# FPGAS DESIGN AND IMPLEMENTATION OF LOCATION ALGORITHM BASE ON FILTERING TECHNIQUES

Wei Chen[1a], Yih-Shyh Chiou*[2], Bai-Yi Guo[1b], Lian-Xin Li[3]
[1] Undergraduate Student, Department of Electronic Engineering, Chung Yuan Christian University
Taoyuan City 320314, Taiwan
Email: [a]webberwchen@gmail.com; [b]a9101262002@gmail.com;

[2] Associate Professor, Department of Electronic Engineering, Chung Yuan Christian University
Taoyuan City 320314, Taiwan
Email: choice@cycu.org.tw

[3] Graduate Student, Institute of Communications Engineering, National Yang Ming Chiao Tung University
Hsinchu City 30010, Taiwan
Email: e0985677354@gmail.com

**KEY WORDS:** FIR filter (finite Impulse Response filter), FPGA (field programmable gate array), IIR filter (infinite impulse response filter), Kalman filter, least-squares approach

**ABSTRACT:** With the rapid development of wireless sensor networks and Internet of Things (IoT) technology, location-based services (LBS) based on positioning have gained increasing attention, which in turn has driven the development of positioning technology. Location-based technologies can be classified into outdoor and indoor positioning systems based on the environment. The development of outdoor positioning technology is already mature, such as Global Positioning System (GPS), but GPS signals cannot penetrate most buildings, making it unsuitable for indoor environments. With the increasing demand for indoor positioning applications, various applications have emerged in daily life, such as public transportation, commercial promotion, and emergency rescue. Common mechanisms used in indoor positioning systems include infrared, ultrasound, Bluetooth, and Wi-Fi technologies.

This article simulates the received signal strength indication (RSSI) of Wi-Fi signals in an indoor environment, converts the signal strength into distance according to the radio propagation model, and then calculates the position of the mobile terminal based on the transmission distance and the Least Square (LS) algorithm. The LS method using the radio propagation model can cause some estimated positions to be unreasonable. In order to reduce unreasonable estimation results, this article uses the Infinite Impulse Response (IIR) filter and Finite Impulse Response (FIR) filter algorithm to filter out the unreasonable signal data, improve the overall positioning accuracy, and then compare the filtered prediction paths using Kalman filter (KF) obtain more accurate predictions. Finally, the algorithm is implemented on a Field Programmable Gate Array (FPGA) on site.

## 1. INTRODUCTION

Localization has a wide range of applications, including GPS, Wi-Fi positioning, and autonomous driving, making it highly important and widely used. As a result, localization systems strive for greater accuracy. In this study, we replace traditional triangulation with the LS algorithm and write a program. LS calculations allow for the use of more Access Points (APs), leading to improved positioning results. In our comparative simulations, we initially assess the performance in different environments and deployment scenarios. Subsequently, we process the obtained RSSI data using IIR and FIR filters to find the most suitable filter coefficients for the given environment. We also compare the accuracy for different numbers of APs, making it easier to adapt the system to various environments. We combine the filtered radio wave data with KF to achieve tracking and prediction capabilities. Lastly, we consider the feasibility of implementing the system on hardware to further enhance computational speed.

### 1.1 IIR and FIR Filter

IIR is defined as (1). In this article, we use the IIR filter formula of BMP384 as our IIR filter (BMP384, 2020), as shown in equation (2), and the moving average filter as our FIR filter, as shown in equation (3). Additionally, we modify $c$ and $M$ to find the coefficients that best suit the current environment. Next, we use equation (4) to transform the signal into distances, which are used for LS localization.

$$yy_n[i] = -\sum_{k=1}^{N} a_k yy_n[i-k] + \sum_{k=0}^{M} b_k xx_n[i-k] \tag{1}$$

$$yy_n[i] = \frac{cyy_n[i-1]+xx_n[i]}{c+1} = \frac{c}{c+1} yy_n[i-1] + \frac{1}{c+1} xx_n[i]$$

$$\Rightarrow a_1 \triangleq -\frac{c}{c+1}, b_0 \triangleq \frac{1}{c+1}, N = 1, M = 0 \quad (2)$$

$$yy_n[i] = \frac{xx_n[i]+xx_n[i-1]+\cdots+xx_n[i-(M-1)]}{M}, \Rightarrow b_k \triangleq \frac{1}{M}, \quad (3)$$

where $M$ = feedforward filter order
$b_k$ = feedforward filter coefficient
$N$ = feedback filter order
$a_k$ = feedback filter coefficient
$xx_n[i]$ = input signal
$yy_n[i]$ = output signal
c = IIR filter coefficient

$$d_n = 10^{(\frac{A-y_n}{10\times n})}, \quad (4)$$

where $A$ = signal strength at a distance of 1 meter from AP
$n$ = path loss

### 1.2 Least Square Approach

The LS positioning algorithm offers mobile computation and is not limited by the number of APs. To use it in three dimensions, you need four or more reference points: $AP_1(x_1, y_1, z_1)$, $AP_2(x_2, y_2, z_2)$, $AP_3(x_3, y_3, z_3)$, $AP_4(x_4, y_4, z_4)$, and $AP_n(x_n, y_n, z_n)$, to estimate the coordinates of an unknown point $P(P_x, P_y, P_z)$. The calculation process is illustrated in steps (3) to (16) as shown in (Cengiz, 2021) (He et al., 2010).

