“Model for Accurate Sensor Node Localization using MCB with Speed and Direction Calculation”

A Dissertation Submitted in Partial fulfillment of the requirement
For the award of Degree of

MASTER OF TECHNOLOGY
in
SOFTWARE TECHNOLOGY
Delhi Technological University, Delhi

SUBMITTED BY

AMIT KUMAR
Roll No: 2K11/SWT/01

Under the Guidance of:

Mr. VINOD KUMAR

Associate Professor
Delhi Technological University

DEPARTMENT OF COMPUTER ENGINEERING
DELHI TECHNOLOGICAL UNIVERSITY
BAWANA ROAD, DELHI-110042
2011-2014

CERTIFICATE

This is to certify that the work contained in this dissertation entitled "**Model for Accurate Sensor Node Localization using MCB with Speed and Direction Calculation**" submitted in the partial fulfillment, for the award for the degree of M. Tech. in SOFTWARE TECHNOLOGY at **DELHI TECHNOLOGICAL UNIVERSITY** by **AMIT KUMAR, Roll No. 2K11/SWT/01** is carried out by him under my supervision. This matter embodied in this project work has not been submitted earlier for the award of any degree or diploma in any university/institution to the best of our knowledge and belief.

(Mr. VINOD KUMAR)
Project Guide
Associate Professor
Department of Computer Engineering
Delhi Technological University

ACKNOWLEDGEMENT

First of all, like to thank, my parents and dear friends who are the most helpful in their support that contributed to the successful completion of this project.

I feel privileged to offer my sincere thanks to Mr. VINOD KUMAR, project guide for having his confidence and let me work on project dealing with latest technologies and providing their class support, guidance & encouragement in studying and completing this project.

I would like to have honor to express the profound sense of gratitude and respect to all those who helped me in this project, throughout life cycle. DELHI TECHNOLOGICAL UNIVERSITY, in particular has been the source of inspiration, I like to thanks and acknowledge the effort of all who have contributed significantly to this project.

AMIT KUMAR

(Roll No.: 2K11/SWT/01)

TABLE OF CONTENTS

List Of Figures

List Of Tables

Abstract

Most problematic factor in wireless network is to find exact geographical location of sensor i.e. termed as localization issue. Many works, algorithm have been done is this regard to find accurate sensor location using Sequential Monte Carlo (SMC) method. This uses Seed/Anchor node to locate other sensor nodes but this algorithm, requirement is to have high density of Seed/Anchor nodes for accurate result, and also it deals with problem related to low sampling rate. Many research are done in same line but lacks to address energy requirement. On similar scope, Bounding Box method is designed which is quite efficient in searching time of candidate samples and hence address time issue to search valid sample set faster. In said thesis, we propose Energy Efficient approach that is faster in terms of searching candidate samples and ignore invalid samples efficiently from sample space. Also it takes more samples into account and hence increases localization accuracy. We will monitor direction of movement as well as speed of sensor to predict the next valid position of sample under study and hence we can easily rule out invalid samples in each iteration and achieve high localization accuracy.

Chapter: 1 Introduction

1.1 Objective

The objective is to design a new efficient localization algorithm that can accurately locate the position of the sensor node. Our algorithm based on direction and speed can provide efficient range free solution for localization problem of sensor node.

As we knew that main fundamental issue in designing sensor network is localization - accurately locating the geographical location of sensors. Traditional methods to determine location, can be to use GPS receiver that can share location based on satellite coordinates or manual - like record the installing location of each sensor. As sensors are deployed in masses the GPS option is high is terms of cost and manual location record is almost impossible and inconvenient because of typical geographical topology. Many algorithms based on sensor networks have been proposed in References [8],[7],[12],[15],[16],[13],[14],[17],[10],[11],[18],[9]. These algorithms mainly work on 2 types of nodes. First some special Seed/Anchor nodes, which knows about their current location and share it to other nodes. And second is sensor (Common) nodes who receive location information from Seed nodes and are capable of sending local information. But mostly all such algorithms are focused to static networks and are not addressing mobility i.e. sensor networks. Also many of these algorithms require costly equipment that measuring distance based on reflex time, speed that require additional HW and cost.

In our approach, we intend to have localization where prior deployment of seed nodes is not available and also nodes are distributed in geographical area irregularly. Also d the density of nodes is not determined and on top of it nodes can move uncontrollably and are not static. The mobility factor adds up complexity in determining the location and said approach took this parameter to improve accuracy and also address to reduce number of seeds requirement.

Consider an environment where we have Seed/Anchor nodes and sensor nodes, then we can have 3 scenarios' as below based on their mobility or static position:

1) Nodes static and seeds moving: Example, a military operation involve in dropping sensor nodes from airplane to effected are of land and seed nodes installed on solders dress are moving. Seed nodes periodically send location info to the sensor nodes and sensor node on getting information estimates its location accurately.

2) Nodes moving and seeds static: Example, Sensor nodes are flowing in the river and we have Seeds nodes installed at fixed interval on the banks of river. The sensor nodes calculate their position based on Seed nodes. As sensor nodes are flowing with the river the old location coordinates become obsolete with the time and hence seed node periodic info will be more accurate results on time basis.

3) Both Sensor nodes and seeds moving: This is ad hoc scenario and most generic in nature, where we have to relative track and calculate both Seed and Sensor nodes position.

Some research already done based on Sequential Monte Carlo (SMC) method for mobile sensor networks, have been discussed in reference [1],[19],[2],[20],[4]. The said SMC algorithm is best suited for mobile nodes as can this algorithm exploit mobile node mobility to improve localization accuracy. ON drawback side in this algorithm we have to keep sampling and filtering till we got valid samples and this is very time consuming activity.

1.2 Problem Statement

The objective is to design a new efficient localization algorithm that can accurately locate the position of the sensor node. Our algorithm based on direction and speed can provide efficient range free solution for localization problem of sensor node. Hence this algorithm can be used as replacement of costly GPS (Global Positioning System).

The desired algorithm must be generic and flexible to be customized for any application. It must support scalability based on changing needs and requirements.

1.3 Motivating Factor

Wireless Sensor Networks (WSNs) can be termed as collection of large number of sensors equipped with a processor, sensor capabilities, wireless communication capabilities, memory and (Battery) power source. WSNs have been in use in many fields like environmental monitoring, animal tracking, habitat monitoring, disaster tracking and in field of agriculture as precision agriculture. In almost all deployment, location awareness capability is essential. Most of existing sensor networks, have static sensors but some modern applications need to have mode sensors. Example, in habitat monitoring applications like sensors are attached to zebras and that helped to collect information about their behavior and migration patterns [6]. In other applications sensors can be monitored to get cellular phones reception quality [6].

The localization problem should be addressed by more accuracy in wireless sensor networks because:

1. Many WSN protocols and applications presumed that all sensor nodes in the system are aware of their location.
2. If any critical event is reported then location awareness is most important thing to address that issue.
3. If application is deployed in geographical scope, then this require location awareness to process data correctly.

Traditional methods to determine location, can be to use GPS receiver that can share location based on satellite coordinates or manual - like record the installing location of each sensor. As sensors are deployed in masses the GPS option is high is terms of cost and manual location record is almost impossible and inconvenient because of typical geographical topology. Many algorithms based on sensor networks have been proposed in References [8],[7],[12],[15],[16],[13],[14],[17],[10],[11],[18],[9]. But mostly all such algorithms are focused to static networks and are not addressing mobility i.e. sensor networks.

Some research already done based on Sequential Monte Carlo (SMC) method for mobile sensor networks, have been discussed in reference [1],[19],[2],[20],[4]. The said SMC algorithm is best suited for mobile nodes as can this algorithm exploit mobile node mobility to improve localization accuracy. ON drawback side in this algorithm we have to keep sampling and filtering till we got valid samples and this is very time consuming activity.

In our thesis, we consider all above challenging parameters and propose efficient algorithm that will provide better results in comparison to existing algorithms in terms of localization accuracy.

1.4 Organization of the Dissertation

This thesis work is organized as follows

Chapter 1 describe objective, problem statement, motivation of undertaking this research work.

Chapter 2 focus on Wireless Sensor Networks and challenges associated with Wireless Sensor Networks. Also describe the localization problem and need to have efficient and accurate localization algorithm.

Chapter 3 focus on mobile network model and various localization strategies, which lay foundation of our approach and provide knowledge of various states involved in finding location.

Chapter 4 describe our research work. It show case basic design of our approach and talk about various states involves in our algorithm and how state machine concept is used in iteration to determine location accuracy.

