

Accuracy Improvement of RSSI-based Distance Localization using Unscented Kalman Filter (UKF) Algorithm for Wi-Fi Tracking Application

<https://doi.org/10.3991/ijim.v14i16.14077>

Syifaул Fuada (✉)

Universitas Pendidikan Indonesia, Bandung, Indonesia
syifaulfuada@upi.edu

Trio Adiono

Institut Teknologi Bandung, Bandung, Indonesia

Prasetyo

Korea Advanced Institute of Science and Technology, Daejeon, South Korea

Abstract—In this report, we perform the digital filter computation using Matlab for Wi-Fi tracking application. This work motivates to improve the accuracy of filter algorithm in the RSSI-based distance localization system. There are several aspects that we can improve, e.g., in the Filter part and Path-loss model. But, in this work, we focus on filter part; Unscented Kalman Filter (UKF) is implemented to replace linear Kalman Filter (KF), which is used in previous work. Based on the performance comparison, UKF has 90% hit ratio while linear KF has only 81.15 % hit ratio. We found that UKF can handle the noise in RSSI. Further work, the UKF algorithm is then embedded on the server system.

Keywords—Unscented Kalman Filter (UKF), RSSI-based Distance Localization, Wi-Fi Tracking System.

1 Introduction

In [1], we have performed some experiment to observe whether the RSSI method is suitable for Wi-Fi tracker system or not. From the experiment results, we found out that we can collect the required information from a smartphone, that are Wi-Fi signal strength and MAC address. Whenever the smartphones turn on the Wi-Fi, they will broadcast the packet data request through Wi-Fi signal in which the packet contains some information like channel, protocol, MAC address data, etc. To use RSSI method in the Wi-Fi tracker system, we can sniff that packet and measure the corresponding RSSI on each node.

In [1], there are several points that become consideration for further evaluation: 1) unstable value of RSSI; 2) interval-time which the packet data request is not constant broadcasted, it depends on the smartphone state; and 3) each smartphone emits different initial power. Those things become main concern whenever use RSSI method in Wi-Fi

tracker system. They keep changing dynamically because the present of noise. From the experiment, the distribution of RSSI value most likely to be Gaussian. We have performed KF to overcome noise in RSSI data that are scanned in sensor. However, the improvement of accuracy is required since Linear KF only reach about 80% hit ratio [2]. Hit ratio is a prediction result of the device location being less than 5 meters from the actual location, where the calculation is by dividing the number of results that are not error by the number of scans.

We try to use Unscented Kalman Filter (UKF) as our filter. Recent works employed UKF for their systems such as F. Tian, et al. [3], Z.I. Ghodousi, et al., [4], H. Lu, et al., [5], S. Wang, et al., [6], J. Yan, et al., [7].

For this work, the set of RSSI data are scanned in multiple sensors/nodes used the Non-Linier Least Square (NLS). This technique is RSSI range-based method in which it is applied to estimate the location of each devices. Based on literature review, including Ref. [3-7], we found that the NLS + UKF technique for RSSI-based Wi-Fi tracking system applications (precise indoor Wi-Fi localization) has not been done by many researchers. This paper is more emphasis on UKF computation in Matlab.

2 Methods

2.1 System block

Fig 2 shows the whole architecture of proposed Wi-Fi tracker system, which is an application that allows us to track the smartphone's location from its Wi-Fi signal. This application is used to analyze the distribution of the smartphones existed in a certain room/place. In this work, we divide the Wi-Fi Tracker system into two main parts: node system and server system. The node system is used as a sensor to scan the information that smartphone emits. The data is then sent to the server. Later, the server will perform some computation to predict the smartphone's location according to the data collected from the node network. We also equip our system with a dashboard (web-based application) to configure the algorithm as well as display the result.

In previous work, we have performed three indoor localization algorithms: Intersection Density, Non-Least Square (NLS), and Kalman Filter (KF). However, we obtained low-accuracy in detecting the smartphone's position. To improve the accuracy result, we try to use the UKF instead of KF. Further, the computation results of UKF in Matlab then implemented on the server system through Phyton programming.

