DYNAMIC LOCALIZATION OF MULTIPLE MOBILE SUBJECTS IN WIRELESS ADHOC NETWORKS

By

RAVI KIRAN GARIMELLA

Bachelor of Technology in Electrical and

Electronics Engineering

Jawaharlal Nehru Technological University

Hyderabad, Andhra Pradesh, India

2004

Submitted to the Faculty of the
Graduate College of the
Oklahoma State University
in partial fulfillment of
the requirements for
the Degree of
MASTER OF SCIENCE
December 2007

DYNAMIC LOCALIZATION OF MULTIPLE MOBILE

SUBJECTS IN WIRELESS ADHOC

NETWORKS

Thesis Approved:

Dr. Weihua Sheng

Thesis Adviser

Dr. R Ramakumar

Dr. Qi Cheng

Dr. A. Gordon Emslie

Dean of the Graduate College

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER I

INTRODUCTION TO LOCALIZATION TECHNIQUES

1.1 Motivation

Localization, in simple words, is defined as the technique through which location awareness is made available for wireless subjects. It is a research topic of growing interest owing to the overwhelming progress achieved in the field of wireless applications over the past few decades. Apart from the technical advancements, the entry of cost-effective wireless devices into consumer markets have necessitated researchers to find out better localization techniques to closely locate the wireless subjects in their respective deployment region.

Tracking soldiers in a battlefield is one such application where the soldiers' locations are found out to issue favorable commands. Livestock tracking [1, 2], which is in implementation stage, is another contemporary application that has indicated the necessity of tracking cattle to make enhancements in farm management practices. While the localization of stationery subjects was dealt even before many decades, the ability to track the motion of mobile subjects in a wireless network is more sought after in recent applications.

Some of the contemporary applications make this possible by incorporating embedded sensory systems in PDAs, cell phones and generally the subject possesses a wearable position tracking system (WePosT). WePosT is a collage of different types of sensors, each with a well defined purpose in localization process and a version of WePosT is currently being developed at the laboratory for Advanced Sensing, Computation and Control at Oklahoma State University.

Recent years have seen the growing interest in mobile sensor networks [3] where all or partial of the sensor nodes have motion capability endowed by robotic platforms. Tracking and self-localizing various types of moving objects has become an important research topic. With the knowledge of the new technology, one might suggest 'Global Positioning System' which acquired a place even in common man jargon. From both the technical view point and the application constraints, one realizes that this option is not feasible all the time. It could be either due to unavailability of Global Positioning System (GPS) signal to all subjects, or the financial overhead involved in equipping a GPS on the all the subjects. This primarily directs the focus onto less expensive and short ranged sensor systems which are gaining momentum in many applications owing to their ease of availability and deployment. Once the issue of sensor networks comes into picture, the next question will be regarding their localization procedure and this calls for a simple yet robust localization algorithm. This way, localization algorithms emerged to be a contemporary research topic.

It is imperative to understand that the localization of mobile subjects relies on many issues, network topology being the most important. The attributes like network mobility, number of subjects in network, connectivity, shape of the network topology and others play a vital role right from selection of sensor type to the localization algorithm to be used. Apart from the network topology, there are many other deciding issues like the number of anchors, outdoor/indoor deployment, radio range and the precision requirement.

This concludes that the efficiency parameters of a localization algorithm cannot be generalized for all the problems. In other words, the efficiency is always defined based on the application and our goal is then to find out the best localization procedure given the constraints on the deployment. An overview of recent works on localization is quite indicative of the vast scope of inter-disciplinary methodologies being employed.

This thesis targets at developing a localization algorithm that can be employed for a dynamic and sparse network wherein the mobile subjects are equipped with moderately accurate sensors. It also assumes that GPS may not be available to most of the subjects. For such an operating environment, very few efficient algorithms were developed in the recent years to the best of our knowledge. Some of them are briefed in the next section and as we shall explain they aren't suitable for some or other specific reasons. Hence the goal is to develop a reasonably efficient localization algorithm and such efforts remain the main motivation behind the formulation of this thesis "Dynamic Localization of Multiple Mobile Subjects in Wireless Adhoc Networks"

Our method will integrate short distance dead reckoning technique with Multidimensional Scaling (MDS) technique [4] to provide accurate location tracking. The short distance dead reckoning is enabled by a wearable position tracking (WePosT) system, which consists of a data processing unit (DPU) and a set of inertial sensor units (ISUs). The DPU and ISUs are compact, light weight, tag-like devices which can be worn by the subject to be tracked. For example, the DPU can be attached to the human belt or arm. The ISU uses inertial sensors (accelerometers, gyros) to collect the acceleration rate and angle velocity as well as a digital compass for heading calibration. The ISUs can be attached to human ankles or shoes where the motion of human body can be detected. The ISUs will communicate with the DPU using Zigbee protocol. In the DPU, multi-sensor fusion scheme [5] is used to correlate the sensing data from both feet to achieve improved dead reckoning accuracy. The prototypes of wearable position tracking system and Inertial Sensor Units are shown in Figure 1.

Figure 1: Prototypes of Wearable Position Tracking System (left) and Data Processing Unit (right) Inertial Sensor unit (Bottom Right)

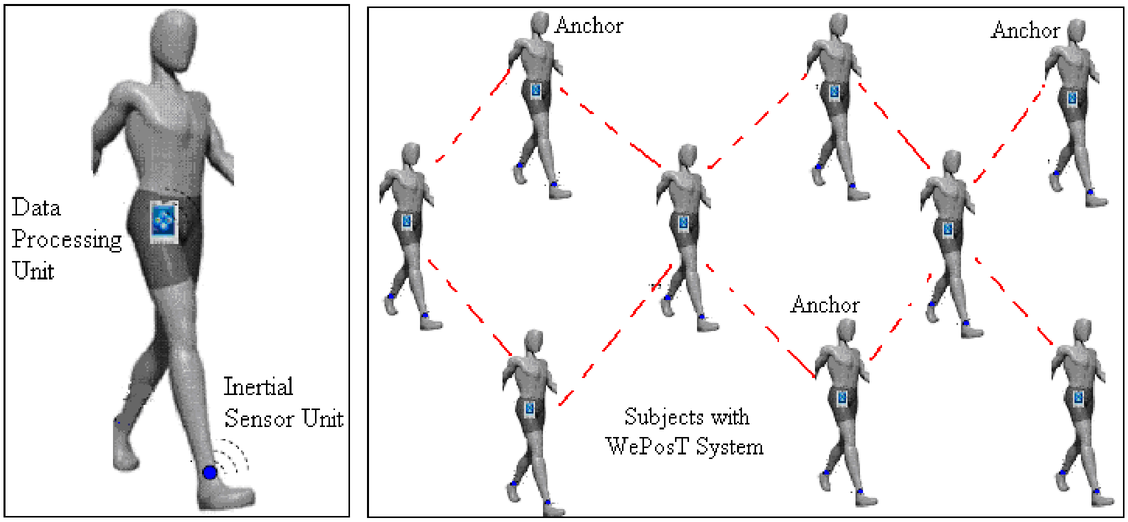

Figure 2: Mobile Subject Localization Problem using Wearable Position Tracking System (WePosT)

We assume the mobile network consists of n subjects. Among them only a small portion, m (m << n, for example, m = 5% of n) subjects, know their own locations. These subjects are called beacons. In the soldier tracking example, these beacons may be officers or military vehicles that are equipped with GPS or other localization techniques. With right sensors such as ultrasonic sensor, each subject can measure the distance to its neighbors through time difference of arrival (TDOA) technique [6].

As illustrated in Figure 2, the problem of multiple mobile subject localization is as follows: *Given a distance graph G =<S_u, S_a, D > where S_u = {x_1, x_2,… x_{n-m}} is a set of subjects in an s-dimensional space (s is 2 or 3), Sa = {x_{n-m+1}, x_{n-m+2},… x_n} is a set of m beacon subjects, D = [d_{ij}] is the distance matrix, find the n - m unknown locations S_u such that $|x_i-x_j|=d_{ij}$.*

Organization of the thesis: The thesis is organized in five chapters. The first chapter introduces the concept of localization and throws light on the need of having robust localization algorithms that would meet the requirements of sparse and dynamic wireless sensor networks. The contemporary localization techniques are also briefed in this chapter and important information pertaining to localization is enlisted.

The second chapter deals with Multidimensional Scaling based localization which forms the basis of this thesis. The theory and the mathematical modeling of Classical Multidimensional Scaling are explained at length and the application of MDS to localization problem is reviewed.

The third chapter comprises the main contribution of the thesis which is about the Dynamic Localization technique. The classical Multidimensional scaling algorithm is modified to suit the localization problem for a sparse and dynamic mobile sensor network. This version of MDS is referred to as Dynamic Multidimensional scaling. Dead reckoning based localization and the Dynamic MDS are fused together to get the final estimates. The methodology and the experimental results are presented to support the stand point.

The fourth chapter is a modification of the Dynamic Localization technique explained in the third chapter. The Dynamic MDS is analyzed to discover the variables which can asses the performance of the algorithm. Subsequently these variables are used in modifying the method in which the results of Dynamic MDS and the dead reckoning based localization are fused together to give the final estimate. This technique is validated through experimental simulations scenario. The thesis concludes with the chapter five where the contribution of thesis is revisited and the scope of further research is defined.

1.2 Related Work

1.2.1 Current Work on Sensor Localization

This section provides a brief overview of the contemporary localization techniques in vogue and subsequently focuses on the necessity of having MDS-based localization. In recent years, researchers have been developing different localization algorithms using triangulation, multilateration or other techniques, mainly for wireless sensor networks [7, 8, 9]. Localization techniques typically require some form of communication between reference points (nodes with known coordinates) and the receiver (node that needs to localize). Some examples of communication technologies are RF-based and acoustic based communication. The localization techniques are broadly classified into two categories: range-based and range-free. In range based techniques, information such as distances (or angles) of a receiver are computed for a number of references points using signal strength or timing based techniques and then position is computed. The current thesis belongs to this genre as it fundamentally requires all the pair wise distances for all the nodes deployed. As we shall explain in the later sections, we also need angle information to assist the MDS-based results with dead reckoning results.

The range-based techniques rely on a method of finding the physical distance between any two nodes in a network that are within communication range. This process is called ranging. There are two basic techniques used to perform ranging: received signal strength and signal propagation time. Received signal strength (RSS) is a way to do ranging by measuring the signal strength of a message at the receiver [8, 9] The receiver then uses knowledge of the sender's signal power (this might be contained within the message) to determine the power loss. Finally the receiver applies its known model for signal propagation behavior to convert the power loss to a distance, thus estimating how far away the sender is. This is an inaccurate technique. Radio signal propagation behavior is highly dependent on the environment (obstacles, signal fading, metals), and hence they are highly variable. Savvides et al [10] describes experiments that tried to get good results this way, but the results are unsatisfactory in most of the cases except for an extremely idealized one. In most real-world ad-hoc networks, ranging by received signal strength is not accurate.

The second method of ranging is possible by measuring the signal propagation time and converting it back to inter-distance with the knowledge of velocity of the signal transmitted. Time of arrival [11] is one such measure where the time taken for wireless signals (or packets) to travel from transmitter to receiver is multiplied by the velocity of signal (almost equal to light velocity) to obtain the inter-node distances. Radio signals travel at the speed of light (essentially instantaneous arrival), so it is not plausible to measure this time without using a high resolution clock to measure the time of flight. This is very commonly used in GPS-based ranging where the GPS receiver estimates distances using TOA from different satellites which needs time synchronization. Given the inter-distances, techniques like multilateration can be used to locate them. To avoid complex time synchronizations between the transmitter and receiver, we can consider return time of flight wherein the receiver retransmits the signal back to transmitter. The transmitter then calculates the TOA as half the return time of flight. But the TOA parameter is affected by latency in receiver response which may be due to processing queue at the receiver.

Time difference of arrival (TDOA) [6] is a variation of time of arrival and it is a preferred way of measuring distance by measuring the propagation time of signals. A sending node will transmit a radio signal and an ultrasonic signal at the same time. Because the radio signal arrives essentially instantaneously and the ultrasonic signal takes much longer, the receiver can measure the time difference between the arrivals, and thus deduce the traveled distance. The Cricket [12] system uses RF/US TdoA ranging. One problem with ultrasound signal propagation is that it is subject to multipath effects, and to variations with changes in the environment. It is desirable to recalibrate TDoA measurements according to these variations. Savvides et al. give a way to perform this calibration, given enough redundancy in the distance data. Some researchers have described the Ad-Hoc Localization System (AHLoS)[10], an iterative way of discovering the absolute position of every node in a network. They assumed an ad-hoc network, in which anchors that know their own location at any given time form some percentage of the nodes. The focus is on two-dimensional localization, and the ranging method is TDoA. Signal processing methods have been developed for localizing a set of static sensor nodes and analyzing the error properties [13, 14, 15], using both TDOA and angle of arrival (AOA) measurements where TOA measures the distances and the AOA tells about the orientation apart from positioning.

Apart from the above mentioned techniques, range-free techniques have also been used widely. An RF based proximity method was developed by [8], in which the location of a node is given as a centroid generated by counting the beacon signals transmitted by a set of beacons pre-positioned in a mesh pattern. Other methods that do not rely on range measurements were also developed. For example, the count of hops is used as an indication of the distance to the beacon nodes in some applications [7, 16]. But the majority of the applications rely on range based localizations.

Coming to the localization techniques, one of the most straightforward localization techniques is Global Positioning System (GPS) based localization that relies on multilateration technique using time of arrival of signals. It has been operative since early 1990's. For localization in an outdoor environment, GPS works extremely well. Unfortunately, the signal from the GPS satellites is too weak to penetrate most buildings, making GPS useless for indoor localization. Likewise it has many other shortcomings. Multipath effects, signal jamming delayed signals, and complex clock synchronization requirements and others have limited the usage of GPS to less applications. Adding to above, the GPS units are very expensive and this makes it almost useless in case of commercial applications where the overheads are mainly specified in terms of financial constraints. This shifted the focus towards less expensive, short ranged sensor network. In recent years, researchers have been developing different localization algorithms to localize these sensor networks.

However, most existing algorithms assume a static sensor network where the nodes do not move and require high node density [17]. Therefore these algorithms can not be used to track the subjects in the above examples, where the network is sparse and constantly changing.

With the above mentioned limitation on the contemporary localization procedures, the goal of this research is then to develop a novel tracking method for mobile subjects in sparse, dynamic wireless networks under the constraint that GPS may not be available to most of the subjects. It can be understood that limited work has been done on mobile sensor network self-localization. Tilak et al. [18] developed dynamic localization protocols for mobile sensor networks. However, their main interest is on how often the localization should be carried out in a mobile sensor network and not on the localization method itself. Recently, Hu and Evans [19] proposed sequential Monte Carlo (SMC) localization method to solve the localization problem and they found that the mobility of the sensors can be exploited to improve the accuracy and precision of the localization. Using a similar approach, simultaneous localization, calibration and tracking (SLAT) of mobile node within a set of static sensor nodes has been developed [20], where both the mobile node and the set of static sensor nodes are localized using range measurements.

As an attempt to design an algorithm that works well in localizing an adhoc network, an interdisciplinary algorithm called ‘Multidimensional scaling’ has been used [21]. But the deployment scenario assumes only static network with considerable node density. This thesis extends the application of Multidimensional Scaling based localization algorithm (with significant changes) to dynamic and sparse adhoc sensor networks.

CHAPTER II

MULTIDIMENSIONAL SCALING BASED LOCALIZATION

2.1 Introduction to Multidimensional Scaling

The roots of the Multidimensional Scaling[22, 23, 24] or MDS lie in the behavioral sciences like Psychometrics and Psychophysics wherein the personal traits of people are analyzed for important underlying distinctive characteristic features. Subsequently MDS proved to be an essential tool for many other researchers in diverse fields like Marketing, Sociology, Geography, and Psychology. Basically MDS is a data visualization algorithm that can describe the structure of the data. It involves multivariate statistical probing to describe proximity between the pairs of objects with the proximity data collected over time.

The term ‘proximity’ is an index defined over a pair of objects to quantity the degree to which the two objects are alike or different. Correlation coefficient, joint probabilities are two such examples of proximity measures which can explain the extent to which two objects show common attributes. A proximity measure helps in differentiating the objects and hence it can indicate either similarities or dissimilarities. Hence the term ‘proximity’ has varied contextual meanings based on the application in which the data visualization algorithms are employed. In general usage, the term proximity indicated by δ_{ij} indicates the dissimilarity between the objects i and j

The Multidimensional Scaling algorithm takes the proximity measures as the input. And the chief output is a spatial representation, consisting of a geometric configuration of points. Each point in the configuration corresponds to one of the objects and the configuration as a whole reflects the hidden structure in the data making them easier to comprehend. This implies larger the dissimilarity between the two objects in comparison, the farther apart they would be placed in the spatial map.

MDS starts with a matrix representing the distances or dissimilarities between 'n' objects. The power of this algorithm lies in its ability to depict the dissimilarities or the proximities between the objects through a placement of points in a low dimensional plane where the Euclidean distances between the points resemble the actual proximities between the objects as closely as possible. The best ever way to demonstrate the capabilities of MDS is by illustrating the classic example of cities and geographic map of Europe which is very widely used.

