Coordinates Determination of Underwater Mobile Sensors Using Cayley-Menger Determinant

Submitted by:

Sadia Zaman

ID:2012-1-60-026

Farihatul Allam Mowsume

ID:2012-1-60-027

Marzia Mohiuddin

ID:2012-2-60-037

Supervised by:

Dr. Anisur Rahman

Assistant Professor

Department of Computer Science Engineering

East West University

A Project Submitted in Partial Fulfillment of the Requirements for the Degree of Bachelors of Science in Computer Science and Engineering to the Department of Computer Science and Engineering

East West University

Dhaka,Bangladesh

Abstract:

Underwater wireless sensor Networks(UWSNs) are usually deployed over a large sea area and the nodes are usually floating, due to their special environment. This results in a lower beacon node distribution density, a longer time for localization, and more energy consumption. Currently most of the localization algorithm in this field do not pay enough consideration on the mobility of the nodes. This paper investigates the problem of localizing submerged sensors that are mobile and provides a new mechanism to determine the coordinates of those sensors using only one beacon node. In underwater wireless sensor networks (UWSN), precise coordinate of the sensors that actuate or collect data is vital, as data without the knowledge of its actual origin has limited value. In this study, the method of determining the underwater distances between beacon and sensor nodes has been presented using combined radio and acoustic signals, which has better immunity from multipath fading. The velocity of unknown node is calculated by using the spatial correlation of underwater object's mobility, and then their locations can be predicted. Moreover, Cayley-Menger determinant is used to determine the coordinates of the nodes where none of nodes have *a priori* knowledge about its location. Simulation results validate the proposed mathematical models by computing coordinates of sensor nodes with mobility with negligible errors.

Declaration

We hereby declare that, this project was done under CSE497 and has not been submitted elsewhere for requirement of any degree or diploma or for any purpose except for publication.

____

Sadia Zaman

ID: 2012-1-60-026

Department of Computer Science and Engineering

East West University

____

Farihatul Allam Mowsume

ID: 2012-1-60-027

Department of Computer Science and Engineering

East West University

____

Marzia Mohiuddin

ID: 2012-2-60-037

Department of Computer Science and Engineering

East West University

Letter of Acceptance

We hereby declare that thesis is from the student's own work and best effort of us,and all other sources of information used have been acknowledged. This thesis has been submitted with our approval.

______________________________ **Supervisor**

Dr. Anisur Rahman

Assistant Professor
Department of Computer Science and Engineering
East West University

______________________________ **Chairperson**

Dr. Ahmed Wasif Reza

Associate Professor and Chairperson (Acting)
Department of Computer Science and Engineering
East West University

Acknowledgement

We begin by praising Almighty Allah since all the praises due to him. It is His mercy that made our work and life easier.

Our sincere gratitude goes to our parents and family members, particularly our mothers, their unbreakable trust in us and continuous support made our life beautiful.

We express special thanks to our friends and seniors whose suggestions and helpful attitude made our thesis more efficient and reliable.

Dr. Anisur Rahman , our supervisor, his considerate and friendly attitude, apt advices, unique direction, and inspiration through the time of our research, helped us a lot.

We thank our teachers of CSE department in East West University because of their helpful attitude.

At last, a grateful thanks for the researches in the field of Wireless Sensor Network (WSN) and Underwater Wireless Sensor Network (UWSN).

Abbreviation and Acronyms

WSN	Wireless Sensor Network
UWSN	Underwater Wireless Sensor Network
GPS	Global Positioning System
LPS	Local Positioning System
RF	Radio Frequency
TDOA	Time Difference of Arrival
TOA	Time of Arrival
RTOA	Roundtrip Time of Arrival

Table of content

Chapter 1: Introduction

Chapter 2: Literature Review

Chapter 3: Proposed Method

List of Figures

List of Tables

Chapter 1

Introduction

The 75% of earth's surface is covered with water in the form of rivers, canals, seas, and oceans. Plenty of precious resources lie underwater which are required to be explored. The key to successful explorations has always been technology dependent. Recent advances in technologies have led the possibilities to do the underwater explorations using sensors at all levels which were not possible previously. Accordingly, underwater sensor network (UWSN) is emerging as an enabling technology for underwater explorations.

1.1 Underwater Wireless Sensor Network (UWSN)

Recently marine rights and interests are receiving more and more attention, since it has witnessed rapid development in many countries. Therefore, researches on UWSNs (Under Water Wireless Networks) has developed quickly, providing basic technical support, such as ocean environment monitoring, resource exploration, natural disaster warning, military defense, health care, mobile communication, surveillance, utilities. Since the majority of the earth’s surface is covered with water, more research is conducted on underwater systems. Data collection and environment monitoring have become major components.

The characteristics of the underwater environment present researchers with many challenges specially developing effective sensor communication and localization techniques.
Since marine life is a vast resource it is therefore crucial to collect accurate data using underwater sensors, and to develop appropriate mechanism for actuation tasks. In addition because of water currents, there is the non-negligible node mobility which may cause frequent changes to the network topology.

Underwater wireless sensor network is a very prominent and interesting research sector in recent years of WSN. Underwater Wireless Sensor Networks consists of a number of sensor nodes, stationary or mobile, connected wirelessly via acoustic communication modules deployed to monitor various events of interest collaboratively. The objective is achieved by having a set of autonomous devices in a network which can self-organize and adapt to deep sea conditions. Underwater communication is mainly done using low frequency and low data rate acoustic modems with a set of nodes transmitting their data to a buoyant gateway that relays the data to nearest coastal monitoring and control station.

The localization of mobile nodes that will collect these data is important and necessary for underwater sensor networks. With random mobility, the sensor nodes could realize the monitoring of the whole area. The sensor nodes located at the sea bed cannot communicate directly with the nodes near the surface level, they require multi-hop communication assisted by appropriate routing scheme. Wireless sensor network consists of some sensing devices that can communicate wirelessly and these devices can process and communicate with its peers.
Depending on the type of deployment, the wireless sensor and actuator network may consist of surface monitoring stations, autonomous underwater vehicles and various environmental sensors . The localization of mobile nodes in this process will help in real time tracking and for identifying the pollutant area. Thus the effective implementation on the area of different sector of our system and management especially in the marine life using WSN localization, it becomes the attraction of the researchers to make it more applicable in different way at low cost. Moreover, the information collection from the underwater devices and transmission are realized by the sensor nodes and the relevant data without location will make no sense, so the localization of mobile sensor nodes has gradually become a research hotspot and focus.

1.2 Challenges for Underwater Wireless Sensor Network

The characteristics of UWSN described above present great difficulties and challenges to the localization of underwater mobile nodes in a large-scale network. Acoustic communication has a bigger propagation delay, lower bandwidth and higher error rate compared to radio communication. This imposes more limits on any localization algorithm. The algorithms based on transmission of tremendous amounts of data or higher real-time communication requirements will not be applicable in this kind of circumstance. In addition, the mobility of nodes makes the topology change frequently.

The algorithms designed for static networks need to run the localization procedure periodically to update the nodes' location, which may cause more energy consumption for communication in an underwater environment. Moreover, the batteries of underwater sensor nodes can almost never be replaced and the energy is strictly limited, so under normal circumstances, it is difficult to achieve higher localization accuracy and localization coverage rate in an underwater environment.

Localization has widely been explored in terrestrial wireless sensor network (WSN). In terrestrial wireless sensor networks, the nodes use Radio Frequency (RF) to establish the communication infrastructure. Underwater sensor networks are quite different from terrestrial sensor networks. In underwater environments, due to water absorption, RF does not deliver the same performance.

If 3D Euclidean distance estimation method is used, it requires the need of a certain number of neighboring nodes to measure inter-node distances and where error is propagated through the system due to its recursive nature. In one method the researchers propose a localization scheme based on buoys moored to the waterbed and mobile nodes that need to communicate directly with these buoys to get their location. This method does not support dynamic environment because buoys need to be deployed in advance in known locations. In [6], four different positions are used to obtain the beacon nodes positions of a 3D local positioning system (LPS).

Another obstacle is sensors' inability to communicate with each other is that the sensors are affected by ocean currents, which have uncertain mobility, so sensors are unable to obtain correct location information of any other sensors in an epoch. All these phenomena result in the entire UWSNs having a high delay, and even a high probability of disruption environment.

1.3 Objective

Aiming at the special environment of UWSN and in mobile node localization, we propose a method based on the Cayley-Menger determinant to find the coordinates of the underwater sensors having the beacon plane and the sensors' made plane in parallel with each other when the sensors are mobile.

Using our method (which is described in chapter 3) at first we will find the coordinates of the sensors from considering one of the sensors coordinate as the origin (0,0,0). Then finally we will compute the coordinates of the sensors in their mobile form i.e. with mobility from the beacon node using Cayley-Menger determinant along with using triangulation and trilateration techniques. As a result we will find the coordinates of the sensors. Following the result what we get from our simulation and will also compare the result with other paper.

1.4 Methodology

Aiming at the special environment of UWSN and the difficulties in mobile node localization, we propose a method based on using Cayley-Menger determinant and compute the coordinates of the sensors. First we measure the distances between the beacon node and sensors. Here, the speed of the sensors are known. Then in this paper, we describe a closed-form solution to determine the coordinates of the underwater sensors having only one beacon node at the surface and the beacon node are assumed to be in a plane, which is parallel to the plane created by the three sensors.

Here we at first calculate the inter sensor distances using Cayley-Menger determinant and compute the coordinates of the sensors with known measurements. Then when the sensors start moving forward in the direction of x-axis we find out their location using triangulation and trilateration techniques. When the sensors and the beacon move forward , we take the sensors back to its previous positions, then we calculate the distance between the beacon's current position to sensors previous position. We then using the previous method we find out the coordinates of the sensors . Then we add the distance that it had covered with its velocity to find out the current actual coordinate of the sensors with respect to the beacon.

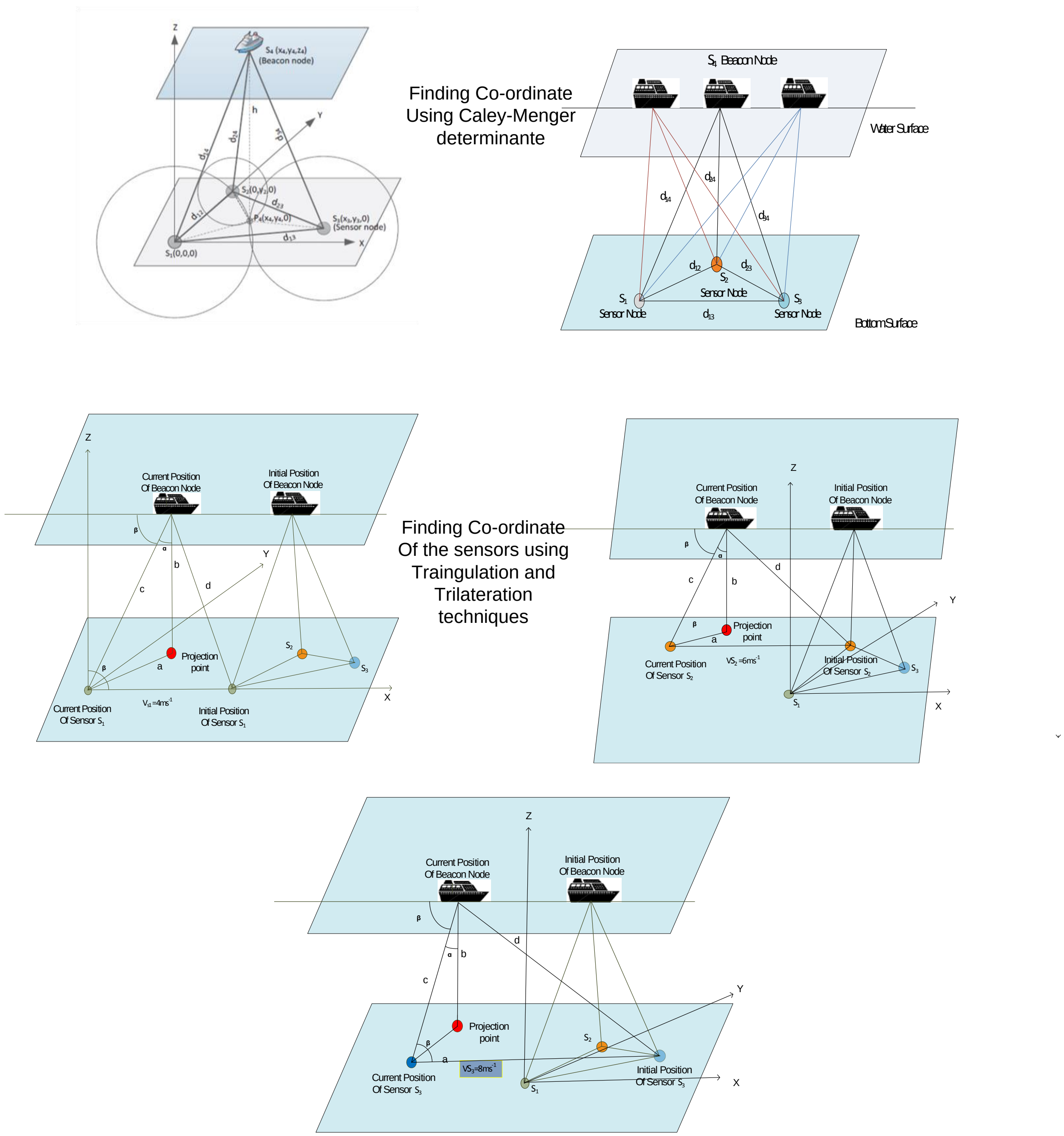

Figure1.1: Finding coordinates of the sensors using proposed method

Chapter 2

Literature Review

The characteristics of UWSN describes about present great difficulties and challenges to the localization of underwater mobile nodes in a large-scale network. Acoustic communication has a bigger propagation delay, lower bandwidth and higher error rate compared to radio communication. Applications of underwater sensing range from oil industry to aquaculture, and include instrument monitoring, pollution control, climate recording and prediction of natural disturbances, search and survey missions, and study of marine life. Wireless information transmission through the ocean is one of the enabling technologies for the development of future ocean-observation systems and sensor networks.