Chapter 5 focus on simulation of our design and parameters used in iterations and explain output with help of graphs. Also show calculated average localization error and standard deviation and in last compare all iterations.

Chapter 6 shares final analysis and outcome of the research and list the problem that is resolved by our algorithm. And the problems which are not addressed can lays foundation for future work in this direction.

Chapter: 2

Literature Review

2.1 Wireless Sensor Network

A WSN is basically a group of sensor nodes that are deployed in ad-hoc manner. These sensor nodes are used to sense physical characteristics like temperature, light, humidity, rain, pollution levels, sunlight etc of that geographical location and share the same to master Base station. On contemporary Base stations send request for data to Sensor nodes, and accepts the response received and interpret data.

Characteristics of a networked sensor include below

(1) Small physical size,
(2) Low power consumption,
(3) Limited processing power,
(4) Short-range communications, and
(5) Small memory.

Individually, these resembles as small resource-constrained devices and looks to be of little value but if deployed in masses in larger area then they can act as most cost efficient solution. These sensors can be placed in very dangerous or uneven geographical location where manual data collection is almost not feasible. And hence with these sensors now Spatial as well as dense monitoring is now feasible. These sensors make ad-hoc network and can be used for collaborative study based on response from users.

Wireless sensor networks provide link between physical geographical world to the digital world. These low-cost sensor nodes will help technology to get characteristic of far location and help to monitor and derive useful information. Who knows next will be different planet and these sensor nodes are providing information of that geographical location. Below are some of the application that can be deployed based on wireless sensor nodes:

- Earthquake monitoring
- Factory automation
- Environmental monitoring
- Home and office controls
- Medicine
- Inventory monitoring
- Security

2.2 WSN Challenges

Although some of application have already given some results, but the wireless sensor networks is giving many challenges to researchers to cover up:

Data storage – Sensors have very small memory and are capable of sampling the environment continuously. As storage is limited, the data cannot be stored permanently. Also in terms of data handling - sensor has to compress, filter and aggregate valid data received from other nodes and reject stale data. So there is need to route data offline to a central server.

Energy efficiency – As these sensors are deployed in large scale and hence need to have energy efficient in terms of power consumption, so that they can be unattended for a long time. Should focus on writing energy-efficient algorithms that saves battery so that life time to sensors increases, but this require optimization at all layers form physical, middle and application layers.

Fault tolerance – Sensor and Seeds nodes should be robust enough against any fault. As there are deployed in almost non reachable geographical location, terrain and hence manual visit for correction is almost impossible. So there should be techniques provided by the system that can enable remote debugging, monitoring or transfer load to other nodes in case any node got malfunction as many nodes may malfunction because they are deployed in most critical environment.

Localization – Location and tracking these are major applications to focus in Mobile Sensor network. TO knew the location of sensors we need to have special equipment like GPS that uses satellite connection to knew the coordinates and hence is on expensive side to deploy. Our approach is based on fact that if some of sensors knew their location then sensors in vicinity to it can determine their location with some factor of accuracy and same factor can be implied in mobile network.

Scalability – Applications developed should be scalable to entertain any situation and should be able to handle large number of sensor nodes. We should use some centralized algorithms that can be deployed in masses easily and also should be scalable to handle any type of situation or problem under survey. Hence deployment and management of large number of sensors should be addressed.

Security – Most important factor to consider in wireless network is security. The date under test or processing or shared over wireless medium if in case is intercepted by any hack then should not be able to interpret the same. So we must have reliable SW that should address privacy and security of data in case some theft is happened. For the same while transferring data we can use encryption technology to ensure in case of theft no data loss occurs.

2.3 Localization Problem

‘The procedure of identifying Nodes location is called localization’

In many applications of wireless network, the success rate depends on the accuracy of node locations. The data coupled with location accuracy is only meaningful in many wireless applications. Consider an application of geographical habitat monitoring; the sensors are dropped from airplane and sensors are capable of sensing environmental changes like humidity, temperature, wind etc. All these parameters are more useful to researches only if they are coupled with location details.

To determine the geographical positions of sensors in a wireless sensor network is called Localization.

Most trivial solution is manual configuration where sensors are installed by human at some predefined locations but with this solution a large number of manpower is required Further it very difficult and sometime not feasible to install sensors in some of the applications like geographical forest or like in earlier case where sensors are dropped from airplane. In both the case exact location on sensor deployment is not known. Hence this manual approach to install sensor at predefined position and then maintaining is very complex.

Another solution can be the introduction of GPS receiver that work on satellite coordinates but having this sensor on masses will increase the deployment cost tremendously. Also with GPS introduction, there is requirement of less energy consumption so that sensor work for longer time. Also as GPS needs to work on satellite so needs to have deployed on open area to get coordinates otherwise need to have assistive GPS that again require another HW and increases cost again. Lastly, the location reported by GPS receive is always coupled with accuracy in terms of meters. The fine tuning of GPS receiver is done to increase accuracy.

Realizing so many challenges in network localization, this dissertation aims to study the problem in sensor networks specially related to mobile Network. We did study the researches done so far and practical problem faced and try to come up with tradeoff between the accuracy and energy cost to achieve localization accuracy in sensor networks.

Chapter: 3

Research Background

3.1 Network Model

We have two kinds of nodes, one is Seed/Anchor node which is capable of knowing it's exact position at any point of time and second is Common nodes which needs to calculate its location based on Seed node location in every unit time. Although both Seed node as well as common node may have been moving but both of them have limited knowledge of their mobility. Hence both nodes are not aware of their moving speed and direction but must be aware of MAX speed v_{max} i.e. in each time unit T nodes can move in any direction with speed v where $0 < v \leq v_{max}$, but both nodes doesn't know actual speed except that their MAX speed is v_{max}.

To start with all modes are deployed randomly over any geographical or network area. For communication between 2 nodes, we define communication range as radius r, the two nodes can communicate only when they are in this radius r. Let's have 2 clarification of neighbors : 1-hop and 2-hop. 1-hop neighbors sensors p are those which can communicate directly with others present in radius r. 2-hop neighbors of sensor p are those which can communicate with the 1-hop neighbors of p directly but not with other sensor q which is not present in communication radius r of sensor p. Let suppose a node q is there which can directly communicate with node p hence q is p’s 1-hop node. Now if there is another node r which can't communicate with p but can communicate with q directly, then r is 2-hop neighbor of p.

3.2 Sequential Monte Carlo Localization

The Sequential Monte Carlo (SMC) method [21] provides solutions based on simulation to estimate the posterior distribution of non-linear discrete time dynamic models. The sample distribution is represented using a set of weighted samples, and the samples are updated gradually on each time unit. In each time unit samples are scanned and updated using the previous samples. And finally new set of samples is validated using the observed seed nodes in current time unit.

The Sequential Monte Carlo Localization (SMCL) algorithm [1], is the first algorithm for addressing localization in mobile sensor networks.

Location Estimation Algorithm [1]:

Initialization: At start node has no information of its location. N denotes the number of samples to maintain and is constant predefined number, based on required accuracy.

$$L_O = \{\text{set of N random locations in the deployment area}\}$$

Steps: Compute new possible location set L_t based on L_{t-1} the possible location set from the previous time step, and the new observations, o_t

$L_t = \{\ \}$

while (size(L_t) < N) **do**

$R = \{l_t^i \mid l_t^i \text{ is selected from } p(l_t|l^i_{t-1}), l^i_{t-1} \in L_{t-1} \text{ for all } 1 \leq i \leq N\}$ **Prediction**

$R\text{filtered} = \{ l^i_t \mid l^i_t \text{ where } l^i_t \in R \text{ and } p(o_t|l^i_t) > 0\}$ **Filtering**

$L_t = \text{choose}(L_t \cup R_{filtered}, N)$

The mobile localization problem can be analyzed in state space as follows. Let t is the discrete time, it resembles the position of node at time t, and o_t resemble observations from seed nodes received between times t-1 and t. A transition equation $p(l_t \mid l_{t-1})$ describes the prediction of node's current position, based on previous position. Observation equation $p(l_t \mid o_t)$ describes the likelihood of the node being at the location in the given time. We will keep estimating recursively in time, the filtering distribution $p(l_t \mid o_0, o_1, \ldots, o_t)$. A set of N samples L_t is used to represent the distribution l_t, and our algorithm recursively computes the set of samples at each time step. Since L_{t-1} resembles all previous observations, we can compute it using only L_{t-1} and o_t.