Consider the following illustration [23]. Let us suppose that we know the inter-city distances accurately for 10 popular cities of Europe. Essentially this would be a 10 by 10 symmetric distance matrix with the principal diagonal being zeros(distance of a city to itself is always zero). The actual distances in miles are given by the following matrix:

Table 1 : Inter-city distance matrix

	1	2	3	4	5	6	7	8	9	10
1	0	569	667	530	141	140	357	396	570	190
2	569	0	1212	1043	617	446	325	423	787	648
3	667	1212	0	201	596	768	923	882	714	714
4	530	1043	201	0	431	608	740	690	516	622
5	141	617	596	431	0	177	340	337	436	320
6	140	446	768	608	177	0	218	272	519	302
7	357	325	923	740	340	218	0	114	472	514
8	396	423	882	690	337	272	114	0	364	573
9	570	787	714	516	436	519	472	364	0	755
10	190	648	714	622	320	302	514	573	755	0

The numbers 1 through 10 correspond to each of the 10 cities shown on the figure 3. This distance matrix is fed to the MDS. The distances are first scaled down suitably with a scaling factor. Now the 10 cities are placed in a 2-D coordinate axes in such a way that their Euclidean distances match very close to their scaled distances and the resulting centroid of the entire configuration of points is at the origin. This forms the Relative Map. Now we may also have specific idea on some of the cities. For instance, assume that we already know the geographical locations of at least three cities say Stockholm, Madrid and Rome. We can now translate and rotate the relative map in such a way that the relative locations of the above mentioned three cities conform very closely to the absolute locations. In this process, we find that the rest of the cities also reach their absolute locations. The accuracy depends on the precision of inter-city distances and the transformation of relative map to absolute map.

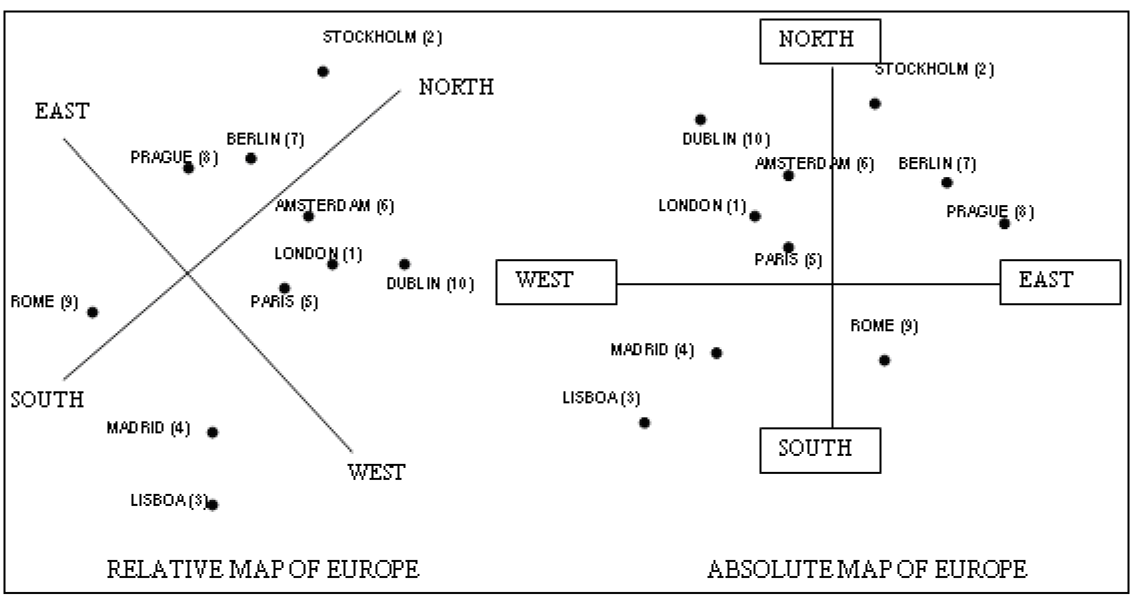

Figure 3 : Relative and absolute maps of cities of Europe

2.2 Types of Multidimensional Scaling

By now, we already understood that the MDS depends heavily on the proximity measure input in the dissimilarity matrix. In the above example, the inter-city distance was the proximity measure. While 'distances' are numerical figures, there are many other types of proximity measures also. Basically, the proximity variables can be divided into four broad categories:

1. Nominal Scale : Classificatory data with no comparisons possible .(Ex: Gender)
2. Ordinal Scale : Comparable data with no quantitative measure(Ex: Grades A–E)
3. Interval Scale: Difference in two values are meaningful but no zero (Celsius scale)
4. Ratio Scale: Same as Interval, but with defined zero.(Kelvin Scale)

Based on these four types of proximity measures, the classical Multidimensional Scaling is further classified into two types:

1. Metric Multidimensional Scaling
2. Non-Metric Multidimensional Scaling

Metric Multidimensional Scaling deals with the Interval and Ratio variables. This would include most of the models that deal with numerical scores, distances and other quantitative measures. This thesis work uses inter-subject distances and hence we will be using on Metric Multidimensional scaling only. The Non-metric Multidimensional scaling deals with the other two types of variables, nominal and ordinal. Mostly this model finds application in subjects involving abstract issues like behavior patters, affinity determination and other issues which are hard to be quantified. This type of MDS is mostly used in Psychology, Marketing and other related fields. Now onwards we will be dealing with only Metric or Classical Multidimensional Scaling in which the Proximity variable refers to the distance between the subjects

Classical Multidimensional Scaling

Let the proximities between any pair of nodes (r, s) be indicated by δ_{rs} where r, s= 1, 2…n and n is the total number of subjects. If the dissimilarities and the distances between the subjects are to be precisely Euclidean distances then Classical Multidimensional Scaling[text book] finds a configuration of points ensuring the equality

$$d_{rs} = \delta_{rs} \tag{1}$$

where d_{rs} is the distance between the two subjects in the configuration. Generally the above equation is not a strict equality and through the configuration of points, MDS always tries to minimize the loss function given by

$$\text{Loss Function} = \left(\left(\sum (d_{rs} - \delta_{rs})^2\right) / \sum (d_{rs}{}^2)\right)^{1/2} \tag{2}$$

2.3 Mathematical Modeling of MDS

This section deals with the mathematical modeling of the classical Multidimensional Scaling algorithm.[22]As explained in the illustration dealing with the geographical map of European cities, we can always find the locations of the subjects. But, only relative to each other. Depending on the necessity, these relative locations can then be transformed to retrieve the absolute locations. The first step is always to recover the relative coordinates.

Let the coordinates of 'n' points in a 'p' dimensional plane be given by x_r (r = 1,2,…n) where $x_r = (x_{r1}, x_{r2} \ldots x_{rp})^T$. Then the Euclidean distance between r^{th} and s^{th} points is given by

$$d_{rs}^2 = (x_r - x_s)^T (x_r - x_s) \tag{3}$$

Let the inner product matrix be B where,

$$[B]_{rs} = b_{rs} = x_r^T x_s \tag{4}$$

From the squared distances $\{d_{rs}\}$, this inner product matrix B is found, and then from B the unknown coordinates.

To Find B:

The first step is to set the centroid of the configuration of points at the origin. This will overcome the indeterminacy of the solution due to arbitrary translation. Hence,

$$\sum_{r=1}^{n} x_{ri} = 0 \quad where (i = 1,2,\ldots p)$$

From equation 1, we get

$$d_{rs}^2 = x_r^T x_r + x_s^T x_s - 2x_r^T x_s \tag{5}$$

The following equalities can be derived from the from the equation (5)

$$\begin{aligned}
\frac{1}{n}\sum_{r=1}^{n} d_{rs}^2 &= \frac{1}{n}\sum_{r=1}^{n} x_r^T x_r + x_s^T x_s \\
\frac{1}{n}\sum_{s=1}^{n} d_{rs}^2 &= x_r^T x_r + \frac{1}{n}\sum_{s=1}^{n} x_s^T x_s \\
\frac{1}{n^2}\sum_{r=1}^{n}\sum_{s=1}^{n} d_{rs}^2 &= \frac{2}{n}\sum_{r=1}^{n} x_r^T x_r
\end{aligned} \tag{6}$$

Substituting into equation (5) gives

$$
\begin{aligned}
b_{rs} &= x_r^T x_s \\
&= -\frac{1}{2}(d_{rs}^2 - \frac{1}{n}\sum_{r=1}^{n} d_{rs}^2 - \frac{1}{n}\sum_{s=1}^{n} d_{rs}^2 + \frac{1}{n^2}\sum_{r=1}^{n}\sum_{s=1}^{n} d_{rs}^2 d) \\
&= a_{rs} - a_{r}. - a_{.s} + a_{..} \\
&\textit{where} \\
a_{r.} &= n^{-1}\sum_{s} a_{rs} \\
a_{.s} &= n^{-1}\sum_{r} a_{rs} \\
a_{..} &= n^{-2}\sum_{r}\sum_{s} a_{rs}
\end{aligned}
\tag{7}
$$

Define the matrix A as follows before proceeding.

$$[A]_{rs} = a_{rs} \tag{8}$$

Hence the inner product B is now

$$B = HAH \tag{9}$$

where,

$H = I - n^{-1}11^T$ with $1 = (1,1,1\ldots1)^T$, a vector of n ones

This derived the matrix B. The next step is to derive the coordinates from B.

To Recover the Coordinates From B:

The inner product matrix B, can be expressed as

$$B = XX^T \tag{10}$$

where $X = [x_1, x_2 \ldots x_n]^T$ is the nXp matrix of coordinates. The rank of B, r(B) is then

$$r(B) = r(XX^T) = r(X) = p \tag{11}$$

Now B is symmetric, positive semi-definite and of rank p, and hence has p non-negative eigen values and n-p zero eigen values.

Matrix B is now written in terms of its spectral decomposition,

$B = V\Lambda V^T$ where $\Lambda = diag(\lambda_1, \lambda_2, ...\lambda_n)$, the diagonal matrix of eigen values $\{\lambda_i\}$ of B and $V = [v_1, v_2, ...v_n]$, the matrix of corresponding eigen vectors, normalized such that $v_i^T v_i = 1$. For convenience the eigen values of B are labeled such that $\lambda_1 \geq \lambda_2 \geq ..... \geq \lambda_{n-1} \geq \lambda_n \geq 0$

Because of the n-p zero eigen values, B can now be written as

$$B = V_1 \Lambda_1 V_1^T \tag{12}$$

where

$$\Lambda_1 = diag(\lambda_1, \lambda_2, ...\lambda_p) \quad \text{and} \quad V_1 = [v_1, v_2, ...v_p]$$

Hence as $B = XX^T$, the coordinates matrix X is given by

$X = V_1 \Lambda_1^{1/2}$ where $\Lambda_1^{1/2} = [\lambda_1^{1/2}, \lambda_2^{1/2}, ...\lambda_{p1}^{1/2}]$ and thus the coordinates of the points have been recovered from the distances between the points. The arbitrary sign of the eigenvectors $\{v_i\}$ leads to invariance of the solution with respect to reflection in origin.

Dissimilarities as Euclidean Distances:

To be of practical use, a configuration of points needs to be found for a set of dissimilarities $\{\delta_{rs}\}$ rather than simply for true Euclidean distances between the points $\{d_{rs}\}$. In the context of thesis, the dissimilarities would be the distances between the subjects measured by the ranging sensors on the WePosT system. From the previous explanation, if B is positive semi-definite of rank p, then $B = V_1 \Lambda_1 V_1^T$ where $\Lambda_1 = diag(\lambda_1, \lambda_2, ...\lambda_p)$

Now the distance between the r[th] and s[th] points of the configuration is given by $(x_r - x_s)^T (x_r - x_s)$ and hence,

$$
\begin{aligned}
(x_r - x_s)^T (x_r - x_s) &= x_r^T x_r + x_s^T x_s - 2x_r^T x_s \\
&= b_{rr} + b_{ss} - 2b_{rs} \\
&= a_{rr} + a_{ss} - 2a_{rs} \\
&= -2a_{rs} = \delta_{rs}^2
\end{aligned}
\tag{13}
$$

by substituting for b_{rs} using equation 7.

The above specified mathematical analysis can be summarized in the form of step-by-step procedure[22]. The practical algorithm for classical scaling will be as follows:

1. Obtain dissimilarities $\{\delta_{rs}\}$
2. Find Matrix A,

   $$A = [-\frac{1}{2}\delta_{rs}^2]$$

3. Find Matrix B,

   $$B = [a_{rs} - a_{r.} - a_{.s} + a_{..}]$$

4. Find the eigen values $\lambda_1, \lambda_2, \ldots \lambda_{n-1}$ and the associated eigen vectors $v_1, v_2, \ldots. v_{n-1}$ where the eigen vectors are normalized so that $v_i^T v_i = \lambda_i$. If B is not positive semi-definite(some of the eigen values are negative) either ignore them and proceed(the assumption adopted in thesis) or change the dissimilarity measure by adding constant and return to step 2 (not used in the Thesis)

5. Choose an appropriate number of dimensions p. When there is no pre-determined p value, use $\sum_1^p \lambda_i \Big/ \sum_1^{n-1} \lambda_i$ to estimate the number of dimensions needed to have a set value of explained variance.

Number of Dimensions and Proportion of Explained Variance:

As indicated previously, the eigen values $\{\lambda_i\}$ indicate how many dimensions are required for representing the dissimilarities. If B is positive semi definite then the number of non-zero eigen values gives the number of dimensions required. If B is not positive semi definite then the number of positive eigen values is the approximate number of dimensions.

The positive eigen values indicate the maximum dimensions of the space required. However to be of practical use, the number of dimensions of the chosen space should be small (generally 2-D). Then we might be interested in knowing the effects of using lesser number of dimensions in the model as compared to using all of the positive eigen values. To answer this question, we use the following measure[22]

$$\text{Proportion of explained Variance (PEV)} = \sum_1^p \lambda_i \Big/ \sum_1^{n-1} \lambda_i \tag{14}$$

And this is a measure of proportion of explained variance by using p dimensions. In an ideal case, if accurate inter-subject distances were collected from a 2D deployment, then we find just the first two eigen values to have significant figures and the rest of the eigens to be zero. In such a situation the above specified variable measuring the proportion of explained variance will work out to be 100%.

It is worthwhile to note that in the later chapters this measure (with slight modification) plays a very major role in appraising the performance of MDS.

2.4 Floyd's Shortest Path Algorithm

This algorithm, which is also referred to as Floyd-Warshall's algorithm [25], compares all possible paths through the connected graph between each pair of vertices. It does so by incrementally improving an estimate on the shortest path between two vertices, until the estimate is known to be optimal. Floyd's shortest path algorithm uses a technique called Dynamic Programming to solve the all-pair shortest path problem. The following explains the procedure involved.

The first step is to create an Adjacency matrix which is computed for any paid of nodes (i, j) as follows:

$$A(i,j) = \begin{cases} 0 & if \quad i = j \\ d_{if} & if \quad i \neq j \end{cases} \tag{15}$$

Note that $d_{ij} = \infty$ when i and j are more than 1-hop away

The adjacency matrix entries are recursively updated by the following function that searches exhaustively for all possible paths and picks the shortest path. The variable k indicates the possible number of iterations and there could be at the most k-1 intermediate nodes in between i and j in any iteration. The recursive function is given by the following expression with k equal to (1, 2…n)

$$d_{ij} = \begin{cases} A(i,j) & when & k = 0 \\ d_{ij} \quad \min(d_{ij}^{(k-1)}, d_{ik}^{(k-1)} + d_{kj}^{(k-1)}) & when & k > 0 \end{cases} \tag{16}$$

This leads to a final updated version of Adjacency matrix with respective shortest path distances as the matrix elements. The computation complexity is given by n^3.

Dijkstra's algorithm is another popular shortest path algorithm that can be used in this application. But Floyd's algorithm is more robust and involves lesser computational overhead in large networks. Moreover practical experience also indicates that Floyd's algorithm is faster than Dijkstra's algorithm in MATLAB simulation.[21]

2.5 Algorithm for MDS-Based Localization

In the case of localization problem, the dissimilarity measure of N subjects is an N x N distance matrix. The distance matrix which is fed to the MDS algorithm must be a symmetric matrix with zeros on its principal diagonal. The symmetry ensures that, for a given pair of nodes the distance between them is always the same when measured from either node and the zeros on the principal diagonal indicate that the distance measured by a node to itself is always zero. Given the above constraints on the inputs, the MDS can then plot these points with origin as the centroid. To get a perceivable output, there must be just 2 or 3 dimensions which are good enough to contain most of the information. Hence singular value decomposition is carried out on the distance matrix and only those dimensional are preserved which convey most of the information. In mathematical terms, these are the dimensions which are associated with correspondingly largest eigen values.

In summary, the localization problem can be addressed by the following steps using MDS [21]

1. The shortest path between the pairs of nodes is computed. The distance measurement capacity of a node is limited by its communication range. A node can measure distances to its neighbors only and for the rest of the nodes which

fall outside the communication range, an “infinite” value is assigned as the distance. (Typically “Infinite” takes the values of few tens of thousand so that it is always contextually large figure)

2. Floyd’s Algorithm is used to compute the shortest paths between any pair of nodes using the connectivity information.

3. The symmetric distance matrix obtained in the above step is input to the Classical MDS. As mentioned earlier, the CMDS does singular value decomposition and eliminates dimensions corresponding to non-significant eigen values, thereby constructing a relative map with 2 or 3 dimensions. An optional refinement step involving least-squares minimization can be included to best conform the inter-distances of the nodes to the measured distances.

4. The relative map obtained can be transformed into an absolute map, if provided with the minimum number of anchor nodes, (3 nodes for a 2-D and 4 nodes for a 3-D networks). First, a transformation function is created by mapping the relative coordinates of the beacons with their known absolute coordinates. This might involve some translations and rotations. The obtained transformation function is then applied to the rest of the nodes. An optional refinement step involving least-squares minimization can be included to conform the inter-distances of the nodes to the measured distances.

2.6 Performance of MDS-Based Localization

The performance of classical Multidimensional Scaling based localization is determined fundamentally by the network topology parameters. It was observed that the density of the network has direct relationship with the performance. Simulation results show that the denser networks exhibit less mean localization error. The second distinctive performance parameter is the shape of the network topology. If the nodes were deployed in a uniform pattern, the results were better. Contrastingly, irregular deployment increases the error and especially c-shaped networks yield highly unsatisfactory results.

