Pune Institute of Computer Technology
Dhankawadi, Pune

A SEMINAR REPORT
ON

COMPARISON OF BAYESIAN FILTERING ALGORITHMS FOR RSSI

SUBMITTED BY

Athang Gupte
Roll No. 31206
Class TE-2

Under the guidance of
Prof. Ashwini Jewalikar

DEPARTMENT OF COMPUTER ENGINEERING
Academic Year 2019-20

DEPARTMENT OF COMPUTER ENGINEERING

Pune Institute of Computer Technology

Dhankawadi, Pune-43

CERTIFICATE

This is to certify that the Seminar report entitled

"COMPARISON OF BAYESIAN FILTERING ALGORITHMS FOR RSSI"

Submitted by

Athang Gupte Roll No. 31206

has satisfactorily completed a seminar report under the guidance of Prof. Ashwini Jewalikar towards the partial fulfillment of third year Computer Engineering Semester II, Academic Year 2019-20 of Savitribai Phule Pune University.

Prof. Ashwini Jewalikar
Internal Guide

Prof. M.S.Takalikar
Head
Department of Computer Engineering

Place:
Date:

ACKNOWLEDGEMENT

I sincerely thank our Seminar Coordinator Prof. B.D.Zope and Head of Department Prof. M.S.Takalikar for their support.

I also sincerely convey my gratitude to my guide Prof. Ashwini Jewalikar, Department of Computer Engineering for her constant support, providing all the help, motivation and encouragement from beginning till end to make this seminar successful.

Contents

List of Tables

List of Figures

List of Abbreviations

GPS Global Positioning System
BLE Bluetooth Low Energy
LoS Line of Sight
NLoS Non-Line of Sight
RSSI Received Signal Strength Indicator
WSNs Wireless Sensor Networks
UWB Ultra Wide Band
KF Kalman Filter
EKF Extended Kalman Filter
UKF Unscented Kalman Filter
PF Particle Filter
SISPF Sequential Importance Sampling
SIRPF Sampling Importance Resampling
HMCM Hidden Markov Chain Model
SMA Simple Moving Average

Abstract

Increasing demand for Indoor Positioning has motivated researchers to explore more efficient and accurate methods. Most contemporary real-time solutions involve measuring RSSI and converting the measurement into distance. Yet RSSI values are not reliable due to high fluctuations caused by environmental factors and multiple reflections.

This raises the need for algorithms and techniques for reducing the RSSI fluctuations and its ill effects on accuracy. This problem is modelled by a state-space representation and Bayesian filtering are used for efficiently solving it with high accuracy. The most commonly used algorithm for this purpose is the *Kalman Filter*, which is ideal for linear systems. For nonlinear systems, simple additions to the *Kalman Filter* framework are incorporated as in the *Extended Kalman Filter* and the *Unscented Kalman Filter*. Then there are nonparametric approaches to the Recursive Bayesian Filtering paradigm, implementations of which include *Particle Filter* and *Non-Parametric Information Filter*. Some extended versions of the Kalman Filter try to improve efficiency and accuracy by incorporating other filters, such as Gaussian-Kalman Linear Filter.

This paper intends to compare between two most commonly used filters for RSSI localization problems, Kalman Filter and Particle Filter, in terms of accuracy, power consumption and efficiency. This study intends to help engineers to select from amongst prevalent technologies depending upon their proposed applications.

Keywords

Indoor Navigation, BLE, RSSI, Kalman Filter, Particle Filter.

1 INTRODUCTION

In recent *Indoor Localization* has seen a growing importance in a large number of applications such as healthcare, monitoring, tracking, etc. Global Positioning System (GPS) has proven to be unparalleled in outdoor contexts, but due to insufficient satellite coverage inside, it soon loses its applicability in indoor contexts. Indoor positioning also faces a lot of challenges due to errors by multipath and Non-Line of Sight (NLoS) conditions caused by shadowing from a greater density of moving obstacles including people. These conditions result in a non-homogenous propagation channel, high attenuation and signal scattering, thus engendering the demand for higher precision and accuracy.

Technologies proposed for solving indoor localization include vision, ultrasound, infrared, RFID, WiFi and Bluetooth [1]. Of these technologies, RFID, WiFi and Bluetooth technologies are the most economical and have thus gained popularity. In addition to this, a sizeable percentage of people presently own smartphones, which have WiFi and Bluetooth embedded in-built. This means that the indoor positioning systems using WiFi and Bluetooth does not require the users or customers to wield any extra gadgets. This makes them favourable for tracking and navigating persons indoors.

The basic idea behind WiFi and Bluetooth based localization is to have a receiver pick up the advertising packets from beacons around it and analyze its received signal strength. The received strength is nothing but the voltage of the received packet's signal. This received strength is measured as Received Signal Strength Indicator (RSSI). RSSI is inherently detected by most present equipment and is measured in dBm (deciBel-meter). The advantage of using RSSI for localization is that it can be used to calculate distance between the transmitter and the receiver. The radio signals involved in calculating RSSI follow the *log-distance Path Loss model*. Thus by applying the inverse equation using the transmitter's TxPower and the RSSI on the receiver side we can estimate the distance between the two nodes.

But the problems mentioned above cause fluctuations in the RSSI which often make the raw reading unreliable. Thus the need arises for devising methods to filter out the noise from the raw values. Thus this paper compares two common state-of-the-art algorithms for filtering, i.e. Kalman Filter and Particle Filter. In section 2 of this report we will describe the various problems faced in measuring RSSI. Section 3 describes related work done in the field with state-of-the-art and some novel methods. Sections 5 and 6 establish the mathematical constructs required for the implementation. The experimental set-up is described in section sec:exp and the results of the experiment are given in section sec:res. Finally section sec:conc concludes the report.

2 MOTIVATION

Despite the advantages of using RSSI, accurate measurement of RSSI for actual use in real-time Indoor Positioning Systems faces many problems. These issues mainly arise because the effect of the environment and the devices themselves is very high on the signal. These effects involve:

1. **Multipath Effect**
   In indoor environments the signal received by the receiver contains not only the waves following direct Line of Sight (LoS), but also a large number of signals that have reflected from walls and other obstacles [2]. These reflected radio waves are delayed and have greater attenuation than the actual LoS signal and thus the RSSI values of these signals are incorrect and add to the noise of the signal.

2. **Shadow Fading Effect**
   The presence of obstacles in the LoS of the transmitter and the receiver causes attenuation of the signal and thus the received RSSI is lower than that of actual LoS. According to [3], the wireless signal strength has shadow fading effect of the *log-normal distribution* (i.e. the logarithm of the shadowing function is normally distributed).

