PARTICLE FILTER APPROACH TO OVERCOME MULTIPATH PROPAGATION ERROR IN SLAM INDOOR APPLICATIONS.

by

Nelyadi Samyak Shetty

A thesis submitted to the faculty of
The University of North Carolina at Charlotte
in partial fulfillment of the requirements
for the degree of Master of Science in
Electrical Engineering

Charlotte

2018

Approved by:

____

Dr. James Conrad

____

Dr. Andrew R. Willis

____

Dr. Thomas P. Weldon

ABSTRACT

NELYADI SAMYAK SHETTY. Particle filter approach to overcome multipath propagation error in SLAM indoor applications. . (Under the direction of DR. JAMES CONRAD)

Robot localization faces the classic chicken or egg problem where it has to either know the location of the robot or the map to localize itself. In most applications this can be using Simultaneous Localization and Mapping(SLAM) through measurements via sensors. Most applications of SLAM involve taking measurements that contain both range and orientation information about its environment. But Range-Only SLAM localization faces an ambiguity because it receives only range measurements from its sensors. Hence in order to implement localization we would need to know atleast an approximate initial location of the landmarks. Previous approaches to this issue involves basic trilateration, probabilistic methods, least squares approximation, multiple hypothesis methods using the Extended Kalman Filter. This however does provide a delay the localization. In order to overcome this there are multiple hypothesis methods that initialize the problem in real time. However, this is computationally expensive and all of these methods do not cope well with multipath noise. Multipath is the propagation phenomenon that results in radio signals reaching the receiving antenna by two or more paths leading to false range measurements.

In this research we explore the problem put forward with the use of radio beacons when applied to RO SLAM and multipath noise that has a non Gaussian distribution. In particular, the use of the particle filter approach that copes with multipath noise as compared to the Extended Kalman Filter that is only applicable to noise that has a Gaussian distribution. This thesis proposes a method using particle filters to perform SLAM and is then compared to the established methods through simulations.

ACKNOWLEDGEMENTS

I would like to thank my advisor, Dr. James M. Conrad for his help and guidance throughout this research and my college career.

I would also like to thank Dr.Andrew Willis for helping over throught the various challenges that I faced through this research, as well as Dr. Tom Weldon, for serving on my committee.

I also want to thank Dr. Samuel Shue and David Grabowsky, who helped set my standards for research, provided technical advice, and friendship.

Finally, I would also like to thank my parents, whose constant support both financially and emotionally has made it possible to get this far. I could not have done this without them.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

EKF	Extended Kalman Filter
RO	Range Only
RSSI	Received Signal Strength Indicator
SLAM	Simultaneous Localization And Maping

CHAPTER 1: INTRODUCTION

Recent advancements in sensors, embedded devices and wireless communication have propelled the field of research regarding robotic autonomy. Because of this industrial need, monumental leaps have been made in everything from intelligent navigation solutions to automated product assembly development. Examples of these solutions include Kuka robots that deploy autonomous logistic navigation in Amazon warehouses and autonomous assembly line robots that manufacture products. Robots in these applications have to accommodate for the erroneous uncertainty in their environment as every environment tends to be highly unpredictable. There are several sources of error, such as sensor noise and control noise, that cause great uncertainity for a robot. Most systems use probabilistic algorithms to estimate their actions to a certain degree. One of the popular actions are ones that involve robot localization.

Localization refers to estimating a robot's position in respect to a global map. The robot obtains its position through odometry measurements given to the robot and sensor measurments obtained from its surroundings. Only once it establishes its position can it decide on the optimal action needed. These odometry and sensor measurements tend to be highly noisy. Hence the positions are represented with a certain level of confidence or certainity. Therefore acquiring its position allows it to establish its momentary uncertainty which determines future uncertainity and only then can it perform the necessary actions to complete a task successfully. However robots face the delima of either knowing the map in order to localize itself. In most industrial applications today various changes to the enviornment are made to help with the localization of the robot. Magnetic strips on the floor or visual markers are placed around the robots enviornment in industrial applications. The robot uses these markers along with prior knowledge of their preset positions in the environment, estimates its distance to the landmarks and localizes itself. This method of localization is impractical as it requires make physical changes in the enviornment or provides a map prior in advance to the robot. SLAM attempts to remove the need for knowledge of the prior positions of the landmarks while simultaneously building a map and localizing itself in it.

SLAM uses the odometry control information given to the robot to predict its position. As these control inputs tend to be highly noisy it furthur tends to correct is position incorporating measurement information from its surroundings. Simultaneous localization and mapping (SLAM) uses probabilistic estimation algorithms that are used by a robot to estimate both its position and also map its environment which is very crucial for path planing algorithms. The robot uses sensors to collect measurement from distinguishable objects in its environment (landmarks) and then determines the position and uncertainty of the robot and landmarks.

A landmark can be any distinguishable feature in the enviornment like a wall, corner, a visual marker or signal measured by the robot. The measurement in the case of a visual feature would generally be through cameras or laser scanners mounted on the robot. In the case of signals it is obtained through a radio transmitter mounted on top of the robot. The measured data using visual features usually gives both orientation and range measurments of its enviornment whereas the radio signals only provide range information. The data obatined from the sensors furthur undergoes processing to identify the differences in the features in an enviornment. The SLAM algorithm that uses radio signals is called Range Only SLAM. In our research we focus on Range-Only SLAM using radio beacons as landmarks. RO SLAM involves receiving only the range data from the sensors when compared to most other implementations where both range and orientation is derived from the measurement. In our application we use signals received from radio beacons to implement SLAM. Here the measured data is the Received Signal Strength Indicator (RSSI) received from the radio beacon. This is used to determine the range measurement to the landmark.

There are different methods of implementing SLAM based on the type of sensor used and filtering method. One of the most common sensors used are range finders such as sonar and lidar, however, recent advances in computing efficiency and parallel computing has resulted in the use of cameras as sensors employing various image processing techniques to build maps. In addition, SLAM methods also vary based on the type of filtering method involved. The filtering methods differ in the way on how they handle uncertainty when they identify landmarks, building a map, or deal with robot motion. Extended Kalman filter (EKF)[3], GMapping [4], Graph SLAM and Particle Filter just to name a few. Each of these algorithms use different filtering methods to handle uncertainties which have their own advantages and disadvantages.

1.1 Motivation

Most SLAM applications tend to run into memory and computational issues as the numbers of landmarks in the enviornment increases. It also faces data association problems where the landmarks in the enviornment become indistinguishable from one another. Utilizing radio beacons as landmarks largely solves memory scaling problems that occurs in most algorithms, since the number of transceivers is finite, limiting the number of landmarks to the number of devices within the network. In addition, the landmark association problem which involves distinguishing different landmarks from one another is entirely solved as each landmark has a network address associated with it. This network address is transmitted in the data packet along with signal. Most sensors require line of sight measurements in order to identify landmarks which is not required when using wireless beacons, also reducing the need for numerous landmarks in the map.

Some of the most popular methods to implement RO SLAM such as the EKF and SEIF use the core bayesian estimation principle that assumes a gaussian representation of data. In addition they require an approximate estimate of the landmarks before application as an initial position of the landmark cannot be determined from range information alone. Multipath fading is also a huge problem when estimating range from wireless beacons, especially in indoor enviornments. Multipath fading occurs when a signal takes multiple paths from the transmitter to the receiver. Depending on the phase of signals when they arrive together, they can either cause either large attenuation or large amplification of the signal. When the signal is suddenly amplified, the model will not hold and will generate highly erroneous readings. Some models account for multipath, but are intended for long range transmission, such as transmission from a cell tower. Multipath fading effects in short range indoor environments are much more difficult to predict due to the many factors that effect its behavior. For mobile robot operating environments, such as a warehouse or a private home, short range multi path is the most prevalent problem in RSSI distance estimation. Most common SLAM methods reply on a Gaussian distribution of the signals and cope poorly with the noise presented by multipath as it tends to be highly non Gaussian. In addition, RO EKF has a high probability of falsely intializing the landmarks in highly noisy indoor environments which also is problematic in common SLAM methods. In this work, we implement a RO SLAM algorithm using range beacons with a emphasis on the multipath problems that occur during its utilization in a indoor environment.

1.2 Objective of this work

The main objective of the work is to implement a SLAM method that functions efficiently even in the presense of multipath signals occur and unknown noise distributions. In addition the SLAM method would also avoid false intializations by dynamically converging to the radio beacons. Thus, it solves the false initialization problem and also avoids the high probability of error due to multipath.

1.3 Contribution

This work provides an in-dept study into the various advantages and disadvantages of utilizing radio beacons for localization in indoor environments. Different localization techniques are discussed with respect with their effectiveness in indoor applications. A novel configuration is introduced to overcome the multipath problems using a particle filter for both the landmarks and the robot thus reducing the effects of non-Gaussian signal error in the localization process. A simulation is built using the new method and metrics such as estimation error and computation time are collected. Simulations are built for the established methods such as Range-Only EKF SLAM and dead reckoning. These are compared with the proposed method and a conclusion is presented.

1.4 Organization

This thesis is organized into five sections. The first chapter provides the motivation for the work presented here. The second chapter provides a background on RO EKF SLAM and the various works related to beacon initializations already present and their drawbacks. The third chapter introduces the proposed new method using Particle Filter SLAM and also the simulation conditions. The fourth chapter covers the simulation of all the methods and the metrics collected. The final chapter concludes by discussing the shortcomings of the particle filter method, the strengths and weaknesses of the simulation method, and the results of the comparisons between the different techniques in the SLAM simulation.

CHAPTER 2: BACKGROUND

Simultaneous Localization and Mapping involves 2 main steps. A prediction step where the next position of the robot is predicted using its kinematic or motion model and the control information given to it. However since odometry information tends to be highy noisy. The second ste is a update step where it tries to correct its position by incorporating the measurement information it obtains from its enviornment. In addition to this SLAM continuously builds a map of the enviornment by appending new landmarks that it encounters. One of the most popular and widely used method uses the Extended Kalman Filter to estimate the robots position and build the map of its enviornment. A high level representation of a general SLAM algorithm is depicted in Figure :2.1.

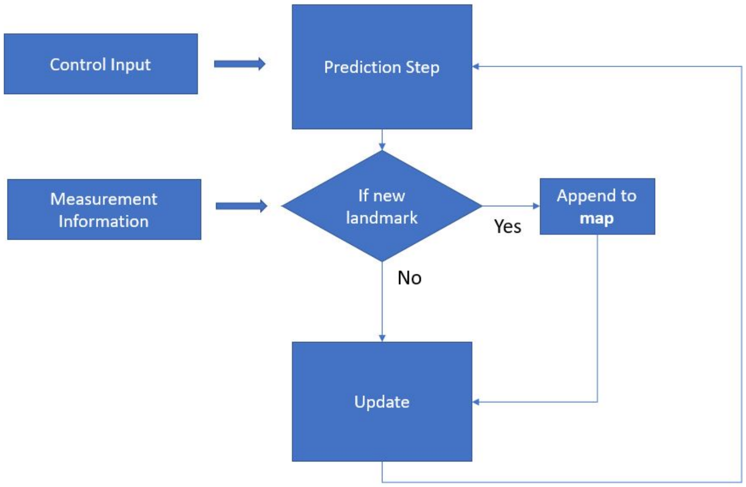


Figure 2.1: Example of a prediction and updation step in SLAM.

2.1 Extended Kalman Filter SLAM

This is one of the most widely used and most reliable implementation of SLAM. It can be easily adapted for various kinds of landmarks, can be easily implemented and computed. EKF is a recursive Bayesian estimation filter which models the input and prediction variables as a Gaussian distributed random variables. In order to understand EKF in a SLAM application the EKF has to be introduced first. The following explains the workings of the basic Kalman filter, the Extended Kalman Filter and concluding with the particulars related to EKF SLAM.

2.1.1 Kalman Filter

The Kalman filter algorithm take a series of noisy measurements over time to estimate the state of unknown variables in a system. It models the noise the signals as Gaussian random variables and updates the state variables each iteration based on the previous state and the uncertainity in measurement. It has 2 phases: a prediction phase and a measurement phase. The prediction phase attempts to update the state variables based on the state transition model which predicts what the state of the variables should be during the next time step and the uncertainities in the variables represented by the error covariance matrix are also updated. The measurement phase corrects the prediction phase by using sensor measurements and taking into account the noise from sensor measurements and the transition model[5].

Algorithm 1 The Kalman Filter $(x_{k-1|k-1}, P_{k-1|k-1}, u_k, z_k)$

```
KF_Prediction:
$x_{k|k-1} = F_k x_{k-1|k-1} + B_k u_k$
$P_{k|k-1} = F_k P_{k-1|k-1} F_k^T + Q_k$
KF_Measurement:
$y_k = z_k - H_k x_{k|k-1}$
$S_k = H_k P_{k|k-1} H_k^T + R_k$
$K_k = P_{k|k-1} H_k^T S_k^{-1}$
$x_{k|k} = x_{k|k-1} + K_k y_k$
$P_{k|k} = (I - K_k H_k) P_{k|k-1}$
return $x_{k|k}, P_{k|k}$
```

2.1.2 The Extended Kalman Filter

A real world environment is highly non-linear. The Kalman filter is only a linear estimator, hence it fails for non-linear applications. However, there are various solutions to the Kalman filter that can adapt it to account for non-linear models. Examples of this are seen in the Extended Kalman Filter, Hybrid Kalman Filter and the Unscented Kalman Filter. One of the most popular non linear forms is the EKF. It accounts for non linearities by linearizing the system through taking he first order taylor series expansion of the observation and state transition model [3]. We obtain the Jacobian's F and H, which contain the first order derivative of a vector function of several variables. They are used to describe how the probability mass is spread during the measurement and prediction processes by approximating the curvature of the non-linear function models with a linear one.

$$F_{k-1} = \frac{\partial f}{\partial x}|_{x_{k-1|k-1},u_k} \tag{2.1}$$

$$H_k = \frac{\partial h}{\partial x}|_{x_{k-1|k-1}} \tag{2.2}$$

The EKF introduces non linearity in its algorithm by replacing the state transition model and the observation matrix by non linear function which describes the state transitions and observation models f and h, in the prediction and measurement steps. Using these the Jacobians are calculated as the values of F and H will change depending on the time of the linearization. Algorithm: 2 shows the EKF in its complete form.

The extended Kalman filter in its complete form is shown in Algorithm: 2. In the section covering EKF SLAM, examples of the f and h functions can be found along with Jacobian calculations based on these functions [3].

Algorithm 2 The Extended Kalman Filter $(x_{k-1|k-1}, P_{k-1|k-1}, u_k, z_k)$

```
EKF_Prediction:
$x_{k|k-1} = f(x_{k-1|k-1}, u_k)$
$P_{k|k-1} = F_k P_{k-1|k-1} F_k^T + Q_k$
EKF_Measurement:
$y_k = z_k - H_k x_{k|k-1}$
$x_{k|k-1} = h(x_{k-1|k-1}, z_k)$
$S_k = H_k P_{k|k-1} H_k^T + R_k$
$K_k = P_{k|k-1} H_k^T S_k^{-1}$
$x_{k|k} = x_{k|k-1} + K_k y_k$
$P_{k|k} = (I - K_k H_k) P_{k|k-1}$ return $x_{k|k}, P_{k|k}$
```

2.1.3 EKF SLAM

One of the most influential and most popular SLAM algorithm is based on the Extended Kalman Filter. EKF SLAM applies the linearization to the kinematic motion model and the measurement model since they are non linear in real world applications. It uses sensor measurements to detect landmarks which is further used to correct odometric error in the robot. Odometry error is obtained from sensors on the robot like the wheel encoders or from sensors used to measure inertial information or from both. However, odometry readings alone aren't reliable because of factors such as wheel slippage or noisy readings from inertial sensors which over time tend to accrued into a large error rendering the information useless. Therefore the EKF utilizes landmarks as inputs to a measurement vector to correct the accrued error from odometry. An example of the prediction and update step is shown in the figure below.