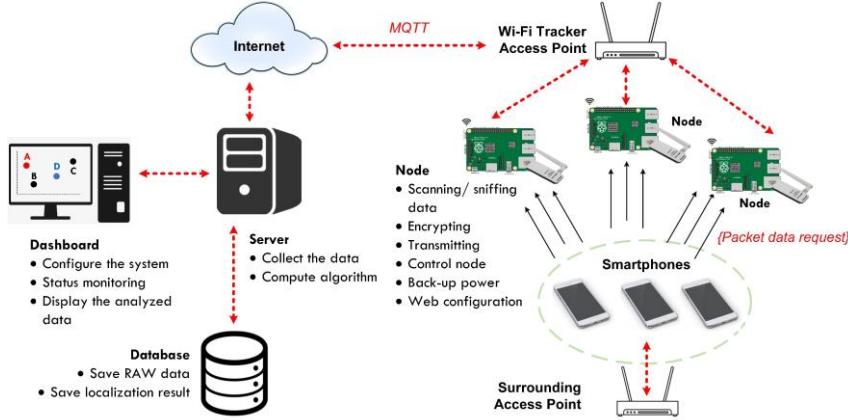


Fig. 1. Proposed Wi-Fi tracker system, we need minimum three nodes in order to set the indoor localization environment

2.2 UKF model

To define UKF, we first introduce the general KF expression. The general KF problem is stated as Eq. 1 and Eq. 2, where X is the state vector system, $f(\cdot)$ is transition function, Z is measurement vector, $h(\cdot)$ is measurement function, w is process noise, and v is measured noise. Both w and v are zero-mean Gaussian distribution with covariance Q and R , respectively.

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

$$Z_k = h(X_k) + v_k \quad (2)$$

There are several types of KF, i.e., 1) Linear KF in which both $f(\cdot)$ and $h(\cdot)$ are linear functions; 2) Extended Kalman Filter (EKF) in which transition $f(\cdot)$ and $h(\cdot)$ are nonlinear functions. The EKF linearizes those function using the Jacobian Matrix. And 3) Unscented Kalman Filter (UKF). Instead of linearizing the transition and measurement function, this method uses sigma-point from the dynamics system, then projects the points into the nonlinear function, later tries to estimate the best fit distribution from the projection result.

If we compare three methods above, the UKF algorithm is the most suitable for our application, where transition function is associated with the smartphone's orientation, smartphone movement, state of smartphone, path loss, and so on, while the measurement function is related to the noise that comes from the node, in which both of them are most likely to be nonlinear functions. The EKF has higher computational complexity due to the calculation of the Jacobian Matrix each time step. In this work, we still use KF as our filter. On the other hand, we have proven that linear KF still has a low accuracy, as reported in [2]. We will try to implement UKF as a filter.

After KF has introduced above, then we include the movement influence on the value of the RSSI that is scanned in the node. We change our state vector into Eq. 3

$$X = \begin{bmatrix} RSSi(t) \\ v(t) \\ P_{init}(t) \end{bmatrix} \quad (3)$$

From the state, we also estimate the initial power that is emitted by devices (smartphones). From the previous work [1], we understood that each smartphone has a different initial power. We include this parameter because we want to have a better estimation of the velocity influence in the RSSI value. We conclude like this based on the derivation as follows:

We take the derivative from the path loss model as Eq. 4,

$$\frac{dP}{dt} = \frac{d}{dt} (P(d_0) - 10 \alpha \log \frac{d}{d_0}) \quad (4)$$

Afterward, we get the rate change on RSSI as Eq. 5. The distance (d) depends on the initial power that transmitted by the smartphone as Eq. 6,

$$\frac{dP}{dt} = -\frac{10 \alpha}{d} v \quad (5)$$

$$d = 10^{\frac{(RSSi - P_{init})}{10\alpha}} \quad (6)$$

Then, by assuming that the velocity and initial power are constant, we get transition model as Eq. 7,

$$\begin{aligned} RSSI(t+1) &= RSSI(t) - 10\alpha * v(t) * 10^{\frac{(P_{init}(t) - RSSI(t))}{10\alpha}} * \Delta t \\ v(t+1) &= v(t) \\ P_{init}(t+1) &= P_{init}(t) \end{aligned} \quad (7)$$

For the measurement vector, we used the same as before, as expressed in Eq. 8. The measurement noise model is expressed as Eq. 9, and the process noise model is formulated as Eq. 10,

$$Z = [RSSI] \quad (8)$$

$$R = \left[\left(\frac{3.0 * RSSI + 340}{70} \right)^2 \right] \quad (9)$$

$$Q = \sigma^2 \begin{bmatrix} \Delta t^3 / 3 & \Delta t^2 / 2 & 0 \\ \Delta t^2 / 2 & \Delta t & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (10)$$

