

Modelling an Indoor Crowd Monitoring System based on RSSI-based Distance

Syifaul Fuada¹

Program Studi Sistem Telekomunikasi
Universitas Pendidikan Indonesia
Bandung, Indonesia

Prasetyo³

School of Electrical Engineering
Korea Advanced Institute of Science and Technology
Daejeon, South Korea

Trio Adiono²

School of Electrical Engineering and Informatics
Institut Teknologi Bandung
Bandung, Indonesia

Hartian Widhanto Shorful Islam⁴

Stream Intelligence Ltd.
London, United Kingdom

Abstract—This paper reports a real-time localization algorithm system that has a main function to determine the location of devices accurately. The model can locate the smartphone position passively (which do not need a set on a smartphone) as long as the Wi-Fi is turned on. The algorithm uses Intersection Density, and the Nonlinear Least Square Algorithm (NLS) method that utilizes the Lavenberg-Marquart method. To minimize the localization error, Kalman Filter (KF) is used. The algorithm is computed under Matlab approach. The most obtained model will be implemented in this Wi-Fi tracker system using RSSI-based distance for indoor crowd monitoring. According to the experiment result, KF can improve Hit ratio of 81.15 %. Hit ratio is predicting results of a location that is less than 5 m from the actual area (location). It can be obtained from several RSSI scans, the calculation is as follows: the number of non-error results divided by the number of RSSI scans and multiplied by 100%.

Keywords—Wi-Fi tracker system; RSSI-based distance; intersection density method; Nonlinear Least Square (NLS) method; Kalman Filter (KF)

I. INTRODUCTION

In 4.0 era, crowd monitoring/tracking system has become very useful application because it provides some summaries and insights about flow, direction, density and activity of people in certain public and private areas. A prior work described the use of a real-time Wi-Fi tracking system for business intelligence in the retail company [1]. The methods can be used in the crowd monitoring system, e.g., image-based and non-image-based method. Generally, image or video-based system requires high-cost and complex computation. Also, that approach has several other disadvantages: it only covers a small line-of-sight (LoS) area and difficult to obtain high estimation accuracy when overlap and occlusion exist in the crowd. In other hands, the video-based method does not work in dark or smoke environments and also less privacy of the target [2]. The non-image-based method can overcome drawbacks in image-based methods, especially in the cost factor, and it can cover a high LoS area.

Nowadays, most people bring their smartphones everywhere they go. The Wi-Fi access point facilities have also been installed in many places; it will trigger most people to turn on their Wi-Fi on smartphones. We can use this fact to track-down them using a Wi-Fi-based approach. The Wi-Fi devices on a smartphone will reveal their MAC address data through probe-request data whenever the Wi-Fi devices on. In other words, when the device activates Wi-Fi, they will eventually broadcast probe request data containing useful information; for example, MAC addresses data of device and time-stamp [3-5]. Because the MAC address is a unique identifier, it can be used to identify the presence of people in a particular location. In this work, we used Wi-Fi-based RSSI localization to estimate or monitor the crowd pattern at certain places. A similar approach has been proposed in work at [6] and [7].

Radio Frequency (RF) technologies such as GPS, RFID, Bluetooth, and Wi-Fi, use radio signals to find the device's location. Generally, we put the sensor (node) to sense the RF signal parameter then estimate the location from that parameter. Some of RF parameter that has been used by researchers: Time difference of Arrival [8], Time of Arrival [9], Angle of Arrival [10], and RSSI [11]. The RSSI-based localization provides a simpler node compared to the other methods. But this approach gives a major problem to the detection accuracy due to the high variance on RSSI value, the techniques to overcome it, e.g., Bayes Filter [12], Particle Filter, and so on.

There is a various method that has been proposed in the Wi-Fi RSSI-based indoor localization, like Finger-printing, Distance-based localization, etc. This work will provide GUI to monitor the indoor crowd, based on the RSSI localization method. However, the biggest problem in the real application is, the RSSI value measured at the node is unstable. They keep changing dynamically because of the presence of noise. The distribution of RSSI value most likely to be Gaussian. Therefore, correction values by using the Kalman Filter (KF) is made.

In this work, the set of RSSI data is scanned in multiple sensors (nodes). One of two recommended localization algorithms: Intersection Density and Non-Linear Least Square (NLS) will be used in this experiment. The localization algorithm is used to estimate the location of each device precisely. Whereas the KF algorithm is used to overcome the noises in RSSI signals that are scanned in sensor.

This paper is composed by four sections: 1) Introduction that discusses why RSSI-based localization method is selected for Wi-Fi tracking application, 2) Methods discusses proposed system and the used algorithms, 3) Results and Analysis, and the last section is 4) Conclusion.

II. METHODS

A. Proposed Systems

Fig. 1 shows a proposed architecture of the Wi-Fi Tracker system. It consists of two main parts: the node system and the server system. The node system has a primary function to sniff the devices (smartphones) data on the site. Then the server will compute and analyze the data and make a summary about the situation on the dashboard. As informed by [3-5], when the devices activate their Wi-Fi, they will broadcast wireless signature data containing unique information about the devices, which is a MAC address. According to our previous experiment, as in [5], the packet request data is Wi-Fi's packet data associated with it. From that data, we also got the information besides MAC address data: Timestamp and RSSI.

In the first step, we placed the node at several points on the site. Afterward, the devices on the sites will broadcast the packet request data to the surrounding access point (AP). The smartphone broadcasts the packet request data when its Wi-Fi is turned on, and they do not have to be connected with Node.

In other cases, the smartphone can be connected with the surrounding AP. As long as they broadcast the packet request data, the system will continue work. Every node on the sites will sniff the packet data emitted by the devices. The node must be in monitoring mode to sniff those packet data. Then the nodes will send the data to the server.