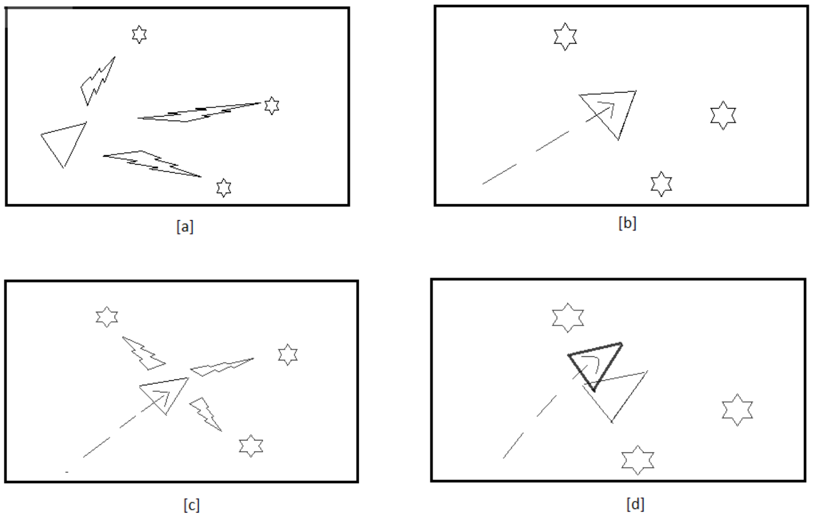


Figure 2.2: Example of a prediction and updation step in SLAM.

In Figure:2.2:a a robot represented by the triangle takes measurements from landmarks represented by the stars. Figure:2.2:b shows the robot moves forward and updates the position from odometry data. In Figure:2.2:c it takes measurments from the landmarks. Figure:2.2:d shows the robot corrects its position (represented by the thicker traingle) based on the data from the landmarks.

Algorithm 3 EKF SLAM $(x_{k-1|k-1}, P_{k-1|k-1}, u_k, z_k)$ [3]

```
$z_0, R_0 = get\_measurements$
**for** $k = 1$ to $steps$ **do**
    $u_k, Q_k = get\_odometry$
    $x_{k|k-1}, P_{k|k-1} = EKF\_Prediction(x_k)$
    $z_k, R_k = get\_measurements$
    $DA_k = data\_association(x_{k|k-1}, z_k, R_k)$
    $x_{k|k-1}, P_{k|k-1} = append\_landmarks(x_{k|k-1}, z_k, R_k)$
    $x_k = EKF\_Update(x_{k|k-1}, z_k, R_k, DA_k)$
**return** $x_{k|k}, P_{k|k}$
```

2.1.4 RO EKF SLAM

Using wireless beacons as landmarks helps reduce the scaling probelms mentioned in regular EKF as the number of landmarks are static. This prevents EKF SLAM from continuously appending landmarks and consuming memory. If the number of beacons is already known at the start it can avoid the appending step of EKF by pre allocating space for all the landmarks. Using nodes also solves the data association problem as each node includes a network address correctly identifying each measurement with each landmark.

In the SLAM algorithm mentioned before it gets both distance and bearing is obtained from measurement. Here when using wireless beacons we can only get range only measurements. This still can be utilized for refining the position of the robot within the world. These algorithms are refered to as RO SLAM algorithms. The EKF based solution is the most common. It uses the same algorithm used for EKF SLAM but only a few modifications to the measurement model, as the measurement is scalar valued rather than vector valued.

$$y_k = z_k - h(x_{k|k-1}, u_k) \tag{2.3}$$

$$h(x_{k|k-1}, u_k) = \begin{bmatrix} (\sqrt{(x_r - x_i)^2 + (y_r - y_i)^2} \end{bmatrix} = \begin{bmatrix} range \end{bmatrix} \tag{2.4}$$

The Jacobian of $h()$ has the bottom row removed, giving the following:

$$H = \begin{bmatrix} \frac{x_r - x_i}{r} & \frac{y_r - y_i}{r} & 0 & \dots & -\frac{x_r - x_i}{r} & -\frac{y_r - y_i}{r} & \dots & 0 & 0 \end{bmatrix} \tag{2.5}$$

A main drawback to RO SLAM is that a approximate location of the beacons must be known in advance for the filter to work. If a false localization occurs it leads to erreous results. To overcome this there are various pre-intialization processes mentioned in 2.1.5. It is also subject to many approximations and limiting assumptions. It makes Gaussian noise assumption for robot motion and perception. Mutipath noise from wireless beacons do not follow the properties of a Gaussian. Also the possibility of having multipath in indoor environments might lead to false localization during the initialization phase. The landmark's initial position is not so easily determined, as the range-only beacon has an unknown orientation relative to the robot. Therefore, the landmark could reside approximately anywhere within the sensed distance radius.

To compensate for this, most beacons go through an initialization phase before appending it to the map and using the RO-SLAM algorithm. This initialization phase is usually performed over several iterations until the position converges to an acceptable degree. If the landmarks are not initialized properly the robot uncertainty increases and the filter diverges. There are various pre initializations processes that delay the localization of the robot by first initializing just the beacons. These methods are further divided into delayed and undelayed initializations.

2.1.5 Delayed/Off-line initializations

Delayed/off-line initializations are methods that attempt to localize the beacon prior to incorporating it into the active RO-SLAM algorithm. Some SLAM algorithms require an approximate location of each landmark before being incorporated into the on line/real time algorithm. Since beacons only provide data regarding range, the location of the beacon is not known upon first observation, and must be initialized. These processes which determines the initial estimated location of the beacon before being incorporated into the SLAM algorithm are known as delayed or off-line initializations. These initialization methods can be undesirable as they require the SLAM algorithm to run without corrective measurements until the first beacon is initialized. Presented in this section are the most prevalent methods for off-line beacon initialization.

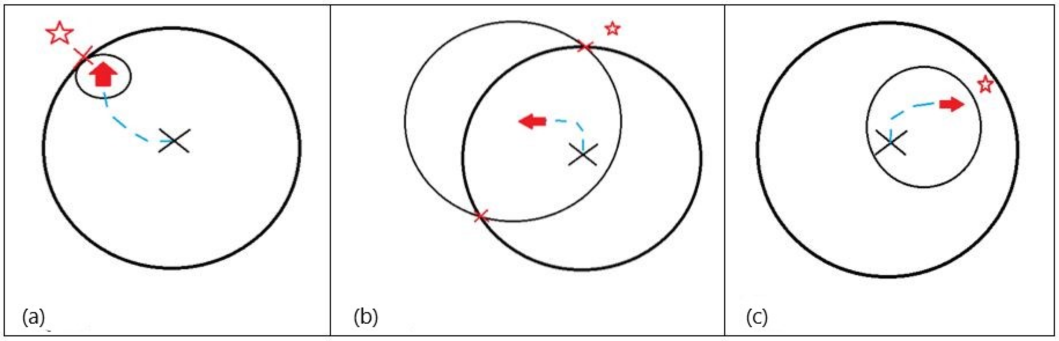


Figure 2.3: Multilateration used for Pre-initialization in RO-SLAM.

Trilateration

Trilateration refers to the process of using the distance to at least three devices at known locations in order to estimate the location of a fourth device. Methods generally use the geometry of circles and triangles in order to accomplish this. [6] mentions a basic trilateration technique that involves geometric intersection in both 2D and 3D state space. If there are more than three measurements its is known as multilateration. Trilateration uses the distance to each device with a known location referred to as a anchor node as well as radii for circles representing the possible locations of the device we wish to localize known as the mobile node. Rings are drawn at different points of the robots path as shown in Figure:2.3:a, where the radius of the rings is the distance measured to each beacon. We need atleast three circles around anchor nodes, ideally there should be a position at which each circle overlaps, resulting in the position of the mobile node. Figure 3.a shows basic trilateration scenario. The equations below describe how to calculate the trilaterated position [7]:

$$(X_1 - X_4)^2 + (Y_1 - Y_4)^2 = r_1^2 \tag{2.6}$$

Where (X_1, Y_1) and (X_4, Y_4) represent the Cartesian coordinates for anchor node 1 and mobile node 4, respectively, and r1 represents the estimated Euclidean distance between the two nodes. This equation can be rearranged to the following form:

$$(X_1 - X_4)^2 + (Y_1 - Y_4)^2 - r_1^2 = 0 \tag{2.7}$$

This equation can be repeated for each remaining anchor node and formed into the following system of equations:

$$\begin{bmatrix} (X_1 - X_4)^2 + (Y_1 - Y_4)^2 \\ (X_2 - X_4)^2 + (Y_2 - Y_4)^2 \\ (X_3 - X_4)^2 + (Y_3 - Y_4)^2 \end{bmatrix} - \begin{bmatrix} r_1^2 \\ r_2^2 \\ r_3^2 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \tag{2.8}$$

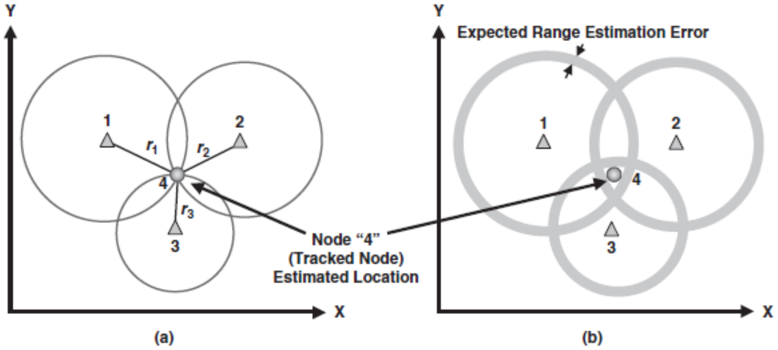


Figure 2.4: An ideal trilateration where we have a single point of intersection (a) and trilateration scenario with expected range estimation error (b) [7].

Where the solution to this equation yields the coordinates X_4, Y_4. However, due to inaccuracies in the equipment, resolution of the RSSI values, fading effects from multipath, or the empirically calculated path loss exponent, distance estimation to each anchor node is usually afflicted by some error. Figure: 2.4 shows a trilateration scenario with distance estimation error. Therefore, we need to examine the error terms instead.

$$abs\left(\begin{bmatrix}(X_1-X_4)^2+(Y_1-Y_4)^2\\(X_2-X_4)^2+(Y_2-Y_4)^2\\(X_3-X_4)^2+(Y_3-Y_4)^2\end{bmatrix}-\begin{bmatrix}r_1^2\\r_2^2\\r_3^2\end{bmatrix}\right)=\begin{bmatrix}e_1^2\\e_2^2\\e_3^2\end{bmatrix}=E \qquad (2.9)$$

Where abs is the absolute value function, e2n is the squared error term for the nth anchor node, and E is the error vector. In this form, the solution becomes finding the (X_4, Y_4) coordinates which minimizes the error vector. This is usually accomplished by utilizing a non-linear least squares method to find that value [8].

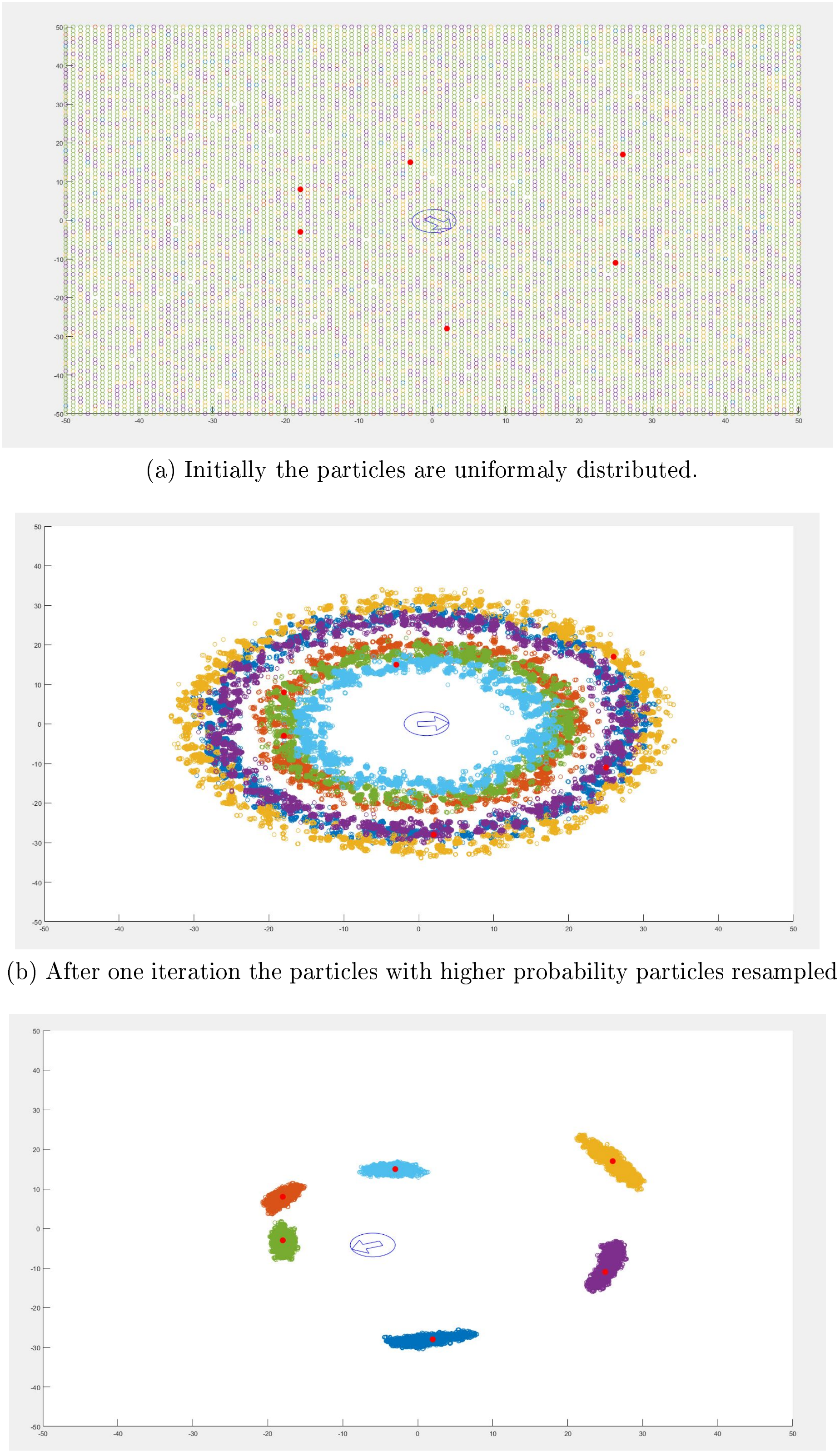
This method of initialization is highly ambiguous as the rings can have more than one point of intersecting as shown in Figure:2.3:b or no intersecting points as shown in Figure:2.3:c. Hence, it doesn't initialize a landmark until it get only one intersecting point in all 3 circles.

Particle Filtering

Particle filters take a probabilistic approach to estimating the initial location of the beacons as mentioned in [9]. Here, a particle refers to a possible hypothesis of position of the landmark. Initially the robot assumes global uncertainty through a set of particles that are distributed all over the map as shown in Figure:2.5(a). Hence the position of the landmarks are uncertain. As the robot moves around and measures sensor data it re-samples the particle set weighted according to the probability of the particle being the corect one based on sensor measurements as shown in Figure:2.5:b. After several iterations the particles that have a higher degree of probability remain and eventually converge to the location of the landmarks as shown in Figure:2.5:c.

The methods presented above tend to be the more common approaches, however, there are several off line algorithms that perform localization. A least squares approach is described in [10], a posed-based EKF (Extended Kalman Filter) method is mentioned in [11], and a moving horizon estimation is presented in [12]. These methods have also proven effective for the beacon types used in their respective applications.

These methods have higher probability of failure when there is a large amount of multi-path interference when used in noisy indoor environments (i.e the landmarks might get initialized to the multi-path signals). Hence even with the pre initialization processes, there will always be a chance where the robot obtains noisy range measurements leading to false localization and erroneous results. An example of which is shown in Figure 2.6. The particle filter represented by the blue particles converges to a false location of the landmark.



(a) Initially the particles are uniformaly distributed.

(b) After one iteration the particles with higher probability particles resampled

(c) Covergence of the particles to the landmark positions.