2.3 UKF calculation step

After the UKF has been modeled well, afterward, we elaborate on the calculation of the UKF algorithm in which it will be computed in Matlab. There are two steps, two steps as follows:

Prediction step

1. Calculate sigma point and weight

The first step in UKF is sigma points calculation referred from the previous prediction. Sigma points are sampling from distribution. The set of sigma points are deterministically chosen so that the mean and covariance of these points match the previous distribution points. We also calculate the weight for each sigma point. Weight determines how we choose the sampling in the distribution. There are two weight vectors: mean weight (W^m), covariance weight (W^c), as expressed by Eq. 11 and Eq. 12, respectively. Later, the rule is seen in Eq. 13,

$$W_0^m = \frac{\lambda}{n+\lambda} \quad (11)$$

$$W_0^c = \frac{\lambda}{n+\lambda} + 1 - \alpha^2 + \beta \quad (12)$$

$$W_i^m, W_i^c = \frac{1}{2(n+\lambda)}, i = 1, 2, \dots, 2n \quad (13)$$

$$\lambda = \alpha^2 (n + k) - n \quad (14)$$

Where n is the state vector dimension ($n = 3$) and, k , α , β , and λ (Eq. 14) are the parameters that determine the sampling distribution, larger α means the wider spread on distribution (with a note: $0 \leq \alpha \leq 1$), and $\beta = 2$ is right choice for the Gaussian problem.

Then, the sigma point is chosen as Eq. 15 for x_0 and Eq. 16 for x_1 , where μ and Σ is mean and covariance from the previous prediction. We scale the previous covariance matrix by $(n + \lambda)$, then take the square root of it. Then adding and subtracting from mean to ensure symmetry sampling. To calculate square the root matrix, we used the Cholesky decomposition.

$$\chi_0 = \mu \quad (15)$$

$$\chi_i = \begin{cases} \mu + [\sqrt{(n+\lambda)\Sigma}]_i, & \text{for } i = 1, 2, \dots, n \\ \mu - [\sqrt{(n+\lambda)\Sigma}]_{i-n}, & \text{for } i = n+1, \dots, 2n \end{cases} \quad (16)$$

2. Using unscented transform from the set of sigma-points

After we get sigma-point from the previous value, then pass the value through the transition function/model, which is a nonlinear function. Then from the set of the transformed value, we find the best fit distribution that matches the previous distribution by calculating its mean and covariance value using weight value. The unscented transform is defined by Eq. 17 and Eq. 18. Then, the predicted mean can be calculated using Eq. 19, while the predicted covariance can be calculated using Eq. 20. Both Eq. 19 and Eq. 20 are transition state.

$$y_i = f(\chi_i) \quad (17)$$

$$z_i = h(\chi_i) \quad (18)$$

$$\bar{y} = \sum_{i=0}^{2n} W_i^m y_i \quad (19)$$

$$P_{yy} = \sum_{i=0}^{2n} W_i^c (\bar{y} - y_i)(\bar{y} - y_i)^T + Q \quad (20)$$

Steps I and II are equivalent with prediction step of Linear KF as in [2]. The next step is correction, to correct to predicted value based on the measurement we have.

Correction step

1. Using measurement model

After the prediction step, we continue the calculation by defining the correction step in the UKF. The parameter used in our measurement model is only RSSI value because our sensor in the node system will be designed to read only the RSSI. We get this value from the first element of unscented transform y_i as Eq. 21. Then we calculate the mean and covariance for our measurement model as shown in Eq. 22 and Eq. 23, respectively.

$$z_i = y_i |_1 \quad (21)$$

$$\bar{z} = \sum_{i=0}^{2n} W_i^m z_i \quad (22)$$

$$P_{zz} = \sum_{i=0}^{2n} W_i^c (\bar{z} - z_i)(\bar{z} - z_i)^T + R \quad (23)$$

2. Compute Kalman gain

To compute Kalman gain, we compute cross covariance between predicted value and measurement using Eq. 24, and then we get Kalman gain as expressed in Eq. 25,

$$P_{yz} = \sum_{i=0}^{2n} W_i^c (\bar{y} - y_i)(\bar{z} - z_i)^T \quad (24)$$

$$K = P_{yz} * P_{zz}^{-1} \quad (25)$$

3. Final Correction step

To correct the predicted value, we first compute the residual between measurement result (Z_k) -in this case, the RSSI values that are scanned in node- with predicted value using Eq. 26. Then we correct our prediction using Eq. 27 and then Eq. 28. The value of X_{k+1} and P_{k+1} is then stored and used in the next prediction.