2.1 Related Works

Localization methods for wireless sensor networks can be divided into two types: range-based and range-free methods. The range-based methods such as the received signal strength indicator (RSSI), time difference of arrival (TDOA) and time of arrival (TOA) use hardware to measure the distance information. These kinds of method have a higher accuracy, but they increase the network cost and energy consumption [8]. The range-free methods use the connectivity of the network to locate the unknown nodes. The typical range-free methods mainly include the DV-HOP, Convex Programming and Centroid Localization algorithm. These methods have no additional hardware requirements, and they have lower energy consumption and shorter positioning time, but their accuracy is usually lower [9]

There are many researches on terrestrial nodes localization. In [10] the authors proposed a range-free localization algorithm based on a sequential Monte Carlo localization method. It can exploit mobility to improve the localization accuracy. In [11], a Monte Carlo localization algorithm with mobility prediction (MCL-MP) was proposed, and it can further improve the accuracy by using prediction and filtering for the unknown nodes based on dynamic sampling. Studies on the localization of underwater mobile nodes always face some challenges, and most of them were designed for small-scale networks. For example, GPS Intelligent Buoys (GIB) based on surface buoys and one-hop communication under the water has been proposed. This approach has a high accuracy but the hardware is complex and the cost is high [12]. In [13], a so-called Silent Localization algorithm was proposed, which does not need time synchronization and is applicable to one-hop underwater networks. In addition, in [14], a Scalable Localization with Mobility Prediction (SLMP) method was proposed, and it is closer to localization in an actual environment. The SLMP algorithm has two stages to locate the unknown nodes. First, the velocity of beacon nodes will be estimated by the Durbin algorithm to perform the online linear prediction. The unknown nodes are then located by using the mobility prediction based on the spatial correlation of sensor nodes.

A three-dimensional deployment space is another important characteristic of UWSNs. In [15], an efficient localization scheme which can transform the three-dimensional localization problem into a two-dimensional counterpart via a projection technique was proposed; it can make all the nodes map to the same plane. Because the depth information can be obtained by a pressure sensor, this scheme not only makes the two-dimensional localization algorithm apply in the three-dimensional space, but also simplifies the amount of calculation for three dimensional localization, and reduces the energy consumption of the process. In this paper, we also use this method to project the nodes on a certain plane, and only consider the velocity of the nodes in one direction along with X-axis. The principle of this method is shown in **Figure 2.1**.

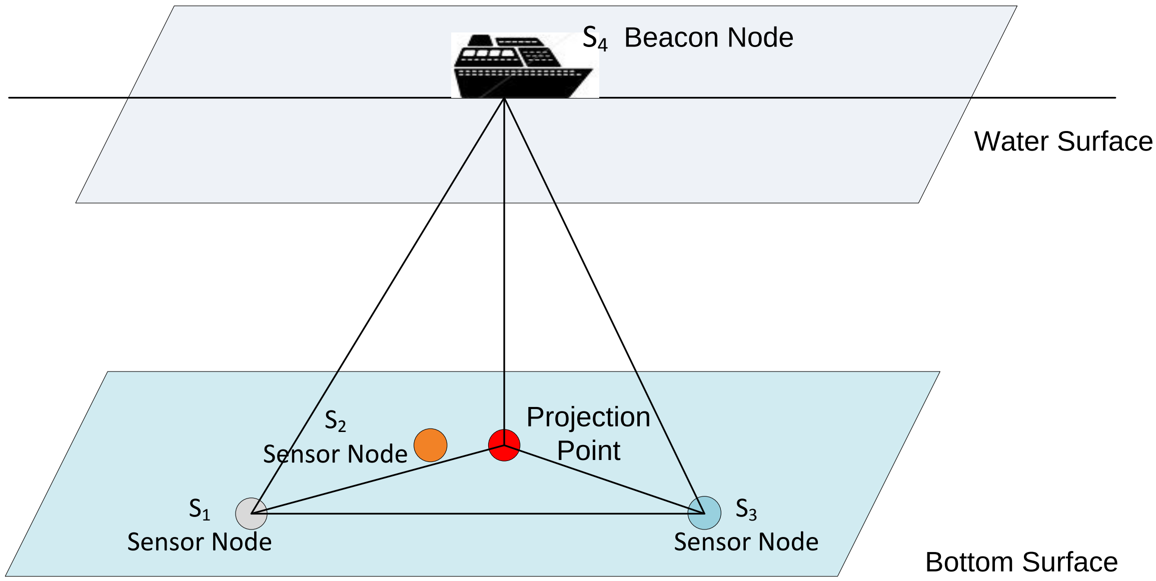

Figure 2.1 A diagram of the projection method

As shown in Figure 2.1, S_1, S_2 and S_3 are sensor nodes in an underwater networks. The projection point of the beacon node(the ship) is shown on the bottom surface of the sea bed. Then we only need to locate the nodes in the surface plane, and the final location can be obtained by simply adding the respective distance information.

2.2 Localization Techniques of UWSN

Localization of sensor nodes is an important aspect in Wireless Sensor Networks (WSNs) .This methods for wirelss sensor networks can be divided into two types

- Range-based approach
- Range-free approach

2.2.1 Range-Based Approach

The range-based methods such as the received signal strength indicator (RSSI), time difference of arrival (TDOA) and time of arrival (TOA) use hardware to measure the distance information. These kinds of method have a higher accuracy, but they increase the network cost and energy consumption.

Received Signal Strength Indicator (RSSI):

RSS is a common technique in localizing sensor nodes; this is due to the fact that almost all nodes have the ability to measure the strength of the received signal. RSS technique benefits from the fact that radio signals diminish with the square of the distance from the signal's source. In other words, the node can calculate its distance from the transmitter using the power of the received signal, knowledge of the transmitted power, and the path-loss model. The operation starts when an anchor node broadcasts a signal that is received by the transceiver circuitry and passed to the Received Signal Strength Indicator (RSSI) to determine the power of the received signal. Since the path loss in underwater acoustic channels is usually time varying and multipath effect can result in significant energy fading, the RSSI method is not the primary choice for underwater localization.

Time Difference of Arrival (TDOA)

TDOA can be measured based on the fact that the distances between the transmitter and different receivers are different. This means that the transmitted signal is delayed in time based on the distance to the receiver. Based on the two received signals, the distance to the transmitter can be determined. However, it is unsuitable for underwater localization because radio does not propagate well in water. Alternatively, the time difference of arrival between beacons from different reference nodes transmitted using acoustic signaling can be used in localization.

Time of Arrival (TOA)

TOA is defined as the earliest time at which the signal arrives at the receiver. It can be measured by adding the time at which the signal is transmitted with the time needed to reach the destination (time delay). The time delay can be computed by dividing the separation distance between the nodes by the propagation velocity. In TOA, the nodes have to be synchronized and the signal must include the time stamp information. To overcome these restrictions, Roundtrip Time of Arrival (RTOA) and Time Difference of Arrival (TDOA) are developed.

2.2.2 Range-Free Approach

Range-free approach, employs to find the distances from the non-anchor nodes to the anchor nodes. This approach use the connectivity of the network to locate the unknown nodes. The typical range-free methods mainly include the DV-HOP, Convex Programming and Centroid Localization algorithm. These methods have no additional hardware requirements, and they have lower energy consumption and shorter positioning time, but their accuracy is usually lower.[17]

There are three basic localization techniques that are used as a base to a more advanced techniques: [21]

1.Trilateration: This method determine the position of a node from the intersection of 3 circles of 3 anchor nodes that are formed based on distance measurements between its neighbours. The radius of the circle is equal to the distance measurement as shown in Figure 1. However, in a real environment, the distance measurement is not perfect; hence, more than three nodes are required for localization.

2. Triangulation: This method is used when the direction of the node rather than the distance is estimated. It uses trigonometry laws of sines and cosines to calculate the nodes position based on the angle information from two anchor nodes and their positions.

3. Maximum Likelihood Multilateration: Trilateration technique cannot accurately estimate the position of a node if the distance measurements are noisy. A possible solution is to use the Maximum Likelihood (ML) estimation, which includes distance measurements from multiple neighbour nodes as in Figure 3. This method intends to minimize the differences between the measured distances and estimated distances.

Three-Dimensional Underwater Target Tracking (3DUT) Scheme

A Three-Dimensional Underwater Target Tracking (3DUT) scheme is also proposed. As shown in Figure2.2, at least three anchor nodes float at the surface of water. One of these nodes is the sink (node *A*) which collects the information from underwater sensor nodes and carries out the calculations. The black nodes collect and send information from the target to the sink. The gray node is the designated projector node. 3DUT is a two phase algorithm. During the first phase, Passive Listening, sensor nodes listen to the underwater environment for potential targets. The second phase of the algorithm, Active Ranging, is to localize the target. 3DUT selects a projector node which sends pings periodically. The target is assumed to be a point target so that the echoes are radiated isotropically. Once the echo is received by the projector, it calculates its distance to the target and transmits to the sink node. Sink node uses trilateration to localize the target. The location and the calculated velocity of the target are then exploited to achieve tracking. Depending on the results of the calculations, sink node selects a new projector node.

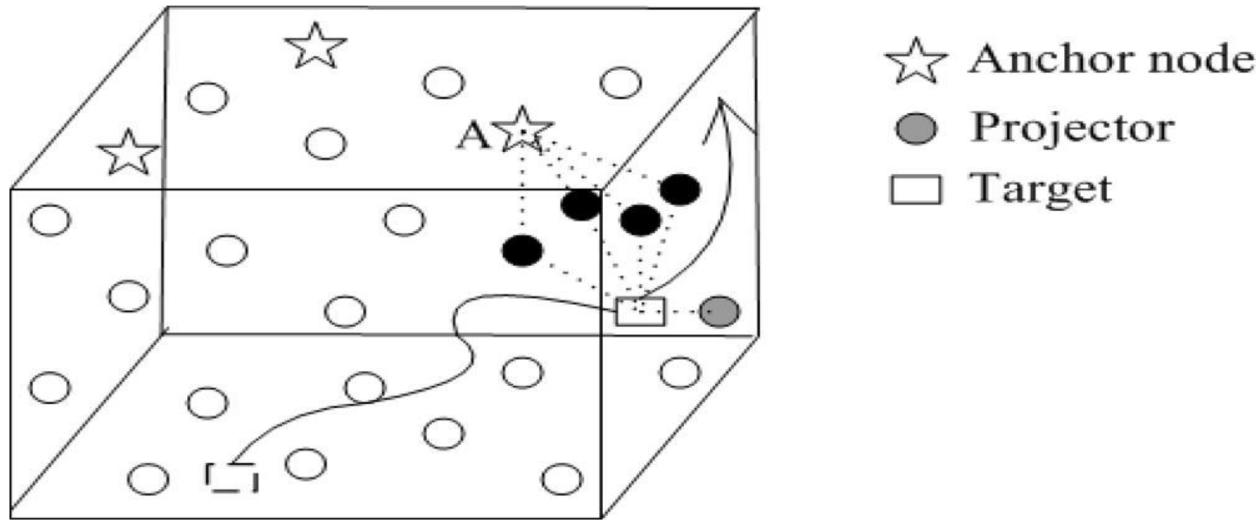

Figure 2.2 The Three-Dimensional Underwater Target Tracking.

To save energy, the nodes which are not located at the network edge have low duty cycles. The nodes which are at the boundary of the sensing region have higher duty cycles in order to detect the target entering into the sensing region immediately. Therefore, to avoid rapid energy depletion of boundary nodes due to continuous surveillance, 3DUT employs an adaptive procedure to find, designate, and activate new boundary nodes. Furthermore, 3DUT does not depend on the number of nodes. The algorithm runs even if the number of sensor nodes changes. However, 3DUT can only track one target at a time. Moreover, the tracking accuracy is heavily influenced by the target's velocity.

Chapter 3

Proposed Method

Research shows that the mobility of underwater object is influenced by water current, temperature and some other factors , so we cannot use a unified model to describe the nodes mobility for all environments. However, under a specific environment, if we specify the nodes' mobility in one direction with known speed we can establish a specific mobility model for this specific environment . Nowadays the UWSN is mainly used in seashore environment, where the water is relatively shallow, usually it is less than 100 m, and the water current is relatively flat, so the nodes' mobility situation is not complex.

3.1 Problem Statement

In this paper, to determine the coordinates of the submerged sensors, our proposed method assumes at least three sensors and a floating beacon. It is also assumed that the distance measurement between the beacon and sensors are possible (which is described later part in this chapter). In the marine environment, a boat or a buoy can be used as a beacon and sensors could be deployed in the water. While measuring the multiple distances between the beacon and sensors, those locations of the beacon are assumed to be in a plane, which is approximately parallel to the plane created by the three sensors (as shown in Figure. 3.1). But if this three sensors are mobile then we will need to find these sensors location using only one beacon.

For finding mobile sensors location from one beacon node ,at first measurements of the distances from six locations of the beacon are taken. As the general properties of a transducer, beacon has the capability of generating radio and acoustic signal, whereas sensors might have the restricted capability of receiving the radio and acoustic signal for timing purpose as well as it is enabled with acoustic transmission. A solvable configuration of one beacon with three submerged mobile sensors is denoted in Figure. 3.2. Our proposed mechanism exploits the advantage of both radio and acoustic signal propagation in sea water in 1.8-323m depth. Since most of the marine explorations take place in shallow water, our proposed model has wide ranging practical applications.

