International Journal of Recent Technology and Engineering (IJRTE)
ISSN: 2277-3878 (Online), Volume-9 Issue-3 September 2020

RSSI Filtering Methods Applied to Localization using Bluetooth Low Energy

Aditya Kaduskar, Omkar Vengurlekar, Varunraj Shinde

Abstract: Bluetooth Low Energy or BLE is a technology targeting mostly small-scale IoT applications including wearables and broadcasting beacons that require devices to send small amounts of data using minimal power. This paper focuses on our implementation, which is a system, designed to filter RSSI (Received Signal Strength Indicator), calculate the co-ordinates of a BLE device that is programmed as a Beacon and display the coordinates. Since RSSI is susceptible to noise and a downgrade in its reliability is unavoidable, several filtration methods have been used. The 'Kalman – Histogram' method, which incorporates the usage of a histogram of the RSSI readings along with the Kalman filter, is our own approach to tackle issues regarding noisy RSSI readings. The localization of stationary 'Assets', has been evaluated using the Trilateration algorithm: a result in mathematics which is used to locate a single point using its distance from three or more other points. The purpose of this research work is to provide a comparative result analysis of the results obtained using the aforementioned filters, indicating the effect of these filters on our localization system. As our research suggests, the 'Kalman – Histogram' filter performs better as compared to other filters and can be used in localization applications for better accuracy.

Keywords: Bluetooth Low Energy (BLE), Raspberry Pi, Localization

I. INTRODUCTION

Since the advent of satellite navigation systems such as the Global Positioning System (GPS), Galileo and GLONASS, locating the whereabouts of an entity in an outdoor, large environment has become a trivial task. This is because most entities are within the Line of Sight of the satellites. However, this task becomes a bit complicated due to factors such as the high density of the population in urban areas, causing a rare Line of Sight. Furthermore, localization [13] [15] becomes even trickier when the entities are located indoors, where there is no Line of Sight.

Revised Manuscript Received on August 12, 2020.
* Correspondence Author
Aditya Kaduskar, Department of Electronics and Telecommunication's, PVG's College of Engineering, Pune, India. E-mail: adityakaduskar@gmail.com
Omkar Vengurlekar, Department of Electronics and Telecommunication's, PVG's College of Engineering, Pune, India. E-mail: omkarvengurlekar66@ggmail.com
Varunraj Shinde, Department of Electronics and Telecommunication's, PVG's College of Engineering, Pune, India. E-mail: varunraj.shinde@gmail.com

In such cases, the ideal solution is to implement an indoor localization system based on radio networks such as Wi-Fi, Zigbee, and BLE which is proposed in [14].
The location of assets is determined using different geometrical arrangements of transmitters and receivers, and mathematical algorithms such as Trilateration which is proposed in [5] [11]. In this paper, we will be dealing with BLE [1] [4]. BLE provides an option for less usage of energy during operation as compared to its counterparts such as Zigbee and Wi-Fi. As this is a low - energy protocol, the transmitters or 'beacons' [9] can be powered by batteries continuously, for longer periods of time, ranging from months to years, even. Also, BLE is inexpensive and easier to implement as compared to Zigbee and Wi-Fi. Its ubiquitous nature makes it a much more general option.

II. THEORITICAL FRAMEWORK

A. Trilateration

There are many techniques related to Localization techniques like Unilateration, Bilateration, Trilateration. The problem with Unilateration is it does not give exact coordinates in three-dimensional space. Bilateration will give two unique solutions in two-dimensional space, whereas in three-dimensional space it still does not give exact coordinates. Given three reference points (coordinates of Raspberry Pi's), with a known distance with each of these three nodes which we will calculate using RSSI, it is possible to determine the precise coordinates of the point at which the three spheres are intersecting. In our case the BLE tag is the point. This process is known as Trilateration. It is more computationally expensive than the other two techniques

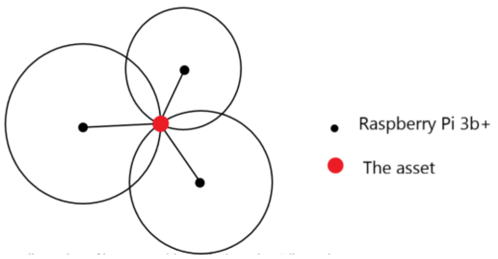

Figure 1: Trilateration

B. RSSI

Received Signal Strength Indicator (RSSI) is one of the most used characteristics for indoor localization. The main principle behind this measurement technique is it calculates the power present in a signal sent from an access point to a client device or the other way round.

Retrieval Number: 100.1/ijrte.C4413099320
DOI:10.35940/ijrte.C4413.099320
Journal Website: www.ijrte.org

Published By:
Blue Eyes Intelligence Engineering and Sciences Publication

As RF waves get attenuated due to the inverse square law, the distance can be approximated based on the relationship between transmitted and received signals strengths. This method will only work as long as no other errors are contributing to the incorrect measurements. The combination of this technique along with a propagation model can help to determine the distance between two devices.

It can be assumed that as the number of devices increases, a greater amount of information can be collected. So, the accuracy could be increased if valid RSSI values are obtained. This has a negative impact on the system. An increase in the number of devices would increase the interference between different signals. The user should consider errors while estimating the distance between the two devices. Because in wireless localization systems the key challenge is the range measurements are often associated with errors. Although, RSSI technique is the cheapest and easiest method to implement, they do not provide the best accuracy. Filtering is necessary to get rid of noise and improve systems accuracy using RSSI based localization.