3. **Interference**
   Indoor environments can have a lot of radio traffic including WiFi signals, Bluetooth radios, cellular transmissions, and a lot of other magnetic interfering sources. These signals especially WiFi and Bluetooth cause interference in the Bluetooth Low Energy (BLE) radio waves and add to the noise of the received signal strength.

4. **Environmental Effects**
   Big magnetic metallic objects, like iron in the RCC structure of buildings, may distort the magnetic field and thus inroduce noise in the nearby signals.

The nature of the noise in RSSI has the following characteristics:

- Even static beacon and receiver arrangements have high errors and variance **??**.
- Different receivers have possess different amplitude signatures. This difference is not stable across distance.
- The noise has no definable characteristics across the distance range, so it can be treated equally for all distances [4].
- For moving arrangement of beacons and receivers, the RSSI may present results that do not necessarily convey the correct motion characteristics, and are thus unreliable for measurements other than average distance and thus the inferred position.

3 LITERATURE SURVEY

The problem of filtering RSSI has been attended to by many researchers since the late 90s. One of the most direct uses of RSSI as well as the focus of this report is in context of localization. There have been many approaches toward filtering of RSSI to make it usable in finding the distance between the transmitter and the receiver.

Some papers try out simple arithmetic estimation algorithms so as to obtain high computational efficiency.

1. In [5] they have tested and compared four algorithms, namely Simple Moving Average (SMA), Exponential Moving Average, Moving Median and Moving Mode, for static nodes in LoS environments. Their results show that SMA and EMA methods had better overall filtering performance than the other two methods. Larger window sizes proved to improve accuracy at the cost of additional memory usage.

2. The researchers of [6] experimented with SMA and Alpha Trimmed Mean and compared with Kalman Filter. Their experiment was conducted with static transmitter and receiver and a moving obstacle between them. They conclude that KF works best for noise reduction followed by SMA. They also compare performance in delays – Settling and Response Delay – results of which place KF at the top followed by Alpha Trimmed Mean.

3. Paper [7] presents a new algorithm, which they named RSSI Moving Median algorithm (RMM), that is based on a common method called Median Filter, and its comparison with Kalman Filter. They use a median filter as a base and then update the mean value for each sample. Their tests conclude that despite the better filtering accuracy provided by the KF in static situations, the RMM algorithm works better in a dynamic context involving mobility and existence of moving obstacles.

4. A novel extension of the Moving Average filter has been presented in [4]. They propose the Adaptive-Bounds Band-Pass Moving-Average Filter. The filter, as derivative from the name, first smooths the signal using a Simple Moving Average, then sorts the window data and cuts the extreme values using a Band-Pass filter, and lastly uses an Adaptive-Bounds mechanism to adapt to the natural variations of the signal. They were able to achieve upto 20% improvement in relative average error as tested on four different devices.

5. In [8], a 2-stage multipath mitigation technique is proposed. The technique consists of RSSI filtering by a low comlexity Weighted Average Filter with a threshold, and RSSI combining method involving the use of space diversity where RSSI values obtained from two BLE tags were combined to enhance signal stability. They employed anchor node optimization which then produced fairly high performance improvements.

Most other research has been along the grounds of stochastic and probabilistic approach such as Bayesian Filtering methods.

6. A simple static Kalman Filter (KF) was implemented by the authors of [9] and their results show that the Kalman Filter provides impressive improvements to system performance. The overall error was improved by upto 79% by applying the filter.

Their observations depict the effect of the size and locality of the geo-fence in the resulting accuracy. Another notable observation is that as the distance between the anchor nodes, i.e. the beacons, increases the system is less affected by noise. Thus their system was suggested for larer indoor environments.

7. A few other papers, including [10] and [11], also implement the Kalman filter in order to solve the multipath and shadowing problems. They also promote the advantages of optimization using KF and their experimental results support their claims.

8. The paper [12] compares Kalman Filter (KF), Extended Kalman Filter (EKF) and the Particle Filter (PF) in the context of a constant velocity motion model. The tracking is done in a 2-Dimensional model where the RSSI from one target node moving with a constant speed and sudden changes in its direction is observed. The results show that EKF exhibits superior performance than KF and the PF, despite being computationally expensive, outperforms EKF and KF by a big margin, especially a when large number of particles are used. For a smaller number of particled EKF may show better performance than PF.

9. Another in-depth comparison has been presented in [13], between Kalman Filter (KF), Extended Kalman Filter (EKF), Unscented Kalman Filter (UKF) and PF using Ultra Wide Band (UWB) positioning system. They inferred that the performance of the filters depends on the system model as well as the actual implementation. The experimental results show that UKF delivered the best performance among all the filters. They found that the PF always performed below par with the Kalman filters. PF can always be applied regardless of linearity or non-linearity of the model, but its exceedingly high runtimes make it less applicable in real-time systems.

10. The solution to RSSI filtering, advocated in [14], is to incorporate two filters namely the Gaussian Filter and a Simple Averaging Filter. The Gaussian filter optimizes the wider fluctuations in the readings and the Averaging filter smoothes out the resulting values. They obtained significant improvement in their localization estimation using the suggested algorithm.

11. [15] demonstrated use of multiple antennas on the receiver, combining the readings and filtering using the PF to improve noise reduction and accuracy. They used different combinations of particle filter, ground-reflection model, log-normal model using different antenna-arrangements. Their results showed that the accuracy was better when using the proposed particle filter along with the ground-reflection model. They also inferred from the experiments that the selection of different resampling methods is not critical for the accuracy of the given algorithm.

12. A novel method is introduced in [16], which takes advantage of two filters – Gaussian Fitting Filter and Kalman Filter. Their analysis posits that the wide fluctuations in the RSSI cannot be completely eliminated by either Gaussian filter or Kalman filter alone. Thus they put forward the Gaussian-Kalman filtering algorithm. They first use the Gaussian Fitting filter with a large number of RSSI values. The estimate and standard deviation are used in the KF and finally linear regression is conducted on the output. Their algorithm was able to surpass both the Gaussian and Kalman filters in accuracy and noise elimination.

13. The authors of [17] apply adaptive Kalman filters to simultaneously estimate the position and channel parameters in Wireless Sensor Networks (WSNs). They use the Unscented Kalman Filter (UKF) [18] along with a time-variant noise process tracked by a Variational Bayesian approximation. The resultant algorithm, the Variational Bayesian Adaptive Unscented Kalman Filter (VBAUKF), does not suffer from performance degradation and the noise estimates match with the channel model. Adaptive filtering enables for optimal tracking performance in scenarios where the target objects traverse regions with different propagation characteristics.

