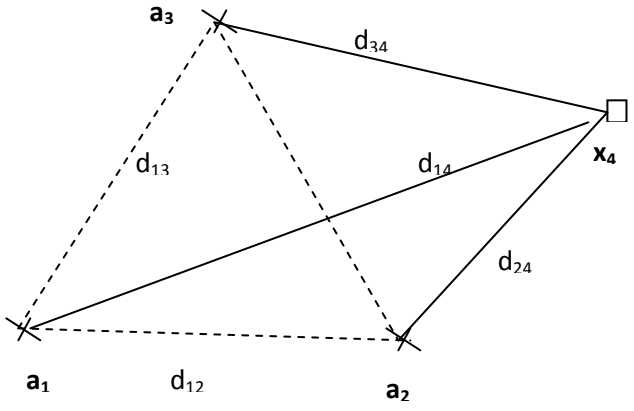


Figure2.9: Three anchors and a node in R2

Dattorro in [50] proposed an iterative form of SDP formulation for a rank-reduction as:

$$minimize \qquad tr(\mathbf{Z}, \mathbf{W}) \tag{2.12a}$$

$$Subject\ to\ tr\left(\widetilde{d_{ij}}\right)^2 \leq \left(\mathbf{F}\left(\mathbf{e_i}-\mathbf{e_j}\right)\left(\mathbf{e_i}-\mathbf{e_j}\right)^T\right) \leq \left(\widehat{d_{ij}}\right)^2,\ \forall (i,j) \in (I_{aa} \cup I_{an} \cup I_{nn})$$

$$tr\left(\mathbf{F}\mathbf{e_i}\mathbf{e_i^T}\right) = \|x_i\|^2, \quad i \in N_a$$

$$tr\left(\mathbf{F}\left(\mathbf{e_i}\mathbf{e_j^T}+\mathbf{e_j}\mathbf{e_i^T}\right)\frac{1}{2}\right) = x_i^T x_{j,} \quad i<j, \qquad \forall i,j \in N_a$$

$$\mathbf{X}(:,2:1+N_a) = [\mathbf{a_1} \ \cdots \ \mathbf{a_{Na}}]$$

$$\mathbf{Z} = \begin{bmatrix} \mathbf{I} & \mathbf{X} \\ \mathbf{X^T} & \mathbf{F} \end{bmatrix} \succcurlyeq 0$$

and

$$minimize\ \langle \mathbf{Z}, \mathbf{W} \rangle \tag{2.12b}$$

$$subject\ to \quad \mathbf{0} \preccurlyeq \mathbf{W} \preccurlyeq \mathbf{I}$$

$$tr\,\mathbf{W} = \mathbf{N}$$

where **W** in equation 2.12a can be initially selected as **W** = **I** or **W** = **O**, that is either identity or all zero matrix. $\check{d}$ and $\hat{d}$ are the lower and upper bounds on distance square of actual distances. We call this formulation hereafter as TSDP, the equations 2.12a and 2.12b are iteratively solved till solution with required rank is obtained.

2.6.2 Minimization of Distance Square Errors

Biswas [11,13,46, 57] formulated convex localization of a problem given in equation 2.3a-c by minimizing distance square errors. Let d_{ij} be the actual Euclidean distance between nodes *i* and *j*, and $\hat{d}_{ij}$ be the measured distance between two nodes. The relationship between square of actual distance and measurement is given by:

$$\left(d_{ij}\right)^2 = \left(\hat{d}_{ij}\right)^2 + \varepsilon_{ij} \quad (i,j) \in I_{an} \cup I_{nn}$$

where ε_{ij} is the square distance error between actual and measurements. Biswas and Ye [57] formulated localization problem by minimizing distance square error as:

$$minimize \sum_{(i,j)\in I_{an}\cup I_{nn}} \left|\varepsilon_{ij}\right|$$

subject to

$$tr\left(\mathbf{F}\left(\mathbf{e_i}-\mathbf{e_j}\right)\left(\mathbf{e_i}+\mathbf{e_j}\right)^T\right) - \varepsilon_{ij} = \hat{d}_{ij}^2, \forall (i,j) \in (I_{an} \cup I_{nn}) \qquad (2.13a)$$

$$\begin{bmatrix} \mathbf{I} & \mathbf{X} \\ \mathbf{X}^{\mathrm{T}} & \mathbf{F} \end{bmatrix} \succcurlyeq 0$$

Instead of minimizing sum of absolute distance square errors substitutions $\varepsilon_{ij} = \varepsilon_{ij}^{+} - \varepsilon_{ij}^{-}$ are made to deal with absolute values [17], with this SDP localization problem 2.13a becomes:

$$minimize \sum_{(i,j)\in I_{an}\cup I_{nn}} \left(\varepsilon_{ij}^{+} - \varepsilon_{ij}^{-}\right)$$

subject to

$$tr\left(\mathbf{F}(\mathbf{e_i} - \mathbf{e_j})(\mathbf{e_i} + \mathbf{e_j})^T\right) - \varepsilon_{ij}^{+} + \varepsilon_{ij}^{-} = \hat{d}_{ij}^2, \forall (i,j) \in (I_{an} \cup I_{nn}) \quad (2.13b)$$

$$\begin{bmatrix} \mathbf{I} & \mathbf{X} \\ \mathbf{X}^{\mathrm{T}} & \mathbf{F} \end{bmatrix} \succcurlyeq 0$$

We call hereafter heuristic given in equation 2.13b as DSDP.

2.7 Impact of Relaxation

By relaxing the original localization problem into SDP convex formulation, realization of unknown nodes can result in a higher dimensional solution (equally known as high rank solution) especially for noisy distance measurements and for a network being only globally rigid. This is a necessary condition for a unique realization in R^d, where d is the desired dimension of the problem. Due to relaxation, if there exists a higher dimensional realization of the problem, SDP formulation will find it. Therefore, global rigidity is not enough criteria for the relaxed SDP formulation guaranteeing a unique realization. Resultantly, more strict constraint known as unique d-localizable is required for the network. A *unique d-localizable* [23,13] is defined as:

Definition 2.15: *Unique d-localizable*: A network is uniquely *d*-localizable, if for a system of equations 2.3a-c has a unique solution and is also a unique solution for any space R^l, where l>*d*. Therefore, for SDP based localization, necessary condition for a unique solution is unique *d*-localizability. Under such conditions, unique solution for a relaxed SDP could be obtained. It has been found in [13] that trilateral graphs defined in 2.4 are unique-*d* localizable.

2.8 Signal Model

Performance of any optimization technique used depends on the channel conditions that induce noise in the received signal. The channel condition further depends on the building type (residential, office or manufacturing) and type of furniture within it. In order to design a sound indoor localization system and for its performance evaluation, a comprehensive channel model is vital.

It should be noted that the generic indoor channel models used to evaluate performance of indoor wireless systems are not suitable for localization problem performance evaluations as they do not fit the empirical models found specifically for localization [39].

Therefore, extensive work has been done in [58-61] to develop indoor channel models for localization applications based on TOA distance measurements.

In TOA based technique, the distance between a node and anchor is measured by estimating the signal propagation delay (τ) in free space, where radio signals travel at the constant speed of light (c). Therefore, distance (d) between anchors and the node is estimated as:

$$d = \tau c$$

Within the indoor environment, measurements are corrupted by four types of sources, that is due to multipaths, thermal noise, fully or partially obstructed signal. All these sources will introduce two types of noises: measurement noise and a positive bias.

Measurement noise is modelled as independent identically distributed (i.i.d) Gaussian with zero mean at all nodes represented as $\mathcal{N}\sim(0, \sigma^2_n)$. Biases in [39] are modelled as Gaussian in regions with a stronger signal normally considered into a coverage area and lognormal distributed elsewhere. In [47], it is modelled as positively distributed where probability distribution function was developed based on measurements, while in [63] positive biases are modelled as gamma distributed.

Based on aforementioned references and channel characterization found in [39], we considered in our analysis the general model for distance measurements ($\hat{d}_{ij}$) based on TOA between two nodes given as:

$$\hat{d}_{ij} = d_{ij} + b_{ij} + n,\ (i,j) \in N_a \cup N_n \qquad (2.14)$$

Where d_{ij} represents the actual Euclidean distance between two nodes, b_{ij} is the biased within the measurements and is a function of distance while n is the random measurement Gaussian noise at all nodes.

2.9 Summary

In this chapter we formulated the localization problem using theory of distance geometry, rigidity theory and EDM. We have also presented the review of indoor localization. The conditions guaranteeing a unique solution for a non convex problem (it must be globally rigid), with at least d+1 non-collinear anchors is also explained. We have also presented SDP relaxed formulation of the original non-convex problem. For a unique solution, network must be uniquely d-localizable. Trilateral graphs are the types of graphs with a property of unique d-localizability, a necessary condition for a unique realization in all dimensions. Furthermore, distance measurements in an indoor environment are always biased.

Chapter 3

Theoretical Bounds and Estimator Performance

Theoretical bounds are the theoretical limits indicating the best estimates achievable with all the noises present within the distance measurements used for position estimates. These bounds also serve as the benchmarks for comparing different estimation techniques against the best theoretical values attainable.

In this chapter, computation of bounds for a localization problem with biased measurements having prior Gaussian or gamma distributions is presented.

We have analyzed the performance of the SDP, and Least Square estimation techniques for localization and compared results with theoretical limits.

The pre-averaging case, where the distance measurements are pre-averaged prior to estimations is also analyzed. It has been found that pre-averaging the measurements prior to estimations can improve performance especially for stationary nodes.

3.1 Estimated Positions as a Random Variable

Distance measurements are never accurate. Additionally, measurements are also always biased in an indoor environment. Due to the presence of noisy measurements resultant estimated positions of the unknown nodes and their estimates are random variables.

These random positions could be completely classified by a probability density function which further depends on factors such as mathematical relations of positions (**X**) with measured distances, probability of noise and estimation technique used.

Although probability density function gives complete information regarding estimated values, it is most of the time sufficient if two parameters: mean of the estimated values and their variances/standard deviation can be used to judge the quality of the estimates and performance of the estimation technique used. Therefore, in subsequent sections we will analyze the performance of the estimation techniques based on these two parameters.

3.1.1 Desirable Estimates

Any optimum estimation technique used for localization is desired to have following two characteristics on any estimated value $\overline{\mathrm{X}}$:

1. Mean value of the estimate should be equal to actual values i.e. $E[\bar{\mathbf{x}}] = \mathbf{x}$, where **x** is actual node positions, and E[.]represents mean
2. Variance or standard deviation of the estimates should be minimum i.e. $\sigma_{\bar{x}}^2 = E[\overline{\mathrm{x}} - E[\mathrm{x}]^2]$ tends to zero.

Any estimator with mean of the estimates satisfying the condition (1) above is known as an *unbiased estimator*. Second condition implies that the average mean square deviations of the estimations from the true values should be small which further implies that the estimates are distributed closely around their mean values.

Conditions 1 and 2 are the desired characteristics of the estimates expected from any optimum estimation technique. Now two questions arise: whether an estimator exist satisfying conditions 1 and 2 for localizing nodes with biased distance measurements, secondly if an estimator does not exist with such properties, then what are the best estimates which could be achieved with measurements having all the noises present. The answer to second question is that if an estimator exists and is unbiased, then the best an estimator could estimate is lower bounded by Cramer Rao Lower (CRL) bounds or (CRLB).

On other words, if condition one is met then the lowest value for condition 2 achievable is lower bounded by CRL bounds.

For the case of biased estimator, its variances around the mean values cannot be compared with those of CRL bounds since variations are around mean of the estimates which are biased and not the actual node positions. For this case, the Circular Error Probable (CEP) (a performance measure metric used in target detection [67] and localization applications) can be used [34, 68, 69]. CEP is basically a confidence interval consisting of fifty percent of the estimates. Normally this interval is defined in terms of a circle centred at actual position of a node with radius encapsulating 50% of the estimated values and is used to quantify the accuracy of the estimator. The smaller the value of CEP (radius) the better is the estimated values and the estimator used.

We have computed theoretical limits i.e. CRLB and CEP using the distance measurement model given in equation 2.14. In the next step, we have analyzed the performance of the SDP and weighted LS estimators based on these performance metrics.

In our analysis, we have also considered pre-averaged measurements and found that pre-averaging can improve variance of the estimations. Pre-averaging could be applied to stationary nodes in order to achieve smaller variances.

3.2 CRL Bounds

CRLB inequality gives the theoretical lower bound for error variances of any *unbiased estimates* of some unknown parameters [38]. These lower bounds can then be used to find out the best achievable accuracy. CRL bounds for localization are investigated in [70-77].

Analysis is based on two types of biased distance measurements given in equation 2.14 for an indoor environment that is Gaussian and Gamma distributions. Let measurable distances between the number of anchors N_a, and a single node $\mathbf{x_0}$, be $d_{i,0}$. According to

signal model (equation 2.14), distance measurements between anchors and a node are given as:

$$d_{i,0} = \hat{d}_{i,0} + b_{i,0} + n, \quad i = 1, 2, \ldots, N_a, \tag{3.1}$$

where $\hat{d}_{i,0}$ denote actual Euclidean distance between i^{th} anchor and a node $\mathbf{x_0}$ in R^2; it is written as.

$$\hat{d}_{i,0} = \sqrt{(x_i - x_0)^2 + (y_i - y_0)^2} \tag{3.2}$$

We compute CRL bounds for a network in R^2 Euclidean space consisting of N_a anchors and an unknown node. If number of nodes is more than one, expressions derived here could be used iteratively to find the CRL bounds for each node. It is also possible to find the CRL bound expressions for a network consisting of N_a anchors and N_n >1 number of unknown nodes. The results obtained will be the same to a case where a network is partitioned into subnets consisting of N_a anchors and an unknown node and bounds found for each subnet. The subnet technique is adopted here due to simplicity.

3.3 CRLB Computations

To compute CRL bounds, we begin with biased distance measurements having prior Gaussian distributions. Therefore, we consider the position coordinates (x_0, y_0) of an unknown node and biases ($b_{i,0}$) within distance measurements to be unknown parameters. The resultant unknown parameters to be estimated consist of two sets i.e. N_a number of biases and (x_0, y_0) coordinates of an unknown node. These unknown parameters can be arranged into a vector (**u**) as.

$$\mathbf{u} = (x_0, y_0, b_{1,0}, b_{2,0}, \ldots, b_{Na,0}) \tag{3.3}$$

Let $\bar{\mathbf{u}}$ denote the estimates of the unknown parameters **u**. CRL bounds for these unknown parameters given in by the vector in equation 3.3 are computed as [38]:

$$Cov(\mathbf{u}) \geq \mathbf{J_T^{-1}} \tag{3.4}$$

where covariance (Cov ( ))is defined as:

$$Cov(\mathbf{u}) \equiv E[(\bar{\mathbf{u}} - \mathbf{u})(\bar{\mathbf{u}} - \mathbf{u})^T]$$

while $\mathbf{J}_T^{-1}$ in equation 3.4 is the Fisher Information matrix, and E[.] is the expectation. According to [78], $\mathbf{J_T}$ matrix for unknown parameters with prior information consists of two parts:

$$\mathbf{J_T} = \mathbf{J_D} + \mathbf{J_P} \tag{3.5}$$

where $\mathbf{J_D}$ in equation 3.5 is information due to data and $\mathbf{J_P}$ is the information due to prior knowledge. Furthermore, $\mathbf{J_D}$ for an unknown parameter **u** is given by:

$$\mathbf{J_D} = E\left[\left(\frac{\partial}{\partial \mathbf{u}} \log f_{\mathbf{u}}(d)\right)\left(\frac{\partial}{\partial \mathbf{u}} \log f_{\mathbf{u}}(d)\right)^T\right] \tag{3.6}$$

In equation 3.6, '*d*' is a measured distances between anchors and a node, $f_{\mathbf{u}}(d)$ is the joint probability density function of measurements '*d*' conditioned on **u**. For a Gaussian measurement noise (n ~$(0,\sigma_n^2)$) joint probability density function is given by

$$f_u(d) \equiv f\left(d|\, x, y, b_{1,0}, \ldots, b_{Na,0}\right) \propto \prod_{i=1}^{Na} \exp\left\{-\frac{1}{2\sigma_n}\left(d_{i,0} - \hat{d}_{i,0} - b_{i,0}\right)^2\right\} \tag{3.7}$$

Whereas elements of a $\mathbf{J}_P$ in equation 3.5 are given by [78]:

$$\mathbf{J_{Pij}} = -E\left[\frac{\partial^2 \ln p_u(u)}{\partial u_i \partial u_j}\right] \quad \forall\ (i,j) \in (1,2,\ldots,N_a+2) \tag{3.8}$$

where $P_u(u)$ in equation 3.8 is a priori probability distribution function of an unknown parameters.

3.3.1 Computation of J_D

With Gaussian measurement errors elements of $\mathbf{J_D}$ matrix (with (Na + 2) unknown parameters of **u**) can be computed as [38]:

$$[\mathbf{J_D}]_{i,j} = \frac{1}{\sigma_n^2}\sum_{k=1}^{Na}\frac{\partial d_{k,o}}{\partial \mathbf{u}_i}\frac{\partial d_{k,o}}{\partial \mathbf{u}_j},\ \ i = 1,2,\ldots,N_a+2, j = 1,2,\ldots,N_a+2 \tag{3.9}$$

After substituting equation 3.1 into 3.9 it is straightforward to obtain all the elements of $\mathbf{J_D}$ as:

$[\mathrm{J_D}]_{11} = \frac{1}{\sigma_n^2}\sum_{k=1}^{Na}(\cos\theta_k)^2$ where $\cos\theta_k = \frac{x_0 - x_k}{\sqrt{(x_0-x_k)^2+(y_0-y_k)^2}}$

$[\mathrm{J_D}]_{12} = \frac{1}{\sigma_n^2}\sum_{k=1}^{Na}\cos\theta_k \sin\theta_k$ where $\sin\theta_k = \frac{y_0 - y_k}{\sqrt{(x_0-x_k)^2+(y_0-y_k)^2}}$

$[\mathrm{J_D}]_{13} = \frac{1}{\sigma_n^2}\cos\theta_1$

$\vdots$

$[\mathrm{J_D}]_{1j} = \frac{1}{\sigma_n^2}\cos\theta_{j-2}$ j=3,4, …, N_a+2

$[\mathrm{J_D}]_{21} = [\mathrm{J_D}]_{12}$

$[\mathrm{J_D}]_{22} = \frac{1}{\sigma_n^2}\sum_{k=1}^{Na}(\sin\theta_k)^2$

$[\mathrm{J_D}]_{23} = \frac{1}{\sigma_n^2}\sin\theta_1$

$\vdots$

$[\mathrm{J_D}]_{1j} = \frac{1}{\sigma_n^2}\sin\theta_{j-2}$ j=3,4, …, N_a+2

$[\mathrm{J_D}]_{31} = [\mathrm{J_D}]_{13}$

$[\mathrm{J_D}]_{32} = [\mathrm{J_D}]_{23}$

$[\mathrm{J_D}]_{33} = \frac{1}{\sigma_n^2}$

$[\mathrm{J_D}]_{3j} = 0$ j=4,5, …, N_a+2

$[\mathrm{J_D}]_{41} = [\mathrm{J_D}]_{14}$

$[\mathrm{J_D}]_{ij} = \frac{1}{\sigma_n^2}\ \forall\ i = j = 4,5,\dots,Na+2$

$[\mathrm{J_D}]_{ij} = 0\ \forall\ i \neq j = 4,5,\dots,Na+2$

All these terms of $\mathbf{J_D}$ can be arranged into a (Na+2)×(Na+2) matrix form as:

$$J_D = \left[\begin{array}{cc:cccc} \frac{1}{\sigma_n^2}\sum_{k=1}^{Na}(\cos\phi_k)^2 & \frac{1}{\sigma_n^2}\sum_{k=1}^{Na}(\cos\phi_k \sin\phi_k) & \frac{1}{\sigma_n^2}\cos\phi_1 & \frac{1}{\sigma_n^2}\cos\phi_2 & \cdots & \frac{1}{\sigma_n^2}\cos\phi_{Na} \\ \frac{1}{\sigma_n^2}\sum_{n=1}^{M-1}(\cos\phi_n \sin\phi_n) & \frac{1}{\sigma_n^2}\sum_{n=1}^{M-1}(\sin\phi_n)^2 & \frac{1}{\sigma_n^2}\sin\phi_1 & \frac{1}{\sigma_n^2}\sin\phi_2 & \cdots & \frac{1}{\sigma_n^2}\sin\phi_{Na} \\ \hdashline \frac{1}{\sigma_n^2}\cos\phi_1 & \frac{1}{\sigma_n^2}\sin\phi_1 & \frac{1}{\sigma_n^2} & 0 & 0 & 0 \cdots \\ \frac{1}{\sigma_n^2}\cos\phi_2 & \frac{1}{\sigma_n^2}\sin\phi_2 & 0 & \frac{1}{\sigma_n^2} & 0 & 0 \cdots \\ \frac{1}{\sigma_n^2}\cos\phi_3 & \frac{1}{\sigma_n^2}\sin\phi_3 & 0 & 0 & \frac{1}{\sigma_n^2} & 0 \cdots \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \cdots \\ \frac{1}{\sigma_n^2}\cos\phi_{Na} & \frac{1}{\sigma_n^2}\sin\phi_{Na} & 0 & 0 & 0 & \frac{1}{\sigma_n^2} \end{array}\right]$$

Or equally, $\mathbf{J_D}$ matrix could be partitioned according to dotted lines and written into blocks matrices form as:

$$\mathbf{J_D} = \frac{1}{\sigma_n^2}\begin{bmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{B^T} & \mathbf{I} \end{bmatrix} \tag{3.10}$$

where **I** is the identity matrix, whereas **A** and **B** are given by:

$$A = \begin{bmatrix} \frac{1}{\sigma_n^2}\sum_{k=1}^{Na}(\cos\phi_k)^2 & \frac{1}{\sigma_n^2}\sum_{k=1}^{Na}(\cos\phi_k \sin\phi_k) \\ \frac{1}{\sigma_n^2}\sum_{k=1}^{Na}(\cos\phi_k \sin\phi_k) & \frac{1}{\sigma_n^2}\sum_{k=1}^{Na}(\sin\phi_k)^2 \end{bmatrix} \tag{3.11a}$$

$$B=\begin{bmatrix} \frac{1}{\sigma_n^2}\cos\phi_1 & \frac{1}{\sigma_n^2}\cos\phi_2 & \cdots & \frac{1}{\sigma_n^2}\cos\phi_{Na} \\ \frac{1}{\sigma_n^2}\sin\phi_1 & \frac{1}{\sigma_n^2}\sin\phi_2 & \cdots & \frac{1}{\sigma_n^2}\sin\phi_{Na} \end{bmatrix} \qquad (3.11b)$$

3.3.2 Computation of J_P

Considering biases according to [39] as Gaussian with non-zero mean $\mathcal{N}\sim(\mu_{b,i0}, (\sigma_{bi0})^2)$, the prior probability of **u** i.e. $P_{\mathbf{u}}(\mathbf{u})$ is given by:

$$p_u(u) \propto \prod_{i=3}^{Na+3} \exp -\frac{1}{2\sigma_{b,i0}}\left(u_i-\mu_{b,i0}\right)^2 \qquad (3.12)$$

By substituting equation 3.12 into equation 3.8, elements of **Jp** are obtained as:

$$J_P=\left[\begin{array}{cc:ccc:c} 0 & 0 & 0 & 0 & 0 & \cdots \\ 0 & 0 & 0 & 0 & 0 & \cdots \\ \hdashline 0 & 0 & \frac{1}{\sigma_{b,10}^2} & 0 & 0 & \cdots \\ 0 & 0 & 0 & \frac{1}{\sigma_{b,20}^2} & 0 & \cdots \\ 0 & 0 & 0 & 0 & \frac{1}{\sigma_{b,30}^2} & \cdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \cdots \end{array}\right]$$

The dimensions of $\mathbf{J_P}$ matrix will be equal to that of $\mathbf{J_D}$; $(N_a+2)\times(N_a+2)$. Likewise, $\mathbf{J_D}$ we can also write $\mathbf{J_P}$ in block matrix form (shown by dotted lines) as:

$$\mathbf{J_P} = \begin{bmatrix} \mathbf{O_1} & \mathbf{O_2} \\ \mathbf{O_2^T} & \frac{1}{\sigma_{b,i0}^2}\mathbf{I} \end{bmatrix} \tag{3.13}$$

Where $\mathbf{O_1}$ and $\mathbf{O_2}$ are all zero block matrices. After combining $\mathbf{J_D}$ and $\mathbf{J_P}$ given in equations 3.10 and 3.13, $\mathbf{J_T}$ is expressed as:

$$\mathbf{J_T} = \mathbf{J_D} + \mathbf{J_P} = \frac{1}{\sigma_n^2}\begin{bmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{B^T} & \mathbf{I} \end{bmatrix} + \begin{bmatrix} \mathbf{O_1} & \mathbf{O_2} \\ \mathbf{O_2^T} & \frac{1}{\sigma_{b,i0}^2}\mathbf{I} \end{bmatrix} = \frac{1}{\sigma_{b,i0}^2}\begin{bmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{B^T} & \mathbf{C_1} \end{bmatrix} \tag{3.14}$$

where

$$\mathbf{C_1} = \mathbf{I} + \frac{\sigma_n^2}{\sigma_{b,i0}^2}$$

Finally, CRL bounds according to equation 3.4 are obtained by evaluating the inverse of $\mathbf{J_T}$ found in equation 3.14. Let $(\mathbf{J_T})^{-1}$ be given as:

$$J_T^{-1} = \begin{bmatrix} \mathbf{D_N} & \mathbf{E} \\ \mathbf{F} & \mathbf{G} \end{bmatrix}$$

where $\mathbf{D_N}$ is:

$$\mathbf{D_N} = \sigma_n^2\left[\mathbf{A} - \mathbf{B}\mathbf{C_1^{-1}}\mathbf{B^T}\right]^{-1} \tag{3.15}$$

The subscript '*N*' with **D** in equation 3.15 is used to represent CRL bounds with biases considered as Gaussian. The first two diagonal elements of $\mathbf{D_N}$ will be the location bounds for an unknown node as

$$[D_N]_{ii}, i = 1,2$$

In terms of real location (x_0 ,y_0) of an unknown node and its estimates $(\bar{x}_0, \bar{y}_0)$, variances in *x* and *y* coordinates are:

$$\bar{\sigma}_{x0}^2 = E[x_0 - \bar{x}_0]^2 \geq [\mathbf{J}_T^{-1}]_{11} \quad (3.16a)$$

and

$$\bar{\sigma}_{y0}^2 = E[y_0 - \bar{y}_0]^2 \geq [\mathbf{J}_T^{-1}]_{22} \quad (3.16b)$$

While mean squared Euclidean distance between the true location $\mathbf{x_0}$= (x_0, y_0) and its estimated location is given by:

$$E[u - \bar{u}]^2 = E[x_0 - \bar{x}_0]^2 + E[y_0 - \bar{y}_0]^2 \quad (3.17)$$

3.3.3 CRL Bounds for Gamma Distributed Biases

From equation 3.5, CRL bounds consists of $\mathbf{J_D}$ and $\mathbf{J_P}$, where $\mathbf{J_P}$ only depends on prior information. If prior information of biases changes, only $\mathbf{J_P}$ matrix will be adjusted accordingly. The biases in [73] for indoor environments are found to be gamma distributed G(α, β) with probability density function:

$$P(b_{i0};\ \alpha_{i0}, \beta_{i0}) = \frac{b_{i0}^{\alpha_{i0}-1}}{\beta_{i0}^{\alpha_{i0}}\Gamma(\alpha_{i0})} \quad i = 1, 2, 3, \ldots, Na \quad (3.18)$$

where $\Gamma(\alpha)$ is the gamma function. To evaluate CRL bounds, only prior information regarding biases has been changed from Gaussian distributions to Gamma, therefore $\mathbf{J_P}$ is re-evaluated using 3.18 and 3.8 as:

$$\mathbf{J_p} = \begin{bmatrix} \mathbf{O_1} & \mathbf{O_2} \\ \mathbf{O_2^T} & \frac{1}{(\alpha_{i0}-2)\beta_{i0}^2}\mathbf{I} \end{bmatrix} \tag{3.19}$$

By combing $\mathbf{J_P}$ from equation 3.19 with $\mathbf{J_D}$ given in equation 3.10, $\mathbf{J_T}$ is obtained as:

$$\mathbf{J_T} = \frac{1}{\sigma_n^2}\begin{bmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{B^T} & \mathbf{C_2} \end{bmatrix} \tag{3.20}$$

where

$$\mathbf{C_2} = (\mathbf{1} + \frac{\sigma_n^2}{(\alpha_{i0}-2)\beta_{i0}^2})\mathbf{I}$$

while **A** and **B** are given in equations 3.11 a, b.