Before the data sent to the server, the data will be encrypted using TLS/SSL with 1024-RSA encryption. The nodes will connect to our proposed access point (Wi-Fi tracker access point) and send the data to the server using Message Queuing Telemetry Transport (MQTT) protocol.

The server will collect the data from all nodes by subscribing to the MQTT broker based on the designed topic. The server will decrypt the packet data, then collect the data and organize them based on the MAC address information. Therefore, in the server, we will get a set of MAC address and their corresponding RSSI in every node. From this set of data, we will compute the location of each device using the algorithm (NLS and KF algorithms). The raw data and processed data (smartphone and its location) will be stored in a database. In our system, we used MongoDB as a database system. We will also provide the dashboard based on a web application that runs on our server. Beside serve basic configuration to the system, the web will also provide the information (analyzed data).

But in this work focuses on the algorithm part. RSSI data is collected and then computed through Matlab simulation. The most optimum algorithm is then implemented in a server using the Python script (further research). Thus, our system will have a high-accuracy in tracking the devices. Three approaches: Intersection Density, NLS, and linear KF are computed.

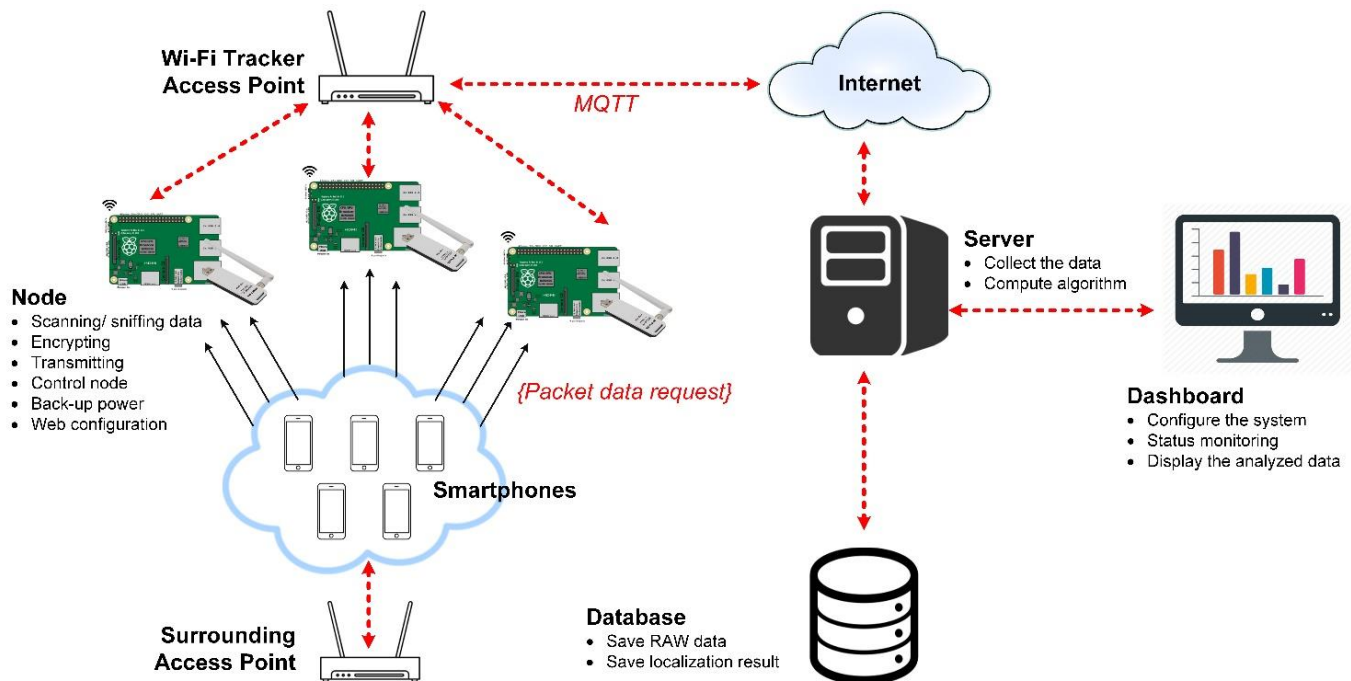


Fig. 1. Wi-Fi Tracker System Architecture.

B. Localization Algorithm

A node network with a known location is expected to get the RSSI value from the devices. Then from this set of RSSI measurements, the device location can be predicted by estimating the distance between the smartphone and the node. First, we estimate the distance between nodes and devices using the path-loss model and RSSI values. We can formulate a mathematical representation of our problem as follows:

- Given m known nodes, each is located at (x_i, y_i) $i = 1, 2, \dots, m$. The RSSI value measured at node i -th are P_i $i = 1, 2, \dots, m$
- From the data set of $\{(x_i, y_i, P_i)\}$ then we estimate the location of devices (x, y)

There are two general steps of RSSI-based distance estimation, first is Distance estimation, by using path-loss model, the value of P_i is used to estimate the distance between the smartphone and node i -th (d_i), and the second one is (b) Location estimation. From the set of measurable distance $\{d_i\}$, the smartphone location (x, y) is estimated. This step can be done using localization algorithm, such as Triangulation, Trilateration, Intersection Density, Linear Least Square, NLS, etc. In this work, we have tried two methods: Intersection Density and NLS. We compare them which is the best one to be used as the localization algorithm.

- Path-Loss model

Path-loss is a reduction in power density of the electromagnetic waves as it propagates through space (attenuation). It represents signal level attenuation caused by free-space propagation, reflection, diffraction, absorption, and scattering. There are various path-loss models that have been established to represent Wi-Fi communication as well as the condition in the room. In this work, we used the log-distance path-loss model as in Eq.1,

$$P(d) = P(d_0) - 10 \alpha \log \frac{d}{d_0} + X_\sigma \tag{1}$$

Where $P(d)$ is power at distance d , $P(d_0)$ is reference power that emitted by smartphone (power at distance $d_0 = 1m$), α is Path-loss exponent (depending on the surrounding environment, related to attenuation factor). Later, X_σ is Zero-mean, and σ -variance is a random variable (from noise, shadowing, multi-path effect).