14. In [19] a hybrid Kalman filter based filtering method is proposed to reduce the effect of the Kalman filter on the peak value of the pulse deviation by the Grubbs deviation test. The Grubbs deviation test eliminates the abnormal data by the range ratio judgement. This has shown to provide significant improvement in performance especially to eliminate the sudden large spikes.

15. A cascaded algorithm combining KF and PF was presented in [20]. The accuracy of localization is improved by 28.16% in 2D and 25.59% in 3D environmental contexts when compared to using only a PF. KF was used to smooth the signal and the smoothed RSSI values were used as input to PF for non-linear tracking.

16. Another novel technique is proposed by the authors of [20] in [21]. They present a cascaded algorithm that combines PF and EKF in series to reduce the impact of multipath effects and noise on the RSSI. When compared with a lone PF, their Particle Filter - Extended Kalman Filter (PF-EKF) algorithm improved localization accuracy by 31.3% anf 33.9% in 3D and 2D environments respectively. They also showed that the PF-EKF algorithm outperforms their previous work, the KF-PF cascaded algorithm in [20].

17. In paper [22], a new algorithm is proposed with Kalman Filter in order to improve localization accuracy for target tracking in real-time applications in indoor and outdoor environments. The algorithm, named as Motetrack InOut system by the authors, outperformed a simple Kalman Filter by significant margin. In this algorithm they estimate the error using average value and then reduce intensity of the error before filtering it using KF. The error reduction in localization was 74.5% for indoor and 75.6% for outdoor environments when using Motetrack InOut system for a moving target.

4 PROBLEM DEFINITION AND SCOPE

4.1 Problem Definition

To compare different algorithms for filtering the Received Signal Strength Indicator (RSSI) of a Bluetooth Low Energy (BLE) beacon as received by a mobile smartphone and infer suitable results.

4.2 Scope

The scope of this seminar and report is to compare Kalman Filter and Particle Filter (Sampling Importance Resampling) in the context of Bluetooth RSSI. The comparison parameters include accuracy, noise reduction and computational efficiency. The sampling is done between two static nodes.

5 MATHEMATICAL MODEL

The State Space Model

State-Space model is a mathematical model used to represent a system using state variables and first-order differential equations or difference.[23, 24] They are defined by a set of inputs, outputs and state variables. The state-space model essentially describes otherwise notationally intractable problems, defined by n-th order differential equations, in a notationally tractable form by reducing to first-order differential equations[25]. The state variables depend on inputs and outputs upto the current time-step but are not themselves measured in the experiment.

For a temporal dynamic system, like Real-time Indoor Positioning, the following generalised state space model of a nonlinear system is considered:

$$x_t = f\left(x_{t-1}, u_t, w_{t-1}\right) \tag{5.1}$$

$$y_t = h\left(x_t, v_t\right) \tag{5.2}$$

where at time t, x_t is the *process state*, w_t is the *system* or *process noise*, y_t is the *measurement* or *observation* and v_t is the *measurement noise*[26].

The equations (5.1) and (5.2) are refered to as the ***process model*** and the ***measurement model*** respectively. The *process model* describes the generation of the current state vector x_t by a nonlinear function f of the previous state x_{t-1}, the current input u_t and the process noise w_{t-1}. The *measurement model* is used to obtain the current measurement from a nonlinear function h of current process state x_t and the measurement noise v_t. These 2 equations form the basis for most linear estimation algorithms, such as the *Kalman Filter*.

The Hidden Markov Chain

The above model of a temporal dynamic system is assumed to be a first-order ***Markov Chain***[27]. A Markov model is a stochastic modek used to model randomly changing systems. This means that the current state x_t only depends on the last state x_{t-1}, and the current observation y_t is determined only by the current state x_t and not on the states before that. Thus the likelihood density of current state x_t conditional to all the previous states $x_{0:t-1} = \{x_1, x_2, \cdots, x_{t-1}\}$ is equal to the transition density of last state to current state, that is —

$$p\left(x_t|x_{0:t-1}\right) = p\left(x_t|x_{t-1}\right)$$

and the probability density of the current observation y_t given all the states $x_{0:t}$ and all the previous measurements $y_{0:t-1}$ is equal to the current transition density, that is —

$$p\left(y_t|y_{0:t-1}, x_{0:t}\right) = p\left(y_t|x_t\right)$$

The figure 1 represents the state-space model as a functional form of the Hidden Markov Chain Model (HMCM). The Markov Chain is hidden as it cannot be altered nor

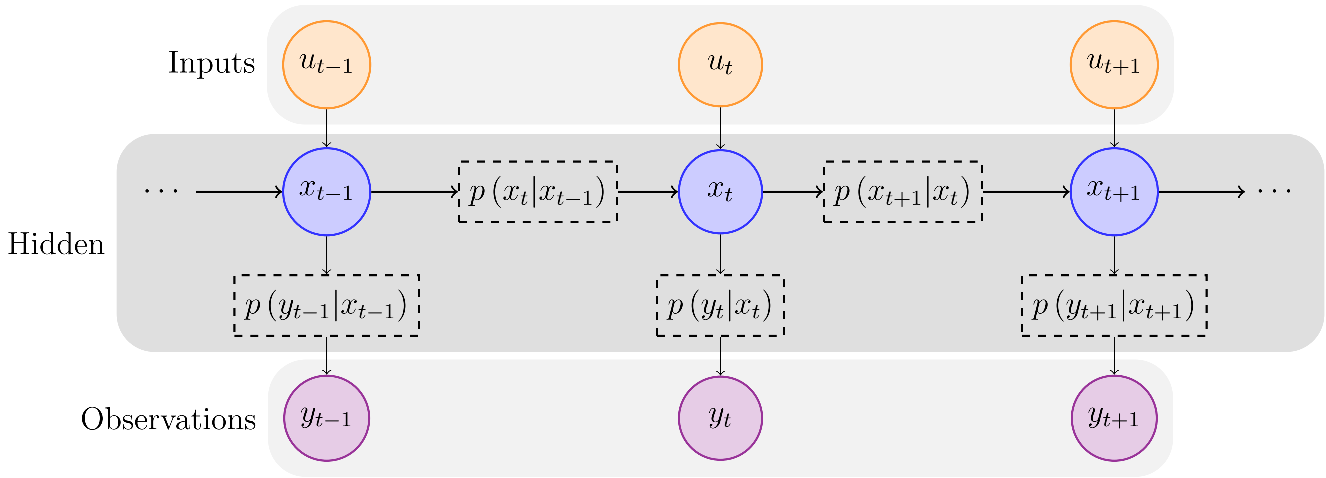


Figure 1: State-Space Model with the Hidden Markov Chain

measured directly. It comprises of the *initial* density $p(x_0)$, *transition* density $p(x_t|x_{t-1})$, and the *likelihood* density $p(y_t|x_t)$.