C. Filtering Methods

Average filtering:

The moving average filter is a simple Low Pass FIR (Finite Impulse Response) filter commonly used for smoothing an array of sampled signals. It takes n samples of input at a time and takes the average of those n samples and produces a single output point. A moving average filter is represented by the following difference equation,

$$Y(n)= \frac{1}{Window\ Size} (x(n) + x(n-1) + \ldots + x(n - (windowSize-1))) \quad (1)$$

It is based on a rational transfer function defined by the numerator and denominator coefficients a and b respectively. Also, it considers the window size which specifies the number of samples that are to be used at a time. In MATLAB there is an inbuilt function to execute moving average of the input data,

$$\text{Avg Filter RSSI} = \text{Filter}(b, a, RSSI_1) \quad (2)$$

where, b = (1/windowSize)*ones(1,windowSize) and it represents the numerator coefficients of the rational transfer function and a represents the denominator coefficients of the rational transfer function and is taken as 1 (mentioned in the MATLAB help section). The output of the moving average filter is the smoothened version of the input that is the raw RSSI data.

Kalman filtering:

The Kalman filter is nothing but a state estimator algorithm that makes an estimate of unobserved variables considering a noisy environment(measurements). It is simply a recursive algorithm that considers the history of measurements. In the present case, a Kalman filter is used to smooth out our raw RSSI data.

Here, every step is a step from current state to the next state and the translation from a state to a measurement is assumed to be linear, i.e., it assumes linear models. The general equation of transition model is given by:

$$X_t=A_t*X_{t-1} + B_t * \mu_t + \epsilon_t \quad (3)$$

Here, X_t represents the current state and is dependent on the previous state X_{t-1} (given some transformation matrix A), a control input μ (that could be the movement of the beacon) and noise ϵ. ϵ is the process noise caused by the system itself. In our application, we assume that the position and the time frame of the measurement is static. That means, we expect a constant RSSI value and everything else is noise. So, we modify the transitional model in such a way that we completely ignore μ and set A to an identity matrix.

Thus, the equation becomes,

$$X_t \approx X_{t-1} + \epsilon t \quad (4)$$

The latter part of implementing the Kalman filter is defining the observation model, that is, what exact transition takes place between state X and the measurement Z. We model RSSI directly, i.e. our state and measurements are equal. This results in the following reduced measurement model:

$$Z_t = C_t * X_t + \delta_t \quad (5)$$

Next comes the prediction/estimation step of the Kalman filter which is nothing but updating the filter. This step describes what we expect our state to be without using any measurements. As we have assumed our system to be static, therefore,

$$\bar{\mu}_t = \mu_{t-1} \quad (6)$$

Here, μ represents our prediction/estimation. The bar above μ denotes that we still must incorporate information from the measurement. The certainty of our prediction/estimation depends on the previous certainty of our prediction and the process noise. It is expressed as:

$$\Sigma_t = \Sigma_{t-1} + \epsilon_t \quad (7)$$

So, if we are uncertain about the previous measurement, we will not be sure about the next one. In our application, we have considered a low value of process noise (0.008) because we assume that all the noise is added during measurement. Now we compute the Kalman gain using the prediction estimate. We have used a simplified formula for our application.

$$K_t=\bar{\Sigma}_t (\Sigma_t * \delta_t)^{-1} \quad (8)$$

The Kalman gain depends on the certainty of our estimate and the certainty of our measurement which is influenced by the measurement noise δ (variance of the RSSI signal).

In the final update step, the final prediction of the system is computed. This is given by,

$$\mu_t=\bar{\mu}_t + K_t(z_t-\bar{\mu}_t) \quad (9)$$

$$\Sigma_t = \bar{\Sigma}_t - (Kt*\bar{\Sigma}_t) \quad (10)$$

In MATLAB, there is a dedicated toolbox to execute the whole process, called "dsp.Kalmanfilter". The input parameters are:

1) ProcessNoiseCovariance
2) MeasurementNoiseCovariance
3) InitialStateEstimate
4) InitialErrorCovarianceEstimate
5) ControlInputPort

Kalman filter using mode estimation:

We can eliminate the uncertainty of our estimate by simply taking the mode of our raw RSSI data.

This will consider the maximum occurrences of a particular RSSI value. Thus, we will not be uncertain about the next state. Thus, the InitialStateEstimate parameter will be equal to the mode of RSSI value.

Kalman filter using Histogram approximation:

One drawback of the mode estimation is that, for a distance, the mode value is different for every iteration as it depends on the raw RSSI data which is unstable and full of undesired values. Also, the mode value can vary from the desired range. For example, if the desired range of RSSI values at 1 meter is from -57 dB to -51 dB and the number of values in this range are 210 but the number of occurrences of -43 dB are 90 times which can be more the occurrences of individual values in the desired range and this can cause significant errors. To eliminate that, we have used the Histogram approximation method. Here we consider an entire range of values where the range has a maximum number of occurrences. The redundant data is dropped out, i.e. the RSSI values out of the range. While extracting the values from a range, we usually get the mean of the extreme values and not the whole range. To tackle this problem, we have considered the values between (mean of the range – n) and (mean of the range + n) where n is the factor that helps to capture the whole range. The factor n depends on the amount of data you are using. This method reduces the difference between the maximum and the minimum RSSI value. Then the Kalman filter is applied over this range to get a smoother output.

D. Literature Survey:

The development of Real-Time Locating Systems (RTLS) has become an important add-on to many existing localization systems. While GPS has solved most of the outdoor RTLS problems, it fails to repeat this success indoors due to a lack of resolution. A number of technologies have been used to address this problem. The ability to accurately track the location of assets indoors has many applications ranging from medical, military, and logistical to entertainment. However, some systems are not capable enough to track moving targets or are not accurate enough when coverage is poor.

Kevin C. et al [11] attempt to provide such a useful comparison by providing a review of the practicalities of installing] certain location sensing systems. They also comment on the accuracy achieved and problems encountered using the position-sensing systems.