The path-loss exponent value depends on the surrounding environment, thus to get the precise value of this parameter, the calibration must be performed. The path-loss exponent value can be determined by measuring the RSSI value for several minutes at a specific distance in the room. From that data, by using the path-loss model in Eq. 1, the value of α can be computed. Table I shows path-loss exponent.

The reference power value emitted by a smartphone has different from other smartphones. Due to our application is intended to detect many smartphone types at once, this parameter value cannot be determined by doing a calibration only. Hence, this is another point that we have to consider when performing the localization algorithm.

TABLE. I. PATH-LOSS EXPONENT VALUE

Environment	Path-Loss Exponent
Free Space	2
Urban area cellular radio	2.7 - 3.5
Shadowed urban cellular radio	3.0 - 3.5
In building LOS	1.6 to 1.8
Obstructed in building	4 to 6
Obstructed in factories	2 to 3

When a Wi-Fi signal encounters another medium with different electrical properties, there is partly reflected signal and partly absorbed signal. The reflection coefficient is a complex function of the material properties and generally depends on signal frequency, polarization, and angle of incidence. The previous model (Eq. 1) is suitable for Line-of-Sight (LOS) indoor environment. For the Non-Line-of-Sight (NLOS) environment, there is a more precise model that considers the attenuation factor from the existing obstacles. The multi-wall model can be expressed in Eq. 2,

$$P(d) = P(d_0) - 10 \alpha \log \frac{d}{d_0} + X_\sigma + \sum_{i=1}^k A_i \tag{2}$$

Where k is a number of obstacles between transmitter and receiver and A_i is an attenuation factor for obstacle i -th. Several examples of attenuation factor value for 2.4 GHz Wi-Fi signal in various materials can be seen in Table II.

Actually, in our application, this factor can only be controlled by setting the path-loss exponent value through a calibration in the room. Several precise path-loss models can represent the actual situation in the real case. Still, we use a simple path-loss model for this work. In a further improvement, it can be used for the more precise model that has higher accuracy to represent path-loss in the room.

- Intersection Density

The Intersection Density method estimates the smartphone location by utilizing pairs of known node locations to generate circles. By using multiple different pairs of the node, multiple circles can be derived, each of which intersects at smartphone location in the absence of noise and measurement errors. Of course, noise and measurement errors always exist in our measurements. This fact causes the intersection point does not intersect in a single point. However, Intersection Density assumes that a number of the intersection will be the highest in the surrounding of a smartphone location. Therefore, the smartphone location will be determined in the area that has the most intersections.

TABLE. II. ATTENUATION FACTOR FROM SEVERAL OBSTACLES

Obstacle	Attenuation factor
Concrete wall 10 cm	-16 dBm
Concrete wall 20 cm	-29 dBm
Metal	-26 dBm
Glass wall	-6 dBm
Wooden wall	-4 dBm
Human body	-3 dBm

First, from the log-distance model, by assuming that the surrounding has the same path exponent value for all directions, then we can obtain power difference as Eq. 3,

$$P_{ij} = P_i - P_j = 10 \alpha \log \frac{d_j}{d_i} \quad (3)$$

Next step we define and measure the distance ratio for all known nodes as expressed by Eq. 4,

$$d_{ij} = \frac{d_i}{d_j} = 10^{\frac{P_j - P_i}{10\alpha}} \quad (4)$$

The Intersection Density algorithm will map the set of node location $\{(x_i, y_i)\}$ and distance ratio $\{d_{ij}\}$ into set of circle with center $\{C_{ij}\}$ and radius $\{r_{ij}\}$ as Eq. 5 and Eq. 6 respectively,

$$C_{ij} = \left(\frac{x_j d_{ij}^2 - x_i}{d_{ij}^2 - 1}, \frac{y_j d_{ij}^2 - y_i}{d_{ij}^2 - 1} \right) \quad (5)$$

$$r_{ij} = \sqrt{\left(\frac{x_j d_{ij}^2 - x_i}{d_{ij}^2 - 1} \right)^2 + \left(\frac{y_j d_{ij}^2 - y_i}{d_{ij}^2 - 1} \right)^2 - \frac{d_{ij}^2 x_j^2 + d_{ij}^2 y_j^2 - x_i^2 - y_i^2}{d_{ij}^2 - 1}} \quad (6)$$

The next step is to find location where the circles intersect each other. This can be done by generate circle equation, then find the location of the intersection by solving the equation for each pair of circles. Then we divide the map into several grid areas. Later, the smartphone location (x, y) can be estimated in the location that has the most intersection points [13]. For example, in Fig. 2, the location of the smartphone can be estimated in the grid with the x mark where on the grid there are the highest intersection point.

- Non-Linear Least Square Algorithm (NLS)

This method finds the smartphone location by forming the objective function that represents the mean square error between the measurement and model. Then by using optimization function, we estimate the smartphone location (x, y) that minimizes our objection function. Because our objective function is nonlinear with the respects of the variable $(x$ and $y)$, then we called it NLS. From the path-loss model, we can calculate the power difference as Eq. 7,

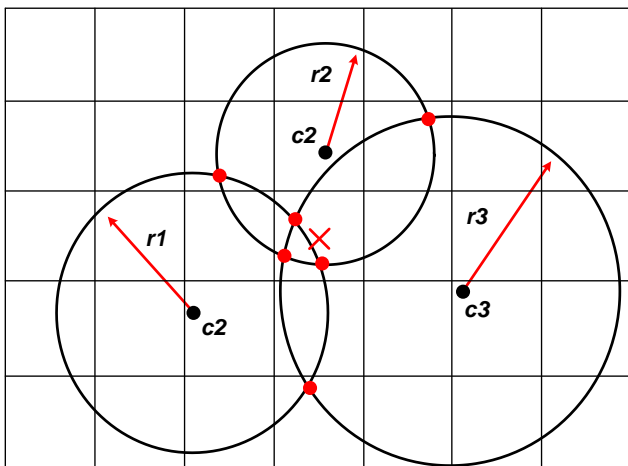


Fig. 2. Illustration of Intersection Density Method.