Initially we assume the node has no information about its position, so the initial samples are selected randomly from all possible locations. At each time step, the location set is updated based on possible movements and new observations. The location of the said node is estimated by computing the average location of all possible nodes in L_t. We assume locations are (x, y) positions in two dimensional space, but this technique could be used for three dimensions or other location representations.

SMCL used 3 step operation for determining location as below:

Initialization: Node has no knowledge about its location in the deployment area. N initial samples L_N are selected randomly to represent p's possible positions.

$$L_0 = \{l_0^1, l_0^2, \ldots\ldots\ldots, l_0^N\}$$

Here N is number of minimum samples to maintain and is a constant.

Prediction: A node starts from the set of possible locations computed in previous step, L_{t-1} and L_t new set of sample is computed using the transition equation. The Transition equation $p(l_t^i \mid l_{t-1}^i)$ is determined from mobility model or other constraints and it is assumed that the node has no information about its speed and direction, but it knows MAX speed v_{max}.

So if l_{t-1}^i is one possible location of a node in previous step, then the current position of node will be within the circular region with origin l_{t-1}^i and radius v_{max}.

In SMCL[1] the Transition equation is given by:

$$P(l_t \mid l_{t-1}) = \begin{cases} 1/\pi v^2_{max} & \text{if } d(l_t, l_{t-1}) < v_{max} \\ 0 & \text{if } d(l_t, l_{t-1}) \geq v_{max} \end{cases} \qquad (1)$$

Where $d(l_t, l_{t-1})$ represent the distance between two samples l_t and l_{t-1} taking at time t-1 and t. Thus the set of n new samples l_t taken in prediction step must have one node that is selected randomly previously in l_{t-1} and this node lie within the circle of radius vmax during previous sampling. The probability of prediction of node's location in successive sampling is very low as we are not having any information of node movement direction. In case we got some information about the node movement direction and speed and we can be in better position to making accurate predictions of valid samples.

Filtering:

In this step as name mention, we remove the invalid locations on the basis of the new observations. Here we assume that all messages are received instantly. Hence once and Seed sensor send location information then all sensor nodes receive the same in real time and these sensor nodes like in radius range. However in real scenario there can be network collusion or signal obstacle, that can drop messages send by Seed Node. Samples weight is computed as $p(l_t^i|o_t)$, where o_t is the current observed seed node at current time. Samples with zero weight are ignored and if in any iteration number of samples goes less then N, then we have to go back to prediction step.

There are four types of seeds:

a) Outsiders: Seeds which are not heard in current or previous sampling.

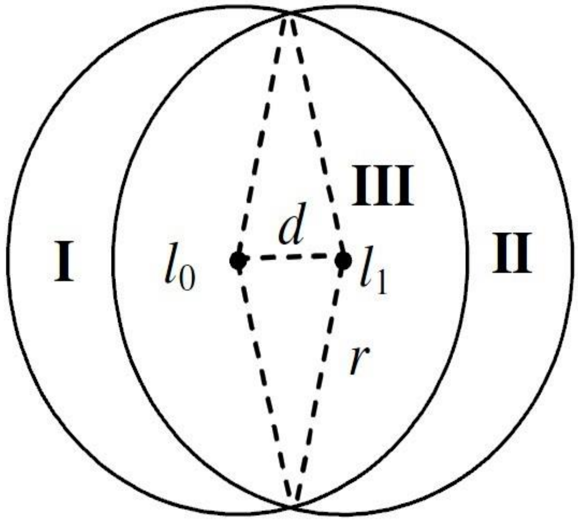

Fig.3.1. Seed Movement

b) Arrivers: Seeds which are not heard in previous sample but are heard in current sample.
c) Leavers: Seeds which are heard in previous sampling but not heard in current sample.
d) Insiders: Heard in both samples and hence is in range radius.

In fig 3.1 the seed movement is from l_0 to position l_1 with time 0 to time 1 respectively. This seed acts as an arriver for region II nodes, and as insider for region III nodes, and for all other nodes outsider, and for region I leave. The most useful information is provided by Arrivers and Leavers that node was within distance r of l_0 at time t_0, and goes out of range distance r of l_1 at time t_1.

Let S represent set of 1-hop seed neighbors of N sample and T represent set of 2-hop seed neighbors of N samples, the filtering condition of l_t is represented as:

$$filter(l_t) = \forall s \in S, d(l, s) \leq r \wedge \forall s \in T, r < d(l,s) \leq 2r \quad (2)$$

The probability distribution is zero, if the filter condition is false and evenly distributed otherwise. Thus, we eliminate the inconsistent locations from possible locations. After filtering, if the possible locations are less than N then prediction and filtering process repeats till we obtain N valid samples. After obtaining N valid samples, p calculates its position as the weighted average of all the samples.

3.3 Monte Carlo Localization Boxed

Monte Carlo have good accuracy, especially when we have low Seed/Anchor Sensors scenario, the efficiency can be further improved. The step of predicting and then filtering took a lot of time and consume a lot of energy from sensor node. Here in Monte Carlo Localization boxed algorithm, use it same MCL algorithm related to Anchor/Seed information of room leavers. As name stated it used Anchor information as well as boxed approach in drawing new sample and the location of sensor.

The Monte Carlo Localization Boxed(MCB)[2] is extended version of SMCL. The steps are similar to MCL and the only difference is how Anchor/Seed information is used and how new samples are sampled. The MCL algorithm in filtering step uses 1-hop and 2-hop neighbor information for rejecting invalid samples. In MCL BOX approach the Anchor/Seed information is used to constrain the sample area, hence this method is fast and easy as compared to MCL because samples once sampled are less like to be rejected. Thereby reducing the number of iterations to be done to find out Valid samples in given scenario.

In SMCL we have two areas, Candidate sample area and Valid sample area (fig 3.2). The Candidate sample area is used to draw new candidate samples into the deployment area whereas the Valid sample area is used to have only valid samples and filter out the invalid candidate samples in prediction step. If the Candidate sample area is large and the Valid sample area is small, the candidate samples drawn in prediction phase have high possibility of being filtered out in the filtering phase. Hence we can conclude that Candidate sample area will be large in case v_{max} is large and Valid sample area will be small in case s_d is large. So both the factors v_{max} and s_d plays an important tole in SMCL and in case both values are larger than SMCL algo will take a lot of iterations to come up with accurate localization.

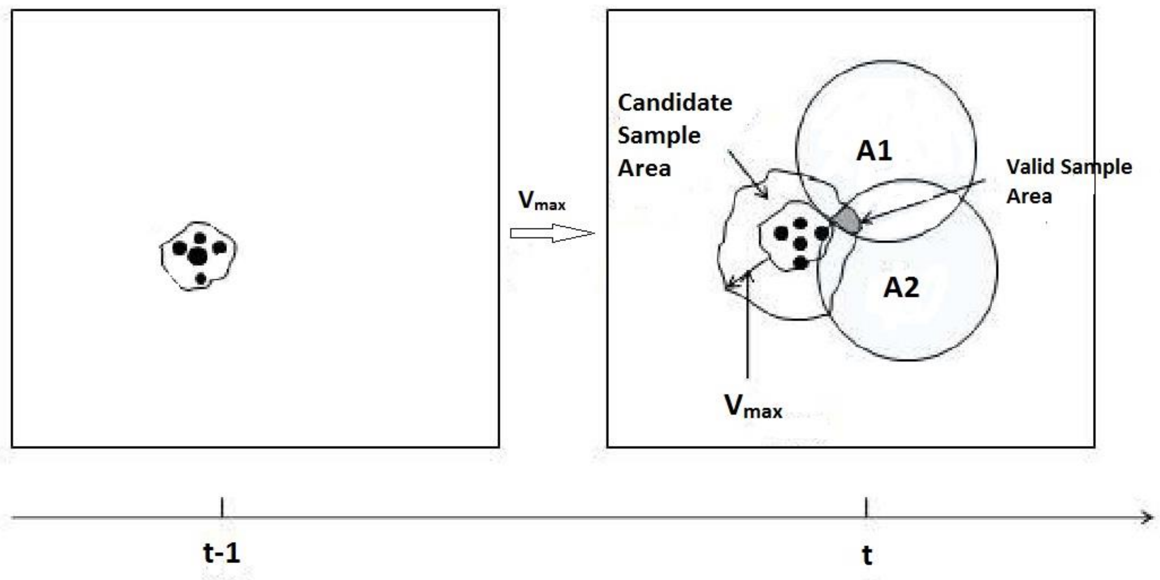

Fig.3.2 How SMCL[1] works

To overcome Candidate and Valid sample area problem, we can use a Bounding Box Method [2] that is an extension of MCL and limit the candidate sample area. This algorithm focus on limiting the Candidate Sample area in to smaller area and then identify the Valid Samples. Infact the best solution can be to have Candidate as well as Valid area as much smaller but in reality it's not possible, so MCLB try to limit the boundary of Samples area and later construct an approximation for Valid sample area are using bounding box.