In the view of the parameters identified, the MDS-based localization procedure is realizable in most of the cases where the nodes are deployed densely and regularly in a static network. Once deployed, the nodes do not change their locations and hence, the regularity of the network can possibly be addressed by a proper initial deployment of nodes. But the density is fixed and moreover this approach cannot be extended to a dynamic network where the nodes move around randomly and might end up forming an irregular network in due course of time.

The performance of the MDS is restrained by the density of network and regularity in the locations of the deployed nodes. Hence this situation calls for an algorithm which comes into picture once the nodes start dispersing. It should be able to accommodate the issues of density and regularity to the best, though irregular networks are always a problem. These two issues are addressed in the following chapters.

2.7 Results and Analysis

Experimental Set-Up

For simulation purpose, 100 mobile nodes are deployed randomly in a field of predetermined dimensions (in this case, a 5r-by-5r square) where ‘r’ is unit length of the placement area. Each of the nodes has a ranging capability of 2.0r, i.e., they can sense the presence of another node within a vicinity radius of 2.0r. Gaussian noise (of standard deviation of 5%) is introduced to the true inter-node distances to depict the inevitable ranging errors.

The simulation results follow more or less the results which were presented in the MDS-based localization algorithm put forth by Yi Shang et al.[19] The performance of the localization algorithm is analyzed with respect to two of the main influencing factors: network topology and density of the network.

(i) Network Topology

To examine the effects of the network topology, 100 nodes were deployment in three different deployment areas. In the first one, the nodes were regularly placed in a grid. The second placement area is in the form of-c-shaped network. The final placement area is of practical use and in this deployment the nodes are dispersed in a random fashion. The localization error is defined as the mean of the distance of estimated location to the actual location. In the error plot, these distances are indicated by a red-line between the actual locations (‘o’) to the estimated location (‘x’).

Figure 4 describes the Regular Grid network and the error works out to be the least value of 0.12 % of the unit length ‘r’.

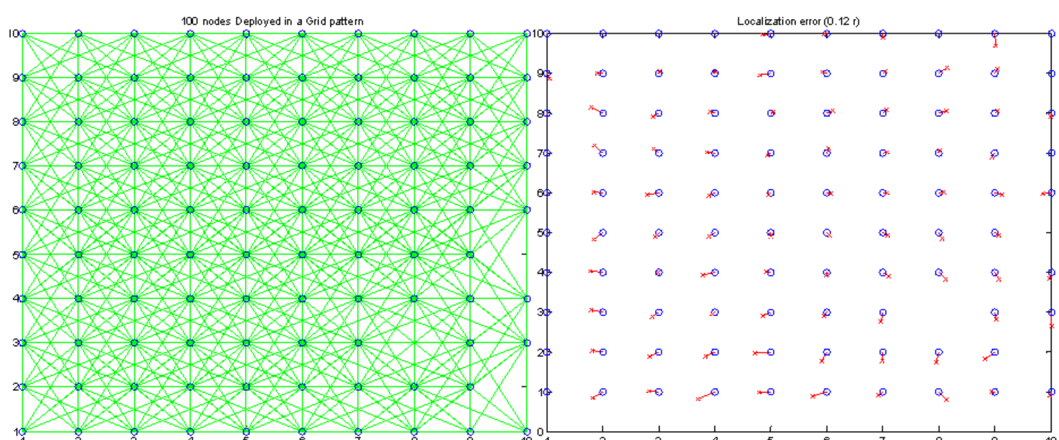

Figure 4: MDS-based Localization of Grid network yields good results.

Figure 5 indicates the other extremity of the network where the nodes are placed in a c-shaped pattern. We expect that this would bear huge errors due to wrong estimates of the inter-node distances. The error value turns out to be the highest and it is about 0.25 % of the unit length.

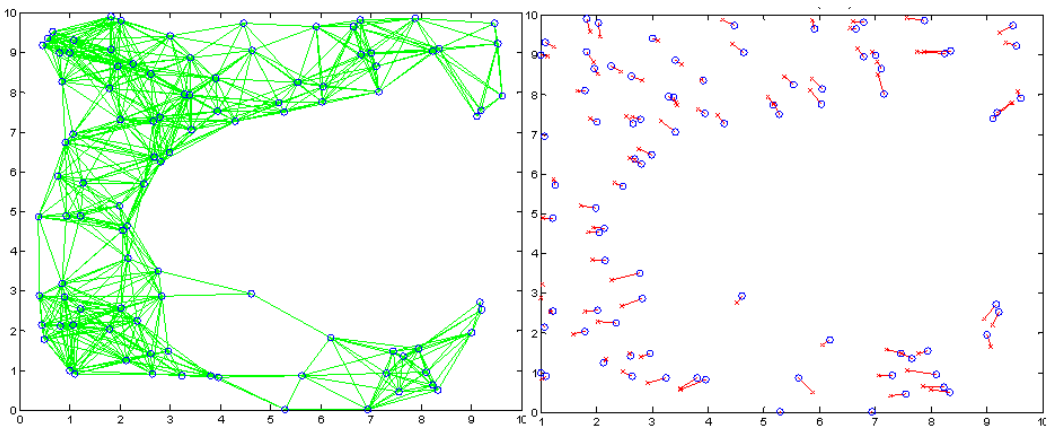

Figure 5: Low performance of MDS-based localization in C-shaped networks

The figure 6 depicts random deployment of the nodes which is more common than the either of the above two types of deployments. For this case, the error is an intermediate value ranging at about 0.2% r.

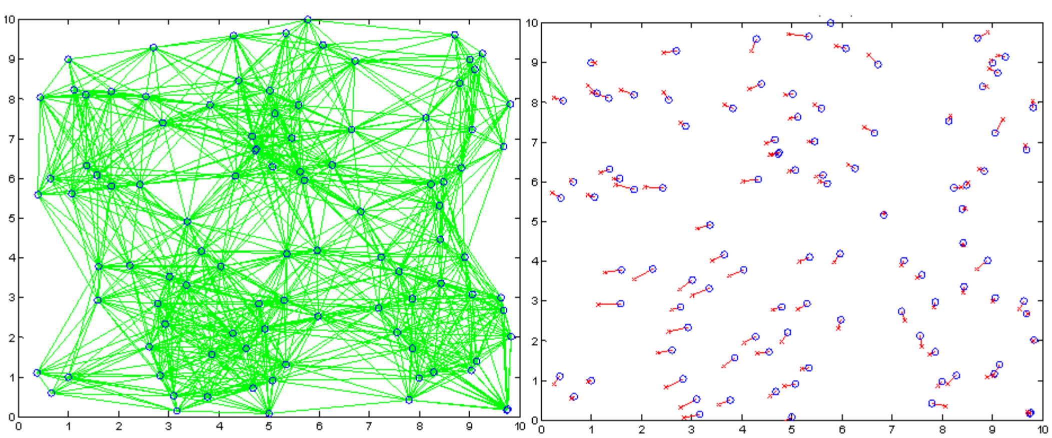

Figure 6: MDS-based localization of random network: A practical situation.

It is observed that if the communication range is decreased, it has highest impact on the c-shaped deployments which yield higher localization errors.

(ii) Density of the network

Density of the network is defined as the number of nodes per unit placement area. The placement area being constant, the density is defined by the number of nodes. We consider only the random deployment due to its practical importance and we can extend the results to the rest of the networks.

The Figure 7 has 100 nodes and the Figure 8 has 175 nodes. All other network parameters are left unchanged.

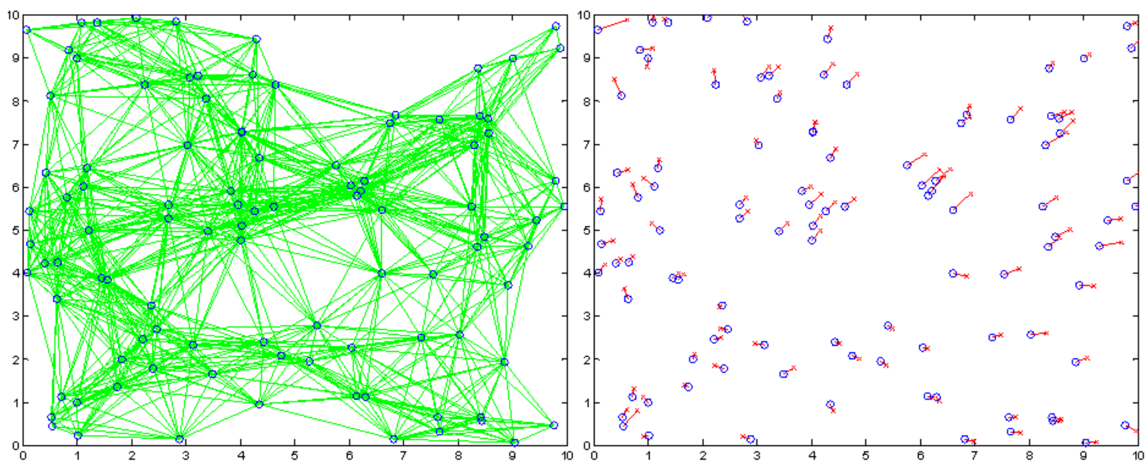

Figure 7: Random network with 100 subjects

It is evident that the increase in density leads to increase in the performance. This can because of the better connectivity between the nodes and this in turn makes Floyd's algorithm yield better estimates. Also MDS tends to work better with dense networks.

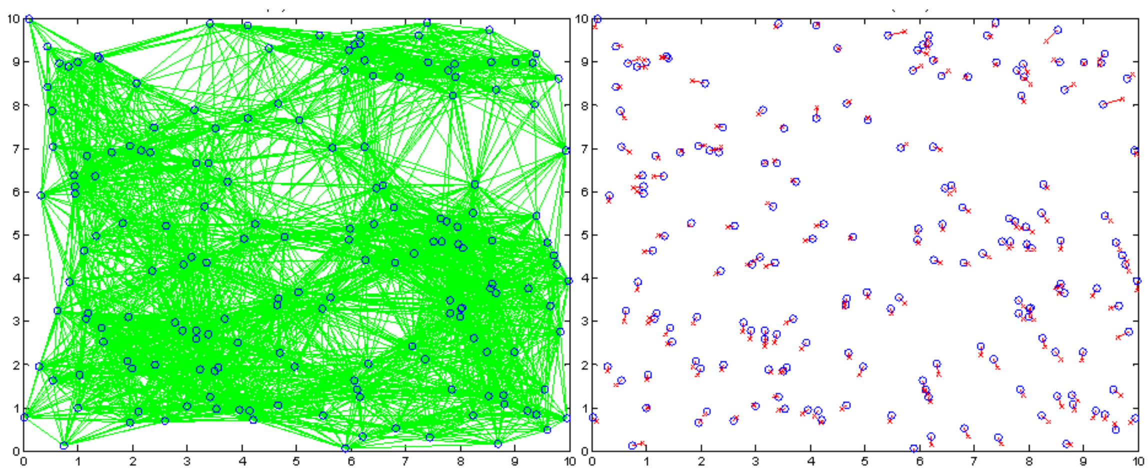

Figure 8: Random network with 175 subjects: performance improved

Hence we can conclude that apart from the quality of the ranging devices employed, the network topology shape and the node density also directly impact the performance of the MDS-based localization algorithm

CHAPTER III

DYNAMIC LOCALIZATION

3.1 Introduction

The applicability of data visualization algorithm like Classical Multidimensional Scaling has been validated for the localization contextual issues [21]. The performance of this type of localization depends heavily on network topology parameters. As explained in the earlier sections of the thesis, network topology, network density and the degree of precision involved in the estimation of distances between the subjects defines the performance of the Classical Multidimensional Scaling.

Obviously one cannot provide a generic solution if the applicability of the localization procedure is constrained by the specific network topology parameters and such solutions are not called-for. For instance, we cannot force that the subjects be always placed in regular pattern just because we know that the algorithm works best in such cases. Such a proposition lacks generality and also it should be understood that the network topology properties are defined by the application specifications, which cannot be bent for the sake of easy computations.

Now we shift our focus onto emerging trends in mobile sensor networks. We have already explained at length the need for the mobile sensors. With growing interests that have been detailed in the Chapter I, we cannot deny that mobile sensor networks need more attention than just the stationery fixed subjects.

What we call ‘Dynamic Localization’ is unique solution for the mobile sensor networks and it also takes care of the network topology constraints of the sensor network. This is an extension to the classical multidimensional scaling based localization procedure and hence it offers all the advantages that are offered by it. All that we need are the inter-node distances and the connectivity information with which a relative mapping can be generated and converted to absolute location with the help of beacons. The only change is that these data are sensed from the network in regular time intervals, which shall be referred to “iterations’ in programming jargon and hence used interchangeably hereafter. In between two consecutive iterations, a fraction of subjects might have moved away from their locations, thereby changing the network topology all together. This dynamic behavior is not only supported but also utilized in increasing the performance of the localization procedure. At this point in time, we introduce the concept of adding “virtual nodes” which makes the network topology more and more suitable for Classical Multidimensional Scaling and this forms the crux of Dynamic Multidimensional Scaling (DMDS)

Once again, it is imperative to note that we cannot set predetermined rules for the moving subjects. For instance, if the subjects refer to livestock then we have to include the high degree of randomness in their movement as the time passes. Keeping the nature of Dynamic Multidimensional Scaling in view, the Dynamic Localization procedure incorporates another technique called Dead Reckoning based localization [26] which produces a parallel estimate of the node locations. The final result is a weighted average and we shall statistically proved that this combination of the two results performs better than the DMDS results at times when the network parameters aren’t favorable for the underlying classical Multidimensional Scaling. The following sections provide detailed descriptions about all of the above mentioned procedures and the conclusion presents the simulation test outputs that stand in accordance with the anticipated performance improvement at various stages.

3.2 Dynamic Multidimensional Scaling

Dynamic Multidimensional Scaling is similar to Classical Multidimensional Scaling except for the nature of the subjects involved. Dynamic Multidimensional scaling or DMDS involves the localization of "virtual" nodes apart from the original nodes.

It can be re-iterated that the most important issue in this type of range-based localization is the density of the network. A node is limited in its communication range and at times it cannot have a single hop communication with most of the other nodes. Floyd's algorithm assigns a shortest path distance based on connectivity. However, the shape of the network can cause the shortest path between two nodes to be much different than the actual Euclidean distance between them. This will yield highly erroneous results in the estimation of the locations as their inter-distances are now different from the actual distances. Dynamic MDS can deal with this situation by utilizing the node mobility to yield better results.

3.2.1 Addition of Virtual Nodes

The density of a static network is fixed. But in case of a dynamic network, the density can be increased by adding "virtual" nodes. Whenever a node moves, the old location is preserved by assuming a virtual node in its place. This way, the method is still associated with previous connectivity information of a node. In other words, the overall density and connectivity of network is increased in every iteration and this leads to better estimation of the inter-node distances. Note that ‘connectivity’ in this context helps for better estimation of inter-point distances which is our goal and it doesn’t have anything to do with the actual communication path if the nodes are supposed communicate with each other. We are trying to improve the localization only.

The following Figure 9 explains the impact of having virtual nodes amidst the real nodes in the deployment area. Figure a shows 4 nodes A, B, C and D initially. Assume that the nodes A and D fall out of range with each other and hence in the absence of any other nodes, the Floyd’s algorithm picks A-B-C-D to be the shortest path between A and B and hence the length of this path becomes their inter-distance. Now the nodes B and C move away from their initial locations in the indicated directions. Figure b depicts the final positions of all the nodes at the end of the iteration and additionally two virtual nodes were introduced at the old locations of the nodes B and C. The number of nodes has increased from 4 to 6.

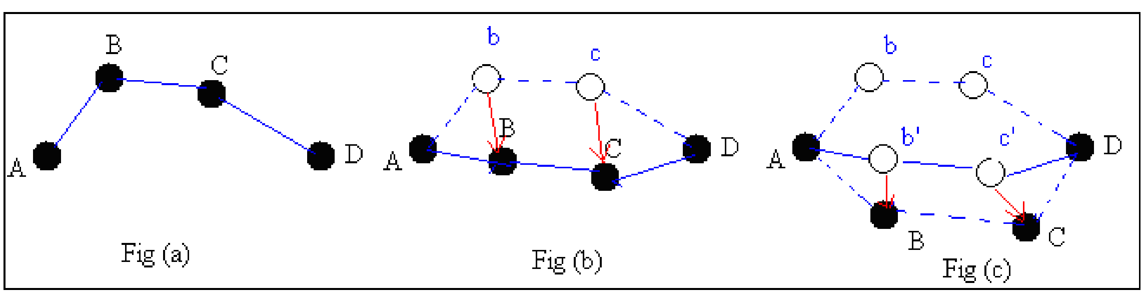

Figure 9: Virtual nodes improve connectivity and hence performance

Now consider Figure c. The nodes B and C have moved as shown in figure b. We also have two virtual nodes b' and c' in place of the old locations of the nodes B and C. Now the shortest distance estimate of A and D is via A-b'-c'-D which is obviously a better estimate. Also the network has become denser with increased connectivity among 8 nodes in total. Note that the Figure conveys information with more accuracy and hence the MDS results are reliable. The enhanced network has become denser with increased connectivity and increased accuracy of inter-node distance estimate. This validates the underlined concept of adding virtual nodes to the network. The other way of explaining the entire effect is by mentioning that whenever the network topology is bad (indicated by c-shaped connectivity), the movement of nodes might suitable modify it over a period of time. If the movement of nodes tends to distort the network negatively (as in the case of the above example) the virtual nodes can preserve the information contained. This way we take advantage of the mobile nodes and their respective virtual nodes.