$$P_{ij} = P_i - P_j = 10 \alpha \log \frac{d_j}{d_i} = 10 \alpha \log \frac{\sqrt{(x-x_j)^2 + (y-y_j)^2}}{\sqrt{(x-x_i)^2 + (y-y_i)^2}} = 5\alpha \log \frac{(x-x_j)^2 + (y-y_j)^2}{(x-x_i)^2 + (y-y_i)^2} \quad (7)$$

This power difference equation is used because we want to eliminate the power reference that emitted by smartphone, which are unknown for us. Still, this can be done by assuming that path-loss exponent is equal in all rooms. From the measured RSSI in nodes, we can compute the measured power different \overline{P}_{ij} . Afterward, we correct the results using KF and then define objective function as the sum of squares of differences between the measured value and theoretical value as Eq. 8,

$$Q(x, y) = \sum_{i < j} \left[\overline{P}_{ij} - 5\alpha \log \frac{(x-x_j)^2 + (y-y_j)^2}{(x-x_i)^2 + (y-y_i)^2} \right]^2, \quad 1 \leq i < j \leq m \quad (8)$$

NLS algorithm finds the value of (x, y) that minimizes the objective function of $Q(x, y)$. Some common methods that can be used to solve the NLS problem, i.e., Gradient descent method, Gauss-Newton method, and Levenberg-Marquardt method. But, in this work, to find the solution, we used the Lavenberg-Marquardt method.

The objective function $Q(x, y)$ is nonlinear. In case, we can linearize the objective function using the Taylor series expansion; this method transforms the NLS into Linear Least Square method. We did not try it yet, however we think that the nonlinear model is more suitable for our systems, so we will use NLS instead. Eq. 9 shows objective function that consider initial power device,

$$Q(x, y) = \sum_{1 \leq j \leq m} [P_{init} - 5\alpha \log(x - x_j)^2 + (y - y_j)^2 - \overline{P}_j]^2, \quad -50 \text{ dBm} < P_{init} < -30 \text{ dBm} \quad (9)$$

C. Kalman Filter

If we observe the RSSI value that is scanned in node, we will find that the value is continuously changing even for the smartphone placed in the same location. There are so many factors that cause it. But in the real case, that is what actually happened; it will decrease the accuracy of our algorithm. To suppress this problem, we used KF to reduce the noise that happens in the node when performing a measurement.

KF works well for the systems which are continuously changing; it can predict uncertain information about a dynamic system, what the system is going to next, and its value. This filter has some advantages: it requires less memory (only needs a previous state value other than the whole history) and has fast computation (suitable for the real-time applications).

Generally, KF has two steps, which is the prediction step and the correction step. In the prediction step, KF makes a prediction based on the previous state. Then in the correction step, KF will correct the prediction value with regard to the measurement in this state. The general KF problem is stated as Eq. 10 and Eq. 11,

$$X_{k+1} = f(X_k) + w_k \quad (10)$$

$$Z_k = h(X_k) + v_k \quad (11)$$

Where X is system state vector, $f(.)$ is transition function, Z is measurement vector, $h(.)$ is measurement function, w is process noise, and v is measurement noise. Both w and v are zero mean Gaussian distribution with covariance Q and R , respectively.

D. Linear Kalman Filter Calculation

In Linear KF, both the transition function $f(.)$ and the measurement function $h(.)$ are linear functions. We can divide into two steps as follows:

- Prediction step: In this step, we have to predict the value based on the previous state. Project the state ahead using Eq. 12, while project the covariance matrix using Eq. 13,

$$X_{k|k-1} = F * X_{k-1} \quad (12)$$

$$P_{k|k-1} = F * P_{k-1} * F^T + Q \quad (13)$$

Where F is a transition matrix model, and the covariance matrix P represents the level of certainty of our prediction value.

- Correction step: In this step, we correct our prediction based on the measurement result at this time. First, we have to compute the Kalman gain as expressed by Eq. 14,

$$K_k = P_{k|k-1} * H^T * (H * P_{k|k-1} * H^T + R)^{-1} \quad (14)$$

Then, correct the prediction based on the measurement as Eq. 15,

$$X_k = X_{k|k-1} + K_k * (Z_k - H * X_{k|k-1}) \quad (15)$$

Additionally, we have to update the covariance matrix as Eq. 16,

$$P_k = (I - K_k * H) * P_{k|k-1} \quad (16)$$

Where H is observation matrix model.

In this application, each time step, we collect the RSSI value from all nodes then uses KF for each before using them for our algorithm. Fig. 3 illustrates how the KF is used in this work; the KF works for each node. Each computation requires the previous step value so that each time the calculation has been made. The value must be stored in memory. We only need previous value, so the other values can be removed after the computation has been done. KF block diagram can be seen in Fig. 4.