Building the Bounding Box: Here we deploy a bounding box, as an intersection of Seed nodes example 3 Seeds nodes and bounding box in fig 3.3, where particular Sensor node has Seed/Anchor nodes as its 1-hop or 2-hop. The deployment area where node localization is possible is bounding box region which overlap neighboring Seed/Anchor radio range and create a box.

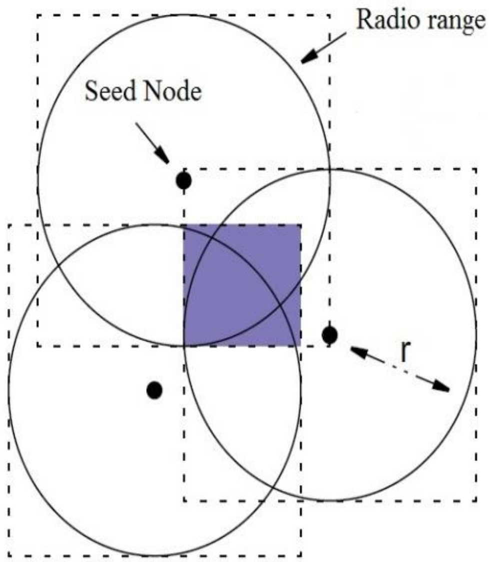

Fig.3.3. MCB[2] shaded region is the valid sample area.

The candidate samples area is reduces using bounding box. Actually SMCB constraints candidate samples into much smaller area upto that extent that this area can be called as valid sample area. Fig 3.3 shows an example of a bounding box, where three one-hop seeds/anchor were heard by a Sensor node. *r* being radio range a Node is considered in square that is build of size *2r* at Seed position for each 1-hop anchor heard. The bounding box consist of coordinates (x_{min}, x_{max}) and (y_{min} , y_{max}) as follows:

$$x_{min} = \max_{i=1}^{n} \{x_i - r\},$$
$$x_{max} = \min_{i=1}^{n} \{x_i + r\},$$
$$y_{min} = \max_{i=1}^{n} \{y_i - r\},$$
$$y_{max} = \min_{i=1}^{n} \{y_i + r\} \quad (3)$$

where (x_i , y_i) is the coordinate of the i^{th} 1-hop seed neighbor. We can further reduce size of bounding box by considering 2-hop seed neighbors and *r* should be replaced with *2r*.

A node has to deploy samples from the bounding box only and hence we can have more accurate location. In MCB the computation cost is on higher side as we have to have more Seed/Anchor and also samples to be reserved for final iteration. Other difference is that in SMB if 1hop samples are absolute ZERO then we can look for 2-hop samples also and hence in this approach we can provide localization support if samples are 2-hop away, which is not possible in SMCL. Hence MCB can still provide localization support in case SMCL is not able to do so, and have better location accuracy than SMCL.

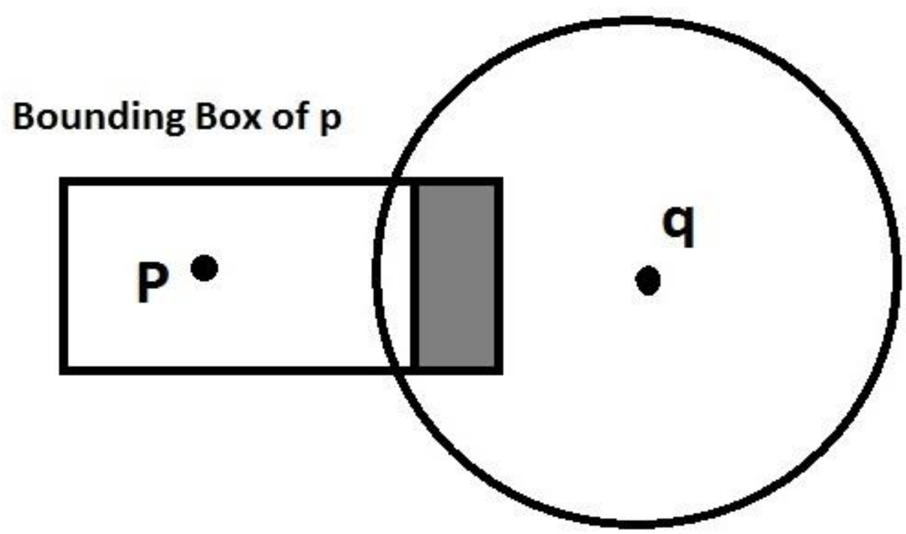

Fig.3.4. Reducing size of Bounding Box

The size of bounding box can be reduced further using negation beacon as follows: Let's have a bounding box (x_{min}, x_{max}, y_{min}, y_{max}) as above. By using 2 beacon negative effect, we can further reduce the size of bounding box (Fig.3.4.). Assume *q* is *p's* 2-hop neighbor, then the shadowed region doesn't contain *p* otherwise *q* will become *p's* 1-hop neighbor. So eliminating shadowed region is possible without having any loss of valid samples.

Chapter: 4

Proposed Approach

In this section, we present our approach which is based on MCL and help to reduce computation cost and increases localization accuracy. Our approach mainly add on direction and speed to the samples. Hence we got the information of the direction of movement of the Sensor node with the help of navigational instrument compass and speed of nodes with the help of accelerometer. Both information direction as well as speed is used in prediction step of MCL to predict N new samples more accurately, hence it improve localization accuracy.

4.1 Compass

A compass is a navigational device that helps to provide direction relative to the surface of the earth. The direction is termed as four cardinal points - north, south, east, and west. Usually, a compass, points to the directions, with names usually abbreviated as N(North), E (East), W (West), S(South) marked on the compass. Compass usually points to N relative to reference location and other locations are determined by angle marks in degrees on the compass. North corresponds to zero degrees, and the angle increase clockwise, hence east corresponds to 90 degrees, south to 180, and west is 270 degrees clockwise.

Fig.4.1 A compass

The magnetic compass used to have magnetized pointer (called as North) which align its direction to the magnetic field of earth and point towards North direction. The magnetic compass The magnetic compass is built as standalone unit with needle turning freely on pivot. The magnetic compass align itself freely along earth magnetic line with one pole pointing toward North and other pole pointing towards south.

4.2 Accelerometer

Accelerometer is a sensor that measures the acceleration to which this is attached. This doesn't necessarily measure the rate of change of velocity but refer the acceleration associated with weight of mass at rest in frame of reference.

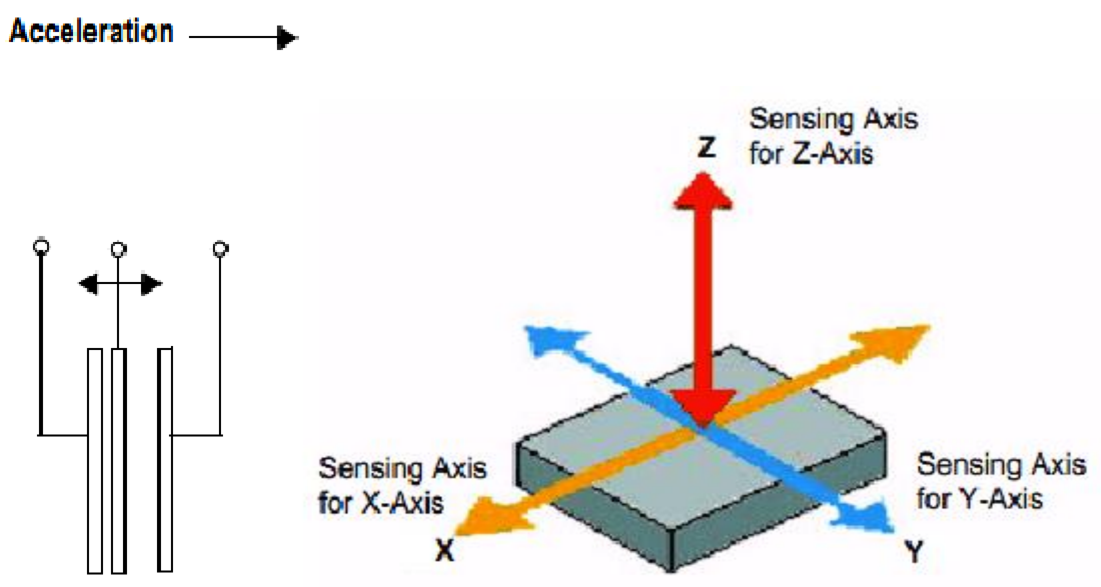

Fig.4.2 Accelerometer principle

Accelerometer can be modeled as set of beams attached to movable central mass that move between the fixed beams. As beam attach to central mass move then distance of central mass will increase from one fixed beam while distance will decrease from other fixed beam. The change in distance is measured as acceleration. The above can also be seen as set of 2 capacitor and as central beam moves with acceleration then capacitance of both capacitor varies that is directly related to acceleration.