3.2.2 Algorithm for Dynamic Multidimensional Scaling

The basic idea of Dynamic Multidimensional Scaling still relies on the Classical Multidimensional Scaling at the root level. As in CMDS, the inter-node distances are used as an input to the Dynamic Multidimensional Scaling algorithm, which then locates these nodes on a relative map of perceivable dimensions(2D or 3D at the most). The presence of necessary number of beacons can convert this relative map into an absolute map. But now, the conspicuous difference is that the network is virtually "growing" in density and connectivity. In every iteration, a percentage of nodes are assumed to be mobile. Actually, any percentage of nodes may move and this number may vary in each iteration. The selection of mobile nodes, the direction of movement and the distance traveled in an iteration are all assumed to be random in nature. Moreover, there is no hard and fast rule that a mobile subject is in constant motion (ex: livestock). Hence in each iteration the mobile nodes are selected randomly irrespective of their history of motion till the current iteration. This simulates the real world situation of mobile subjects with more meaningful assumptions. The following steps explain the step-by-step procedure involved:

1. For the first iteration, the nodes are just deployed and it is assumed that they haven't started moving yet. So, use the distances and the connectivity information as it is to get the relative mapping of the nodes.
2. Obtain the absolute locations of the nodes by creating a mapping function using best three beacons.
3. A certain percentage of nodes move per iteration. Assume virtual nodes in their old locations. This way, we observe that the number of nodes increase in the first few iterations and hence the network gets denser.
4. Run the CMDS-based localization algorithm explained in the previous chapter.
5. When the network becomes satisfactorily dense (which is generally identified by threshold value of mean connectivity), start forgetting the oldest virtual nodes to accommodate the virtual nodes introduced in the current iteration. This essentially maintains a constant number of total nodes (actual and virtual) thereafter and avoids the uncontrolled build up of network.
6. Repeat from step 2.

As verified from the simulation results, the DMDS brings down the mean localization error significantly over the iterations. This is an obvious outcome as we know that the network is getting denser with better topology. To talk in terms of quantitative measure we can have a look at the eigen values returned by the Multidimensional scaling procedure. It will be observed that the eigen values corresponding to the first two dimensions increases considerably as we add more and more virtual nodes. This directly implies that MDS is giving more and more importance to the first two dimensions (X and Y), which is a positive indication. Note that the simulations were carried out on 2-D deployment area and hence ideally only two dimensions must be used by the MDS estimates. We can also observe that the mean connectivity value significantly show a raising pattern as the time progresses.

Another important issue also comes into light through simulation results. It is observed that the decrease in the mean localization error is not consistent at times. This is denoted by sudden shoot-ups of the error in some of the iterations. This can be attributed to the randomness involved in the node movement and the nature of Multidimensional Scaling algorithm itself. This cannot be predicted beforehand but we must be ready with an alternative that can suppress or reduce these unanticipated errors. This reasoning forms the building block for using WePosT system which uses Dead Reckoning technique to provide estimates of the nodes. The final result is an ensemble result which integrates the location estimate from DMDS and the location estimate from the WePosT system. But it is essential to note that the Dead Reckoning estimates are not stand alone results and they are considerable dependent on the DMDS results. Once again we authenticate the above explained sensor fusion technique using simulation results which indicate that the combined result is a much better estimate when Dynamic Multidimensional Scaling estimate return an inconsistent result with increased mean error. The following section deals with the details of the sensor fusion technique explained here.

3.3 Fusion of Estimates from WePosT and Dynamic MDS

WePosT system constitutes ranging sensors and inertial sensors which can describe the nature of the node's movement with regards to the direction of the movement and the traversed distance. When a node moves, the heading direction is provided by the gyro and the compass on the WePosT system. Using Dead Reckoning, the position are then estimated.

Dead Reckoning based localization technique is not a stand alone method and it depends on the Dynamic Multidimensional Scaling to an extent. WePosT estimates are calculated by using simple coordinate geometry. Whenever we have the knowledge of the initial location of a mobile node apart from the distance traveled and the orientation angle, we can estimate the final location of the node. The WePosT system can sense the orientation angle and the distance traversed, but the initial estimate is provided by the Multidimensional scaling and this makes the Dead Reckoning method dependent on the MDS result. But then it is only for the first iteration that the Dead Reckoning method entirely borrows the initial estimates form the DMDS. Thereafter it relies on the fused estimation for the same purpose. The following puts this in better words.

For the first iteration we have the nodes just deployed and hence we have the position estimates only from the CMDS which is also treated as the final fused result. From the second iteration onwards, the Dead Reckoning uses fused estimates from previous iteration as the initial locations and estimates the final locations using distance and angle information. In mathematical expression, the estimate of new location $\hat{p}_w(k)$ of the moving node is expressed as follows:

$$\hat{p}_w(k) = [\hat{x}_w(k), \hat{y}_w(k)]^T = [\hat{x}(k-1), \hat{y}(k-1)]^T + [d(k).(\sin\theta(k)), d(k).(\cos\theta(k))]^T \quad (17)$$

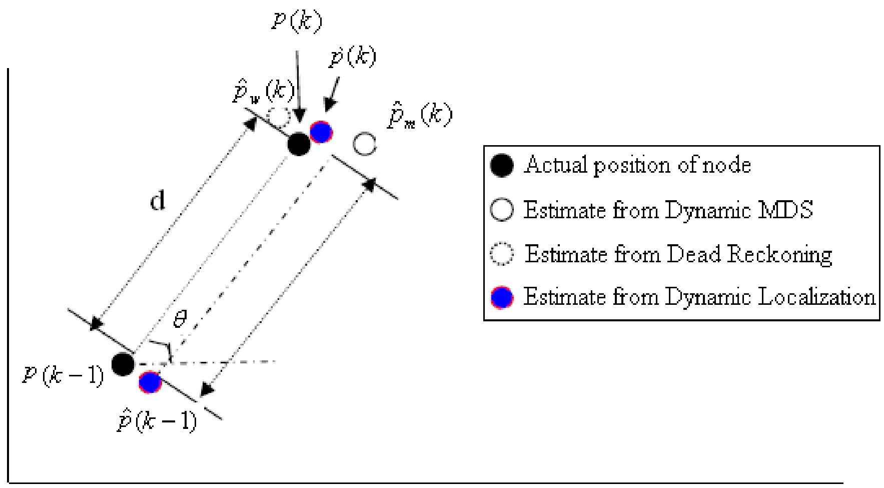

Figure 10 : Dynamic Localization estimations

Here $\hat{p}_w(k-1) = [\hat{x}(k-1), \hat{y}(k-1)]^T$ is the fused estimate in the last step. Since the first iteration has only DMDS result, the initial estimates for dead reckoning in the second iteration are chosen to be the previous DMDS result Thereafter we have the fused estimates as the initial locations. $d(k)$ and $\mu(k)$ are the distance and angle measurements which are corrupted by Gaussian noises for simulation purpose. From the second iteration onwards, the concept of sensor fusion comes into the picture. For the fraction of nodes that didn't move in a particular iteration, the modified algorithm takes the average of current DMDS results and the past results of the DMDS over the last few iterations (2 or 3 iterations in which these nodes didn't move). For those of the nodes which moved in a particular iteration, there are now two different estimates of their final locations. One estimate $\hat{p}_m(k)$ is from dynamic MDS that relies solely on the distances information. The other estimate $\hat{p}_w(k)$ comes from the WePosT system that uses the previous location estimate and the dead reckoning in the current step. The new location estimate $\hat{p}(k)$ of the moving subject is a weighted average of the results of both estimates as shown in the following equation

$$\hat{p}(k) = \frac{w_w \hat{p}_w(k) + w_m \hat{p}_m(k)}{w_w + w_m} \tag{18}$$

where W_w and W_y are the two weights designated for the dead reckoning based localization estimate and Dynamic MDS estimate respectively.

To keep it simple, we choose equal weights for both of the estimates. Many other possible weights were also tried in the simulation. But one has to realize that the weights cannot be generalized and hence we zeroed in on equal weights. In the next chapter the selection of appropriate weights turns out to be the most crucial issue as it is done at run time.

This procedure of having a fused estimate of the WePosT and the Dynamic MDS is referred to as Dynamic Localization. In summary, the Dynamic Localization algorithm is as follows:

1. For the first iteration the Classical Multidimensional scaling is applied on the data to get the initial configuration of nodes. This is treated to be the final result ad also as the initial estimated for the Dead Reckoning technique in the second iteration.

2. From the second iteration onwards, whenever a node moves to a new location, the heading of the movement and the distances moved are measured. Also old locations of the nodes are preserved by assuming virtual nodes in their places
3. Run Dynamic MDS to get the estimate of the new positions $\hat{p}_m(k)$
4. From the heading and the distance moved, the new location $\hat{p}_w(k)$ of a mobile node can be estimated according to Equation for $\hat{p}_w(k)$.(equation 17)
5. For the nodes in motion, the localization is done by taking the weighted average of the above two estimates. For the stationary nodes, the estimate is just the average of current results and the past results of the Dynamic MDS over the last few iterations, in which the nodes were stationery.
6. Repeat from step 2.

One of the important changes made in the algorithm is the absence of refinement step which was used in the original MDS-based localization algorithm [21]. It has been noted from the simulation that though refinement improves the performance initially, the final settling error is not significantly different. Moreover the refinement step demands high computation time and it has been proved that it is much more expensive than any other steps as the number of nodes increase. Hence in order to reduce computation time, the refinement step is excluded at the cost of performance which is slightly affected in the first few iterations. After reaching the threshold connectivity, the performance is more or less the same as the results after refinement step.

3.4 Beacon Selection in Absolute Mapping Transformation

The presence of beacons amongst the nodes helps us to retrieve absolute mapping from the relative locations of the nodes. The absolute locations of the network are obtained through a transformation which maps the relative map of network (from MDS results) to the absolute map with least error. This transformation is generated by using beacons whose relative locations are subjected to translation and/or rotations to conform them to the actual absolute locations with least error. These translation and the rotation components will be directly generated if Procrustes analysis [23] is used. Procrustes analysis takes two configurations of points and conforms them to each other by essentially centering them at origin. In doing so, it may translate and/or rotate the configuration of points. Since we have the absolute and the relative estimates of the beacons, we can use Procrustes analysis to best conform each other. The resulting transformation function can then be applied over the rest of the relative locations of the nodes to estimate their absolute locations.

For a two dimensional representation (which is the present case), at least 3 beacons are necessary to find the transformation. In Shang's paper [21], all the beacons are used to calculate the transformation function. This may lead to significant errors because the relative locations of some beacons may carry large errors as a result of the localization algorithm. Once again we cannot limit these beacons to fixed locations and hence even if they are placed in suitable locations, they might move over time and place themselves in an unfavorable way. Since the transformation function is a very important step in the process, care should be taken by avoiding the beacons that are not localized properly. In order to reduce the effects of these "bad" beacons, we propose a *beacon selection* algorithm to pick just three "good" beacons to be used in the transformation calculation.

By terming the chosen three beacons "good", we imply that they satisfy the following two requirements: (1) *Goodness*. The triangle formed by the three beacons should be as close to an equilateral triangle as possible. If the three beacons are almost collinear, then such a placement of nodes doesn't yield satisfactory transformation function. Ideally they should form an equilateral triangle which ensures that the nodes are not collinear. (2) *Similarity*. The corresponding triangle formed by the estimated beacon locations should be as close as possible to the triangle formed by the actual beacons. In summary, the goodness requirement will guarantee the selected beacons are not close to co-linear, a formation that can lead to large errors in the transformation. The dissimilarity requirement will guarantee the selected beacons have less error in the relative locations.

The Beacon Selection algorithm examines all the triangles which can be exhaustively formed from the existing number of beacons. Assuming the number of beacons to be *n*, the total number of triangles is C_n^3.

The following task is to set up a common measure on which all of these triangles can be compared. In this algorithm we have used the concept of variance for the sake of comparison. The average of lengths of the three sides of a given triangle in the relative map is computed. The variance of each of the 3 sides from this average length is measured. The obtained measure is then added (with some weights) to the actual variance measure in the absolute map. Ensuring least variance, one can pick a triangle which is nearly equilateral and which is similar in both relative and absolute map. Such a triangle qualifies to be a desired candidate.

While the above method ensures a triangle with least variance on it sides, an ideal selection would pick the biggest triangle from these triangles. Hence a second comparison measure is set up that would choose the biggest triangle from the above best possible triangles. Weights can be included to these two assessment measures to bias the triangle selection. In this paper, no dedicated weights are considered and the final measure for each of the triangles is an average value of the above two measures.

Then the goodness measure *and* the dissimilarity measure are calculated according to the following

Average of sides of the triangle in absolute map $s_{avg} = (S_1 + S_2 + S_3)/3$

Average of sides of the triangle in relative map $s'_{avg} = (S_1' + S_2' + S_3')/3$

Variance of triangle in relative map $V_{MDS} = \dfrac{\sqrt{\left((S_1' - S'_{Avg})^2 + (S_2' - S'_{Avg})^2 + (S_3' - S'_{Avg})^2\right)/3}}{S'_{Avg}}$

Variance of triangle in absolute map $V_{ABS} = \dfrac{\sqrt{\left((S_1 - S_{Avg})^2 + (S_2 - S_{Avg})^2 + (S_3 - S_{Avg})^2\right)/3}}{S_{Avg}}$

$$\text{Total variance, } V = (W_1.V_{MDS} + W_2.V_{ABS}) \big/ (W_1 + W_2) \qquad (19)$$

In the present case, the above weights W1 and W2 are chosen to be 1:1

The total variance measure takes care of choosing three beacons such that triangle formed by them is nearly equilateral in both relative map and absolute map. The next step is to check if these beacons form similar triangles in both the maps, which is indicated by the following dissimilarity measure:

Average length, $L_{Avg} = (S_1 + S_2 + S_3 + S_1' + S_2' + S_3')/6$

$$\text{Dissimilarity, D} = \frac{\sqrt{\left((S_1^{'} - S_1)^2 + (S_2^{'} - S_2)^2 + (S_3^{'} - S_3)^2\right)/3}}{L_{Avg}} \tag{20}$$

Dissimilarity measure makes sure that the corresponding sides of the triangles in two maps agree with each other very closely.

Finally, we have two measures on which the 3 beacons are chosen and these two measures are unified by weighted average using appropriate weights.

$$\text{Beacon\_selection} = \left(W_3.V + W_4.D\right) \big/ \left(W_3 + W_4\right) \tag{21}$$

Once again we zero-in on equal weights.

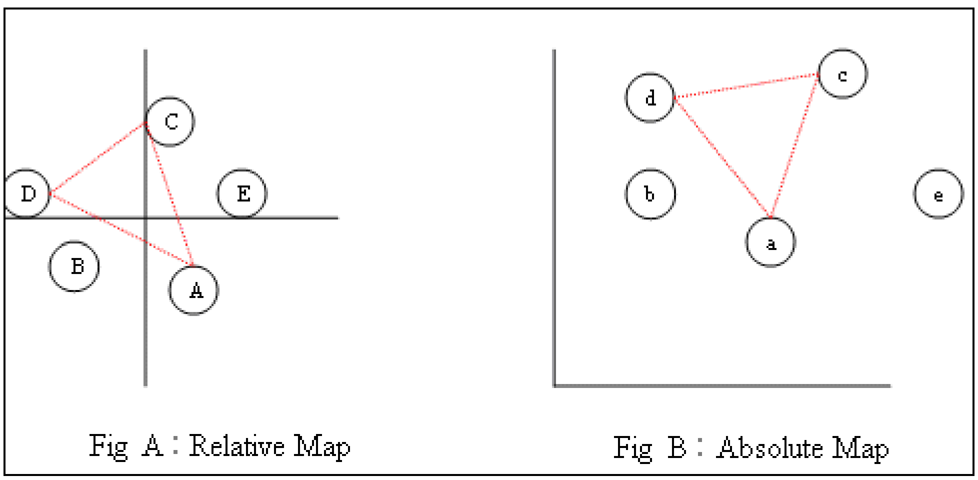

Figure 11: Beacon Selection Illustration

For instance, consider Figure 11. Figure A is the relative map of beacons as estimated by localization algorithm and figure B indicates their actual positions. Out of the 5 beacons, best 3 can be picked by looking at the 10 triangles formed by the in either maps. Triangle formed by nodes A, B, D match very closely with that of nodes a, b, d but they are very close to being collinear. The Nodes a, e, c are actually located favorably in absolute positions, but A, E, C weren't estimated correctly by the localization algorithm. So the best choice would be A, D, C which form similar triangles in both the maps and they are ideally spaced apart. Hence suitable rotation and translation function on these beacons will determine a transformation function to map relative positions to absolute positions. By avoiding beacons B and E, we will be creating a better function.

3.5 Results and Analysis

Experimental Set-Up

For simulation purposes, initially 100 mobile nodes are deployed randomly in a field of predetermined dimensions (in this case, a 5r-by-5r square) where 'r' is the unit length of the placement area. Each of the nodes has a ranging capability of 1.0r, i.e., they can sense the presence of another node within a vicinity radius of 1.0r These nodes are referred to as neighbors and a node can measure the distance to its neighbors. However it is essential to note that these distances are prone to inevitable ranging errors which are modeled in the experiment by adding Gaussian noise to the true distances.

Each of the iteration indicates a time instant in the continuous changes occurring in the network and the data is collected when the mobile nodes traverse an average length of distance say 1.75r. In practice, the inter-node distances would be computed in regular time intervals, which are referred to as 'iteration' in the simulation process.

In an iteration, any percentage of nodes may move and this number may vary in each iteration. For our convenience, it is assumed that on an average, 75% of the nodes (75% of 100 = 75) move in each iteration. The direction and the length traveled by a node in an iteration are chosen randomly. Apart from this, the selection of mobile nodes is also done randomly, taking care not to choose virtual nodes in the process. This way, the simulation mimics the randomness existing in real world situations, wherein the motion of the mobile subjects (livestock or soldiers) is not limited by any predetermined rules.