$$d_1{}^2 = (P_x - x_1)^2 + (P_y - y_1)^2 + (P_z - z_1)^2 \quad (3)$$

$$d_2{}^2 = (P_x - x_2)^2 + (P_y - y_2)^2 + (P_z - z_2)^2 \quad (4)$$

$$d_3{}^2 = (P_x - x_3)^2 + (P_y - y_3)^2 + (P_z - z_3)^2 \quad (5)$$

$$d_4{}^2 = (P_x - x_4)^2 + (P_y - y_4)^2 + (P_z - z_4)^2 \quad (6)$$

$$\vdots$$

$$d_n{}^2 = (P_x - x_n)^2 + (P_y - y_n)^2 + (P_z - z_n)^2 \quad (7)$$

$$P_x \times 2(x_2 - x_1) + P_y \times 2(y_2 - y_1) + P_z \times 2(z_2 - z_1) = d_1{}^2 - d_2{}^2 + x_2{}^2 + y_2{}^2 + z_2{}^2 - x_1{}^2 - y_1{}^2 - z_1{}^2 \quad (8)$$

$$P_x \times 2(x_3 - x_1) + P_y \times 2(y_3 - y_1) + P_z \times 2(z_3 - z_1) = d_1{}^2 - d_3{}^2 + x_3{}^2 + y_3{}^2 + z_3{}^2 - x_1{}^2 - y_1{}^2 - z_1{}^2 \quad (9)$$

$$P_x \times 2(x_4 - x_1) + P_y \times 2(y_4 - y_1) + P_z \times 2(z_4 - z_1) = d_1{}^2 - d_4{}^2 + x_4{}^2 + y_4{}^2 + z_4{}^2 - x_1{}^2 - y_1{}^2 - z_1{}^2 \quad (10)$$

$$\vdots$$

$$P_x \times 2(x_n - x_1) + P_y \times 2(y_n - y_1) + P_z \times 2(z_n - z_1) = d_1{}^2 - d_n{}^2 + x_n{}^2 + y_n{}^2 + z_n{}^2 - x_1{}^2 - y_1{}^2 - z_1{}^2 \quad (11)$$

$$\boldsymbol{Pa} = \boldsymbol{b} \quad (12)$$

$$\boldsymbol{P} = \begin{bmatrix} P_x \\ P_y \\ P_z \end{bmatrix} \quad (13)$$

$$\boldsymbol{a} = 2 \times \begin{bmatrix} x_2 - x_1 & y_2 - y_1 & z_2 - z_1 \\ x_3 - x_1 & y_2 - y_1 & z_2 - z_1 \\ x_4 - x_1 & y_2 - y_1 & z_2 - z_1 \\ \vdots & \vdots & \vdots \\ x_n - x_1 & y_n - y_1 & z_2 - z_1 \end{bmatrix} \quad (14)$$

$$\boldsymbol{b} = \begin{bmatrix} {d_1}^2 - {d_2}^2 + {x_2}^2 + {y_2}^2 + {z_2}^2 - {x_1}^2 - {y_1}^2 - {z_1}^2 \\ {d_1}^2 - {d_3}^2 + {x_3}^2 + {y_3}^2 + {z_3}^2 - {x_1}^2 - {y_1}^2 - {z_1}^2 \\ {d_1}^2 - {d_4}^2 + {x_4}^2 + {y_4}^2 + {z_4}^2 - {x_1}^2 - {y_1}^2 - {z_1}^2 \\ \vdots \\ {d_1}^2 - {d_n}^2 + {x_n}^2 + {y_n}^2 + {z_n}^2 - {x_1}^2 - {y_1}^2 - {z_1}^2 \end{bmatrix} \tag{15}$$

$$\boldsymbol{P} = (\boldsymbol{a}^T\boldsymbol{a})^{-1}\boldsymbol{a}^T\boldsymbol{b} \tag{16}$$

### 1.3 Kalman Filter Tracking Algorithm

The KF is an optimal linear estimator of signals in measurement noise (Kalman, 1960). The Kalman filtering equations provide a real-time algorithm for estimating an unknown recursive state vector, with the measurement quality of each noisy data based on the minimization of mean square error to track its target (Chen et al., 2005). It has the advantage of low memory footprint (it only needs to retain the previous state, aside from the current state) and operates at high speeds, making it well-suited for applications involving real-time constraints and embedded systems. The system's state is represented as a vector with real-valued elements.

With each discrete-time step, this linear transformation acts on the current state, producing a new state along with some noise. Additionally, control information from known controllers is integrated into the system. The KF algorithm is a special case of Bayesian filtering algorithms under linear and Gaussian distribution conditions. It utilizes the Bayesian Approach to predict the observation points$x_{0:k}$ of the state model at time *k* and obtain the predicted distribution for the target $x_{0:k+1}$. In the correction step, it uses known observation points $z_{0:k+1}$ to obtain the predicted distribution for the target$x_{0:k+1}$ (Huang et al., 2005). Therefore, the various stages of the KF's cycle can be derived from Markov Chain and Bayesian Theory.