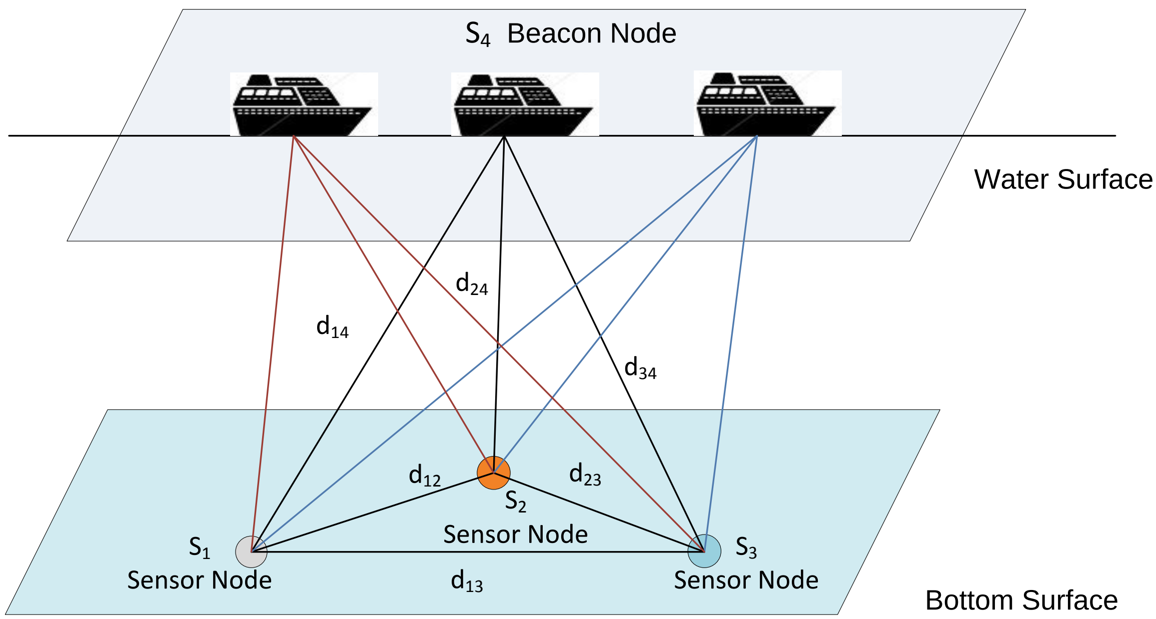

Figure. 3.1 A solvable configuration of one beacon with three submerged sensors

Here the position where the first sensor(S_1) was deployed, we denoted that position as (0,0,0) on x-axis. The position where the second sensor(S_2) was deployed, we denoted that position as (0,y,0) and the position where the third sensor(S_3) was deployed, we denoted that position as (x,y,0). The discussion is elaborated in chapter 3.6 along with the figures.

3.2 Environmental Limitations

Normally, underwater environment is more adverse than terrestrial environment; despite those limitations, it poses some merits that could be exploited in determining coordinates. Water body is relatively more homogeneous because the usual obstacles present in water are smaller in size than that of in terrestrial environment. The region of interest on the ground is more likely occupied with buildings and trees which are the major factors for multipath propagation [1].

Regarding signal propagation in water, acoustic signal propagates much further compare to radio signal; however the speed of the acoustic signal is much slower than that of radio signal. TABLE 3.1 shows some limitations and typical measurements for radio and acoustic signals.

TABLE 3.1 Properties of Radio and Acoustic signal

	Radio signal		Acoustic signal	
	Vacuum	Water	Vacuum	Water
Velocity	3×108 m/s	≈2.25×106 m/s	-	VA
Range	-	1.8-323 m	-	1 - 100 km

The main environmental variable that we assume in our method to determine distances is the speed of acoustic signals in water. It depends on the temperature, salinity and permeability. How the speed of acoustic will vary because of aforesaid factors is not considered in this study, but our mathematical model assumes it as a variable v_A.

3.3 Distance Measurement

Despite the limitation of both radio and acoustic signal propagation in water in different aspect, we will exploit each of their merits in our proposed method to increase the accuracy of the distance measurements. Differential speed between radio and acoustic signals will be used to calculate the distance, while acoustic signal will be used for communication purposes. This method will require a short communication in between the beacon and sensor nodes.

Even though the speed of radio signal is slightly less than that of in the vacuum, considering the problem domain, the speed variation will not have significant impact on the proposed localization method. Moreover, the speed of acoustic signal, which varies due to different environmental factors, is the main variable that we need to use for coordinate determination [1].

Assumptions needed:

- The beacon can generate radio and acoustic signals simultaneously.
- The environmental factors that affect the acoustic signal will be considered while measuring inter-node distances.
- Sensor nodes are mobile.
- Sensors speed are known.
- Sensors follow the same direction towards X-axis.
- Base for all the sensors is same and the base is of tetrahedron shape.
- Each sensor node will have a unique ID.

The distance measurement calculation mentioned below :

(i) Simultaneous generation of radio and acoustic signals by the beacon , j = 4,5.... at t_0

(ii) For any submerged sensors Si , i = 1,2,3

a. Sensors receive the radio signal at

$t_{R(rec)} = t_0+\epsilon$

b. Sensor receives the acoustic signal at

$t_{A(rec)}$; here $t_{A(rec)} >> t_{R(rec)}$

(iii) Time taken for the acoustic signal to travel from beacon to sensor is :

$T_{ij(Travel),i=1,2,3;j=4,5,6.....} = t_{A(rec)} - t_{A(tra)}$

$= t_{A(rec)} - t_{R(tra)}$ Since, $t_{A(rec)} - t_{R(tra)}$

So, $T_{ij(Travel)} \approx t_{A(rec)} - t_{R(tra)}$ Since, $t_{R(rec)} = t_0 + \epsilon \approx t_{R(tra)}$

(iv) Sensor nodes send back the time $T_{ij(Travel)}$ with individual sensor's ID to the beacon using acoustic signal.

(v) Beacon nodes compute the distance between the beacon and sensors:

$d_{ij} = v_A \times T_{ij(travel)}$ here, is average acoustic signal speed.

Being aware of the limitations of radio and acoustic signals in water, each of its merit has been used in our proposed method to determine the distances in the problem domain. The method is relatively simple but precise enough when both the beacon and sensors are capable of transmitting/receiving radio and acoustic signals. To be precise, the beacon should be capable of acoustic (transmit (Tx) & receive (Rx)) and radio (transmit (Tx)) only. On the other hand, sensors should be capable of acoustic (transmit (Tx) & receive (Rx)) and radio (receive (Rx)) only. Considering most of the practical applications, this assumption is considered pragmatic [1].

For the acoustic velocity, we are using vA in our mathematical model, which depends on several factors and how these factors affect vA is left for future investigation. However, in our approach, we are able to measure the time in between transmission and reception using radio signals more accurately. Furthermore, our model has very low overhead. Figure. 3.2 shows the sequence of action that each node performs; where at the end only one message with the value of Tij (travel) and sensor ID is transferred via acoustic signal from sensor to beacon for distance calculation.

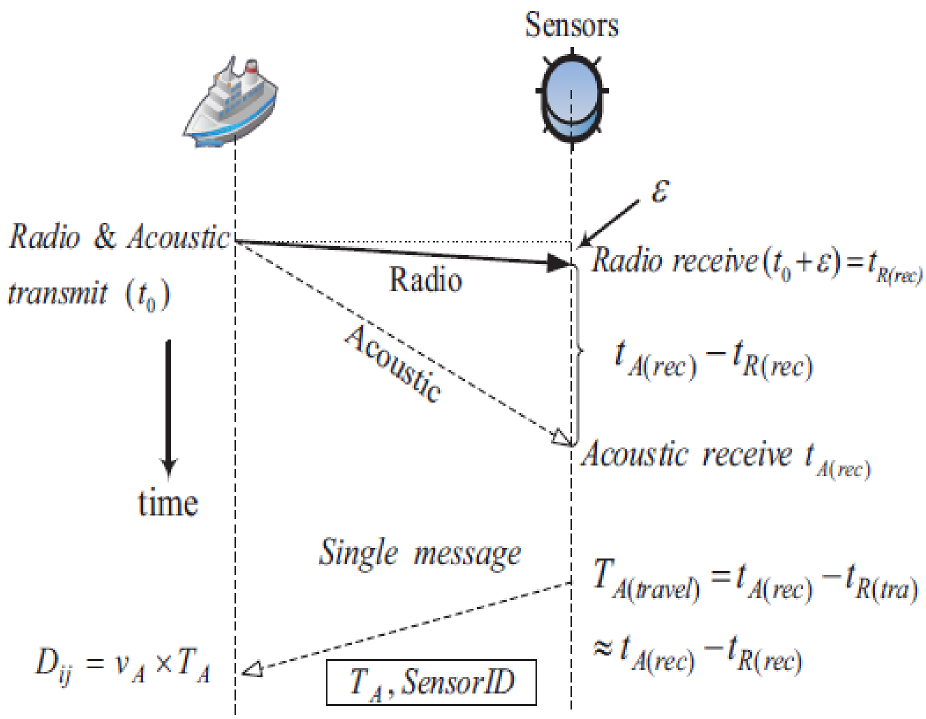

Figure. 3.2 Message transmission for distance calculation

3.4 Coordinates of the Sensors

The objective of localization algorithms is to obtain the exact position or coordinates of all the sensor nodes by measuring distances between beacon and them. Only measurements available here to compute is the distance and typically it is considered as optimization problem where objective functions to be minimized have residuals of the distance equations.

The variables of any localization problem are the 3D coordinates of the nodes. In principle more number of distance equations are needed than number of variables to solve this kind of problem. However, this approach known as degree of freedom analysis may not guarantee the unique solution in a nonlinear system [1].

Trilateration or multilateration techniques that are nonlinear system usually used to determine the location or coordinates of the sensors in partial or full. According to Guevara et al. [10] the convergence of optimization algorithms and Bayesian methods depend heavily on initial conditions used and they circumvent the convergence problem by linearizing the trilateration equations.

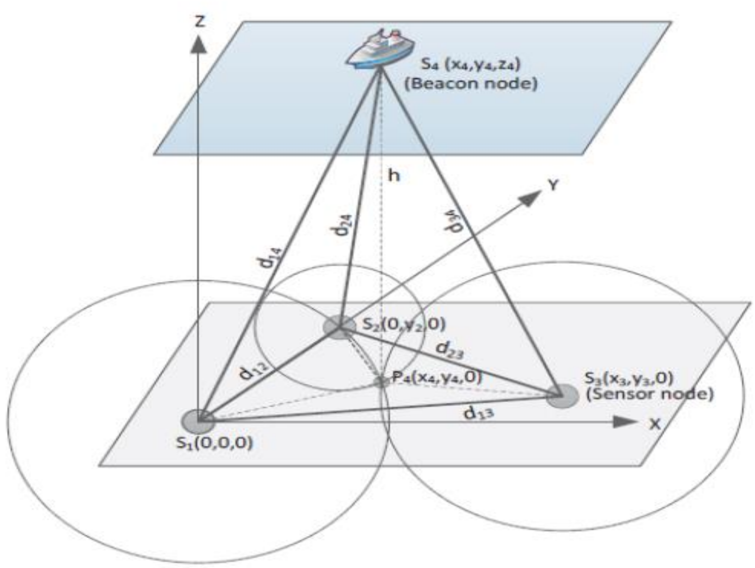

Figure 3.3: Coordinates determinations

Figure 3.3 shows the initial subset composed of the beacon node S_j, j = 4,5....9 and three sensor nodes S_i, i =1,2,3. Without loss of generality, a coordinate system can be defined using one of the sensor nodes S_i, i =1,2,3. as the origin (0,0,0) of the coordinate system. Now the trilateration equations can be written as a function of two groups of distance measurements. The distance between beacon and sensors d_{14}, d_{24}, d_{34}.... which are measured data, and inter sensor distances d_{12}, d_{13}, d_{23} and the volume of tetrahedron V_t (formed by the beacon and sensors), are unknown. By expanding and grouping known-unknown variables of (1), we obtain;

$$d_{34}^2(d_{12}^2 - d_{23}^2 - d_{13}^2) + d_{14}^2\left(\frac{d_{23}^4}{d_{12}^2} - d_{23}^2 - \frac{d_{13}^2 d_{23}^2}{d_{12}^2}\right) + d_{24}^2\left(\frac{d_{13}^4}{d_{12}^2} - \frac{d_{13}^2 d_{23}^2}{d_{12}^2} - d_{13}^2\right) - (d_{14}^2 d_{24}^2 + d_{14}^2 d_{34}^2 - d_{24}^2 d_{34}^2 - d_{14}^4)\frac{d_{23}^2}{d_{12}^2} - \left((d_{34}^2 d_{24}^2 - d_{14}^2 d_{34}^2 + d_{14}^2 d_{24}^2 - d_{24}^4)\frac{d_{13}^2}{d_{12}^2}\right) + \left(144\frac{v_t^2}{d_{12}^2} + d_{13}^2 d_{23}^2\right) = (d_{24}^2 d_{34}^2 - d_{34}^4 + d_{14}^2 d_{34}^2 - d_{14}^2 d_{24}^2)$$

Here, $(d_{12}^2 - d_{23}^2 - d_{13}^2)$, $\left(\frac{d_{23}^4}{d_{12}^2} - d_{23}^2 - \frac{d_{13}^2 d_{23}^2}{d_{12}^2}\right)$, $\left(\frac{d_{13}^4}{d_{12}^2} - \frac{d_{13}^2 d_{23}^2}{d_{12}^2} - d_{13}^2\right)\frac{d_{23}^2}{d_{12}^2}, \frac{d_{13}^2}{d_{12}^2}$ and $\left(144\frac{v_t^2}{d_{12}^2} + d_{13}^2 d_{23}^2\right)$ are unknown terms.