Following similar steps in deriving equation 3.15, lower bounds for position estimates by considering biases as Gamma distributed are obtained by using equation 3.20.Resultantly position bounds will be the first two diagonal elements of $\mathbf{D_G}$ given by:

$$[\mathbf{D_G}]_{ii} = \sigma_n^2\left[\mathbf{A} - \mathbf{B}\mathbf{C_2^{-1}}\mathbf{B^T}\right]^{-1} \quad i = 1,2 \tag{3.21}$$

where subscript G with **D** in equation 3.21 is used to represent Gamma distribution.

By observing CRL bound expressions given in equation 3.15 and 3.21, it is deduced that bounds only depend on variances of the measurement noise and biases, while mean of the biases has no effects on the bounds.

As an example, we consider a case with three anchors located at vertices of an equilateral triangle with sides 10 m at (0, 0), (10, 0) and (5, 8.66) respectively. While an unknown node could move freely between them. The contour of square root of the Euclidean distance (equation 3.17) is given in figure 3.1, where measurement variance is 4e-4m and biases are considered to be Gaussian distributed and function of distance with variance at 1m as 7.84e-4 from [39].

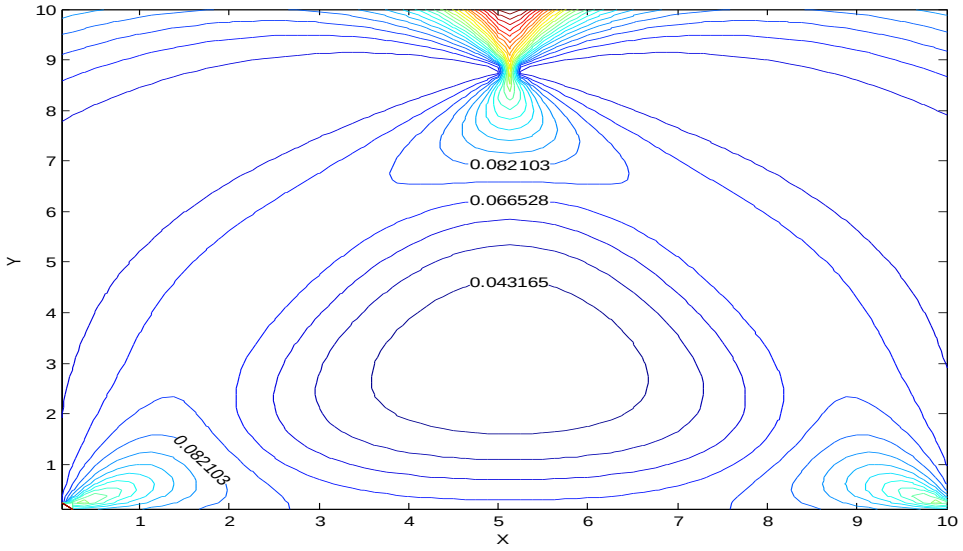


Figure3.10: Contour of square root of the Euclidean distance for the case where known sensors are placed at the vertices of an equatorial triangle with sides 10 m.

Similarly, we also considered a case with six anchors located at the vertices of a regular hexagon with side length 10m at (-5, -8.66), (5, -8.66), (10, 0), (5, 8.66), (-5, 8.66) and (-10, 0), respectively. The variance of measurements and biases is same to that of

aforementioned three anchors example. It is found that centre of the hexagon is not the optimum minimum error location area as shown in figure 3.2.

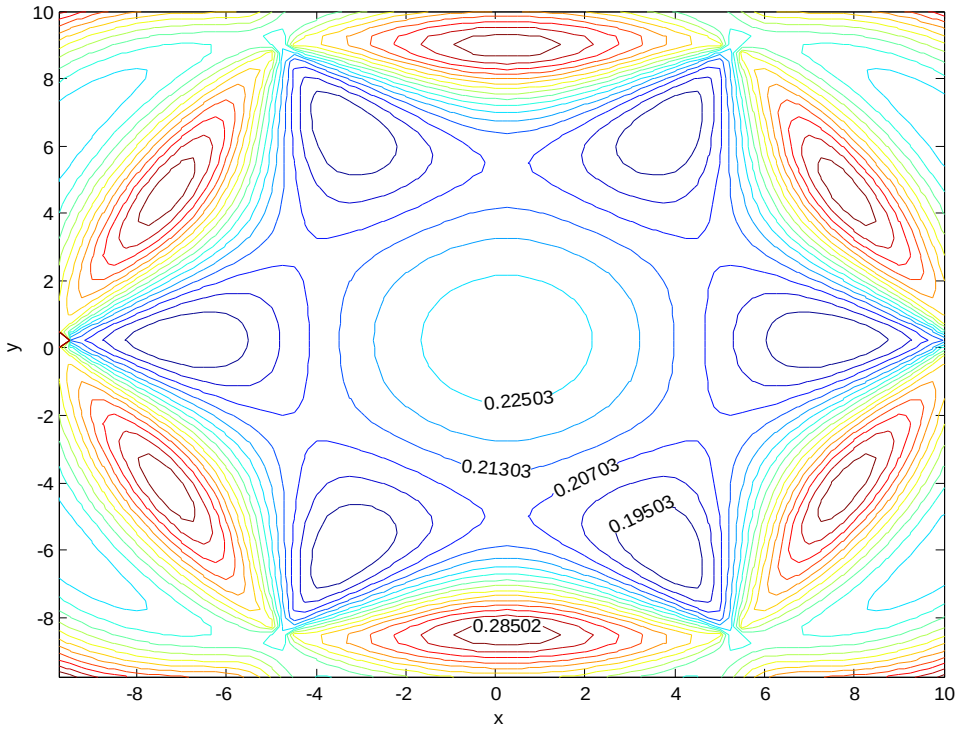


Figure 3.2: Contour of square root of the Euclidean distance for the case where 6 known sensors are placed at the vertices of a regular hexagon with side length 10m.

3.3.4CRL Bounds for Averaged Measurements

If WLAN is not densely populated and nodes are static then by pre-averaging the measurements, performance of the localization can be improved. In this section, we have found the theoretical CRL bounds for the case where distance measurements are pre-averaged prior to applying localization algorithm.

The pre- averaged distance measurements of a signal model given in equation 2.11 are:

$$\frac{1}{M}\sum_{k=1}^{M} d_{ij,k} = \hat{d}_{ij} + \frac{1}{M}\sum_{k=1}^{M} b_{ij,k} + \frac{1}{M}\sum_{k=1}^{M} n_k \;\; \forall\, i,j \in \{I_{aa}, I_{an}, I_{nn}\} \tag{3.22}$$

where *M* is the number of measurements averaged. Furthermore, equation 3.22 can be written into a simplified form as:

$$d_{ij,M} = d_{ij} + b_{ij,M} + n_M \;\; \forall\, i,j \in \{I_{aa}, I_{an}, I_{nn}\} \tag{3.23}$$

where if measurement noise n_k is Gaussian $n \sim N(0, \sigma_n^2)$,then n_M in equation 3.23 will be:

$$n_M \sim N(0, {}^{\sigma_n^2}/_{M})$$

If each bias $b_{ij,k}$ for any k^{th} measurement is normally distributed i.e. $\mathcal{N}\sim(\mu_b, \sigma_b^2)$, then $b_{ij,M}$ will also be normally distributed with $\mathcal{N}\sim(\mu_b, \sigma_b^2/M)$ [79]. Lower bounds for pre-average measurements given by equation 3.22, and 3.23 are found by replacing variances in $\mathbf{J_D}$ and $\mathbf{J_P}$ given in equations 3.10 and 3.13 respectively with new variances of averaged biases and averaged measurement noises. Following the steps in deriving the expressions for CRL bounds in previous section, it can be easily proved that pre-average measurements are 1/*M* times of a single measurement $\mathbf{D_N}$ as:

$$[\mathbf{D_{AN}}] = \frac{1}{M}[\mathbf{D_N}] \tag{3.24}$$

If biases are considered gamma distributed as in section 3.3.3 and *M*>>1, then using central limit theorem (CLT), averaged gamma biases could be approximated by Gaussian distribution $\mathcal{N}\sim(\mu_i, \alpha_i\beta_i^2/M)$. *M*>30 is considered by most of the statisticians as sufficient number for any averaged distribution to be approximated as normal [79].

Approximating gamma distribution to be normal (for M>30) and following the similar steps as in the aforementioned section, $\mathbf{J_D}$ and $\mathbf{J_P}$ are written for gamma distribution as:

$$\mathbf{J_D} = \frac{M}{\sigma_n^2}\begin{bmatrix}\mathbf{A} & \mathbf{B}\\ \mathbf{B^T} & \mathbf{I}\end{bmatrix} \tag{3.25}$$

$$\mathbf{J_p} = \begin{bmatrix}\mathbf{O_1} & \mathbf{O_2}\\ \mathbf{O_2^T} & \frac{1}{(\alpha_{io})\beta_{i0}^2}\mathbf{I}\end{bmatrix} \tag{3.26}$$

Combining equation 3.25- 3.26, $\mathbf{J_T}$ is given by:

$$\mathbf{J_T} = \frac{M}{\sigma_n^2}\begin{bmatrix}\mathbf{A} & \mathbf{B}\\ \mathbf{B^T} & \mathbf{C_3}\end{bmatrix} \tag{3.27}$$

where

$$\mathbf{C_3} = \mathbf{I} + \frac{\sigma_n^2}{\alpha_{io}\beta_{io}^2}\mathbf{I} \tag{3.28}$$

Following the equation 3.15, the CRL bounds for pre-averaged measurements with gamma biases from equation 3.27 will be the first two diagonal elements of $\mathbf{D_{AG}}$ given as:

$$[\mathbf{D_{AG}}]_{ii} = \frac{\sigma_n^2}{M}\left[\mathbf{A} - \mathbf{B}\mathbf{C_3^{-1}}\mathbf{B^T}\right]^{-1} \quad i = 1,2 \tag{3.29}$$

with $\mathbf{C_3}$ given in equation 3.28.

3.4 Accuracy of the Estimators

CRL bounds only give theoretical limits. To quantify the accuracy of the estimator, circular error probable (CEP) is computed, which is basically a location precision. It is defined to be the radius (R_c) of the smallest circle with centre at actual position of the node which has a 50% probability of containing the estimated locations [34]. Thus for 50% of the time, the estimates of the unknown node will lie within the circle drawn around the actual position.

To derive mathematical expressions for CEP=R_C: radius of a circle (we will be using CEP and R_C interchangeably to mean R_C within this thesis), let us consider a single random variable *x* with Gaussian probability density function (pdf) *f(x)*. Figure 3.3 depicts two cases of *x* one with zero mean and other with a non zero mean value.

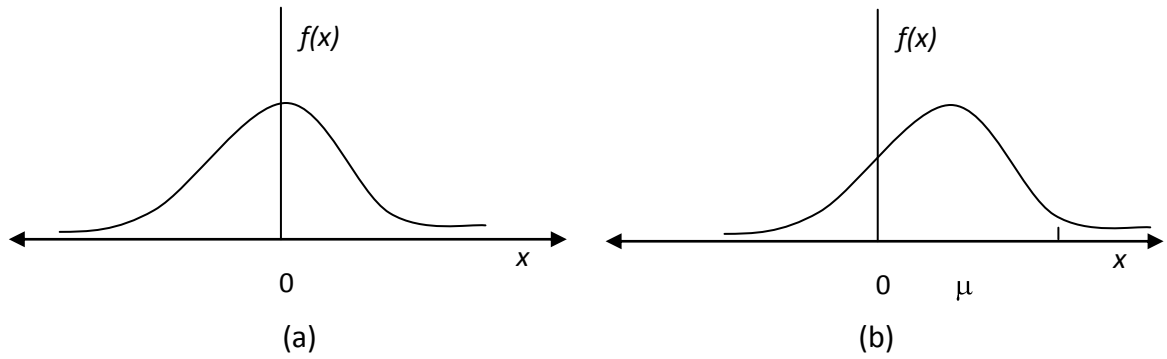


Figure 3.3: (a) Gaussian pdf with zero mean, (b) Gaussian pdf with μ mean

If there are two Gaussian random variables (*x*, *y*) with mean (μ_x, μ_y) and variances$\left(\sigma_x^2, \sigma_y^2\right)$, then their joint probability density function is given by [88]:

$$f_{xy}(x,y) = \frac{1}{\sqrt{2\pi}\sigma_x} e^{-\frac{1}{2\sigma_x}(x-\mu_x)^2} \quad \frac{1}{\sqrt{2\pi}\sigma_y} e^{-\frac{1}{2\sigma_y}(y-\mu_y)^2} \tag{3.30}$$

Figure of equation 3.30 will be a 3-dimensional (3D) bell shape. If variances in *x* and *y* are equal, plot of equal probability will be a circle and if $\sigma_x \neq \sigma_y$, equal probability results in an ellipse. Figure 3.4 depicts various cases.

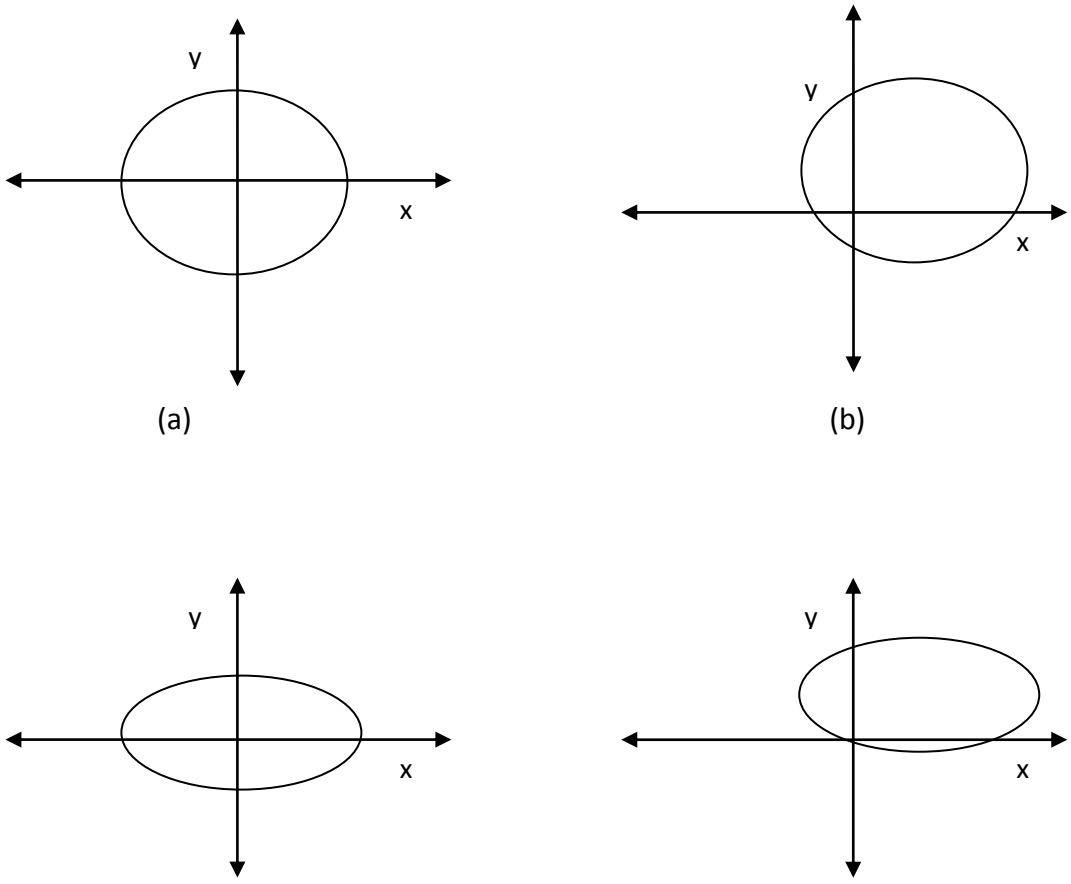


Figure 3.4: (a) equal probability case when $\sigma x = \sigma y$ and mean of both variables is zero (b) equal probability case when $\sigma x = \sigma y$ and mean of both variables is not zero, (c) equal probability case when $\sigma x \neq \sigma y$ and mean of both variables is zero (d) equal probability case when $\sigma x \neq \sigma y$ and mean of both variables is non zero.

Let $\mathbf{x_1} = (x_1, y_1)$ be the actual position of a node, and $\bar{\mathbf{x}}_1$ be its estimate as shown in figure 3.5, then R_c will be the radius of a circle centred around $\mathbf{x_1}$. For the unbiased position estimations, centre of *Rc* circle and circles of equal probability values coincide as shown in figure 3.5 a, while figure 3.5b shows a case of biased estimates when centres of both circles do not coincide.

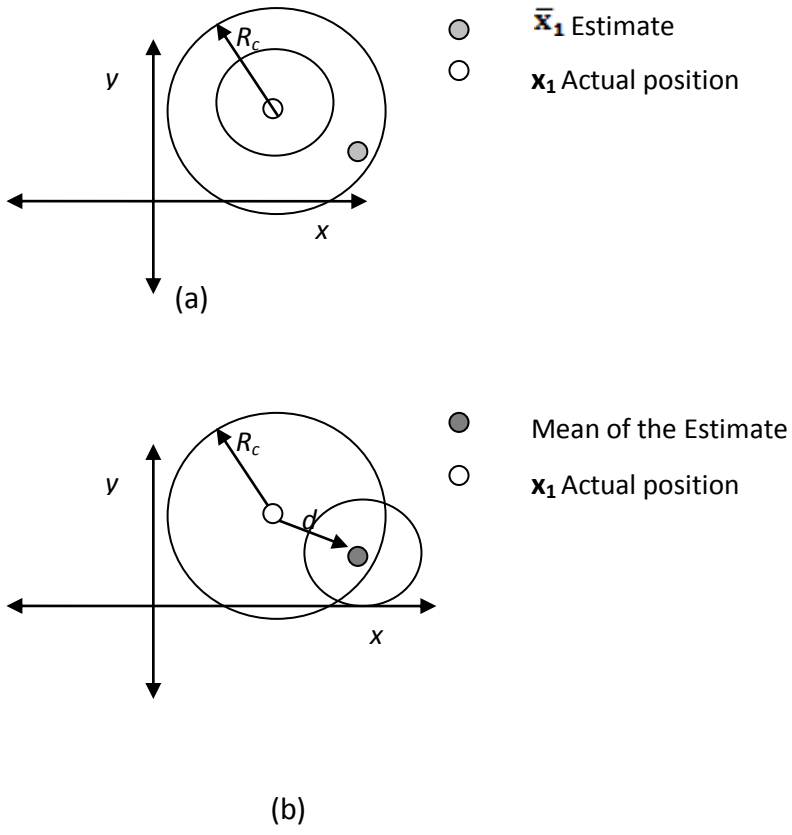


Figure 3.5: (a)Rc circle around actual position for unbiased estimate (b) Rc circle around actual position for biased estimate

The CEP, a circle with a radius containing 50 % estimates will be the area within curves (i.e. circle or ellipse) as shown in figures 3.4 and 3.5. For simplicity and without any loss, we assume unknown node to be present at an origin, then R_C is obtained by solving the integral given in equation 3.31 [67,80, 88].

$$F_{Rc} = \iint_{R_c} f_{xy}(x,y)dxdy \ = 0.5, \ \ with\ R_c \geq \sqrt{(x)^2 + (y)^2} \tag{3.31}$$

where F_{Rc} is a cumulative distribution function. Setting F_{Rc} = 0.5 for 50% probability and substituting 3.30 into 3.31, we get:

$$0.5 = \iint_{R_c} \frac{1}{2\pi\sigma_x\sigma_y} e^{-\frac{1}{2}\left(\frac{(x-\mu_{\bar{x}})^2}{\sigma_x} + \frac{(y-\mu_{\bar{y}})^2}{\sigma_y}\right)} dxdy \tag{3.32}$$

where $(\mu_{\bar{x}},\ \mu_{\bar{y}})$ are the mean of estimate in *x* and *y* coordinates.

R_C is found by solving equation 3.32. If biases are not zero then it is not possible to obtain a closed form solution [80]. If $\sigma_x = \sigma_y = \sigma$, then equation 3.32 will be the area of a circle having radius R_C and can be numerically approximated by [80] as:

$$F_{R_c} = e^{-\frac{d^2}{2\sigma^2}} \sum_{n=0}^{\infty} \frac{\left(\frac{d^2}{2\sigma^2}\right)^n}{n!} \left[1 - e^{-\frac{R_c^2}{2\sigma^2}} \sum_{m=0}^{n} \frac{\left(\frac{R_c^2}{2\sigma^2}\right)^m}{m!}\right] \tag{3.33}$$

where

$$d = \sqrt{(x-\mu_{\bar{x}})^2 + \left(y-\mu_{\bar{y}}\right)^2}$$

Not all the time variances are equal and resultant equal probable shape is an ellipse instead of a circle as shown in figure 3.4c-d. In [80], authors replaced an ellipse by an equivalent circle to simplify calculations and suggested that:

$$\sigma = 0.5\left(\sigma_x + \sigma_y\right) \; for \; 0.2 \leq \frac{\min\left(\sigma_x, \sigma_y\right)}{\max\left(\sigma_x, \sigma_y\right)} \leq 1 \tag{3.34}$$

This will produce a standard circle which will effectively replace the standard error ellipse, that is by setting value of σ as given by equation 3.34 and solving equation 3.33, R_c value can be found for unequal variances.

In [34] CEP is approximated more simply as:

$$CEP = \; d + 0.75\sqrt{\sigma_x^2 + \sigma_y^2} \tag{3.35}$$

We have used this approximation to generate initial value within the algorithm to solve equation 3.31 numerically for fast convergence.

3.5 Least Square (LS) Estimation

LS estimator has been used for localization problems as it does not require probabilistic assumptions about data. Only signal model is sufficient to formulate the problem. In this section, we analyze LS estimations for the indoor signal model given in equation 2.14 by finding mean and variance of the estimated values.

For analysis, we assume a trilateral WLAN in R^2, with N_a number of anchors and for simplicity we consider one unknown node, although LS estimations presented here could easily be extended for $N_n > 1$.