Recursive Bayesian Filtering

The objective of filtering is to estimate the optimal current state at a time t based upon the observations upto time t, which is in fact the *posterior probability distribution*, i.e. $p(x_t|y_{0:t})$ [28]. Thus by considering the values obtained from the HMCM, we get —

$$p(x_t|y_{0:t}) = \frac{p(y_{0:t}|x_t)\, p(x_t)}{p(y_{0:t})}$$

By using *Bayes Rule* and the properties of the HMCM discussed above we can convert the above equation into the *Chapman-Kolmogorov* equation which can be computed recursively —

$$p(x_t|y_{0:t}) = \frac{p(y_t|x_t)\, p(x_t|y_{0:t-1})}{p(y_t|y_{0:t-1})} \tag{5.3}$$

where

$$p(x_t|y_{0:t-1}) = \int p(x_t|x_{t-1})\, p(x_{t-1}|y_{0:t-1})\, dx_{t-1} \tag{5.4}$$

$$p(y_t|y_{0:t-1}) = \int p(y_t|x_t)\, p(x_t|y_{0:t-1})\, dx_t \tag{5.5}$$

The equations (5.4) and (5.5), respectively the predict the next state and update the prediction with the latest observation. These three equations represent the ***recursive Bayesian filtering framework***. As given by equation (5.3) *posterior* density is defined by – *prior* density $p(x_t|y_{0:t-1})$, *likelihood* $p(y_t|x_t)$ and *evidence* $p(y_t|y_{0:t-1})$ [28].

Although the model described above provides a notationally tractable representation of the filtering problem, whose complete solution is given by the posterior probability density (5.3), the problem may remain intractable since the posterior density is a function and not a finite-dimensional point estimate [28]. The tractability depends on whether the integrations in the equations (5.4) and (5.5) can be solved or not. These intergrations are determined by the characteristics of the dynamic system which also form the classification parameters for the existing filtering techniques [26].

6 ALGORITHMS

6.1 Kalman Filter

One of the most well-known and frequently applied stochastic estimation tools is the ***Kalman Filter***, named after the Rudolph E. Kalman who first devised the method in his paper [29]. This filter works under the assumption that the system is a *linear gaussian* system. Thus the equations (5.1) and (5.2) of the *state-space model* are reconstructed as —

$$x_t = \mathbf{F}x_{t-1} + \mathbf{G}u_t + w_{t-1} \tag{6.1}$$

$$y_t = \mathbf{H}x_t + v_t \tag{6.2}$$

where

$$\begin{aligned}
\mathbf{F} &\rightarrow \text{constant linear transition matrix} \\
\mathbf{G} &\rightarrow \text{constant input matrix} \\
\mathbf{H} &\rightarrow \text{constant observation matrix} \\
w_t &\rightarrow \text{process noise distribution} \sim N\left(0, \mathbf{Q}\right) \\
v_t &\rightarrow \text{measurement noise distribution} \sim N\left(0, \mathbf{R}\right) \\
\mathbf{Q} &= E\left[w_t w_t^T\right] \\
\mathbf{R} &= E\left[v_t v_t^T\right]
\end{aligned}$$

Gaussian distribution is closed under linear transformation, thus the transition density is $p\left(x_t|x_{t-1}\right) \sim N\left(\mathbf{F}x_{t-1}, \mathbf{Q}\right)$ and the likelihood is $p\left(y_t|x_t\right) \sim N\left(\mathbf{H}x_t, \mathbf{R}\right)$. This enusres that the integrations in the equations (5.4) and (5.5) will produce gaussian distributions and therefore, sequentially the *posterior* density will be gaussian.

The general framework of the Kalman Filter involves two main steps – ***Time update*** or *"Predict"* and ***Measurement update*** or *"Correct"*. Considering $\hat{x}'_t$ to be the *a priori* estimate at time t with P'_t as the *a priori* error covariance; and $\hat{x}_t$ to be the *a posteriori* estimate at time t with P_t as the *a posteriori* error covariance, we get the equations —

Time Update

— *Priori* state estimate

$$\hat{x}'_t = \mathbf{F}\hat{x}_{t-1} + \mathbf{G}u_t \tag{6.3}$$

— *Priori* error covariance estimate

$$\mathbf{P}'_t = \mathbf{F}\mathbf{P}_{t-1}\mathbf{F}^T + \mathbf{Q} \tag{6.4}$$

Measurement Update

— *Kalman Gain*

$$\mathbf{K}_t = \mathbf{P}_t'\mathbf{H}^T\left(\mathbf{H}\mathbf{P}_t'\mathbf{H}^T + \mathbf{R}\right)^{-1} \tag{6.5}$$

— *Posteriori* state estimate

$$\hat{x}_t = \hat{x}_t' + \mathbf{K}_t\left(y_t - \mathbf{H}\hat{x}_t'\right) \tag{6.6}$$

— *Posteriori* error covariance estimate

$$\mathbf{P}_t = \left(\mathbf{I} - \mathbf{K}_t\mathbf{H}\right)\mathbf{P}_t' \tag{6.7}$$

6.2 Extended Kalman Filter

We have seen the working of the Kalman filter for linear systems, but in nature processes are rarely linear and gaussian. While the Kalman filter works excellently for linear gaussian processes, it fails for nonlinear systems. The most intuitive step would be to linearize these nonlinear processes into a linear form. An often-used method for linearization of an equation is to use the first-order terms of the Taylor series expansion of the system. This is exactly what the ***Extended Kalman Filter*** does. Without loss of generalization we can write the equations of the nonlinear *state-space model* as —

$$x_t = f\left(x_{t-1}, u_t\right) + w_{t-1} \tag{6.8}$$
$$y_t = h\left(x_t\right) + v_t \tag{6.9}$$