The above expression can be written as follows,

$$d_{14}^2 X_1 + d_{24}^2 X_2 + d_{34}^2 X_3 - (d_{14}^2 - d_{34}^2)(d_{24}^2 - d_{14}^2)(d_{24}^2 - d_{14}^2)(d_{34}^2 - d_{24}^2)X_5 + X_6 = (d_{24}^2 - d_{34}^2)(d_{34}^2 - d_{14}^2) \quad \text{..................(1)}$$

Based on the local positioning system configuration of Figure. 3.4, we need to write equations that will include all known and unknown distances. For that matter, we express the volume of tetrahedron V_t using Cayley-Menger determinant as following:

$$288V_t^2 = \begin{vmatrix} 0 & 1 & 1 & 1 & 1 \\ 1 & 0 & d_{12}^2 & d_{13}^2 & d_{14}^2 \\ 1 & d_{12}^2 & 0 & d_{23}^2 & d_{24}^2 \\ 1 & d_{13}^2 & d_{23}^2 & 0 & d_{34}^2 \\ 1 & d_{14}^2 & d_{24}^2 & d_{34}^2 & 0 \end{vmatrix} \text{ ....(2)}$$

The Equation (2) in fact resembles the linear form of $a_1x_1 + a_2x_2 + \cdots + a_nx_n = b_1$.As we have six unknown in (1), we need at least six measurements, which could be done following the same procedure described earlier steering the beacon node S j , j=4,5....9 to six different locations and measuring the distances in the vicinity of S4. Finally, we get m-linear equations of the form;

$$\begin{aligned} a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n &= b_1, \\ a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n &= b_2, \\ &\vdots \\ a_{m1}x_1 + a_{m2}x_2 + \cdots + a_{mn}x_n &= b_m, \text{ ....(3)} \end{aligned}$$

If we omit reference to the variables, then system of equations in (3) can be represented by the array of all coefficients known as the augmented matrix of the system, where the first row of the array represents the first linear equation and so on. That could then be expressed in AX =b linear form. Then, the system of equations can be written as:

$$A = \begin{bmatrix} d_{14}^2 & d_{24}^2 & d_{34}^2 & -(d_{14}^2 - d_{34}^2)(d_{24}^2 - d_{14}^2) & -(d_{24}^2 - d_{14}^2)(d_{34}^2 - d_{24}^2) & 1 \\ d_{15}^2 & d_{25}^2 & d_{35}^2 & -(d_{15}^2 - d_{35}^2)(d_{25}^2 - d_{15}^2) & -(d_{25}^2 - d_{15}^2)(d_{35}^2 - d_{25}^2) & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ d_{19}^2 & d_{29}^2 & d_{39}^2 & -(d_{19}^2 - d_{39}^2)(d_{29}^2 - d_{19}^2) & -(d_{29}^2 - d_{19}^2)(d_{39}^2 - d_{29}^2) & 1 \end{bmatrix}$$

$$X = \begin{bmatrix} \left(\frac{d_{23}^4}{d_{12}^2} - d_{23}^2 - \frac{d_{13}^2 d_{23}^2}{d_{12}^2}\right) \\ \left(\frac{d_{13}^4}{d_{12}^2} - \frac{d_{13}^2 d_{23}^2}{d_{12}^2} - d_{13}^2\right) \\ (d_{12}^2 - d_{23}^2 - d_{13}^2) \\ \frac{d_{23}^4}{d_{12}^2} \\ \frac{d_{13}^4}{d_{12}^2} \\ \left(144\frac{V_t^2}{d_{12}^2} + d_{13}^2 d_{23}^2\right) \end{bmatrix} \qquad b = \begin{bmatrix} (d_{24}^2 - d_{34}^2)(d_{34}^2 - d_{14}^2) \\ (d_{25}^2 - d_{35}^2)(d_{35}^2 - d_{15}^2) \\ \vdots \\ (d_{29}^2 - d_{39}^2)(d_{39}^2 - d_{19}^2) \end{bmatrix}$$

From the above representation, after finding X_1, X_2 , X_3 , X_4, X_5 and X_6 we calculate d_{12},d_{13} and d_{23} as follows:

$$d_{12}^2 = \frac{X_3}{(1-X_4-X_5)}, d_{13}^2 = \frac{X_3 X_5}{(1-X_4-X_5)}, d_{23}^2 = \frac{X_3 X_4}{(1-X_4-X_5)}$$

If we let the coordinates of the submerged sensors S_1,S_2 and S_3 are (0,0,0), (0,y_2 ,0) and (x_3, y_3 , 0) respectively, then the inter-sensor distances could be written with respect to coordinates of the sensors as follows:

$$d_{12}^2 = y_2^2, d_{13}^2 = x_3^2 + y_3^2, d_{23}^2 = x_3^2 + (y_3 - y_2)^2$$

From the above values the unknown variables can be computed as follows:

$$y_2 = d_{12}, \;\; y_3 = \frac{d_{12}^2 + d_{13}^2 - d_{23}^2}{2d_{12}}, x_3 = \sqrt{\left(d_{13}^2 - \left(\frac{d_{12}^2 + d_{13}^2 - d_{23}^2}{2d_{12}}\right)^2\right)}$$

where d_{12}, d_{13} and d_{23} are known computed distances Table 3.2 summarizes the coordinates of the sensors for this system.

Table 3.2 Coordinates of the sensors with known measurements

Sensors	Coordinates
S_1	(0,0,0)
S_2	$(0,d_{12},0)$
S_3	$(\sqrt{\left(d_{13}^2-\left(\frac{d_{12}^2+d_{13}^2-d_{23}^2}{2d_{12}}\right)^2\right)},\frac{d_{12}^2+d_{13}^2-d_{23}^2}{2d_{12}},0)$

3.5 Coordinates of the Sensors with respect to the Beacon initially:

Up to now we have been able to determine the coordinates of the sensor nodes with respect to S_1 . In order to find the coordinate with respect to the beacon node we follow the following steps.

We assume that with the use of appropriate sensors , the depth *h* can be measured [12].After measuring the vertical distance *h* in between the beacon node S_4 (x_4 , y_4 , z_4) and the *XY* plane, we can assume the projected coordinate of the beacon node S_4 (x_4 , y_4 , z_4) on the plane *XY* is P_4 (x_4, y_4,0) . To find x_4 and y_4 , we can apply trilateration in the following manner assuming the distances between S_1, S_2, S_3 and P_4 are D_{14} , D_{24} and D_{34} respectively.

$$D_{14}^2 = x_4^2 + y_4^2 \qquad \text{..........(4)}$$

$$D_{24}^2 = x_4^2 + (y_4 - y_2)^2 \qquad \text{..........(5)}$$

$$D_{14}^2 = (x_4 - x_3)^2 + (y_4 - y_3)^2 \qquad \text{.........(6)}$$

From equation (4), (5) and (6) we obtain the projected beacon's coordinates P_4 (x_4, y_4,0).

$x_4 = \sqrt{\frac{1}{2D}(2d_{12}D_{14}^2 - D_{14}^2 + D_{24}^2 + d_{12}^2)}$,

$y_4 = \frac{1}{2d_{12}}(D_{14}^2 - D_{24}^2 + d_{12}^2)$

As d_{14} , d_{24} and d_{34} are hypotenuse of the $\Delta S_1P_4S_4$, $\Delta S_2P_4S_4$ and $\Delta S_3P_4S_4$ respectively , so it is possible to obtain D_{14} , D_{24} and D_{34} using Pythagorean Theorem. So the coordinate of the beacon node S_4 (x_4 , y_4 , z_4) would be S_4 (x_4 , y_4 , h) where all the elements are known.

$$\text{So, } S_4\,(x_4 , y_4 , 0) = S_4\left(\begin{matrix}\left(\sqrt{\frac{1}{2D}(2d_{12}D_{14}^2 - D_{14}^2 + D_{24}^2 + d_{12}^2)}\right), \\ \left(\frac{1}{2d_{12}}(D_{14}^2 - D_{24}^2 + d_{12}^2)\right), h\end{matrix}\right)$$

If the origin of the Cartesian system is transferred on to the coordinate of the beacon node, then it is possible to find the coordinates of other sensors with respect to the beacon node S_4. A linear transformation would give the results as in Table 3.3.

Table 3.3 Coordinates of the sensors with respect to beacon for parallel situation

Sensors	**Coordinates**	**Sensors**	**Coordinates**
S_4	(0, 0, 0)	S_2	(x_2-x_4 , $y_2 - y_4$, -z_4)
S_1	(-x_4 ,- y_4 , -z_4)	S_3	(x_3-x_4 , $y_3 - y_4$, -z_4)

3.6 Coordinates of the Sensors with respect to the Beacon when they are mobile:

Upon finding the coordinates of the sensors initially i.e when they were static, in this chapter using those measurements we find out their coordinates with their velocity in their mobile form. In order to do that , when the sensors and the beacon move from their previous position to their next , letting them do that, we get the sensor to its previous position for the measurement purpose. So, after that we use the triangulation technique and measure the distance between the beacon's current position to sensor's previous position.

Finding the distance we again use the Cayley-Menger determinant to find out its position then we add the distance that it covered with its speed to find out its actual positions coordinates. That's how we localize the mobile sensors.

Now finding out the sensors positions based on the axises .

When the sensor S_1 is on X-axis:

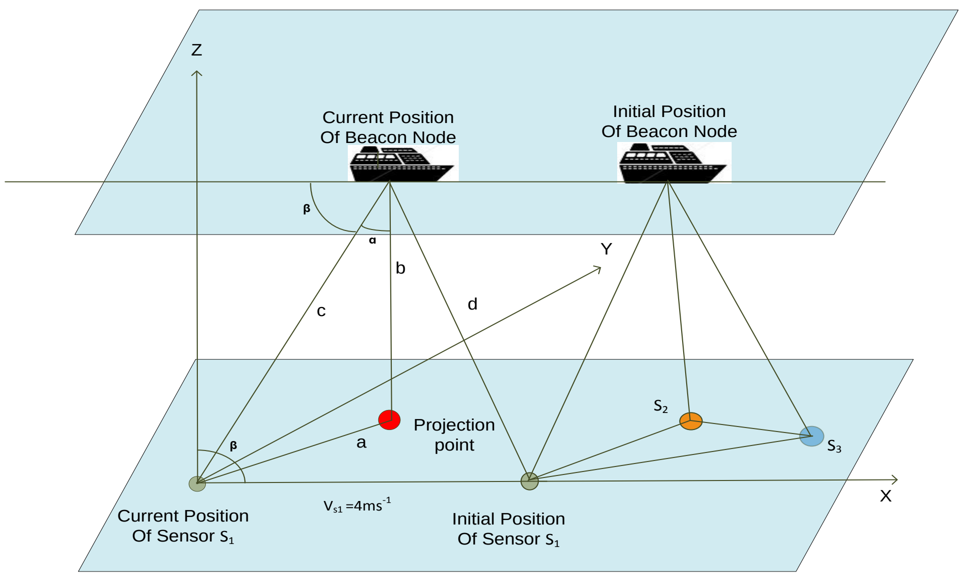

Figure 3.4: sensor S_1 on X-axis

From the above figure ,we first find out the angle **α**,

$$a^2 = b^2+c^2-2.b.c.\cos\alpha$$

then we find out the angle **β**,

$$\beta = 90^0 - \alpha$$

then finally we find out the distance between the beacon's current position to sensor's previous position by using the equation which is given below:

Let, e= V_{s1} ms^{-1}

$$\mathbf{d^2 = c^2+e^2-2.c.e.cos\beta}$$

$$\text{so, } \mathbf{d} = \sqrt{c^2 + e^2 - 2.c.e.\cos\beta}$$

When the sensor S_2 is on Y-axis:

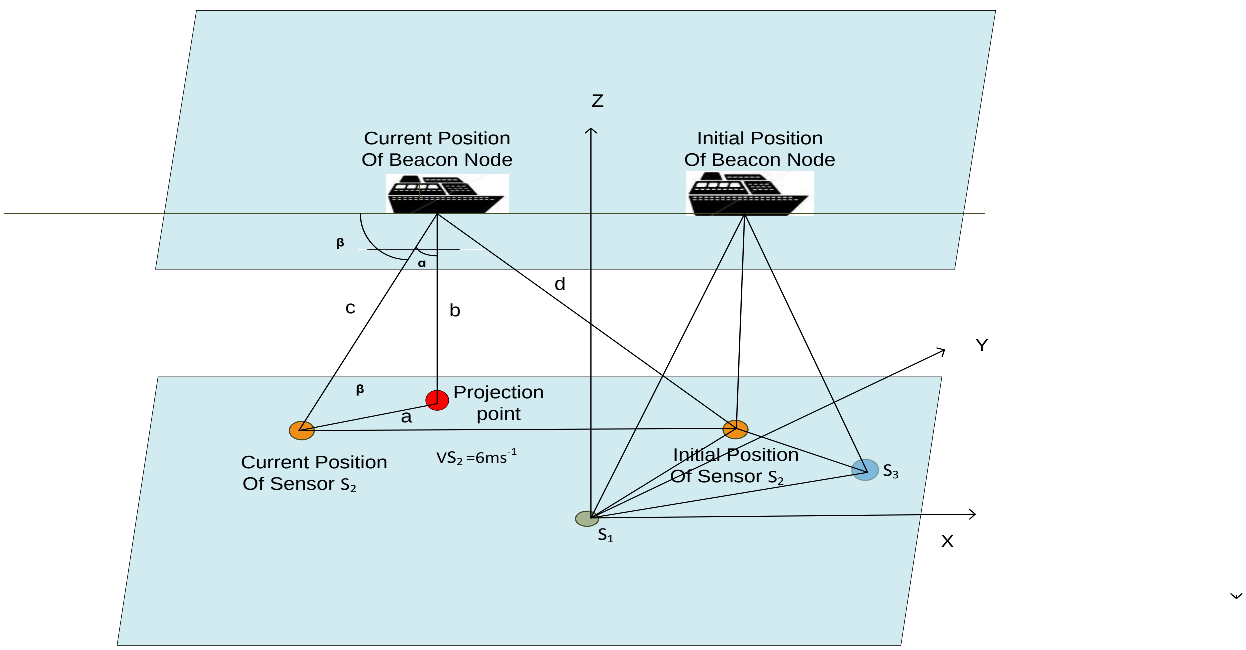

Figure 3.5: sensor S_2 on Y-axis

From the above figure ,we first find out the angle **α**,

$$\mathbf{a^2 = b^2+c^2-2.b.c.cos\alpha}$$

then we find out the angle **β**,

$$\mathbf{\beta = 90^0 - \alpha}$$

then finally we find out the distance between the beacon's current position to sensor's previous position by using the equation which is given below:

Let, e= V_{s2} ms^{-1}

$$\mathbf{d^2 = c^2+e^2-2ce\cos\beta}$$

$$\text{so, } d = \sqrt{c^2 + e^2 - 2.c.e.\cos\beta}$$

<u>When the sensor S_3 is out of axis:</u>

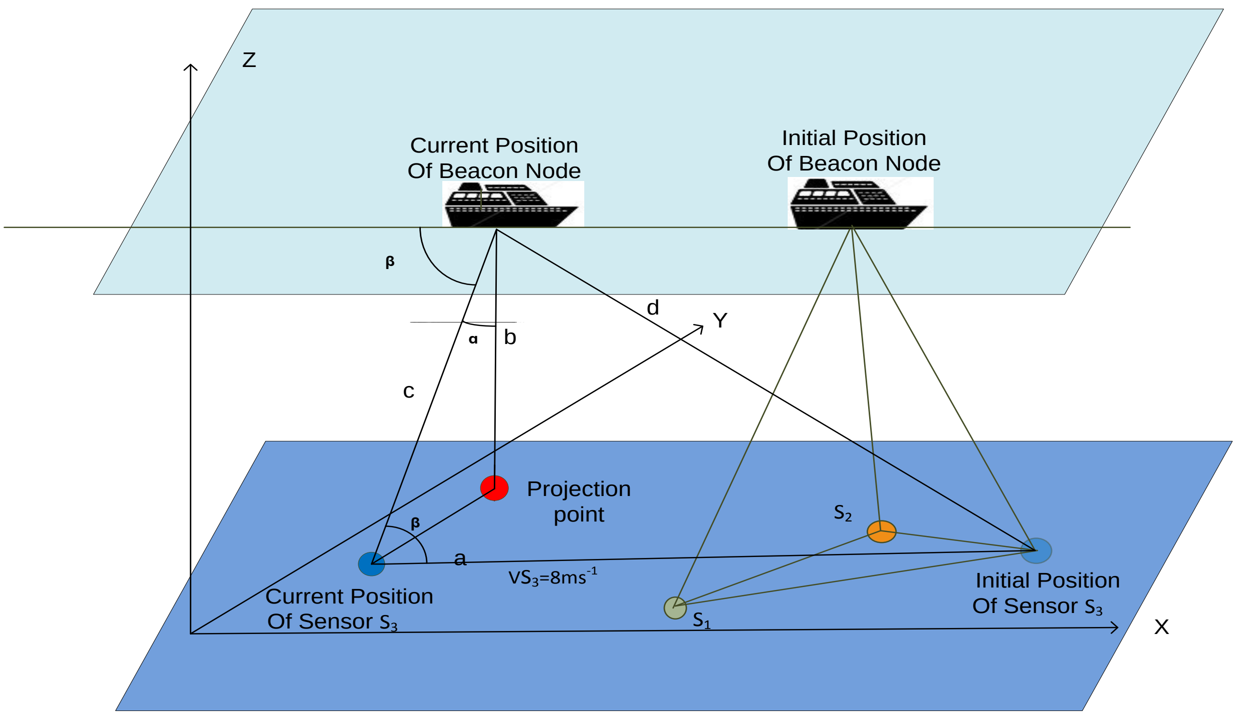

Figure 3.6 : sensor S_3 out of axis

From the above figure ,we first find out the angle **α**,

$$\mathbf{a^2 = b^2+c^2-2.b.c.\cos\alpha}$$

then we find out the angle **β**,

$$\mathbf{\beta = 90^0 - \alpha}$$

then finally we find out the distance between the beacon's current position to sensor's previous position by using the equation which is given below:

Let, $e= V_{s3}\ ms^{-1}$

$$\mathbf{d^2 = c^2+e^2-2ce\cos\beta}$$

so, $\mathbf{d = \sqrt{c^2 + e^2 - 2.c.e.\cos\beta}}$

Chapter 4

Computational Results & Analysis

A simulation of the proposed method to determine the coordinates of submerged sensors which are mobile as described in chapter 3 was performed to verify the method. The experiment was designed based on 3-D space. As the distance measurement between beacon and sensors is possible. So at first we assume the three sensors are placed at (0, 0, 0), (0, 70, 0) and (85, 90, 0) and a floating beacon moved towards one direction , assumed to be X-axis which is in a plane and that plane is parallel to the imaginary XY plane where the sensors are in 3D- space. The coordinates of the sensors are randomly chosen. Z-coordinates of sensors is always kept zero to satisfy that all sensors are situated in same plane and for computational simplicity one of the coordinates are placed at the origin. Then, all the sensors with their respective velocity start moving towards the same direction as of the beacon and we assumed it to be the X-axis.

So, during that we calculate the final coordinate of the sensors from the beacon according to our proposed method. This proposed method has been simulated using Matlab (v2010). To simulate the proposed method 18 datasets were taken. While calculating the true Euclidian distance from six different beacon nodes to sensors S_1, S_2 and S_3. The simulation result of our proposed method is given below:

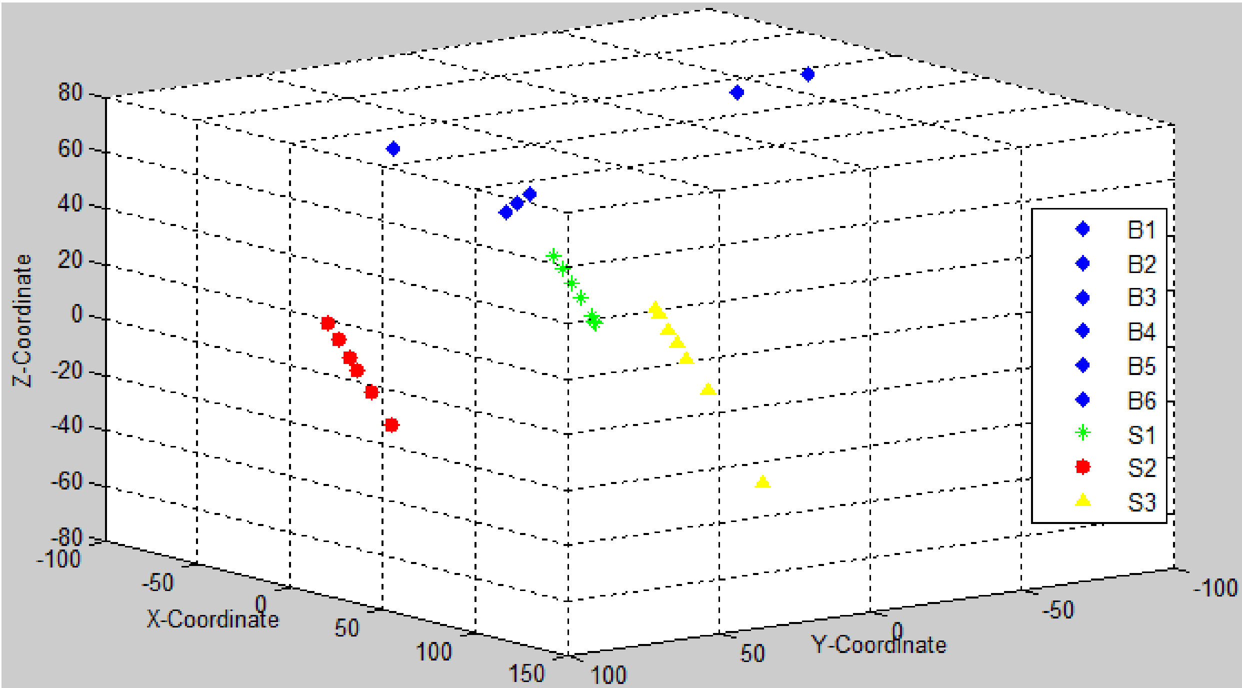

Figure 4.1: Calculated sensors positions with proposed method

To calculate the coordinates of sensors we need the inner distances between sensors S_1, S_2 and S_3. After solving the linear equation (3) which is formed by Cayley-Menger determinant equation (2) we find the inner distances between sensors S_1, S_2 and S_3. These distances may contain some Gaussian error because of it calculated from the Gaussian error distances between beacon and sensors.

After that when the beacon and the sensors start moving, we calculate the distance between beacon's current position to the sensors previous position using triangulation and trilateration technique. Calculating the distance we use it to find the latest coordinate of the sensor. In order to do that we again used Cayley-Menger determinant. Then we add up the distance with the coordinates, that the sensors had crossed and we can get it from its velocity. That's how we will get the final locality of the sensors.

It is necessary to mention that both the sensors and the beacon were moving towards X-axis.