Afterward, we define our state model which represent the RSSI value and the velocity of smartphone movement. This variable is intended to predict the RSSI when the smartphone is moved or not. We define our state vector as two variables: RSSI – velocity \dot{x} (Eq. 17), and observation vector as the result of scanned RSSI from node (Eq. 18),

$$X = \begin{bmatrix} RSSI \\ \dot{x} \end{bmatrix} \quad (17)$$

$$Z = [RSSI] \quad (18)$$

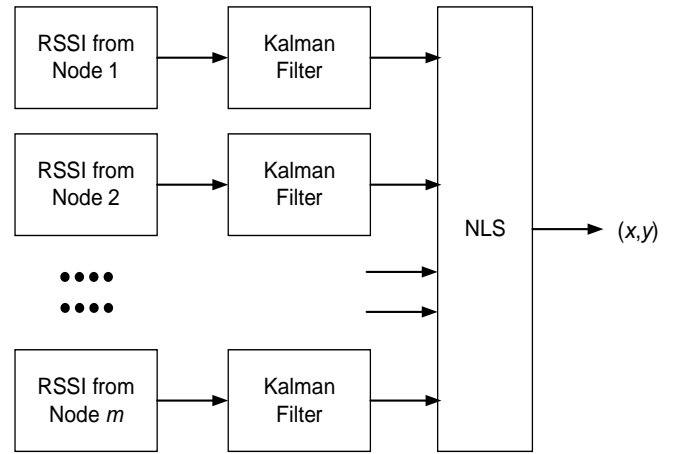


Fig. 3. Algorithm Architecture.

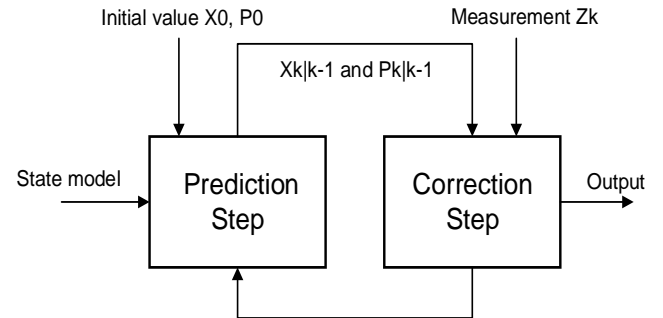


Fig. 4. Steps of KF.

Then KF model used in this work can be expressed as Eq. 19 to Eq. 22, where dt is time interval between the current step and the previous step.

$$F = \begin{bmatrix} 1 & -8 * dt \\ 0 & 1 \end{bmatrix} \quad (19)$$

$$H = [1 \ 0] \quad (20)$$

$$Q = \sigma^2 \begin{bmatrix} dt^3/3 & dt^2/2 \\ dt^2/2 & dt \end{bmatrix} \quad (21)$$

$$R = \left[\left(\frac{3.0 * RSSI + 340}{70} \right)^2 \right] \quad (22)$$

III. RESULTS AND DISCUSSION

A. Path-Loss Simulation

We first examine the path-loss exponent value in the room to get a suitable path-loss model. We measure the RSSI value based on two parameters, i.e., several times against several known distances in the room. Fig. 5 visualizes a sample of obtained RSSI from the observed device. The measurement method is elaborated in [5].

By using the path-loss model as Eq. 1, we can compute the path-loss exponent value for each measurement as shown in Fig. 6. We can estimate the path-loss exponent value of the room to be 2.4 – 2.7. We compute it in the room based on the selected room reference that is Research and Community Service (CRCS) Institut Teknologi Bandung building 1st floor.

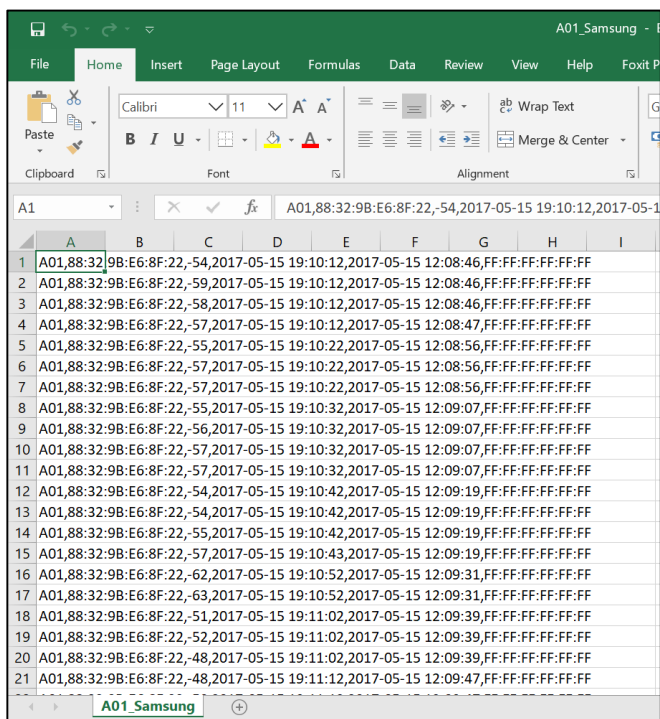


Fig. 5. Example of RSSI Data Logger from “X Smartphone”.

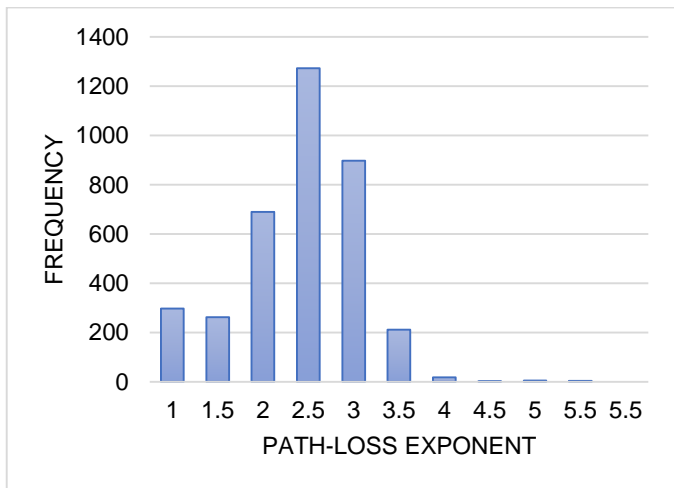


Fig. 6. Path-Loss Exponent Distribution.