1. Improvement in Network Topology: The following figures demonstrate the improvement in density (thereby better connectivity and favorable topology)

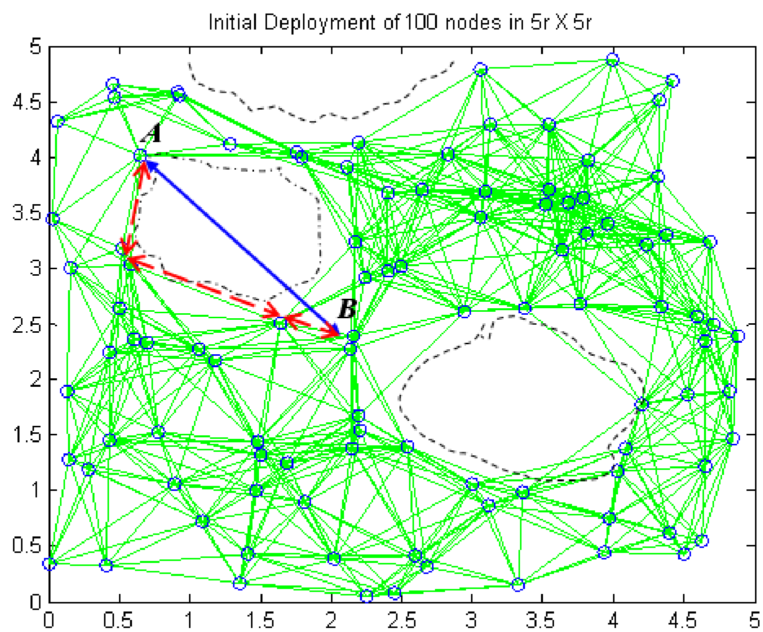

Figure 12 : Large errors in shortest path estimation for sparse network

The blue circles indicate the actual nodes and the green lines indicate the one-hop connectivity. The above deployment gives us an impression that the inter-node distances for the nodes which fall out of range will obviously be prone to errors after running Floyd's algorithm. This is indicated by the free-form dashed lines on the graph. These are the potential areas causing concern. For instance, consider the nodes A and B indicated on the plot. The shortest path distance is indicated by red dashed line which is obviously very different when compared to their actual Euclidean distance indicated by the solid blue line drawn between the two. Hence we can expect the MDS result to be erroneous.

Now the concept of DMDS is brought into picture. For all the nodes, we have the neighbor information and estimates of their respective distances. Now we introduce the concept of adding virtual nodes to the network. Before a node sets out to a new location, we capture the connectivity and the one-hop neighbor distances and use them in subsequent iterations. This in effect is equivalent to perceiving a 'virtual' node in the old location of the node. This is because we have taken a back-up of all the pertinent information like connectivity and 1-hop distances, which would have been available when a 'real' node is present at that location.

The Figure 13 indicates the above deployment after two iterations when 75 nodes move to new locations in every iteration. This time, we have added 75 virtual nodes in each iteration. (Indicated by red boxes as in contrast with the blue circles) Clearly, the density of the network has improved when compared to the figure 11. But the network didn't reach the threshold limit yet. Hence we add more nodes and we observe that with 225 virtual nodes the network reaches the threshold and it is visible that the shortest path distances between most of the pairs of actual nodes have significantly improved.

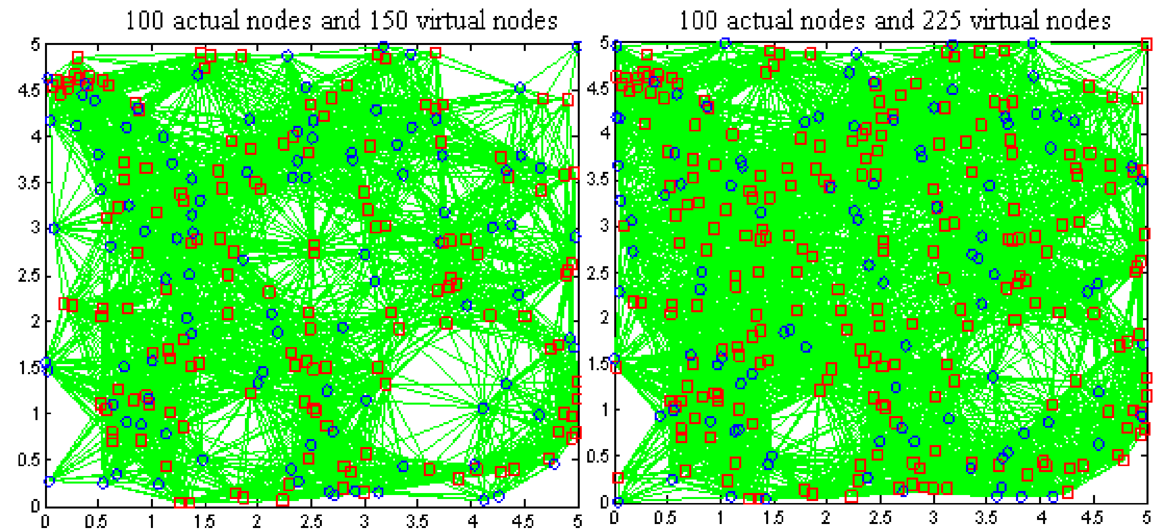

Figure 13: Improved network with 150 and 225 virtual nodes

It is to be understood that the communication radius of the nodes is one major factor that is related to the connectivity of the network. If this radius were high the inter-point estimates are much better as more number of nodes fall within each other's communication range. In the above figures, the radius was 1.0 unit distance. If this radius is increased to three as in shown in Figure 14, we see that the initial deployment itself is pretty good and addition of virtual nodes makes it even better. This can help in choosing right sensors for an application.

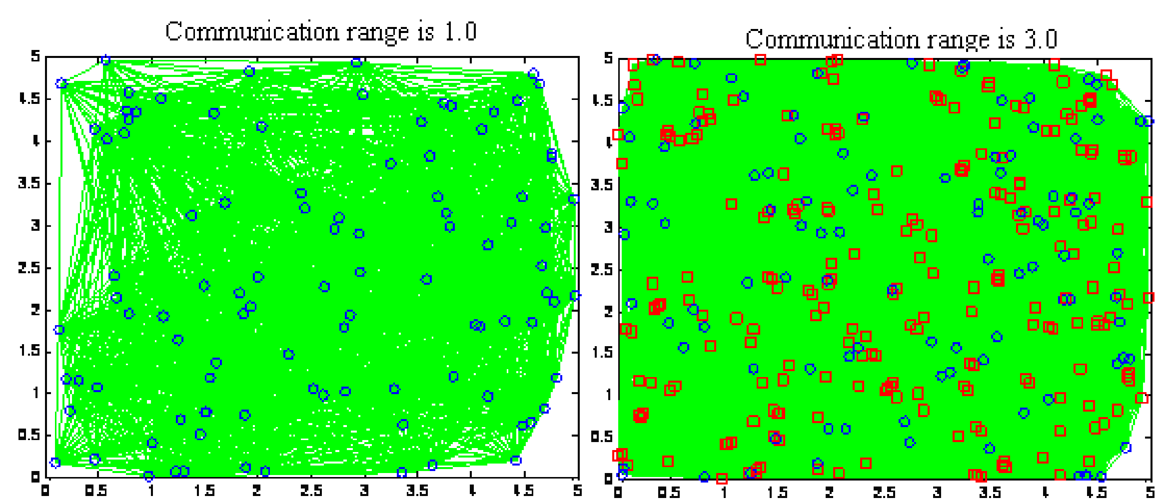

Figure 14: Positive Effect of Increased Communication Range

2. Removal of Refinement Step.

It can be recollected form the Dynamic MDS algorithm that we have eliminated the refinement step which was used in Multidimensional Scaling based location algorithm for static networks. Though this might raise some concerns about the quality of the Dynamic MDS, the following observations reinforce our stand.

It is true that the Refinement step yields better estimates than the model with its exclusion. But for all practical purposes, the emphasis is not just on the performance alone. In general, one has to also take care of the computation time overhead for the optional steps like the refinement step. It has been proved that the refinement step is the costliest step compared to the rest of the processes. But it is only through simulations that we can quantity such an effect. The following Figure 15 indicates the error plots for the Dynamic MDS with and without the refinement step:

The first impression would immediately tell us that the refinement step yields better results. This is very clearly seen in the first iteration. From a closer perspective, we realize that overtime, the unrefined results approach the refined result very closely. And given more number of virtual nodes, it might get even closer. From a practical view-point this explanation would be sufficient to validate our decision of excluding the refinement step as it doesn’t produce significantly better results as compared to unrefined results after few iterations.

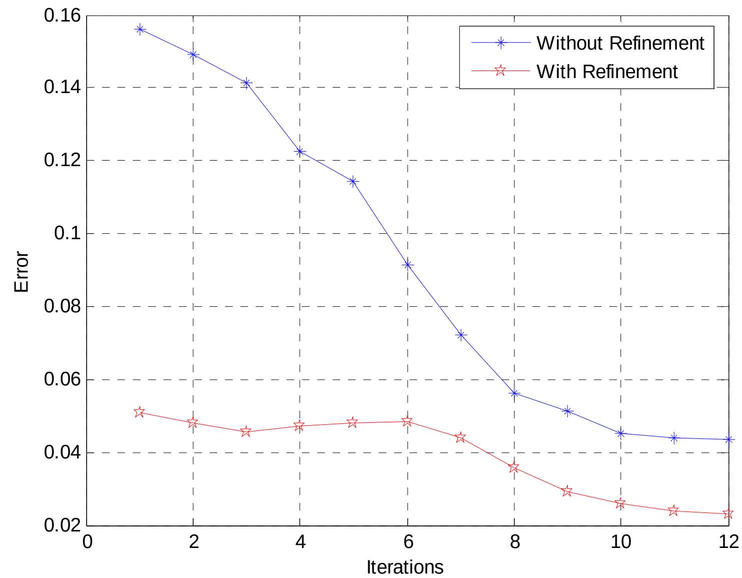

Figure 15: High Impact of Refinement Step on Localization error

Though the above explanation is quite self-supporting, we need to equip our reasoning with better explanation. For this purpose, the two processes: with and without refinement step are analyzed from the computation time point of view. The following figure 16 details the analysis:

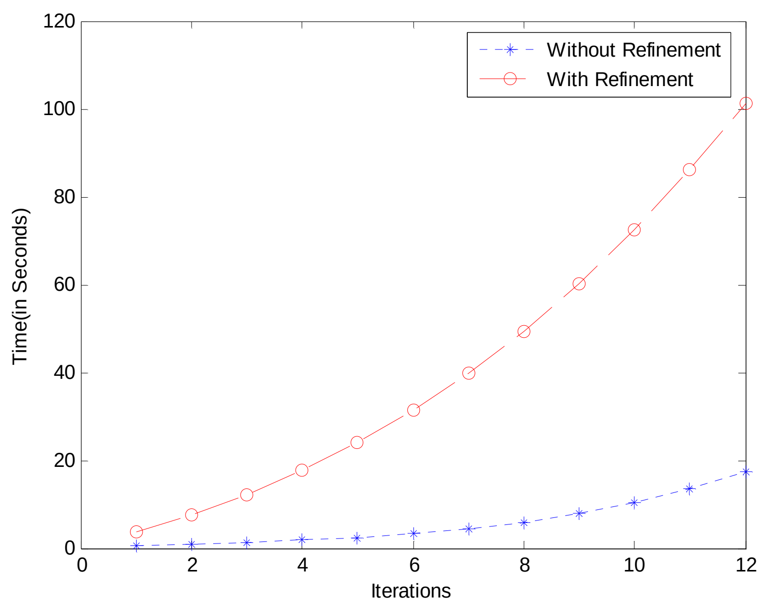

Figure 16: High Computation Overhead of Refinement Step

Now we have a better explanation as to why the refinement step can be excluded. The vital concept behind the DMDS is the addition of virtual nodes by which the network literally grows with the number of iterations. With the increase in the number of nodes, the refinement step takes bigger and bigger chunks of computation time. After a reasonable number of iterations, we realize that it is the biggest overhead even when compared to the basic concept of Classical Multidimensional scaling or the Floyd's algorithm. Hence this is not a good indication and we predict that it gets worse as the actual number of nodes in the initial deployment increases due to application requirements. Hence from now onwards, the refinement step is eliminated from the succeeding procedures.

3. Results of the Dynamic MDS and the Dynamic MDS with sensor fusion.

The results of the Classical MDS followed by the absolute mapping gives the first estimate of the initial node locations, based on which the Dead Reckoning fusion is done. Thereon, for every iteration we have two estimates for the locations of the nodes, one from the DMDS and the other one form the Dead Reckoning fusion. The final result is an ensemble result or the combination of these two results, which seems to do a better job even when the Dynamic MDS gives inconsistent results.

As explained earlier, the results of Dynamic MDS indicate a notable decrease in the mean localization error as the iterations increase. But this is not consistent and we see that there might be unexpected shoot-up of error for some iterations. This is evident from the two sample results which indicate an increase in mean localization error for some of the iterations.

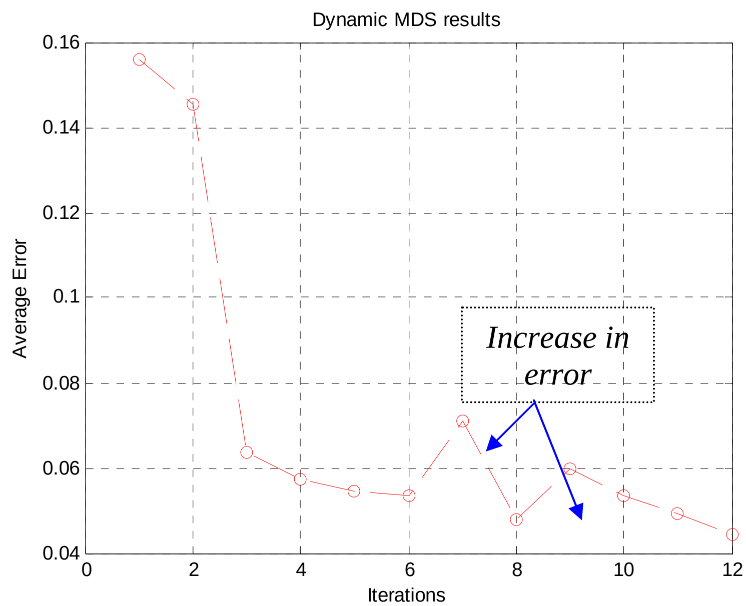

Figure 17: Inconsistency in Dynamic MDS

The figure 17 indicates that the final settling error is significantly lower as compared to the mean localization error in the first iteration. But there is an inconsistency at iteration 7 and 9 in this example. By using the sensor fusion technique, we can demonstrate the positive effects. This is explained in the figure 18.

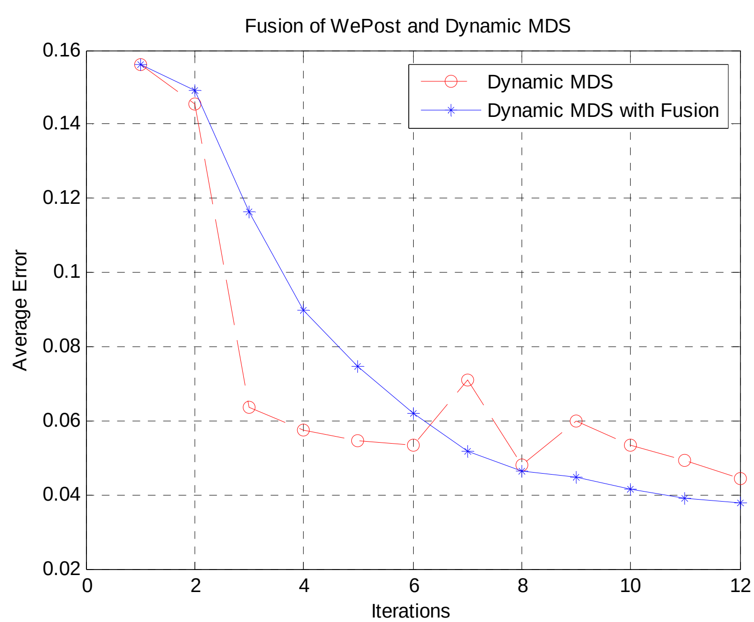

Figure 18: Improved performance of Dynamic Localization over Dynamic MDS

The figure 19 depicts another set of results and here also we can notice that the inconsistency in the error pattern can be minimized by using sensor fusion which can reduce the shoot up of error. The red curve with ‘o’ markers is the Dynamic MDS error plot and the blue curve with ‘*’ markers is the error for Dynamic MDS with Sensor fusion.

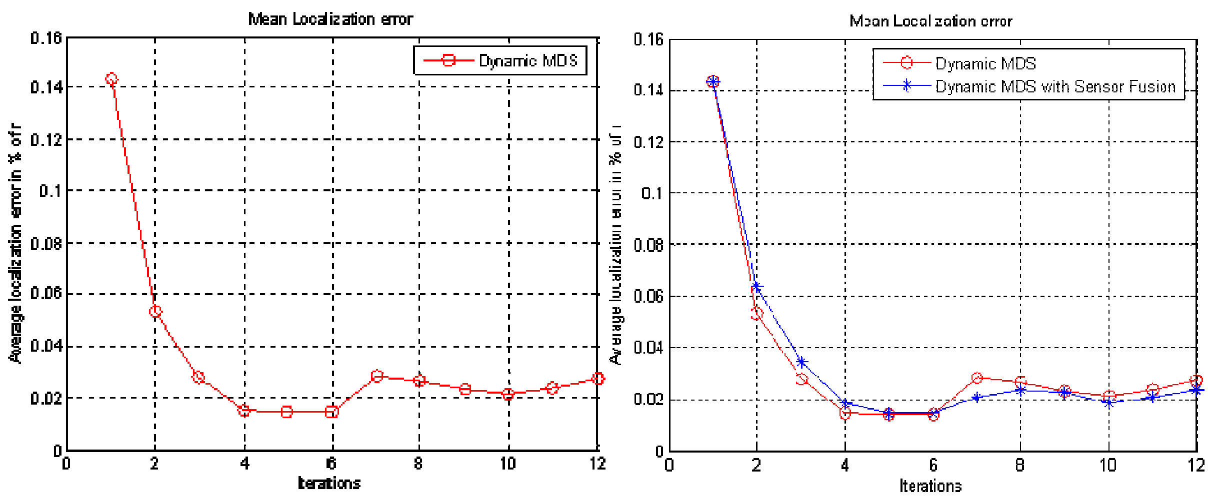

Figure 19: Comparison of performance of Dynamic MDS with /without sensor fusion

Looking at these results, it might be hard to accept that sensor fusion has substantial impact. But the following results can give better insight as to why the fused results prove useful. It is observed that the fused results demonstrate better efficiency when compared to the results of the Dynamic MDS localization algorithm.