$$\begin{bmatrix} x_{1,k+1} \\ x_{2,k+1} \\ x_{3,k+1} \\ S_{1,k+1} \\ S_{2,k+1} \\ S_{3,k+1} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & \Delta_k & 0 & 0 \\ 0 & 1 & 0 & 0 & \Delta_k & 0 \\ 0 & 0 & 1 & 0 & 0 & \Delta_k \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_{1,k} \\ x_{2,k} \\ x_{3,k} \\ S_{1,k} \\ S_{2,k} \\ S_{3,k} \end{bmatrix} + \begin{bmatrix} \frac{{\Delta_k}^2}{2} & 0 & 0 \\ 0 & \frac{{\Delta_k}^2}{2} & 0 \\ 0 & 0 & \frac{{\Delta_k}^2}{2} \\ \Delta_k & 0 & 0 \\ 0 & \Delta_k & 0 \\ 0 & 0 & \Delta_k \end{bmatrix} \begin{bmatrix} \eta_{1,k} \\ \eta_{2,k} \\ \eta_{3,k} \end{bmatrix} \tag{17}$$

$$E\{\Delta_n\boldsymbol{\eta}_n(\Delta_n\boldsymbol{\eta}_n)^T\} = \begin{cases} \mathbf{Q}_k & \text{for } n = k \\ 0 & \text{for } n \neq k \end{cases} = \delta(k-n)\,\mathbf{Q}_k \tag{18}$$

$$\mathbf{x}_{k+1} = \boldsymbol{\Phi}_k\mathbf{x}_k + \Delta_n\boldsymbol{\eta}_n,\ \ \boldsymbol{\eta}_k \sim N\ (\mathbf{0}, \mathbf{Q}_k) \tag{19}$$

Equations (17)-(19) are used to calculate the position prediction for the received signal location in Wi-Fi sampling. The prediction of the next time step's position is based on the previous time step's position, taking into account possible noise. In these equations, $x_{1,k}$, $x_{2,k}$, and $x_{3,k}$ represent the position of the mobile device in the X, Y, and Z-axis directions, respectively. $s_{1,k}$, $s_{2,k}$, and $s_{3,k}$ represent the velocity of the mobile device in the X, Y, and Z directions. $X_k$ is the state matrix, $\boldsymbol{\Phi}_k$ is the model noise matrix, $\Delta_k$ represents the time between signal receptions, $\boldsymbol{\eta}_k$ is the mobile state process noise matrix, and $\mathbf{Q}_k$ is the model noise covariance matrix.

$$\begin{bmatrix} z_{1,k} \\ z_{2,k} \\ z_{3,k} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_{1,k} \\ x_{2,k} \\ x_{3,k} \\ s_{1,k} \\ s_{2,k} \\ s_{3,k} \end{bmatrix} + \begin{bmatrix} v_{1,k} \\ v_{2,k} \\ v_{3,k} \end{bmatrix} \tag{20}$$

$$E\{\mathbf{v}_n(\mathbf{v}_n)^T\} = \begin{cases} \mathbf{R}_k & \text{for } n = k \\ 0 & \text{for } n \neq k \end{cases} = \delta(k-n)\,\mathbf{R}_k \tag{21}$$

$$\mathbf{Z}_{k+1} = \mathbf{H}_k\mathbf{x}_k + \mathbf{v}_n,\ \ \mathbf{v}_k \sim N\ (\mathbf{0}, \mathbf{R}_k) \tag{22}$$

Equations (20)-(22) involve the actual measured positions. To correct potential errors in predictions and achieve optimal positioning, it is necessary to use actual positions for reasonable prediction adjustments. $\mathbf{Z}_k$ represents the actual measurement matrix, $\mathbf{H}_k$ is the measurement transition matrix, $\mathbf{V}_k$ is the measurement noise matrix, and $\mathbf{R}_k$ is the measurement noise covariance matrix.

$$\tilde{\mathbf{x}}_k = \mathbf{\Phi}_{k-1}\hat{\mathbf{x}}_{k-1} \tag{23}$$

$$\tilde{\mathbf{P}}_k = \mathbf{\Phi}_{k-1}\hat{\mathbf{P}}_{k-1}\mathbf{\Phi}_{k-1}^T + \mathbf{Q}_{k-1} \tag{24}$$

$$\mathbf{e}_{k|j} = \mathbf{x}_{k|j} - \mathbf{x}_k, \quad \mathbf{e}_{k|k} = \hat{\mathbf{e}}_k, \quad \mathbf{e}_{k|k-1} = \tilde{\mathbf{e}}_k \tag{25}$$

$$\hat{\mathbf{P}}_k = E\{\hat{\mathbf{e}}_k\hat{\mathbf{e}}_k\}, \qquad \tilde{\mathbf{P}}_k = E\{\tilde{\mathbf{e}}_k\tilde{\mathbf{e}}_k{}^T\} \tag{26}$$

$$\mathbf{K}_k = \tilde{\mathbf{P}}_k\mathbf{H}_k^T[\mathbf{H}_k\tilde{\mathbf{P}}_k\mathbf{H}_k^T + \mathbf{R}_k]^{-1} \tag{27}$$

$$\hat{\mathbf{x}}_k = \tilde{\mathbf{x}}_k + \mathbf{K}_k(\mathbf{z}_k - \mathbf{H}_k\tilde{\mathbf{x}}_k) \tag{28}$$

$$\hat{\mathbf{P}}_k = [\mathbf{I} - \mathbf{K}_k\mathbf{H}_k]\tilde{\mathbf{P}}_k \tag{29}$$

Subsequently, the process enters the Kalman filtering cycle, where $\hat{\mathbf{x}}_k = \mathbf{x}_{k|k}$ and $\tilde{\mathbf{x}}_k = \mathbf{x}_{k|k-1}$ are the state estimation matrices and state prediction matrices, respectively. $\mathbf{U}_k$ is the model noise matrix, $\mathbf{e}_{k|j}$, $\hat{\mathbf{e}}_k$, and $\tilde{\mathbf{e}}_k$ are the state error matrices, estimation error matrices, and prediction error matrices, respectively. $\hat{\mathbf{P}}_k$, $\tilde{\mathbf{P}}_k$, and $\mathbf{K}_k$ are the estimation error covariance matrices, prediction error covariance matrices, and Kalman gain (KG) (Chiou et al., 2012) (Hou et al., 2016) (Al-Fuqaha et al., 2015).