4.3 Benefit

We will have both navigational device compass and Accelerometer attached with each sensor node. So that it will have information about its direction of movement as well as acceleration. Now with direction information along with velocity, the predication of next possible position of the sensor node can be done more accurately. In MCL and MCB both we didn't get any information about direction and velocity of sensor nodes, and hence a lot of inaccuracy happen while we predict sensor movement based on prior location reference. This lack of proper prediction affects the localization results badly for both MCL and MCB.

In our approach we will remove this inaccuracy in prediction of the current location by using the direction of movement information provided by the compass and acceleration information provided by accelerometer. Thereby leading us to achieve high localization accuracy.

4.4 Our Approach

MCB is fundamental base of our approach, steps for localization calculations (Prediction, Filtering) is similar to that used in MCB. Only difference come in the prediction phase, where a node starts from the set of possible locations computed in the previous step L_{t-1}, and applies the mobility model to each sample to get a set of new samples in step L_t. The set of new samples obtained in the prediction phase are more accurate as compared to MCB as we have information about the direction of movement as well as velocity of the node. In MCB we do not have any information about the direction of movement and velocity, so MCB takes any random direction for the samples, but with direction as well as velocity we can predict next sample area more accurately in our approach. Hence our approach gives better accurate localization results as compared to MCB.

Steps for localization:

As stated above, the steps for localization are same as MCB, and the differentiation comes in the prediction phase. The detailed steps are as follows:

1. **Initialization:** Node has no knowledge about its location in the deployment area. Initial samples are selected randomly to represent p's possible positions.

$$L_0 = \{l_0^1, l_0^2, \ldots\ldots\ldots, l_0^N\}$$

Here N is a constant which represents the number of minimum samples to maintain.

2. **Prediction:** A node starts from the set of possible locations computed in previous step L_{t-1} and computes a set of n new samples in current step L_t. The node has information about its speed (less than v_{max}) and also the direction of its movement. So, if l_{t-1}^i is one possible location of a node in previous step, then the possible current positions will be in the same direction as information provided by the navigational device compass attached with the node and it must be contained in the circular region with origin l_{t-1}^i (range decided by velocity) and radius v_{max}.

The uncertainty about the node’s location is very less as we have direction as well as velocity of node’s motion. Now we can predict node’s current position in its actual direction of movement, which is not possible in case of MCL or MCB.

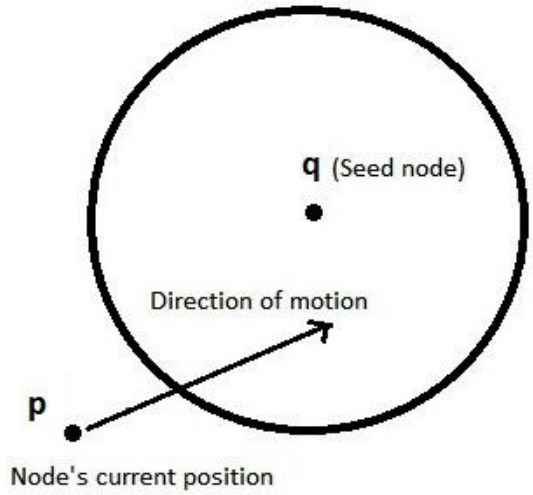

Fig.4.3. Prediction Example

In fig 4.4, current position of node p is shown, now to predict next position we have information about its direction of motion as well as velocity. So we will predict its next position in that direction only considering velocity.

3. **Filtering:** In filtering, we remove the invalid samples which are inconsistent with the current observation. In our approach as we will have information about the direction of motion, we will remove all those samples which are not in the same direction as provided by the navigational device compass. Thus, we eliminate the inconsistent locations from possible locations. After filtering, if the possible locations are less than N then prediction and filtering process repeats till we obtain N valid samples. After obtaining N valid samples, p calculates its position as the weighted average of all the samples.

Chapter: 5
Experimental Results

In this section, we will evaluate the performance of proposed approach through extensive simulations, by monitoring how it check location errors, and results varies with various network and algorithm parameters. The main objective for evaluating a localization approach, is the location estimation accuracy versus the deployment area and communication cost. The location accuracy can be improved by increasing the density of the seeds Anchor/Seed Sensors but to determine appropriate deployment parameters the tradeoffs need to be understood.

5.1 Simulation Parameters

In our experiments, we vary parameters of both the sensor network and sensor nodes.
The various simulation parameters are as follows:

1) **The terrain area:**

   For all our experiments, the deployment area is of 500m x 500m rectangular region. The sensor nodes are distributed randomly in rectangular region.

2) **The node information:**

   Node Density (n_d) - Node density is the average number of nodes including common nodes and seeds modes in 1-hop range. For our experiments, we have taken Node Density n_d = 10.

Seed Density (s_d) – The average number of seed nodes in 1-hop transmission range is called as Seed Density. We have taken Seed Density $s_d = 1$

In our experiments, as per the above Node and Seed density, we have taken 320 Nodes which includes both the common node and the seed nodes, so we have 32 seed nodes.

3) The Mobility Model:

We have implemented the movement of sensors, as per Hu and Evans[1] which states that nodes can't stop in between or push during movement between 2 points. Speed can be classified as below:

Speed of Nodes and Seeds (v_{max} , v_{min} , s_{max} , s_{min})

We have represented speed as the moving distance per time unit. We have taken minimum speed for both nodes and seed as 0 and the maximum speed is taken as 10. A nodes speed is randomly selected from [v_{max} , v_{min}] and the seeds speed from [s_{max} , s_{min}].

4) The Radio Model:

We have set the communication range as 50 for both common and seeds nodes for our experiments. We have assumed that a node can calculate if it lies in radio range *r* of particular seed node but in not capable of calculate the distance of nodes from seed node. We also assumed that the radio range of nodes and seeds as a perfect circle of radius *r*.

5.2 Simulation Results

We will analyze the localization error based on our simulation results. The localization error will be the difference between the estimated location of the node and the actual location of the node. As we have used the random mobility model for our experiments, the movement of the nodes in the deployment area is random, hence the results of our experiments is also random. So we have taken results for 5 iterations, in which each iteration is of 100 steps. In each iteration we have considered result for 100 steps of the node, we have calculated error in each step, and after 100 steps we have calculated average of all 100 steps for each iteration and considered it as the average localization error for that iteration. Now when all the 5 iterations are finished then we have calculated average error for all the iteration, and considered it as the average localization error for our algorithm.

In order to show the simulation results we have plot a graph for each iteration. The graph is plotted between localization error and the steps for each iteration. In the end of this section we have shown results in tabular form. In which we have shown result of each step for different iterations.