Solution to a set of quadratic non-linear equations 2.3 a-c, with a distance measurement model given in equation 2.14, can be simplified by approximating them with a linear set of

equations. Linearity is obtained by expanding quadratic equations 2.3 a-c using Taylor series expansion and retaining terms below second order. For this purpose, let the Euclidean distance between an anchor located at $\mathbf{x}_i$ in R^2 and an unknown node located at $\mathbf{x}$, be denoted by $\hat{d}_i(\mathbf{x_i}, \mathbf{x})$ is written as:

$$\hat{d}_i(\mathbf{x_i}, \mathbf{x}) = \sqrt{(x_i - x)^2 + (y_i - y)^2}$$

Its Taylor series approximation at the point $\mathbf{x_0}$ will be:

$$\hat{d}_i(\mathbf{x_i}, \mathbf{x}) = \hat{d}_i(\mathbf{x_i}, \mathbf{x_0}) + \left.\frac{(x_i - x)}{\sqrt{(x_i - x)^2 + (y_i - y)^2}}\right|_{\mathbf{x}=\mathbf{x}_0} (x - x_0) + \left.\frac{(y_i - y)}{\sqrt{(x_i - x)^2 + (y_i - y)^2}}\right|_{\mathbf{x}=\mathbf{x}_0} (y - y_0)$$

or equally be written in simplified form as:

$$\hat{d}_i = \varepsilon_i + h_{x,i}x + h_{y,i}y \tag{3.36}$$

where,

$$\varepsilon_i = d_i(\mathbf{x_i}, \mathbf{x_0}) - \left.\frac{(x_i - x)}{\sqrt{(x_i - x)^2 + (y_i - y)^2}}\right|_{\mathbf{x}=\mathbf{x}_0} x_0 - \left.\frac{(y_i - y)}{\sqrt{(x_i - x)^2 + (y_i - y)^2}}\right|_{\mathbf{x}=\mathbf{x}_0} y_0$$

and

$$h_{x,i} = \left.\frac{(x_i - x)}{\sqrt{(x_i - x)^2 + (y_i - y)^2}}\right|_{\mathbf{x}=\mathbf{x}_0} \qquad h_{y,i} = \left.\frac{(y_i - y)}{\sqrt{(x_i - x)^2 + (y_i - y)^2}}\right|_{\mathbf{x}=\mathbf{x}_0}$$

After substituting approximate equation 3.36 into equation 2.14, we get the following linear equation in terms of unknown coordinates (*x*, *y*) as:

$$d_i = \varepsilon_i + h_{x,i}x + h_{y,i}y + b_i + n_i \quad \forall\, i \in N_a \tag{3.37}$$

For a set of N_a number of linear equations 3.37, there are N_a + 2 unknowns, that is N_a number of biases and two position coordinates of an unknown node in R^d. Therefore resultant problem is underdetermined. One way to overcome this is to merge biases with a measurement noise considering biases as noises unlike we treated biases as unknowns to compute CRL bounds. For Gaussian distributed biases and merging with measurement noise, equation 3.37 results in:

$$d_i = \varepsilon_i + h_{x,i}x + h_{y,i}y + \mu_i + n_{bi} \quad \forall\, i \in N_a \tag{3.38}$$

where n_{bi} is $N \sim (0, \sigma^2 = \sigma_{bi}^2 + \sigma_n^2)$

further combining terms in equation 3.38 results in:

$$d_i = s_i + h_{x,i}x + h_{y,i}y + n_{bi} \quad \forall\, i \in N_a \tag{3.39}$$

where s_i in equation 3.39 is given by:

$$s_i = \varepsilon_i + \mu_i$$

The set of resultant equations 3.39 can also be written into vectors as:

$$\mathbf{d} = \mathbf{s} + \mathbf{Hx} + \mathbf{n} \tag{3.40}$$

where matrix **H** is

$$\mathbf{H} = \begin{bmatrix} h_{x,1} & h_{y,1} \\ \vdots & \vdots \\ h_{x,Na} & h_{y,Na} \end{bmatrix} \text{ and a noise vector } \mathbf{n} = \begin{bmatrix} n_{b1} \\ \vdots \\ n_{bNa} \end{bmatrix}$$

LS estimator minimizes the square difference between the observed measurements given in equation 3.40 and the assumed signal model, mathematically written as:

$$J(\mathbf{x}) = (\mathbf{d} - (\mathbf{Hx} + \mathbf{s}))^T(\mathbf{d} - (\mathbf{Hx} + \mathbf{s})) \tag{3.41}$$

LS estimator is obtained by setting the gradient of the cost factor given in equation 3.41 equal to zero. Resultant LS estimator giving estimates of an unknown node location coordinates is given by [38]:

$$\bar{\mathbf{x}} = (\mathbf{H^T H})^{-1}\mathbf{H^T}(\mathbf{d} - \mathbf{s}) \tag{3.42}$$

An improvement in LS could be made if statistical characteristics of the measurements are known. The resultant LS is known as weighted LS and is obtained by minimizing the cost factor:

$$J(\mathbf{X}) = (\mathbf{d} - (\mathbf{Hx} + \mathbf{s}))^T\mathbf{W}(\mathbf{d} - (\mathbf{Hx} + \mathbf{s})) \tag{3.43}$$

where **W** can be selected as the variance of noises. The resultant weighted LS estimator is obtained as:

$$\bar{\mathbf{x}} = (\mathbf{H}^{\mathrm{T}}\mathbf{C}^{-1}\mathbf{H})^{-1}\mathbf{H}^{\mathrm{T}}\mathbf{C}^{-1}(\mathbf{d} - \mathbf{s}) \tag{3.44}$$

where **C** in equation 3.44 is a covariance matrix if noises are uncorrelated, it is given as:

$$\mathbf{C} = \sigma_{ii}\mathbf{I}, \qquad \text{with } \sigma_{ii}^2 = \sigma_{bi}^2 + \sigma_n^2$$

Substituting **s** in equation 3.44 and rearranging terms results in:

$$\bar{\mathbf{x}} = (\mathbf{H}^{\mathrm{T}}\mathbf{C}^{-1}\mathbf{H})^{-1}\mathbf{H}^{\mathrm{T}}\mathbf{C}^{-1}(\mathbf{d} - \boldsymbol{\varepsilon}) - (\mathbf{H}^{\mathrm{T}}\mathbf{C}^{-1}\mathbf{H})^{-1}\mathbf{H}^{\mathrm{T}}\mathbf{C}^{-1}\boldsymbol{\mu} \tag{3.45}$$

If mean of biases are known, the estimator will account for biases by the last term in equation 3.45. However, it has been observed that mean values are a function of indoor environment and distance between two nodes. Thus for a generic system used in diverse environments, it is difficult to assign a fixed value to biases. Therefore, we assume mean values of the biases to be unknown and equate them to zero values. The resultant estimator with unknown mean will then be:

$$\bar{\mathbf{x}} = (\mathbf{H}^{\mathrm{T}}\mathbf{C}^{-1}\mathbf{H})^{-1}\mathbf{H}^{\mathrm{T}}\mathbf{C}^{-1}(\mathbf{d} - \boldsymbol{\varepsilon}) \tag{3.46}$$

For **n** to be normal, the distributed $\bar{\mathbf{x}}$ in equation 3.46 will also be normal with covariance given as:

$$\mathbf{C}_{\bar{\mathbf{x}}} = (\mathbf{H}^{\mathbf{T}}\mathbf{C}^{-\mathbf{1}}\mathbf{H})^{-1} \tag{3.47}$$

While expected values and bias of the estimates given in equation 3.46 are:

$$E[\bar{\mathbf{x}}] = (\mathbf{H}^{\mathbf{T}}\mathbf{C}^{-\mathbf{1}}\mathbf{H})^{-\mathbf{1}}\mathbf{H}^{\mathbf{T}}\mathbf{C}^{-\mathbf{1}}\boldsymbol{\mu}_{\mathbf{b}} + \mathbf{x} \tag{3.48a}$$

$$\mathbf{b}_{\bar{\mathbf{x}}} = (\mathbf{H}^{\mathbf{T}}\mathbf{C}^{-\mathbf{1}}\mathbf{H})^{-\mathbf{1}}\mathbf{H}^{\mathbf{T}}\mathbf{C}^{-\mathbf{1}}\boldsymbol{\mu}_{\boldsymbol{b}} \tag{3.48b}$$

For the case of Gaussian measurement noise and Gaussian biases LS estimator given by equation 3.45 will also be Gaussian with probability density function [88, section 4.11.7]:

$$f_{\bar{\mathbf{x}}}(\bar{\mathbf{x}}) = \frac{1}{(2\pi)^{\frac{1}{2}}|\mathbf{C}_{\bar{\mathbf{x}}}|^{\frac{1}{2}}} e^{-\frac{1}{2}(\bar{\mathbf{x}}-\mathbf{m})^{T}\mathbf{C}_{\bar{\mathbf{x}}}^{-\mathbf{1}}(\bar{\mathbf{x}}-\mathbf{m})^{T}} \tag{3.49}$$

where $\mathbf{C}_{\bar{\mathbf{x}}}$ is given in equation 3.47 and $\mathbf{m} = \mathbf{x} + \mathbf{b}_{\mathbf{x}}$, with $\mathbf{b}_{\mathbf{x}}$ given in equation 3.48b.

In the absence of biases, estimator will be unbiased with a mean value equal to actual positions and its variance will satisfy the CRL bounds. While in the presence of biases with or without prior knowledge, variances of the LS estimate is given in equation 3.47. Although these variances are comparable with CRL bounds but one should note that these variances are not around actual positions. They are around mean value given by equation 3.48a which are further biased. Thus the covariance of the estimates given in equation 3.47 are not the variances around actual positions.

Furthermore, for conditions where biases either not known or are assumed to zero, have no effect on variances. However, it does affect the accuracy which can be measured using CEP.

3.5.1 Error Analysis WLS

The mean of the WLS estimates for biased distance measurements are not the actual positions, therefore, WLS for biased measurements will not be an unbiased estimator. Due to this, variances, even if they are comparable with CRL bounds, are not the variances around actual positions and thus do not represent real errors. Alternatively, CEP metric will be used for performance measure.

3.6 SDP Formulation for Localization Analysis

In this section, we analyze the SDP optimization technique for localization given in equation 2.13 based on a mean and variances of the estimated values.

For simplification, we consider a trilateral WLAN in R^2 with three non collinear anchors ($\mathbf{a_1}$, $\mathbf{a_2}$, and $\mathbf{a_3}$) and an unknown node at $\mathbf{x_0}$. For the case where distance measurements are corrupted with zero mean measurement noise, mean and variance of the SDP estimate is found in [57]. Here we have extended the given results for a biased indoor distance measurements as given in equation 2.14.

We rewrite a distance measurement model equation 2.14 for this purpose as:

$$d_{i,0} = \hat{d}_{i,0} + b_{i,0} + n\,, i = 1,2,3 \tag{3.50}$$

where measurement noise and biases are assumed to be Gaussian distributed such as:

$n \sim N(0, \sigma_n^2)\ and\ b_{i,0} \sim N(\mu_{i,0}, \sigma_{bi}^2)$

By merging biases with the measurement noise equation 3.50 can be written as:

$$d_{i,0} = \hat{d}_{i,0} + \mu_{i,0} + n_i\ , i = 1,2,3 \tag{3.51}$$

where

$$n_i \sim N(0, \sigma_i^2 = \sigma_n^2 + \sigma_{bi}^2)$$

The SDP optimization function given in equation 2.13 will be minimized by solving following system of linear equations:

$$\overline{\mathbf{G}} - 2\bar{\mathbf{x}}_0^T \mathbf{a_1} + \|\mathbf{a_1}\|_2 = \left(d_{10} + \mu_{1,0}\right)^2 + n_1^2 \tag{3.52a}$$

$$\overline{\mathbf{G}} - 2\bar{\mathbf{x}}_0^T \mathbf{a_2} + \|\mathbf{a_2}\|_2 = \left(d_{20} + \mu_{2,0}\right)^2 + n_2^2 \tag{3.52b}$$

$$\overline{\mathbf{G}} - 2\bar{\mathbf{x}}_0^T \mathbf{a_3} + \|\mathbf{a_3}\|_2 = \left(d_{30} + \mu_{3,0}\right)^2 + n_3^2 \tag{3.52c}$$

where **G** as before is a Gram matrix. To find the mean of the estimates, we take expectations of both sides of equations 3.52a-c and get the following results:

$$E[\overline{\mathbf{G}}] - 2E[\bar{x}_0]^{\mathrm{T}}\, \mathbf{a_1} + \|\mathbf{a_1}\|_2 = \left(d_{10} + \mu_{1,0}\right)^2 + E[n_1^2] \tag{3.53a}$$

$$E[\overline{\mathbf{G}}] - 2E\left[\bar{x}_0\right]^{\mathrm{T}}\, \mathbf{a_2} + \|\mathbf{a_2}\|_2 = \left(d_{20} + \mu_{2,0}\right)^2 + E\left[n_2^2\right] \tag{3.53b}$$

$$E[\overline{\mathbf{G}}] - 2E[\bar{x}_0]^{\mathrm{T}}\, \mathbf{a_3} + \|\mathbf{a_3}\|_2 = \left(d_{30} + \mu_{3,0}\right)^2 + E[n_3^2] \tag{3.53c}$$

By solving system of linear equations 3.53a-c, mean of the estimated positions in *x* and *y* coordinates is obtained as:

$$E[\bar{x}_0] = \frac{1}{2}\left[\frac{(v_1-v_2)(y_3-y_1)-(v_1-v_3)(y_2-y_1)}{(x_2-x_1)(y_3-y_1)-(x_3-x_1)(y_2-y_1)}\right] \quad (3.54a)$$

$$E[\bar{y}_0] = \frac{1}{2}\left[\frac{(v_1-v_2)(x_3-x_1)-(v_1-v_3)(x_2-x_1)}{(y_2-y_1)(x_3-x_1)-(y_3-y_1)(x_2-x_1)}\right] \quad (3.54b)$$

where

$$v_i = \left(d_{i0} + \mu_{i,0}\right)^2 + E[n_i^2] - \|\mathbf{a_i}\|_2 \quad (3.54c)$$

If measurements were unbiased, then $\mu_{i,0}$ in equation 3.54c will be zero and mean of the positions obtained from equations 3.54a,b would be the actual positions. The biases within position estimates can be obtained as:

$$b_{x0} = \bar{x}_0 - x_0 = \frac{1}{2}\left[\frac{(v_1-v_2)(y_3-y_1)-(v_1-v_3)(y_2-y_1)}{(x_2-x_1)(y_3-y_1)-(x_3-x_1)(y_2-y_1)}\right] - \frac{1}{2}\left[\frac{(v_1'-v_2')(y_3-y_1)-(v_1'-v_3')(y_2-y_1)}{(x_2-x_1)(y_3-y_1)-(x_3-x_1)(y_2-y_1)}\right] \quad (3.55a)$$

$$b_{y0} = \bar{y}_0 - y_0 = \frac{1}{2}\left[\frac{(v_1-v_2)(x_3-x_1)-(v_1-v_3)(x_2-x_1)}{(y_2-y_1)(x_3-x_1)-(y_3-y_1)(x_2-x_1)}\right] - \frac{1}{2}\left[\frac{(v_1'-v_2')(x_3-x_1)-(v_1'-v_3')(x_2-x_1)}{(y_2-y_1)(x_3-x_1)-(y_3-y_1)(x_2-x_1)}\right] \quad (3.55b)$$

where v_i' is as given in equation 3.54c with $\mu_{i,0}$ equal to zero. Since SDP estimator is biased, therefore, variance of the estimates will be around the biased estimated values and not around the actual positions. Therefore, we cannot compare variance of the

estimates to that of CRL bounds, instead we will compute CEP to quantify the accuracy of the SDP estimations.

3.6.1 Error Analysis SDP

The SDP estimations due to biased distance measurements will be biased, with biases within a position coordinates given in equations 3.55a-b. Therefore, CEP will be used to quantify the performance of the estimation technique.

3.7 CEP Computations for SDP and WLS

We have computed CEP for two SDP formulations given in equations 2.12, 2.13 and WLS optimization technique given in equation 3.46 and compared results with theoretical CEP computations using equation 3.33.

For simulations, we considered an arbitrary network with N_a anchors and Nn=30 unknown nodes. Distance measurement model given in equation 2.14 is considered with measurement noise assumed to be Normal distributed with zero mean while biases are considered as Normal distributed with mean and variance function of distances.

Figure 3.7 depicts the simulation results of R_C for WLS, TSDP, DSDP and theoretically computed R_C using equation 3.33 (we have represented theoretical computed R_C by a 'CEP' legend in all figures). Results have been obtained by assuming a network consisting of three anchors placed arbitrary at the vertices of the equilateral triangle with 30 unknown nodes placed on two linear lines as shown in figure 3.6. Within simulation, biases are assumed to be Gaussian function of distance with mean (0.058 at 1 meter) and variance function of distance (0.028 at 1 meter) from [39].

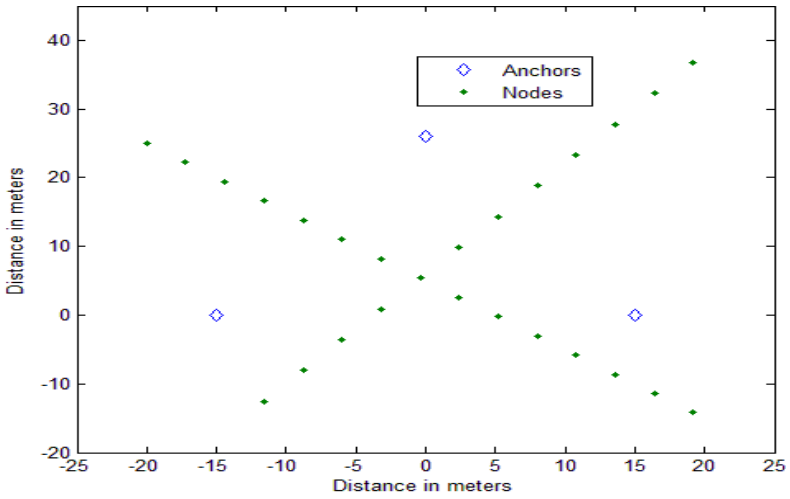


Figure 3.6: Network in R^2 consisting of 3 anchors, and 30 nodes

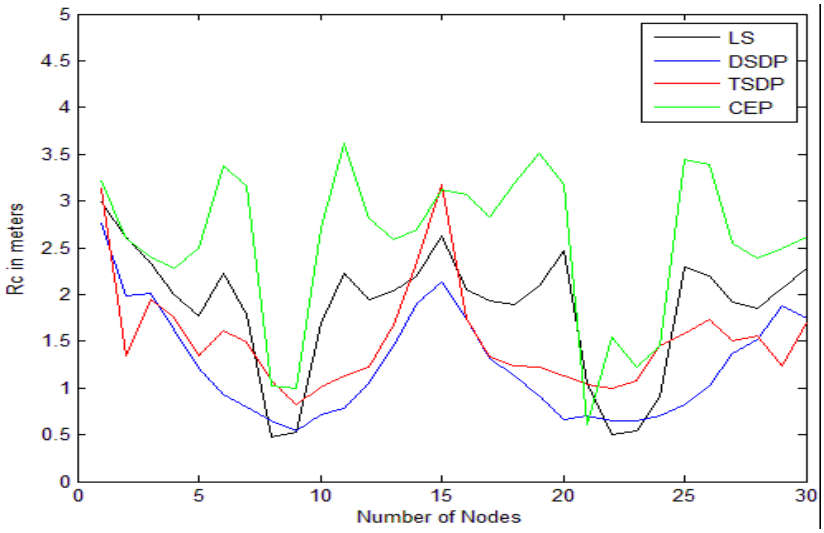


Figure 3.7: Rc for the network given in figure 3.6, CEP green curve is obtained by solving equation 3.33

Simulations were also carried out to obtain R_C for 40 unknown nodes placed randomly as shown in figure 3.8, CEP obtained is depicted in figure 3.9.

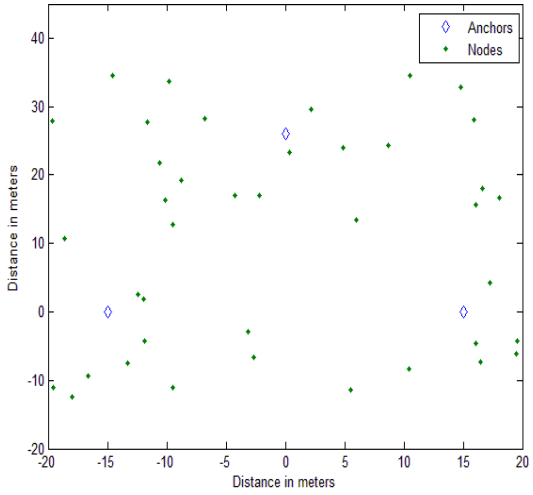


Figure 3.8: Network with 40 randomly distributed nodes.

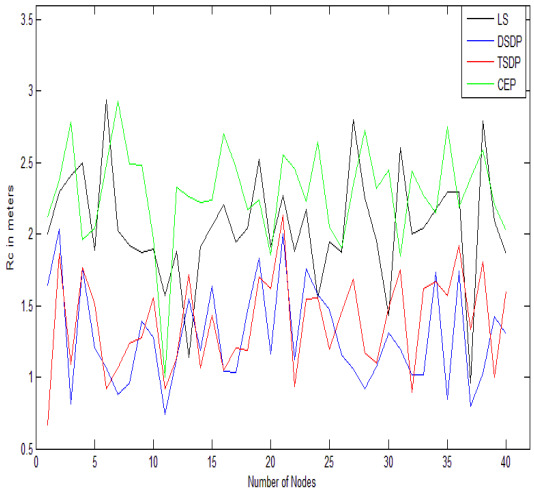


Figure 3.9: Rc for the network shown in figure 3.8.

Similarly CEP was computed for six anchors placed arbitrary at the vertices of a regular hexagon. In simulations, again 40 unknown nodes were considered placed randomly and on two linear lines as shown in figure 3.10 and figure 3.12. Simulation results for the two networks are given in figure 3.11 and 3.13.

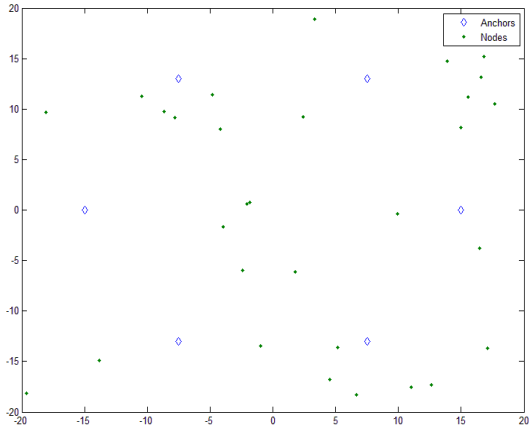


Figure 3.10: Network with 40 randomly distributed nodes and 6 anchors.

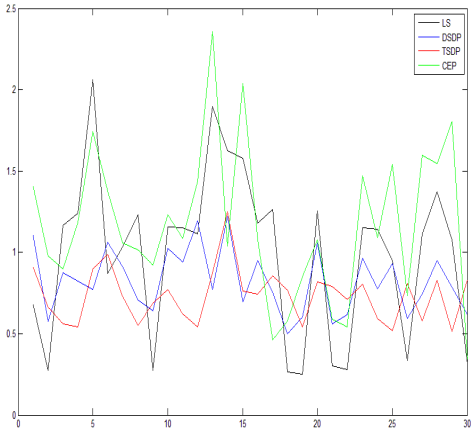


Figure 3.11: Rc for the network shown in figure 3.10.

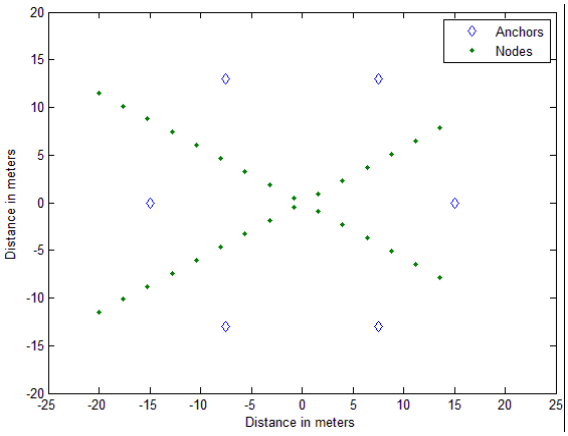


Figure 3.12: Network with 40 randomly distributed nodes and 6 anchors

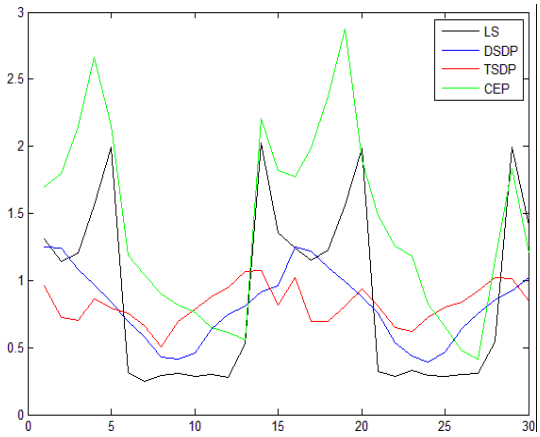


Figure 3.13: Rc for the network shown in figure 3.12.

From simulations and observation of the computed *Rc,* it can be seen that CEP for WLS match with that obtained by solving the equation 3.31. While the CEP of SDP approach is

less than that of WLS. Moreover, theoretical *Rc* values are the upper bounds for all estimators.

3.8 The Effects of SDP Relaxations on Localization

The original localization problem based on distances measurements is a non convex problem, it is converted into a relaxed convex SDP formulations given in equations 2.12 and 2.13.

Due to the presence of noises within distance measurements, it is quite possible that distance constraints contradict each other resulting in no realization in R^d**;** this results in $\mathbf{G} \neq \mathbf{X}^T\mathbf{X}$. However, in SDP formulation, this equality constraint is relaxed into $\mathbf{G} \succcurlyeq \mathbf{X}^T\mathbf{X}$, therefore, it is possible for the SDP optimization to find a realization in higher space $\mathbf{X} \in R^l$ (resulting in high rank solution) than desired space R^d (i.e. $l > d$) by making an objective value zero. For the case of TSDP, it uses an iterative technique given in equations 2.12a, b for a low rank realizations.

Figure 3.14 shows an example of the effects of a high rank solution for ideal distances that is with no errors for a DSDP optimizations. The method to force the solution into a low rank for DSDP has been discussed in [16]; where two solutions have been discussed. One is to place anchors such that all of the unknown nodes are enclosed within the parameter of the anchors. It might be possible for sensor networks but not for WLAN where we assumed no additional hardware available for localization; access points already used for traffic can be used for localization service as well. In this case access points (anchors) positions are optimized for throughput and coverage. The second method is to add regularization to the cost function. Results have demonstrated that using the regularization technique performance is sound, but disadvantage of this method is the difficulty in selecting the value of the regularization parameter, which further depends on size, geometry and availability of the distance information. Apart from this, the technique

of bounding away constraint is also discussed in [16] to achieve a low rank solution. With this technique, the number of constraints and computation complexity increases.

We have found that spreading out anchors at the boundary of the indoor environment; all the unknown nodes are within the parameter, is not feasible for WLAN as discussed earlier. If however, the virtual nodes are placed at the centre and near the boundary (i.e. outer walls of the building), a solution in lower dimension can be achieved. Virtual nodes are not the real nodes, we just place them to stretch out the problem. With addition of just one virtual node within the centre of the network shown in figure 3.14, a low dimension solution is obtained as shown in figure 3.15.