Elena S. et al [12] analyzes the fluctuations in received signal strength and positioning accuracy in indoor environments for three types of wireless area networks: Wireless Local Area Networks (WLANs) in essence: Wi-Fi at 2.4 GHz and 5 GHz frequency, respectively, and the Bluetooth Low Energy (BLE). In this paper, the similarities, and differences between WLAN and BLE signals operating at 2.4 and 5GHz have been investigated, in terms of path-loss modeling and RSSI-based positioning accuracy. The general conclusion is that the WLAN and BLE signal propagation in multi-floor buildings is very similar.

Ramsey F. et al [13] investigated the impact of Bluetooth Low Energy devices in advertising/beaconing mode on fingerprint-based indoor positioning schemes. This paper has used the fingerprinting algorithm for localization. Fingerprinting is a method that does not require the coordinates of the relaying nodes. The paper has shown that significant positioning improvement over that available from existing Wi-Fi infrastructure is possible even using a relatively sparse deployment of beacons once the characteristics of BLE signals are accounted for. Since Wi-Fi is a power-hungry technology, BLE is seemingly preferred in this case.

Hyunwook P. et al [14] propose a novel three-dimensional positioning system using Bluetooth low-energy– based beacons. Conventional methods estimate distances by obtaining two-dimensional coordinates (x,y) on the basis of received signal strength indicator value, the proposed method improves the measurement accuracy by obtaining three-dimensional coordinate values (x,y,z). In this article, the receiving rate for different signal powers has also been recorded at different distances. This paper is a reference for preliminary RSSI recording and future developments in our projects, should we consider implementing 3 - dimensional positioning instead of two - dimensional positioning.

Jinze D. et al [7] propose a received signal strength indicator (RSSI)-based parameter tracking strategy for constrained position localization is proposed. To estimate channel model parameters, the 'Least Mean Squares' method (LMS) is associated with the trilateration method. In this paper, the distance is estimated using the Median RSSI values and the trilateration algorithm is adopted to estimate the position. This tracking strategy uses grid correction and the Least Mean Squares method.

III. SYSTEM DESIGN

A. Hardware Specifications

Manufacturers such as Texas Instruments, Nordic Semiconductor, Bluegiga, make BLE Beacons. A beacon consists of a Bluetooth chipset (including its firmware), a battery providing power supply, and an antenna. We have used BLE enabled dongles made by Nordic Semiconductors, which are programmed as BLE beacons in our project. These dongles, due to their form factor can be easily placed at a certain location or embedded inside an object. They broadcast BLE radio signals which can be received and interpreted by any other BLE enabled receiver device, unlocking its application in indoor localization. To be able to perceive these beacons, it is necessary to have a device that supports Bluetooth 4.0 or higher. The devices we are using as beacons are the Nordic Semiconductor nrf52840 dongles. The nRF52840 Dongle is a small, low-cost USB dongle that supports Bluetooth 5, Bluetooth mesh, Thread, ZigBee, 802.15.4, ANT, and 2.4 GHz proprietary protocols. It is based on the nrf52840 SoC.

When programmed, a beacon simply transmits BLE packets with the payload as its identity. These packets are called advertisements and they are broadcasted to all the devices that are within its range at regular intervals of time. Beacons do not communicate with the surrounding devices using any other sophisticated way. These Advertisements contain the following data:

1) MAC Address.
2) Universally Unique Identifier (UUID) – common for a single deployment at a venue.
3) Major number- designated for dividing the beacon sets into smaller segments.
4) Minor numbers- designated for dividing the segments into smaller subsegments.

The beacon parameters such as its identity, advertising interval, and TX power can be configured in Configuration Mode. In the configuration mode, beacons use advanced bidirectional communication with a master device (e.g., a smartphone) with the aid of which they are configured. Furthermore, beacons can also be hard configured by modifying the contents of the code they are programmed with.

At the physical layer, BLE uses the industrial, scientific, and medical (ISM) band with 40 channels, each 2.0 MHz wide. 37 channels are used for data exchange among paired devices and 3 channels are designated for broadcasting advertisements. These 3 channels are thus primarily used by beacons and are chosen deliberately so that they collide with other communication channels as little as possible. The beacon broadcasts its advertisement packet repetitively based on the selected advertising interval while hopping over the 3 designated channels.

As receiving devices in our indoor localization system, we have chosen the Raspberry Pi. The Raspberry Pi is essentially an onboard microcomputer which comes with an on-chip Bluetooth 4.2 antenna. Hence, it is BLE enabled. For our research, we have used the Raspberry Pi 3 B+ model.