1) Iteration 1

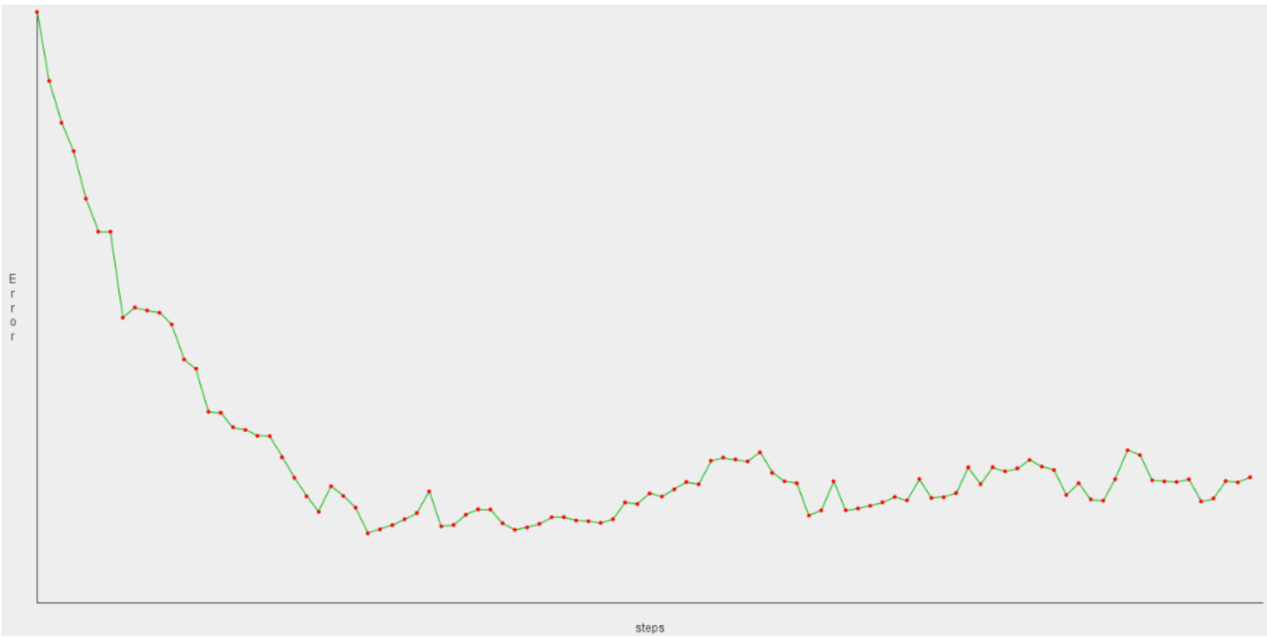

Fig.5.1 Simulation Result: Iteration 1

In iteration 1, the average estimation error in Modified MCB algorithm is: 25.84. And, the standard deviation in modified MCB algorithm is: 15.12

```
in iteration 1, in Step 99......
node 36, not enough sample
node 45 has no sample points
node 52, not enough sample
node 52, not enough samples
node 58 has no sample points
In step 99, in Modified MCB algorithm, the average estimation error is:
25.84
```

2) Iteration 2

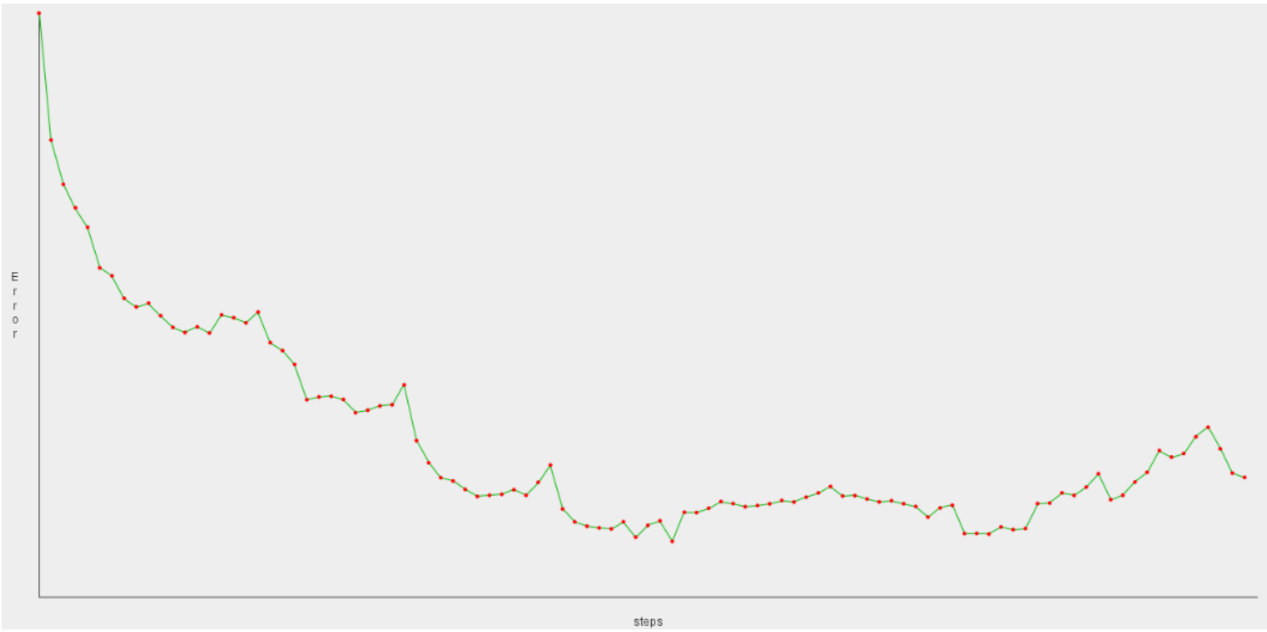

Fig.5.2 Simulation Result: Iteration 2

In iteration 2, the average estimation error in Modified MCB algorithm is: 24.82. And, the standard deviation in modified MCB algorithm is: 16.63.

```
in iteration 2, in Step 99......
node 49, not enough sample
node 49, not enough samples
In step 99, in Modified MCB algorithm, the average estimation error is:
24.82
```

3) **Iteration 3**

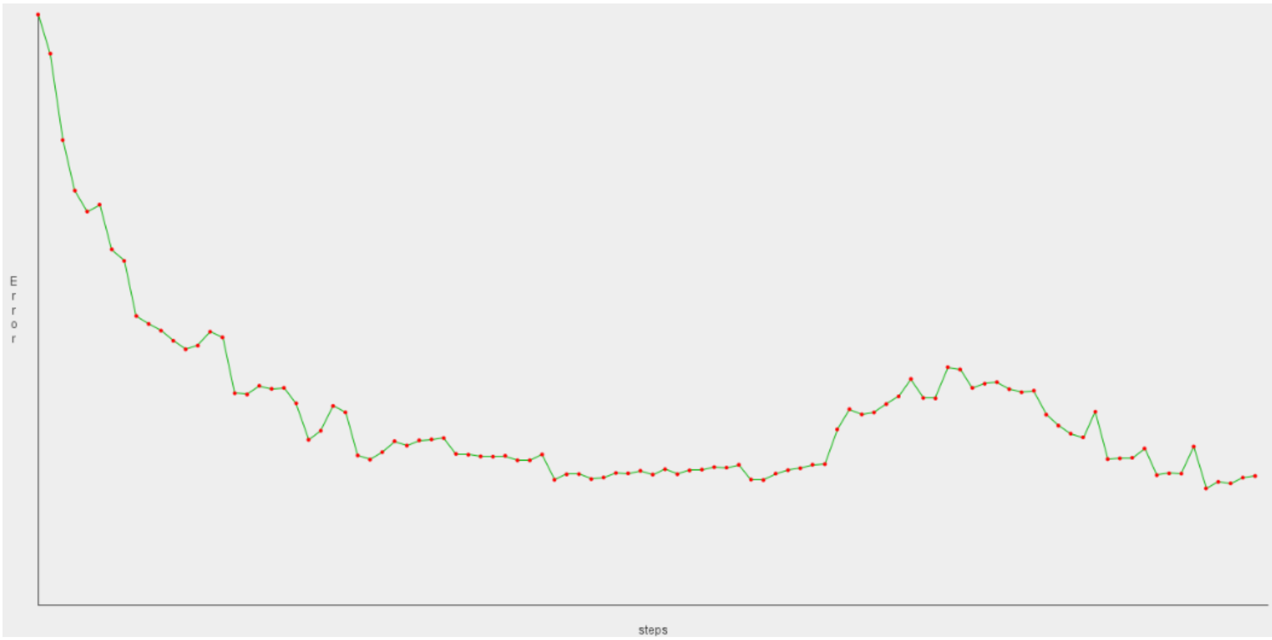

Fig.5.3 Simulation Result: Iteration 3

In iteration 3, the average estimation error in Modified MCB algorithm is: 21.76. And, the standard deviation in modified MCB algorithm is: 17.26

```
in iteration 3, in Step 99......
In step 99, in Modified MCB algorithm, the average estimation error is:
21.76
```