Figure 2.5: Particle Filter used for pre-intialization in SLAM (without mutipath).

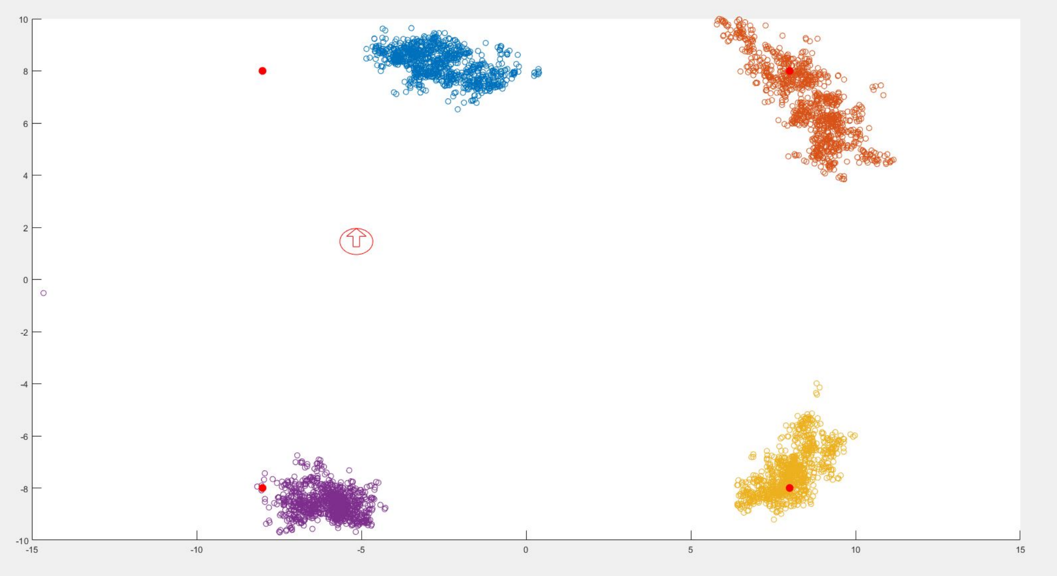


Figure 2.6: Example of a false intialization when it converges to a mutipath signal.

2.1.6 Undelayed/On-line initializations

With undelayed/on-line initialization it simultaneously initializes the landmarks and localizes itself and the beacons as it moves, as mentioned in [13]. A Gaussian Mixture Model is integrated into the EKF to create a multiple hypothesis state representation. This multiple hypothesis approach is used to solve the RO-SLAM problem and its use allows for the un-delayed initialization of landmarks, however a large number of hypothesis can lead to increased computational burden. [14] presents a method that models a Gaussian mixtures for beacon initialization in EKF. The paper proposes that under different circumstances the landmark could be represented using different parameters. This representation would lead to reduced computation cost for large scale range only SLAM as opposed to the high computation cost of multiple hypothesis landmark representations. The update function in a single EKF update step would have a computational complextity $O(d^3)$. If there were N hypothesis the computatinoal complexity become $O(N * d^3)$; Hence there are scalling limitations depending on compiutational complexity.

2.2 RSSI Models

Recently, considerable attention has been focused on wireless sensor networks (WSN) for autonomous robot navigation in flexible, indoor environments as they have the potential to be extremely cheap, easily deployed and distributed monitoring tools. The sensors range from acoustic ranging systems to radio frequency systems. Once these wireless beacons are deployed and initialized, the robot can navigate autonomously without the needing a map and by acquiring the information from the pre-set radio emission sensors deployed in the indoor environment. By measuring the distance from the sensor nodes, the robot locates itself and knows its pose. There are several ways of obtaining range measurements from signals. One method is measuring RSSI strength decay and the other is measuring time of flight. However, these signals cannot be predicted accurately due to attenuation in indoor environments from signal interference, absorption and reflection from objects within the environment. These factors cause unnecessary amplification or attenuation at the receiver known as multipath interference.

There are several mathematical models that derive the distance from RSSI but most do not take the multipath fading into account. There are models to predict multipath in long distance transmission such as Ricean Fading model or the free space path loss model. Indoor short range environments are more difficult to predict. Some of the mathematical models try modeling $RSSI$ variance by taking into account the odometry of the robot such as the Menegatti filter [15]. It uses the odometry information from the robot to generate a estimated RSSI multipath value that combines the estimated RSSI with the measured RSSI to a particular node. Hence when the robot moves a smaller distance the RSSI value cannot undergo an extreme change relative to the frame of the world map and vice versa. In addition to this there are several other factors that effect the accuracy of the robot such as odometry error.

$$RSSI_e = 10n \log d \pm u + A \tag{2.10}$$

$$RSSI_p = \frac{RSSI_e + RSS_m}{2} \tag{2.11}$$

$$d_f = 10^{\frac{RSSI_p - A}{10n}} \tag{2.12}$$

Where $RSSI_e$ is the expected $RSSI$ strength, $RSSI_m$ is the measured signal strength at a given distance d or minus the displacement, u. The filtered distance, d_f, is the filtered distance by rearranging the log distance model and applying the predicted RSSI value. However this model has been heavily criticized for relying heavily on odometry data.

There are several solutions available to solve the multi-path issue that either by wireless modulation of the hardware [16] or with the use of directional antenna arrays [17]. But these cause an increase in the implementation costs due to the hardware modifications. These implementations also involve of non standard communication protocols and hence cannot be successfully used in existing networks.

CHAPTER 3: Particle Filter for RO SLAM

The EKF assumes the noise to be gaussian and hence it can represent the state of the variable with a mean and a variance. But as it encouners mutipath in indoor enviornments the data becomes highly non gaussian and hence the mean and variance estimated by EKF fails to represent the distribtuion. The particle filter represents the state variable in random samples drawn from its posterior, allowing it to represent data in all dstributions.

Particle filters are a type of genetic algorithms which estimates the state of a random variable. They are based on the same core principle of Bayesian estimation. The fact that every model no mater how complex fails to completely capture even the most simplistic environment has led to specific tricks and techniques for the success of particle filter, especially in robotics. Particles filters are a category of monte carlo algorithms that are used to estimate states in partially observable Markov chains as shown in [18]. Early successes of particle filters can be seen in area of robot localization where a robot has to localize itself in a known environment using measurements from sensors. They have also been successful in solving global localization problems [19] or the kidnapped robot problem [20], where a robot has to recover its state from global uncertainty.

A particle here represents the hypothesis of a particular state variable. The particle filter used for global localization is graphically depicted in Figure 3.1. It describes the process where a robot localities itself inside a known map. The first image, (a), shows the map of an interior building filled with uniformly scattered black dots. The dots represent a particle or hypothesis of the state, or position, of the robot. Over time, as the robot moves, The particles "condense" or converge over iterations as the

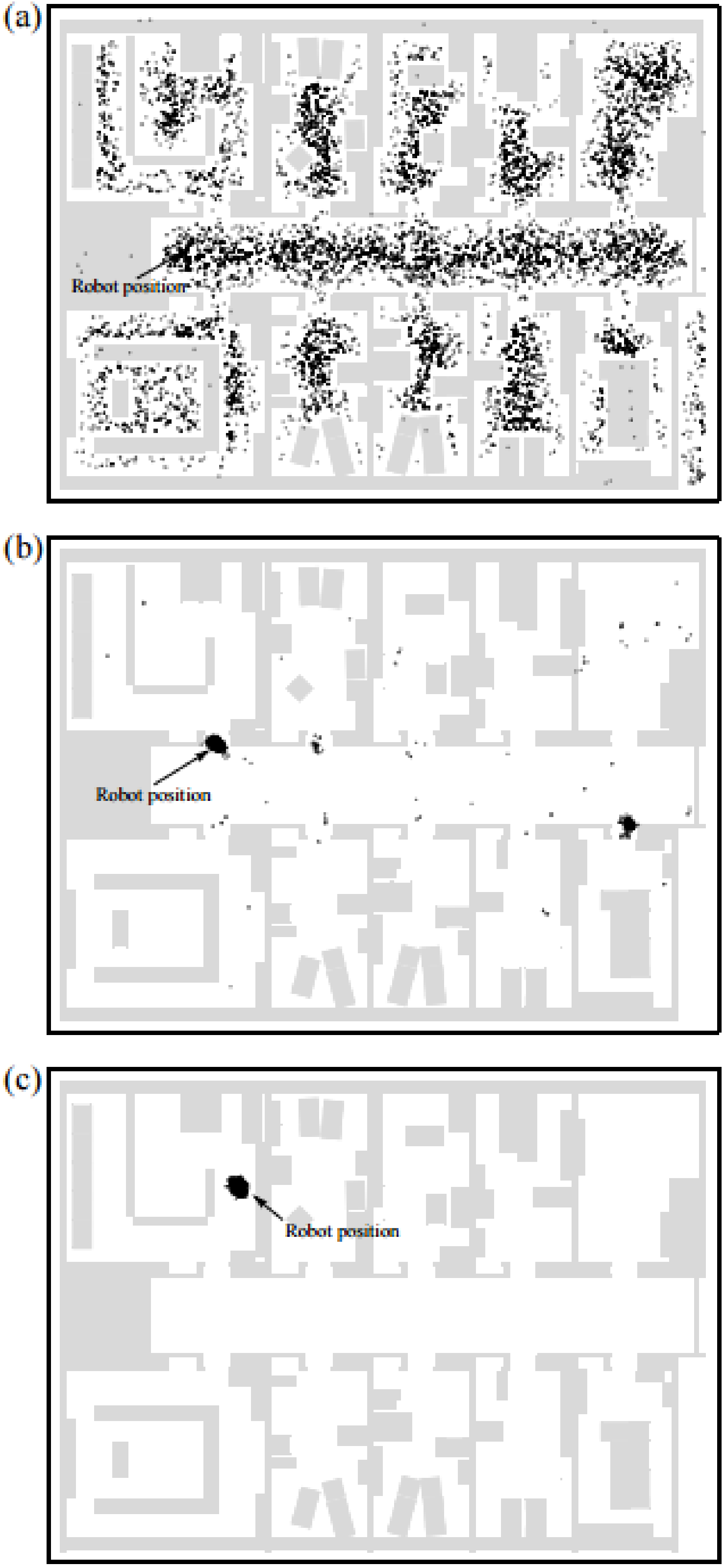


Figure 3.1: Particle filter for mobile robot localization [3].

erroneous particles predicted measurements generate a low importance weight when compared to the sensor measurements, hence, they are resampled less and less each iteration. In Figure 3.1 image b, if there are two possible locations of the robot as indicated by 2 clusters. The symmetrical nature of the hallway introduces two nearly equally likely position for the robot to exist. Once the robot moves to a structurally unique position on the map the the filter converges into one cluster as shown in Figure 3.1:c.

Particle filters have been successfully able to solve applications in the robotics domain for more than one reason. They can be applied to any model that is probabilistic and can be represented as a Markov chain. Secondly particle filters do not require a fixed computational time rather their accuracy increases with increase in computational resources as mentioned in [21] and [22].

They also overcome several drawbacks that are faced by traditional SLAM using EKF. It can process negative information unlike the EKF. They provide better data association outcomes than standard EKF and handle non-gaussian noise. It has also been at the core higher dimensional computational problems. In addition, in the context of RO SLAM using beacons it can minimize the error that can be caused due to false localization that occur in RO SLAM using EKF due to multipath.

Algorithm 4 The Particle Filter (X_{t-1}, z_t)[3]

```
$\bar{X}_t = X_t = \emptyset$
for $m = 1$ to $M$ do
    sample $x_t^{[m]}$ $p(x_t|u_t, x_{t-1}^{[m]})$
    $w_t^{[m]} = p(z_t|x_t^{[m]})$
    $\bar{X}_t = \bar{X}_t + < X_t^{[m]}, w_t^{[m]}$
for $m = 1$ to $M$ do
    draw $i$ with probability $\approx w_t^{[i]}$
    add $x_t^{[i]}$ to $X_t$

return $X_t$
```

A particle filter algorithm is mentioned in [3] All the particles are contained within the set of data X_t. Initially algorithm, X_t and $\bar{X}_t$ are initialized as empty sets. These are later populated as particles are drawn from the previous set, X_{t-1}. Depending on the number of particles initially the filter assumes global uncertainty by either by uniformly distributing the particles or according to a certain distribution over the entire map. The more particles used the faster the filter converges and more accurately it represents the variables. The prediction is performed in the sample process, where x_t is calculated based on x_{t-1} and control input to the robot u_t. The Bayesian posterior is also calculated $p(x_t|u_t, x_{t-1}^{[m]})$ for all the particles. Once it receives a measurement, z_t, it calculates a importance weight for each particles. The importance weight is assigned from the distribution $p(z_t|x_t)$, which uses the measurement vector and the calculated posterior for x_t. The set $\bar{X}_t$ is then populated with M hypothesis, x_t, and associated importance weights, w_t. The next step re samples M particles from $\bar{X}_t$ according to the importance weights assigned.

The particle filters scale exponentially with the amount of variables in state space. However, since indoor robot localization using range only measurements involve low dimensional state variables a standard particle filteris an appropriate solution. Also since the particle filter eventually converges to actual localization of the robot and the landmarks, it avoids false initializations of landmarks due to mutipath signals as seen in RO SLAM using EKF. As the robot traverses its environment and receives a line of sight signal from a beacon, the particle filter would eventually converge to the true location

3.1 Implementation

.

The implementation uses the same core algorithm mention in Algorithm: 4. The particle filter for the robot uses importance weights derived from the measurement obtained by the robot and the actual distance between each particle representing the robot to the mean of the particles representing the landmark. The process is repeated for the landmarks but by taking the weighted average of the particles representing the robot and each particle representing the landmarks. These importance weights are used to resample the particles for the robot and landmark respectively. Hence a particle with a higher importance weight is resampled more compared to one with a smaller importance weight. As we can see the particles for the state variables eventually converge to the positions of the robot and the landmark as shown in 3.2. The red dots and red circle are the actual landmark and robot positions. The blue, purple, yellow, orange particles represent each landmark. The green set of particles represent the robot particle filter and the black circle represents the average of the particles.

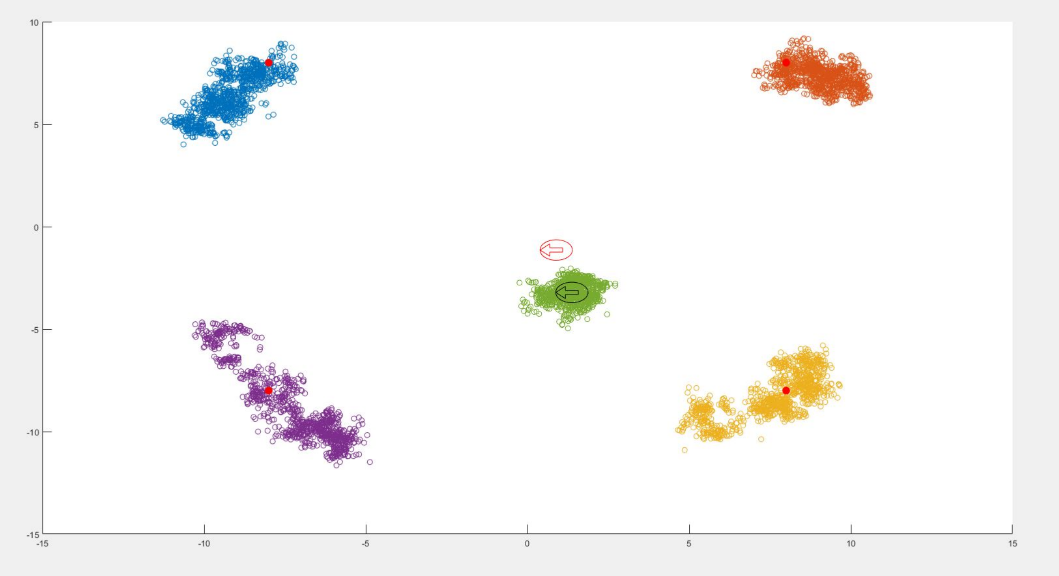


Figure 3.2: Implementation of Particle Filter used to perform SLAM.