B. Working and Architecture

1. The BLE indoor positioning system is composed of three elements. The first element is one or several BLE beacons, who send the BLE advertisements. We are using the nrf52840 dongle manufactured by Nordic Semiconductors as the BLE beacon [13]. The second element is a set of at least three receivers. Each receiver is a Raspberry Pi 3B+ [15] Finally, a platform (server) is needed to receive the RSSI [12] obtained by each receiver to estimate the current position of the sender nodes (BLE beacons). The three receivers are connected to a console device namely a laptop using Wi-Fi. The three Raspberry Pi's and the laptop need to be connected to the same Wi-Fi network.
2. The nrf52840 dongle is a BLE enabled device that can be programmed as a BLE beacon using the proprietary SDK provided by Nordic Semiconductors. As the device is programmed as a beacon, its period of advertisement and identifier can also be modified.
3. The signals sent by the BLE beacons are received by the Raspberry Pi's and then further computation is done. In our proposed system, it is the BLE beacon that is to be located. E.g. Suppose an asset is a box of screws, the BLE beacon will be placed on the box.
4. The Raspberry Pi 3B+ [15] is a device that comes with Bluetooth 4.2; hence it is a BLE enabled device. For the purpose of trilateration, a minimum of three receivers are required, therefore, three Raspberry Pi's are used. 'Bluez' is a Linux based Bluetooth library that needs to be installed on the receivers.
5. A shell script is run on each receiver that scans for nearby BLE [[1][14] devices. The results contain several attributes such as the MAC address of the device, the Device name, Timestamp of that result, and the RSSI.
6. Several such readings are recorded after a certain time interval which corresponds to the advertisement duration of the beacon.
7. These results are dumped into a 'Comma Separated Values' (.csv) file and transferred to the console using an FTP server. This process is carried out on each receiver.
8. The console receives the RSSI measurements from the receivers using FTP. However, these measurements are quite noisy and need to be filtered according to the environment they are measured in. Here, the Kalman filter comes into the picture. Although the Kalman filter is used for dynamic measurement scenarios, it can be modified for static measurements as well.
9. All filtering and computations are done in MATLAB using a program we have designed.
10. After Kalman filtering, the filtered RSSI values are used to calculate the distance of the beacon from each receiver. Once the distances are calculated, the values obtained are used to compute the coordinates of the beacon using the Trilateration algorithm.
11. Trilateration is a localization technique that requires distances of a point from 3 fixed reference points. In our system, the beacon is the mobile point, and the receivers are the fixed points. Let us consider the RSSI values of the beacon obtained from each receiver as $RSSI_1$, $RSSI_2$, and $RSSI_3$ respectively, then the distances calculated are d_1, d_2, and d_3, respectively. Hence, a system of three equations is formed [8].
12. The algorithm uses a circle of radius d_i to compute circles from each fixed reference point. The Center of each circle represents the coordinates of the Raspberry Pi's. Ideally, the intersection of all three circles should be a single point, which is the coordinates (x,y) of the BLE tag that has been attached to the Asset in which the user is interested. The distances d_1, d_2, and d_3 are calculated from respective RSSI values using:

$$\begin{aligned} (x-x_1)^2 + (y-y_1)^2 &= d_1^2 \\ (x-x_2)^2 + (y-y_2)^2 &= d_2^2 \\ (x-x_3)^2 + (y-y_3)^2 &= d_3^2 \end{aligned} \tag{11}$$

13. The system is implemented on hardware as described in Section 2.2.1.
14. To test the performance of the designed system it has been deployed in a live area. This section describes the layout of the live area.
15. Our system is tested in a closed environment. Testing area is a Living room with a lot of wooden furniture and walls. Layout of the testing area is presented in figure 3 The red squares represent the Raspberry Pi 3B+ (node) which are placed at a calibrated distance from a reference point.

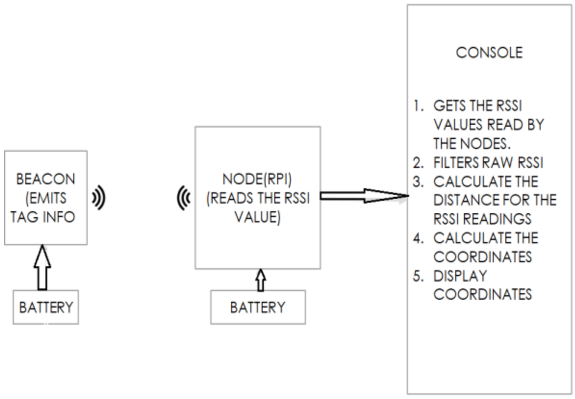

Figure 2: Block Diagram

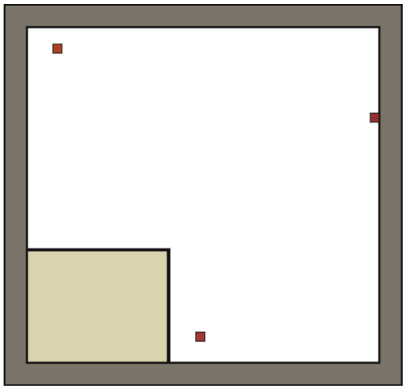

Figure 3: Testing Environment

IV. RESULTS AND CONCLUSION

The calculations in this section are dependent on the experimental estimations of the environment constant 'n' and the RSSI reading at one meter 'A_0'. Other values of n and A_0 give different results from the ones presented in Table 2.1. However, no matter how these constants are decided, RSSI readings are noisy. Therefore, an assumption can be made that similar results would be achieved with other estimations of n and A_0 as well. For our test environment the suitable 'A_0' and 'n' readings are -55.1 dBm and 4 respectively.

A. Results:

Table 1: Beacon at 1 meter

Method	Min RSSI	Max RSSI	Mean RSSI	Min Distance	Max Distance	Mean Distance
Unfiltered	-74 dBm	-32 dBm	-49.53 dBm	0.26	2.96	0.82
Averaging Filter	-61.20 dBm	-41 dBm	-49.32 dBm	0.44	1.42	0.74
Mode Estimation Method	-55.49 dBm	-48.22 dBm	-49.86 dBm	0.60	0.86	0.66
Kalman Filter	-54.54 dBm	-48.22 dBm	-49.84 dBm	0.67	0.968	0.74
Histogram Method	-56.34 dBm	-54.03 dBm	-55.77 dBm	0.94	1.07	1.03

The results presented in Table 2.1 indicate the RSSI readings obtained for placing the beacon at 1m from the receiver. The unfiltered RSSI at one-meter distance takes values between -32 dBm and -74 dBm, indicating that the environment is noisy, which makes using RSSI for distance estimation a difficult task using the equation:

$$RSSI = -10n \log (d/d_0) + A_0 \qquad (12)$$

with A_0 = -55.1 dBm and n = 4, $RSSI_1$= -32 dBm and $RSSI_2$ = -74 dBm, we get d_1 = 0.26 m and d_2 = 2.96 m, which gives a distance difference of $\Delta d = d_1 - d_2$ = 2.76 m. In this case, the noise in the RSSI measurements can cause a potential error of 276% in the distance estimation. Such a large potential error requires filtering techniques to get good accuracy in a localization application.