## 2. RESEARCH METHODOLOGY

### 2.1 Developing with MATLAB Software

In the MATLAB simulation, we introduced Gaussian noise with different variances to the RSSI to simulate real-world environments. With the same number of APs, we applied both IIR and FIR filters while varying $c$ and $M$ to determine the optimal filter coefficients, as shown in Figure 1 and Figure 2. Under the same taps count, IIR filtering outperforms FIR filtering, as demonstrated in Figure 3, and it performs almost equally to FIR with a higher taps count, as shown in Figure 4. FIR exhibits greater delay, and increasing the tap count to enhance filtering performance would require more registers in hardware, as depicted in Figure 5. Therefore, our hardware implementation will focus on the IIR filter.

Subsequently, we compared the accuracy of the best-performing filter coefficients under different numbers of APs, as shown in Figure 6 and Table 1. It can be observed that under the best coefficients, there is minimal difference in accuracy between the two filters. Additionally, we found that LS converges at 5APs, and increasing the number of APs has little impact on the localization results. Under the foundation of LS positioning, the KF algorithm is introduced to enhance the overall algorithm's accuracy using its recursive properties and achieve tracking effects. The simulation tests the CDF, as shown in Figure 7.

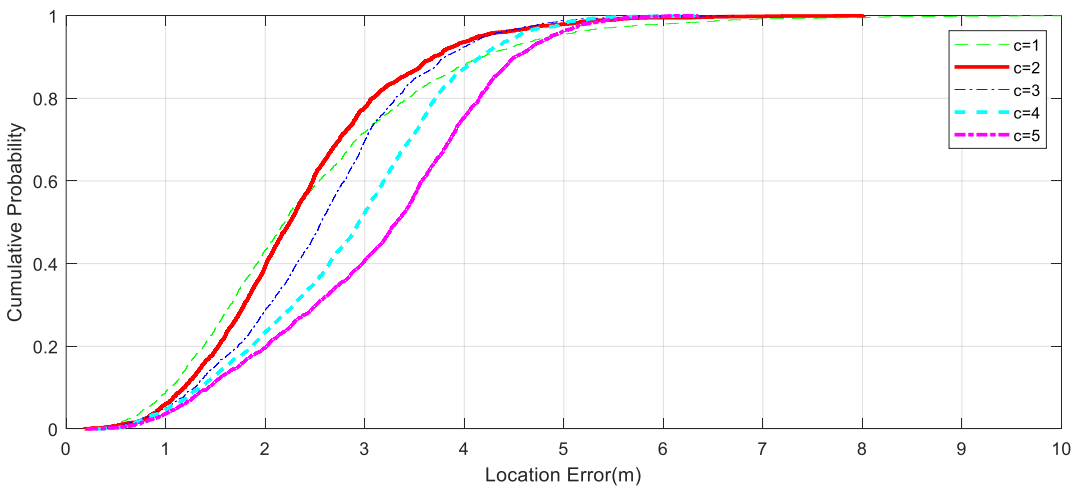


Figure 1. Cumulative probability distribution of the error distances, 5AP $c$ from 1 to 5, when the noise variance is 2dB.

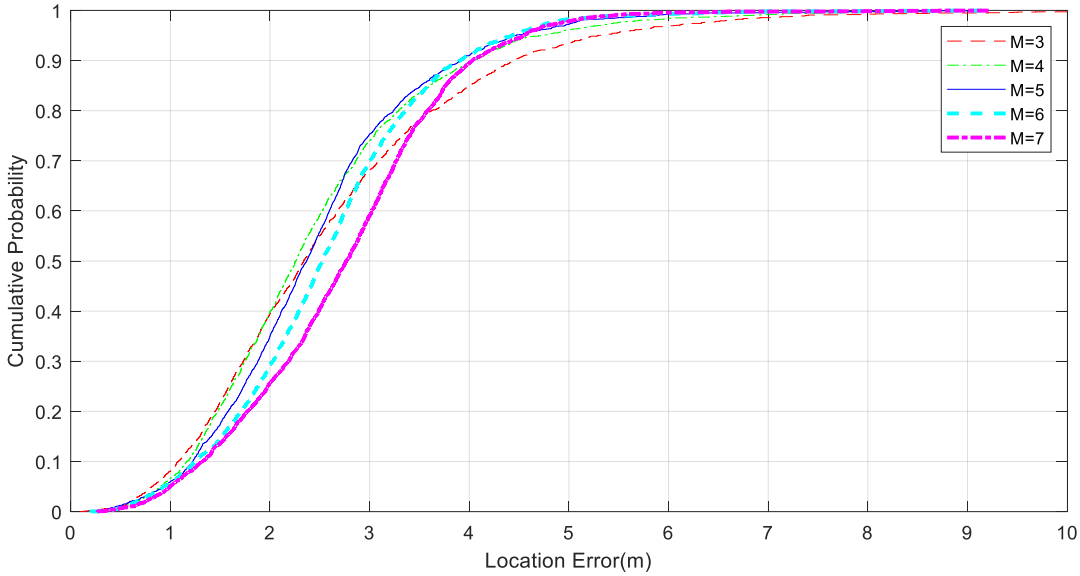


Figure 2. Cumulative probability distribution of the error distances, 5AP *M* from 3 to 7, when the noise variance is 2dB.