4) Iteration 4

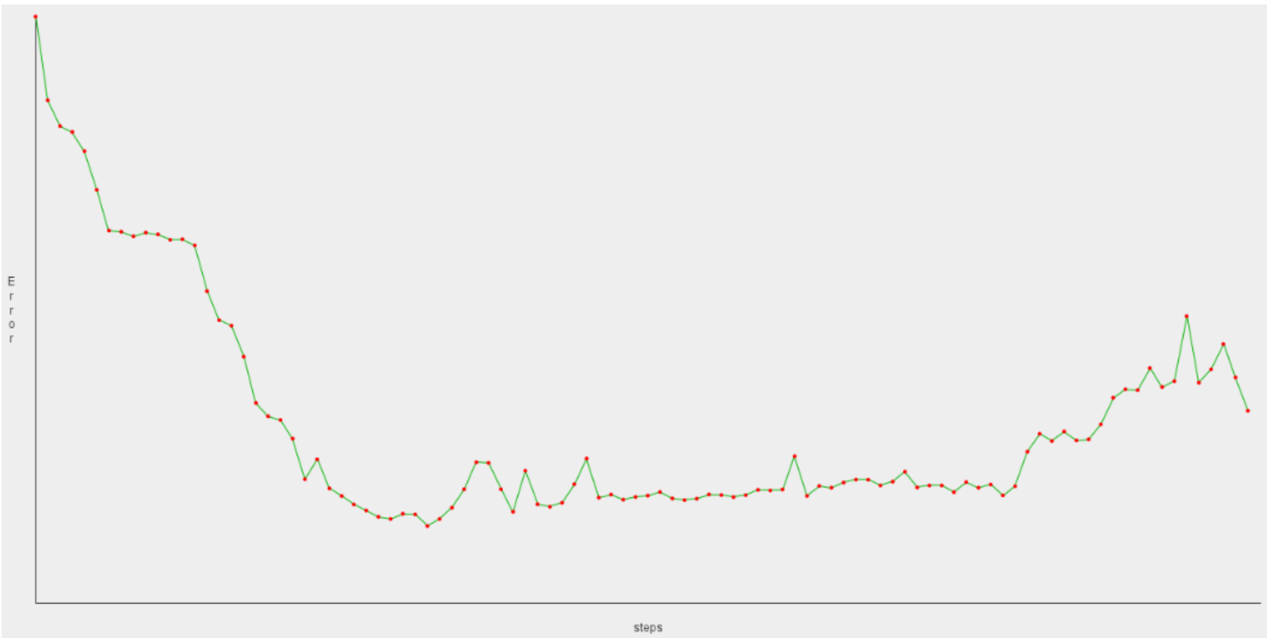

Fig.5.4 Simulation Result: Iteration 4

In iteration 4, the average estimation error in Modified MCB algorithm is: 22.75. And, the standard deviation in modified MCB algorithm is: 15.34

```
in iteration 4, in Step 99......
node 49 has no sample points
In step 99, in Modified MCB algorithm, the average estimation error is:
22.75
```

5) Iteration 5

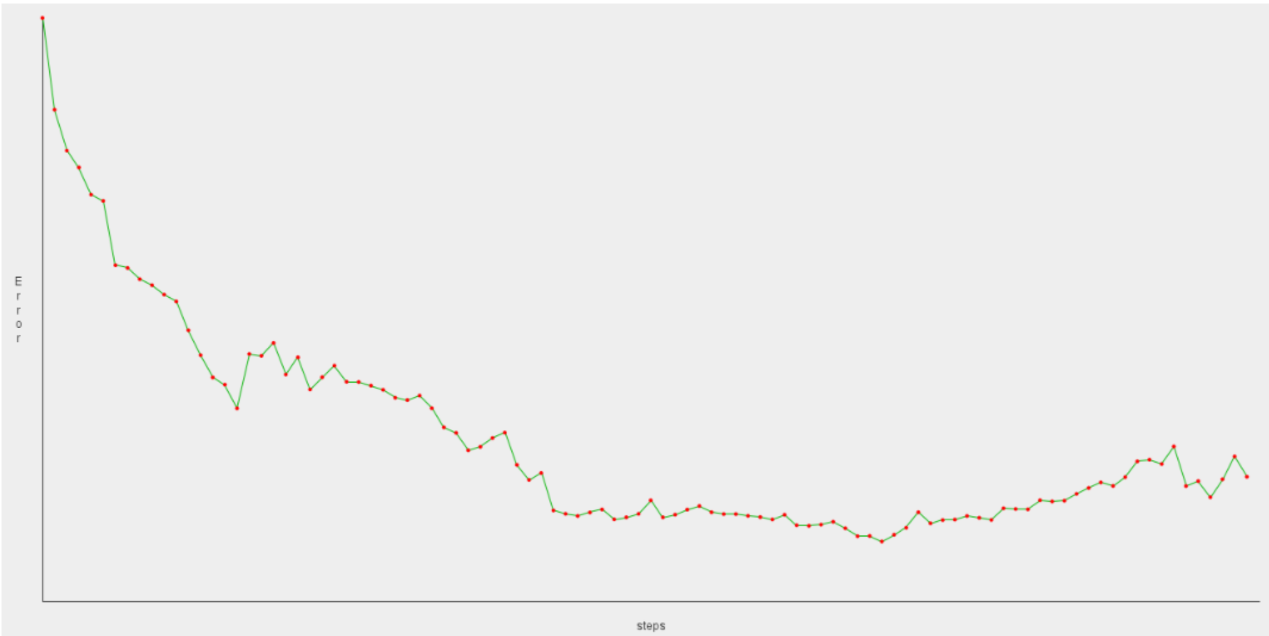

Fig.5.5 Simulation Result: Iteration 5

In iteration 5, the average estimation error in Modified MCB algorithm is: 23.07. And, the standard deviation in modified MCB algorithm is: 15.08

```
in iteration 5, in Step 99......
node 37 has no sample points
In step 99, in Modified MCB algorithm, the average estimation error is:
23.07
```

6) Average for all 5 Iteration

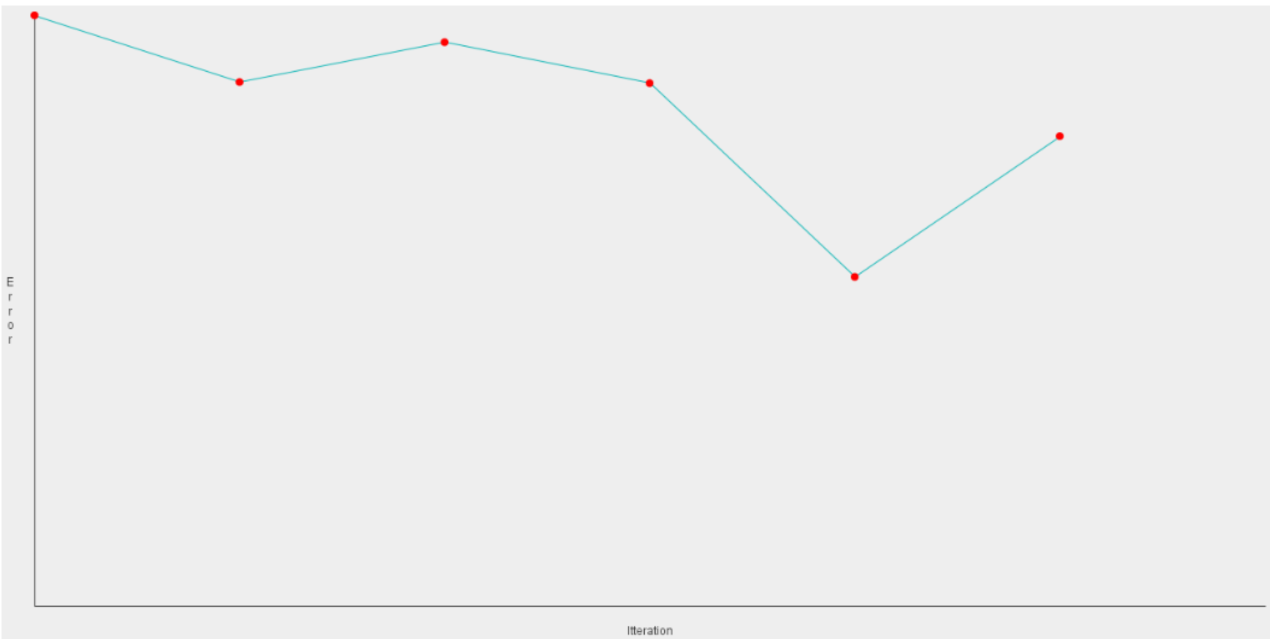

Fig.5.6 Simulation Result: Average for all Iterations

Average estimation error in Modified MCB algorithm is: 22.29044246142673

We can show our simulation results in the tabular form as follows:

Serial No	Iteration Number	Estimated Error
1	Iteration 1	25.84
2	Iteration 2	24.82
3	Iteration 3	21.76
4	Iteration 4	22.75
5	Iteration 5	23.07
6	Average for all 5 Iteration	22.29

Table 5.1. Estimated error for all Iterations

Serial No	Iteration Number	Standard Deviation
1	Iteration 1	15.12
2	Iteration 2	16.63
3	Iteration 3	17.26
4	Iteration 4	15.34
5	Iteration 5	15.08

Table 5.2. Standard Deviation for all Iterations

Chapter: 6 Conclusion and Future Work

6.1 Conclusion

Localization in wireless network has received much attention in the past years. In this thesis, we have presented an efficient and accurate range-free localization algorithm for wireless sensor networks which can be deployed in both mobile sensor as well as static sensor Network. The problem with existing localization algorithm is high computation cost and low accuracy.