B. Intersection Density Simulation

In second step, we measure the Intersection Density algorithm performance. At this simulation test, we take non-real-time data from each node. We get RSSI data for 5 minutes for each location, as illustrated in Fig. 5, then we take its average to represent the scanned RSSI value on each node. This data will be used in the Intersection Density algorithm. The computation is done using Matlab (offline computation). Suppose “X smartphone” and “Y smartphone” as detected devices.

Fig. 7 illustrates several results of the Intersection Density algorithm. The real-location smartphone is located in a green mark. The estimated location lays in the area that has the highest intersection point.

When the scanned RSSI does not represent the real power, the intersection point will be parted to each other. In this case, it is difficult to estimate the location because the intersection area will be “large”. Then if we used the real-time data, most of them do not intersect in our valid area. Hence, we assumed that this algorithm has low accuracy for real-time data, and we decide to try to use different methods, i.e., using NLS approach.

C. NLS Simulation

In this test, we measure the performance in our algorithm. We use real-time data that are measured in our server, although the computation is done in offline mode. Fig. 8 depicts the simulation result of data collected by our server for 5 minutes. The “X smartphone” is placed in the center of the room (pointed by a red mark). And four nodes are placed in the corner of the room (points A, B, C, and D). The blue circle represents the estimated point of the smartphone location.

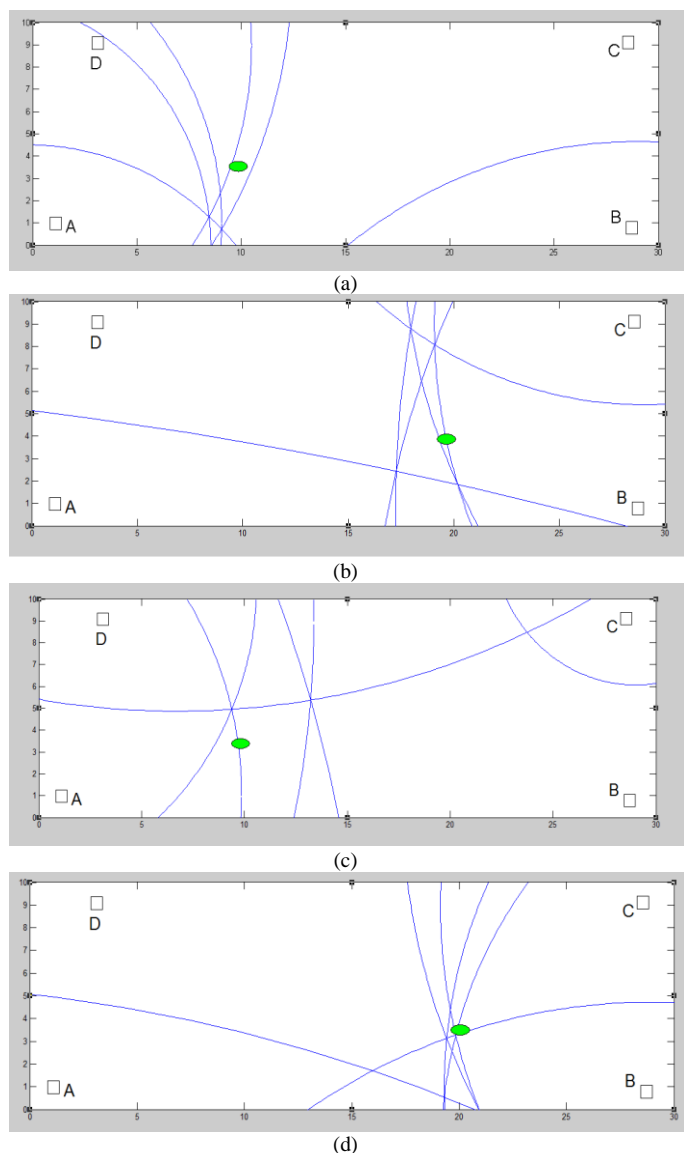


Fig. 7. Intersection Density Result: (a) X Smartphone Location I; (b) X Smartphone Location II; (c) Y Smartphone Location I; (d) Y Smartphone Location II.

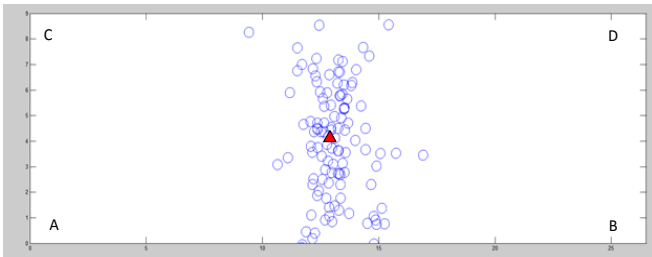


Fig. 8. NLS Simulation Result (X Smartphone) –Location I, using Eq. 8 Approach.

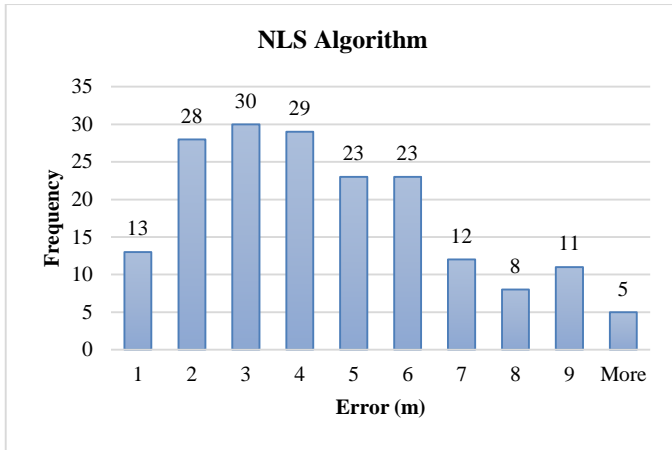


Fig. 9. Error Measurement of NLS Algorithm–Location I, using Eq. 8 Approach.