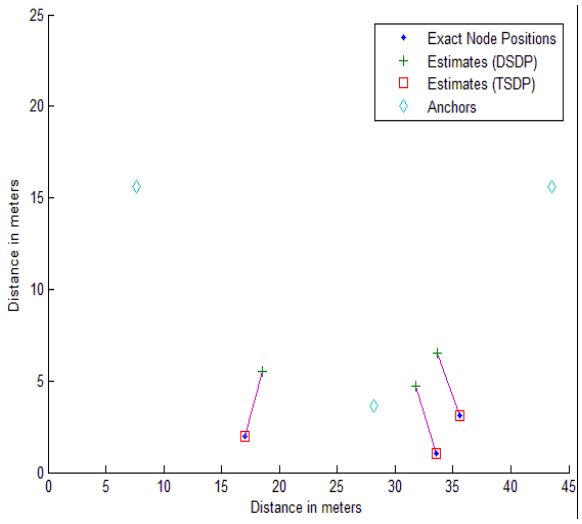


Figure 3.14: Effect of relaxations on DSDP localization

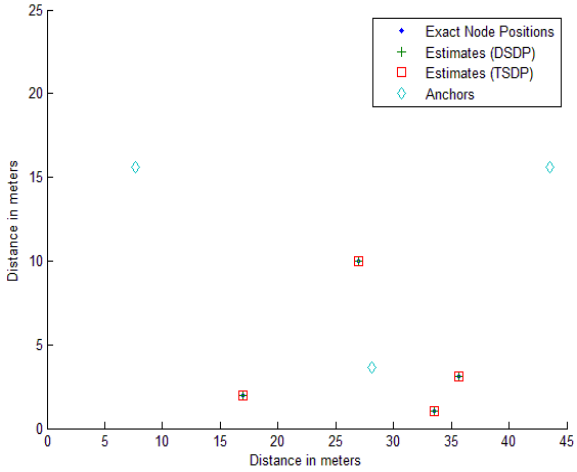


Figure 3.15: Localization of a network of figure3.14 after adding virtual nodes.

Similarly for the case where large number of nodes are present within the network and outside the parameter of the anchors are shown in figure 3.16. The effect of adding virtual nodes (encircled) encapsulating anchors and unknown nodes is shown in figure 3.17. From the figure 3.17, an improvement in estimations has been observed. TSDP is based on iterative rank minimization technique, due to iterations it is computationally expensive compare to DSDP. Therefore, one has to compromise between computational complexity and accuracy.

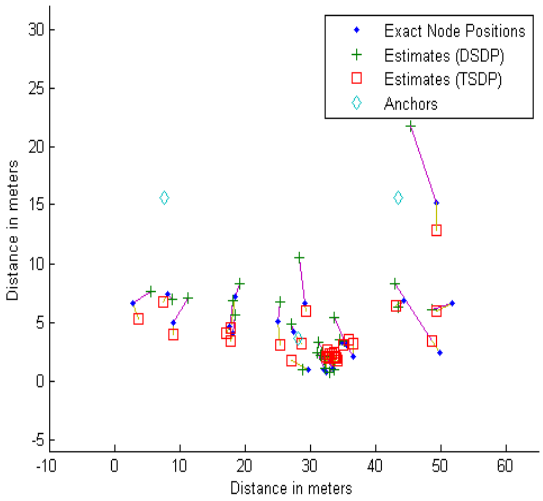


Figure 3.16: Localization of nodes without adding virtual node

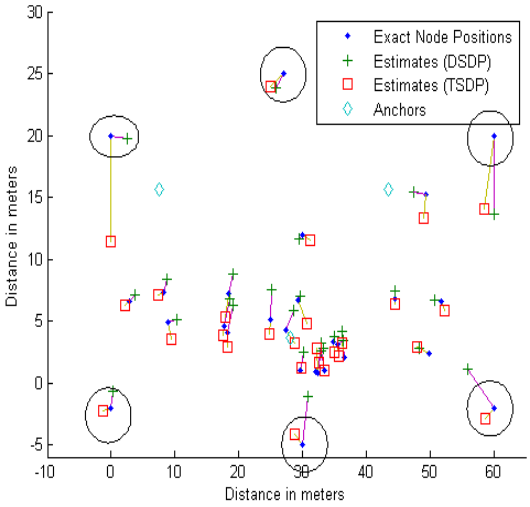


Figure 3.17: Localization of nodes with virtual nodes indicated by circles

3.9 Analysis and Discussions

In this chapter, the localization performance evaluation metrics have been explained which are used exclusively for biased estimators. We have found that CRL bounds cannot be used as benchmarks for performance evaluation. Therefore, CEP metric, which is confidence interval containing 50% estimates, is used for performance evaluation. Simulations are used to evaluate performance of the least square and two SDP based convex optimization techniques. Results show convex based techniques outperform the least square technique. In addition, we have also found that accuracy of estimates depends on various factors such as:

- *number of anchors* as shown in figures 3.7 and 3.13 respectively where R_C values of six anchors is lower than three anchors
- *on location of a node,* such as nodes located around the centre of an area have minimum errors with anchors at the boundaries (as shown in figures 3.12 and 3.13, where nodes 6 to 13 and nodes 21 to 27 are located around the centre have lower R_C values)
- on quality of *distance measurements*.

The high rank problem arising due to relaxation is solved by developing the concept of virtual nodes. Results show that low rank estimations can be obtained by adding virtual nodes.

3.10 Summary

In this chapter, we have computed theoretical bounds and then analyzed two estimation techniques that is WLS and SDP for localizing nodes in WLAN. We have found that for the biased indoor distance measurements, both estimators will be biased. Therefore, their variations around mean value could not be compared to that of the CRL bounds. Instead, CEP (a performance measure metric) is used to analyze the performance of the

estimators. We have also found that SDP estimators outperform the LS estimations based on the CEP metric. Furthermore, due to the SDP relaxation, a higher rank solution could be obtained. For DSDP, we have found that with the placement of virtual nodes at outer boundary a low rank solution can be obtained.

After evaluating the performance of the convex SDP based localization and its comparison with non-convex technique, a new convex based distributed localization technique to achieve highly accurate real time localization has been developed and is presented in next chapter.

Chapter 4

Distributive Localization Technique and Performance Evaluation

In this chapter a novel distributive WLAN localization algorithm based on SDP developed during the research work is described. Although original problem relaxed to the convex SDP form can produce highly accurate results as demonstrated in chapter 3, there are two problems associated with relaxed SDP optimization:

i. If a network is not universally rigid then a high rank solution with large estimation errors can be generated.

ii. SDP based optimizations are computationally demanding.

The proposed heuristic divides a WLAN consisting of large number of nodes into subnets consisting of small number of nodes as SDP based optimizations for small number of nodes is computationally efficient. Subnets are created based on geographical positions and mobility levels of the nodes, and are further classified as mobile and stationary subnets. Mobile subnets include all those nodes whose positions are continuously changing with time. On the other hand, stationary subnets consist of nodes with fixed positions (the status of these nodes can change at any time). Furthermore, mobile subnets are more frequently localized as compared to stationary nodes. Overall localization process becomes computationally efficient as not all the nodes present within the WLAN need to be localized all the time.

In addition, developed distributive localization technique is analyzed for its performance based on scalability, capacity, accuracy and frequency with which nodes can be localized.

Various parameters' selection like number of nodes within a subnet, threshold value for classifying nodes as mobile and stationary. Finally a case study by considering WLAN localization is included.

4.1 WLAN Node Distributions

To develop a novel algorithm for WLAN localization, distribution of nodes and their dynamics need to be taken into account. In a typical WLAN, number of active nodes is variable as it can vary from no-node to hundreds of nodes. As an example, consider a University campus, where during weekends and/or at night. It is highly probable that there is no one in the campus apart from few security personnel who may be using WLAN devices to communicate with the central office, and central office is able to locate security personnel roaming within the building using locations based application. The positions of security personnel can be estimated if the device used by the security personnel can directly measure distances between at least (d+1) anchors in R^d Euclidean space. Perhaps within the same campus during office timings there may be more than 100 WLAN device users. Therefore WLAN node distributions are highly diverse. Furthermore, not all the WLAN device users are on the move, some of them might be stationary for some time before changing their status.

The main traffic flow among the nodes within a WLAN provides broadband services. These services include typical real time applications such as video conferencing, VOIP video streaming etc and are computationally intensive.

It has been found that SDP optimizations for number of nodes greater than 50 cannot be solved efficiently due to computational complexity [8]. Thus within a network which is already providing broadband services, this additional computationally demanding location

service can overload the network. Therefore computation-efficient heuristic is needed for the WLAN network providing multimedia and localization service.

If a network is densely populated with uniformly distributed nodes then it is not a necessary condition that each node measures distances with at least (d+1) anchors to create a trilateral network., If node-node measurements are measurable between closely located nodes such that a resultant network becomes universally rigid, then nodes can be uniquely localized without requiring larger anchor range. Such techniques for densely populated networks are discussed in [83].

What we have seen from aforementioned discussion is that the number of nodes in WLANs could vary a lot. For such a network, all nodes must be able to measure distances between at least (d+1) access points (anchors) because in WLANs, dense node distribution is not always guaranteed. If closely located nodes exist, we assume that they can measure distances among themselves; with this additional measurements a better accuracy could be achieved.

4.2 Assumptions

While designing a novel localization heuristic, it has been assumed that the WLAN network is planned and optimized for capacity and coverage where each access point (with known positions) has a coverage range of Ra to provide broadband services other than the localization services. For localization services there exist at least (d+1) access points in R^d with known positions (anchors). Nodes within the coverage area of a network can measure distances with respect to d anchors. It is also assumed that all the nodes periodically measure distances between (d+1) anchors, thus for distance measurements only node to anchor range (Rr) is extended. Therefore, for distance measurements, only anchor-node range is extended periodically beyond Ra for a short duration. Node-node distances are also measurable if nodes are close to each other, we assume this range to be 3 meters (based on an average room length)., We have found using simulations that a

better accuracy can be achieved with these additional measurements. These nodes are known hereafter as neighbour nodes. If a large number of nodes is within the neighbour of a certain node then only two such node-node distance measurements will be considered and additional distance measurements will be sparsified.

To create the trilateral network, only those nodes will be localized having at least (d+1) anchor-node distance measurements, otherwise node(s) will not be localized.

We also assume localization algorithm runs at one of the anchors for estimating nodes positions; these estimated positions are then accessible to a location based application as and when required. We define a subnet as a subset of nodes present within a WLAN.

4.3 Distributed WLAN Localization Heuristic

An algorithm to localize nodes within a WLAN starts by comparing total number of nodes present within a network to a certain threshold number (N_T), selection of this number is described in detail in subsequent sections. For a network with nodes greater than N_T, it is divided into subnets based on geographical positions, i.e. by using node-node distances subnets are formed. If a node can measure a distance between a neighbouring node(s), then it is near to it and all such cluster of nodes are included within a subnet. The maximum number of nodes a subnet can have is N_T. Each subnet is sparsified such that anchor-node distances are always kept for localization and if for a certain node, more than two neighbouring nodes exist then only two arbitrary node-node measurements are kept. For the case where the number of nodes is greater than N_T and no neighbouring nodes exist, then subnets are formed by combining the nodes arbitrarily.

When the algorithm starts for the first time, each subnet is created based on the geographical positions and is then localized iteratively in time division multiplexed form. After this, by using new distance measurements, subnets created in previous iteration are localized. The position estimations obtained are compared with the previous estimations. If

for a certain node, the Euclidean distances between two estimates is greater than some threshold value say d_{Δ}, we assume it to be mobile otherwise it is marked stationary (see section 4.9 for determining d_{Δ}). All mobile nodes added within subnet(s) (if number of nodes are greater than N_T then the number of mobile subnets could be more than one) are marked as mobile.

The subnets are localized in a time division multiplexed (TDM) form, that is time slots are created such that within a slot subnet could be localized. All the subnets are scheduled within these time slots in a TDM way and are explained in section 4.12. The frequency (f_m) with which mobile subnet is localized depends on factors such as: total number of nodes within a subnet, number of subnets, technique used for localization and on processing capabilities of the processor. We have explained frequency determination with examples in subsequent sections.

Furthermore, after iteration each node (even stationary node) is checked for its state by comparing its new estimated position with previous estimate. If a mobile node becomes stationary or vice versa, it is placed into an appropriate subnet. With this technique, only mobile subnets that are time critical are updated frequently, therefore, not all the nodes within the network need to be localized frequently. Resultantly a localization algorithm is computationally optimized. Only nodes detected as mobile are segregated from the stationary nodes and are treated differently by localizing them more frequently compared to stationary nodes.

If total number of nodes within a WLAN is less than N_T, we do not create separate mobile and stationary subnets. The whole network is localized iteratively by treating it as mobile, this is done to reduce overheads. Due to possible change in status of the nodes, number of nodes within subnets will change. if there exist two subnets of identical type with total number of nodes less than $N_T/2$, then these subnets will be merged., This is done to reduce creation of large number of subnets with only few nodes. For a stable network, merge operation can be performed with very low frequency.

Main steps of the algorithm are:

Algorithm:

Input: A network with total number of nodes $N=N_a+N_n$, anchors with their respective positions and measured distances i.e. anchor-node, node-node.

Output: $\overline{\mathbf{X}}$ i.e. estimates of the unknown node positions.

Step 1: Create trilateral network by screening out nodes with less than (d+1) anchor-node distances in R^d.

Step 2: Check if $N < N_T$, then sparsify and localize the network, by treating it as mobile.

Else: Create subnets based on geographical positions, sparsify subnets, merge smaller subnets and then localize all subnets using SDP.

Step 3: Get new distance measurements and perform steps 1 and 2 (excluding creation of subnets in step 2).

Step 4: Find Euclidean distance between two successive position estimations for each node found in steps 2 and 3.

Step 5: If $N > N_T$, and Euclidean distance between two estimates is greater than d_Δ add it into a new subnet marked as mobile.

Step 6: If new node enters the network, check it for a neighbouring node, if so, then add it in a mobile subnet which contains its neighbour. If a subnet containing neighbour is not mobile, then add it in a mobile subnet with least number of nodes.

Step 7: Schedule subnets for localization (see section 4.12)

Step 8: Localize all subnets according to a scheduled time.

Step 9: Find Euclidean distance between two successive position estimations for each node, and check for a change in status of the node(s), if there exists a change in status then place node(s) in appropriate subnet.

Step 10: Based on new distance measurements perform step 1.

Step 11: Check if $N < N_T$, then sparsify and localize the network, by treating it as mobile, go to step 7.

Else: go to step 6

The resultant distributive algorithm giving high priority to mobile nodes can be used to track mobile nodes in real time.

We have applied this iterative, distributive localization algorithm to a WLAN with different node densities and dynamics. Simulation results, performance analysis and scalability of the algorithm are explained in subsequent sections.

In next section, brief introduction of a software programming tool used for simulation is presented.

4.4 Convex programming tool used within simulations

Simulations are performed using CVX [84], Matlab based routines developed for solving convex optimizations. CVX is a modelling system to solve convex optimization problems which are described using a limited set of construction rules and enables convex problems to be analyzed and solved efficiently [84].

CVX is implemented in Matlab, turning it into an optimization modelling language. Using this tool within Matlab, convex problem specifications are constructed using common Matlab operations and functions. Standard Matlab code can be freely mixed with these specifications. This combination makes it simple to perform the calculations needed to form optimization problems or to process the results obtained from their solutions.

Convex specific commands are separated from ordinary Matlab statements by enclosing them within a *cvx_begin* statement and a *cvx_end* statement. Within these two cvx specific statements, any ordinary Matlab statements, as well as cvx-specific commands can be included.

Within a CVX specification, optimization variables have no numerical value; instead, they are special Matlab objects. This enables Matlab to distinguish between ordinary commands and CVX objective functions and constraints. As Matlab reads a CVX specification, it builds an internal representation of the optimization problem. If it encounters a violation of the rules of convex programming (such as an invalid use of a composition rule or an invalid constraint), an error message is generated. When Matlab reaches the cvx_end command, it completes the conversion of the CVX specification to a canonical form, and calls the underlying core solver to solve it.

If the optimization is successful, the optimization variables declared in the CVX specification are converted from objects to ordinary Matlab numerical values which can be used in further Matlab calculations. In addition, CVX also assigns a few other related Matlab variables. One, for example, gives the status of the problem (*i.e.*, whether an optimal solution was found or the problem was determined to be infeasible or unbounded). Another gives the optimal value of the problem.

4.4.1 Constraints and SDP LMI constraints

Three types of convex constraints may be specified in CVX environments i.e.

1. An *equality constraint*, constructed using ==, where both sides are affine.
2. A *greater-than inequality constraint*, using either >= or >, where the left side is concave and the right side is convex.

3. A *less-than inequality constraint*, using either <= or <, where the left side is convex and the right side is concave.

Non-equality constraints, constructed using ~=, are never allowed. (Such constraints are not convex.)

In *semidefinite programming* (SDP) the constraints are typically expressed using *linear matrix inequality* (LMI) notation.

CVX provides a special *SDP mode* which allows this LMI convention to be employed inside CVX models using Matlab's standard inequality operators >=, <=, *etc.* In order to use it, one must simply begin a model with the statement *cvx_begin sdp* or cvx_begin SDP instead of simply cvx_begin. When SDP mode is engaged, CVX interprets certain inequality constraints in a different manner. To be specific:

Equality constraints are interpreted the same (*i.e.*, element wise). Inequality constraints involving vectors and scalars are interpreted the same; that is element wise. Inequality constraints involving non-square matrices are *disallowed*; attempting to use them causes an error. Inequality constraints involving real, square matrices are interpreted as follows:

X >= Y and X > Y become X - Y == semidefinite(n), where n=max(size(X,1),size(Y,1)).
X <= Y and X < Y become Y - X == semidefinite(n), where n=max(size(X,1),size(Y,1)).

If either side is complex, then the inequalities are interpreted as follows:

X >= Y and X > Y become X - Y == hermitian_semidefinite(n), where n=max(size(X,1),size(Y,1)).

X <= Y and X < Y become Y - X == hermitian_semidefinite(n), where n=max(size(X,1),size(Y,1)).

In effect, CVX enforces a stricter interpretation of the inequality operators for LMI constraints. Furthermore the CPU used for simulation has features, CPU speed = 3 GHz, with 4 GB RAM (Memory).

4.5 Experimental Test bed

In analyzing the distributive heuristic, we arbitrarily considered an indoor WLAN within PTCL, Academy, Pakistan. The floor plan details of the building in 2D are depicted in figure 4.1. The building consists of laboratories (labs), a library, class rooms and offices (OF). The building is divided into two portions by a corridor, on one side of the corridor are all labs, and other side comprises of offices and class rooms.

The distance measurement model given in equation 2.14 are used within the simulations with measurement noise assumed as zero mean Gaussian with σ = 0.02, and biased as Gaussian distributed with mean and variances as a function of distances.

4.6 Scalability Versus Speed

By considering an indoor environment depicted in figure 4.1 (with three anchors and randomly distributed nodes), we localized nodes increasing from 5 to 100 using SDP relaxed localization formulations given in equations 2.12a, b and 2.13. Furthermore, for iterative TSDP formulation (equations 2.12a,b) we considered three cases:, first with no iteration (only equation 2.12a is solved once we call it zero iteration), second and third cases are with one and two iterations respectively. The resultant average CPU time to localize nodes for all cases is tabulated in table 4.1. From the table it can be seen for DSDP (equation 2.13) and TSDP with no iterations that CPU time is almost equal except for 100

nodes. While with iterations TSDP consumes more processing time. Overall DSDP is computationally efficient compared to TSDP, but as we have seen in chapter 3 that if anchors are not placed at the outer boundary of the indoor environment, a low rank solution is not always guaranteed for a DSDP resulting in large errors. Dashed lines in table 4.1 under TSDP technique with 100 nodes indicate memory full errors which means that software cannot handle the problem.

TSDP is recommended to be used only with the condition that all nodes exist within the parameter of the anchors. If still for computation efficiency it needs to be used then virtual nodes (as discussed in chapter 3) can be added to stretch the resultant graph for improvements.

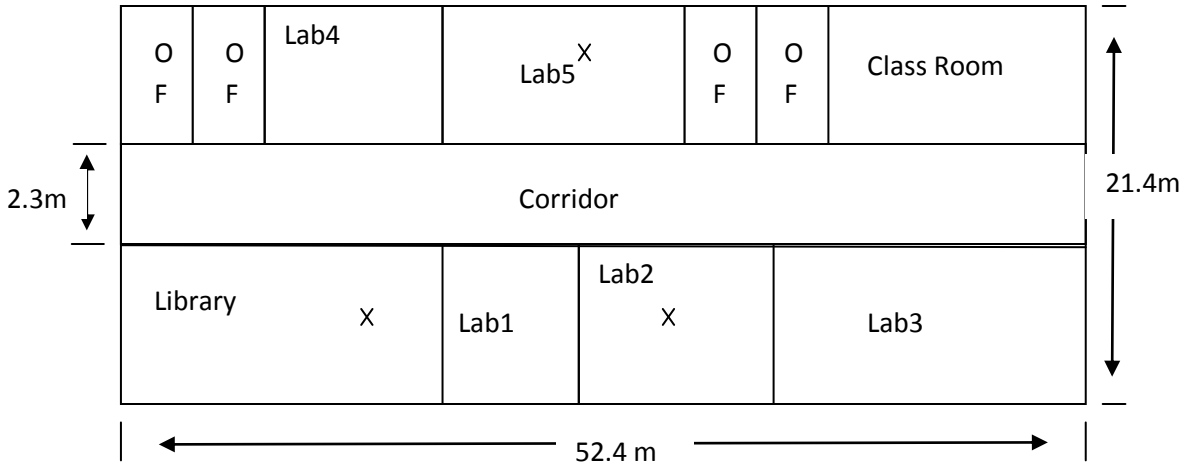


Figure 4.1: Floor plan of PTCL Academy building

Table 4.1: Effect of number of nodes on computational speed

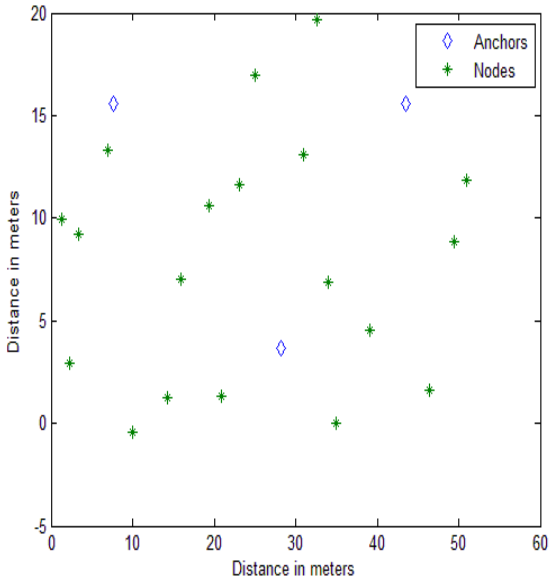
Total Number of nodes = Unknown Nodes + Anchors	TSDP Average CPU Time in Seconds (0 iterations)	TSDP Average CPU Time in Seconds (1 iterations)	TSDP Average CPU Time in Seconds (2 iterations)	DSDP Average CPU Time in Seconds
5+3	0.3011	0.6834	1.0094	0.2126
10+3	0.3249	0.8479	1.2792	0.3288
20+3	0.5594	1.4162	2.3506	0.5614
25+3	0.7135	1.9115	3.1422	0.7008
30+3	0.98	2.72	4.5313	0.9
40+3	1.5850	5.4502	8.9677	1.30
50+3	2.6861	12.8628	23.5855	1.9657
60+3	4.4957	35.9480	64.4469	2.9417
70+3	6.9885	82.2741	168.4219	6.7774
100+3	51.0660	-	-	27.6103

4.7 Impact of Creating Subnets

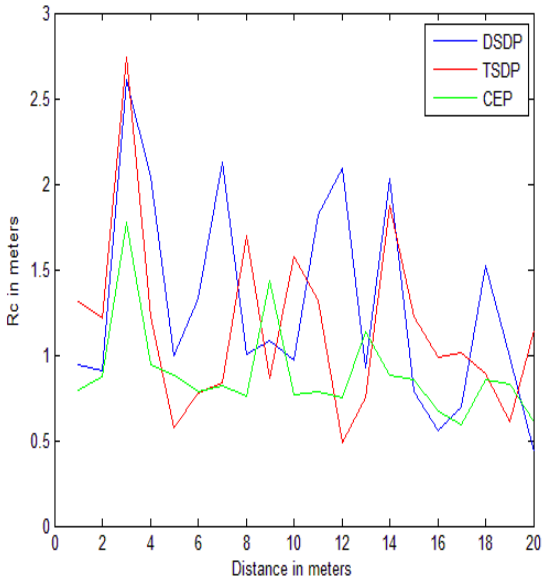
Let us consider a case of localizing 100 nodes. With reference to table 4.1 in order to localize these nodes using TSDP and DSDP, average CPU time of 51 and 27.6 seconds is required respectively. On the other hand if a network with 100 nodes is divided into two subnets consisting of 50 nodes each, then the time required will be approximately 1.965 + 1.965 ≅ 4 seconds for a DSDP technique, or 5.36 seconds for a TSDP with no iterations. This shows huge amount of reduction in computation complexity. Although for the case of sensor networks, not all the nodes can measure distances with anchors. Sensor-sensor distance measurements in addition to sensor-anchor measurements are used for localization. Dividing the network into subnets can generate localization errors, but WLANs are not similar to sensor networks due to diverse range of node densities. Therefore, by dividing a WLAN into subnets the localization computation complexity can be reduced.

4.8 Impact of Noise on Accuracy

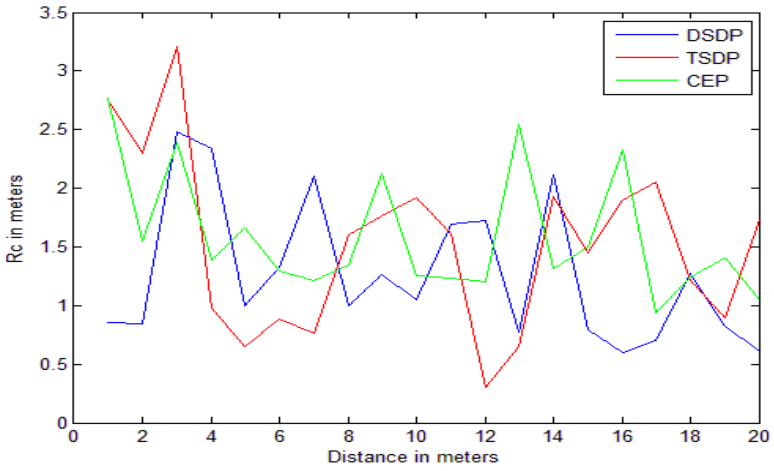
We have used CEP to measure the accuracy of the TSDP and DSDP estimators, i.e. we have found a confidence interval with 50% successful localization estimations. This interval is a function of measurements, node positions and type of estimation used. It is also upper bounded by theoretical values obtained by using equation 3.33. The impact of bias values within distance measurements is found using simulations by considering a subnet consisting of 20 randomly distributed nodes with three anchors as depicted in figure 4.2 (a). Figure 4.2 (b) shows simulation results for bias at 1 meter considered as 0.029m as per suggestion [39]. Figures 4.2 (c)-(e) show increase in bias values as 0.058 (that is 100% increase in biases), 0.087m, and 0.116m respectively.