3.1.1 Particle Filter algorithm

The algorithm used for implementation is shown in algorithm: 5. Particle sets X_1 to X_N represent the N landmarks while the set $X_N + 1$ represent the robot. Lines 1-5 in algorithm:5 refer to the initial generation of the particles uniformly across the entire region of interest for the landmarks. It is assumed that the initial location of the robot is known, as the world is generated from the initial position of the robot

Algorithm 5 Implementation

```
Landmarks:
for $n = 1$ to number of landmarks do
    for $t = 1$ to number of particles do
        $X_{(t)(n)}$ = particles uniformly distributed
        $\bar{X}_{(t)(n)} = \emptyset$
 Robot:
for $t = 1$ to number of particles do
    $X_{(t)(N+1)}$ = particles at the origin
Measurement Step:
Landmarks:
for $n = 1$ to number of landmarks do
    for $t = 1$ to number of particles do
        $w_t^{[n]} = p(z_t | X_t^{[n]}, mean(X_t^{[N+1]}))$
        $\bar{X}_t^{[n]} = \bar{X}_t^{[n]} + < X_t^{[n]}, w_t^{[n]}$
    for $t = 1$ to $T$ do
        draw $i$ with probability $\approx w_t^{[i]}$
        add $x_t^{[i]}$ to $X_t^{[n]}$

return $X_t^{[1:N]}$
Robot:
Prediction Step :
for $t = 1$ to number of particles do
    $x_t^{[N+1]} \approx p(x_t | u_t, x_{t-1}^{[N+1]})$
if (landmarks converge) then
    for $n = 1$ to number of landmarks do
        for $t = 1$ to number of particles do
            $w_t^{[n]} = p(z_t | X_t^{[N+1]}, mean(X_t^{[n]}))$
            $\bar{X}_t^{[N+1]} = \bar{X}_t^{[N+1]} + < X_t^{[N+1]}, w_t^{[n]}$
        for $t = 1$ to $T$ do
            draw $i$ with probability $\approx w_t^{[i]}$
            add $x_t^{[i]}$ to $X_t^{[N+1]}$

    return $X_t^{[N+1]}$
```

(0,0). The particles for the robot are generated at the origin. In the measurement phase for the landmarks importance weights are computed for each particle based on the error between distance from the received RSSI measurement and the distance from each particle to the mean position of the particles $X[N+1]$ representing the robot. Once these importance weights are computed they are normalized and the particles representing the landmarks are resampled based on the weights represented by $w_t^{[n]}$ with the particles with higher importance weights being sampled more than the rest. Now the particles representing the robot undergo a prediction step based on the motion model used and the control input given. If the particles for the landmarks converge then previous process for generating weights is repeated for the particles representing the robot but by taking the mean position of the particles representing the landmarks in lines 23 to 29. Then particles representing the robot are resampled according to the weights. If the landmarks have not converged the algorithm is repeated from line 6. A high level representation of the algorithm is depicted in Figure:3.3.

3.2 Simulation

MATLAB was used to simulate realistic RSSI signals in different environments using a Markov chain training method mentioned in [1]. A XBee IEEE 802.15.4 transmitter and receiver was used to record RSSI strengths in different environments and in increasing increments of 0.25m positions at 2.4GHz. The environments represented varying levels of multipath propagation. This data was used to train a markov chain that provided us with realistic RSSI signals in our simulations. As shown in Figure:3.4 from [1] the simulated RSSI signal closely matched the actual measured values collected in the environment. The radiation patterns generated by the model Figure:3.5. This method of using actual data tracks multipath very accurately and overcomes the need of accurately modeling different test environments.

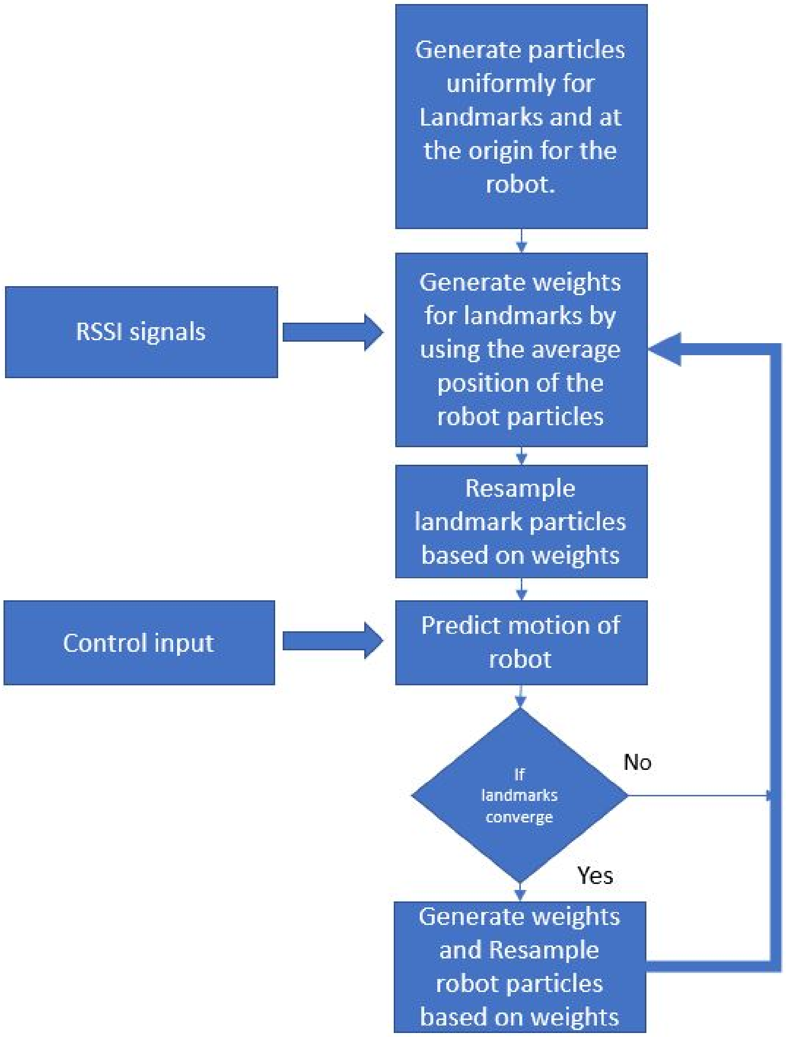

Generate particles uniformly for Landmarks and at the origin for the robot.
RSSI signals
Generate weights for landmarks by using the average position of the robot particles
Resample landmark particles based on weights
Control input
Predict motion of robot
If landmarks converge
No
Yes
Generate weights and Resample robot particles based on weights

Figure 3.3: High Level Representation.

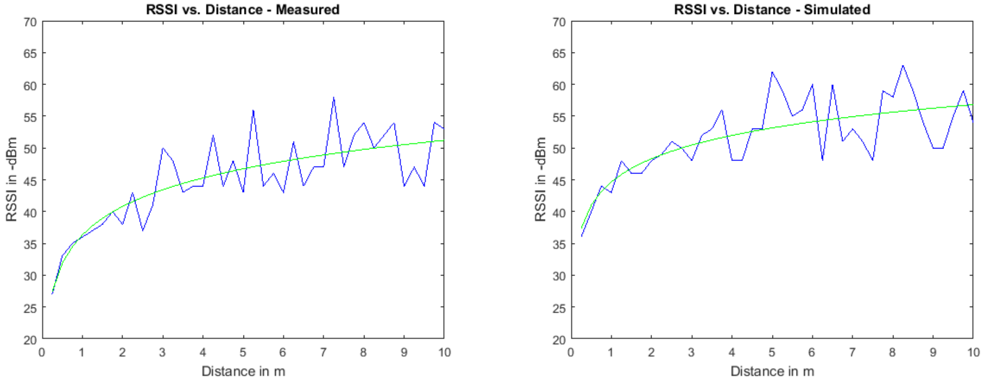


Figure 3.4: Measured and simulated RSSI data over distance[1].

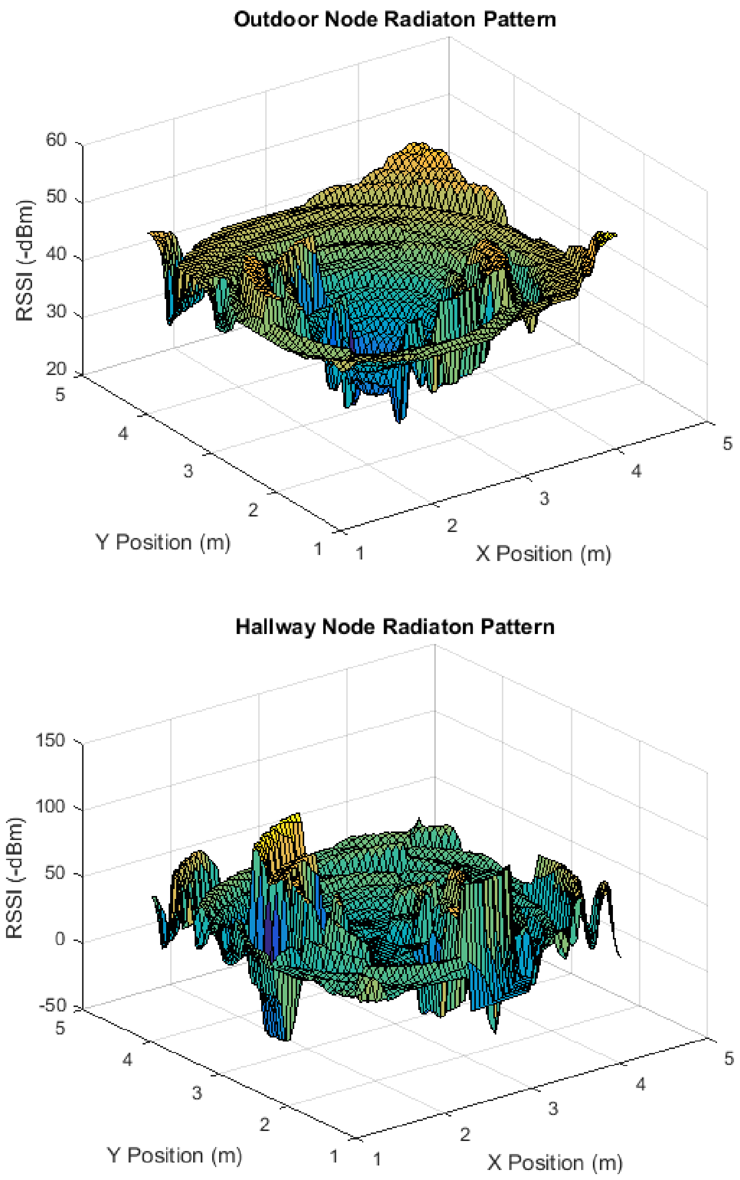


Figure 3.5: Simulated radiation patterns from Markov chains trained on various environment data [2]

CHAPTER 4: Results

The simulation is run for the data sets representing a lab, a hallway, an open room and an outdoor environment. The test cases for each environment are further divided into landmarks initialized at true locations and falsely initialized landmarks. Both cases use the same set of RSSI signals generated by our simulator. The first graph in each case represents the histogram of the error in distance derived from the RSSI value from the radio beacons. This is obtained through the difference between the actual robot and landmark to the distance derived from the signal generated by our simulator. The following graphs represents the accumulated robot position error for the different methods, landmark localization error, the actual path followed by ground truth, dead reckoning, RO SLAM and Particle filter. In all cases the robot is assumed to start at the same location. In each case the robot is driven with constant linear velocity of 0.1 m/sec in a 'S' Shaped track and a angular velocity of 30 degrees during the turns. It is assumed the landmarks have already been initialized.

4.0.1 Case: Outdoors

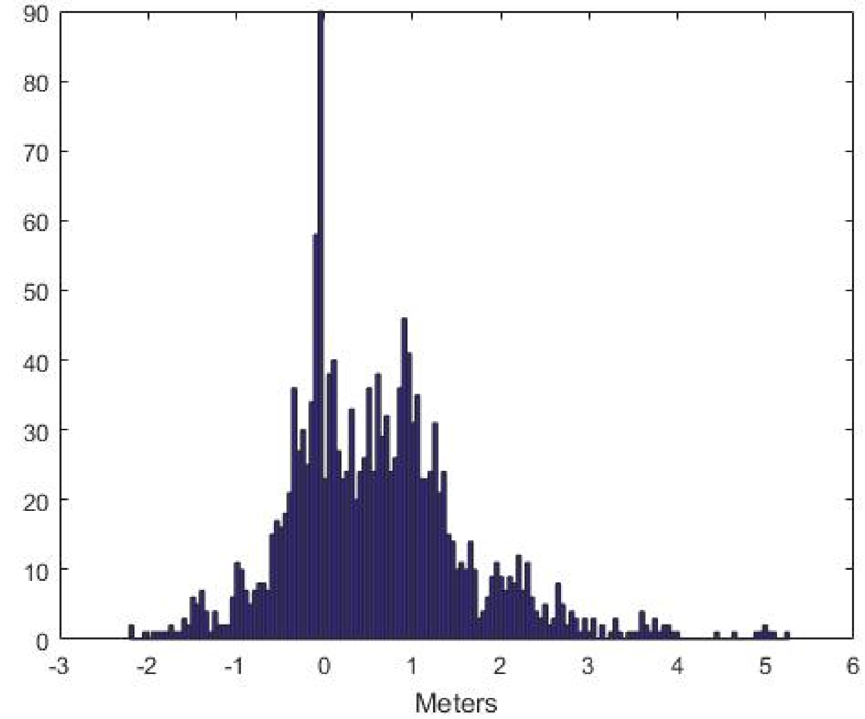


Figure 4.1: The histogram of the error in RSSI signals (in meters) generated by the simulator by using the data set representing outdoor environments at 2.4 GHz.

Since the histogram of the beacon noise has a variance of 2.25 meters, it is assigned as the variance of the measurement noise in RO EKF. Also a Gaussian zero mean odometry noise with a variance of 0.1 meters linear velocity is introduced, a zero mean gaussian noise of 10 degrees is used for the angular velocity noise. It uses a particle filter that maintains 1000 particles for each landmark and also the robot.

True Landmark Initialization

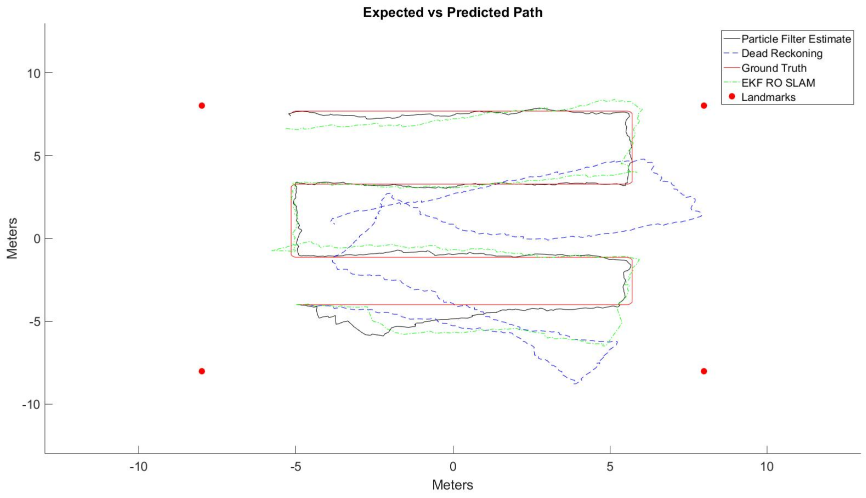


Figure 4.2: The error in path followed in outdoor environments.

In this case we assume the landmarks are intialized approximately to the true locations. The landmarks are represented by the red dots shown in Figure: 4.2. The black trace indicates the mean position of the particle filter representing the robot, red line represents the true robot position, the dashed blue line represents the dead reckoned position and the dashed green line represents the position given by RO-EKF.