Then we measure the error by comparing the result as obtained in Fig. 8 to the real-location. If we define the hit ratio as the number when the estimated location has error 5 m or less in several periods, then this measurement has hit ratio = 67.5 %. Fig. 9 shows the obtained graph that represents the error calculation. For NLS simulation, we only take Fig. 7(a) as a sample.

In the latest observations (real-time test), we found a problem using the above algorithm (Eq. 8). Some devices that are placed outside the room, can also be detected as inside the room. After making some observations and analysis, we found what caused it. In Eq. 8, we tried to eliminate the initial power value that smartphone transmits. We did this because initial power is not a fixed variable that different smartphone gives different value. Let’s say that our sensor A, B, C, and D read RSSI value from a smartphone of -50 dBm, -50 dBm, -51 dBm, and -51 dBm, respectively. Then we have the same smartphone, but we place it in different locations, and our sensor A, B, C, and D read RSSI of -60 dBm, -60 dBm, -61 dBm, and -61 dBm respectively. The above algorithm (Eq. 8) will predict the same location for both conditions; this is because they have the same power different.

To overcome this problem, we involve additional cases when using the NLS algorithm. First, we use the objective function as Eq. 9, which considers the initial power of the smartphone to decide whether a smartphone is placed inside or not. We use several initial power values and decide which one is the most optimum by looking at cost function (residual error from objective function in the solution), from that we decide

which smartphone is inside or outside the room. Then if the device is inside, we use the previous objective function (Eq. 8) to get a better estimate location by eliminating initial power value.

D. KF Simulation

As in previous step, we first measure the RSSI value of a node, then we use the KF. Lastly, we integrate it with the NLS algorithm. Fig. 10 visualizes the result comparison of raw data before and after filtering. The data is captured for a specific time, and the interest-device (we only interested in one device) which is moving around the node. The filtering technique is used to remove ‘random’ signal strength, which captured in a single node. As we observe, raw signal strength (blue-colored line) can be filtered out using KF (red-colored line).

In line to Fig. 10, we can conclude that KF is able to overcome the ‘random’ signal strength. Later, we apply KF in each node. We compare the result of the algorithm before and after applying the KF as shown in Fig. 11. It illustrates the room with a dimension of 30 m x 10 m (the CRCS ITB room size). While A, B, C, and D point represents node system and placed on each corner. The smartphone is placed in the center of room (red mark). The blue circle indicates the result of the algorithm for several times. Later, we analyze the error of the predicted location with the actual location of the smartphone. Fig. 12 depicts the analysis result. In the accuracy of 5 m, we have a hit ratio of 68.115 %.

Then for the same data, we use KF and the result of the algorithm is shown in Fig. 13, whereas the obtained graph is shown in Fig. 14. Using a filter, we have a hit ratio of 81.15 %. According to the simulation result, we can summarize that KF can improve the hit ratio from 68.115% to 81.15 %.

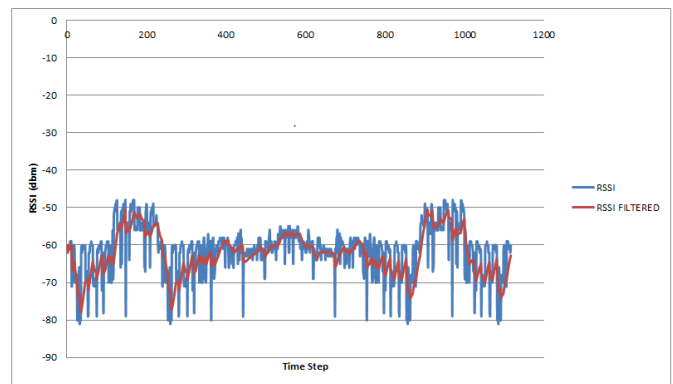


Fig. 10. The Scanned RSSI in the Node before and after using KF.

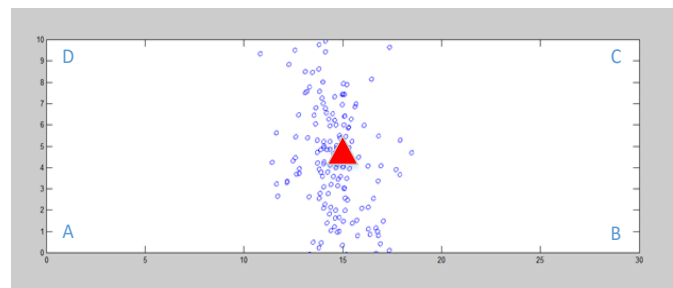


Fig. 11. The Predicted Location without KF, using Eq. 9 Approach.

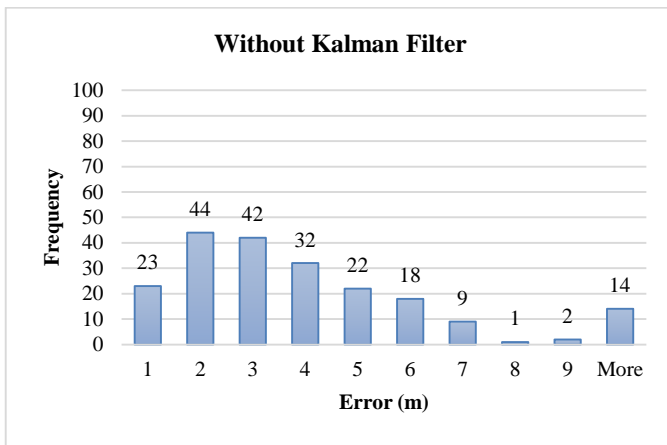


Fig. 12. Error Analysis of the Algorithm without KF, using Eq. 9 Approach.