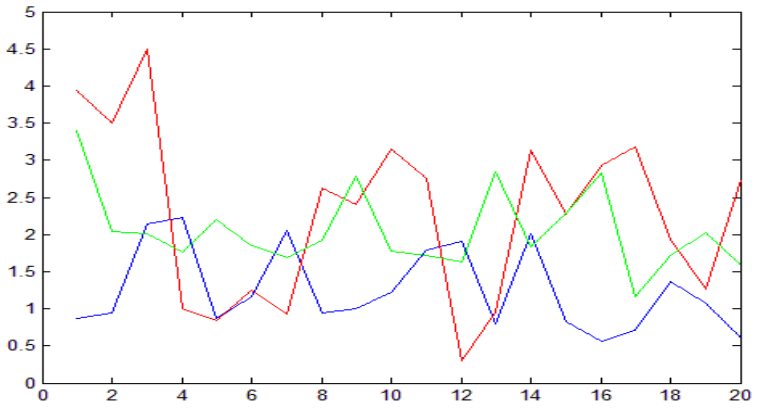
(a)



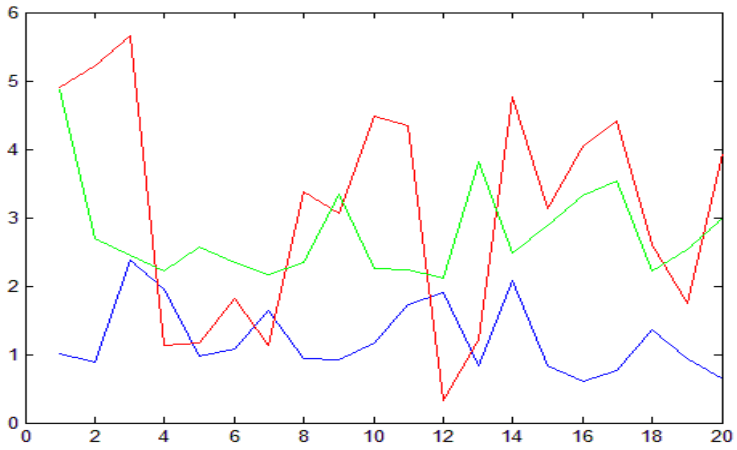
(b)


DSDP
TSDP
CEP
Rc in meters
Distance in meters
0
0.5
1
1.5
2
2.5
3
3.5
0
2
4
6
8
10
12
14
16
18
20

(c)


0
0.5
1
1.5
2
2.5
3
3.5
4
4.5
5
0
2
4
6
8
10
12
14
16
18
20

(d)



(e)

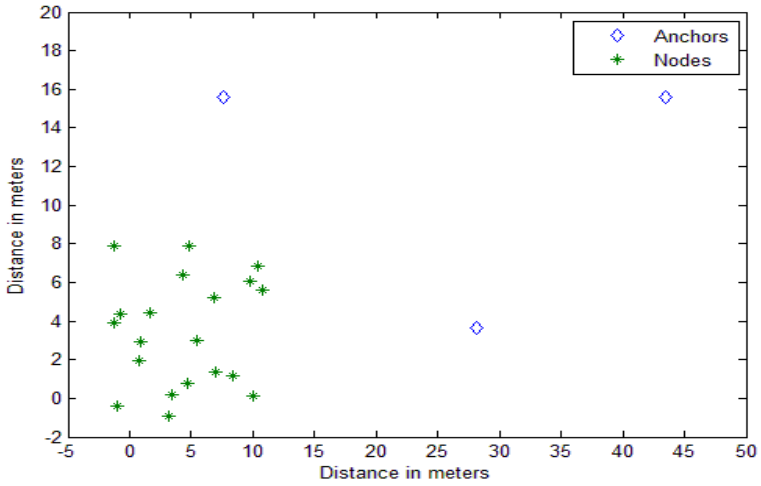
Figure 4.2 : CEP for the random node distributions, in all figures green line represents theoretical calculated Rc, blue line DSDP Rc, and red line TSDP Rc (a) network with 20 random unknown nodes (b)CEP with bias at 1 meter considered as 0.029 (c) CEP with bias at 1m = 0.058 (d) CEP with bias at 1m = 0.0870 (e) CEP with bias at 1m = 0.1160

Figures 4.2 (b-e) show that the increase in bias values has no effect on the CEP of a DSDP technique compared to TSDP which is much sensitive to bias. Hence DSDP gives better estimations compared to TSDP in addition to be computationally efficient. The disadvantage of higher rank solution of DSDP can be compensated by adding dummy or virtual nodes as discussed in chapter 3 and by creating stationary subnets by randomly selecting nodes, not based on geographical positioning as discussed previously. It may result in a non-stretched cluster. Example 4.1 demonstrates a high rank localization for DSDP optimizations.

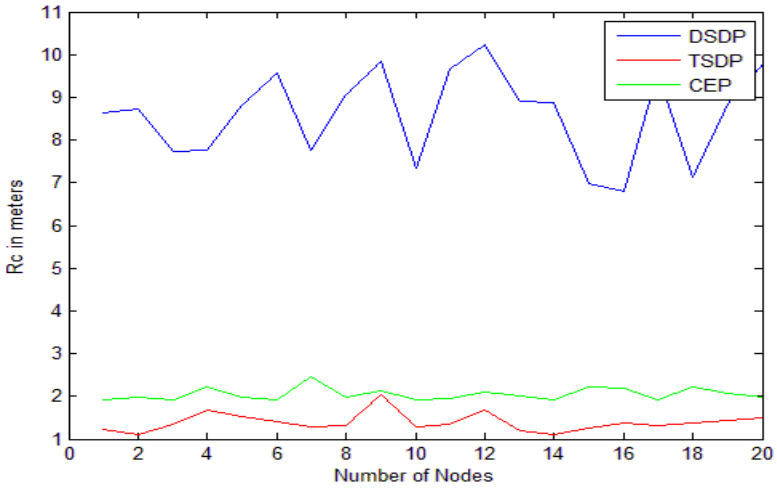
Example 4.1

Consider a network consisting of three anchors and 20 unknown nodes located very close to each other outside the parameter of anchors as shown in figure 4.3a. Simulated R_C values for TSDP, and DSDP are shown in figure 4.3b. As clearly seen from figure 4.3b, large

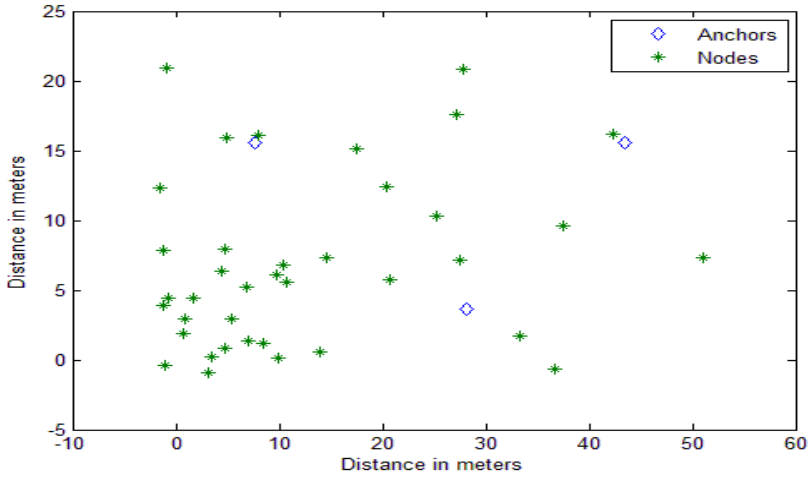
values of R_C indicate a high rank solution. We have added 20 random virtual nodes to the network of figure 4.3a as shown in figure 4.3c, resultant R_C values are given in figure 4.3d, the improvement in R_C is visible.



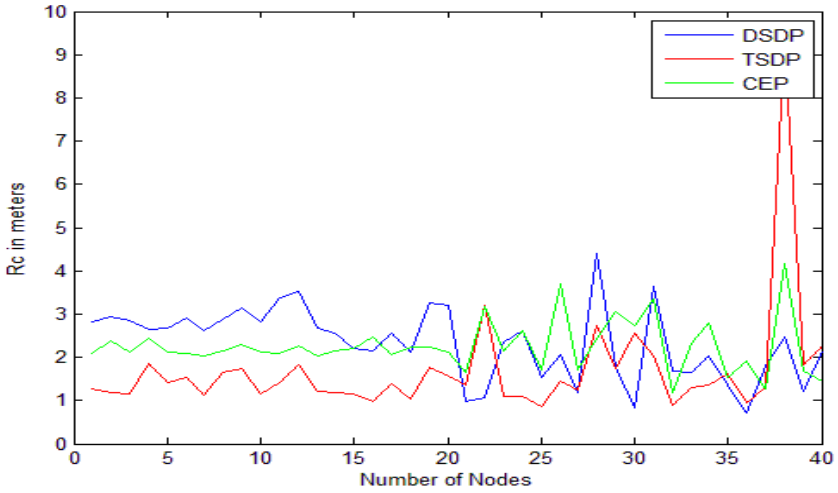
(a)



(b)



(c)



(d)

Figure 4.3: (a) network with 20 unknown closely located nodes (b) Rc of the network of figure 4.3a (c) network of figure 4.3a with 20 virtual nodes (d) Rc of the network of figure 4.3c.

In figure 4.3 d, nodes numbered 1 to 20 are unknown nodes while nodes from 21 to 40 are virtual nodes. From example 4.1 and from aforementioned discussions, if DSDP is to be used, it is recommended to add randomly distributed virtual nodes.

4.9 Identification of Mobile Nodes

Mobile nodes are separated from the stationary nodes by comparing Euclidean distance between two position estimations obtained from successive iterations (that is *i*+1, and *i*, where *i* is the i^{th} iteration) against a threshold value d_{Δ} given as:

$$if \quad |\bar{\mathbf{x}}_{i+1} - \bar{\mathbf{x}}_i|_2 > d_{\Delta}, \qquad node\ will\ be\ mobile \tag{4.1}$$

else node is stationary.

The value of d_{Δ} must be greater than R_C (i.e. 50% CEP), therefore threshold for detecting mobile node from the stationary node is given by:

$$|\bar{\mathbf{x}}_{i+1} - \bar{\mathbf{x}}_i|_2 > d_{\Delta} > R_c \tag{4.2}$$

Hence, d_{Δ} depends on distance measurements and type of localization technique used.

4.10 Selection of a Maximum Number of Nodes within Subnets (subnet size)

SDP based localization is computationally intensive for large number of nodes as can be seen from table 4.1. CPU time required for localization increases by increasing number of nodes. In section 4.3, we have described a distributive technique to reduce computational complexity by dividing a larger network into subnets with small number of nodes to be localized. The selection of maximum number of nodes within a subnet, (hereby called as *subnet size),* depends on factors like type of SDP formulation used i.e. DSDP or TSDP. For the case of TSDP, number of iterations used, and at the frequency with which updated localization of nodes are also required to be considered For WLANs, the mobility level is considered to be an average speed of human walk, in addition as discussed previously, not

all the nodes will be mobile all the time. Considering these factors, maximum number of nodes within a subnet can be selected.

By referring to table 4.1 for using TSDP with one iteration and three anchors, the choice of maximum number of nodes within a subnet 20+3 (nodes+ anchors) seems reasonable. It can be validated by considering the following example 4.2.

Example 4.2

Let there be only 60 stationary nodes within a network. If we consider a subnet of size 30, it will result in two subnets, adjusting parameters by considering table 4.1 and using TSDP with one iteration we obtain following results:

Total time required for localization = 2.72 + 2.72 = 5.44 seconds

If we reduce the maximum number of nodes within subnets and fix it as 20, this will result in 3 subnets, then

Total time required for localization = 1.5 + 1.5 + 1.5 = 4.5 seconds

Difference between the CPU time required for localization of subnets with 30 and 20 nodes is 0.94 seconds.

Let there be 30 total nodes within a network then for a 30 subnet size

Total time required for localization = 2.72 seconds

While for the case where subnet size is 20, it will result in two subnets with

Total time required for localization = 1.5 + 1.5 = 3 seconds

There is only a difference of 0.03 seconds in this case as compared to 0.94 seconds for the previous case. Number of nodes within subnets can be made adoptable for further

reduction in slot duration in some cases., For example, consider a network with 30 nodes, let subnet size be 20 then two subnets will be created. Instead of creating two subnets of size 20 and 10, if we select subnet size as 15 for each, it can result in further reduction in computation speed. Slot duration can be adjusted according to 15 subnet size which is less as compared to case of 20 subnet size.

For DSDP we recommend a subnet of size 40 with 20 unknown nodes and 20 virtual nodes.

4.11 Selecting Frequency of Iteration of Slots

In proposed distributive localization technique, the subnets are iteratively localized with a certain frequency. For this purpose we used time division multiplexing technique. Furthermore, we defined a *frame* in time domain as fixed time interval (T_f) of some duration. The frame is further divided into slots of equal duration (T_s). A subnet is localized within a slot; therefore duration of the slot must be adjusted so that a subnet of known size can be localized. The frame and slot concept is depicted in figure 4.4.

The size of the frame i.e. its duration depends on number of slots it contains, therefore, frame duration is the integer multiple of slot duration. The minimum size of a frame is one slot duration.

The duration of the slot depends on the maximum number of nodes it can have. By using table 4.1, the slot duration can be selected based on the localization technique used.

The frequency with which a subnet is localized depends on number of factors such as the number of nodes within a network, number of subnets, their types that is mobile or stationary (stationary nodes are less frequently localized) and number of slots allocated within a frame.

We have considered in our analysis a 1 : *n* ratio of subnet localizations within frames, that is each mobile subnets will be localized in every frame, while stationary subnets will be localized once in '*n*' number of frames. Example 4.3 explains how it works.

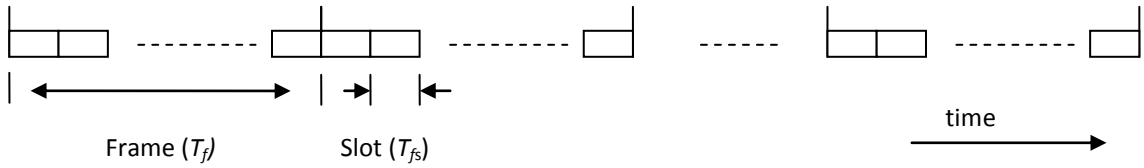


Figure 4.4: Frame and slot concept.

Example 4.3

As an example, consider a network of 100 unknown nodes, with 40 mobile and 60 stationary nodes. We set maximum number of nodes within a subnet to be 20, with this there will be two mobile subnets and three stationary subnets. If we are using TSDP with one iteration, then T_s can be selected from table 4.1 as 1.5 seconds. This network can be designed for a distributed localization by considering a ratio of 1:3, (a specific stationary subnet will be localized after every 3 frames) the resultant frame must consist of at least three slots, two for mobile subnet localization and one for a stationary subnet localization. All the five subnets scheduled within the frame are shown in figure 4.5, where m_i stands for i^{th} scheduled mobile subnet and s_i for i^{th} scheduled stationary subnet. Same notation will be used thereafter.

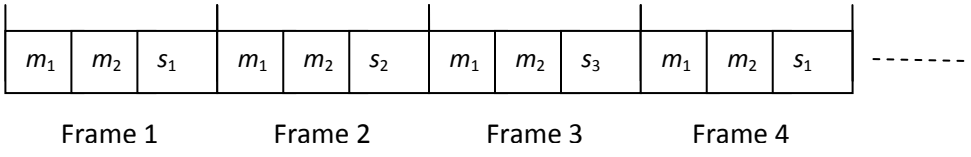


Figure 4.5: Frame structure of subnets of example 4.3

For this example using the distributive approach, the best we could achieve is:

Minimum Frame duration (T_f) = $3\times T_s$

Maximum Frequency of each mobile node localization = $1 / T_f$

Maximum Frequency of each stationary node localization = $(1 / 3T_f)$

The frame duration found in example 4.3 is the minimum duration for the desired network. If the requirement of localization update is not high, the frame duration can be increased by appending a frame with an empty time space at the end with the shaded interval as shown in figure 4.6, no localization is performed.

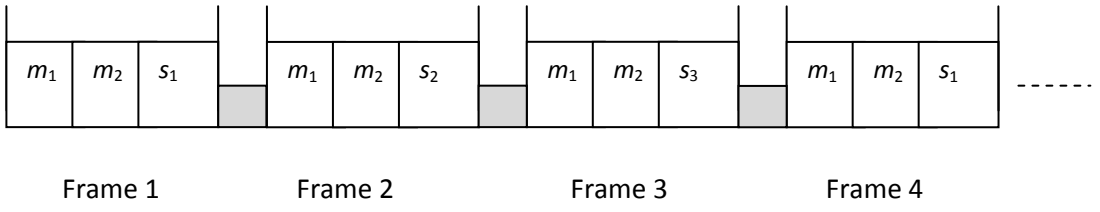


Figure 4.6 Increase in frame duration of example 4.3 by appending an empty space at the end of each frame.

As we have seen from example 4.3 that after designing a frame, frequency of subnet localization can be determined which depends on number of subnets and mobile / stationary subnet ratio.

4.12 Scheduling of Subnets

Scheduling of mobile and stationary subnets is done based on frequency of iterations of both types of subnets. This is defined based on localization ratios such as 1:2, 1:3, 1:4, … 1 : *n* of mobile and stationary subnets where 1:2 ratio means every mobile subnet will be localized once in a frame while every stationary subnet will be localized once in two frames. Figure 4.7 shows subnet ratio and frame relationships for 1:*n*, consisting of two mobile subnets, and one stationary subnet.

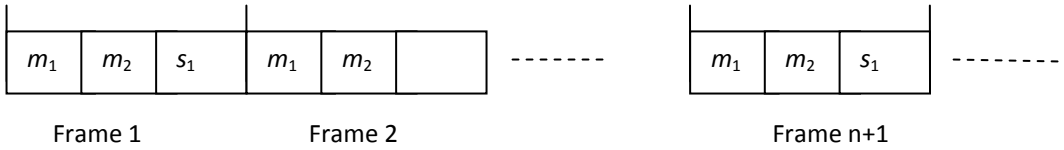


Figure 4.7 Scheduling of 1 : n ratio of mobile and stationary subnets within frames.

Let M_s be the number of mobile subnets and S_s be the number of stationary subnets, then scheduling of subnets within frame slots using mobile / stationary subnet ratio 1:*n* is performed as:

If $S_s \leq n$, then

Minimum number of slots per frame = $M_s + 1$ (4.3a)

If $n < S_s \leq k.n$, where k is integer, then

Minimum number of slots per frame = $Ms + k$ (4.3b)

Hence the frequency of frame and node localization is obtained as:

Minimum Frame duration (T_f) = $(M_s + 1) \times T_s$, if $S_s \leq n$ or (4.4a)

$$(M_s + k) \times T_s, \quad \text{If} \quad n < S_s \leq k.n \tag{4.4b}$$

Maximum Frequency of each mobile node localization = $1 / T_f$ (4.5a)

Maximum Frequency of each stationary node localization = $(1 / nT_f)$ (4.5b)

Example 4.4 to 4.6 further clarify scheduling of subnets within frames.

Example 4.4

Consider a network with M_s=1, and S_s=2 subnets with 1:3, the number of slots are computed as:

Here $S_s < n$, then

Minimum number of slots per frame = M_s+1 = 2

Example 4.5

Consider a network with M_s=1, and S_s=3 subnets with 1:3, the number of slots are computed as:

Here $S_s \leq n$, then

Minimum number of slots per frame = M_s+1 = 2

Example 4.6

Consider a network with M_s=1, and S_s=4 with 1:3, the number of slots are computed as:

n=3 < S_s = 4 ≤ 2.n = 6 then

Minimum number of slots per frame = M_s+2 = 3

The subnets of example 4.4 to 4.6 are scheduled within slots as depicted in figure 4.8 a-c. For the case of example 4.4 and 4.6, there will be ideal slots within frames. These ideal slots can be removed to decrease the frame durations as explained in next section.

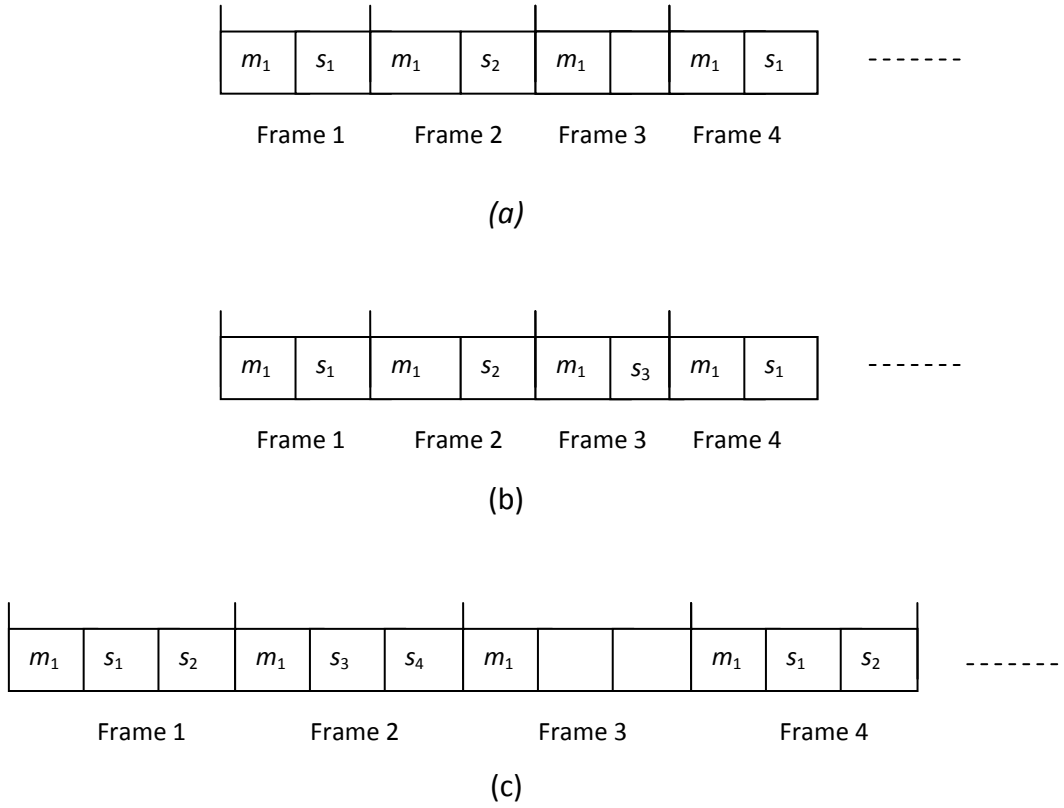


Figure 4.8: Scheduling of mobile and stationary slots within frames of example 4.4 – 4.6

4.13 Reducing Frame Time -- An Early Start Approach

The frame time can be further reduced in some cases such as given in examples 4.4 and 4.6, where instead of waiting during idle slot, the slot can be removed and next frame can start earlier as shown in figure 4.9, we call this approach as early start approach.

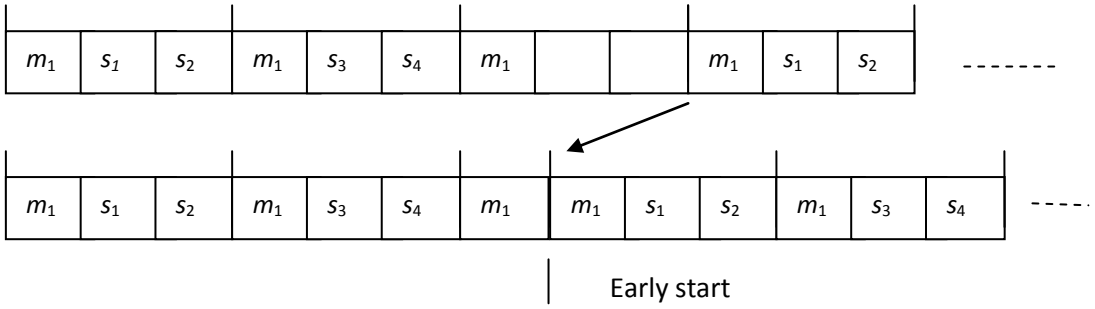


Figure 4.9: Frames with early start approach.

4.14 Scalability versus Capacity

Scalability as defined in chapter 1 is the response of the localization algorithm to large number of nodes, while capacity refers to number of node estimations, a system can process per unit time. Our proposed distributive time division multiplexed algorithm is highly scalable. As the number of nodes increases it results in more subnets which further increases time slots within frames and the frame duration also increases.

Our distributive system with time division multiplexed scheduling is always traceable, we can always measure its capacity. The capacity of the system can be determined from the number of slots within a frame, subnet size, type of localization technique used and frame duration as:

Let S_T be the total number of slots within a frame of fixed duration T_s, and subnet size be S_{size}. From this we can calculate total number of nodes that can be localized within a frame, and frame duration T_f as:

Total number of nodes, a frame can localize = Total slots within a frame × Subnet size = S_T × S_{size} (4.6)

Frame Duration (T_f) = Total slots within a frame × Slot duration = $S_T \times T_s$ (4.7)

Capacity is computed as:

Capacity = Total number of nodes a frame can localize / Frame Duration =

$$(S_T \times S_{size}) / (S_T \times T_s) = S_{size} / T_s \quad (4.8)$$

The capacity of our method is always fixed and depends on subnet size and time required to localize a subnet.

4.15 Scalability versus Frequency of Localization

As we have seen from equation 4.8 that scalability has no effect on the capacity of the system which is always fixed. The factor that is affected will be the frequency with which we can re-localize the nodes. We have classified the nodes as mobile and stationary with mobile nodes being more frequently being localized as compared to stationary nodes. Frequencies of mobile and stationary node localization are given in equations 4.5a and 4.5b respectively. Both frequencies depend on frame duration which further is a function of number of nodes, hence by using equation 4.4 and 4.5 frequency of node localization can be found.