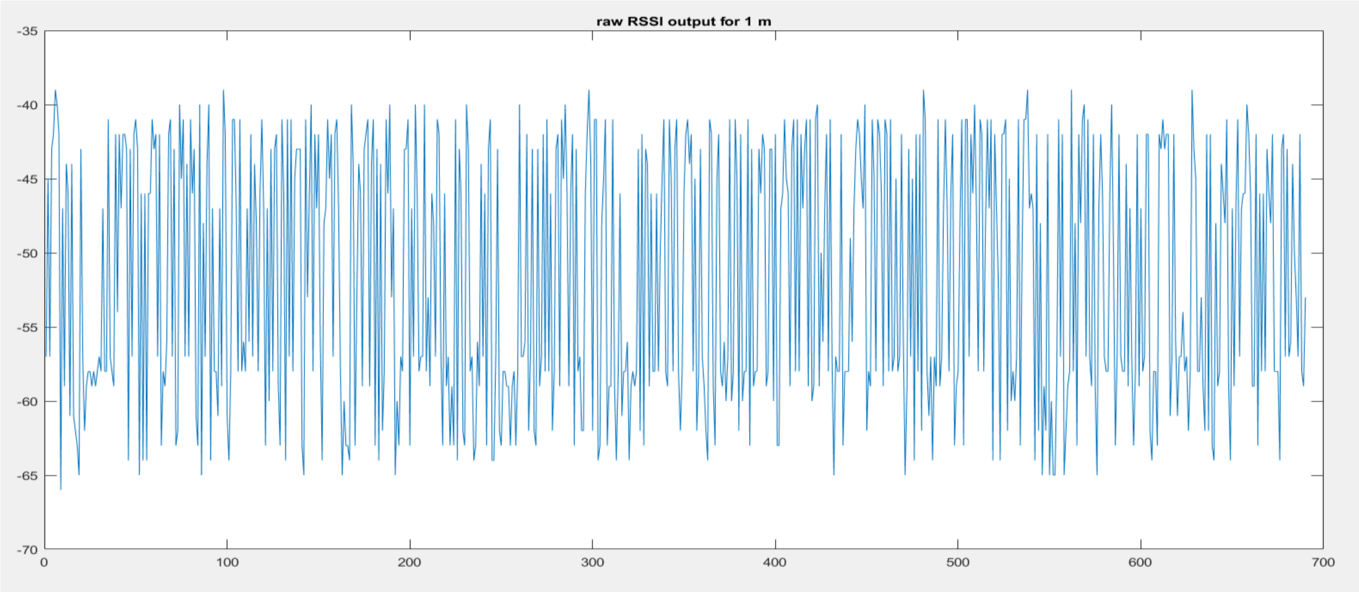

Figure 4: Unfiltered RSSI at 1m distance

After the average filter is applied, keeping the same conditions, the RSSI values for a beacon placed at one-meter distance from receiver vary from -41 dBm to -61.2 dBm. Hence, we get d_1 = 0.44 m and d_2 = 1.42 m, and $\triangle$d = 0.98 m. In this case, the potential error percentage is 98% which is a large improvement compared to the case where no filter was used.

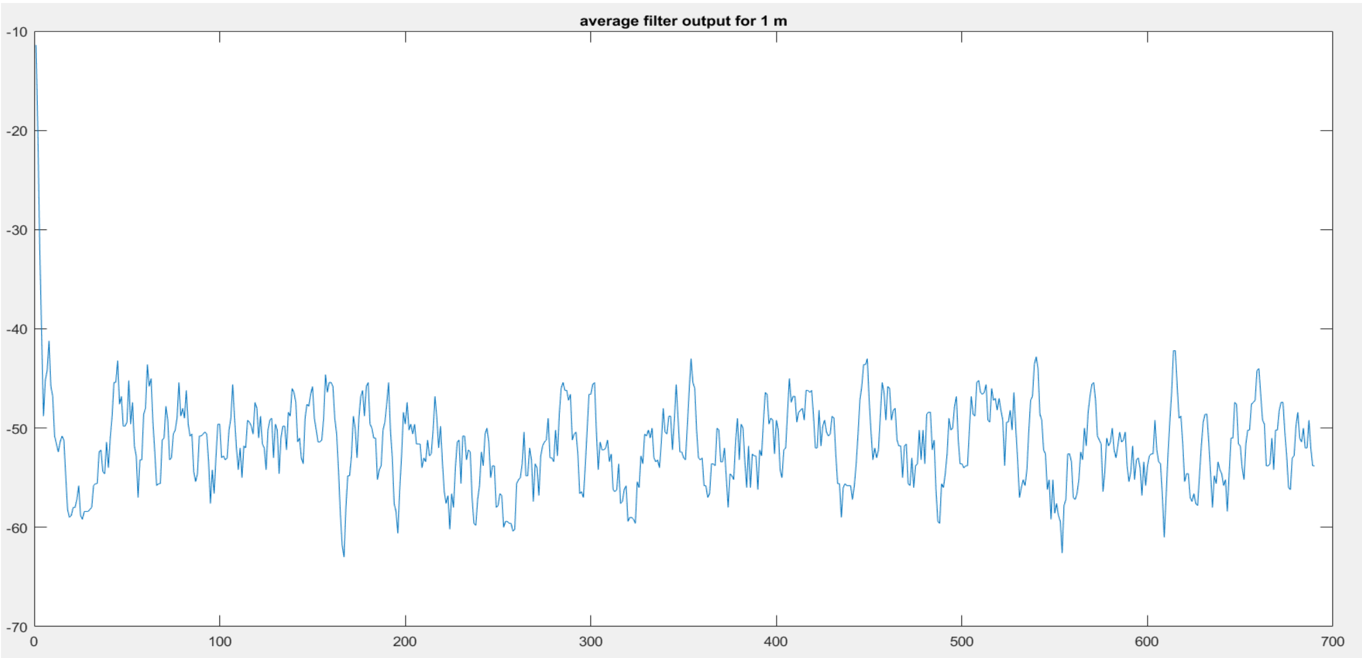

Figure 5: Average filter RSSI at 1m distance.