$$Res = Z_k - \bar{z} \quad (26)$$

$$X_{k+1} = \bar{y} + K * Res \quad (27)$$

$$P_{k+1} = P_{yy} - K * P_{zz} * K^{-1} \quad (28)$$

3 Results and Analysis

3.1 RSSI analysis

We have implemented UKF on Matlab and also carried out some simulation with several data. The RSSI data in this simulation is obtained from the smartphone that is placed in fix location (No movement), in which the method to get RSSI data is presented in [1]. Although the smartphone is in steady condition, the RSSI value that are scanned in the node keeps changing with high variance. This is the major problem of using RSSI-based localization method. Moreover, it will make the localization has lower accuracy. To overcome this problem, UKF is used to reduce the high variance on RSSI value.

Fig. 7 shows that UKF give better value on RSSI. It can reduce the noise and variance. We got RSSI value from steady smartphones which are placed in 10 meters distance from the node system.

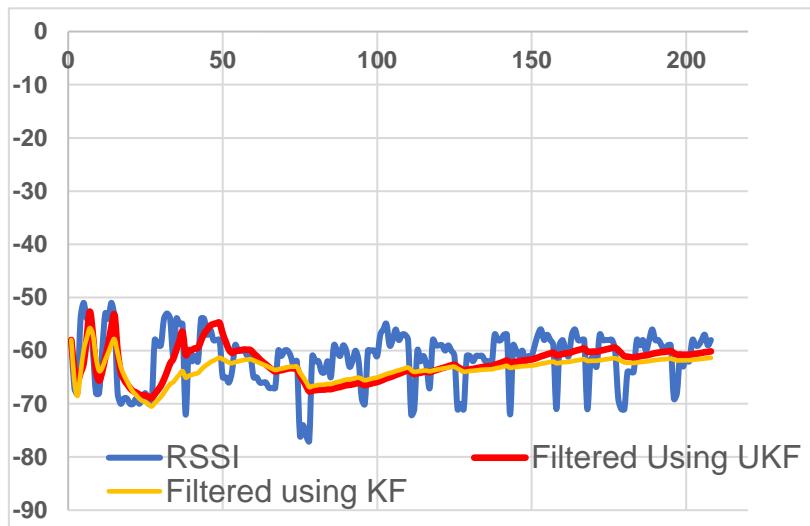


Fig. 2. Performance of UKF compared to Linear KF, the UKF can filter a high variance on RSSI

3.2 Estimation experiment using UKF

As introduced in the research background part, we designed the UKF to handle the movement influence to the estimated RSSI value. From the previous work [2], when the linear KF is at convergence state, it will handle the movement poorly (slow response). In this UKF simulation, we placed the smartphone on “point A” then move it to “point B”. Afterward, we estimated the location using UKF and the algorithm. The result is depicted in Fig. 8. In this experiment, the data is required to move from the

point A to B. Then we estimated the location using our algorithm. From this experiment, the algorithm needs around 30 – 40 sampling data to move from point A to B.

If we used linear KF as filter in the Wi-Fi tracker system, once it reached the convergence value, it required higher sampling data. Therefore, UKF has faster response than KF.