The proposed algorithm improves the performance of existing Monte Carlo (MCL) Localization algorithm. In our proposal, we have used information about direction of the movement of the node along with speed, so that we are able to predict the next position of the node more accurately and faster as compared to existing Monte Carlo Localization algorithm. We have attached a device called Compass to each node so that, we can easily get this information about direction of the movement and accelerometer to get the acceleration of every node. Our proposal outperforms existing Sequential Monte Carlo (SMC) based algorithms in terms of location accuracy. Our proposed algorithm can produce more accurate localization results under high node density. Also, even when there are a very few seed nodes, most nodes still get accurate position estimations.

6.2 Future Work

In the future, we are planning to further enhance the performance of our algorithm. We have used accelerometer and Compass to get high localization accuracy but it has increased the cost factor. Many issue remain to be explored in future including the most appropriate compass and accelerometer to be used for mobile sensor network. We need to understand the tradeoff between the accuracy and cost for proposing excellent localization algorithms for sensor networks.

We can also try to improve the performance of our algorithm by considering the information of the sensors mobility patterns. If we know the mobility pattern, then it will help us a lot in the prediction phase. This will make our prediction more accurate, hence we can achieve high localization accuracy.

We have to think of a solution which will produce accurate localization results and it must be energy and cost efficient as well.

Publications

[1] Paper 1:

Conference: 4th INTERNATIONAL CONFERENCE On "Software Solutions, E-Learning, Computer Sciences, Information & Communication Technology" (SECICT–2014) was held at Jawaharlal Nehru University, New Delhi, on 14th -15th June, 2014

Paper Title: Model for accurate Sensor Node Localization using MCB with Speed & Direction calculation.

Author: Vinod Kumar, Amit Kumar, Satyendra Yadav

Online link: http://www.ripublication.com/irph/VolumeSPL/ijictspl.htm

[2] Paper 2:

Journal: International Journal of Engineering Science and Innovative Technology ISSN: 2319-5967

Paper Title: Model for accurate Sensor Node Localization using MCB with Speed & Direction calculation.

Author: Vinod Kumar, Amit Kumar, Satyendra Yadav

Online link: We have got the acceptance, publication awaited.

Acceptance Letter Paper ID:	IJESIT1909201404_1845
Paper Title	Model for accurate Sensor Node Localization using MCB with Speed & Direction calculation
Author Name ‘s	Vinod Kumar, Amit Kumar, Satyendra Yadav

References

[1] L. Hu and D. Evans. Localization for mobile sensor networks. In Tenth International Conference on Mobile Computing and Networking (MobiCom'04), pages 45–57, Philadelphia, Pennsylvania, USA, Sept. 2004. - *describe MCL algrothim.*

[2] Aline Baggio and Koen Langendoen. Monte-carlo localization for mobile wireless sensor networks. In MSN06, pages 317–328, 2006.- *Describe MCB algrothim.*

[3] Z. Shigeng, J. Cao, C. Lijun, and C. Daoxu, "Location Nodes in Mobile Sensor Networks More Accurately and Faster," Proc. Ann. IEEE Conf. sensor and Ad Hoc Comm. and Networks, 2008 - *Weighted MCB algrothim.*

[4] M. Rudafshani and S. Datta, "Localization in Wireless sensor Networks," Proc. Sixth Int'l Conf. Information Processing in Sensor Networks (IPSN '07), pp. 51-60, 2007. *MCB with position of sensor.*

[5] P. Juang, H.Oki, Y.Wang, M. Martonosi, L.S.Peh and D. Rubenstein. Energy-efficient computing for wildlife tracking: design tradeoffs and early experiences in zebranet. In proceedings of the 10th international conference on Architectural support for programming languages and operating systems(ASPLOS), pages 96-107, 2002. *Zebra network with moving sensors and data collection.*

[6] David K. Goldenberg, Pascal Bihler, Ming Cao, Jia Fang, Brian D.O. Anderson, A. Stephen Morse, and Y. Richard Yang. Localization in sparse networks using sweeps. In Proceedings of the 12th annual international conference on mobile computing and networksing (MobiCom' 06), 2006.

[7] Tian He, Chengdu Huang, Brian M. Blum, John A. Stankovic, and Tarek Abdelzaher. Range-free localization schemes for large scale sensor networks. In Proceedings of the 9th Annual International Conference on Mobie Computing and Networking(Mobicom'03), pages 81–95, 2003.

[8] P. Biswas and Y Ye. Semidefinite programming for ad hoc wireless sensor network localization. In Proc. of 3rd International Symposium on Porcessing in Sensor Networks(IPSN), 2004.

[9] Lance Doherty, Kristofer S.J. Pister, and Laurent El Ghaoui. Convex position estimation in wireless sensor networks. In Proceedings of the 20th Annual Joint Conference of the IEEE Computer and Communications Societies (Infocom'01), volume 3, pages 1655–1663, 2001.

[10] Mo Li and Yunhao Liu. Rendered path: Range-free localization in anisotropic sensor networks with holes. In Proceedings of the 13th annual ACM international conference on Mobile computing and networking(Mobicom'07), pages 51–62, Montreal, Que´ebec, Canada, September 2007. ACM.

[11] H. Lim and J.C. Hou. Localization for anisotropic sensor networks. In Proc. of 24th Annual Joint Conference of the IEEE Computer and Communications Societies(INFOCOM 2005), pages 138–149, 2005.

[12] Dragos Niculescu and Badri Nath. Ad-hoc positioning system(aps). In Proceedings of 2001 IEEE Global Telecommunications Conference (Globecom'01), pages 2926–2931, San Antonio, TX, USA, May 2001. IEEE Communications Society.

[13] Andreas Savvides, Chih-Chieh Han, and Mani B. Srivastava. Dynamic fine-grained localization in ad-hoc sensor networks. In Proceedings of the 7th Annual international Conference on Mobile Computing and Networking(Mobicom'01), pages 166–179, Rome, Italy, 2001. ACM.

[14] Y. Shang and W. Ruml. Improved MDS-Based localization. In Proc. of 23rd Annual Joint Conference of the IEEE Computer and Communications Societies(Infocom'04), pages 2640–2651, 2004.

[15] Y. Shang, W. Ruml, Y. Zhang, and Fromherz MPJ. Localization from mere connectivity. In Proceedings of the 4th ACM Internatioal Symposium on Mobile Ad Hoc Networking and Computing(Mobihoc'03), 2003.

[16] Chen Wang and Li Xiao. Locating Sensors in Concave Areas. In Proceedings of 25th IEEE International Conference on Computer Communications(Infocom'06), pages 1–12, April 2006.

[17] Hongyi Wu, Chong Wang, and Nian-Feng Tzeng. Novel selfconfigurable positioning technique for multihop wireless networks. IEEE/ACM TRANSACTIONS ON NETWORKING, 13(3):609–621, June 2005.

[18] Bram Dil, Stefan Dulman, and Paul Havinga. Range-based localization in mobile sensor networks. In ewsn06, pages 164–179, 2006.

[19] Jiyong Yi, Sungwon Yang, and Hojung Cha. Multi-hop-based monte carlo localization for mobile sensor networks. In Proceedings of the 4th Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks (SECON’07), pages 162–171, June 2007.

[20] J. E. Handschin. Monte Carlo Techniques for Prediction and Filtering of Non-Linear Stochastic Processes. Automatica 6. pp. 555-563.1970.

[21] http://en.wikipedia.org/wiki/Compass

[22] T. Camp, J. Boleng, and V. Davies. A survey of mobility models for ad hoc network research. Wireless Communications and Mobile Computing (WCMC): Special issue on Mobile Ad Hoc Networking: Research,Trends and Applications, 2(5):483-502, 2002.

[23] en.wikipedia.org/wiki/Accelerometer