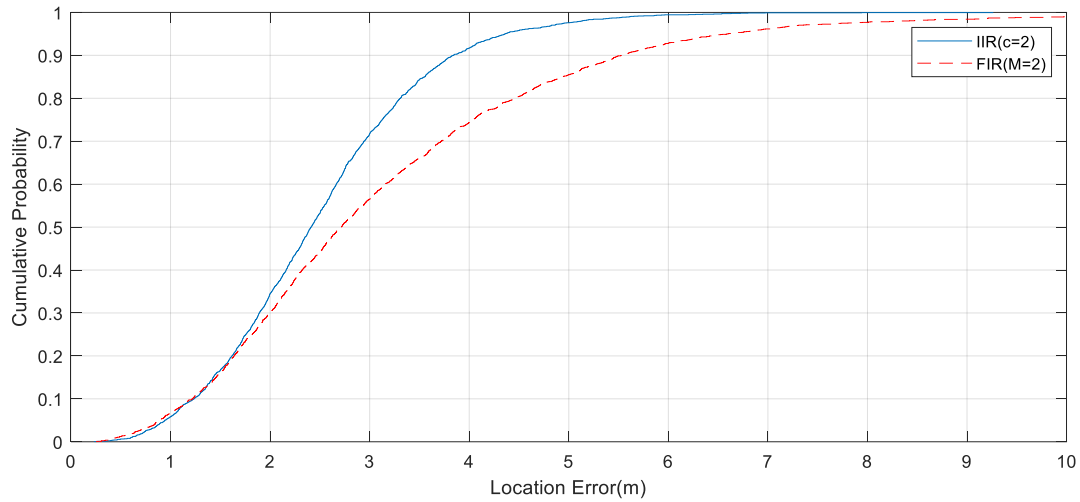


Figure 3. Cumulative probability distribution of the error distances, when IIR and FIR under same taps.

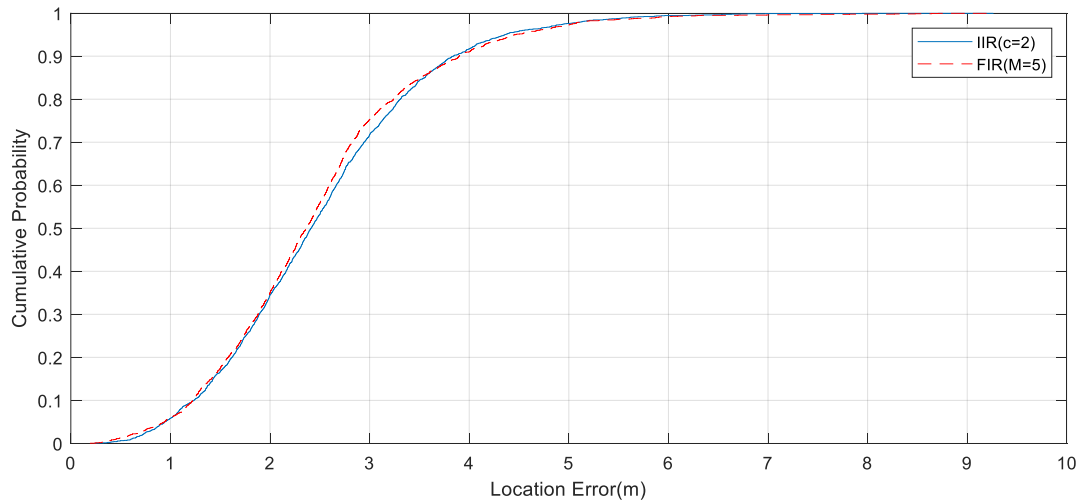


Figure 4. Cumulative probability distribution of the error distances, best coefficient of IIR and FIR.

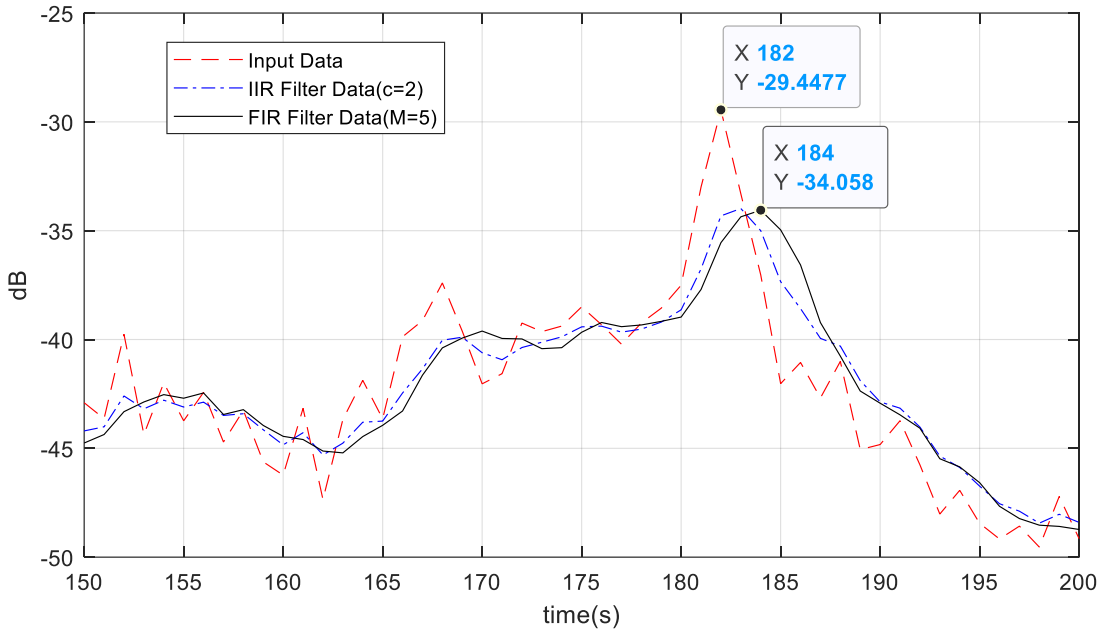


Figure 5. Compare the delay between the IIR filter and the FIR filter.