4.16 Scalability versus Time Required to Localize all Nodes

In our distributive algorithm we localize the mobile and stationary nodes with frames scheduled 1 : n, which means that all stationary and mobile nodes will be localized within n number of frames. So the total time required to localize all the nodes can be computed as:

Total time required to localize all nodes = frame duration × n = $T_f \times n$ (4.9)

Again in equation 4.9, frame duration depends on total number of nodes and their types. Also note that within this time if there exist mobile subnets, then mobile nodes will be localized *n* times.

Next we present a case study showing how aforementioned parameters can be used in designing the distributive localization solution.

4.17 Case Study

In this section, we consider an arbitrary WLAN within an indoor environment as shown in figure 4.1 with three anchors and 300 randomly distributed nodes, consisting of mobile and stationary nodes. Let there are 50 mobile nodes and they are only present within a corridor, we also assume a distance measurement model of equation 2.14. It is also assumed that prior statistical characteristics of distance measurements are known.

For this case, we will design a distributive TDM localization solution by finding the accuracy of the positions, and the maximum frequency with which nodes can be localized. Let us set the mobile and stationary subnet localization ratio to be 1:5.

To design a distributive localization solution, we begin by calculating accuracy of the localization. As we have seen in chapter 3, for biased distance measurements given in equation 2.14, resultant estimations will be biased., Therefore, CRL bounds cannot be used in determining accuracy. For this case, CEP is used and R_C values are determined which further depend on measurements and technique used for localization. However, upper bounds of *Rc* values can be found using statistical characteristics of measurements and equation 3.33. Therefore accuracy of this network can be calculated by using equation 3.33 with prior statistical characteristics of measurements.

For maximum node localization frequency, we first of all design a distributive solution by finding the number of mobile and stationary subnets. From section 4.11, we have seen that subnet size (maximum number of nodes a subnet can have) of 20 is better than other

sizes, therefore, we set subnet size as 20. Let us select TSDP with one iteration as localization technique. Then from table 4.1, we set slot duration (T_s) as 1.5 seconds.

There are 50 mobile nodes which will result in three mobile subnets (M_s), while for remaining 250 nodes, there will be 13 stationary subnets (S_s), therefore, total number of subnets will be 16.

For 1 : 5 scheduling, the frame and slots are determined by using expressions given in section 4.12 as:

$5 < S_s < 3 \times 5$

Therefore k=3

Minimum number of slots per frame = $Ms + k = 3 + 3 = 6$

Minimum Frame duration (T_f) = $6 \times 1.5 = 9$ seconds (1.5 seconds is slot duration)

Maximum Frequency of each mobile node localization = $1 / T_f = 0.11$ Hz

Frequency of each stationary node localization = $(1 / 5T_f) = 0.022$ Hz

Time after which each mobile node will be re-localized = 9 seconds

Time after which each stationary node will be re-localized $= 9 \times 5 = 45$ seconds

Capacity = $S_{size} / T_s = 20 / 1.5 = 13.33$ nodes per second.

4.17.1 Real Time Traceability

As we have seen in the case study that every mobile node will be localized after 9 seconds. If a mobile user is walking with an average speed around 80 meters/ minute, then within two localization iterations, a mobile user might have covered a distance of 12 meters. The best we can achieve is reduction in 9 seconds interval by localizing only one stationary

subnet within a frame compared to three as we considered previously for the network under study. With this, mobile to stationary scheduling ratio becomes 1:13 as there are total 13 stationary subnets and we have to localize at least one stationary subnet per frame. With 1:13 we get new values as:

Slots per frame = 3+1 =4

Frame duration (T_f) = 4 × 1.5 = 6 seconds (1.5 seconds is slot duration)

Maximum Frequency of each mobile node localization = 1 / T_f = 0.166 Hz

Frequency of each stationary node localization = (1 / 13T_f) = 0.0128 Hz

Time after which each mobile node will be re-localized = 6 seconds

Time after which each stationary node will be re-localized =6 × 13 = 78 seconds

The distance travelled by the mobile node user within successive iterations (assuming average human walk) = (80/60) × 6 = 8 meters

The time and distance covered in this case are reduced to 3 meters, but it is the best that could be achieved.

4.18 Summary

In this chapter, we have explained the novel distributive WLAN localization heuristic in where localization process is made computationally efficient by dividing the larger WLAN into smaller subnets consisting of small number of nodes. Furthermore, two types of subnets are created, mobile and stationary. This characterization is made to localize mobile nodes more frequently compared to stationary nodes. With this technique, only small portion of the nodes are localized at one time with highest priority given to mobile nodes, and all of the nodes present within the network are localized in an incremental

manner. The resultant technique can be used for tracking mobile nodes within the network in real time.

Furthermore, we have explained how developed distributive technique works and how its computation performance can be calculated. Scheduling of subnets is also explained in detail along with the method of determining maximum speed with which nodes can be localized. A case study is presented in which we designed a distributive localization solution and determined its accuracy and capacity (number of node estimations per unit time).

Chapter 5

3D localization

In this chapter, we extend the analysis of localization to three dimensional space. The problem is reformulated for 3D localization and necessary conditions for unique realization are discussed. Simulation based analysis are performed to localize nodes in 3D. A computation efficient distributive TDM technique for 3D localization is also proposed.

5.1 Problem Formulation for 3D

Different 3D localization techniques such as multidimensional scaling based techniques for sensor networks discussed in literature can be found in [94]. Anchor free 3D localization is presented in [85]. Another technique known as COLA which is based on non-convex localization is proposed in [93]. COLA reduces computational complexity of localization by introducing pair of anchors known as super anchors. Super anchors are pair of anchors placed in 3D space with a constraint that each pair is located at the same *x* and *y* location but at different *z* coordinates. This configuration can be used to transform 3D localization problem into 2D localization resulting in lower computational complexity. Constraint of COLA approach is the requirement of super anchors. In addition errors are induced during 3D to 2D transformation. Cause of the errors is use of trigonometric laws during 3D to 2D transformation instead of using any optimization technique to mitigate errors. Our proposed technique for 3D localization avoids such constraints and proposes computational efficient heuristic. Moreover COLA technique can be improved by investigating the possibility of using convex optimization.

Here we have analyzed localization for WLANs based on convex optimization. In many applications 2D localization will be sufficient. For example, in a single floor indoor environment, 2D localization indicates position of the nodes within rooms without any

height estimates of the nodes, where height of the node might not be of any interest. Also, in most indoor cases every floor is covered by access points whose range is limited to that floor only, therefore, 2D localization in such cases will be sufficient. But in some other applications, like in a multi-floor building where coverage of access points is not limited to a single floor, floor information may be of interest and will be required for complete positioning. The 3D localization for multi-floor environment would estimate the floor in which node exists in addition to its planar 2D positioning.

For the case of 3D localization, realization matrix **X** consisting of *N* number of total nodes will have the dimensions $R^{3\times N}$, and distance measurements will be as given in equations 2.3a-c, rewritten here as:

$$\left\|\mathbf{x}_i - \mathbf{x}_j\right\|_2^2 = d_{ij}^2 \quad \forall\ (i,j) \in I_{an} \tag{5.1a}$$

$$\left\|\mathbf{x}_i - \mathbf{x}_j\right\|_2^2 = d_{ij}^2 \quad \forall (i,j) \in I_{nn} \tag{5.1b}$$

$$\left\|\mathbf{x}_i - \mathbf{x}_j\right\|_2^2 = d_{ij}^2 \quad \forall (i,j) \in I_{aa} \tag{5.1c}$$

where vector $x_i \in R^3$ represent i^{th} node position and is expressed as:

$$\mathbf{x}_i = (x_i, y_i, z_i)$$

The conditions under which a unique solution to 3D equations 5.1a-c exists will be the presence of at least (*d*+1=4) number of *non-planar* anchors, and a trilateral graph formed with measured distances. One example of a trilateral graph with non-planar anchors can be as shown in figure 5.1. Where anchors connected by a shaded triangle are assumed to lie in a plane.

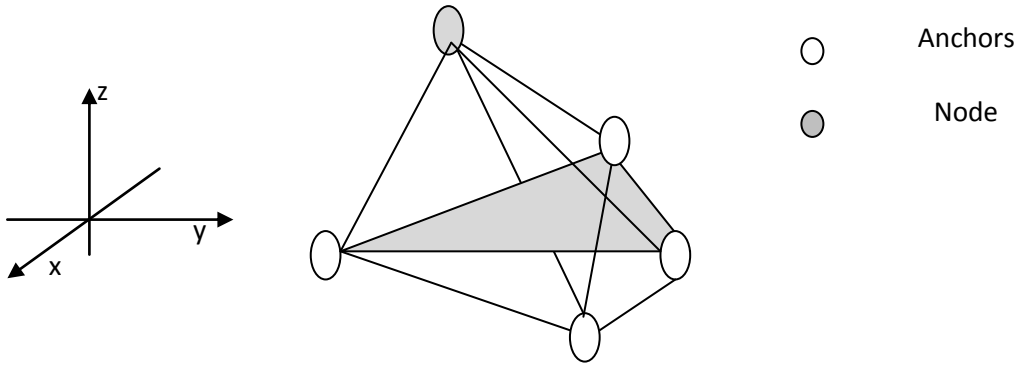


Figure 5.1: Anchors and a node in 3D environment forming a trilateral graph.

In the absence of any noise and assuming only the anchor-node distance measurements, for a trilateral 3D network having four anchors and an unknown node, the graphical solution of equations 5.1a-c will be the intersection of four spheres centred at each anchor having a radius equal to anchor-node distance.

As noise is always present therefore estimation techniques are used for localization. We can easily extend SDP techniques given in equations 2.12 and 2.13 for a 3D case by adding *z*-coordinates in **X** realization matrix. For example 3D formulation of an example 2.2 would be:

Example 5.1

Let us extend example 2.2 for 3D localization by considering four nodes placed in a non-planer network in R^3 Euclidean space with anchors located at $\mathbf{a_1}$= (x_1, y_1, z_1), $\mathbf{a_2}$= (x_2, y_2, z_2), and $\mathbf{a_3}$= (x_3, y_3, z_3) and a node at $\mathbf{x_4}$=(x_4, y_4, z_4) in a Cartesian coordinate system as shown in the figure 5.2. Let all the distances are measurable then the localization problem is formulated using TSDP as:

$$minimize \qquad tr(\mathbf{Z}, \mathbf{W})$$

$$Subject\ to\ tr\left(\widetilde{d_{ij}}\right)^2 \leq \left(\mathbf{F}\left(\mathbf{e_i}-\mathbf{e_j}\right)\left(\mathbf{e_i}-\mathbf{e_j}\right)^T\right) \leq \left(\widehat{d_{ij}}\right)^2,\ \forall (i,j) \in (I_{aa} \cup I_{an} \cup I_{nn})$$

$$tr\left(\mathbf{F}\mathbf{e_i}\mathbf{e_i^T}\right) = \|x_i\|^2, \quad i \in N_a$$

$$tr\left(\mathbf{F}\left(\mathbf{e_i}\mathbf{e_j^T}+\mathbf{e_j}\mathbf{e_i^T}\right)\frac{1}{2}\right) = x_i^T x_{j,} \quad i<j, \qquad \forall i,j \in N_a$$

$$\mathbf{X}(1{:}\,3, 2{:}\,1+N_a) = [\mathbf{a_1} \ \cdots \ \mathbf{a_{Na}}]$$

$$\mathbf{Z} = \begin{bmatrix} \mathbf{I} & \mathbf{X} \\ \mathbf{X^T} & \mathbf{F} \end{bmatrix} \succcurlyeq 0$$

and

$$minimize\ \langle \mathbf{Z}, \mathbf{W} \rangle$$

$$subject\ to \quad \mathbf{0} \preccurlyeq \mathbf{W} \preccurlyeq \mathbf{I}$$

$$tr\,\mathbf{W} = \mathbf{N}$$

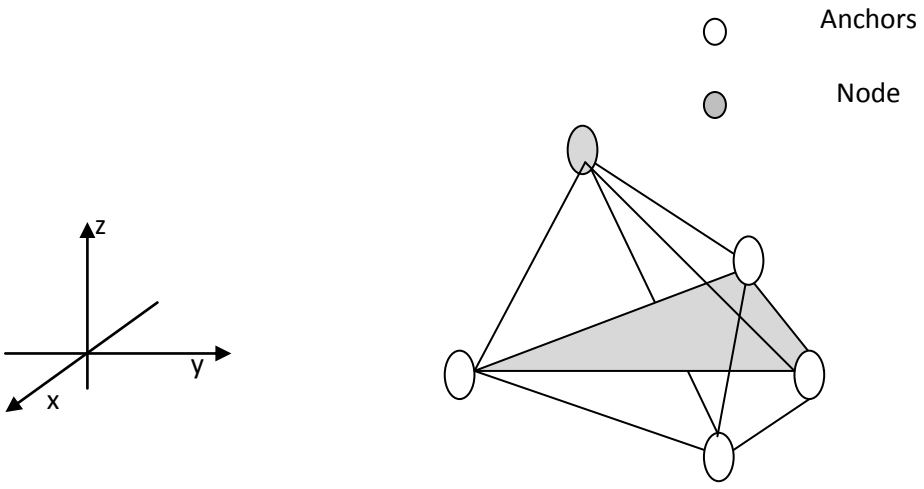


Figure 5.2: Anchors and a node in 3D environment forming a trilateral graph for example 5.1.

5.2 Simulations for 3D Localization

In this section, we present the simulation of a 3D localization by extending the WLAN given in figure 4.1, to a 3D network space. Let us assume single floor of an indoor environment given in figure 4.1, having a height of 15 feet as shown in figure 5.3. We have selected *x-y* coordinates to represent floors while *z*-coordinate indicating height from the ground as depicted in figure 5.3.

Let there be 3 anchors fixed arbitrarily along the walls at equal height of 10 feet from the ground, while fourth anchor be fixed along the roof at a height of 15 feet. We assume to localize forty randomly placed nodes within this indoor environment with an average height of 3 feet (height of nodes uniformly varies between one to five feet). The resultant estimated positions in *x-y* coordinates, *x-z* coordinates, CEP values in *x-y* direction and CEP values in *x-z* direction are depicted in figures 5.4. In figures 5.4 a- b, lines between the estimates and actual node positions indicate localization errors. Within simulations we assumed biases within measured distances to be normally distributed having mean

function of distance as found in [39], and considered mean value at 1m as 0.058m from [39] and σ = 0.028. Furthermore, for the TSDP optimization technique we considered no iterations.

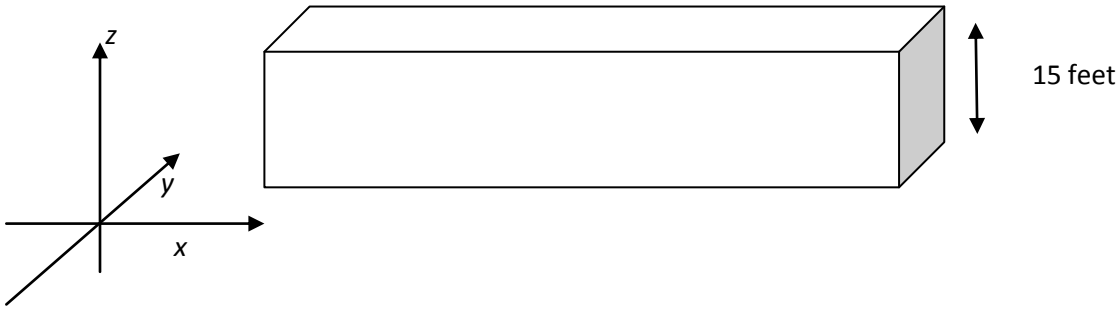
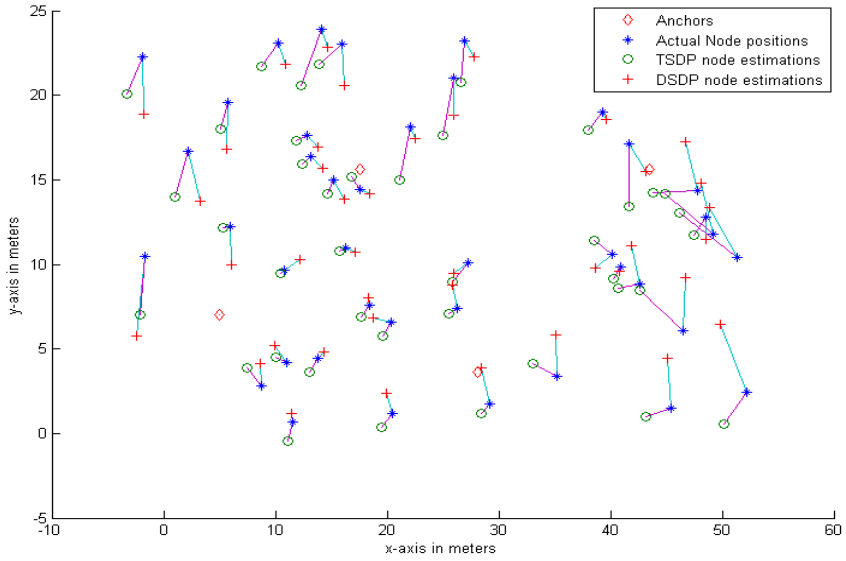
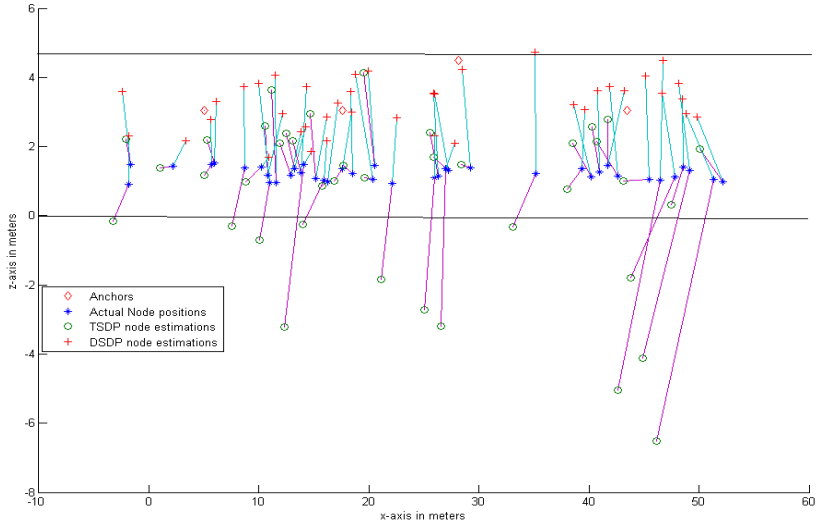


Figure 5.3: 3D representation of figure 4 .1 with one floor

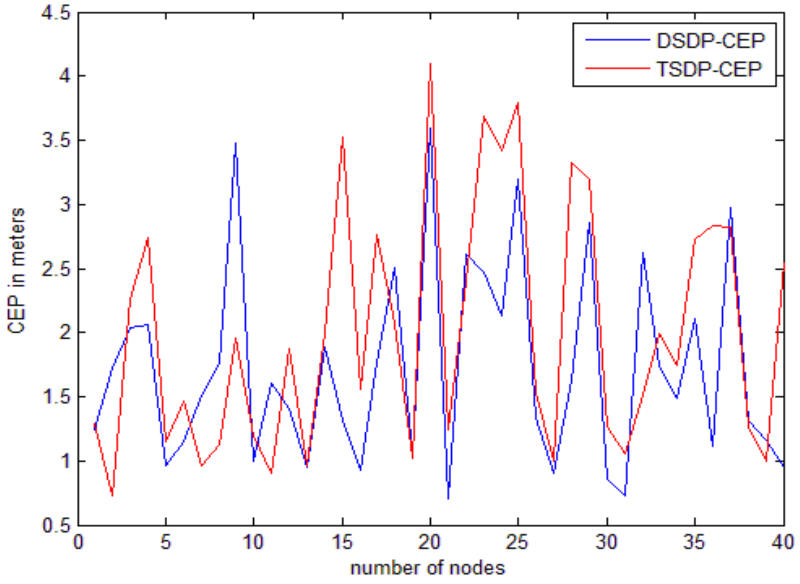
We have found from simulations that the errors in position estimations in *x-z* directions compared to *x-y* directions by comparing CEP figures 5.4 c and d are large.


Anchors
Actual Node positions
TSDP node estimations
DSDP node estimations
y-axis in meters
x-axis in meters
25
20
15
10
5
0
-5
-10
0
10
20
30
40
50
60

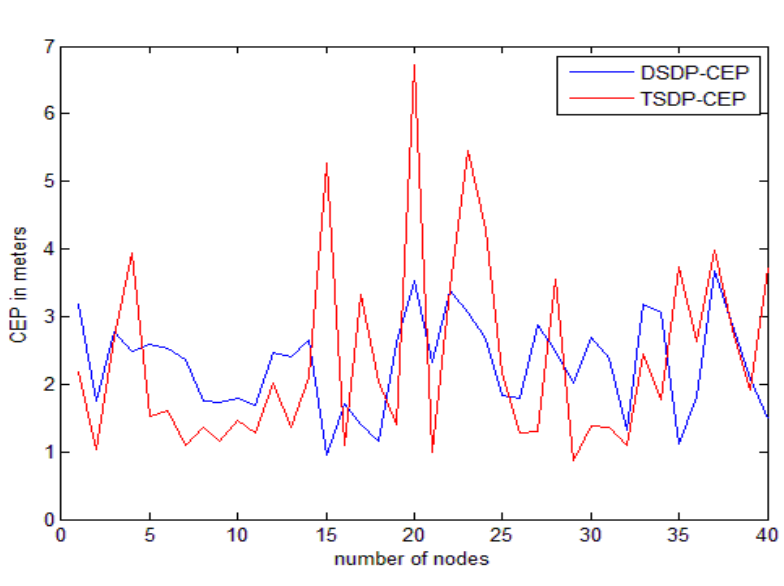
(a)


Anchors
Actual Node positions
TSDP node estimations
DSDP node estimations
z-axis in meters
x-axis in meters
6
4
2
0
-2
-4
-6
-8
-10
0
10
20
30
40
50
60

(b)



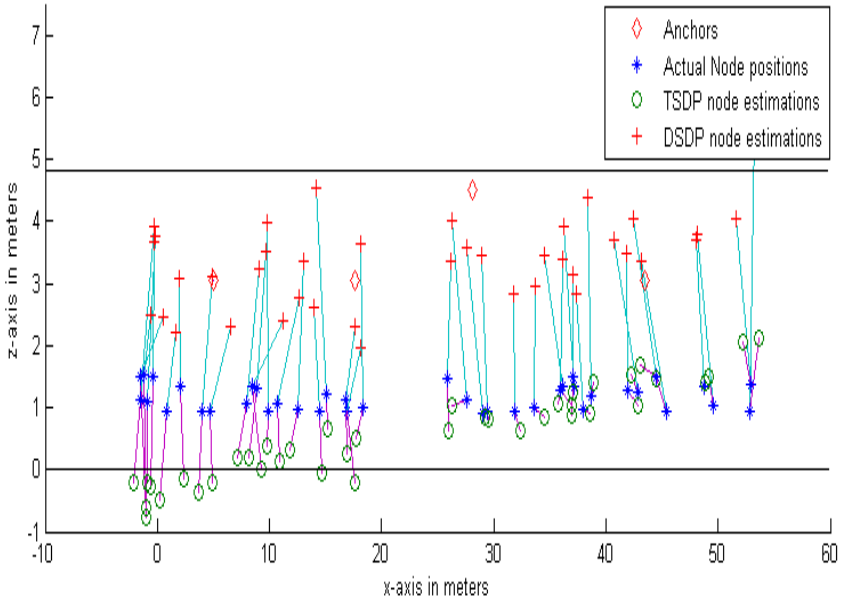
(c)



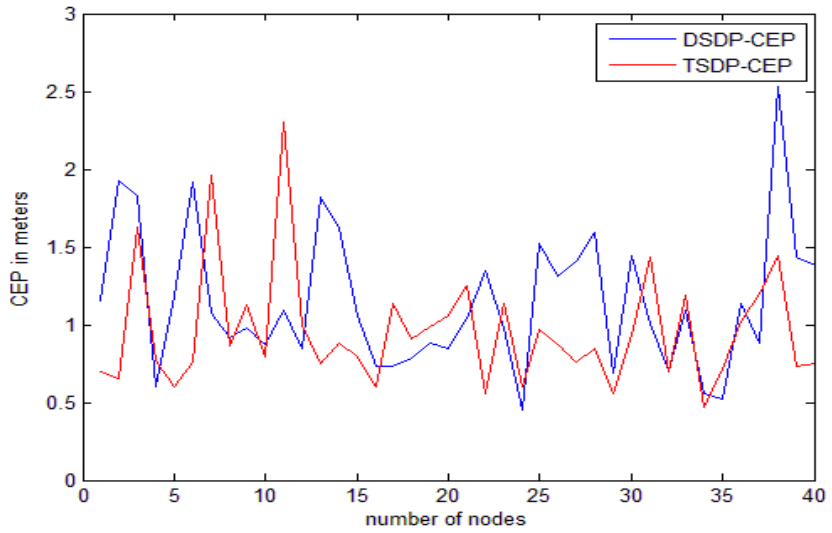
(d)

Figure 5.4: (a) x-y position estimations (b) x-z position estimations, two horizontal lines show a ground and a floor of the building (c) Rc values in x-y coordinates, (d) Rc values in x-z coordinates.

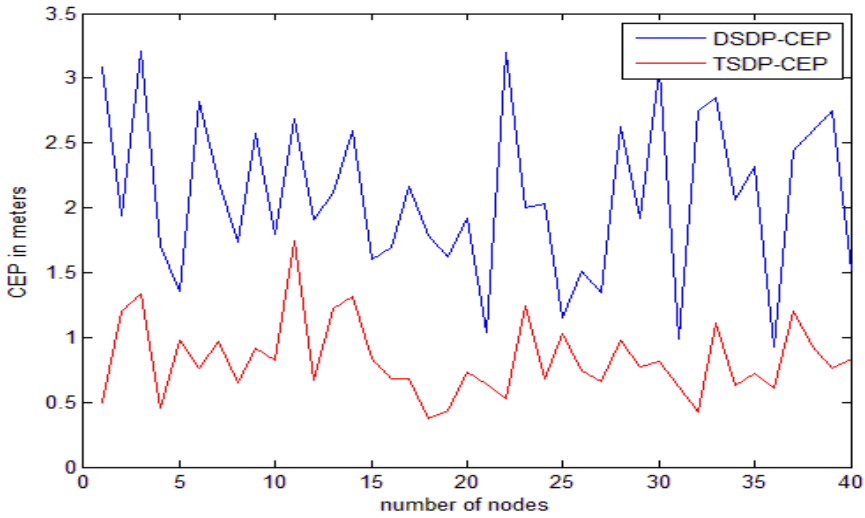
We have increased the iterations for a TSDP technique to six, resultant simulation results of position estimations in *x-z* directions and corresponding CEP are depicted in figures 5.5 a-c. There were significant improvements in position estimations in *x-z* coordinates for a TSDP estimations, see for example figure 5.5c, but at the cost of computation complexity. CEP values for TSDP in *x-z* coordinates are reduced from 1.5m to less than 1m. On the other hand, more iterations has no effect in *x-y* position estimations between two estimation techniques, see for example figure 5.5 b.