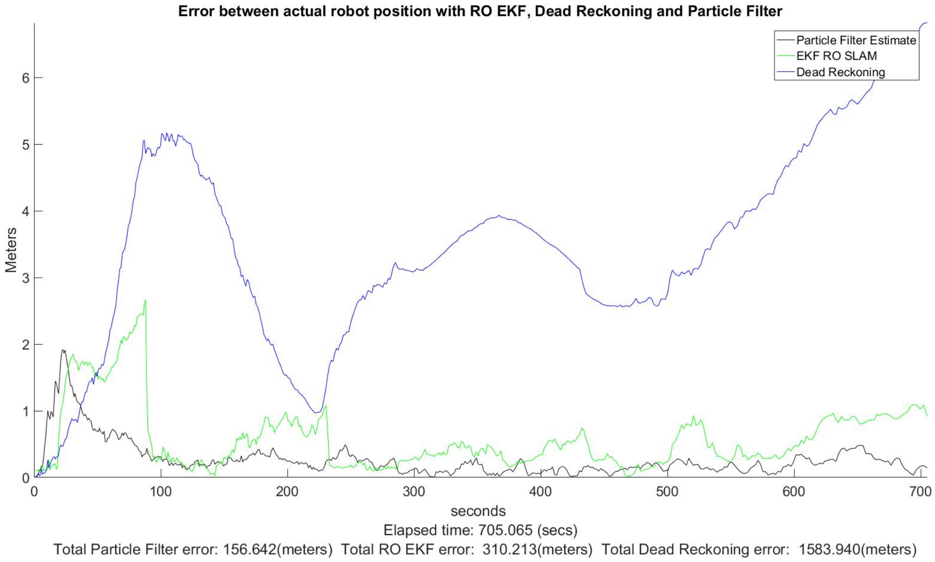


Figure 4.3: The error in the robots position accumulated over the entire path in an outdoor environment.

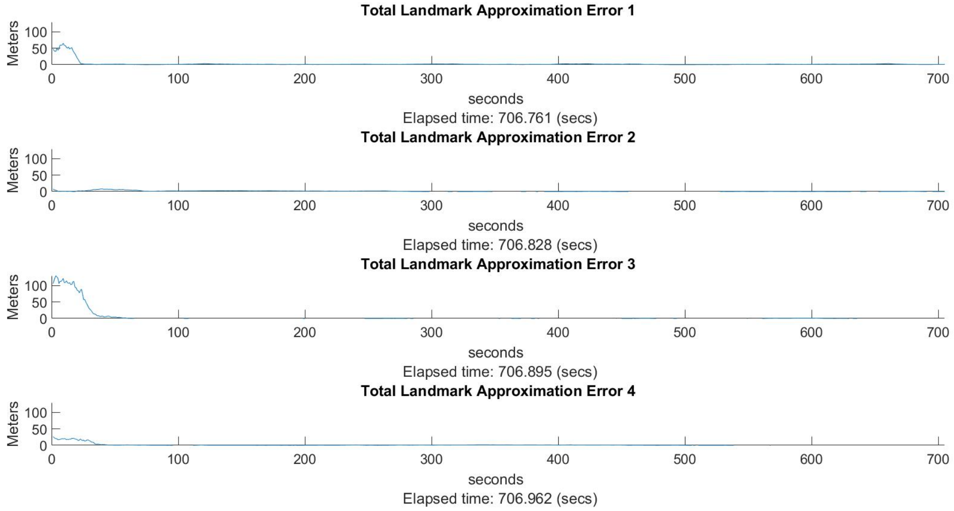


Figure 4.4: The error in the landmark localization in an outdoor environment.

False Landmark Initialization

It is clear from the Figure:4.1 that the noise is highly non gaussian, and we can see burst of multipath signals even at a distance of 3-4 meters from the landmark position. Now we initialize two landmarks to false locations and measure the accumulated error over the entire path.

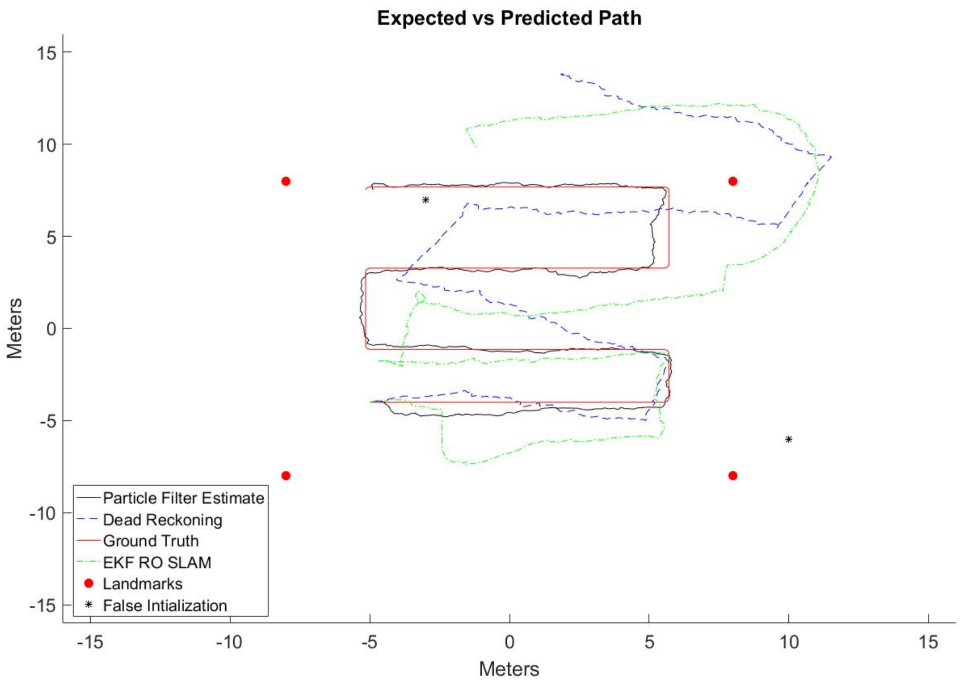


Figure 4.5: The error in path followed in an outdoor environment with false initialization.

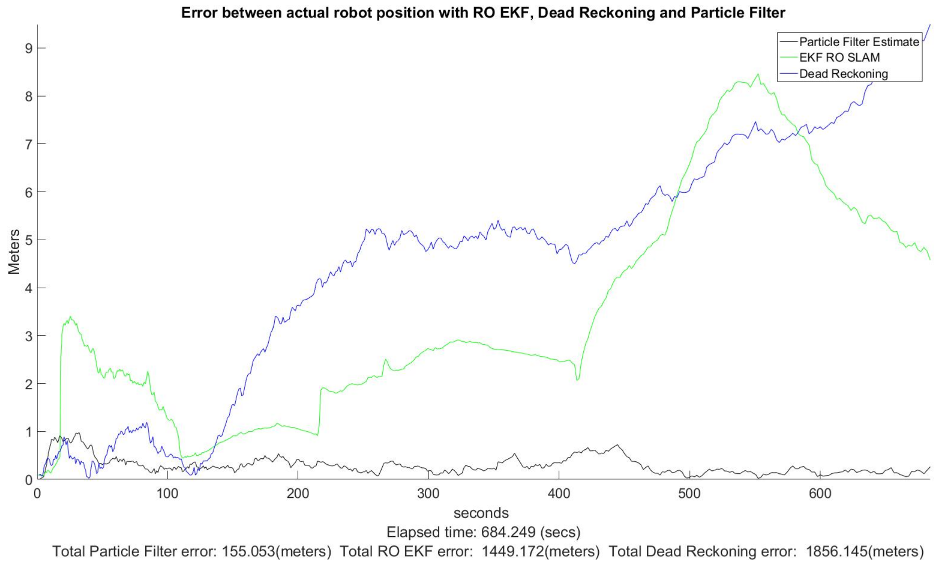


Figure 4.6: The error in the robots position accumulated over the entire path in outdoor environments.

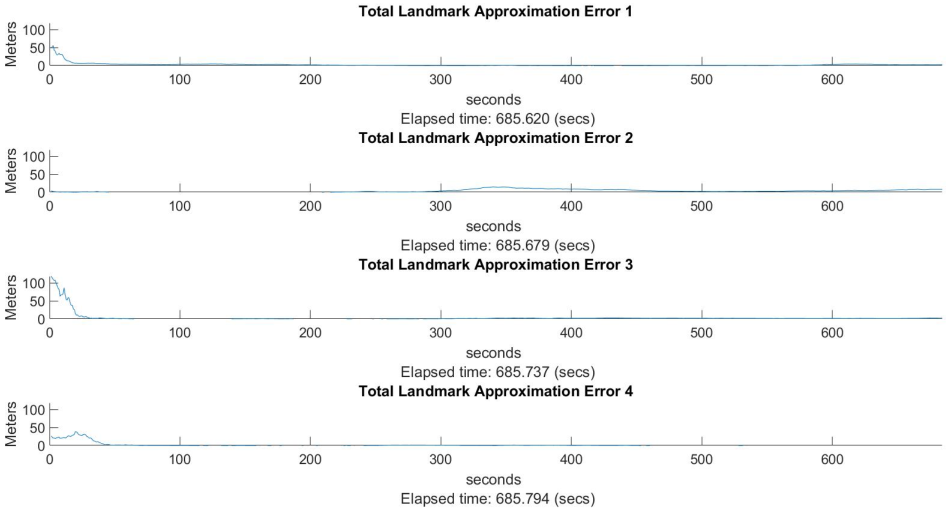


Figure 4.7: The error in the landmark localization in an outdoor environment.

From Figure: 4.7 we can see that even if the particle filter converges to a false location due to mutipath, it would eventually converge to the true locations of the landmarks as the robots moves across the map.

4.0.2 Case: Hallway

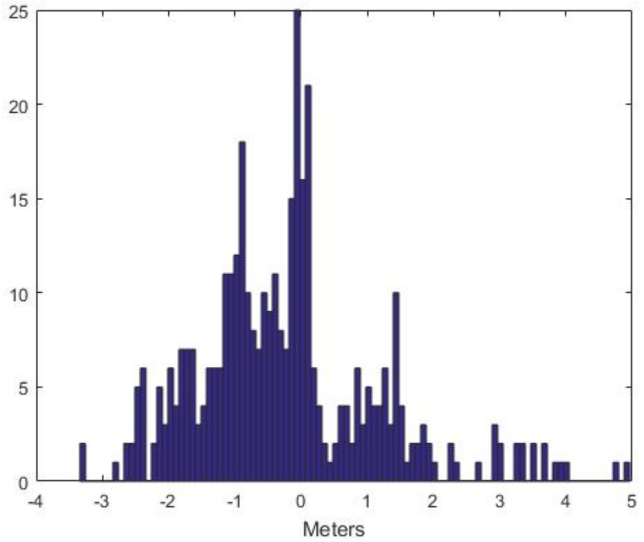


Figure 4.8: The histogram of the error in RSSI signals (in meters) generated by the simulator by using the data set representing hallway environments at 2.4GHz.

In this case we have used a measurement noise with a variance of 3.25 in our RO EKF and have repeat the process mentioned in the previous test case.

True Landmark Initialization

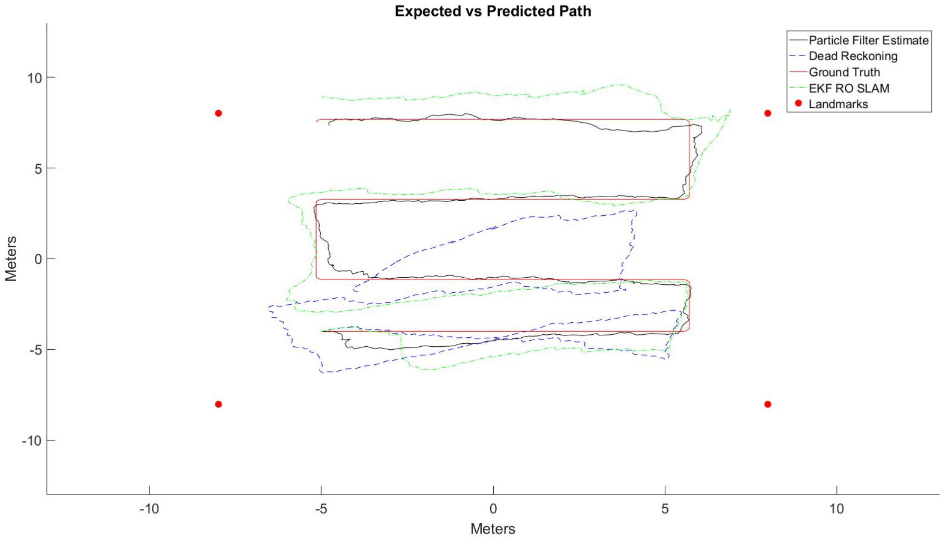


Figure 4.9: The error in path followed in Hallway environments.

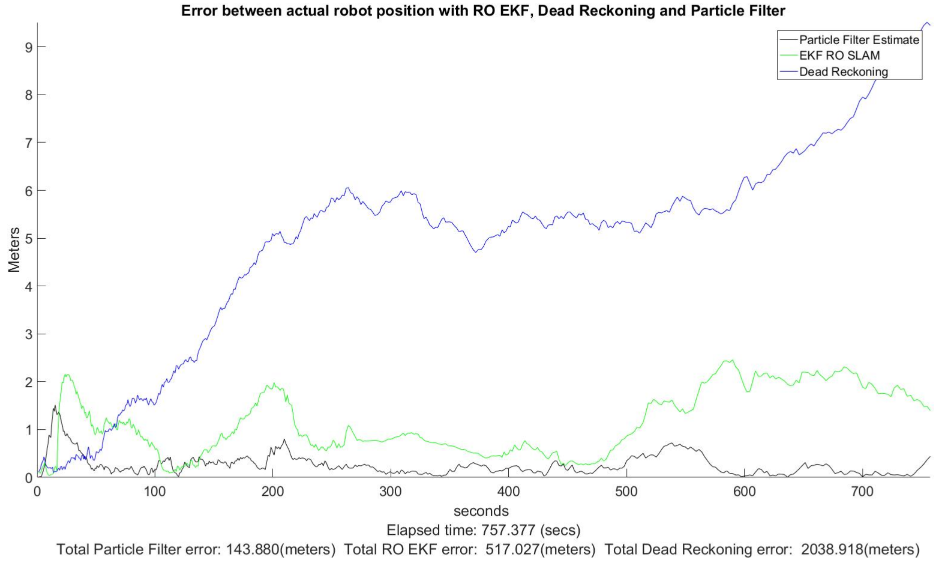


Figure 4.10: The error in the robots position accumulated over the entire path in a Hallway environment.

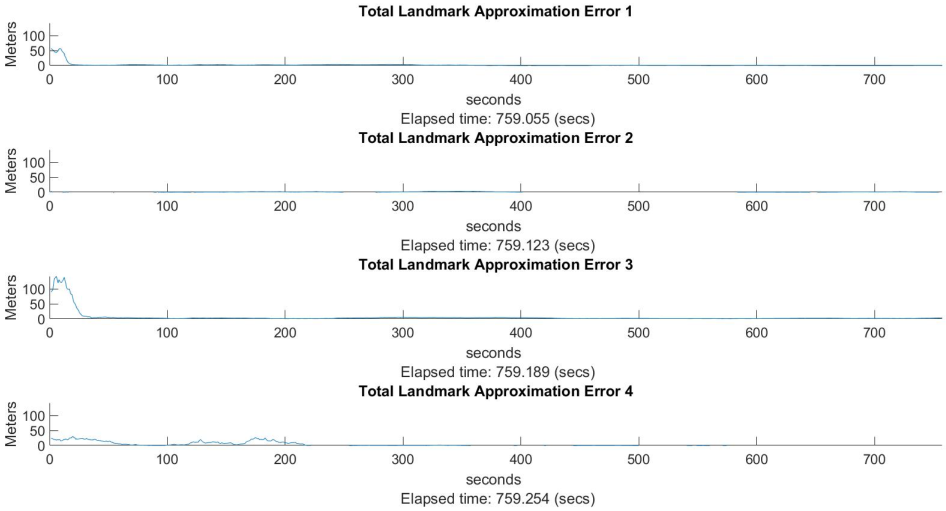


Figure 4.11: The error in the landmark localization in a Hallway environment.