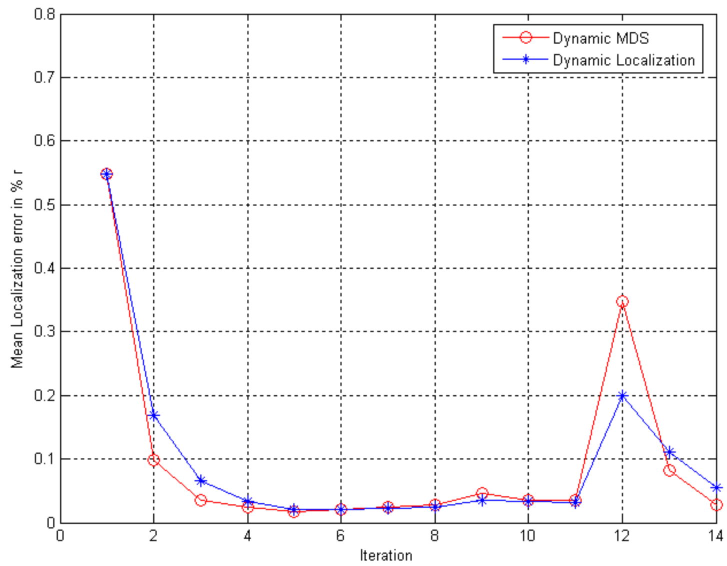

Figure 20: Extreme result in Dynamic MDS

In Figure 20, it is clearly visible that the Dynamic MDS is absolutely bad for the 12th iteration whereas the Dynamic MDS with Sensor fusion suppresses the error by about 50% which is a significant improvement. From this view point, we can rely on the sensor fusion results with a compromise in the initial few iterations. But in the next chapter, we come up with another methodology where we improve even the sensor fusion estimate to always come with better location estimates.

4. Comparison of results for various percentages of mobile nodes.

A meaningful conclusion can be made about the number of nodes and the mean Localization error if we have a common platform to compare the respective situations. This can be simulated by asking different percentages of mobile nodes per iteration and then comparing the mean errors. The following figures are three plots for three different percentages of mobile nodes:25%, 50% and 75%. Care was taken to have similar initial deployment at the beginning without which the comparisons would not be sensible. This makes sure that the initial deployment properties are all same for the three situations. The respective Dynamic MDS results are compared in the Figure 21 and those of the Dynamic MDS with sensor fusion are compared in Figure 22.

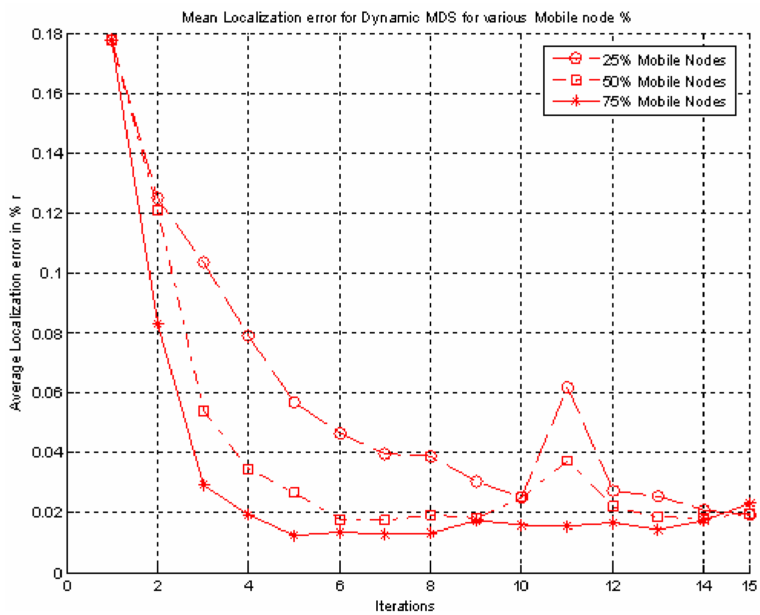

Figure 21 : Dynamic MDS for 25%, 50% and 75% Mobile nodes: Better performance with high %

From the Figure 21 we observe that as the number of mobile nodes increases, we have a better Dynamic MDS result which is of direct implication as we know that the connectivity increases with more mobile nodes. This can be visually felt by looking at the 11th iteration, where the shoot-up of mean error in 25% plot is gradually diminished over 50% and 75% plots.

Now that we know that the Dynamic MDS has indicated an improvement, we focus our attention onto the Dynamic MDS with sensor fusion.(after all we have decided on relying on it more than the results of Dynamic MDS). From the Figure 22, we observe that the fused results have exhibited a noticeable improvement over the increase in the percentage of the mobile nodes. This can be explained as follows. As the number of nodes increase, we have better MDS results and hence better initial estimates for the Dead reckoning procedure and over period of time, we can expect better results.

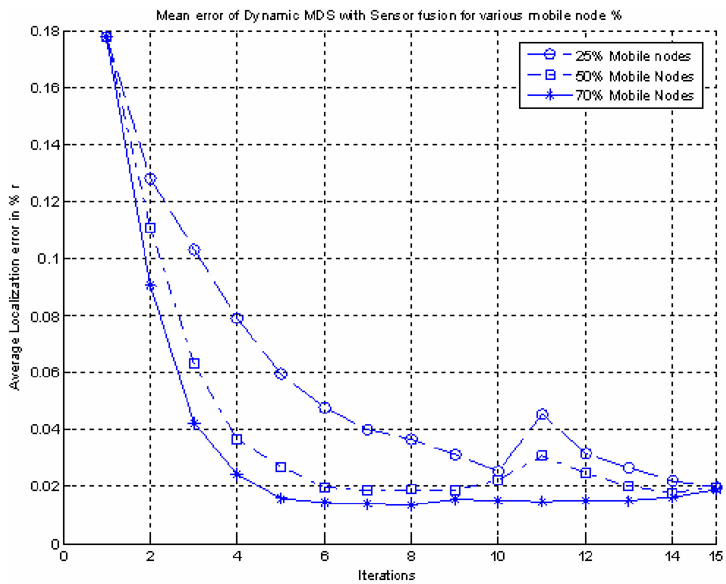

Figure 22: Dynamic Localization for 25%, 50% and 75% Mobile nodes: Better performance with higher mobility

CHAPTER IV

DYNAMIC LOCALIZATION WITH ADJUSTED WEIGHTS

4.1 Introduction

The procedure of applying equal weights for Dynamic Multidimensional Scaling and Dead Reckoning works decently well for most of the time. As explained earlier, it ensures consistency in the final results as compared to the DMDS results. But, from a closer perspective, we realize that this way of equal weights for the two results doesn't provide very satisfactory results all the time.

There is no denying that the previous equal weights work out to provide results better than DMDS, but they could be made even better. For instance when the DMDS yields an abnormally large error and on the other hand if Dead Reckoning yields a low error, equal weighted results force us to get satisfied with a midway result, even though we have better results from Dead Reckoning based localization. On the same note, we might end up with other way when we have better results from DMDS and unsatisfactory results from Dead Reckoning. This is very much visible in the Figure 22.

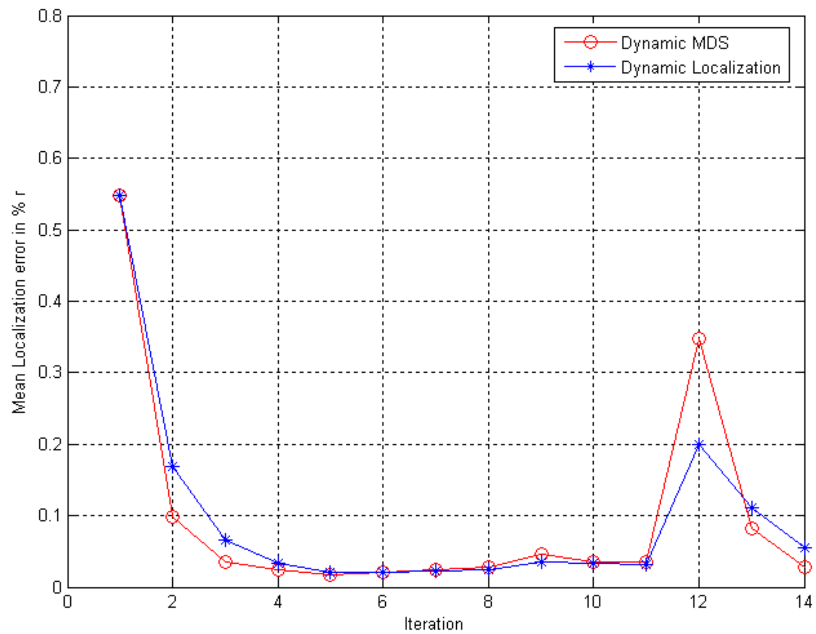

Figure 23: Extreme result in Dynamic MDS

The Figure 23 is still in compliance that equal weights resulted in better results but not the best ones. As one could observe, for the 12th iteration we could have achieved better results if Dead-Reckoning results were given more weight than the Dynamic MDS results. Similarly for the iteration 14, Dynamic MDS outperformed the Dead-Reckoning, but due to equal weights it was not adequately utilized in the final result. This calls for the need of modifying the Dynamic Localization algorithm to always come up best results at any time.

The first and the foremost step is to identify the parameters which allow us to appraise the performance of the DMDS and thereby providing a methodology to adjust weights dynamically to approach the better results out of DMDS and Dead Reckoning. This section of the thesis throws sufficient light on these issues.

The first part of the problem is to identify the parameters which are proved to reflect the performance of DMDS. This is a vital subject as this also means that we have to identify those issues which can potentially contribute to the total localization error. The following step would then relate these parameters with the Dynamic MDS mean error. This essentially means that we can now predict the patterns in the Dynamic MDS error approximately thereby giving us a chance to apply suitable weighting method.

4.2 Measures to Estimate the Performance of DMDS

The results of Dynamic Multidimensional Scaling are to be analyzed in depth to uncover the underlying trends and patterns of dominant quantitative attributes of DMDS. This requires a lot of experimental results which are profiled to deduce some association rules, validated from statistical standpoint. To start off with, one has to identify the crucial issues which can contribute to the error.

From what was explained in the earlier sections of the thesis, the first step before executing classical Multidimensional Scaling is to find out the shortest path distances between all pairs of the nodes with the help of Floyd's Algorithm. For those of the nodes which fall out of the range, Floyd's algorithm returns the distance of the shortest path traversed between the two nodes. As long as the network is reasonably connected, this estimate is fine. But it turns out to be a point of concern when the topology of the nodes leads to huge errors in the shortest path distance estimate. This in turn leads to erroneous inputs to classical Multidimensional scaling which outputs corresponding placement of points that closely resemble the erroneous distances. Hence the classical Multidimensional scaling yields unreliable results. It turns out that this error is quite evident from the non-zero eigen vectors corresponding to third and higher dimensions required to locate the nodes.

The above mentioned erroneous estimates are used for the absolute mapping procedure to obtain the absolute locations of all the nodes. The relative positions of the best three beacons are used in forming a transforming function and obviously this leads to further misrepresentation of the data points.

Now that the potential causes of the error are identified, the step deals with the quantification of the same, through which we always deduce important conclusions on the performance of the DMDS. The following section deals with the three important measures identified in appraising the performance of the localization algorithm in the view of above mentioned errors.

3.2.1 Proportion of Unexplained Variance

When the relative locations of the N number of nodes are derived from their respective inter-node distances, theoretically N-1 dimensions are needed to represent the node locations to stand in compliance with the inter-node distances.[22] But not all these dimensions are actually useful. The eigen value associated with a dimension is indicative of the usage of that dimension in differentiating the node location. For instance, let us consider a simple example. Let 100 nodes be dispersed in a 2-D plane. If the true inter-node distances of these 100 nodes are input to MDS, then it will be observed that the eigen values of the first two dimensions appear significant and the rest of the eigen values are not dominant (ideally zero).

However in practical applications the ranging errors are inevitable. Moreover, depending on the network topology, the Floyd's algorithm also leads to false representation of the inter-node distances. For the above mentioned 100 nodes in 2-D plane, MDS tries to forcibly use extra dimensions to truly match the input distances. This is seen by the presence of non-zero eigen values for the higher order dimensions. Clearly this is indicative of error. When we suppress the higher order dimensions, we tend to lose some information and hence the inter node distances no longer match the input NXN distance matrix. Since we are already aware that the nodes were dispersed in a 2-D plane, we can infer that the need of 3rd and higher order dimensions raises concern and if this contribution of non-dominant eigen values is quantified, we can get a feel of the data representation error by MDS. This is the first measure to gauge the performance of the localization algorithm.

We have a direct measure to quantify the explained variance using p dimensions out of the possible n-1 dimensions. This measure is give by:

$$\text{Proportion of explained Variance} = \sum_1^p \lambda_i \Big/ \sum_1^{n-1} \lambda_i \qquad (22)$$

In an ideal situation, we need only two dimensions for the above example and hence using just the first two eigen values we would have been able to explain 100% variance. The counterpart of this measure gives the percentage of unexplained variance. Referring to this counterpart measure as 'Proportion of Unexplained Variance'(PUV), the following mathematical expression is derived to quantify the contribution of the superfluous dimensions which is essentially caused due to erroneous inter-node distances.

$$\text{Proportion of Unexplained Variance} = \frac{\text{Sum of eigen values corresponding to 3 or higher dimensions}}{\text{Sum of all the eigen values}}$$

In mathematical expression,

$$PUV = \frac{\sum_{i=3}^{i=n-1} \lambda_i}{\sum_{i=1}^{i=n-1} \lambda_i} \qquad (23)$$

where λ refers to eigen value for respective dimension. This equation gives the proportion of unexplained variance and it denotes the contribution of the higher order dimensions in the estimation made by the MDS algorithm. Ideally this figure should be close to zero and significantly high value cautions about the probable misrepresentation involved in the localizing process.

While some of the applications just need relative positions of the subjects involved, most of the other common applications necessitate on absolute locations. This calls for a suitable transformation function to map the relative locations to absolute locations and as we shall see in the next section, this leads to impending errors.

3.2.2 Position Error of Beacons

Beacons play a major role in the latter part of the localizing technique. A percentage of nodes have GPS equipment through which they can independently estimate their coordinates. Since we also have their relative positions from MDS results, we can compute a transformation function (involving translation and rotation) to best conform the relative locations to the absolute locations. The derived transformation function is then applied on the rest of the relative positions to obtain the actual coordinates of all the nodes.

The presence of distortion in the MDS relative estimates impacts the transformation functions also. Due to this there could be two possible issues. Firstly, the lateral translation of the beacons might not conform to the actual locations and hence there might be a finite difference between the beacon positions even after their absolute mapping. Clearly this measure indicates the possible error that will be introduced in all other nodes too. Hence it is vital to measure this difference which is important in forecasting the error. The mathematical expression used to estimate this difference is:

$$\text{Beacon Lateral Error} = \frac{\text{Total Difference in Predicted and Actual Beacon Locations}}{\text{Perimeter of the Triangle formed by the Actual Beacons}}$$

In mathematical expression,

$$BLE = \frac{\sum_{i=1}^{i=3} \delta d_i}{\sum_{i=1}^{i=3} l_i} \tag{24}$$

where δ_i(i=1, 2, 3) is the distance between the actual coordinates and the estimated coordinates for the i^{th} beacon.

Apart from the error in the translation component of the transformation function, there can be error in the rotation component too. At first glance this might seem to be a direct outcome of the presence of error in translation. But from a closer look, we realize that for the same measure of translation error, there could be many ways in which the estimated triangle differs from the actual triangle. Hence it becomes imperative to capture this angle information as it can be of substantial help in future.

$$\text{Beacon Rotational Error} = \text{Sum of the three offset angles in Beacon triangle}$$

In mathematical expression,

$$BRE = \sum_{i=1}^{i=3} \sum_{j=1}^{j=3} \theta_{ij} \tag{25}$$

where θ_{ij} is the difference in slope (expressed in degrees) of the line joining the locations of i^{th} and j^{th} beacons (i,j=1, 2, 3).

With regard to the beacon estimation, the above mentioned two measures can aptly describe the error that will be introduced when the absolute mapping process is used.

All the three variables i.e., Proportion of unexplained variance, Beacon Lateral error and Beacon Rotational error are together called as performance parameters as they can measure the performance of Dynamic MDS.

3.3 Correlation Analysis

At this juncture, we have an intuition that the above mentioned three measures are adequate to assess the performance of DMDS. In other words, all of these variables share a relation with the mean localizing error of DMDS method. Though this has been established through numerous simulations, the procedural method of proving their relationship requires the back up of statistical explanation. In other words, we need to adopt a methodology through which we not only illustrate the existence of a linear relation, but also quantify it with a numerical figure. But this doesn't imply that we consider complex data fitting procedures which are not generic and which are tough to comprehend unless we have a preconceived notion. To keep the matter simple, it is a better way to examine the variables for a simple linear relationship.

Pearson Correlation Coefficient [27] is one such apt measure to determine the linear relationship between a pair of continuous variables. It is a popular tool used in most of the situations when a researcher tries to conceptualize the outcomes of the experimental results.