(a)



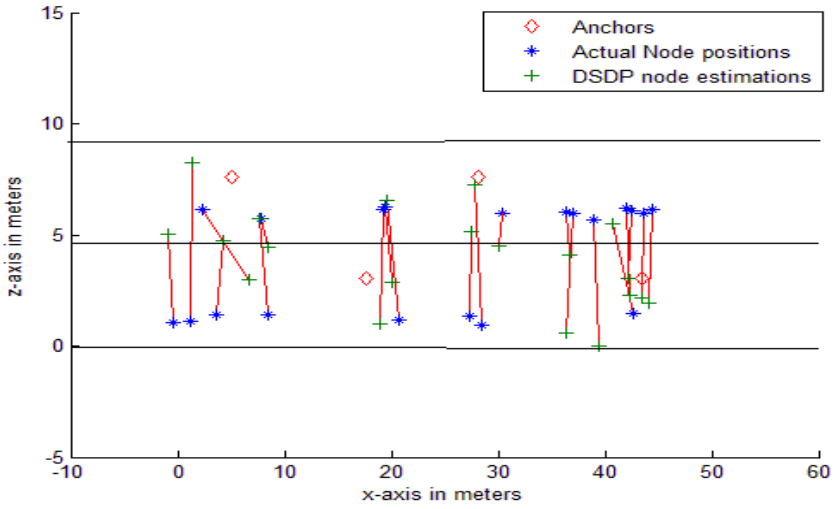
(b)



(c)

Figure 5.5: (a) x-z position estimations, two horizontal lines show a ground and a floor of the building (b) R_C values in x-y coordinates with 6 iterations, (c) R_C values in x-z coordinates with 6 iterations.

Moving forward, instead of considering a single floor, we added another floor to our simulation model. We placed two anchors at each floor and localized twenty five randomly placed nodes at each floor, again uniformly distributed between one to five feet. Position estimations in *x-z* coordinates are given in figure 5.6.



(a)

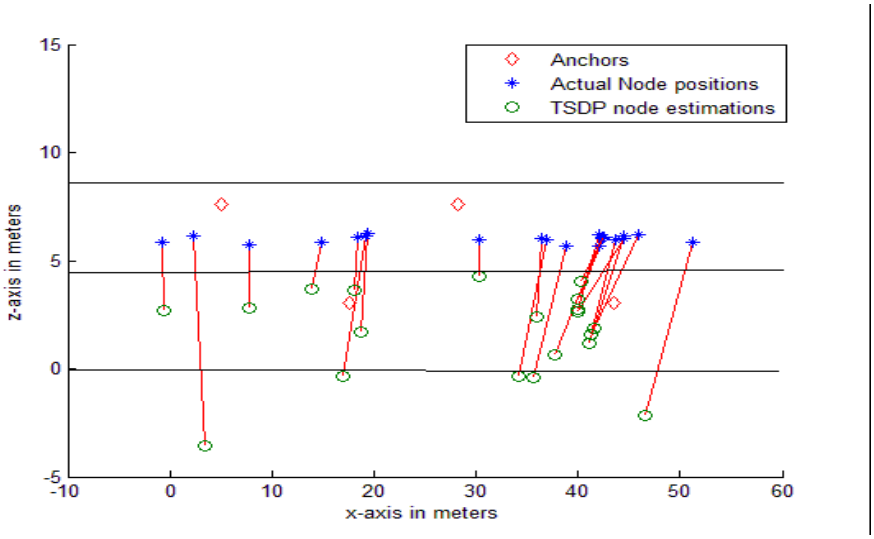
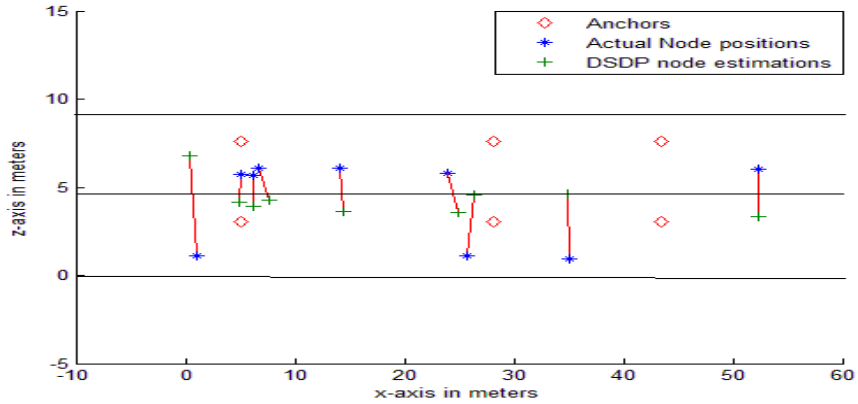
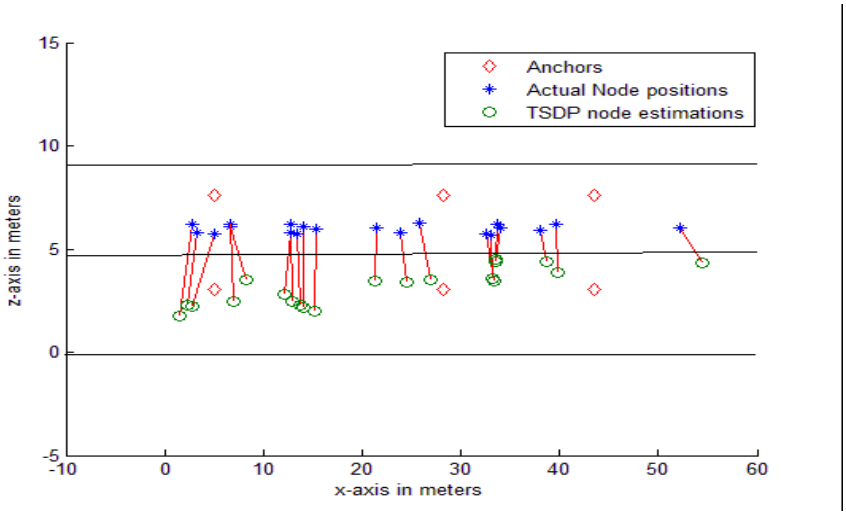


Figure 5.6: (a) Nodes with floor estimation errors by using DSDP (b) nodes with floor estimation errors by using TSDP.

We repeated simulations by adding one anchor at each floor to the previous case that is we placed a total of six anchors, three at each floor, we found improvements in DSDP results for estimating floor of the nodes as shown in figure 5.7a-b.



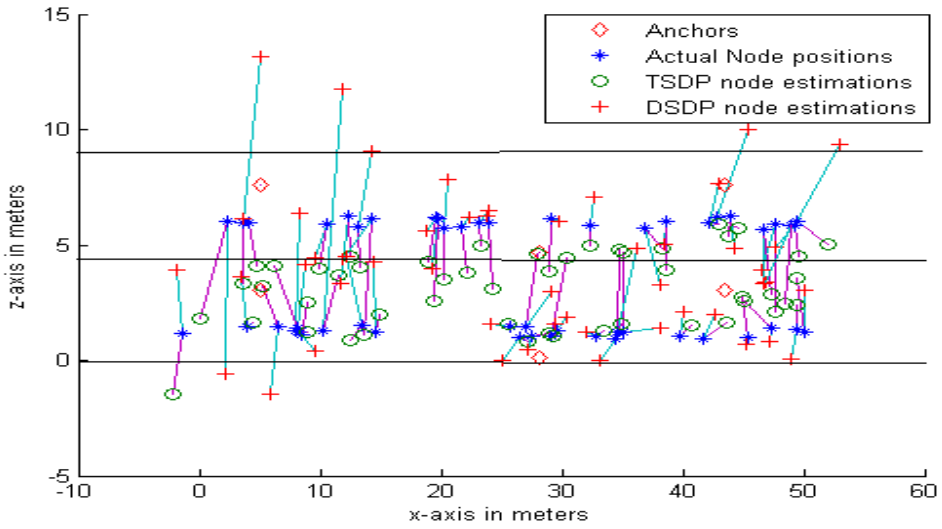
(a)



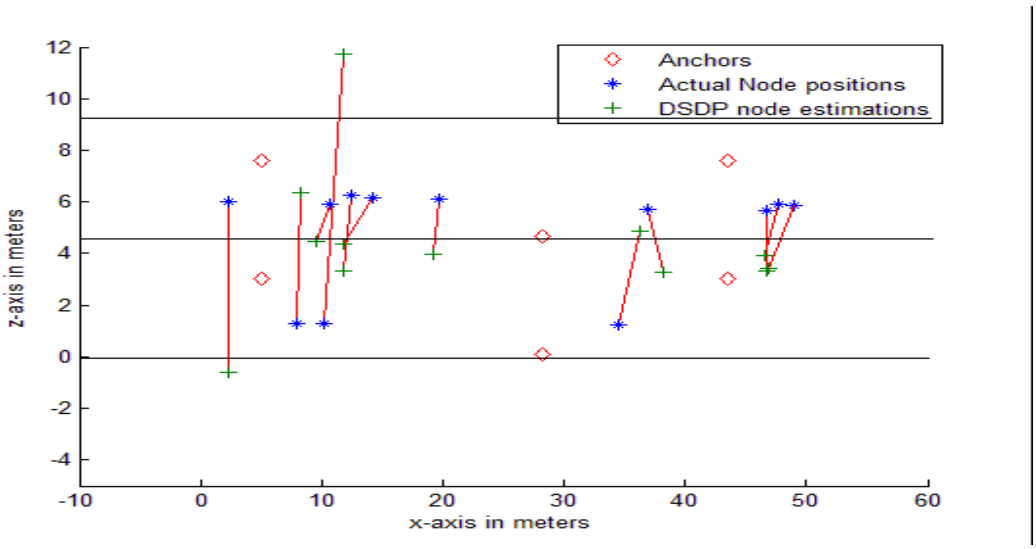
(b)

Figure 5.7: (a) Nodes with floor estimation errors by using DSDP with 6 anchors (b) nodes with floor estimation errors by using TSDP with 6 anchors.

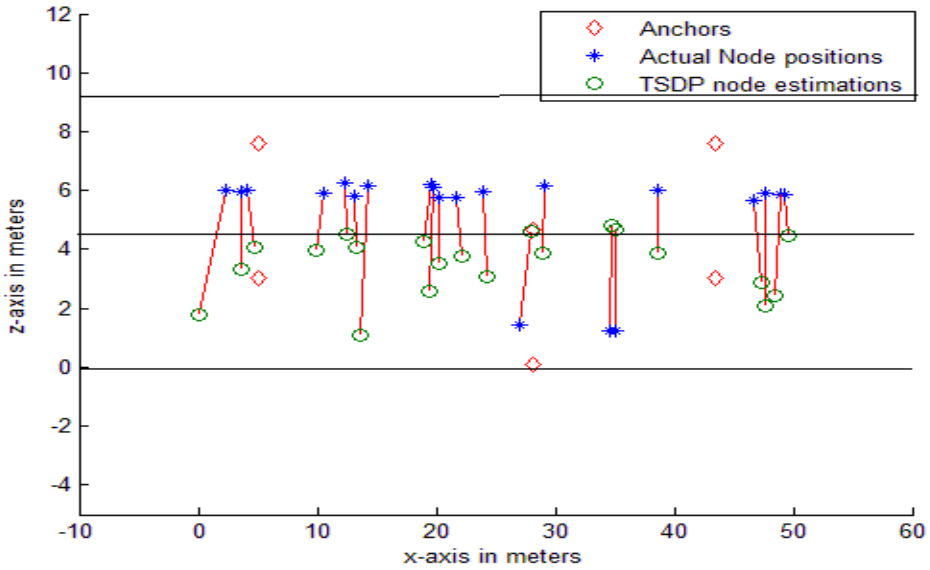
To analyze the effects of anchor positions, we placed one anchor at a ground of each floor instead of placing at the walls near roof, simulation results with ground located anchors are depicted in figure 5.8 a-c.



(a)



(b)



(c)

Figure 5.8: (a) x-z position estimations (b) nodes with floor estimation errors by using DSDP with 6 anchors (c) nodes with floor estimation errors by using TSDP with 6 anchors.

Effects of number of anchors on the performance of localization in floor identification are analyzed by simulating an indoor environment consisting of two floors with dimensions given in section 5.2. Within simulations twenty randomly distributed nodes are placed in each floor and anchors are varied from four to eleven. Nodes are localized and floor identification is performed by using TSDP (with two iterations) and DSDP techniques. Within simulations we assumed biases within measured distances to be normally distributed having mean function of distance as found in [39], and considered mean value at 1m as 0.058m from [39] and σ = 0.028. The percentage of incorrect floor identification is computed and is given in table 5.1. From table 5.1, it could be observed that DSDP estimations outperformed TSDP Furthermore in order to improve percentage of floor identification number of anchors needs to be increased.

Table 5.1: Effects of number of anchors on floor estimations

No of Anchors	Percentage of floor estimation errors using TSDP	Percentage of floor estimation errors using DSDP
4	46	44
6	44	27
7	43	25
9	31	23
11	25	18

5.3 2D Approximations of a 3D Problem

We have also analyzed the localization problem by distributing it dimension wise that is 3D problem is formulated to be localized in 2D planar *x*-*y* space. In this case, distance

measurements used are same as for 3D localization, that is, no transformation is performed on measured distances. The performance of this 2D approximation is evaluated by simulating the experimental setup as described in section 5.2. The nodes are placed randomly within a 3D space, measured distances between nodes and anchors are generated using signal model given in equation 2.14. These generated measured distances for 3D space are used to localize nodes within *x*-*y* planer space by using 2D TSDP and DSDP techniques given in equations 2.12 and 2.13 respectively. Simulated results show that there is an Euclidean difference of around 0.4m only between estimates obtained by solving a problem in 3D and its approximations using 2D optimizations for the case of DSDP as shown in figure 5.9. It means that if only planar *x*-*y* position of nodes is to be estimated for nodes present in 3D then these estimates can be approximated by solving the problem by formulating it in 2D. Error incurred will not be more than 0.4 meters. The advantage of this approximation is that it minimizes computational load. In most cases, the rate with which mobile nodes change floors will be very less compared to the rate of change in *x*-*y* plane. However for the case of TSDP, errors are significant as depicted in figure 5.10. Consequently it cannot be used to approximate *x*-*y* planar positions of nodes within 3D space by using 2D optimizations.

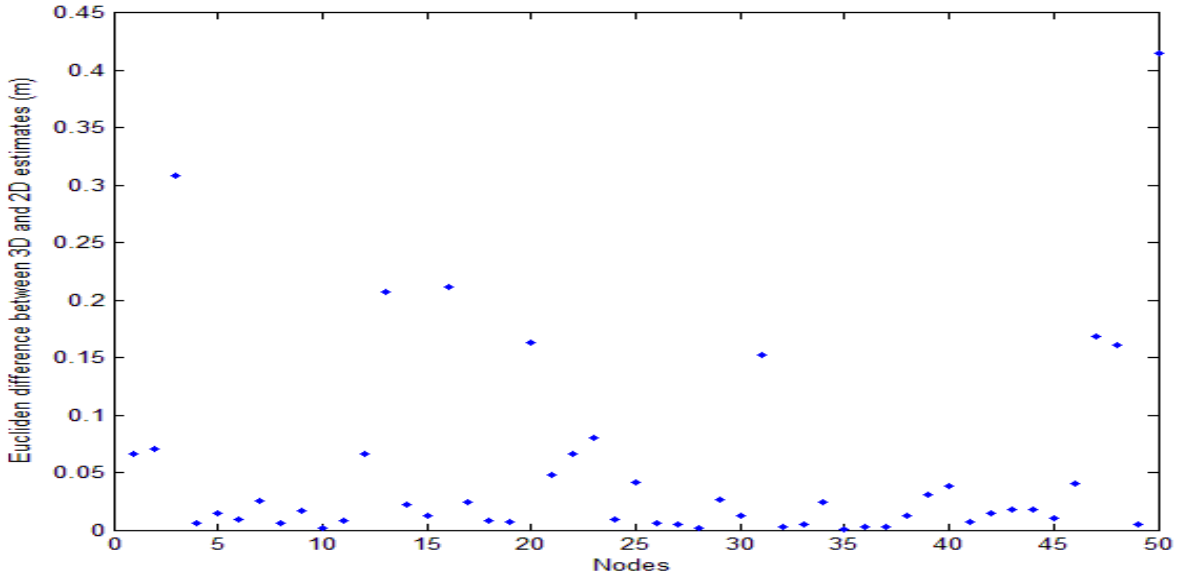


Figure 5.9: Euclidean distance between 3D and its 2D approximate estimates of 50 nodes in x-y plane by using DSDP technique

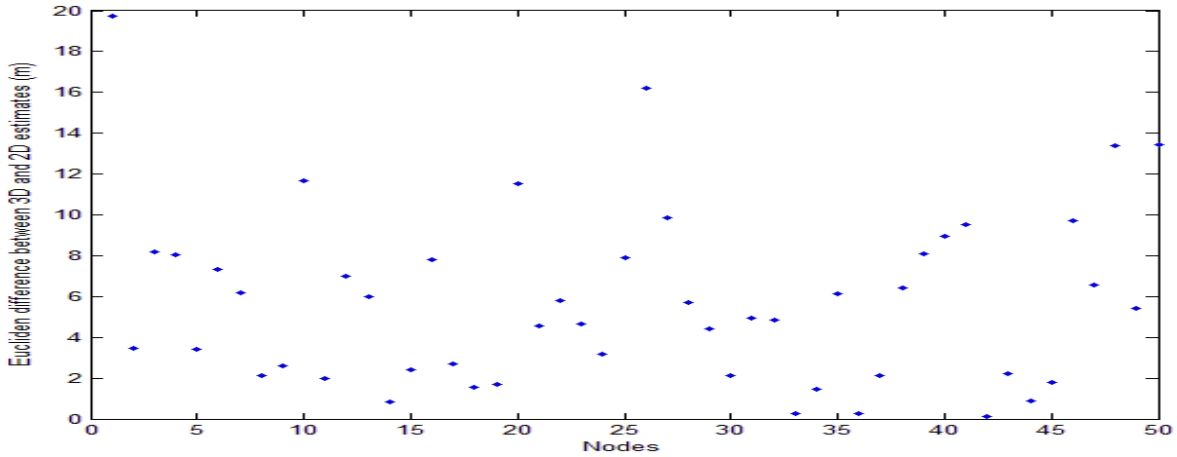


Figure 5.10: Euclidean distance between 3D and its 2D approximate estimates of 50 nodes in x-y plane by using TSDP technique.

5.4 Analysis of 3D Localization

Simulation results obtained for 3D localization show that accuracy of position estimates in *x-y* plane is comparable to that of 2D localization, while estimates in vertical position (*z-*

axis) are coarse. Therefore coarse height estimates can only be used to identify floor of a node for the case of a multi-storey building.

In addition to this, simulated results of DSDP and TSDP optimizations show that in 3D localization, estimates in *x-y* plane of a TSDP technique are better than DSDP. On the other hand height estimates of DSDP are more accurate compared to TSDP as given in Table 5.2. Overall 3D CEP values obtained show DSDP to be more accurate, but with large percentage of errors in floor identification compared to TSDP. The number of anchors and their locations also has impact on accuracy in height estimates.

We have also simulated a 3D localization problem in 2D to obtain only position estimates in *x-y* plane. Distance measurements are generated for 3D environment and are then used to obtain position estimates in *x-y* plane. Results show DSDP estimates node positions with an error of 0.4m compared to 3D estimates of *x-y* positions, whereas TSDP 3D to 2D approximations do not perform well and cannot be used.

We have developed TDM localization for 3D, exploiting 3D to 2D approximation if DSDP is to be used to reduce computational complexity.

Considering these observations from simulated results, a distributive 3D localization technique is developed. 3D localization technique works similar to 2D localization algorithm in which nodes are identified as mobile and stationary; furthermore mobile nodes are more frequently localized compared to stationary nodes. In order to classify nodes as stationary and mobile, nodes are localized whereas in SPACELOC [17] nodes are classified based on variations in measured distances. In WLANs, coverage range is larger than sensor networks and errors in distance measurements are function of distances. Therefore, it is not possible to identify mobile nodes by just observing changes in measured distances between two time intervals as is done in SPACELOC for sensor networks.

Although accuracy in identifying stationary and mobile nodes is much greater for the case where identification is done after localization using optimization technique, but there is a disadvantage of computational complexity. For the case of 2D localization, we have developed a TDM based localization technique where stationary nodes are localized less frequently as compared to mobile nodes. In reality, nodes exist in 3D environment; therefore localization must be achieved in 3D space. To make localization process computationally efficient and scalable, a 3D TDM distributive technique is developed and is explained in section 5.6.

5.5 Scalability versus Speed

Similar to the section 4.6, we have found the average speed required to localize nodes in 3D by using DSDP and TSDP techniques for the nodes increasing from 5 to 60. The resultant average CPU time to localize nodes is tabulated in table 5.2. In order to compare CPU time of 2D and 3D localization, table 6.2 is also generated. From table 5.3 it can be deduced that slot duration to localize 40 nodes using DSDP has to be increased from 1.5 seconds to 2 seconds. Similarly to localize 20 nodes in 3D using TSDP with two iterations, slot duration has to be increased to 3.5 seconds.

Table 5.2: Effect of number of nodes on computational speed

Total Number of nodes = Unknown Nodes + Anchors	TSDP Average CPU Time in Seconds (0 iterations)	TSDP Average CPU Time in Seconds (1 iterations)	TSDP Average CPU Time in Seconds (2 iterations)	DSDP Average CPU Time in Seconds
5+4	0.3837	0.6746	1.0886	0.2425
10+4	0.5183	1.0933	1.5817	0.3991
15+4	0.6623	1.5889	2.5131	0.5612
20+4	0.9486	2.3228	3.4636	0.7554
25+4	1.1808	3.1405	5.0887	0.9263
30+4	1.6154	4.6550	7.2641	1.1914
40+4	2.8360	9.0679	14.3421	1.7961
50+4	5.2117	22.9168	40.2334	3.1258
60+4	7.8146	59.1901	100.1660	6.1965

Table 5.3: Comparison of a computational speed of 2D and 3D localization

Total Number of nodes = Unknown Nodes + Anchors	TSDP Average CPU Time in Seconds (0 iterations) 2D	TSDP Average CPU Time in Seconds (0 iterations) 3D	DSDP Average CPU Time in Seconds 2D	DSDP Average CPU Time in Seconds 3D
5+4	0.3011	0.3837	0.2126	0.2425
10+4	0.3249	0.5183	0.3288	0.3991
15+4	0.4325	0.6623	0.4314	0.5612
20+4	0.5594	0.9486	0.5614	0.7554
25+4	0.7135	1.1808	0.7008	0.9263
30+4	0.98	1.6154	0.9	1.1914
40+4	1.5850	2.8360	1.30	1.7961
50+4	2.6861	5.2117	1.9657	3.1258
60+4	4.4957	7.8146	2.9417	6.1965

5.6 Time Division Multiplexed Distributive Localization

We have developed two separate distributive localization techniques for TSDP and DSDP optimizations. First of all DSDP based technique is explained.

5.6.1DSDP Based 3D Localization

As seen from tables 5.2 and 5.3, 3D localization is more computationally demanding compared to 2D localization. One method to reduce computation complexity for 3D localization can be based on exploiting the rate with which a mobile node user will be changing floors compared to its planar *x-y* coordinate changes in its position. For a mobile user rate of change in *x-y* coordinate will always be greater than a change in floors. Based on this we can reduce the localization time by applying alternative technique to a time division multiplexed distributive algorithm. In this technique, not all the time localization is performed in 3D instead subnets are localized in 3D to 2D ratio. Each subnet is localized within frame(s) once in a 3D space then in a 2D space in an alternative way, if DSDP is to be used. When nodes are localized in 2D coordinates, their 3rd dimension estimates are retained from previously estimated values. With this alternative method, overall localization speed can be improved to track the mobile nodes in real time applications. Figure 5.11 depicts this 3D TDM distributive technique for a network with one mobile and one stationary subnet. When nodes are localized in frame 2 only, *x-y* coordinates of the node positions are estimated, their 3rd dimension estimates i.e. *z*-coordinates are kept same to that of frame 1.

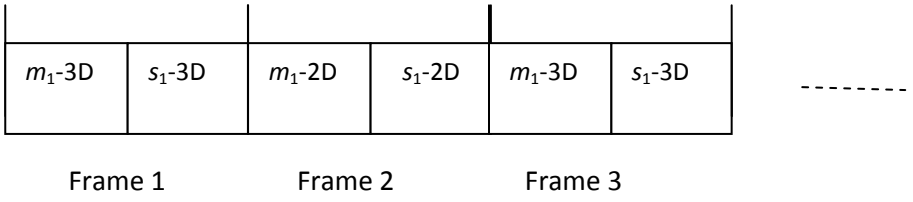


Figure 5.11: Frame structure of subnets for a proposed 3D TDM technique.

Complexity can be further reduced if 3D to 2D approximations are used. In this case, all the subnets are localized initially in first two frames in 3D space. It is done for two purposes: to estimate node positions in *x-y* plane and floor identification and to identify

mobile and stationary nodes. Subnets are then localized in 1:n such that all the mobile subnets are localized in 3D while stationary subnets are localized in 2D. Subsequently stationary and mobile node identifications are achieved by comparing only estimations in *x-y* plane.