Thus by linearizing the state estimation equation at some arbitrary *posterior* estimate $\hat{x}_{t-1}'$, we get —

$$x_t \approx \hat{x}_t' + \mathbf{F}_t\left(x_{t-1} - \hat{x}_{t-1}\right) + w_{t-1}$$
$$y_t \approx h\left(\hat{x}_t'\right) + \mathbf{H}_t\left(x_t - \hat{x}_t'\right) + v_t$$

where

$$\begin{aligned}
x_t &\rightarrow \text{actual state vector}\\
y_t &\rightarrow \text{actual measurement vector}\\
\hat{x}_t' &\rightarrow \text{current } \textit{a priori} \text{ estimate}\\
\hat{x}_{t-1} &\rightarrow \text{previous } \textit{a posteriori} \text{ estimate}\\
w_t &\rightarrow \text{process noise distribution} \sim N\left(0, \mathbf{Q}\right)\\
v_t &\rightarrow \text{measurement noise distribution} \sim N\left(0, \mathbf{R}\right)\\
\mathbf{F}_t &\rightarrow \text{Jacobian of } f \text{ with respect to } x \text{ at } \hat{x}_{t-1}'\\
&= \frac{\partial f}{\partial x}\left(\hat{x}_{t-1}', u_t\right)\\
\mathbf{H}_t &\rightarrow \text{Jacobian of } h \text{ with respect to } x \text{ at } \hat{x}_t'\\
&= \frac{\partial h}{\partial x}\left(\hat{x}_t'\right)
\end{aligned}$$

Thus substituting the linearized model into the Kalman Filter Framework gives us the Extended Kalman Filter equations as follows —

Time Update

— *Priori* state estimate

$$\hat{x}'_t = f\left(\hat{x}_{t-1}, u_t\right) \tag{6.10}$$

— *Priori* error covariance estimate

$$\mathbf{P}'_t = \mathbf{F}_t \mathbf{P}_{t-1} \mathbf{F}_t^T + \mathbf{Q} \tag{6.11}$$

Measurement Update

— *Kalman Gain*

$$\mathbf{K}_t = \mathbf{P}'_t \mathbf{H}_t^T \left(\mathbf{H}_t \mathbf{P}'_t \mathbf{H}_t^T + \mathbf{R}\right)^{-1} \tag{6.12}$$

— *Posteriori* state estimate

$$\hat{x}_t = \hat{x}'_t + \mathbf{K}_t \left(y_t - h\left(\hat{x}'_t\right)\right) \tag{6.13}$$

— *Posteriori* error covariance estimate

$$\mathbf{P}_t = \left(\mathbf{I} - \mathbf{K}_t \mathbf{H}_t\right) \mathbf{P}'_t \tag{6.14}$$

6.3 Particle Filter

The filters described above are based on the Kalman Filter Framework and are thus all limited by the underlying Gaussian assumption, as real world systems are rarely Gaussian. Thus, for estimating more generalized processes, we use ***Monte Carlo*** methods.

6.3.1 The Monte Carlo Principle

The Monte Carlo Principle is that, if we take enough samples or particles to get a representative sample of the problem and run the particles through the system model then the transformed points can be uesd to compute the results we require. Here each particle represents a possible state of the process and the next state is estimated by performing weighted statistical analysis of the particles [26, 30]. Another advantage of Monte Carlo methods is that integration of any function or curve is trivial. A bounding box is generated around the curve within the limits and the box is filled with a large number of random particles. The ratio of the particles that lie inside the curve to the total number of particles is equal to the ratio of the area under the curve to the area of the bounding box [30]. Consider the following numerical integration problem —

$$E(f) = \int f(x)\,p(x)\,dx \tag{6.15}$$

In Monte Carlo methods, we draw N random samples $\left\{x^1, x^2, \cdots, x^N\right\}$ from a target probability distribution $p(x)$ defined on a state-space, and then estimate the problem using these samples as follows —

$$\hat{E}(f) = \frac{1}{N}\sum_{i=1}^{N} f\left(x^{(i)}\right) \tag{6.16}$$

and the probability of a state x can be determined by delta-dirac point mass function —

$$\hat{p}(x) = \frac{1}{N}\sum_{i=1}^{N} \delta_{x^{(i)}}(x) \tag{6.17}$$

6.3.2 Generic Particle Filter Framework

1. **Generate random particles**
   A large number of particles and their corresponding weights are generated. Initially all weights can be considered equal. The sum of all weights should be 1 so as to maintain total probability as 1.

2. **Predict**
   Predict the next state of the particles based on the internal system model.

3. **Update**
   Update the particle weights based on actual measurements. Particles closer to the actual measurements are weighted higher than the particles otherwise.

4. **Resample**
   To avoid *degeneracy* in the algorithm we have to discard the particles with very low likelihood, given by their weights, and replace them with more probable particles.

5. **Estimate**
   Compute the weighted average of all the particles to get an estimate of the system state.

6.3.3 Sequential Importance Sampling

The ***Particle Filter*** is a *Sequential Monte Carlo Sampling* method, where the Monte Carlo Sampling methods are applied iteratively at each step for estimating sequential states of an arbitrary system [26]. One of the most widely used Particle Filters is the ***Sequential Importance Sampling (SISPF)*** [28], in which the important region of the posterior distribution is approximated recursively. The target density $p(x)$ for most real processes is very difficult to sample. The idea behind Importance Sampling is to choose a proposal density $q(x)$ such that $q(x)$ is positive for all x where $p(x)$ is positive. With this knowledge, the equation (6.15) can be rewritten as —

$$E(f) = \int f(x) \frac{p(x)}{q(x)} q(x)\, dx \tag{6.18}$$

Applying the Monte Carlo Estimation (6.17) we get —

$$\hat{E}(f) = \frac{1}{N} \sum_{i=1}^{N} W\left(x^{(i)}\right) f\left(x^{(i)}\right) \tag{6.19}$$

where $W\left(x^{(i)}\right)$ are called *Importance Weights* and are given by —

$$W\left(x^{(i)}\right) = \frac{p\left(x^{(i)}\right)}{q\left(x^{(i)}\right)} \tag{6.20}$$

But in most real world problems the normalizing factor of $p(x)$ is not known. Thus to ensure that $\sum_{i=1}^{N} W\left(x^{(i)}\right)$, the weights are normalized. The normalized equations are —

Sequential Importance Sampling

$$\begin{aligned} \hat{E}(f) &= \frac{\frac{1}{N}\sum_{i=1}^{N} W\left(x^{(i)}\right) f\left(x^{(i)}\right)}{\frac{1}{N}\sum_{i=1}^{N} W\left(x^{(i)}\right)} \\ &\equiv \frac{1}{N} \sum_{i=1}^{N} \tilde{W}\left(x^{(i)}\right) f\left(x^{(i)}\right) \end{aligned} \tag{6.21}$$

where

$$\tilde{W}\left(x^{(i)}\right) = \frac{W\left(x^{(i)}\right)}{\sum_{i=1}^{N} W\left(x^{(i)}\right)} \tag{6.22}$$

Degeneracy

The SISPF method faces a major drawback called *Weight Degeneracy* problem. The unconditional covariance of the importance (weight) distribution increases with time. Thus over time, the number of non-zero weights reduces to such an extent that very few or even just one of the weights in $W\left(x^{(i)}\right)$ remains positive. This leads to wastage of computation time in calculating the trivial unimportant particle weights all the time.