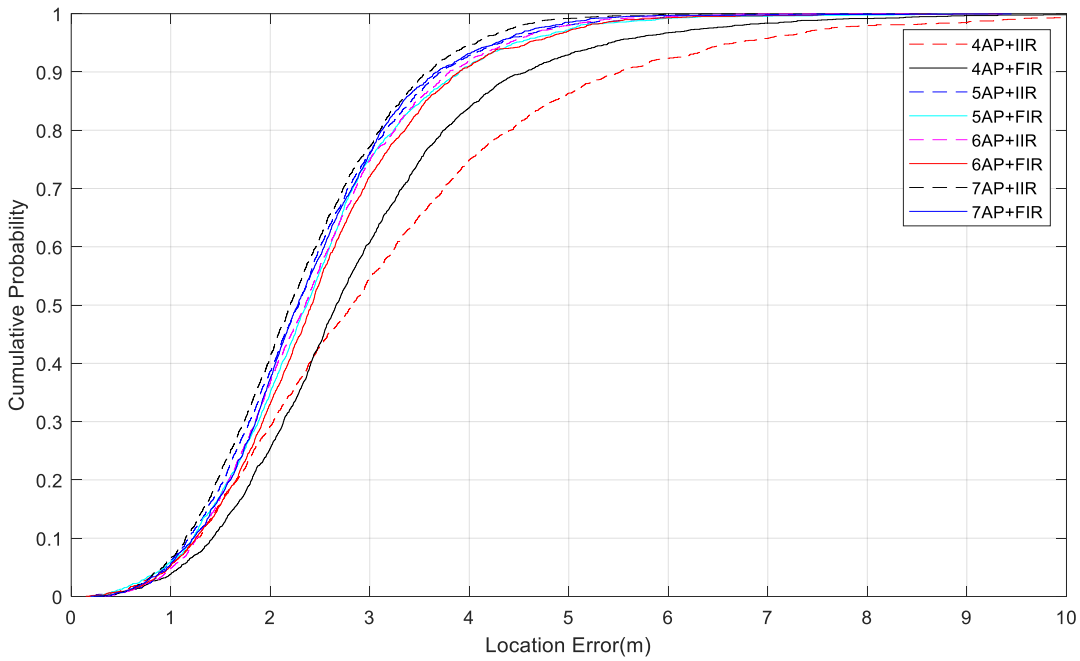


Figure 6. Cumulative probability distribution of the error distances, when the noise variance is 2dB.

Table 1. Cumulative probability distribution of the error distances, LS 5AP with IIR and FIR filter in different environments.

| Method / CDF | VAR=2 5AP+IIR $c$=1 | VAR=2 5AP+IIR $c$=2 | VAR=2 5AP+FIR $M$=5 | VAR=4.53 5AP+IIR $c$=2 | VAR=4.53 5AP+IIR $c$=3 | VAR=4.53 5AP+FIR $M$=6 |
|---|---|---|---|---|---|---|
| 90% | 4.52m | 3.71m | 3.90m | 5.37m | 4.78m | 5.10m |
| 50% | 2.33m | 2.28m | 2.37m | 2.94m | 2.96m | 3.00m |

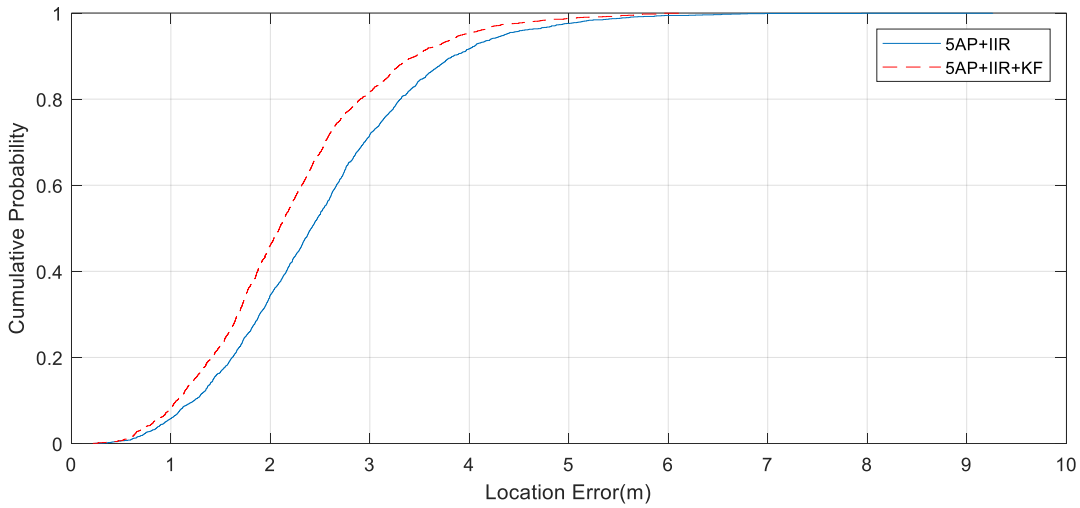


Figure 7. Cumulative probability distribution of the error distances, 5AP+IIR, 5AP+IIR+KF, when the noise variance is 2dB.

### 2.2 The Proposed Localization Algorithms Base on FPGA

This section introduces the hardware implementation results of IIR, LS and KF, examining the experimental paths in a simulated indoor 3D environment. In terms of signal processing, we employ the IIR filter to filter the received signals and then convert them into distance data using the RSSI formula. For localization, we utilize the LS with more APs to achieve higher accuracy, and we split the matrix operations of the algorithm to operate on FPGA, as shown in Figure 8.