Using the Mode Estimation method, the range of RSSI values obtained for a beacon placed at one-meter distance from receiver is -52.56 dBm to -65 dBm. The distance d_1 = 0.89 m and d_2 = 1.76 m. $\triangle$d = 0.87 m. Therefore, the potential error is 87%. Although this is an improvement over the previous result, it is not ideal.

When we use the Kalman filter, the RSSI values obtained at one-meter range between -53.43 dBm to -68.07 dBm. The distances d_1 and d_2 are 0.9 m and 2.10 m respectively, and $\triangle$d = 1.2 m. The potential error percentage is 120%. Although the potential error indicates a high percentage, the readings approach the one-meter mark gradually.

Retrieval Number: 100.1/ijrte.C4413099320
DOI:10.35940/ijrte.C4413.099320
Journal Website: www.ijrte.org

Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication

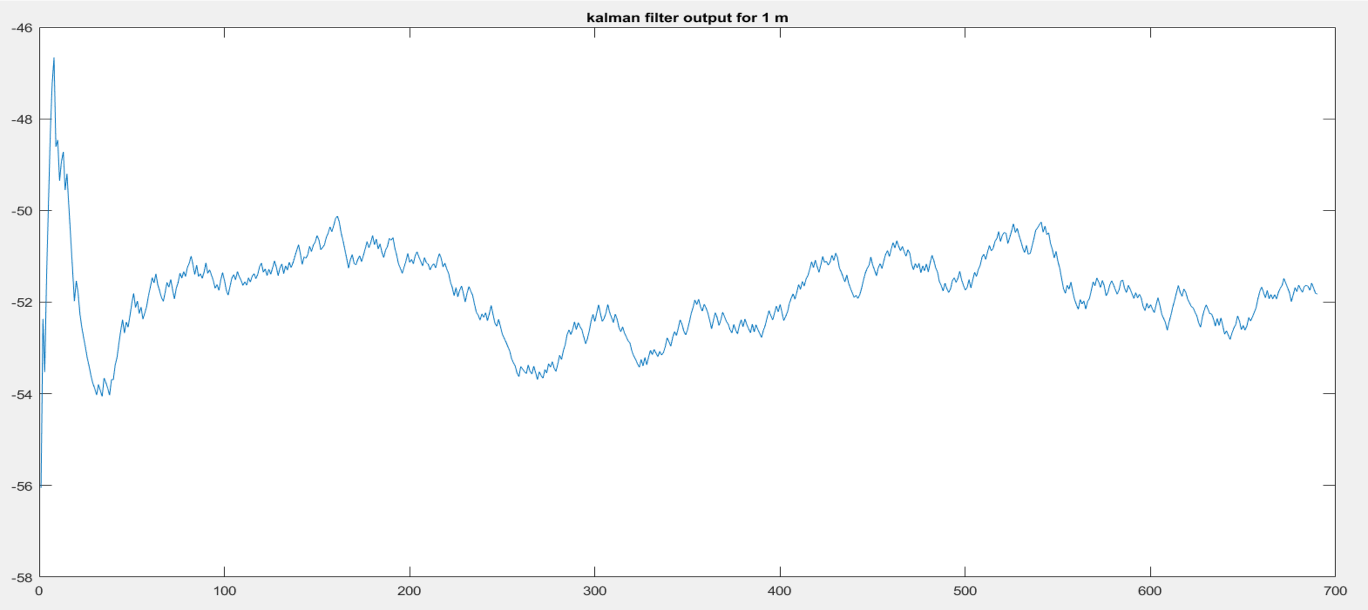

Figure 6: Kalman filter at 1m distance.

Using the Kalman – Histogram method, the RSSI values obtained lie within the range -61.62 dBm to -62.66 dBm. The distance d_1 = 1.455 m and d_2 = 1.54 m, $\triangle d$ = 0.085 m. Hence, the potential error percentage is 8.5%. In this method, a histogram of all RSSI readings post Kalman filtering is plotted, and a certain range of RSSI values having a higher frequency than other values is chosen. As a result, the noisier values are discarded, and fewer values are chosen for estimating the distance. Due to this filtering method, the error percentage drops down drastically as observed, which is expected.