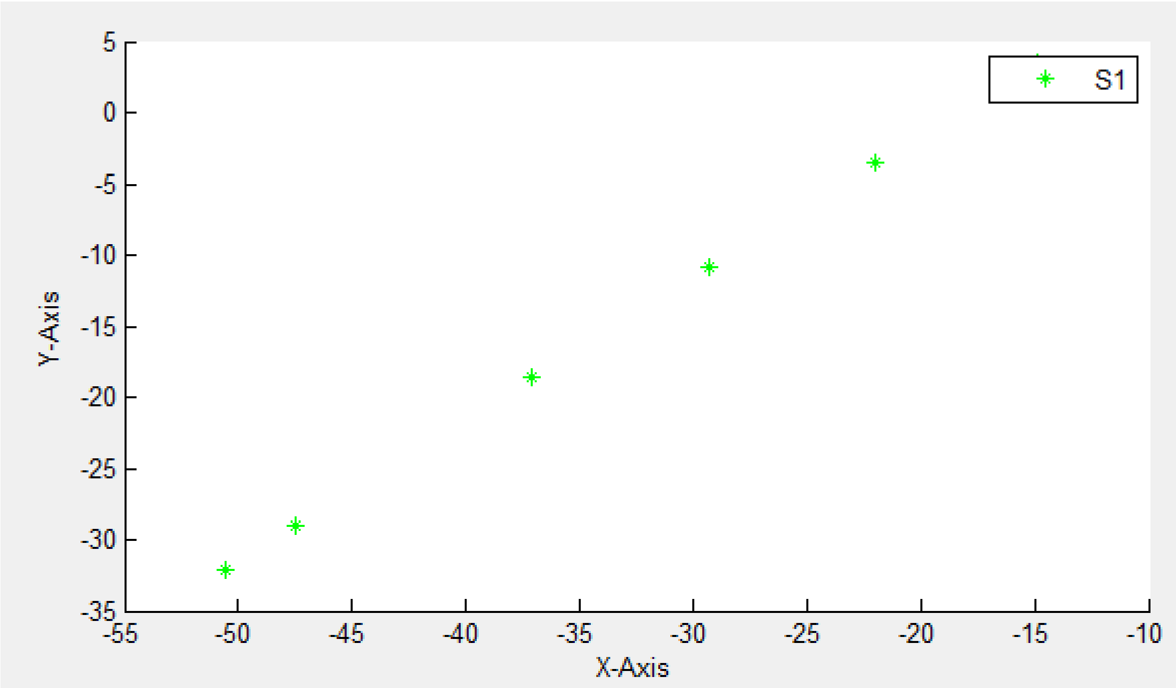

Figure4.2: Distance of sensor S1 from the original position

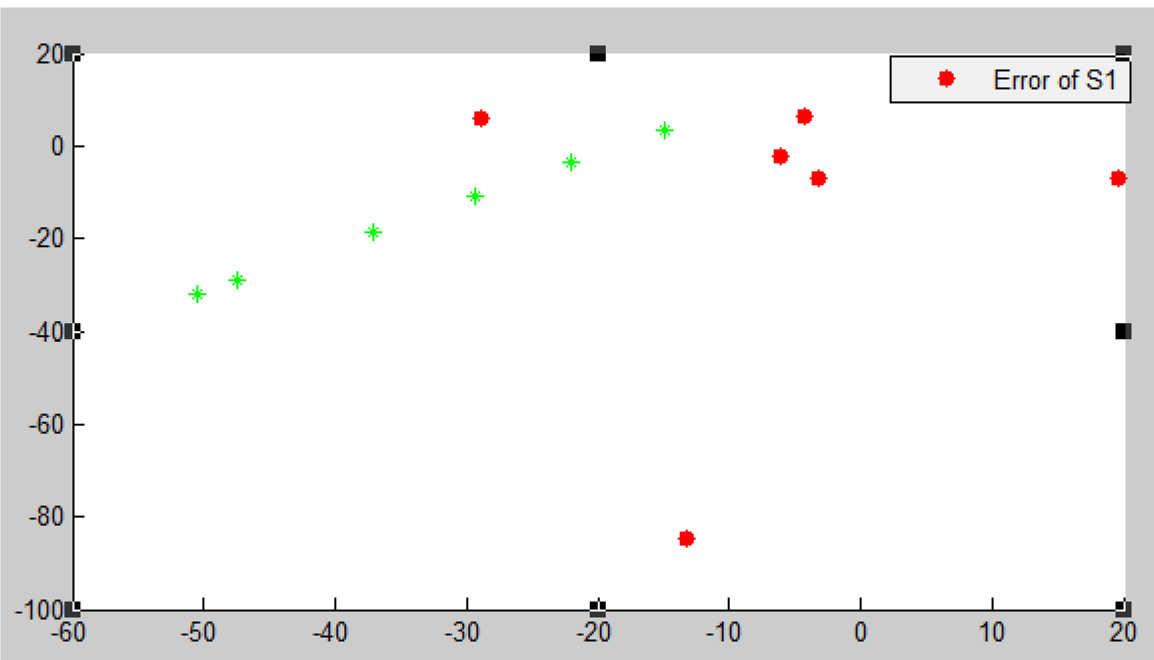

Figure4.3: Distance errors of sensor S1 from the original position

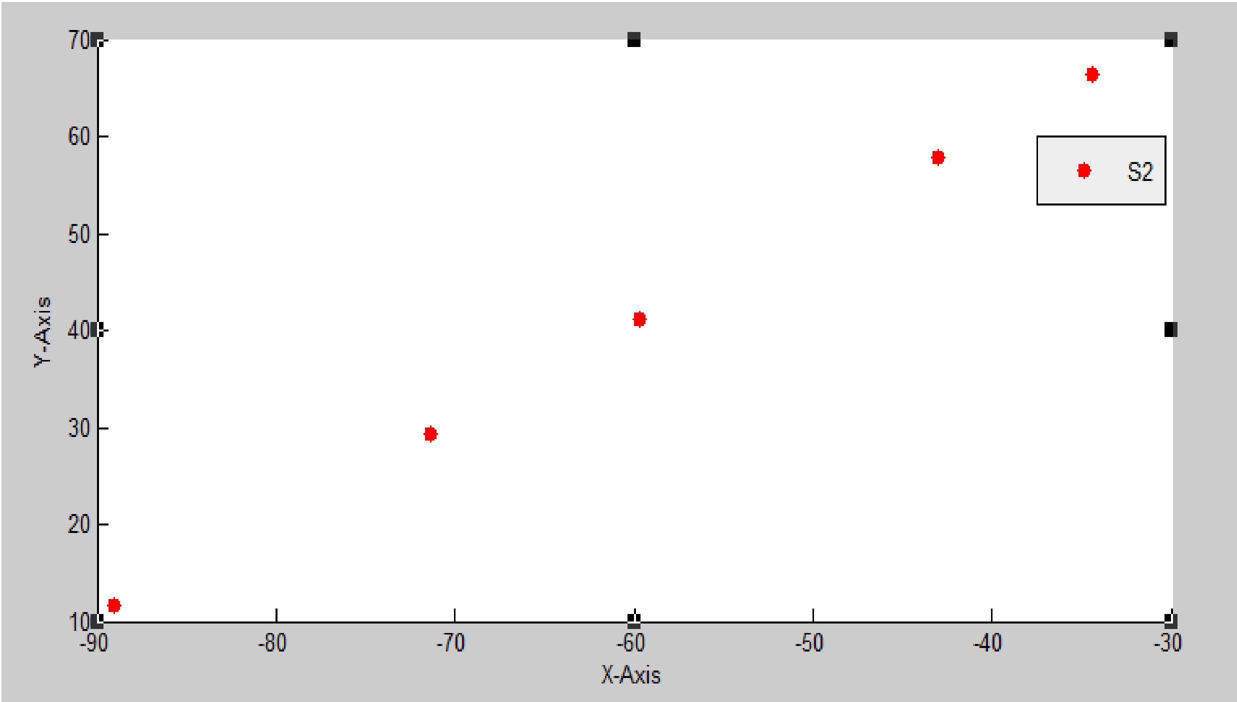

Figure4.4: Distance of sensor S2 from the original position

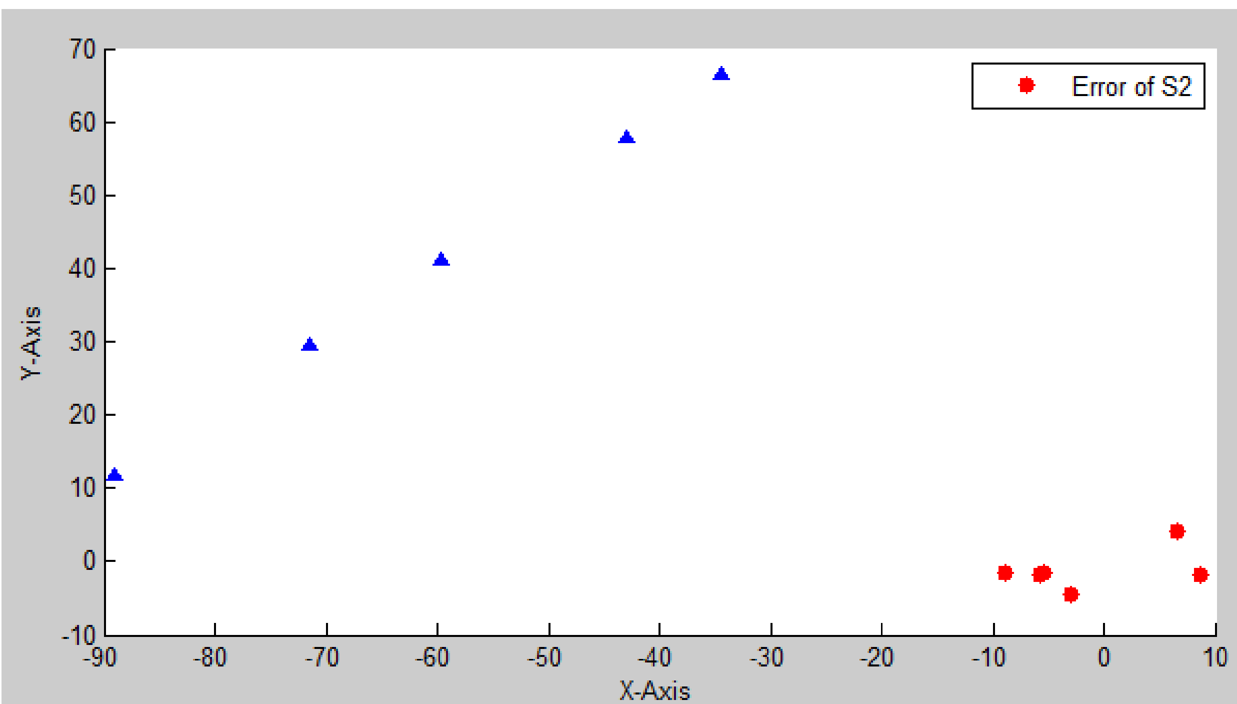

Figure4.5: Distance errors of sensor S2 from the original position

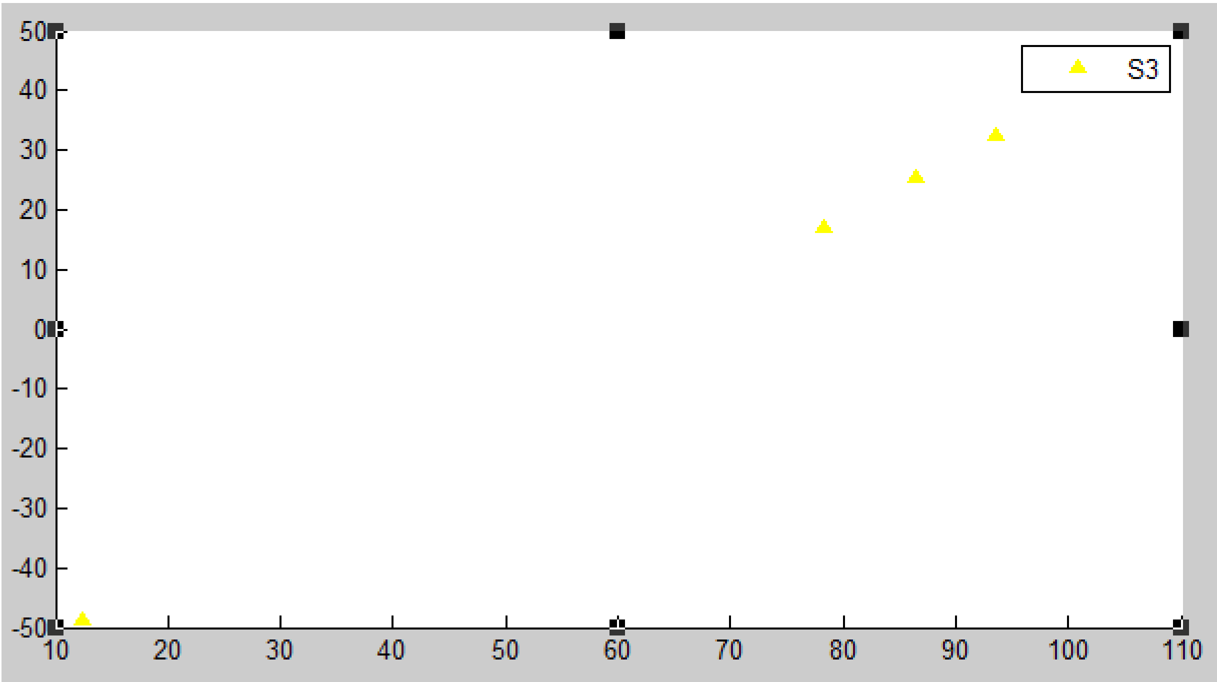

Figure 4.6: Distance of sensor S3 from the original position

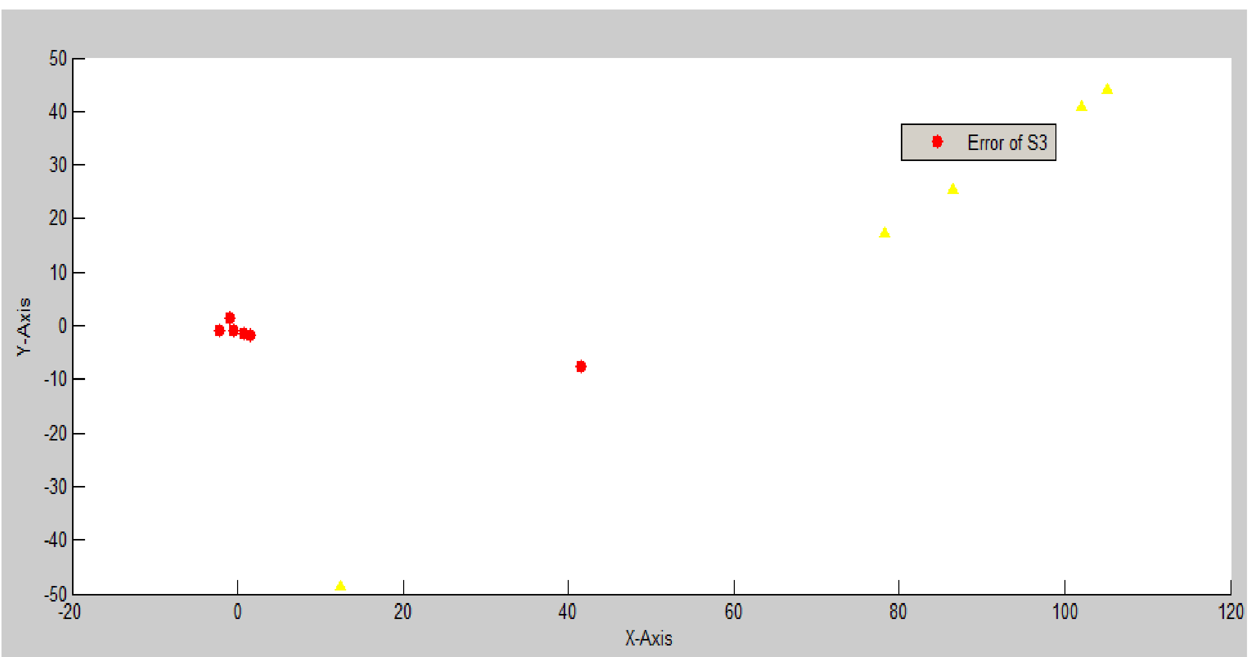

Figure4.7: Distance errors of sensor S3 from the original position

In the above figures, we showed the plots from simulation for all the three sensors in their mobile form. From simulation we tried to show the sensors' (S1, S2, & S3) mobility pattern.
In Figure 4.1 we showed the sensors' mobility pattern along with the beacon's, all together in 3D form.

Here, the initial position of the beacon were taken randomly . So the position of the beacon can be changed and it will not cause any discrepancy to our proposed method and thus can still be applied. The calculation and the technique will be performed accordingly.

Table 4.1 compares the positional errors of sensor S1, S2 and S3 when they are mobile & the distances between the beacon's current position and sensors' previous positions are calculated.

Table 4.1 : Matlab Simulation and Percentage of Error Result

	Experimental coordinates of sensors			**Actual coordinates of sensors**			**Percentage of error in distance**		
	Ex	Ey	Ez	Ax	Ay	Az	ErX	ErY	ErZ
S1	15.589	7.589	-10.410	13.034	8.154	-9.324	19.603	-6.929	11.647
	22.882	14.882	-3.117	23.912	14.001	-4.523	-4.307	6.292	-31.086
	30.002	22.002	4.001	31.000	23.654	3.999	-3.2	-6.984	0.050
	-5.661	-13.661	-31.661	-6.023	-14.001	-29.672	-6.010	-2.428	6.703
	7.801	-0.199	-18.199	8.995	-1.312	-20.101	-13.274	-84.832	-9.462
	-2.598	-10.598	-28.598	-3.655	-9.991	-30.001	-28.919	6.075	-4.677
S2	-26.535	37.464	-50.535	-24.886	35.990	-51.535	6.626	4.096	-1.940
	-14.835	49.164	-38.835	-13.654	50.010	-37.110	8.649	-1.692	4.648
	1.883	65.883	-22.116	1.999	66.997	-20.776	-5.803	-1.663	5.883
	10.442	74.442	-13.557	11.045	77.882	-13.661	-5.459	-1.612	-3.123
	10.441	74.441	-13.558	11.453	75.661	-13.995	-8.836	-1.614	-3.123
	-44.181	19.819	-68.180	-45.549	20.734	-70.000	-3.003	-4.413	-2.6
	76.593	87.593	-0.406	77.012	88.425	-0.761	-0.544	-0.941	-46.649

S3	61.256	72.256	-15.743	60.351	73.572	-14.618	1.499	-1.789	7.696
	85.047	96.047	8.047	84.454	97.482	9.065	0.702	-1.472	-11.230
	69.459	80.459	-7.541	70.158	79.294	-8.992	-0.996	1.469	-16.137
	88.241	99.241	11.241	90.162	100.021	11.201	-2.131	-0.779	0.357
	-4.506	6.493	-81.506	-3.183	7.026	-80.472	41.565	-7.586	1.285

Percentage of error in distance:

((Experimental value – Actual value)/ Actual value)*100

Chapter 5

Conclusion and Future Work

Focusing on the problems of underwater WSNs, like sparse deployment and mobility of the nodes, in this paper we presented a mathematical model to localize submerged mobile sensors using single beacon. The method computes the coordinates with respect to the beacon and sensor node that alleviates a number of problems in the domain of localization since they are moving in one direction. Besides, our distance measurement model and technique contributes minimum error and potentially avoid multipath fading effects in localization.

Finally, simulation results validate that proposed mathematical model generates negligible error in coordinate determination of the sensors when distances between beacon and sensors are true Euclidean distance. The simulation results show that this method can provide highly accurate localization of beacon nodes, and the positioning accuracy and positioning coverage rate can be kept at a better level.

While UWSN is a promising new field and may help in exploring the unfathomed world that lies underwater, there are many challenges and opportunities as well. Facing the challenges of underwater WSNs, a mathematical model to determine the coordinates of submerged sensors which are all mobile using single beacon is presented. We use multilateration, trilateration and triangulation technique to determine the location of the sensors with respect to the beacon node where distance between them is measured considering the acoustic and radio signal. Cayley–Menger determinant is used to determine the nodes coordinates , it reduces the impact of distance measurement error on the location estimation.Simulation results validate that proposed mathematical model though it generates negligible error in coordinate determination of the sensors when distances between beacon and sensors are true Euclidean distance. It also shows that coordinates are within acceptable error range.

In future work we plan to consider sensors involuntary mobility, moving towards random directions due to water currents in the proposed model. The amount of challenges in designing of UWSNs makes it an interesting area for researchers to work on. With the advancement in sensor and wireless technologies, UWSNs have attracted a lot of researchers and have contributed significantly to this field. However, the window is still wide open for upcoming research and opportunities.