The formula for the Pearson's Coefficient for a pair of continuous variables x and y over N observations is given as follows:

$$\rho = \frac{N\sum xy - (\sum x)(\sum y)}{\sqrt{[N\sum x^2 - (\sum x)^2][N\sum y^2 - (\sum y)^2]}} \tag{26}$$

This correlation coefficient (ρ) ranges between -1 to 1. As a general rule of thumb, a value in the range of -0.7 to -1.0 and +0.7 to +1.0 indicates a strong association, -0.3 to -0.7 and +0.3 to +0.7 indicates a medium association and anything in the range of -0.3 to +0.3 is indicative of weak or no relationship.

Many tests were conducted to confirm the validity of linear relationship of the variables with mean error and it has been statistically proved that the above mentioned three measures of DMDS performance share a strong positive association with the mean localizing error. (Pearson's Correlation coefficient: +0.7 to +1.0) Hence they are good predictors of the mean error in localizing and this will help us in real-time computation of the weights for the DMDS and Dead Reckoning results.

3.4 Regulation of Weight Adjustment

This section forms the crux of the final phase of the research work. Now that the DMDS is evaluated in terms of the impending errors in the representation of the data, we can use this information to adjust the weights between the results of Dynamic Multidimensional Scaling and Dead Reckoning estimates. It is anticipated that this method of real-time adjustment of the weights will yield better results when compared to the simple 1:1 weight ratio that was adopted in the previous phase of the research work.

The initial step was to substantiate the presence of linear relationship between the three performance parameters through the use of Pearson's Correlation coefficient. The simulation results confirmed this and thereby the next issue comes into light. The weights are to be chosen dynamically for the Dynamic MDS and Dead Reckoning based on the pattern of the three measures identified: Proportion of unexplained variance, Beacon Lateral Error and Beacon Rotational Error.

A simple methodology is adopted. From the patterns of the performance parameters, a weight is estimated by each of them through the measurement of the amount of increase or decrease in the variable as the iteration progresses. The three weights are then averaged to get the final weight which is used for the DMDS result. The expressions that describe the explained procedure are as follows:

The weight for Dynamic MDS as estimated by the variable Proportion of unexplained variance is given by

$$\text{w\_mds1} = \frac{\text{Proportion of unexplained variance in last iteration}}{\text{Proportion of unexlained variance in current iteration}}$$

For kth iteration, this can be expressed in mathematical terms as follows

$$w\_mds1(k) = \frac{PUV(k-1)}{PUV(k)} \tag{27}$$

Whenever Dynamic MDS shoots up in mean error the Proportion of unexplained variance in the current iteration goes up and this would bring down the weight assigned to the Dynamic MDS.

The weight for Dynamic MDS as estimated by the Beacon Lateral error variable is given by

$$\text{w\_mds2} = \frac{\text{Beacon Lateral error in Last Iteration}}{\text{Beacon Lateral error in Current Iteration}}$$

For kth iteration, this can be expressed in mathematical terms as follows

$$w\_mds2(k) = \frac{BLE(k-1)}{BLE(k)} \tag{28}$$

The weight for Dynamic MDS as estimated by the Beacon Rotational error variable is given by

$$\text{w\_mds3} = \frac{\text{Beacon Rotational error in Last Iteration}}{\text{Beacon Rotational error in Current Iteraion}}$$

For kth iteration, this can be expressed in mathematical terms as follows

$$w\_mds3(k) = \frac{BRE(k-1)}{BRE(k)} \tag{29}$$

A sudden increase in any of Beacon related error variable indicates that the Dynamic MDS error might shoot up. Essentially this error would be contributed by a compromised transformation function used to get the absolute locations.

The overall weight estimated for the Dynamic MDS is an average of all of the above three weights which were calculated by the rate of changes in each of the three performance parameters

$$\text{w\_mds} = \frac{\text{w\_mds1} + \text{w\_mds2} + \text{w\_mds3}}{3} \tag{30}$$

The weights for Dynamic MDS and Dead Reckoning are inversely related. This is because, if we have analyzed that Dynamic MDS has performed well, then we don't have a reason to rely more on Dead Reckoning and vice versa. This way, the weight for Dead Reckoning is estimated as follows:

$$w\_dr = \frac{1}{w\_mds} \tag{31}$$

By computing three different weights and taking their average, we assure nearly accurate results. It is to be understood that we can never assure that the three measures devised can capture the trend in mean localizing error accurately all the time. In other words, we cannot expect that the plots of all the three variables exactly mimic the mean error. There is a significant probability that at times, one of the three variables can indicate a false pattern which might generate corresponding misappropriate weight. In such a situation the weights from the other two variables can sufficiently neutralize this negative effect. On the same note, it should be mentioned that all the three variables go wrong only in a very rare event, which could be accounted to the randomness that exists in every type of experimental results.

3.5 Results and Analysis

Experimental Set-Up

This chapter is an extension of previous chapter and hence they should have commonalities so that the final results are comparable. For this purpose, we take care so that the initial deployment is similar to that of previous chapter. Hence we still deploy 100 mobile nodes in the field of predetermined dimensions (in this case, a 5r-by-5r square) where 'r' is the unit length of the placement area with communication radius being 1.0r. In summary, all the characteristic features of the network are retained to make the results comparable.

As in Dynamic Localization algorithm we get two estimates: one from Dynamic MDS and the other from Dead-reckoning technique. But then the final result was an ensemble result with equal weights to both these estimates. In Dynamic localization with adjusted weights, we demonstrate the selection of appropriate weights which are not fixed. This selection is done dynamically by assessing the performance of the Dynamic MDS results in real-time and then estimating corrective weights.

3.5.1 Appraisal of Dynamic MDS Performance:

Three important parameters were identified to capture the patters in the mean localization error of Dynamic MDS: Proportion of unexplained variance, Conformation error in Beacon mapping measured by Lateral Difference and Rotation Difference. Their relationships with the mean error of Dynamic MDS can be substantiated by employing correlation analysis. Since continuous variables are involved we can use Pearson's Correlation coefficient as explained in earlier sections.

Correlation of Proportion of Unexplained Variance to the Mean Error

It is through obvious understanding that these two variables are related to each other. We can get a feel of this statement by looking at the dominating patterns in the plots of mean localization error and the Proportion of Unexplained Variance.

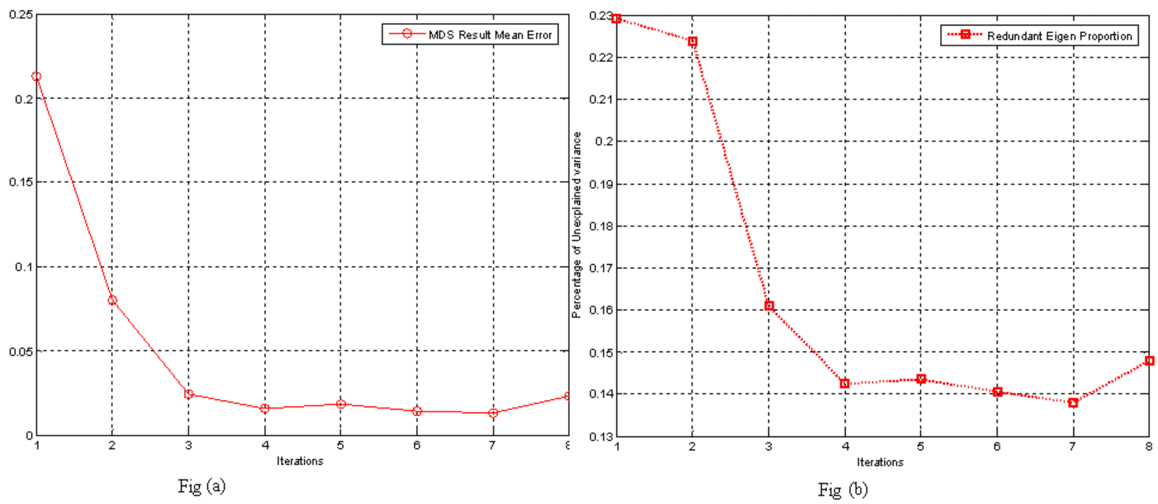

Figure 24: Positive Correlation of proportion of unexplained variance

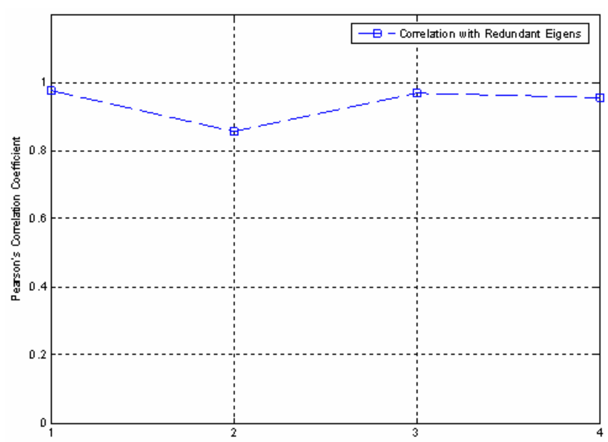

Figure 25: High Pearson correlation values for Proportion of Unexplained Variance

Consider Figure 24. The Figure (a) is the error from Dynamic MDS. The Figure (b) is the plot of unexplained variance.. By looking at the two plots, it is quite evident that the proportion of unexplained variance curve is able to capture most of the notable changes in the DMDS. It can be observed that the Mean localization error slightly increases for the 8th iteration in Figure (a). We can see a similar increase in the Figure (b) for the 8th iteration.

Now we turn our attention towards explaining the relation in the light of Statistics. For the demonstrated graphs, the correlation coefficient works out to be about 0.93 which indicates a high degree of association. Figure 25 shows the variation in the correlation coefficient of proportion of unexplained variance with the mean localization error for 4 simulations. It can be seen that all the four are positive and greater than 0.7. Hence we can conclude that the patterns in proportion of unexplained variance can be used to predict the pattern in the total localization error.

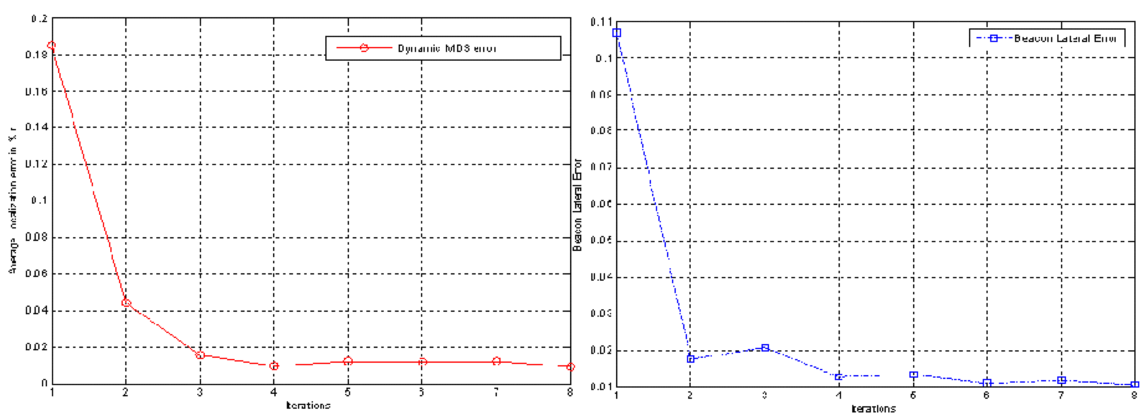

Fig (a) Fig (b)

Figure 26: Positive correlation of Beacon Lateral Error with localization error

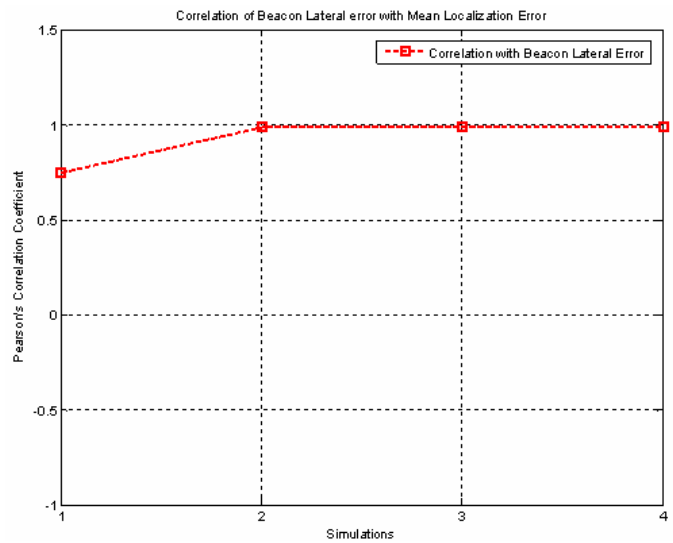

Figure 27: High Pearson values for Beacon Lateral Error

Similarly the errors which are associated with the absolute mapping can also be associated with the mean localization error. Figure 26 and Fig 28 compare the trends of the Dynamic MDS error with those of Beacon Lateral error and Beacon Rotational error respectively. The correlation analyses for 4 simulations also indicate that the strength of linear association of these errors with the mean error is relatively stable (always positive). This is evident form figure 27 and 29.

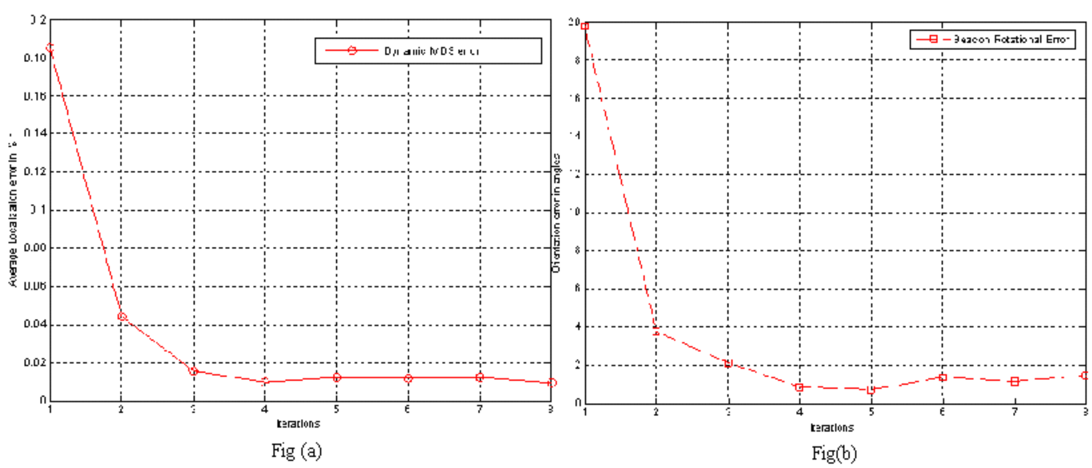

Figure 28 : Positive correlation Beacon Rotational Error with localization error

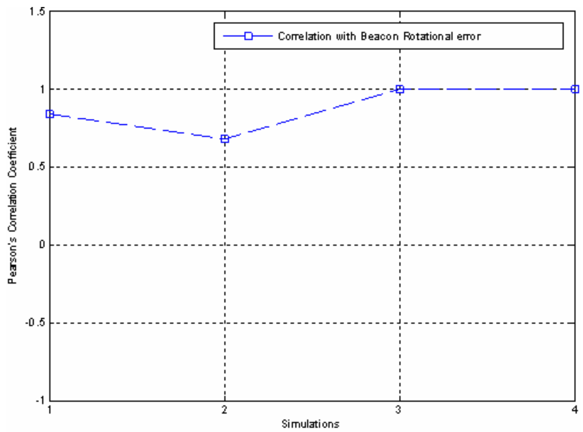

Figure 29 : High Pearson's values for Beacon Rotational Error

At this juncture we might see a potential concern from s statistical view point. The above three variables which exhibited strong correlation with the mean error, are correlated with each other as well. This situation is often referred to as multi-correlation. In simpler words, it indicates that using one of the three variables would be good enough.

But we can prove that such step would adversely affect the model. This is because, each of the three variables show some sort of inconsistency at times. By averaging the weights obtained from all the three variables, we have an opportunity to dampen such inconsistencies. Hence we shouldn't rely entirely on a single variable which would over fit the model to pattern of that variable which would then follow the inconsistencies in the model as well.

3.5.2 Performance of Dynamic localization with adjusted weights

The figure 30 is the same figure which was used in the beginning of the chapter to explain the need of modifications to the Dynamic Localization algorithm. Now the figure had an additional graph of Dynamic localization with adjusted weights to compare the two versions of the algorithm.

In our previous explanation, the iteration 12 was a cause for concern due to unsatisfactory performance of Dynamic Localization with equal weights to Dynamic MDS and Dead Reckoning. Now it is quite evident that the real time weights to the two estimates helped the model to give more weight to the Dead Reckoning method for iteration 10 which brought down the error considerably.

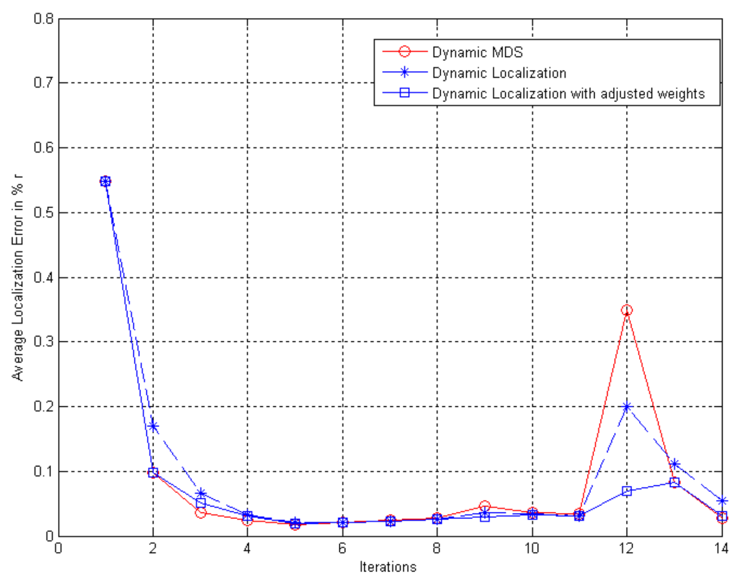

Figure 30: Reduced Localization error with Dynamic Localization with adjusted weights

Working backwards we observe that this result was possible due to the similarities in the pattern of mean error with the three specified variables. Whenever any one of the three measures indicates any unpredicted behavior, the other two variables dampen such effects to a comparable degree. This way the modified algorithm adjusts the weights from each of the three variables during run time.