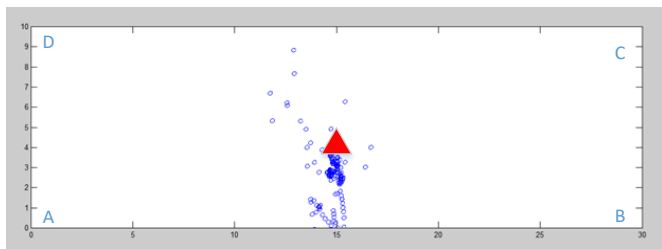


Fig. 13. The Predicted Location using KF.

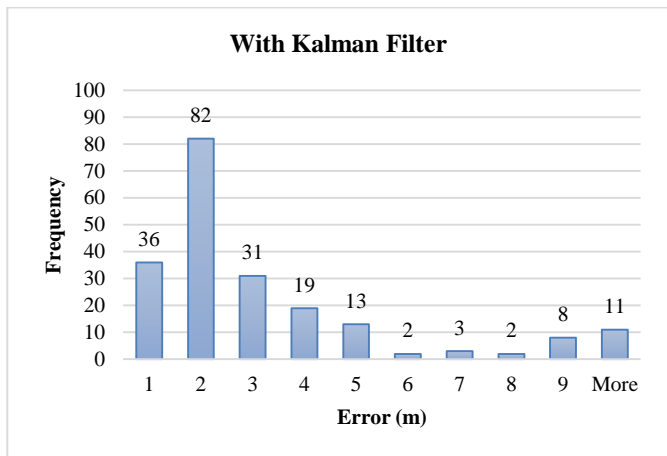


Fig. 14. Error Analysis of the Algorithm with KF.

Based on the obtained information, it is confirmed that the NLS method has a better accuracy compared to the Intersection Density method in a real-time case, and KF can improve the accuracy of the NLS almost 13%. We predict that later, UKF will perform better performance than KF. The most optimum algorithm will be implemented in the RSSI-based Wi-Fi tracking application for indoor environment, such as a system presented by recent works: G. Pipelidis, et al. [14], and Fernandez, et al. [15]. But, our system will has a complete features compared to [14-15].

IV. CONCLUSION AND FUTURE WORKS

The localization algorithm for the Wi-Fi tracker system has been modeled in this paper. It has the primary function to determine the location of devices (e.g., smartphone, laptop, tablet, etc.) in an indoor environment based on the information which is collected in the server. Two localization algorithm using distance-based methods (i.e., Intersection Density and NLS algorithms) have been tried under a Matlab simulation. The Intersection Density algorithm performs well in non-real time data (with error within 3 – 5 m). But it has a low-accuracy for real-time data (with error of >7 m or most intersection lies in outside valid area), while the NLS algorithm performs better than Intersection Density; it has the hit ratio (55 – 70%) for the real-time data. The use of KF can improve the hit ratio of 81.15 %.

According to the experiment, we assume that our algorithm still has a low-accuracy for the real-time data case; it is because of the RSSI instability. In the next work, we will try to overcome this problem by using the Unscented Kalman Filter (UKF) to pre-process the RSSI. We can put the UKF algorithm after the localization result to get better results.

REFERENCES

- [1] V. Vinhas, "Business Intelligence Based on A Wi-Fi Real Time Positioning Engine A Practical Application In A Major Retail Company," Proc. of the 11th Int. Conf. on Enterprise Information Systems, 2009.
- [2] Y. Yuan, et al., "Crowd Monitoring Using Mobile Phones," Proc. of the 6th Int. Conf. on Intelligent Human-Machine System and Cybernetics, 2014.
- [3] A.B.M. Musa and J. Eriksson, "Tracking Unmodified Smartphones using Wi-Fi Monitors," Proc. of the 10th ACM Conf. on Embedded Network Sensor Systems, pp. 281-294, November 2012.
- [4] M. Cunche, "I know your MAC address: targeted tracking of individual using Wi-Fi," J. of Computer Virology and Hacking Techniques, Springer, 2013.
- [5] S. Fuada, et al., "Your MAC Address can be detected easily when connected to the Wi-Fi," Unpublished.
- [6] J. Weppner, et al., "Monitoring Crowd Condition in Public Spaces by Tracking Mobile Consumer Devices with WiFi Interface," Proc. of the UbiComp, pp. 1363-1371, 2016.
- [7] Y. Fukuzaki, et al., "A Pedestrian Flow Analysis System using Wi-Fi Packet Sensors to a Real Environment," Proc. of the UbiComp, pp. 721-730, 2014.
- [8] N.E. Gemayel, et al., "A Weighted Least Squares Algorithm for Passive Localization in Multipath Scenarios," Proc. of VTC2015-Fall, 2015.
- [9] C. Yang and H. Shao, "WiFi-Based Indoor Positioning," IEEE Communication Magazine, Vol. 53(3), pp. 150-157, 2015.
- [10] R. Ma, et al., "An Improved WiFi Indoor Positioning Algorithm by Weighted Fusion," Sensors, pp. 21824-21843, Vol. 15(9), 2015.
- [11] N. Hernandez, et al., "WiFi-based indoor localization and tracking of a moving device," Proc. of UPINLBS, 2014.
- [12] L. Khalil and P. Jung, "Scaled Unscented Kalman Filter for RSSI-Based Indoor Positioning and Tracking," Proc. of NGMAST, 2015.
- [13] M. Shchekotov, "Indoor localization methods based on Wi-Fi lateration and signal strength data collection," Proc. of the 17th of fruct association, pp. 187-191, 2015.
- [14] G. Pipelidis, et al., "HuMan: Human Movement Analytics via WiFi Probes," Proc. of PerCom Workshops, 2019.
- [15] D.N. Fernandez, et al., "Implementation of a WiFi-based indoor localization system on a mobile device for a university area," Proc. of Intercon, 2019.