References

[1] A. Rahman, V. Muthukkumarasamy and E. Sithirasenan, "Coordinates Determination of Submerged Sensors Using Cayley-Menger Determinant," 2013 IEEE International Conference on Distributed Computing in Sensor Systems, Cambridge, MA, 2013, pp. 466-471.

[2] H. P. Tan, R. Diamant, W. K. G. Seah, and M. Waldmeyer, "A survey of techniques and challenges in underwater localization," Ocean Engineering, vol. 38, pp. 1663-1676, 2011.

[3] J. H. Cui, J. Kong, M. Gerla, and S. Zhou, "The challenges of building mobile underwater wireless networks for aquatic applications," Network, IEEE, vol. 20, pp. 12-18, 2006.

[4] P. Xie, J. H. Cui, and L. Lao, "VBF: vector-based forwarding protocol for underwater sensor networks," Networking 2006. Networking Technologies, Services, and Protocols; Performance of Computer and Communication Networks; Mobile and Wireless Communications Systems, pp. 1216-1221, 2006.

[5] Z. Zhou, J. H. Cui, and S. Zhou, "Efficient localization for large-scale underwater sensor networks," Ad Hoc Networks, vol. 8, pp. 267-279, 2010.

[6] T. C. Austin, R. P. Stokey, and K. M. Sharp, "PARADIGM: a buoybased system for AUV navigation and tracking," in OCEANS 2000 MTS/IEEE Conference and Exhibition, 2000, pp. 935-938 vol.2.

[7] J. C. Prieto, A. R. Jiménez, J. Guevara, J. L. Ealo, F. Seco, J. O. Roa, and F. Ramos, "Performance evaluation of 3D-LOCUS advanced acoustic LPS," Instrumentation and Measurement, IEEE Transactions on, vol. 58, pp. 2385-2395, 2009.

[8]. AlHajri, M.I.; Goian, A.; Darweesh, M.; AlMemari, R.; Shubair, R.M.; Weruaga, L.; Kulaib, A.R. Hybrid RSS-DOA technique for enhanced WSN localization in a correlated environment. In Proceedings of the 2015 IEEE International Conference on Information and Communication Technology Research (ICTRC), Abu Dhabi, United Arab Emirates, 17–19 May 2015; pp. 238–241.

[9]. El Assaf, A.; Zaidi, S.A.R.; Affes, S.; Kandil, N. Range-free localization algorithm for heterogeneous wireless sensor networks. In Proceedings of the 2014 IEEE Wireless Communications and Networking Conference (WCNC), Istanbul, Turkey, 69 April 2014; pp. 2805–2810.

[10]. Hu, L.; Evans, D. Localization for mobile sensor network. In Proceedings of the ACM 10th annual international conference on Mobile computing and networking, Philadelphia, PA, USA, 26 September–1 October 2004; pp. 45–57.

[11]. Soltaninasab, B.; Sabaei, M.; Amiri, J. Improving Monte Carlo localization algorithm using time series forecasting method and dynamic sampling in mobile WSNs. In Proceedings of the 2010 Second International Conference on Communication Systems, Networks and Applications, Hong Kong, China, 29 June–1 July 2010; pp. 389–396.

[12]. Alcocer, A.; Oliveira, P.; Pascoal, A. Study and implementation of an EKF GIB-based underwater positioning system. Control Eng. Pract. **2007**, 15, 689–701.

[13]. Cheng, X.; Shu, H.; Liang, Q.; Du, D.H.C. Silent positioning in underwater acoustic sensor networks. IEEE Trans. Veh. Technol. **2008**, 57, 1756–1766.

[14]. Zhou, Z.; Peng, Z.; Cui, J.H.; Shi, Z.; Bagtzoglou, A.C. Scalable localization with mobility prediction for underwater sensor networks. IEEE Trans. Mob. Comput. **2011**, 10, 335–348.

[15]. Cheng, W.; Teymorian, A.Y.; Ma, L.; Cheng, X.; Lu, X.; Lu, Z. Underwater localization in sparse 3D acoustic sensor networks. In Proceedings of the 27th IEEE Conference on Computer Communications INFOCOM, Phoenix, AZ, USA, 13–18 April 2008; pp. 236–240.

[16]. Yang, H.; Sikdar, B. A Mobility Based Architecture for Underwater Acoustic Sensor Networks. In Proceedings.

[17] J. Guevara, A. R. Jimenez, A. S. Morse, J. Fang, J. C. Prieto, and F. Seco, "Auto-localization in Local Positioning Systems: A closedform range-only solution," in Industrial Electronics (ISIE), 2010 IEEE International Symposium on, 2010, pp. 2834-2840.

[18] J. Guevara, A. Jiménez, J. Prieto, and F. Seco, "Auto-localization algorithm for local positioning systems," Ad Hoc Networks, 2012.

[19] X. Che, I. Wells, G. Dickers, P. Kear, and X. Gong, "Re-evaluation of RF electromagnetic communication in underwater sensor networks," Communications Magazine, IEEE, vol. 48, pp. 143-151, 2010.

[20] I. Vasilescu, K. Kotay, D. Rus, M. Dunbabin, and P. Corke, "Data collection, storage, and retrieval with an underwater sensor network," in Proceedings of the 3rd international conference on Embedded networked sensor systems, 2005, pp. 154-165.range-only solution," in Industrial Electronics (ISIE), 2010 IEEE International Symposium on, 2010, pp. 2834-2840.

[21] A. R. Kulaib and R. M. Shubair, M. A. Al-Qutayri and Jason W. P. Ng, “An Overview of Localization Techniques for Wireless Sensor Networks”.

[22] P. Duff and H. Muller, "Autocalibration algorithm for ultrasonic location systems," in Wearable Computers, 2003. Proceedings. Seventh IEEE International Symposium on, 2003, pp. 62-68.

[23] E. Olson, J. Leonard, and S. Teller, "Robust range-only beacon localization," in Autonomous Underwater Vehicles, 2004 IEEE/OES, 2004, pp. 66-75.

Appendix

Code:

```
% The depth of senors S1 ,S2 & S3 are same H1=H2=H3=H=60 ;

H=70;

% Beacon Nodes

B1 = [ 100 90 H];

B2 = [90 80 H];

B3 = [ 80 70 H];

B4 = [ -10 60 H];

B5 = [ -20 -60 H];

B6 = [ -30 -90 H];

 % Sensor Coordinates

S1 = [0 0 0 ] ;

S2 = [0 70 0];

S3 = [85 90 0];

% from  Beacon's 1st position

d14 = pdist2(B1,S1,'euclidean');

d24 = pdist2(B1,S2,'euclidean');

d34 = pdist2(B1,S3,'euclidean');
```

Dist_Between_Senosrs_and_B1 = [d14 d24 d34];

% from Beacon’s 2nd position

d15 = pdist2 (B2,S1,'euclidean');

d25 = pdist2 (B2,S2,'euclidean');

d35 = pdist2 (B2,S3,'euclidean');

Dist_Between_Senosrs_and_B2 = [d15 d25 d35];

% from Beacon’s 3rd position

d16 = pdist2 (B3,S1,'euclidean');

d26 = pdist2 (B3,S2,'euclidean');

d36 = pdist2 (B3,S3,'euclidean');

Dist_Between_Senosrs_and_B3 = [d16 d26 d36];

% from Beacon’s 4th position

d17 = pdist2 (B4,S1,'euclidean');

d27 = pdist2 (B4,S2,'euclidean');

d37 = pdist2 (B4,S3,'euclidean');

Dist_Between_Senosrs_and_B4 = [d17 d27 d37];

% from Beacon’s 5th position

d18 = pdist2 (B5,S1,'euclidean');

d28 = pdist2 (B5,S2,'euclidean');

d38 = pdist2 (B5,S3,'euclidean');

Dist_Between_Senosrs_and_B5 = [d18 d28 d38];

% from Beacon's 6th position

d19 = pdist2 (B6,S1,'euclidean');

d29 = pdist2 (B6,S2,'euclidean');

d39 = pdist2 (B6,S3,'euclidean');

Dist_Between_Senosrs_and_B6 = [d19 d29 d39];

%Adding Gaussian Error

% 1st Beacon Node

d14 = d14+erf(d14);

d24 = d24+erf(d24);

d34 = d34+erf(d34);

% 2nd Beacon Node

d15 = d15+erf(d15);

d25 = d25+erf(d25);

d35 = d35+erf(d35);

% 3rd Beacon Node

d16 = d16+erf(d16);

d26 = d26+erf(d26);

d36 = d36+erf(d36);

% 4th Beacon Node

d17 = d17+erf(d17);

d27 = d27+erf(d27);

d37 = d37+erf(d37);

% 5th Beacon Node

d18 = d18+erf(d18);

d28 = d28+erf(d28);

d38 = d38+erf(d38);

% 6th Beacon Node

d19 = d19+erf(d19);

d29 = d29+erf(d29);

d39 = d39+erf(d39);

% From Cayley - Menger Determinant

a11=d14^2; a12=d24^2; a13=d34^2; a14=-(d14^2-d34^2)*(d24^2-d14^2); a15=-(d24^2-d14^2)*(d34^2-d24^2); a16=1; b1=(d24^2-d34^2)*(d34^2-d14^2);

a21=d15^2; a22=d25^2; a23=d35^2;a24=(-(d15^2-d35^2))*(d25^2-d15^2); a25=(-(d25^2-d15^2))*(d35^2-d25^2);a26=1; b2=(d25^2-d35^2)*(d35^2-d15^2);

a31=d16^2; a32=d26^2; a33=d36^2;a34=-(d16^2-d36^2)*(d26^2-d16^2);a35=-(d26^2-d16^2)*(d36^2-d26^2); a36=1;b3=(d26^2-d36^2)*(d36^2-d16^2);

a41=d17^2; a42=d27^2; a43=d37^2;a44=-(d17^2-d37^2)*(d27^2-d17^2);a45=-(d27^2-d17^2)*(d37^2-d27^2); a46=1;b4=(d27^2-d37^2)*(d37^2-d17^2);

a51=d18^2; a52=d28^2; a53=d38^2; a54=-(d18^2-d38^2)*(d28^2-d18^2);a55=-(d28^2-d18^2)*(d38^2-d28^2); a56=1;b5=(d28^2-d38^2)*(d38^2-d18^2);

```
a61=d19^2;    a62=d29^2;    a63=d39^2;a64=-(d19^2-d39^2)*(d29^2-d19^2);    a65=-(d29^2-d19^2)*(d39^2-d29^2);a66=1; b6=(d29^2-d39^2)*(d39^2-d19^2);

% Augmented Matrix

A = [a11 a12 a13 a14 a15 a16
     a21 a22 a23 a24 a25 a26
     a31 a32 a33 a34 a35 a36
     a41 a42 a43 a44 a45 a46
     a51 a52 a53 a54 a55 a56
     a61 a62 a63 a64 a65 a66];

Matrix = A;

cond = cond(A);

% Result

B = [b1
     b2
     b3
     b4
     b5
     b6];

% value of x

x = A\B;

% x = pinv(A)*B;
```

% Unknown Inner Distances Between Sensors

% Distance Between S1 & S2

d12 = sqrt(x(3)/(1-x(4)-x(5)));

% Distance Between S1 & S3

d13 = sqrt((x(3)*x(5))/(1-x(4)-x(5)));

% Distance Between S2 & S3

d23 = sqrt((x(3)*x(4))/(1-x(4)-x(5)));

Dist_Between_Senosrs_and_B1 = [d14 d24 d34];

% Final Coordinates Respect of Submerged Sensors S1 , S2 , S3

y2 = d12;

y3 = (d12^2+d13^2-d23^2)/(2*d12);

x3 = sqrt(d13^2-((d12^2+d13^2-d23^2)/(2*d12))^2);

% Matrix Representation of Sensors

```
S = [ 0 0 0
    0 y2 0
    x3 y3 0];
```

% Coordinates of the sensors with respect to the Beacon

% Distance Between Projected Beacon's coordinates P4 and Sensors Using Pythagorean Theorem

D14 = sqrt(d14^2-H^2);

```
D24 = sqrt(d24^2-H^2);

D34 = sqrt(d34^2-H^2);

Dist_Between_Sensors_and_P4 = [D14 D24 D34];

% Coordinates of Beacon

y4 = (D14^2 - D24^2 + d12^2)/(2*d12);

x4 = sqrt(D14^2 - y4^2);

% Projected Coordinates of Beacon -> P4

P4 = [x4 y4 0];

% Coordinates of Beacon B1

z4=H;

S4 = [x4 y4 z4];

% After Cartesians Transformation, Sensors Coordinates with respect to Beacon

fS4 = [0 0 0];                      % Considering Beacon position on Origin

fS1 = [-x4 -y4 -z4];                % Coordinates of S1 respect of Beacon

fS2 = [-x4  (y2-y4) -z4];           % Coordinates of S2 respect of Beacon

fS3 = [(x3-x4) (y3-y4) -z4];        % Coordinates of S3 respect of Beacon

Final_Sensors_Coordinates =  [ fS4

fS1

fS2

fS3];
```

% Distance between 1st Beacon Node and Calculated Sensors Coordinates

```
Cal_d14 = pdist2(fS4,fS1,'euclidean');

Cal_d24 = pdist2(fS4,fS2,'euclidean');

Cal_d34 = pdist2(fS4,fS3,'euclidean');

Cal_Dist_Between_Sensors_and_B1 = [Cal_d14  Cal_d24  Cal_d34];
```

%Part: 2 :