6.3.4 Sampling Importance Resampling

The intuitive idea for solving the *weight degeneracy* problem is to eliminate the samples with small weights and duplicate the samples with larger weights. This derives motivation from the *Bootstrap and Jackknife* techniques, which involves a set of computationally intensive methods based on resampling from the observed data [31]. Usually resampling is inserted between two importance sampling steps, and this method is called as ***Sampling Importance Resampling (SIRPF)*** [26, 28].

Resampling methods can have *deterministic* or *dynamic* schedules. Deterministically scheduled methods perform resampling every $k \geq 1$ time steps, whereas dynamic schedules utilise set of thresholds on the variance of the weights to decide whether resampling should take place [30, 28]. There are many resampling methods available including *Multinomial resampling* [32], *Systematic resampling* [33] and *Residual resampling* [34].

CODE

Please refer to the source code accompanying this report for specific implementational details. The source is in Python and available as IPython notebook.

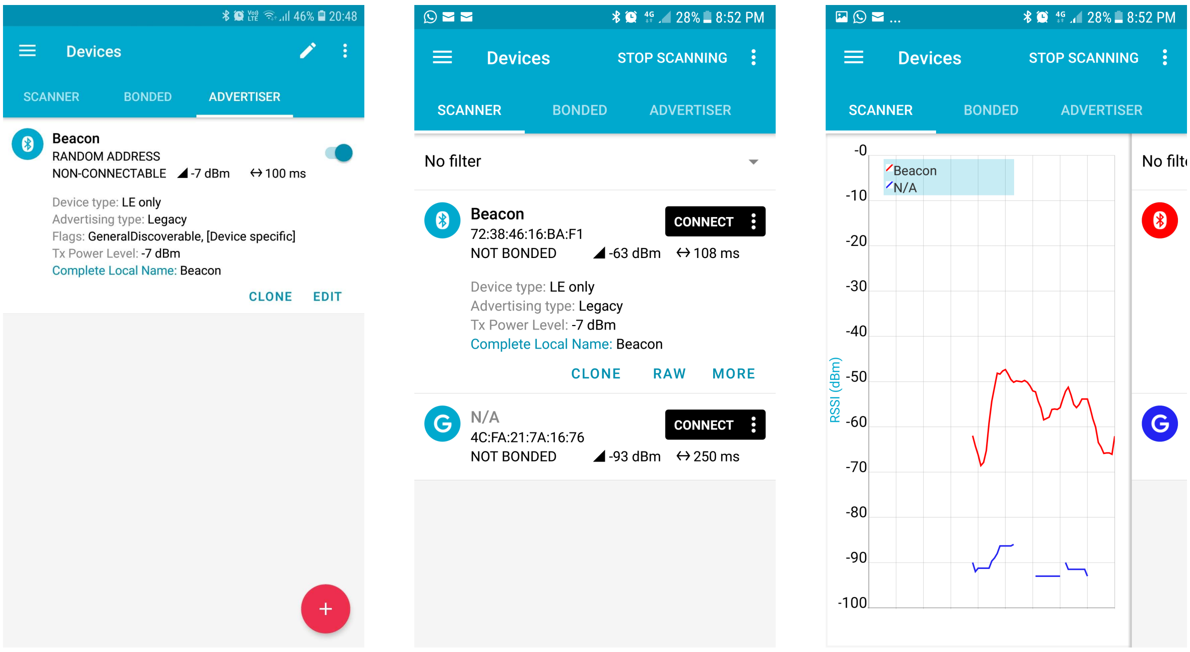
7 EXPERIMENTAL SET-UP

To perform the experiment, 2 mobile smartphones with BLE support were used, one as transmitter or *beacon* and the other as receiver. To simulate the BLE beacon advertising and store the RSSI values from the received packets, the Android application **nRF Connect** [35], by Nordic Semiconductor®, was used. The description of the two devices is given in table 1.

Role	**Transmitter**	**Receiver**
Device Name	Samsung Galaxy J7 Max	Samsung Galaxy Note 5
Model	SM-G615F	SM-N920G
Android Ver.	**8.1.0**	**7.0**
Board	mt6757	universal7420
BLE	**Yes**	**Yes**
Peri. Mode	**Yes**	**Yes**

Table 1: Device Information

To start collecting RSSI on the devices some basic configuration is requried inside the nRF Connect app as shown in figure 2.



(a) Configuration of beacon

(b) Scanning for nearby transmitters

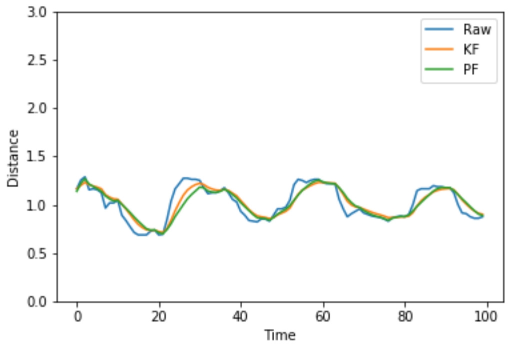
(c) RSSI graph generated

Figure 2: Configuration in nRF Connect

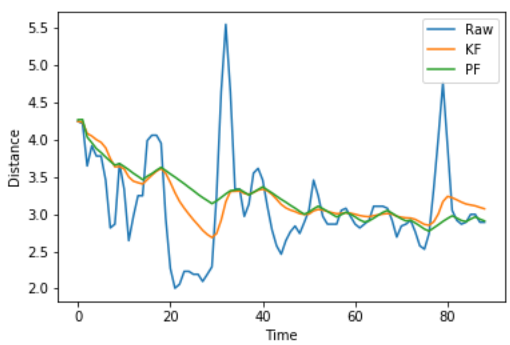
The RSSI values were collected at four different distances – $1m$, $2m$, $3m$ and $4m$