Regarding KF tracking, Figure 9 demonstrates the circuit structure based on the (23)-(29). We split the matrices according to the computation cycles in the algorithm. From the software simulation results, this tracking algorithm can accurately reconstruct the predicted path. Hardware testing using the DE2-115 FPGA board also correctly calculates the values. We use shifting to amplify the values, reducing many floating-point operations, and leverage the parallel processing capabilities of hardware to decrease computation time. Using hardware computation offers advantages such as low hardware cost and high speed.

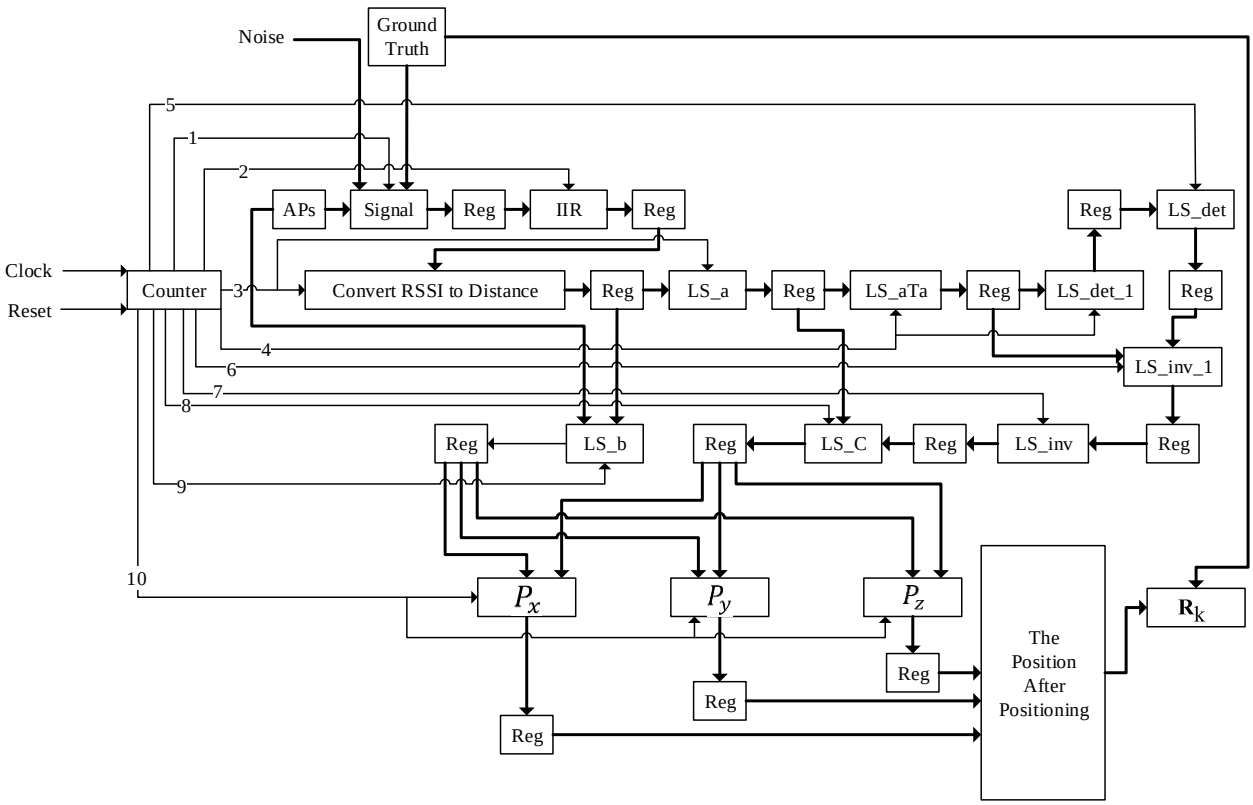

Figure 8. FPGA architecture of the IIR filter and LS positioning algorithm.

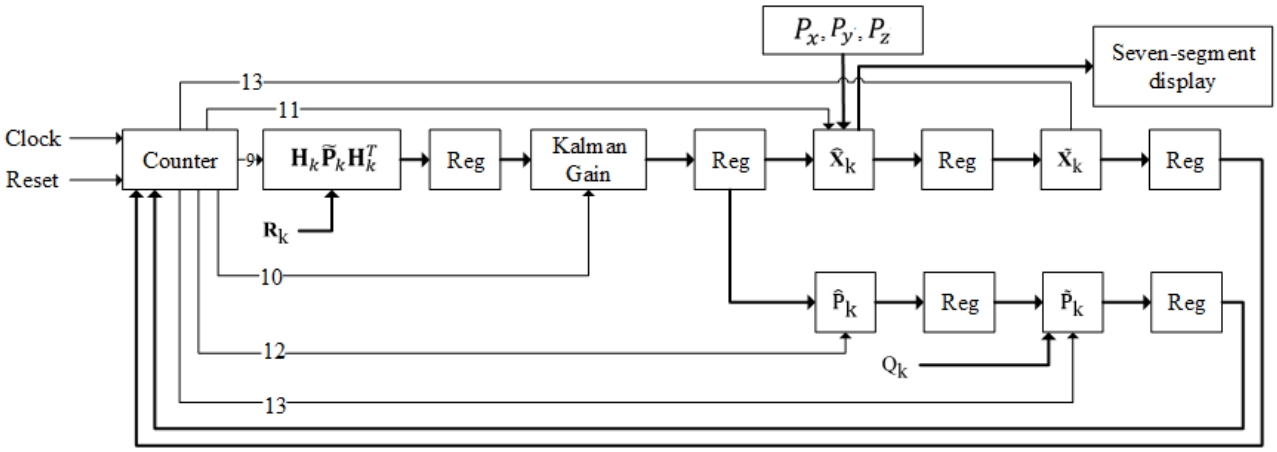


Figure 9. FPGA architecture of the KF tracking algorithm.