False Landmark Initialization

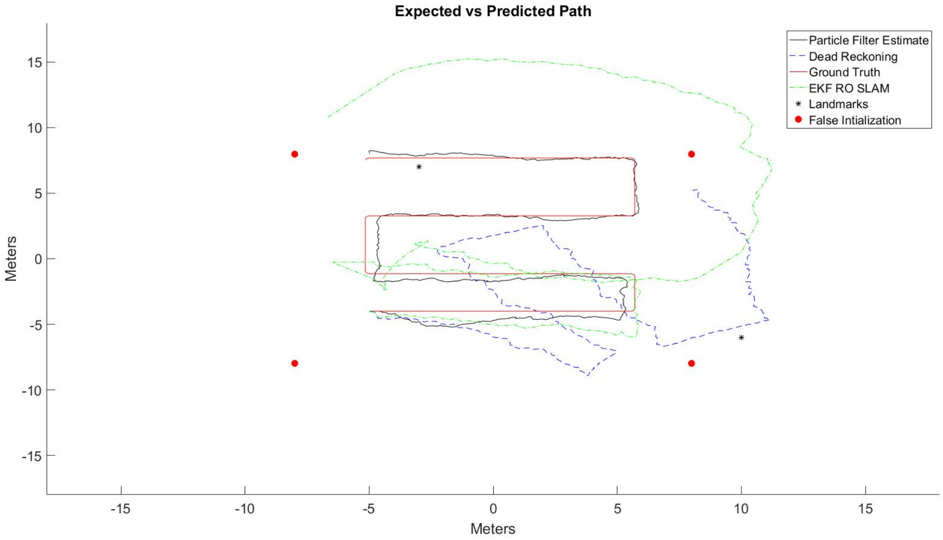


Figure 4.12: The error in the path followed in a Hallway environment with false initializations.

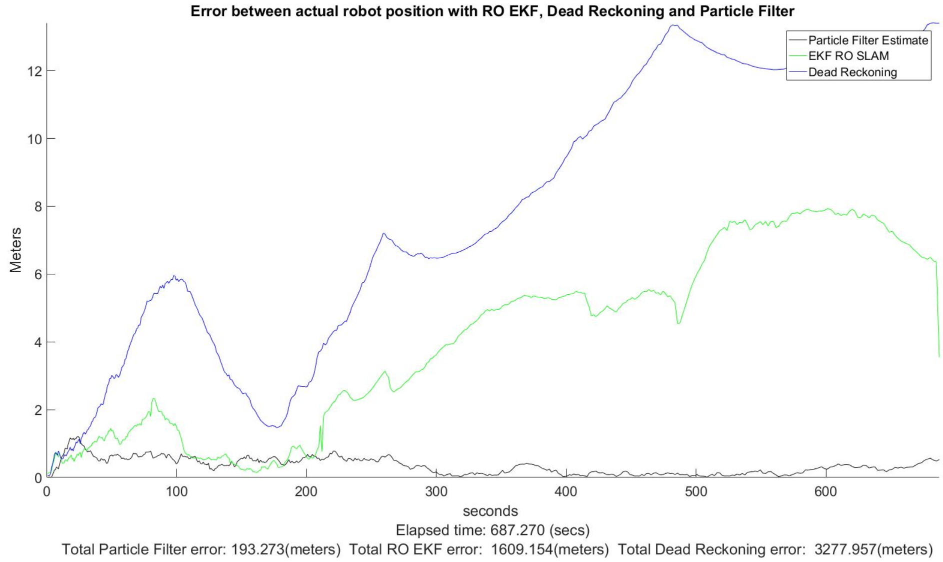


Figure 4.13: The error in the robots position in a Hallway environment with false initialization.

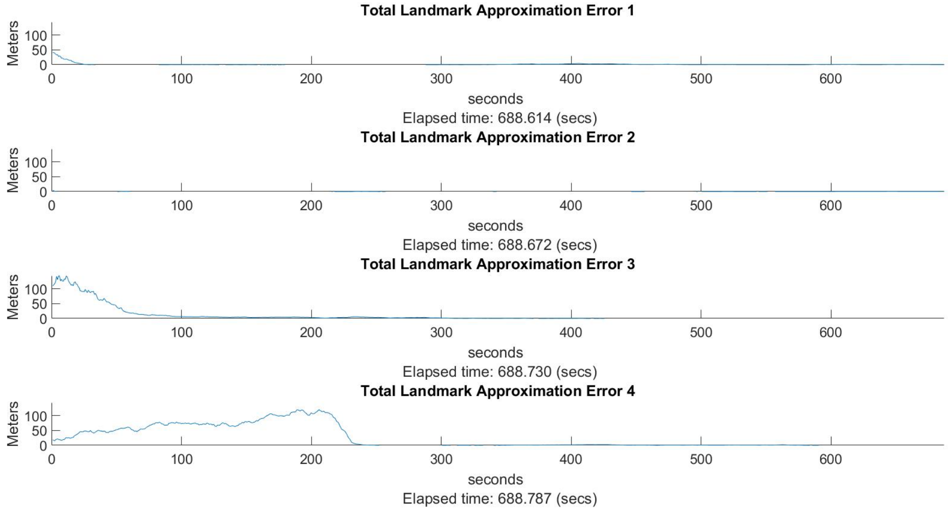


Figure 4.14: The error in the landmark localization in a Hallway environment with false initialization.

4.0.3 Case: Lab environment

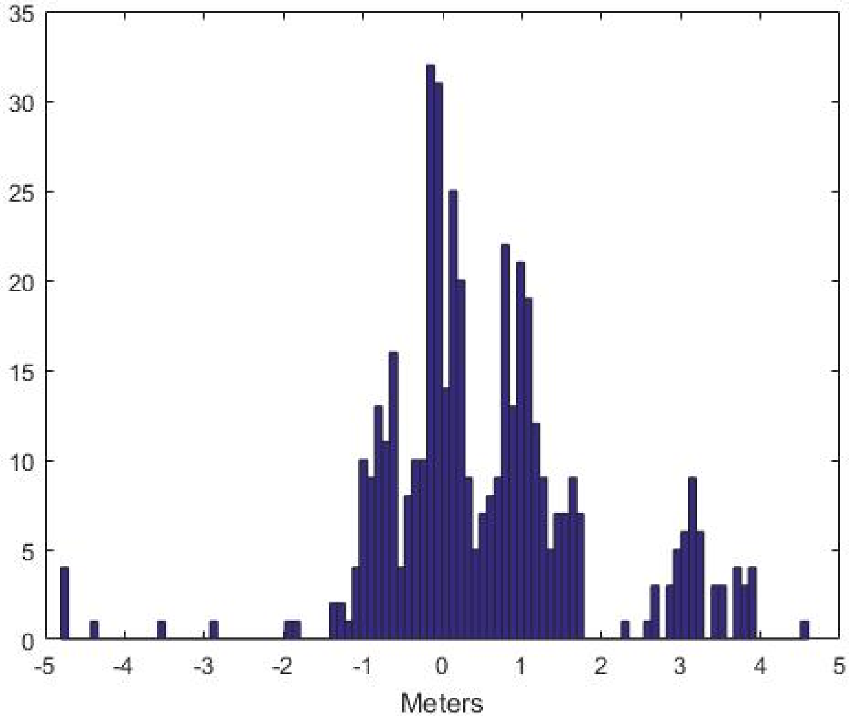


Figure 4.15: The histogram of the error in RSSI signals (in meters) generated by the simulator by using the data set representing a lab at 2.4GHz.

In this case we have used a measurement noise having variance of 2 meters variance in our RO EKF and have repeated the process mentioned in the previous test case.

True Landmark Initialization

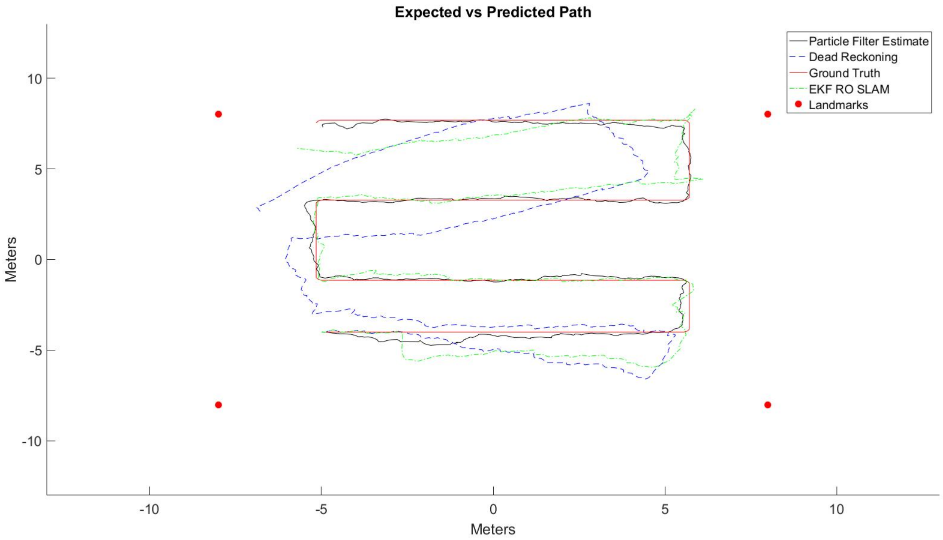


Figure 4.16: The error in path followed in a Lab environment.

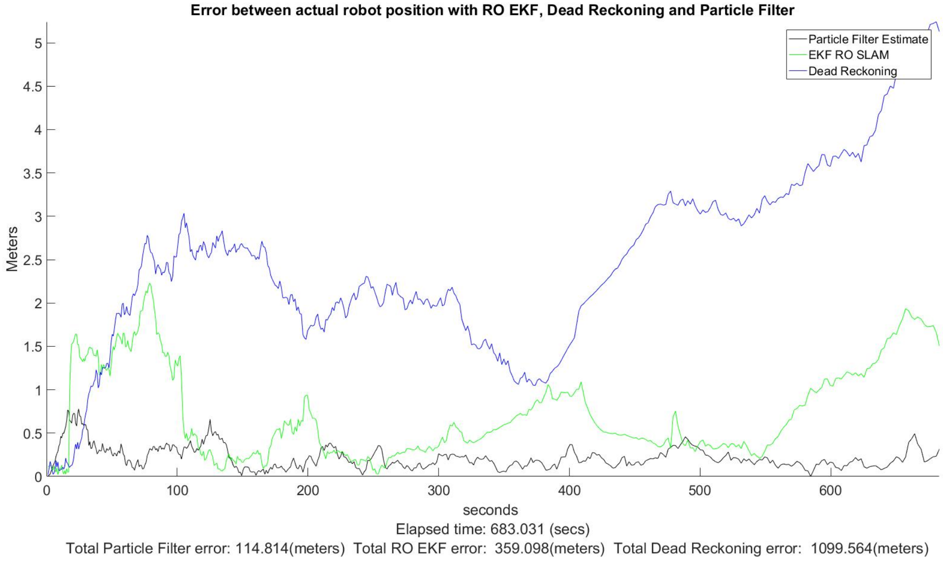


Figure 4.17: The error in the robots position accumulated in a Lab environment over time.

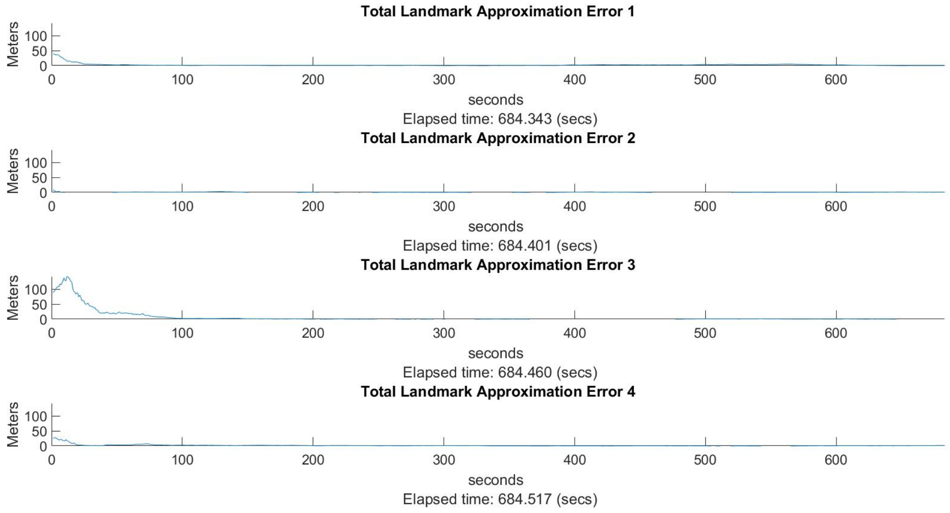


Figure 4.18: The error in the landmark localization in a Lab environment over time.

False Landmark Initialization

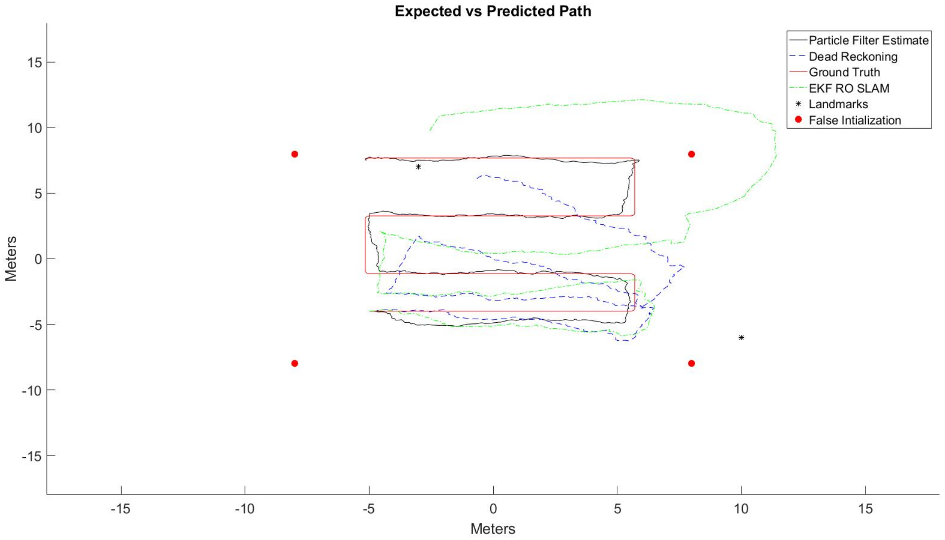


Figure 4.19: The error in path followed in Lab environments with false initialization.

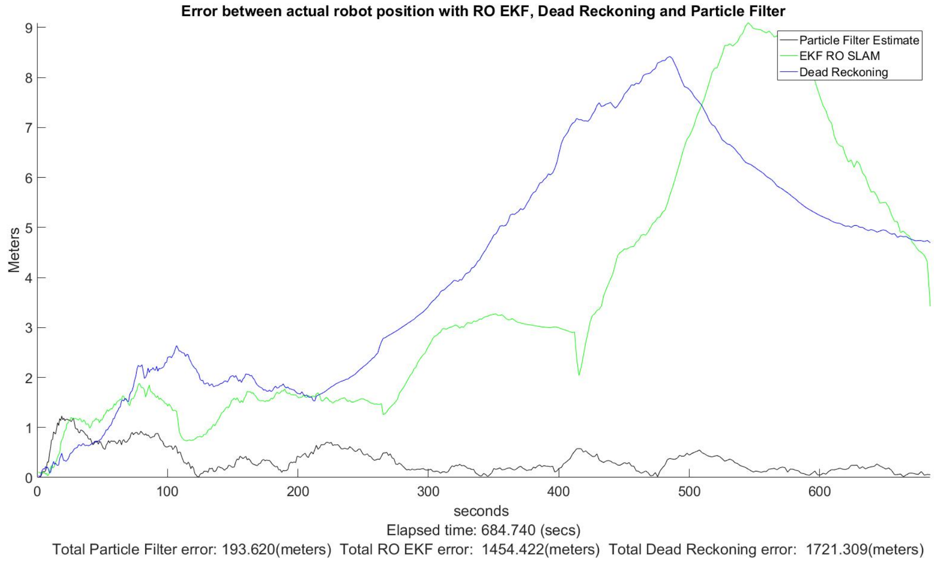


Figure 4.20: The error in the robots position in a Lab environment with false initialization over time at 2.4GHz.