8 RESULTS AND INFERENCE

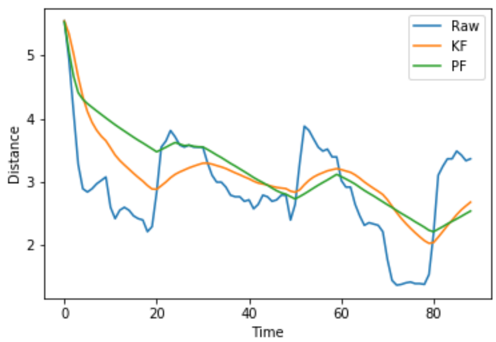
8.1 Implementation Results



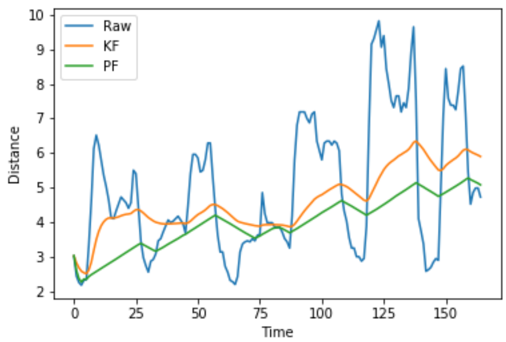
(a) Actual Distance : 1m



(b) Actual Distance : 2m

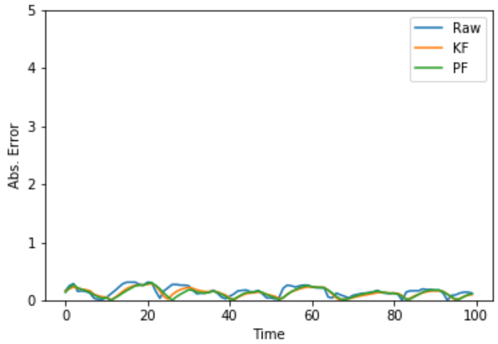


(c) Actual Distance : 3m

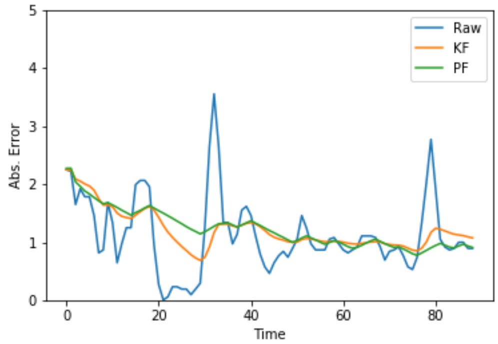


(d) Actual Distance : 5m

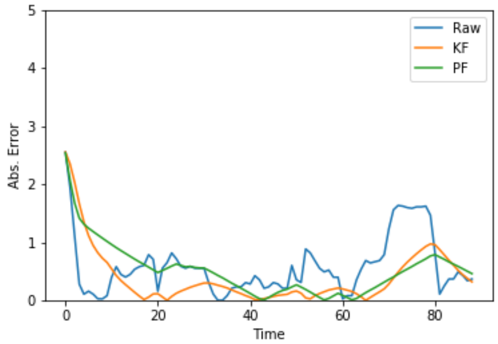
Figure 3: Results when using KF and PF on four data sets measured at different distances



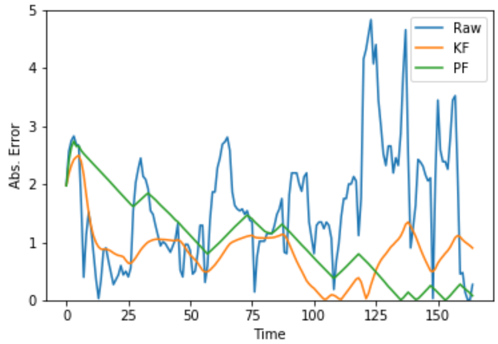
(a) Actual Distance : 1m



(b) Actual Distance : 2m



(c) Actual Distance : 3m



(d) Actual Distance : 5m

Figure 4: Absolute Errors when using KF and PF on four data sets measured at different distances

Distance		PF
1m	0.147183	0.147053
2m	1.276428	1.274565
3m	0.676280	0.678469
5m	1.269959	1.266580

Table 2: RMS Error (in m)

Distance	KF	PF
1m	1.690730	1.654867
2m	61.650579	61.555335
3m	6.454328	6.523930
5m	20.526050	20.387668

Table 3: Relative Error (in %)

8.2 Performance Testing

To test the computational performance of the algorithms, both the modules were run for N times on each of the data sets above and the time was measured. The results are given in table 4

Iterations	KF	PF
5	0.005468	4.351748
10	0.014060	9.046966
25	0.034104	21.116861
50	0.060803	42.273397

Table 4: Performance Test (in sec)

The table shows that the KF algorithm takes approximately 0.001 seconds to compute all four data sets, whereas the PF algorithm takes approx. 0.96 seconds for the same. Although this does not make PF unsuitable for online implementation but may add some overhead to the complete localization framework.

9 CONCLUSION

In this report, we explained the different causes of noise in RSSI and why the need of filters arises. The research revealed Bayesian filters, specifically Kalman Filter and Particle Filters, to be the most preferred approach by many researchers for their performance and accuracy.

Thus we presented a comparison between those two methods for filtering and estimation of distance derived from RSSI of a BLE transmitter-receiver pair. The results of the experiment showed that post filtering the readings become much more reliable and can be used for real-time distance estimation in localization. Both the algorithms produced strikingly similar results in terms of error correction and noise reduction. KF follows the curve closer than PF which often helps in reaching the actual value faster, thus reducing the time of error, whereas PF showed better noise reduction characteristics, thus flattening the noisy curve. KF showed much better performance than PF in terms of computational efficiency.

Future scope of the study will be to test the performances in different environments and with mobile nodes. The combination of the filters with smoothing methods also shows promising scope for future research.

References

[1] Yanying Gu, Anthony Lo, and Ignas Niemegeers. A survey of indoor positioning systems for wireless personal networks. *IEEE Communications surveys & tutorials*, 11(1):13–32, 2009.

[2] Jean Paul Linnartz. Wireless communication - JPL's reference website, 1996-2010. http://www.wirelesscommunication.nl/reference/chaptr03/fading/fading.htm.

[3] *TGn Channel Models*, 2003.

[4] Diego Ordóñez-Camacho and Edwin Cabrera-Goyes. An adaptive-bounds band-pass moving-average filter to increase precision on distance estimation from bluetooth rssi. In Álvaro Rocha and Teresa Guarda, editors, *Proceedings of the International Conference on Information Technology & Systems (ICITS 2018)*, pages 823–832, Cham, 2018. Springer International Publishing.

[5] Moses A Koledoye, Daniele De Martini, Simone Rigoni, and Tullio Facchinetti. A comparison of rssi filtering techniques for range-based localization. In *2018 IEEE 23rd International Conference on Emerging Technologies and Factory Automation (ETFA)*, volume 1, pages 761–767. IEEE, 2018.