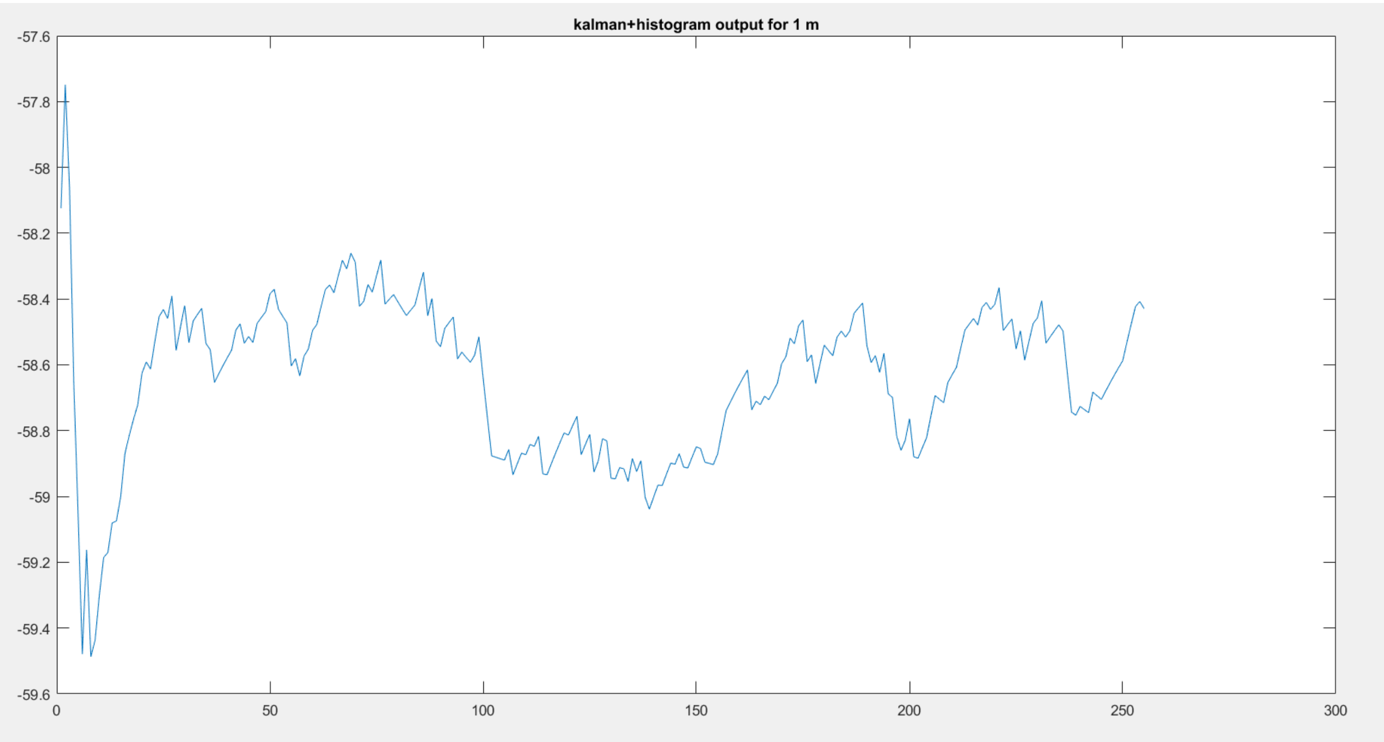

Figure 7: Histogram post Kalman filter at 1m distance.

Table 2: Beacon at 2 meters

Method	Min RSSI	Max RSSI	Mean RSSI	Min Distance	Max Distance	Mean Distance
Unfiltered	-79 dBm	-42 dBm	-56.88 dBm	0.47	3.95	1.25
Averaging Filter	-67.20 dBm	-47.40 dBm	-56.66 dBm	0.64	2	1.13
Mode Estimation Method	-65 dBm	-52.56 dBm	-56.84 dBm	0.89	1.76	1.10
Kalman Filter	-68.07 dBm	-53.43 dBm	-56.89 dBm	0.90	2.10	1.1
Histogram Method	-62.66 dBm	-61.62 dBm	-62.16 dBm	1.455	1.54	1.50

The localization algorithm implemented in our system is the Trilateration algorithm. Therefore, the analysis in this section mainly focuses on the effect of various filtering techniques on the localization accuracy. The localization accuracy is judged on the basis of the error distance. The error distance is the Euclidean distance between the True Position and the Estimated Position.

As RSSI is known to be highly unreliable for the purpose of distance estimation and localization, error distances ranging from 1 m to 1.5 m are considered to be fair results. An error distance below 1 m for a noisy environment indicates high accuracy. As the distance of the beacon from the receiver increases, error distance increases, and localization accuracy decreases.

Table 3: Localization test results using Trilateration and Unfiltered RSSI

TRUE POSITION	ESTIMATED POSITION	ERROR DISTANCE
(2.25,5.19)	(1.92,3.30)	1.91m
(5.03,5.19)	(2.36,3.32)	3.25m
(1.34,0.92)	(2.19,1.67)	1.13m
(2.34,2.74)	(1.97,2.75)	0.36m
(0.43,3.36)	(1.95,2.84)	1.61m
(2.25,3.68)	(2.09,2.96)	0.73m
(0.98,5.8)	(1.66,3.03)	2.85m

The localization test results for unfiltered RSSI from table 2.3 indicate that Unfiltered RSSI leads to a large error distance for majority of the readings, ranging from 0.36 m to 3.25 m. This suggests that further filtering is required to acquire more accuracy. Once the RSSI values are average filtered, some noticeable changes are observed.

Table 4: Localization test results using Trilateration and Average filtered RSSI

TRUE POSITION	ESTIMATED POSITION	ERROR DISTANCE
(2.25,5.19)	(1.16,4.48)	1.29m
(5.03,5.19)	(3.99,5.74)	1.17m
(1.34,0.92)	(1.19,2.89)	1.98m
(2.34,2.74)	(2.24,2.65)	0.12m

(0.43,3.36)	(1.47,2.58)	1.29m
(2.25,3.68)	(2.21,2.93)	0.74m
(0.98,5.8)	(1.26,3.34)	2.47m

According to table 4, the error distance ranges between 0.12 m and 2.47 m. This assures us of the fact that filtering RSSI values leads to better localization accuracy. Nevertheless, the frequency of readings having lesser error distance is still low. Therefore, other filtering techniques also must be implemented.

Table 5: Localization test results using Trilateration and Kalman Filter

TRUE POSITION	ESTIMATED POSITION	ERROR DISTANCE
(2.25,5.19)	(2.23,3.91)	1.27m
(5.03,5.19)	(2.41,3.46)	2.33m
(1.34,0.92)	(1.99,1.72)	1.03m
(2.34,2.74)	(1.71,2.77)	0.62m
(0.43,3.36)	(1.88,2.99)	1.50m
(2.25,3.68)	(1.14,3.30)	1.17m
(0.98,5.8)	(1.14,3.31)	2.49m

The localization test results using the Kalman Filter, from table 2.5, indicate a better performance as compared to Average filtering. The error distance ranges from 0.62m to 2.47 m. However, several observations indicate error distances within the range of 1 m to 1.5 m. This indicates that the Kalman filter normalizes the RSSI values within a certain range.