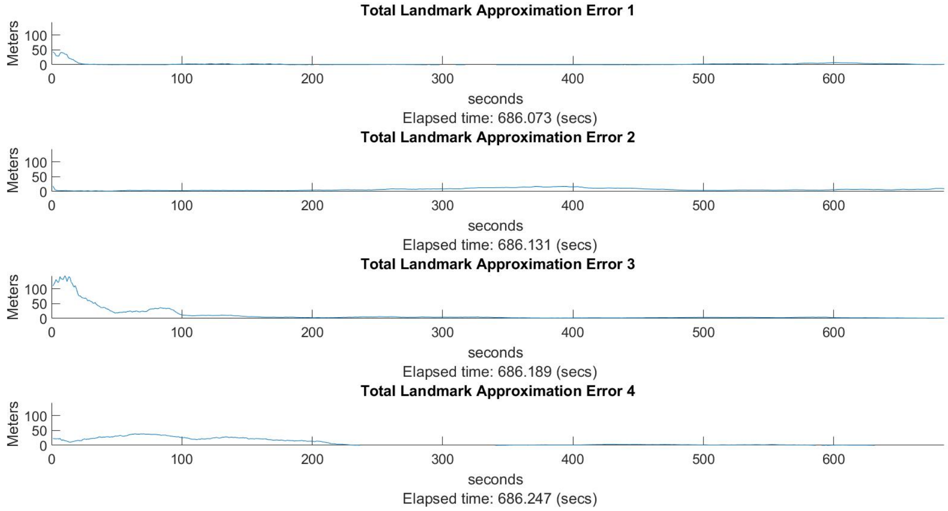


Figure 4.21: The error in the landmark localization in a lab environment over time.

4.0.4 Case: Large Open Room

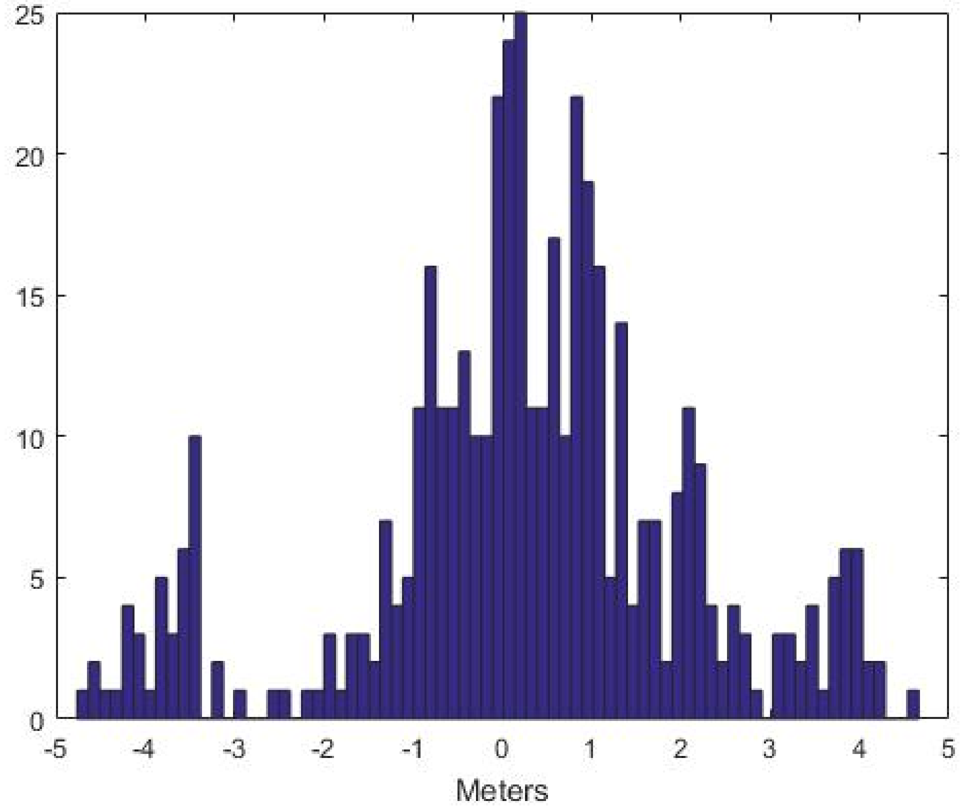


Figure 4.22: The histogram of the error in distance derived from RSSI (in meters) in a large open room environment at 2.4GHz.

In this case we have used a measurement noise of 3.5 meters variance in our RO EKF and have repeated the process mentioned in the former test cases.

True Landmark Initialization

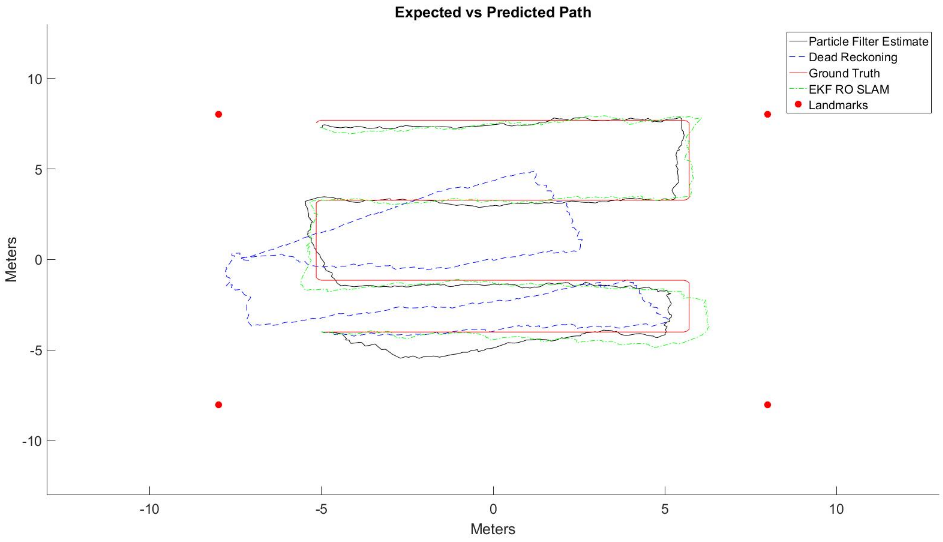


Figure 4.23: The error in path followed in a large open room environment.

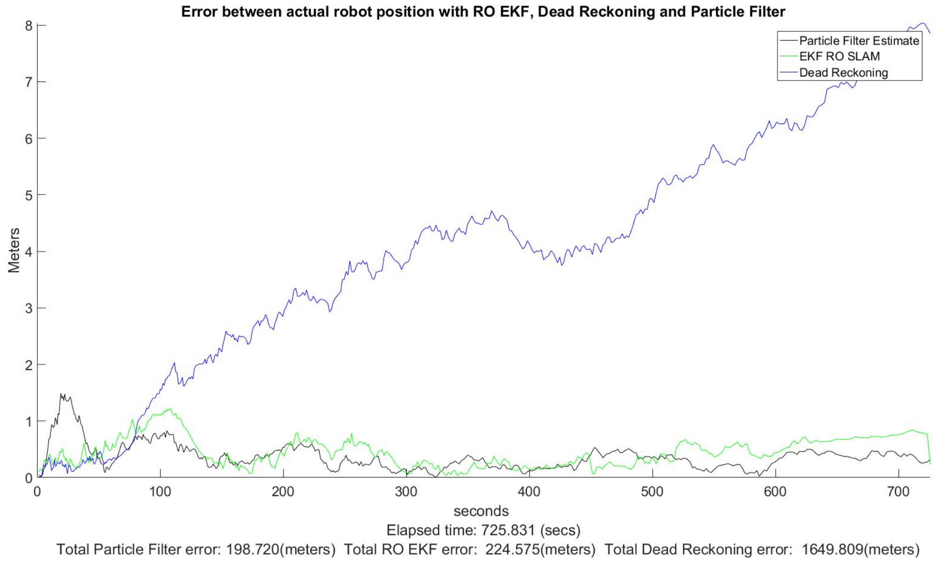


Figure 4.24: The error in the robots position accumulated over the path in a large open room environment over time.

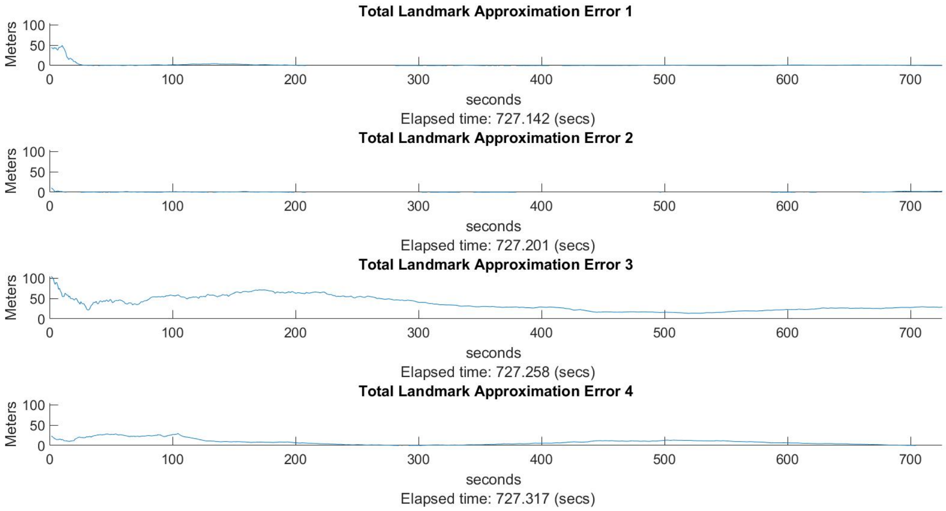


Figure 4.25: The error in the landmark localization in a large open room environments over time.

False Landmark Initialization

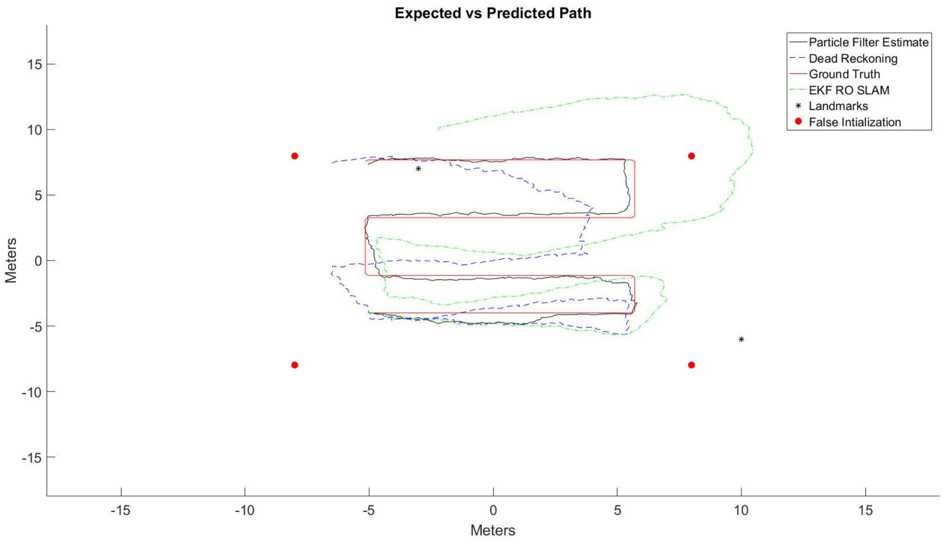


Figure 4.26: The error in path followed in a large open room environments with false initialization.

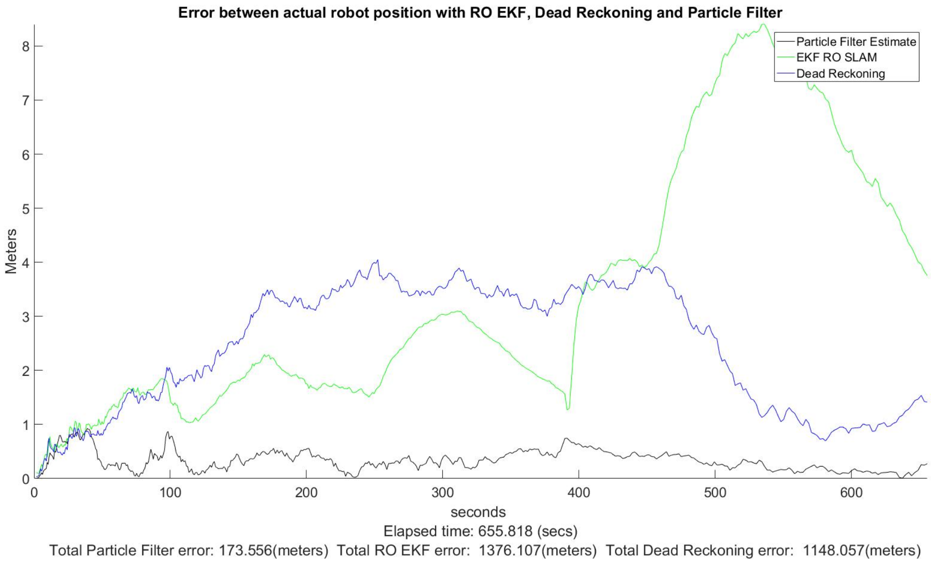


Figure 4.27: The error in the robots position in a large open room environment with false initialization over time.

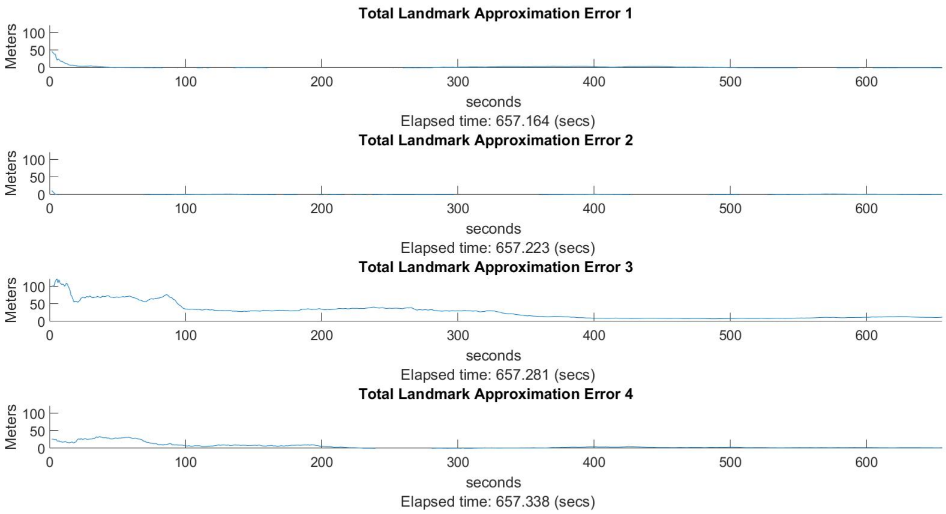


Figure 4.28: The error in the landmark localization in a large open room environment with false initialization over time.

4.1 Comparision of environments

As shown in Table: 4.1 the EKF becomes comparable to the particle filter when the landmarks are correctly initialized depending on the amount of mutipath it comes across as it traverses through the path. But when falsely initialized it becomes comparable to dead reckoning.

Table 4.1: Comparison of accumulated robot error over the entire path using particle filter, dead reckoning and EKF methods in different enviornments collected at 2.4GHz.

RO SLAM	Particle Filter (meters)	Dead Reckoning (meters)	EKF (meters)
Outdoor with true initialization	156.642	1583.940	310.213
Outdoor with false initialization	155.053	1856.145	1449.172
Lab with true initialization	114.814	1099.564	359.098
Lab with false initialization	193.620	1721.309	1454.422
Large open room with true initialization	198.720	1649.809	224.575
Large open room with false initialization	173.566	1148.057	1376.107
Hallways with true initialization	143.880	2038.918	517.027
Hallways with false initialization	193.273	3277.957	1609.154

4.2 Analysis on the size of the particle set

Below we have presented the differences in the efficiency when we use particles sets of different sizes. We ran the particle filter, dead reckoning and Range Only EKF for a lab environment. We repeated the process for the particle filter using 2000, 1000, 500, 300 and 150 particles. The cases were run using the same look up table generated by the RSSI signal simulator.