[6] Rafhanah Shazwani Rosli, Mohamed Hadi Habaebi, and Md Rafiqul Islam. Comparative analysis of digital filters for received signal strength indicator. In *2018 7th International Conference on Computer and Communication Engineering (ICCCE)*, pages 214–217. IEEE, 2018.

[7] Youssef Aiboud, Issam Elhassani, Hafid Griguer, and M'hamed Drissi. Rssi optimization method for indoor positioning systems. In *2015 27th International Conference on Microelectronics (ICM)*, pages 246–248. IEEE, 2015.

[8] Juthatip Wisanmongkol, Ladawan Klinkusoom, Taweesak Sanpechuda, La-or Kovavisaruch, and Kamol Kaemarungsi. Multipath mitigation for rssi-based bluetooth low energy localization. In *2019 19th International Symposium on Communications and Information Technologies (ISCIT)*, pages 47–51. IEEE, 2019.

[9] Andrew Mackey, Petros Spachos, and Konstantinos N Plataniotis. Enhanced indoor navigation system with beacons and kalman filters. In *2018 IEEE Global Conference on Signal and Information Processing (GlobalSIP)*, pages 947–950. IEEE, 2018.

[10] Laial Alsmadi, Xiaoying Kong, and Kumbesan Sandrasegaran. Improve indoor positioning accuracy using filtered rssi and beacon weight approach in ibeacon network. In *2019 19th International Symposium on Communications and Information Technologies (ISCIT)*, pages 42–46. IEEE, 2019.

[11] Sam Shue and James M Conrad. Reducing the effect of signal multipath fading in rssi-distance estimation using kalman filters. In *Proceedings of the 19th Communications & Networking Symposium*, pages 1–6, 2016.

[12] MW Khan, N Salman, A Ali, AM Khan, and AH Kemp. A comparative study of target tracking with kalman filter, extended kalman filter and particle filter using received signal strength measurements. In *2015 International Conference on Emerging Technologies (ICET)*, pages 1–6. IEEE, 2015.

[13] Mathias Pelka and Horst Hellbrück. Introduction, discussion and evaluation of recursive bayesian filters for linear and nonlinear filtering problems in indoor localization. In *2016 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, pages 1–8. IEEE, 2016.

[14] Ranjan Kumar Mahapatra and NSV Shet. Localization based on rssi exploiting gaussian and averaging filter in wireless sensor network. *Arabian Journal for Science and Engineering*, 43(8):4145–4159, 2018.

[15] Janja Svečko, Marko Malajner, and Dušan Gleich. Distance estimation using rssi and particle filter. *ISA transactions*, 55:275–285, 2015.

[16] Kaibi Zhang, Yangchuan Zhang, and Subo Wan. Research of rssi indoor ranging algorithm based on gaussian-kalman linear filtering. In *2016 IEEE Advanced Information Management, Communicates, Electronic and Automation Control Conference (IMCEC)*, pages 1628–1632. IEEE, 2016.

[17] Thorsten Nowak, Markus Hartmann, and Jörn Thielecke. Simultaneous position and channel parameter estimation applying adaptive kalman filters. In *2018 11th German Microwave Conference (GeMiC)*, pages 115–118. IEEE, 2018.

[18] Eric A Wan and Rudolph Van Der Merwe. The unscented kalman filter for nonlinear estimation. In *Proceedings of the IEEE 2000 Adaptive Systems for Signal Processing, Communications, and Control Symposium (Cat. No. 00EX373)*, pages 153–158. Ieee, 2000.

[19] Liu Yi, Liang Tao, and Sun Jun. Rssi localization method for mine underground based on rssi hybrid filtering algorithm. In *2017 IEEE 9th International Conference on Communication Software and Networks (ICCSN)*, pages 327–332. IEEE, 2017.

[20] Faheem Zafari, Ioannis Papapanagiotou, Michael Devetsikiotis, and Thomas Hacker. An ibeacon based proximity and indoor localization system. *arXiv preprint arXiv:1703.07876*, 2017.

[21] Faheem Zafari, Ioannis Papapanagiotou, and Thomas J Hacker. A novel bayesian filtering based algorithm for rssi-based indoor localization. In *2018 IEEE International Conference on Communications (ICC)*, pages 1–7. IEEE, 2018.

[22] K Vadivukkarasi and R Kumar. Investigations on real time rssi based outdoor target tracking using kalman filter in wireless sensor networks. *International Journal of Electrical and Computer Engineering*, 10(2):1943, 2020.

[23] Wikipedia contributors. State-space representation — Wikipedia, The Free Encyclopedia, 2020. Available at https://en.wikipedia.org/w/index.php?title=State-space_representation&oldid=941945953.

[24] The MathWorks, Inc. What are state-space models? - MATLAB & Simulink - MathWorks India, 2020. Available at https://in.mathworks.com/help/ident/ug/what-are-state-space-models.html.

[25] Gary Bishop, Greg Welch, et al. An introduction to the kalman filter. *Proc of SIGGRAPH, Course*, 8(27599-23175):41, 2001.

[26] Yali Wang, Brahim Chaib-draa, Philippe Giguere, and Christian Gagné. Nonparametric techniques for bayesian filtering. 2011.

[27] Paul A Gagniuc. *Markov chains: from theory to implementation and experimentation*. John Wiley & Sons, 2017.

[28] Zhe Chen et al. Bayesian filtering: From kalman filters to particle filters, and beyond. *Statistics*, 182(1):1–69, 2003.

[29] Rudolph Emil Kalman. A new approach to linear filtering and prediction problems. 1960.

[30] Roger Labbe. Kalman and bayesian filters in python, 2014. https://github.com/rlabbe/Kalman-and-Bayesian-Filters-in-Python.

[31] Jun Shao and Dongsheng Tu. *The jackknife and bootstrap*. Springer Science & Business Media, 2012.

[32] Adrian FM Smith and Alan E Gelfand. Bayesian statistics without tears: a sampling–resampling perspective. *The American Statistician*, 46(2):84–88, 1992.

[33] James Carpenter, Peter Clifford, and Paul Fearnhead. Improved particle filter for nonlinear problems. *IEE Proceedings-Radar, Sonar and Navigation*, 146(1):2–7, 1999.

[34] Jun S Liu and Rong Chen. Sequential monte carlo methods for dynamic systems. *Journal of the American statistical association*, 93(443):1032–1044, 1998.

[35] Nordic Semiconductor. nRF Connect for Mobile. Source code available at https://www.github.com/NordicSemiconductor/Android-nRF-Connect.

attach your review and visit log here......

attach plagiarism report here.....