% distance from beacon's current position to 1st sensor's previous position on X-Axis

```
a1=50 ;a2=60 ;a3=70; a4=80; a5=90; a6=100;

 b1=70;b2=70;b3=70; b4=70;b5=70;b6=70;

 c1=80 ;c2=90 ;c3=100;c4=71;c5=88;c6=83;

A1 = ((a1^2-b1^2-c1^2)/(-2*b1*c1));

A2 = ((a2^2-b2^2-c2^2)/(-2*b2*c2));

A3 = ((a3^2-b3^2-c3^2)/(-2*b3*c3));

A4 = ((a4^2-b4^2-c4^2)/(-2*b4*c4));

A5 = ((a5^2-b5^2-c5^2)/(-2*b5*c5));

A6 = ((a6^2-b6^2-c6^2)/(-2*b6*c6));

AB1=acosd(A1);

ang1=90-AB1;
```

```
AB1=acosd(A1);

ang1=90-AB1;

AB2=acosd(A2);

ang2=90-AB2;

AB3=acosd(A3);

ang3=90-AB3;

AB4=acosd(A4);

ang4=90-AB4;

AB5=acosd(A5);

ang5=90-AB5;

AB6=acosd(A6);

ang6=90-AB6;

q1=80 ;q2=90 ;q3=100;q4=71;q5=88;q6=83;

r1=4;r2=8;r3=12;r4=16;r5=20;r6=24;

distp1=sqrt(q1^2+r1^2-2*q1*r1*cosd(ang1));

distp2=sqrt(q2^2+r2^2-2*q2*r2*cosd(ang2));

distp3=sqrt(q3^2+r3^2-2*q3*r3*cosd(ang3));

distp5=sqrt(q5^2+r5^2-2*q5*r5*cosd(ang5));

distp4=sqrt(q4^2+r4^2-2*q4*r4*cosd(ang4));

distp6=sqrt(q6^2+r6^2-2*q6*r6*cosd(ang6));
```

```
% distance from beacon's current position to 2nd sensor's previous position on Y-axis

xxx1=40 ;xxx2=45 ;xxx3=55; xxx4=58; xxx5=67;xxx6=80;

yyy1=70;yyy2=70;yyy3=70; yyy4=70;yyy5=70;yyy6=70;

zzz1=40 ;zzz2=56 ;zzz3=77;zzz4=88;zzz5=93;zzz6=54;

X1 = acosd((xxx1^2-yyy1^2-zzz1^2)/(-2*yyy1*zzz1));

X2 = acosd((xxx2^2-yyy2^2-zzz2^2)/(-2*yyy2*zzz2));

X3 = acosd((xxx3^2-yyy3^2-zzz3^2)/(-2*yyy3*zzz3));

X5 = acosd((xxx5^2-yyy5^2-zzz5^2)/(-2*yyy5*zzz5));

X4 = acosd((xxx4^2-yyy4^2-zzz4^2)/(-2*yyy4*zzz4));

X6 = acosd((xxx6^2-yyy6^2-zzz6^2)/(-2*yyy6*zzz6));

Yang1=90-X1;

Yang2=90-X2;

Yang3=90-X3;

Yang4=90-X4;

Yang5=90-X5;

Yang6=90-X6;

zz1=40 ;zz2=56 ;zz3=77;zz4=88;zz5=93;zz6=54;

yy1=6;yy2=12 ;yy3=18; yy4=24;yy5=30;yy6=36;

distxx1=sqrt(yy1^2+zz1^2-2*yy1*zz1*cosd(Yang1));

distxx2=sqrt(yy2^2+zz2^2-2*yy2*zz2*cosd(Yang2));
```

```
distxx3=sqrt(yy3^2+zz3^2-2*yy3*zz3*cosd(Yang3));

distxx4=sqrt(yy4^2+zz4^2-2*yy4*zz4*cosd(Yang4));

distxx5=sqrt(yy5^2+zz5^2-2*yy5*zz5*cosd(Yang5));

distxx6=sqrt(yy6^2+zz6^2-2*yy6*zz6*cosd(Yang6));
```

% distance from beacon's current position to 2nd sensor's previous position on out-of-axis

```
f1=34;f2=45;f3=55;f4=65;f5=76;f6=91;

g1=70;g2=70;g3=70;g4=70;g5=70;g6=70;

h1=90;h2=80;h3=77;h4=98;h5=80;h6=50;

F1= acosd((f1^2-g1^2-h1^2)/(-2*g1*h1));

F2= acosd((f2^2-g2^2-h2^2)/(-2*g2*h2));

F3= acosd((f3^2-g3^2-h3^2)/(-2*g3*h3));

F4= acosd((f4^2-g4^2-h4^2)/(-2*g4*h4));

F5= acosd((f5^2-g5^2-h5^2)/(-2*g5*h5));

F6= acosd((f6^2-g6^2-h6^2)/(-2*g6*h6));

Nang1=90-F1;

Nang2=90-F2;

Nang3=90-F3;

Nang4=90-F4;

Nang5=90-F5;

Nang6=90-F6;
```

```
hh1=90;hh2=80;hh3=77;hh4=98;hh5=80;hh6=50;

gg1=8;gg2=16;gg3=24;gg4=32;gg5=40;gg6=48;

distff1=sqrt(hh1^2+gg1^2-2*hh1*gg1*cosd(Nang1));

distff2=sqrt(hh2^2+gg2^2-2*hh2*gg2*cosd(Nang2));

distff3=sqrt(hh3^2+gg3^2-2*hh3*gg3*cosd(Nang3));

distff3=sqrt(hh3^2+gg3^2-2*hh3*gg3*cos(Nang3));

distff4=sqrt(hh4^2+gg4^2-2*hh4*gg4*cosd(Nang4));

distff5=sqrt(hh5^2+gg5^2-2*hh5*gg5*cos(Nang5));

distff6=sqrt(hh6^2+gg6^2-2*hh6*gg6*cosd(Nang6));
```

% After Adding Distance coordinates of sensors with respect to beacon

%Coordinates of S1 with respect to Beacon

```
FS1_1 = [-x4+distp1-4 -y4+ distp1 -z4+ distp1];

FS1_2 = [-x4+distp2-4 -y4+ distp2 -z4+ distp2];

FS1_3 = [-x4+distp3-4 -y4+ distp3 -z4+ distp3];

FS1_4 = [-x4+distp4-4 -y4+ distp4 -z4+ distp4];

FS1_5 = [-x4+distp5-4 -y4+ distp5 -z4+ distp5];

FS1_6 = [-x4+distp6-4 -y4+ distp6 -z4+ distp6];
```

%Coordinates of S2 with respect to Beacon

```
FS2_1 = [-x4+distxx1-6 (y2-y4)+distxx1  -z4+distxx1];

FS2_2 = [-x4+distxx2-6 (y2-y4)+distxx2  -z4+distxx2];
```

```
FS2_3 = [-x4+distxx3-6 (y2-y4)+distxx3  -z4+distxx3];

FS2_4 = [-x4+distxx4-6 (y2-y4)+distxx4  -z4+distxx4];

FS2_5 = [-x4+distxx5-6 (y2-y4)+distxx5  -z4+distxx5];

FS2_6 = [-x4+distxx6-6 (y2-y4)+distxx6  -z4+distxx6];

%Coordinates of S3 with respect to Beacon

FS3_1 = [(x3-x4)+ distff1-8 (y3-y4)+ distff1 -z4+ distff1];

FS3_2 = [(x3-x4)+ distff2-8 (y3-y4)+ distff2 -z4+ distff2];

FS3_3 = [(x3-x4)+ distff3-8 (y3-y4)+ distff3 -z4+ distff3];

FS3_4 = [(x3-x4)+ distff4-8 (y3-y4)+ distff4 -z4+ distff4];

FS3_5 = [(x3-x4)+ distff5-8 (y3-y4)+ distff5 -z4+ distff5];

FS3_6 = [(x3-x4)+ distff6-8 (y3-y4)+ distff6 -z4+ distff6];

x1 = 0;

y1 = 0;

z1 = 0;

x2 = 0;

z2 = 0;

z3 = 0;

%%%for figure

figure,

scatter3(100,90,70,'d','filled', 'b');
```

hold on, scatter3(90,80,70,'d','filled', 'b');

hold on, scatter3(80,70,70,'d','filled', 'b');

hold on, scatter3(-10,60,70,'d','filled', 'b');

hold on, scatter3(-20,-60,70,'d','filled', 'b');

hold on, scatter3(-30,-90,70,'d','filled', 'b');

hold on, scatter3(x1,y1,z1,'*', 'g');

hold on; scatter3(x2,y2,z2,'o','filled','r');

hold on; scatter3(x3,y3,z3,'^','filled','y');

legend('B1','B2','B3','B4','B5','B6','S1','S2','S3');

%figure for S1 sensor with respect to Beacon in its mobile form

hold on, scatter3(-x4+distp1-4,-y4+ distp1,-z4+ distp1,'*','g');

hold on, scatter3(-x4+distp2-4,-y4+ distp2,-z4+ distp2,'*','g');

hold on, scatter3(-x4+distp3-4,-y4+ distp3,-z4+ distp3,'*','g');

hold on, scatter3(-x4+distp4-4,-y4+ distp4,-z4+ distp4,'*','g');

hold on, scatter3(-x4+distp5-4,-y4+ distp5,-z4+ distp5,'*','g');

hold on, scatter3(-x4+distp6-4,-y4+ distp6,-z4+ distp6,'*','g');

%figure for S2 sensor with respect to Beacon in its mobile form

hold on, scatter3(-x4+distxx1-6,(y2-y4)+distxx1,-z4+distxx1,'o','filled','r');

hold on, scatter3(-x4+distxx2-6,(y2-y4)+distxx2,-z4+distxx2,'o','filled','r');

hold on, scatter3(-x4+distxx3-6,(y2-y4)+distxx3,-z4+distxx3,'o','filled','r');

```
hold on, scatter3(-x4+distxx4-6,(y2-y4)+distxx4,-z4+distxx4,'o','filled','r');

hold on, scatter3(-x4+distxx5-6,(y2-y4)+distxx5,-z4+distxx5,'o','filled','r');

hold on, scatter3(-x4+distxx6-6,(y2-y4)+distxx6,-z4+distxx6,'o','filled','r');
```

%figure for S3 sensor with respect to Beacon in its mobile form

```
hold on, scatter3((x3-x4)+ distff1-8,(y3-y4)+ distff1,-z4+ distff1,'^','filled','y');

hold on, scatter3((x3-x4)+ distff2-8,(y3-y4)+ distff2,-z4+ distff2,'^','filled','y');

hold on, scatter3((x3-x4)+ distff3-8,(y3-y4)+ distff3,-z4+ distff3,'^','filled','y');

hold on, scatter3((x3-x4)+ distff4-8,(y3-y4)+ distff4,-z4+ distff4,'^','filled','y');

hold on, scatter3((x3-x4)+ distff5-8,(y3-y4)+ distff5,-z4+ distff5,'^','filled','y');

hold on, scatter3((x3-x4)+ distff6-8,(y3-y4)+ distff6,-z4+ distff6,'^','filled','y');
```

%with error

%figure for S1 sensor with respect to Beacon in its mobile form with error

```
figure,

hold on, scatter3(19.603, -6.929, 11.647, 'o','r')

hold on, scatter3(-4.307, 6.292, -31.086, 'o','r')

hold on, scatter3(-3.2, -6.984, 0.050, 'o','r')

hold on, scatter3(-6.010, -2.428, 6.703, 'o','r')

hold on, scatter3(-13.274, -84.832, -9.462, 'o','r')

hold on, scatter3(-28.919, 6.075, -4.677, 'o','r')
```

```
hold on, scatter3(-x4+distp1-4,-y4+ distp1,-z4+ distp1,'*','g');

hold on, scatter3(-x4+distp2-4,-y4+ distp2,-z4+ distp2,'*','g');

hold on, scatter3(-x4+distp3-4,-y4+ distp3,-z4+ distp3,'*','g');

hold on, scatter3(-x4+distp4-4,-y4+ distp4,-z4+ distp4,'*','g');

hold on, scatter3(-x4+distp5-4,-y4+ distp5,-z4+ distp5,'*','g');

hold on, scatter3(-x4+distp6-4,-y4+ distp6,-z4+ distp6,'*','g');

legend('Error of S1');
```

%figure for S2 sensor with respect to Beacon in its mobile form with error

```
figure,

hold on, scatter3(6.626, 4.096, -1.940,'o','filled','r');

hold on, scatter3(8.649, -1.692, 4.648, 'o','filled','r');

hold on, scatter3(-5.803, -1.663, 5.883, 'o','filled','r');

hold on, scatter3(-5.459, -1.612, -3.123, 'o','filled','r');

hold on, scatter3(-8.836, -1.614,-3.123, 'o','filled','r');

hold on, scatter3(-3.003, -4.413, -2.6, 'o','filled','r');

hold on, scatter3(-x4+distxx1-6,(y2-y4)+distxx1,-z4+distxx1,'^','filled','b')

hold on, scatter3(-x4+distxx2-6,(y2-y4)+distxx2,-z4+distxx2,'^','filled','b')

hold on, scatter3(-x4+distxx3-6,(y2-y4)+distxx3,-z4+distxx3,'^','filled','b')

hold on, scatter3(-x4+distxx4-6,(y2-y4)+distxx4,-z4+distxx4,'^','filled','b')
```

hold on, scatter3(-x4+distxx5-6,(y2-y4)+distxx5,-z4+distxx5,'^','filled','b')

hold on, scatter3(-x4+distxx6-6,(y2-y4)+distxx6,-z4+distxx6,'^','filled','b')

legend('Error of S2');

%figure for S3 sensor with respect to Beacon in its mobile form with error

figure,

hold on, scatter3(-0.544, -0.941, -46.649, 'o','filled','r')

hold on, scatter3(1.499, -1.789, 7.696, 'o','filled','r')

hold on, scatter3(0.702, -1.472, -11.230, 'o','filled','r')

hold on, scatter3(-0.996, 1.469, -16.137, 'o','filled','r')

hold on, scatter3(-2.131, -0.779, 0.357, 'o','filled','r')

hold on, scatter3(41.565, -7.586, 1.285, 'o','filled','r')

hold on, scatter3((x3-x4)+ distff1-8,(y3-y4)+ distff1,-z4+ distff1,'^','filled','y');

hold on, scatter3((x3-x4)+ distff2-8,(y3-y4)+ distff2,-z4+ distff2,'^','filled','y');

hold on, scatter3((x3-x4)+ distff3-8,(y3-y4)+ distff3,-z4+ distff3,'^','filled','y');

hold on, scatter3((x3-x4)+ distff4-8,(y3-y4)+ distff4,-z4+ distff4,'^','filled','y');

hold on, scatter3((x3-x4)+ distff5-8,(y3-y4)+ distff5,-z4+ distff5,'^','filled','y');

hold on, scatter3((x3-x4)+ distff6-8,(y3-y4)+ distff6,-z4+ distff6,'^','filled','y');

legend('Error of S3');