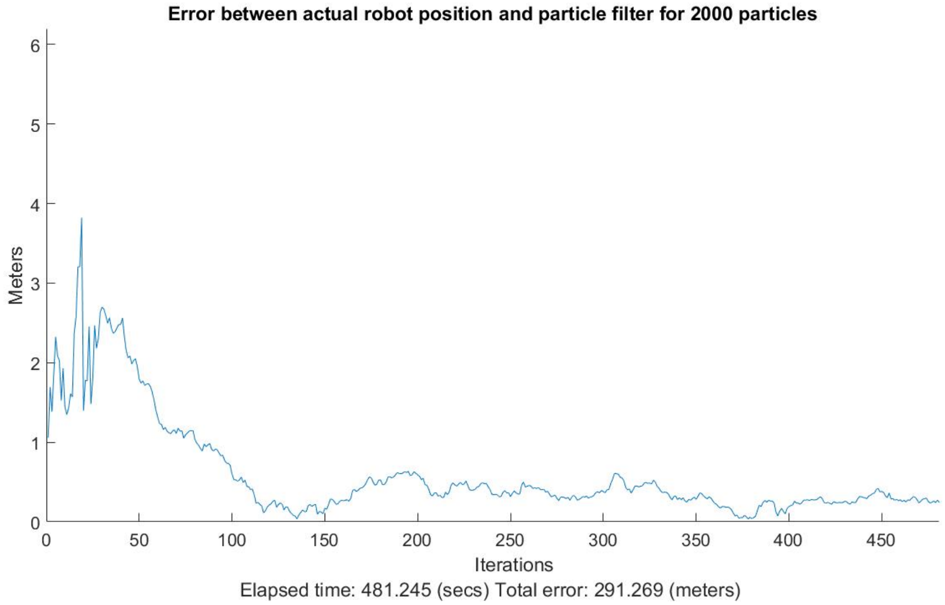


Figure 4.29: The error in the robots position in a large open room environment with 2000 particles.

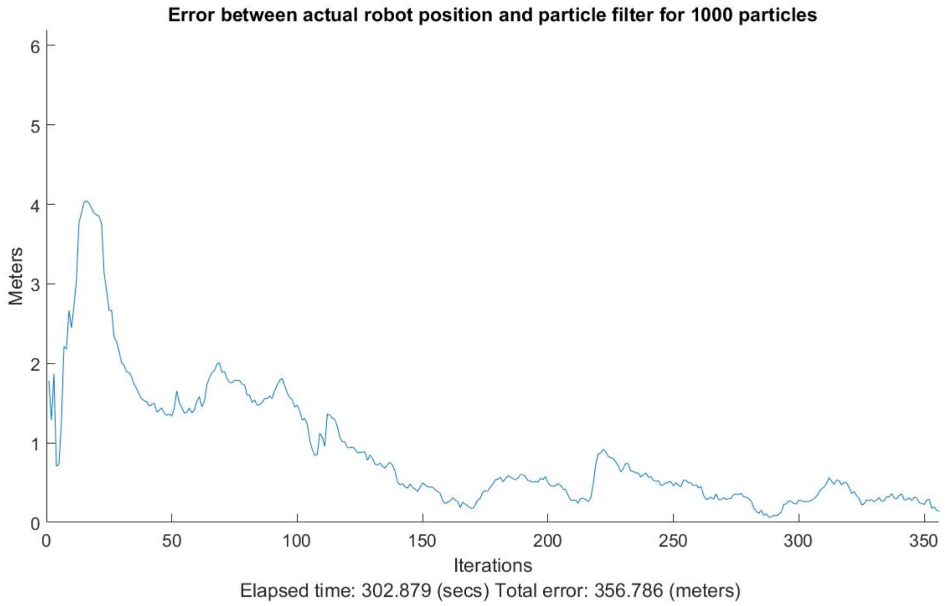


Figure 4.30: The error in the robots position in a large open room environment with 1000 particles.

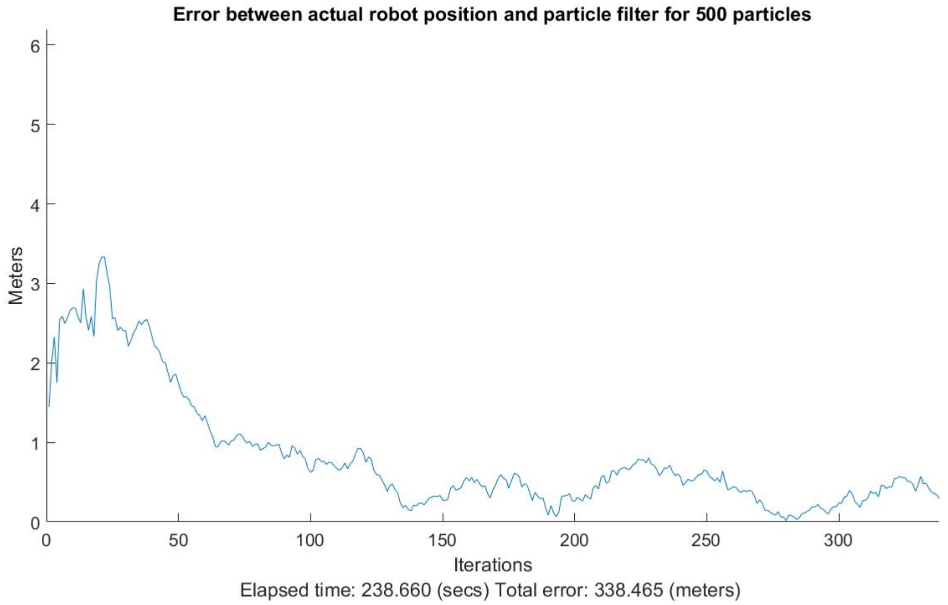


Figure 4.31: The error in the robots position in a large open room environment with 500 particles.

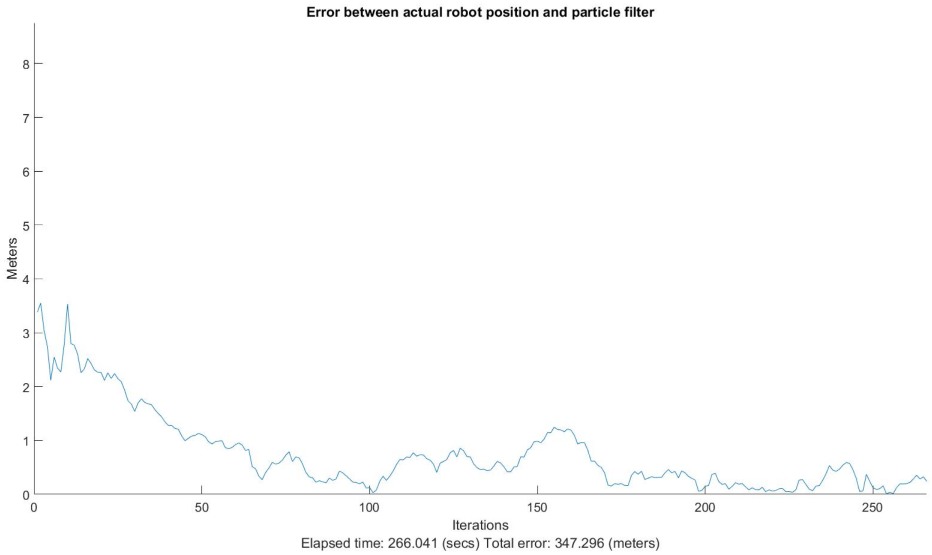


Figure 4.32: The error in the robots position in a large open room environment with 300 particles.

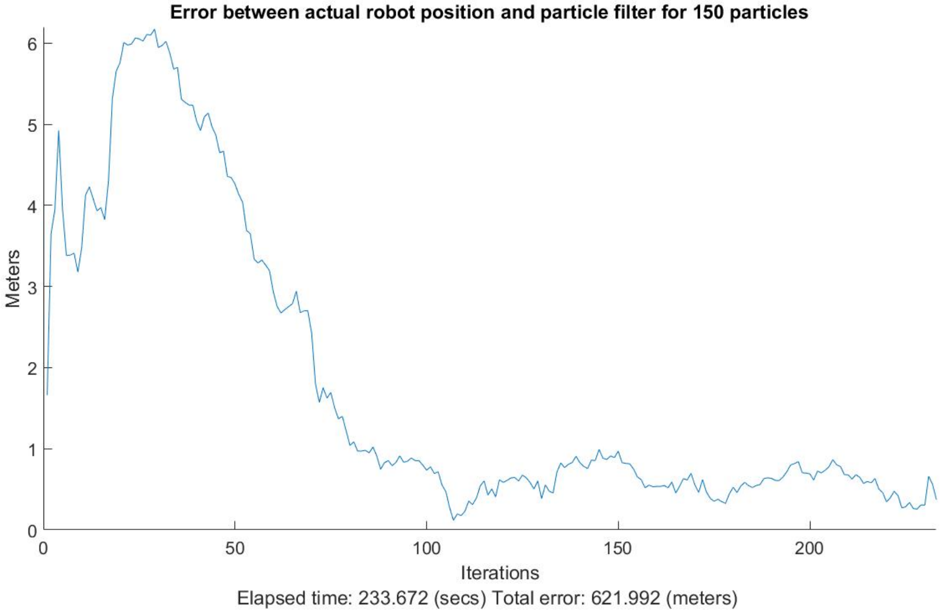


Figure 4.33: The error in the robots position in a large open room environment with 150 particles.

The Figure: 4.29, Figure: 4.30 have the fastest convergence time and least error collected over the path with a negligible difference. As we reduce the number of particles we have a trade off in the convergence time and error incured over the path as seen in Figures: 4.29 - 4.33 with the particle filter with 150 particles taking the most time to converge along with the most accumulated error incured.

CHAPTER 5: CONCLUSION

Existing pre intilization techniques for EKF such as multi-lateration and particle filter techniques are explored. These methods have a high probability of failure in highly noisy environments due to landmarks being intialised at false locations. This occurs when the convergence or multilateration occurs on a multipath signal. This causes the EKF to diverge and hence resulting in false results. Existing RSSI models to model multipath fading into account is discussed. But most of the models are more suited to combat mutipath in outdoor applications, failing to correctly model indoor enviornments.

A novel configuration for the particle filter is proposed and implemented in a simulation using realistic RSSI measurements. It is compared to RO EKF from [23] and dead reckoning in different environments varying in the degrees of multipath propagation. It is tested for both true and false landmark initializations. Performance metrics such as total error incured over the path and landmark localization error is collected and compared. The noise from the beacons are also compared to show the distribution in the error.

In conclusion, with true landmark initialization the EKF and the particle filter incurs comparable amounts of error depending on the mutipath signals received. However the EKF copes poorly with the non Gaussian noise presented by mutipath. When it is falsely initialized in RO EKF it barely updates the robots position from the measurement and hence it accumulates the error incured from odometery and becomes comparable to dead reckoning. The non-Gaussian distribution of the measurement noise furthur causes the landmark to be inaccurately corrected. The results also determine the minimum number of particles required to represent the error distribution in the RSSI signals so that positions related to multipath signals have some probability of being re-sampled along with the line of sight signals. Thus represnting the entire distribution. In our case anything greater than 1000 particles have negligible improvement in results.

5.1 Future work

Further research on this topic includes physical implementation of the particle filter using turtlebots for the robot and UVW radio beacons. Ultra wideband solutions have proven to provide a much more robust distance estimation due to the ability to filter out multipath by examining how each frequency is attenuated for the same transmission. Another line of research can also be developing efficient computational ways of parallel computation using GPUs as the particle filter is a computationally expensive process.

REFERENCES

[1] S. L. Shue, *Utilization of wireless signal strength for mobile robot localization in indoor environments*. PhD thesis, 2017.

[2] S. Shue and J. M. Conrad, "Procedurally generated environments for simulating rssi- localization applications," in *Proceedings of the 20th Communications & Networking Symposium*, CNS '17, (San Diego, CA, USA), pp. 7:1–7:11, Society for Computer Simulation International, 2017.

[3] S. Thrun, "Probabilistic robotics," *Communications of the ACM*, vol. 45, no. 3, 2002.

[4] G. Grisetti, C. Stachniss, and W. Burgard, "Improved Techniques for Grid Mapping With Rao-Blackwellized Particle Filters," *IEEE Transactions on Robotics*, vol. 23, pp. 34–46, Feb 2007.

[5] J. Aulinas, Y. Petillot, J. Salvi, and X. Lladó, "The slam problem: A survey," in *Proceedings of the 2008 Conference on Artificial Intelligence Research and Development: Proceedings of the 11th International Conference of the Catalan Association for Artificial Intelligence*, pp. 363–371, IOS Press, 2008.

[6] R. Sabatini and S. Ramasamy, "Indoor Navigation using Distributed Ultrasonic Beacons," *Australian International Aerospace Congress 2017 (AIAC17)*.

[7] H. Farahani, *ZigBee Wireless Network and Transceivers*. 2008.

[8] W. Murphy and W. Hereman, *Determination of a position in three dimensions using trilateration and approximate distances*. PhD thesis, 1995.

[9] S. Thrun, "Particle Filters in Robotics," *Proceedings of Uncertainty in AI*, vol. 1, pp. 511–518, 2002.

[10] P. Newman and J. Leonard, "Pure range-only sub-sea SLAM," in *2003 IEEE International Conference on Robotics and Automation (Cat. No.03CH37422)*.

[11] S. E. Webster, R. M. Eustice, H. Singh, and L. L. Whitcomb, "Advances in single-beacon one-way-travel-time acoustic navigation for underwater vehicles," *International Journal of Robotics Research*, vol. 31, no. 8, pp. 935–950, 2012.

[12] G. Vallicrosa, P. Ridao, D. Ribas, and A. Palomer, "Active Range-Only beacon localization for AUV homing," in *IEEE International Conference on Intelligent Robots and Systems*, pp. 2286–2291, 2014.

[13] F. Caballero, L. Merino, and A. Ollero, "A general Gaussian-mixture approach for range-only mapping using multiple hypotheses," *Proceedings - IEEE International Conference on Robotics and Automation*, pp. 4404–4409, 2010.

[14] L. Geneve, O. Kermorgant, and E. Laroche, "A composite beacon initialization for EKF range-only SLAM," in *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 1342–1348, IEEE, Sep 2015.

[15] E. Menegatti, A. Zanella, S. Zilli, F. Zorzi, and E. Pagello, "Range-only SLAM with a mobile robot and a Wireless Sensor Networks," in *2009 IEEE International Conference on Robotics and Automation*, pp. 8–14, IEEE, May 2009.

[16] A. Bhandari, A. Kadambi, R. Whyte, C. Barsi, M. Feigin, A. Dorrington, and R. Raskar, "Resolving Multi-path Interference in Time-of-Flight Imaging via Modulation Frequency Diversity and Sparse Regularization," Apr 2014.

[17] G. Seco-Granados, *Antenna Arrays for Multipath and Interference Mitigation In GNSS Receivers.* PhD thesis, Universitat Politecnica De Catalunya, Department of Signal Theory and Communications, July 2000.

[18] A. Doucet, N. De Freitas, and N. Gordon, "Sequential Monte Carlo Methods in Practice," *Springer New York*, pp. 178–195, 2001.

[19] J. J. Borenstein, H. R. Everett, and L. L. Feng, *Navigating mobile robots : systems and techniques.* A K Peters, 1996.

[20] S. Engelson and D. McDermott, "Error correction in mobile robot map learning," in *Proceedings 1992 IEEE International Conference on Robotics and Automation*, pp. 2555–2560, 1992.

[21] T. Dean and M. Boddy, "An analysis of time-dependent planning," in *Proceedings of the Seventh AAAI National Conference on Artificial Intelligence*, AAAI'88, pp. 49–54, AAAI Press, 1988.

[22] S. Zilberstein and S. Russell, *Approximate Reasoning Using Anytime Algorithms*, pp. 43–62. Boston, MA: Springer US, 1995.

[23] A. Willis, "uncc mobile robotics." `https://github.com/uncc-visionlab/uncc_mobile_robotics_tk`, 2017.