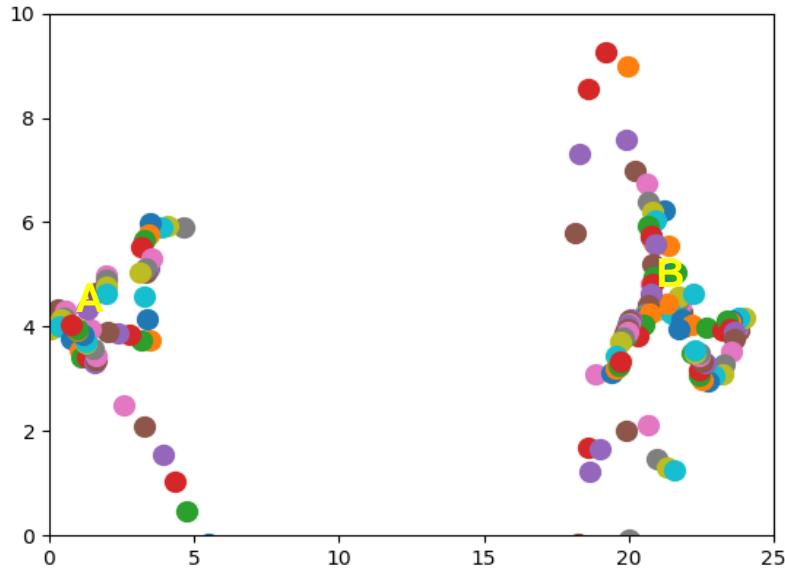


Fig. 3. UKF experiment on Matlab

4 Conclusion

Through literature studies, we define the algorithm that can be selected in our system further. We have two candidates, i.e., UKF and EKF, except Linear KF due to the performance is still low-accuracy as proven in [2]. In this work, we selected UKF instead of EKF with the consideration of low-computation aspect. In UKF, the nonlinearity on $f(\cdot)$ and $h(\cdot)$ are estimated using stochastic linearization that is simpler than EKF (using the Jacobian Matrix calculation).

The UKF computation in Matlab has been performed well. The UKF is able to reduce high variance on RSSI value. According to the results, the localization algorithm using UKF can be functioned to determine the devices location accurately (reach 90%) compared to previous work [2], i.e., Intersection Density, NLS, and Linear KF. In conclusion, UKF is the most optimum algorithm which can be applied in our Wi-Fi tracker system. Further we will provide/realize the server system embedded with UKF and node system.

5 References

- [1] S. Fuada, et al., "Your MAC Address can be detected easily when connected to the Wi-Fi," Unpublished.
- [2] S. Fuada, et al., "Modelling an Indoor Crowd Monitoring System based on RSSI-based Distance," Int. J. of Advanced Computer Science and Application (IJACSA), Vol. 11(1), pp. 660-667, 2020. <https://doi.org/10.14569/IJACSA.2020.0110181>
- [3] F. Tian and H. Li, "Study on the Improved Unscented Kalman Filter Ultra-Wideband Indoor Location Algorithm based on Two-Way Time-of-Flight," J. of Engineering Science and Technology Review, Vol. 11(5), 93-99, 2018. <https://doi.org/10.25103/jestr.115.11>
- [4] Z.I. Ghodousi, et al., "An Efficient Target Tracking in Directional Sensor Network using Adapted Unscented Kalman Filter," Wireless Personal Communication, 2019. <https://doi.org/10.1007/s11277-019-06660-z>
- [5] H. Lu, B. Ba, and W-J. Cui, "A Novel Fusion Visible Light Location Algorithm based on RSSI and Imaging of LEDs," Procedia Computer Science, Vol. 107, pp. 848-854, 2017. <https://doi.org/10.1016/j.procs.2017.03.180>
- [6] S. Wang, et al., "A Data Fusion Method of Indoor Location based on Adaptive UKF," Proc. of the 7th Int. Conf. on Digital Home, pp. 257-263, 2018. <https://doi.org/10.1109/ICDH.2018.00052>
- [7] J. Yan, H. Zhao, X. Luo, and Y. Wang, "Asynchronous Localization of Underwater Target using Consensus-based Unscented Kalman Filtering," IEEE J. of Oceanic Engineering, 2019. <https://doi.org/10.1109/JOE.2019.2923826>

6 Authors

Syifaul Fuada is with the Program Studi Sistem Telekomunikasi Universitas Pendidikan Indonesia (UPI) as a Lecturer. His research interests include analog circuit design and instrumentation, circuit simulation, engineering education, IoT, multimedia learning development and Visible Light Communication.

Trio Adiono is a Full professor and a senior lecturer at the School of Electrical Engineering and Informatics, and formerly serves as the Head of the Microelectronics Center, Institut Teknologi Bandung. His research interests include VLSI design, signal and image processing, VLC, smart cards, and electronics solution design and integration. Email: tadiono@stei.itb.ac.id

Prasetyo received the B.S. degree in electrical engineering from Institut Teknologi Bandung, Indonesia, in 2015. Currently, he is master student at Korea Advanced Institute of Science and Technology. His research interests include Wireless communication, VLSI, analog integrated circuits design, and CMOS technology. Email: Prasetyo@kaist.ac.id

Article submitted 2020-03-04. Resubmitted 2020-07-05. Final acceptance 2020-07-28. Final version published as submitted by the authors.