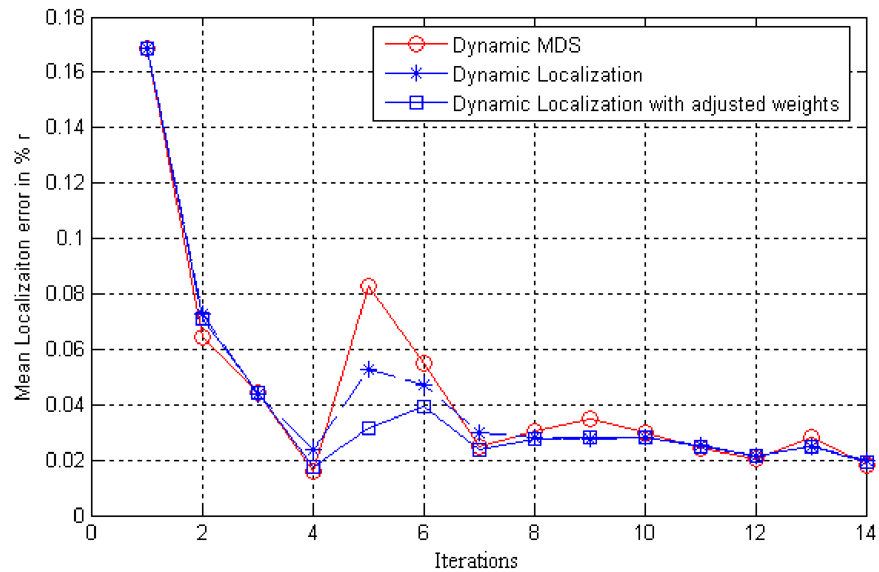

Figure 31: Reduced Localization error with Dynamic Localization with adjusted weights

Figures 31 and 32 depict two other situations where the Dynamic localization with adjusted weights outperforms the Dynamic Localization algorithm, which reflects the stability of the proposed algorithm.

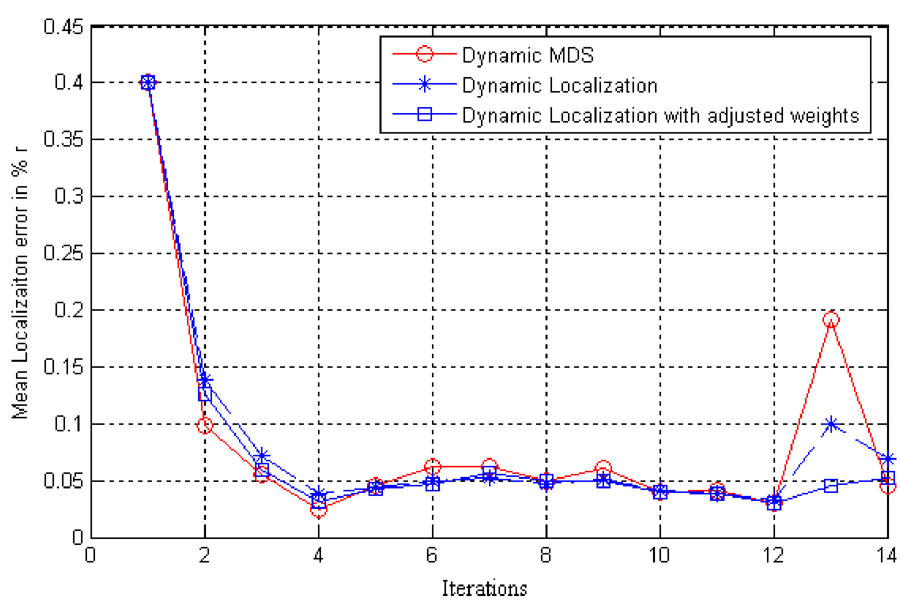

Figure 32: Reduced Localization error with Dynamic Localization with adjusted weights

Time complexity: The Floyd's shortest path algorithm records the maximum share of execution time at every step. It is a significant contributor of computation overhead and it builds up very quickly with the number of virtual nodes being added.. In general, the time complexity is $O(n^3)$. The following table enlists the time complexity introduced by running a simulation with 100 nodes with 75 mobile nodes in each iteration. We see that the computational time increases till the network reaches the set limit of connectivity level(5th iteration) after which the addition of virtual nodes is halted Thereafter the execution time is relatively constant.

Table 2 : Time Complexity

Number of nodes (Real+Virtual)	Time in seconds
100	0.859
175	1.672
250	2.844
325	6.688
400 (required connectivity met)	11.765
400	11.969
400	11.844

CHAPTER V

CONCLUSIONS

4.1 Contributions

The thesis work attempts to solve the localization problem for random deployment of mobile sensors in wireless environment considering all of the given constraints on the deployment.

With Classical Multidimensional Scaling being the crux of the entire work, adequate studies were conducted to understand the working of the algorithm and the fundamental mathematical backing involved. With the range of solutions offered by this algorithm to many of the inter-disciplinary issues, it was then decided that it can significantly contribute to the localization problem as well.

Initially, the thesis focused on validating the applicability of Classical Multidimensional Scaling for localization of static sensor networks. Apart from making an extension to existing work on MDS-based localization, some important conclusions were made regarding the performance issues.

Network topology parameters, node density and precision in measurements are some of the key factors which are found to have significance influence on the performance of the MDS-based localization techniques.

As explained in the first chapter, the admissibility of MDS to localization of wireless mobile sensor networks was the next step to be dealt. With the concept of adding ‘virtual’ nodes to the network, this issue was adequately addressed and the resulting technique was called as Dynamic MDS-based localization. However the simulation results pointed the inconsistency in the overall results. This raised a concern and hence additional work was carried out to finally come with sensor fusion technique that aids the MDS-based localization in bringing down the inconsistency levels. Now the new algorithm is referred to as ‘Dynamic Localization’ technique.

While the above explained Dynamic Localization technique was found to perform fairly well, but not the best. Moreover, it was important to determine and analyze the causes that would dictate the performance of Dynamic Localization technique. Three parameters were found out to be predictive of the mean localization error. Using the patterns in the three parameters, real-time weights are assigned to the Dynamic MDS and the dead reckoning results to come up with final results. The final results of this algorithm with adjusted weights are found out to be much better than those of the algorithm which uses simple 1:1 weights. The simulation results stand in accordance to all of the specified conclusions over a variety of deployment scenario.

4.2 Future Work

The distances, orientation angles of the mobile nodes and other pertinent information are all analyzed by a central processing device (like PDA) in the light of the explained algorithms. In essence, the localization technique implemented in the thesis had one important underlying supposition that the entire process involves centralized computations. While this works for most of the contemporary applications, we foresee the extended usage of the algorithm in forthcoming applications, entailing on the distributed computations. Hence the methodology encompassed in the thesis should significantly accommodate distributed computations. We propose the following to pave way for the future work in the similar lines of this thesis.

We still advocate the usage of popular data visualization algorithms like classical Multidimensional Scaling for the localization of ad-hoc mobile sensor networks. In fact, the DMDS method reinforced by dynamic fusion of the Dead Reckoning estimation can provide a concrete foundation for Distributed Localization Technique too.

The change that would be called for this situation would be in the Dynamic Multidimensional scaling procedure. Entire deployment as a single entity should no longer be used in one-time localization step. Instead, for selected subjects, the one-hop neighbor's information alone is used to create a relative map, which keeps growing with the addition of the relative maps of one-hop neighbors of different nodes. In simpler words, the process starts by selecting a node, which can safely be a node somewhere in the middle of the deployment area. The relative map is generated for the 1-hop neighbors of that node. Then we choose another node and its respective one-hope neighbor relative mapping in such a way that it has a considerable number of common nodes with the central node relative mapping. These two relative maps can be merged together by using the common nodes locations. This process continues till all the nodes in the entire deployment get added to the relative map. Then we have the process of transforming the relative locations into absolute locations with the help of beacons. This would complete first iteration. Thereon, we add virtual nodes to build up the number of nodes and then repeat the procedure of building up the relative map as explained above.

We predict the following vital implications of this Distributed Dynamic localization technique. Firstly, this is a distributed version of the methodology explained in the thesis and hence it offers the common advantages available in any distributed process. On the other hand, we see that this method would be more robust to errors. This is because, in the centralized localizing technique, we saw that the sparse, irregular topologies suffer from misrepresentation of the shortest path distances to 2-hop or any n-hop (n>2) neighbors. Though we have minimized this effect by Dead Reckoning, we still didn't eliminate it. But it seems that the Dynamic Localizing technique is less affected by this as it uses the only 1-hop neighbor distances.

But, there are some concerns about the proposed Distributed Dynamic Localization technique at this point of time. Firstly, it is very obvious that this method is going to take large processing time. This is because, for every patch of 1-hop relative map we need to execute CMDS and then patch it up with the central relative map by mapping technique.

And in general, most of the applications place restriction on computation time apart from their specifications. Also we must have stable transforming function to patch up the relative maps most of time and it is imperative to note that this leaves a considerable scope of errors creeping in the form of bad mapping. Moreover, the concept of adding virtual nodes must be carefully advocated in the distributed technique as it changes the topology of the network very quickly. So these issues are to be dealt carefully in formulating the Distributed version of the algorithm. On a concluding note, the Distributed Dynamic Localization technique can sustain only when we have concrete evidence that setbacks are significantly out-weighed by the advantages and this is possible with extensive simulations with different possible situations can that arise in the real-time deployment scenario.

REFERENCES

[1] Trevarthen, A. 2007. The national livestock identification system: the importance of traceability in E-business. *J. Theor. Appl. Electron. Commer. Res.* 2, 1 (Apr. 2007), 49-62.

[2]http://news.minnesota.publicradio.org/features/2005/04/27_bensonl_registry/ (20/11/2007)

[3] J. Cortes, S. Martinez, T. Karatas, and F. Bullo. Coverage control for mobile sensing networks. In *IEEE Trasactions on Robotics and Automation*, pages 243–255, 2004.

[4] X. Ji and H. Zha. Sensor positioning in wireless ad-hoc sensor networks using multidimentional scaling. In *IEEE Conference on Computer Communications (InfoCom04)*, 2004.

[5] Duncan Smith and Sameer Singh. Approaches to multisensor data fusion in target tracking: A survey. *IEEE Transactions on Knowledge and Data Engineering*, 18(12):1696–1710, 2006.

[6] C. Savarese, J.M. Rabaey, and J. Beutel. Locationing in distributed ad hoc wireless sensor networks. In *Proc. 2001 Int'l Conf. Acoustics, Speech, and Signal Processing (ICASSP 2001)*, volume 4, pages 2037–2040, May 2001.

[7] F. Zhao and L. Guibas. *Wireless Sensor Networks: An Information Processing Approach*. Elsevier and Morgan Kaufmann Publishers, 2004

[8] N. Bulusu, J. Heidemann, and D. Estrin. Gps-less low cost outdoor localization for very small devices. Technical Report 00-729, Computer science department, University of Southern California, Los Angles, CA, 2000 .

[9] X. Nguyen, M.I. Jordan, and B. Sinopli. A kernel-based learning approach to ad hoc sensor network localization. *ACM Transactions on Sensor Networks*, 1(1):134–152, 2005.

[10] A. Savvides, C. C. Han, and M. Strivastava. Dynamic fine-grained localization in ad-hoc networks of sensors. In *ACM SIGMOBILE*, 2001.

[11] J. Beutel. *Master Thesis,Geolocation in a PicoRadio Environment*.UC Berkeley,1999.

[12] Priyantha, N. B., Chakraborty, A., and Balakrishnan, H., 2000, "The cricket location-support system," *Proceedings of the Annual International Conference on Mobile Computing and Networking, MOBICOM*, pp. 32–43.

[13] R. L. Moses, D. Krishnamurthy, and R. Patterson. A self-localization method for wireless sensor networks. *Journal on Applied Signal Processing*, 2003(4):148 – 158, March 2003.

[14] N. Patwari, J. N. Ash, A. O. Hero, R. Moses, and N. S. Correal. Locating the nodes. *IEEE Signal Processing Magazine*, 54, July 2005.

[15] A. Savvides, W. L. Garber, R. L. Moses, and M. B. Srivastava. An analysis of error inducing parameters in multihop sensor node localization. *IEEE Transactions on Mobile Computing*, 4(6), November/December 2005.

[16] T. He, C. Huang, B.M. Blum, J.A. Stankovic, and T. Abdelzaher. Range-free localization schemes for large scale sensor networks. In *MobiCom'03*, 2003.

[17] C. Wu, W. Sheng, and Y. Zhang. Mobile self-localization using multidimensional scaling in robotics sensor networks. *International Journal on Intelligent Control and Systems*, 11(3):163–175.

[18] S. Tilak, V. Kolar, N. B. Abu-Ghazaleh, and K. D. Kang. "dynamic localization control for mobile sensor networks". In *Proceedings of IEEE International Workshop on Strategies for Energy Efficiency in Ad Hoc and Sensor Networks*, April 7-9 2005.

[19] L. Hu and D. Evans. Localization for mobile sensor networks. In *Proceedings of MobiCom'04*, pages 45–57, 2004.

[20] C. Taylor, A. Rahimi, and J. Bachrach. Simulatenous localization, calibration

and tracking in an ad hoc sensor network. In *5th International Conference on Information Processing in Sensor Networks (IPSN06)*, 2006.

[21] Y. Shang and W. Ruml. Improved mds-based localization. In *Proceedings of IEEE InfoCom'04*, 2004.pages 1-12

[22] Multidimensional scaling by Cox, Trevor F, Chapman & Hall**.** ISBN**:** 0412491206, chapters 1, 2 , 3 , 4

[23] Modern multidimensional scaling : theory and applications by Borg, Ingwer, Springer ISBN: 0387948457 Chapters 1,4,5

[24] J.A. Costa, N. Patwari, and A. Hero. Distributed weighted multidimensional scaling for node localization in sensor networks. *ACM Transactions on Sensor Networks*, 2(1), February 2006.

[25] M. J. Atallah. *Algorithms and Theory of Computation Handbook*. CRC Press, 1998.

[26]Feng-Ji, Z., Hai-Jiao, G., and Abe, K. 2004. Localization using combining sensors and dead-reckoning. In *Focus on Computational Neurobiology*, L. Li, Ed. Nova Science Publishers, Commack, NY, 167-176.

[27] www.statsoft.com (20/11/2007)

ACKNOWLEDGEMENTS

1. Members of the thesis committee, Dr Weihua Sheng (Advisor), Dr Rama Kumar and Dr Qi Cheng for their invaluable guidance
2. Department of Electrical and Computer Engineering
3. The entire work group of ASCC laboratory.
4. Dr Yi Shang from University of Missouri at Columbia, for providing a portion of the simulation code
5. My parents, sisters and my friends for being my moral strength all through.

VITA

Ravi Kiran Garimella

Candidate for the Degree of

Master of Science

Thesis: DYNAMIC LOCALIZATION OF MULTIPLE MOBILE SUBJECTS IN WIRELESS ADHOC NETWORKS

Major Field: Electrical Engineering

Biographical:

Personal Data:
Born on January 20th, 1983 in Hyderabad, the capital city of Andhra Pradesh, India.

Education:
Pursued Bachelor of Technology degree in Electrical and Electronic engineering from Jawaharlal Nehru Technological University in May 2004
Completed the requirements for the Master of Science in Electrical and Computer Engineering at Oklahoma State University, Stillwater, Oklahoma in December, 2007.

Experience:
Worked as a Teaching Assistant at Oklahoma State University for an undergraduate level course (ECEN 3113) for a period of three semesters. Also worked as a Research Assistant in Laboratory for Advanced Sensing, Computation and Control. Currently employed as a Computer Lab assistant in CEAT Labs, Oklahoma State University

Name: Ravi Kiran Garimella Date of Degree: December 2007

Institution: Oklahoma State University Location: Stillwater, Oklahoma

Title of Study: DYNAMIC LOCALIZATION OF MULTIPLE MOBILE SUBJECTS IN WIRELESS ADHOC NETWORKS

Pages in Study: 85 Candidate for the Degree of Master of Science

Major Field: Electrical Engineering

Scope and Method of Study: The thesis proposes a localization algorithm that can be used to track and locate mobile subjects in a wireless ad hoc network. An interdisciplinary algorithm called Multidimensional Scaling (MDS) is reconstructed through a variation and the resulting algorithm called Dynamic MDS is used for localization.

The essential idea of Dynamic MDS is to reduce the localization error by addition of virtual nodes till the network turns adequately dense and thereby better connected. The shortcomings were identified in the form of random inconsistencies in the results which were suitably dampened by implementing dead reckoning method enabled by WePosT through sensor fusion technique. This procedure is now referred to as Dynamic Localization procedure which works reasonably better than the Dynamic MDS procedure. While this procedure has exhibited adequate accuracy, we identified some key parameters which can appraise the Dynamic MDS. This further helped the research in revamping the Dynamic Localization algorithm leading to Dynamic Localization with adjusted weights.

Findings and Conclusions: The Dynamic localization with adjusted weights was tried on various deployment situations and it is found out that the results stand in accordance. While this brings a completion to the scope of this thesis, the experience gained through research and the contemporary advancements made in the MDS-based localization techniques provide an opportunity to extend this algorithm to distributed networks which is briefed in the concluding chapter.

ADVISER’S APPROVAL: Dr. Weihua Sheng