Table 6: Localization using Kalman Filter and Histogram method

TRUE POSITION	ESTIMATED POSITION	ERROR DISTANCE
(2.25,5.19)	(2.07,3.96)	1.23m
(5.03,5.19)	(3.19,4.00)	2.18m
(1.34,0.92)	(1.57,1.28)	0.42m
(2.34,2.74)	(1.72,2.59)	0.63m
(0.43,3.36)	(1.87,2.91)	1.49m
(2.25,3.68)	(1.93,3.17)	0.59m
(0.98,5.8)	(1.14,3.33)	2.47m

Table 2.6 shows us the localization test results using the Kalman – Histogram method. This method is a self-developed approach to minimize the effect of noisy readings on localization accuracy. As the table indicates, the error distances range from 0.42 m to 2.47 m. Several readings are under the 1 m error distance mark, indicating a higher accuracy than any other filter used during the testing of this system.

V. CONCLUSION

The results discussed above clearly show that filtering is beneficial for the purpose of RSSI distance estimation to remove as much noise as possible. The potential error in RSSI distance estimation increases as the distance of the beacon from the receiver increases. The Kalman filter does not always provide a better performance as compared to the Averaging filter.

The Bluetooth SIG norms [18] allow us to design our own filter for the purpose of achieving maximum accuracy as possible. Hence, a new approach in the form of the Kalman – Histogram method was taken, which has provided seemingly better results than any other filter. This research work enables us to incorporate the usage of this method in the process of indoor localization, to provide better accuracy.

ACKNOWLEDGEMENT

We are thankful to our project guide Prof. Mrs. Shilpa Litake for her timely constructive criticism and guidance without which the accomplishment of the task would have been difficult. We also thank her for renewing her faith in us and motivating us when completing this project seemed like an impossible task.

REFERENCES

1. O. Silver, "An Indoor Localization System Based on BLE Mesh Network", 2016.
2. R. Heydon, "Detailed Bluetooth Low Energy: The Developer's Handbook", 2013.
3. F. Zafari, A. Gkelias, and K. K. Leung, A Survey of Indoor Localization Systems and Technologies," IEEE Communications Surveys & Tutorials, vol. 21, no. 3, pp. 2568-2599, 2019.
4. A. Pal, "Localization Algorithms in Wireless Sensor Networks: Current Approaches and Future Challenges.", Network Protocols & Algorithms, vol. 2, no. 1, pp. 45-73, 2010.
5. P. Kriz, F. Maly, and T. Kozel, "Improving Indoor Localization Using Bluetooth Low Energy Beacons", Mobile Information Systems, vol. 2016, 2016.
6. C. M. El Amine, O. Mohamed, and B. Boualam, "The Implementation of Indoor Localization Based On An Experimental Study of RSSI Using a Wireless Sensor Network", Peer-to-Peer Networking and Applications, vol. 9, no. 4, pp. 795-808, 2016.
7. J. Du, J.-F. Diouris, and Y. Wang, "A RSSI-Based Parameter Tracking Strategy for Constrained Position Localization", EURASIP Journal on Advances in Signal Processing," vol. 2017, no. 1, pp. 1-10, 2017.
8. P. Askari and F. Barekat, "Trilateration and Bilateration in 3D and 2D Space Using Active Tags," 10 2017.
9. V. Canton Paterna, A. Calveras Auge, J. Paradells Aspas, and M. A. Perez Bullones, "A Bluetooth Low Energy Indoor Positioning System With Channel Diversity, Weighted Trilateration And Kalman Filtering," Sensors, vol. 17, no. 12, p. 2927, 2017.
10. S. Subedi, G.-R. Kwon, S. Shin, S.-s. Hwang, and J.-Y. Pyun, "Beacon Based Indoor Posi-tioning System Using Weighted Centroid Localization Approach", in 2016 Eighth International Conference on Ubiquitous and Future Networks (ICUFN), pp. 1016-1019, IEEE, 2016.
11. K. Curran, E. Furey, T. Lunney, J. Santos, D. Woods, and A. McCaughey, "An Evaluation of Indoor Location Determination Technologies", Journal of Location Based Services, vol. 5, no. 2, pp. 61-78, 2011.
12. J. Talvitie, M. Renfors, and E. S. Lohan, "A Comparison Of Received Signal Strength Statistics Between 2.4 GHz and 5 GHz Bands For Wl-Based Indoor Positioning", in 2015 IEEE globecom workshops (GC Wkshps), pp. 1-6, IEEE, 2015.
13. R. Faragher and R. Harle, "Location Fingerprinting With Bluetooth Low Energy Beacons", IEEE journal on Selected Areas in Communications, vol. 33, no. 11, pp. 2418-2428, 2015.
14. H. Park, J. Noh, and S. Cho, "Three-Dimensional Positioning System Using Bluetooth Low-Energy Beacons", International Journal of Distributed Sensor Networks, vol. 12, no. 10, p. 1550147716671720, 2016.
15. M. WENGER, J. CARRERA, and Z. ZHAO, Indoor Positioning Using Raspberry Pi With uwb", Bachelor Thesis, University of Bern, Institute of Computer Science, 2019.
16. X. Zhu, Y. Feng, et al., "RSSI-Based Algorithm For Indoor Localization", Communications and Network, vol. 5, no. 02, p. 37, 2013.
17. Google, Bluetooth SIG, "https://www.bluetooth.com/blog/proximity-and-rssi/",
18. (accessed September 21, 2015).
19. Google, Bluetooth SIG, "https://www.bluetooth.com/blog/introducing-bluetooth-mesh-networking/", (accessed July 18, 2017).

AUTHORS PROFILE

Aditya Kaduskar. Final year student at department of Electronics and Telecommunications at PVG's college of engineering, Pune, India.

Omkar Vengurlekar. Final year student at department of Electronics and Telecommunications at PVG's college of engineering, Pune, India.

Varunraj Shinde. Final year student at department of Electronics and Telecommunications at PVG's college of engineering, Pune, India.