5.6.2 TSDP Based 3D Localization

As we have seen in aforementioned simulations, TSDP based estimates in *x-y* plane are more accurate compared to DSDP based technique. However, 2D approximation of a 3D problem using TSDP is not feasible. Therefore, distributive technique developed for DSDP cannot be used. Hence, we have developed a novel distributive technique of localization if TSDP is to be used. In this case, initially all the nodes are localized by creating subnets if number of nodes are large and then localized using 3D localization. Then subnets are localized once again in 3D. This is done to identify mobile and stationary nodes. After this, all the mobile subnets are localized in 3D in every frame while stationary nodes are localized less frequently in 2D space. The 2D estimates of stationary nodes are not used to determine estimates of positions but just to detect change in status of the node, while position estimates of stationary nodes are obtained from previous 3D estimates only.

5.7 Summary

In this chapter we have analyzed 3D localization and deduced that performance of floor identification increases by increasing number of anchors. Simulations results show that CEP in three dimensions are not equal, that is CEP values in *x-z* coordinates representing heights is much greater than CEP values in *x-y* coordinates representing a ground. We have also presented a time division multiplexed technique for a 3D localization in which 3rd coordinate position estimations are performed at a slower rate compared to 2D position estimations. , It is based on fact that mobile nodes will not be changing floors any faster than their positions in *x-y* coordinates.

Chapter 6

Conclusions and Future Research

In this thesis, we have developed a time division multiplexed distributive localization algorithm based on convex optimizations for WLANs. This chapter summarizes the developed algorithm and discusses future research directions.

6.1 Contributions

The proposed algorithm is a distributive approach in localizing WLANs using convex optimizations. Although convex optimizations are more accurate as compared to other techniques such as least square, but are computationally inefficient for a network with large number of nodes.

We have proposed localization algorithm for WLANs planned and optimized for providing multimedia services and requiring no extra hardware or relocation of access points for providing additional localization service.

A lot of existing work done in indoor localization is for sensor networks, which are different for WLANs in terms of node densities, mobility and operational range. Hence those techniques do not provide optimum solution for WLANs. We have optimized a localization technique for WLANs having diverse node densities with different mobility levels.

In order to apply a convex optimization to WLANs with large number of nodes, we proposed to divide a larger network into smaller networks known as subnets with computationally manageable sizes. Within WLANs, not all the nodes are mobile all the time. Therefore, locations of all the nodes will not be changing almost of the time; we

have exploited this characteristics of WLANs to further optimize the localization process by classifying nodes as mobile and stationary. Mobile nodes are more frequently localized compared to stationary nodes whose positions are not changing, resulting in a localization solution that can track mobile nodes in real time. With this technique, total number of nodes to be localized is less thus the computational complexity is reduced.

To make a localization algorithm scalable we have developed a time division multiplexed based scheduling technique in which subnets are localized one after another by assigning a time slot to each subnet. In this technique, time axis is divided into frames of fixed duration; frame is further divided into slots such that within slot duration, a subnet can be localized. The frame duration and number of slots is a function of total number of mobile and stationary nodes present within the network. For a dynamic network with changing number of nodes, frame size varies resulting in a highly scalable localization solution. The duration of slots depends on type of localization technique used. We have found the semi definite programming technique minimizing distance square errors given in equation 2.13 to be more computationally efficient as compared to iterative trace minimization technique given in equation 2.12 a , b. One of the drawbacks of distance square error based technique is a high rank realization for the subnets consisting of cluster of nodes distributed very close to each other or subnets with nodes present at outside the parameter of anchors. One method to circumvent this is by adding randomly placed virtual nodes to subnets, which results in performance improvements.

Scheduling of subnets within frames is based on the type of a subnet:, mobile subnets are given more time slots compared to stationary subnets. We have set a ratio of mobile and stationary node scheduling within frames as 1:n, which means that within n number of frames all mobile subnets will be localized n times, while all stationary node, if present, will be localized once. With this approach, only mobile nodes will be more frequently localized as compared to stationary nodes. Our algorithm works by an incremental approach in which only portion of the nodes present within each slot is localized at a time

and a network is localized incrementally by localizing the nodes present within a subnet in a time division multiplexed way.

Our algorithm is always traceable; we can compute the delay parameter, the time required to re-localize a node, or equivalently a maximum frequency with which a node can be localized, an important parameter for real time node tracking.

The accuracy of the localization depends on the measured distances and its upper bounds can be determined by using CEP parameter given in equation 3.33.

We have also analyzed a 3D localization and found that errors in determining the heights of the nodes are much greater compared to estimations along the surface of the floors However it is possible to estimate a floor in which node exists in a multi-storey building, an estimate which is sufficient for most of the applications. We have also proposed a TDM distributive algorithm for 3D localization where the computational complexity is reduced by localizing the nodes in 3D to 2D ratio. During 2D localization, 3^{rd} dimension estimates are kept from previous estimates. It is based on the fact that the rate of change of floor is very less compared to the changes along the ground plane.

6.2 Future Research

Some of the future directions in WLAN localization are highlighted below:

WLANs are deployed by optimizing coverage and throughput for providing the bandwidth hungry multimedia services, we have proposed a localization technique for such a network. We have also found that proper placement of anchors which we assumed as access points can improve position estimations which are normally placed only to optimized throughput and coverage. The developed localization technique assumes proper placement of anchors for optimized throughput and coverage; a joint optimizations of anchor positions for coverage, throughput and localization can be a future area of research.

Our distributive algorithm is based on nature of WLANs, further research work can be done with reference to statistical characteristics of node densities and node mobility to further optimize the localization algorithm.

We have found coarse position estimations in 3D environments and developed a technique that identifies the discrete floor only. Further work may be carried out by applying hybrid-solution of convex and non convex techniques like least square, where convex techniques can be used as initial starting point for least square estimations for achieving accuracy in 3D.

In addition to above the developed technique can be extended for cellular wide area networks having diverse nature of node mobility from stationary users to users travelling at high speeds.

Sum Of Square (SOS) optimization method has been proposed for sensor network localization [107], performance is demonstrated to be promising. This technique can be investigated to be applied for WLAN localization. More specifically SOS can be analysed for 3D localization.

References

[1] T. Budinger, “Biomonitoring with wireless communications”, Annu. Rev. Biomed. Eng., vol. 5, no. 1, pp. 383–412, Aug. 2003.

[2] J. Caffery and G. Stuber, “Radio location in urban CDMA microcells,” in Proc. IEEE Int. Symp. Personal, Indoor Mobile Radio Commun., Toronto, ON, Canada, Sep. 1995, vol. 2,pp. 858–862.

[3] J. Warrior, E. McHenry, and K. McGee,” They know where you are-location detection,” IEEE Spectr., vol. 40, pp. 20–25, Jul. 2003.

[4] Z Zhu, A M-C So, Y Ye, “Universal Rigidity: Towards Accurate and Efficient Localization of Wireless Networks’, IEEE INFOCOM - INFOCOM , pp. 2312-2320, 2010

[5] K. Kaemarungsi, ”Design of Indoor Positioning Systems Based on Location Fingerprinting Technique”, PhD thesis, University of Pittsburgh, 2005.

[6] G. Sun; J. Chen, W. Guo and K.J.R Liu “Signal processing techniques in network-aided positioning: a survey of state-of-the-art positioning designs” IEEE, Signal Processing Magazine, 2005.

[7] G. Mao, B. Fidan, B D. O. Anderson, ‘Wireless sensor network localization techniques’, Computer Networks: The International Journal of Computer and Telecommunications Networking Volume 51 , Issue 10 , July 2007.

[8] A M-C So, Y Ye, 'Theory of Semidefinite Programming for Sensor Network Localization'. Mathematical Programming, Series B (2007) 109:367-384.

[9] J. Aspnes, T. Eren, D. Goldenberg, A. Morse, W. Whiteley, Y. Yang,B. Anderson, and P. Belhumeur, 'A theory of network localization," IEEE TRANSACTIONS ON MOBILE COMPUTING, VOL. 5, NO. 12, December 2006.

[10] B. Hendrickson, 'Conditions for unique graph realizations', SIAM Journal on Computing, v.21 n.1, p.65-84, Feb. 1992.

[11] J. Aspnes, D. Goldenberg, and Y.R. Yang, 'On the Computational Complexity of Sensor Network Localization,' Proc. First Int'l Workshop Algorithmic Aspects of Wireless Sensor Networks, July 2004.

[12] T. Eren, and et al., Rigidity, Computation, and Randomization in Network Localization, *IEEE INFOCOM 2004,* Hong Kong, March 2004.

[13] Z. Zhu, A. M.-C. So, and Y. Ye, "Universal Rigidity and Edge Sparsification for Sensor Network Localization," 29th IEEE Conference on Computer Communications., March**,** 2010**.**

[14] P. Biswas and Y. Ye, "A distributed method for solving semidefinite programs arising from ad hoc wireless sensor network localization,". Technical report, Dept of Management Science and Engineering, Stanford University, October 2003.

[15] S. Kim, M. Kojima, and H. WakiI. 'Exploiting sparsity in SDP relaxation for sensor network localization'. SIAM J. Optim., 20(1):192–215, 2009.

[16] P. Biswas, T.-C. Liang, K.-C. Toh, T.-C. Wang, and Y. Ye. “Semidefinite programming approaches for sensor network localization with noisy distance measurements,” IEEE Transactions on Automation Science and Engineering, 2006.

[17] M. W. Carter, H. H. Jin, M. A. Saunders, and Y. Ye, “SPACELOC: An adaptive subproblem algorithm for scalable wireless sensor network localization,” *SIAM J. OPTIM*, pp. 1102–1128, 2006.

[18] A. U. Khan, and M. Al. Akaidi, ”A Distributive Algorithm for WLAN Localization”, IEEE, ICET 6th Int conference, 2010.

[19] K. Pahlavan, Xinrong Li, J. P. Makela, ”Indoor geolocation science and technology” IEEE Communications Magazine, Vol. 40, No. 2. (Feb 2002), pp. 112-118.

[20] B. Schulze, Combinatorial and geometric rigidity with symmetry constraints, PhD thesis, YORK UNIVERSITY, TORONTO, ONTARIO, MAY 2009.

[21] R Connelly, “Generic Global Rigidity,” Springer-Verlag, Discrete Comput. Geom., Volume 33, Number 4, April 2005

[22] R. Connelly, T. Jordan, and W.J. Whiteley**, “**Generic Global Rigidity of Body-Bar Frameworks”, Technical report, TR-2009-13. Published by the Egervary Research Group, 2009.

[23] A. Singer and M. Cucuringu, “Uniqueness of Low-Rank Matrix Completion by Rigidity Theory,”SIAM J. MATRIX ANAL. APPL. Vol 31, No 4 pp, 1621-1641, 2010.

[24] T. Eren, D. Goldenberg, W. Whiteley, Y.R. Yang, A.S. Morse, B.D.O. Anderson, P.N. Belhumeur, “Rigidity, computation, and randomization in network localization”,

Proceedings of the International Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM), March 2004.

[25] J Aspnes, T Eren, D K. Goldenberg, A. S Morse, W Whiteley, Y R Yang, B. D.O. Anderson, and P N. Belhumeur, “A Theory of Network Localization,” IEEE TRANSACTIONS ON MOBILE COMPUTING, VOL. 5, NO. 12, DECEMBER 2006

[26] B. Hendrickson “Conditions for Unique Graph Realizations”, SIAM J. COMPUT. Vol. 21, No. 1, pp. 65-84, February 1992.

[27] Y. Shang, W. Ruml and m. P. J. Fromherz,” Positioning using local maps”, Elsevier, ADHOC 109, 2004.

[28] Y. Shang and W. Ruml,” Improved MDS-Based localization”, IEEE INFOCOM, 2004.

[29] A. A. Ahmed, Y. Shang and H. Shi,” MDS-Based methods for an ad hoc network localization”, Journal of interconnection Networks, vol 7, pp 5-19, 2006.

[30] M. Cao, B. D. O. Anderson, and A. S. Morse,” Sensor network localization with imprecise distances”, ELSEVIER, Sys & cont letters, 2006.

[31] M. Cao, B. D. O. Anderson, and A. S. Morse, "Localization with imprecise distance information in sensor networks," in Proc. 44th IEEE Conf. Decision Control and European Control Conf. 2005.

[32] D. Niculescu and B. nath,” Ad hoc positioning system,” IEEE, GLOBECOM, 2001.

[33] W. H. FOY, “Position-Location Solutions by Taylor-Series Estimation,” IEEE Trans on Aerospace and Elect Sys VOL. AES-12, NO. 2 March 1976.

[34] D. J. TORRIERI, “Statistical theory of passive location systems,” IEEE Trans on Aerospace and Elect Sys VOL. AES-20, NO. 2 MARCH 1984.

[35] N. Khajehnouri and A. H. Sayed, “A Non-Line-of-Sight Equalization Scheme for Wireless Cellular Location”, IEEE, ICASSP 2003.

[36] W.Gander, “Least Squares with a Quadratic Constraint”, Springer-Verlag Numer. Math. 36, 291-307 , 1981.

[37] G. GOLUB, ”Numerical Methods for Solving Linear Least Squares Problems”, Numerische Mathematik 7, 206--21 6, 1965.

[38] S. M. Kay, Fundamentals of Statistical Signal Processing: Estimation Theory, Prentice Hall, 1990.

[39] N. Aslandi, Indoor Cooperative Localization for Ultra Wideband Wireless Sensor Networks, PhD thesis, WORCESTER POLYTECHNIC INSTITUTE, 2008.

[40] A H. Sayed, A. Tarighat, and N. Khajehnouri ,”Network-Based Wireless Location”, IEEE Signal Processing Magazine JULY 2005.

[41] L. Doherty, K. S. J. Pister, and L. El Ghaoui, ”Convex Position Estimation in Wireless Sensor Networks”, Proc. IEEE in INFOCOM 2001.

[42] S. KIM, M. Kojima, and H. Waki, ”Exploiting sparsity in SDP relaxation for sensor network localization. SIAM J. Optim., 20(1):192–215, 2009.

[43] Z. Wang, S. Zheng, S. Boyd, and Y. Ye, ”Further relaxations of the semidefinite programming approach to sensor network localization”, SIAM J. Optim., pp.655–673, 2008.

[44] J. Nie, “Sum of squares method for sensor network localization”, Comput. Optim. Appl., 2009.

[45] Y. Ding, N. Krislock, J. Qian, and H. Wolkowicz,” Sensor network localization, Euclidean distance matrix completions, and graph realization”, Proc of MELT08, San Francisco, pages 129–134, 2008.

[46] A W. Harrow, Lecture Notes 15, @ http://www.maths.bris.ac.uk/~csawh/

[47] L. Vandenberghe and S. BOYD,“Semidefinite Programming”, SIAM REVIEW Vol. 38, No. 1, pp. 49-95, March 1996.

[48] S. Al-Homidan and H. Wolkowicz, ”Approximate and exact completion problems for Euclidean distance matrices using semidefinite programming, Elsevier, Linear Algebra and its Applications, vol 406 pp 109-141, 2005.

[49] A.Y Alfakih, and A Khandani “Solving Euclidean distance matrix completion problems via semidefinite programming”, Comput Optim Appl vol 12, no 1-3 pp 13-30, 1999.

[50] J. Dattorro, Convex Optimization & Euclidean Distance Geometry, Meboo Publishing, 2011.

[51] J.C Gower, “Euclidean Distance Geometry”, Elsevier, Linear Algebra and its Applications, 1982.

[52] S. Boyd and L. Vandenberghe, Convex Optimization, Cambridge University Press, March 2004.

[53] R Balaji, “On Euclidean distance matrices**”** Elsevier, Linear Algebra and its Applications, 2007.

[54] B Recht, and M Fazel, P. A. Parrilo “Guaranteed Minimum-Rank Solutions of Linear Matrix Equations via Nuclear Norm Minimization”, preprint, 2007 - arxiv.org.

[55] EJ Candès,” Exact matrix completion via convex optimization”, Springer, Foundations of Computational Mathematics, 2009

[56] Y. Kim and M. Mesbahi,” On the rank minimization problem”, Proc American Control Conf, 2004.

[57] P. Biswas and Y. Ye, "Semidefinite programming for ad hoc wireless sensor network localization," in Proceedings of the third international symposium on information processing in sensor networks, ACM press, 46-54 (2004).

[58] N. Alsindi, X Li, and K. Pahlavan, “Analysis of Time of Arrival Estimation Using Wideband Measurements of Indoor Radio Propagations”, IEEE Trans on Instrumentation and Measurement, vol. 56, no. 5, October 2007.

[59] K. Pahlavan, P. Krishnamurthy, and J. Beneat, “Wideband radio propagation modeling for indoor geolocation applications,” *IEEE Commun. Mag.*, vol. 36, no. 4, pp. 60–65, Apr. 1998.

[60] S. J. Howard and K. Pahlavan, “Measurement and analysis of the indoor radio channel in the frequency domain,” *IEEE Trans. Instrum. Meas.*,vol. 39, no. 5, pp. 751–755, Oct. 1990.

[61] B. Alavi, and K. Pahlavan, “Modeling of the TOA-based Distance Measurement Error Using UWB Indoor Radio Measurements”, IEEE Communications Letters, VOL. 10, NO. 4, April 2006.

[62] Jourdan, D. Dardari, D. Win, M. “Position error bound for UWB localization in dense cluttered environments’ IEEE Transactions on Aerospace and Electronic Systems, vol 44, No 2, April 2008, pp 613-626.

[63] Y. Qi and H. Kobayashi, “On Geolocation Accuracy with Prior Information in Non-line-of-sight Environment’, Proc. IEEE VTC, pp 285-288, 2002.

[64] I Güvenç, C. C. Chong, F Watanabe, and H Inamura, “ NLOS Identification and Weighted Least-Squares Localization for UWB Systems Using Multipath Channel Statistics”, EURASIP Journal on Advances in Signal Processing Volume 2008.

[65] B. Roth,” Rigid and flexible frameworks”, Amer. Math. Monthly, 88 (1981), pp. 6-21.

[66] J. C. Gower, “Properties of Euclidean and Non-Euclidean Distance Matrices”, Elsevier, Linear Algebra and its Applications 1985.

[67] W. Nelson,”Use of Circular Error Probability in Target Detection”, ESD-TR-88109, Hanscom Air Force Base, Massachusetts,May 1988.

[68] Koorapaty, H. Grubeck, H. Cedervall, M. **"**Effect of biased measurement errors on accuracy of position location methods," Proc IEEE, GLOBECOM 98, Vol: 3, pp 1497-1502, 1998.

[69] M. Gavish, E. FOGEL, "Effect of Bias on Target Location Bearing-Only" IEEE Trans on Aerospace and Electronic Sys vol. 26, no. 1 January 1990.

[70] E. G. Larsson. "Cramer-Rao bound analysis of distributed positioning in sensor networks,' IEEE Signal Processing Letters, vol 11, no 3, pp 334-337, March 2004.

[71] X.Li, K. Pahlavan, and J. Beneat, "Performance of TOA estimation algorithms in different indoor multipath conditions", IEEE, Wireless Communications and Networking Conference, July 2004.

[72] D. Koks , "Numerical Calculations for Passive Geolocation Scenarios", Australian Government , Dept of Defence, DSTO–RR–0319, May, 2007.

[73] Jourdan, D. Dardari, D. Win, M. "Position error bound for UWB localization in dense cluttered environments' IEEE Transactions on Aerospace and Electronic Systems, vol 44, No 2, April 2008, pp 613-626.

[74] F. Gustafsson and F. Gunnarsson "Mobile positioning using wireless networks: possibilities and fundamental limitations based on available wireless network measurements", IEEE, Signal Processing Magazine, vol 22, pp41-53, 2005.

[75] Y. Qi and H. Kobayashi, "On Geolocation Accuracy with Prior Information in Non-line-of-sight Environment', Proc. IEEE VTC, 2002, pp 285-288.

[76] Y. Qi and H. Kobayashi. "A unified analysis for Cramer-Rao Lower Bound for geolocation," The Proceeding of the 36th Annual Conference on Information Sciences and Systems (CISS 2002), Princeton University, March 2002.

[77] Y. Qi and H. Kobayashi. “Cramer-Rao lower bound for geolocation in non-line-of-sight environment,” In Proceedings of the International Conference on Acoustics, Speech and Signal Processing, Orlando, FL, May 2002, 2473-2476.

[78] H. L. Van Trees, Detection, estimation and modulation theory, Part 1, John Wiley & Sons, Inc., 2001.

[79] Chang, H J, and Wu, C H. “On Sample Size in Using Central Limit Theorem for Gamma Distribution, ”International Journal of Information and Management Sciences (IJIMS), Volume 19, Number 1, 2008, pp. 153-174.

[80] M. E. Shultz,”Circular Error Probability of a Quantity Affected by a Bias”, USAF Aeronautical Chart and Information Center Chart Research Division Gaeophysical and Space Sciences Branch, June 1963.

[81] M. Evans, Z. Govindarajulu, and J. Barthoutlot, “Estimates of circular error probabilities”, Dept of Stats, Stanford university, Technical Report 367, 1985.

[82] J. T. Gillis, “Computation of the circular error probability integral” Harvard university, technical research report, 1991.

[83] H. Wymeersch, J. Lien, and M. Z. Win, “Cooperative Localization in Wireless Networks”, Proceedings of the IEEE Vol. 97, No. 2, February 2009

[84] http://cvxr.com/

[85] R. Mautz, W. Ochieng, G. Brodin, and A. Kemp , ‘3DWireless Network Localization from Inconsistent Distance Observations,” Ad Hoc & Sensor Wireless Networks – AHSWN, vol. 3, no. 2-3, pp. 141-170, 2007.

[86] R. T. Rockafellar, Convex Analysis, Princeton Landmarks in Mathematics,1972.

[87] *L. Vandenberghe and S. Boyd, "Semidefinite Programming", SIAM Review*, 38(1): 49-95, March 1996.

[88] Hongyang Chen, K. Sezaki, P. Deng, and H. C. So, "An improved DV-hop localization algorithm for wireless sensor networks," *Proc. IEEE Conference on Industrial Electronics and Applications (ICIEA 2008), pp.1557-1561, June 2008, Singapore.*

[89] X. Bao, F. Bao S. Zhang L. Liu," An Improved DV-Hop Localization Algorithm for Wireless Sensor Networks," IEEE- WiCOM, 6th International Conference, 2010.

[90] D. Moore, J. Leonard, D. Rus, S. Teller, "Robust distributed network localization with noisy range measurements," ACM, Conference On Embedded Networked Sensor Systems - SenSys , pp. 50-61, 2004.

[91] K. pahlavan and X. Li, "Indoor Geolocation Science and Technology," IEEE Comm magazine, pp 112-118, Feb, 2002.

[92] R. W. Ouyang, A.K-S. Wong, C-T.Lea , " Received Signal Strength-Based Wireless Localization via Semidefinite Programming: Noncooperative and Cooperative Schemes," IEEE Vehicular Technology Society, vol 59, No 3, March 2010.

[93] C-Y. Shih and P. J. Marron, "COLA: Complexity-Reduced Trilateration Approach for 3D Localization in Wireless Sensor Networks,", 2010 Fourth International Conference on Sensor Technologies and Applications, 2010.

[94] N. Dou, G. Bo, Z. Wei and Y. Daquan, "3D localization method based on MDS-RSSI in wireless sensor network," IEEE International Conference on Intelligent Computing and Intelligent Systems (ICIS), December, 2010.

[95] T. Wang, "Ranging Energy Optimization for a TDOA-Based Distributed Robust Sensor Positioning System", Hindawi, International Journal of Distributed Sensor Networks, Vol 2010.

[96] K.C.Y. Shum, Q.J. Cheng, J.K.Y. Ng and D. Ng, "A signal Strength based Location Estimation Algorithm within a Wireless Network", IEEE Comp Soc, 2011 Int. conf on advanced Info Networking and applications, 2011

[97] M. Cypriani, F. Lassabe, P. Canalda, and F Spies, “Wi-Fi-Based Indoor Positioning: Basic Techniques, Hybrid Algorithms and Open Software Platform”, IEEE Int Conf on Indoor Positioning and Indoor Navigation (IPIN), 15-17 SEPTEMBER 2010.

[98] S. Mazuelas, A. Bahillo, R. M. Lorenzo, P. Fernandez, F. A. Lago, E. Garcia, J. Blas, and E. J. Abril, “Robust Indoor Positioning Provided by Real-Time RSSI Values in Unmodified WLAN Networks”, IEEE Journal of Selected Topics in Signal Processing, vol. 3, no. 5, October 2009.

[99] S.H. Fang and T.N.Lin, “Accurate WLAN Indoor Localization Based on RSS Fluctuations Modeling”, WISP 2009, 6th Int Symp on Intelligent Signal Proc, August 2009.

[100] C.B. Lim, S.H. Kang, H.H. Cho, S.W. Park and J. G. Park, “An Enhanced Indoor Localization algorithm Based on IEEE 802.11 WLAN using RSSI and multiple parameters”, IEEE Comp Soc, 2010.

[101] F. Gu, J.Shang, G. Zheng and L. Zhu, “An improved fingerprinting method for localization WLAN-Based”, IEEE, International Conference on Computer Science and Service System (CSSS), 2011.

[102] G. Lui, T. Gallagher, B. Li, A. G. Dempster and C. Rizos, ”Differences in RSSI readings made by different Wi-Fi chipsets: A limitation of WLAN localization”, IEEE International Conference on Localization and GNSS (ICL-GNSS), 2011

[103] H. K. Lee, H. S. Kim, and W. S. Choi, ”Modeling heterogeneous signal strength characteristics for flexible WLAN indoor localization”, ICROS-SICE Int Joint Conference, Japan, 2009.

[104] C.H. Lim, Y. Wan, B.P. Ng, and C.M. S.See, “A real time indoor WiFi localization system utilizing smart antennas”, IEEE Trans on Consumer Electronics, vol. 53, No. 2, May 2007.

[105] F. Izquierdo, M. Ciurana, F. Barcclo, J. Paradells and E. Zola, ”Performance evaluation of TOA-based trilateration method to locate terminals in WLAN”, IEEE 1st International Symposium on Wireless Pervasive Computing, 2006.

[106] K-T Lay, and W-K Chao, “Mobile Positioning Based on TOA/TSOA/TDOA Measurements with NLOS Error Reduction”, IEEE Proceedings of International Symposium on Intelligent Signal Processing and Communication Systems, December 13-16, 2005.

[107] J. Nie, “Sum of Squares Method for Sensor Network Localization”, Computational Optimization and Applications, pp 151-179, Volume: 43, Issue: 2, 2006.