To address the issue of floating-point arithmetic in hardware, we adopted a shifting technique to amplify numerical values, reducing the problems associated with floating-point arithmetic and compensating for the loss of precision in hardware implementation. Figure 10 compares the results between MATLAB simulation and ModelSim simulation. Figure 11 demonstrates that the errors introduced by hardware shifting are acceptable. In comparison, it offers a faster computation speed, making it more suitable for real-time location-based services.

The comparison in Figure 11 also shows that using KF for localization and tracking enhances accuracy. From the experimental results, it's evident that the FPGA-based localization and tracking algorithm can effectively meet the requirements of real-time positioning and tracking.

| | | | | | | |
|---|---|---|---|---|---|---|
| MATLAB simulation: X-axis | -0.95459 | 0.323975 | 0.813477 | 0.181274 | 0.355347 | 0.707031 |
| Y-axis | 0.625488 | -0.594116 | -0.6521 | 0.354858 | -1.575195 | -0.404663 |
| Z-axis | -0.635498 | -2.292114 | -0.87561 | -1.062134 | -1.372925 | -0.567749 |
| ModelSim simulation X-axis | -0.954964 | 0.324077 | 0.813784 | 0.181384 | 0.355536 | 0.707349 |
| Y-axis | 0.625011 | -0.595568 | -0.65326 | 0.353672 | -1.575897 | -0.372503 |
| Z-axis | -0.636009 | -2.293414 | -0.87598 | -1.062431 | -1.374343 | -0.567935 |

Figure 10. MATLAB simulation results and ModelSim simulation results.

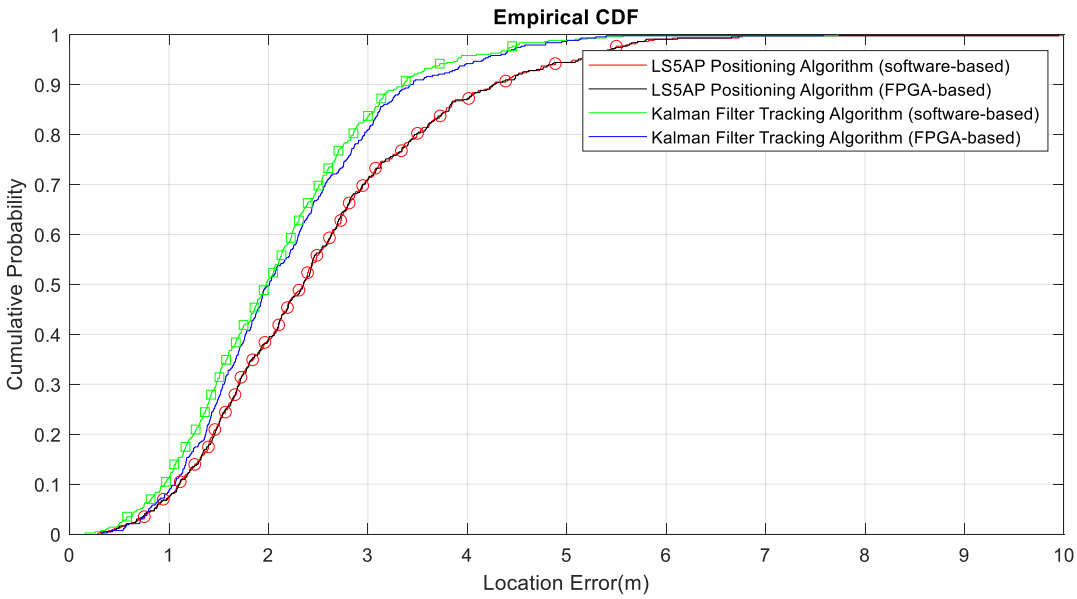


Figure 11. Cumulative probability distribution of the error distances, 5AP+IIR software-based, 5AP+IIR FPGA-based, 5AP+IIR+KF software-based and 5AP+IIR+KF hardware based.

Table 2. Comparison of Execution Times for Positioning and Tracking Localization Algorithms: Software-Based vs. FPGA-Based Approaches.

| Approach / Executing Time | Tradition Software Program (CPU i7 / RAM 16GB) | The Proposed Approach (Hardware Design and FPGA) |
|---|---|---|
| IIR+LS Executing Time | 57.552us | 3.132us |
| IIR+LS+KF Executing Time | 112.543us | 3.770us |

## 3. CONCLUSIONS

Regarding localization, using the LS algorithm for localization allows us to use more APs compared to traditional triangulation, improving localization accuracy. It enables us to adjust the number of APs used for localization based on different environments and hardware capabilities. In terms of tracking, in an ideal environment, adding an IIR filter at the signal end can reduce the estimation error of the KF filter. Although IIR filtering introduces nonlinear phase delays, the magnitude of the error caused by IIR in localization depends on the sampling rate and object movement speed. Leveraging the fast processing speed of hardware can mitigate the localization error caused by IIR filter computation delays. On the hardware side, to optimize and improve the implementation of FPGA methods, we used shifters instead of multipliers and dividers to overcome hardware issues related to floating-point calculations. As shown in Figure 11 and Table 2, the FPGA achieves comparable accuracy to software but offers a significant advantage in processing speed.

**ACKNOWLEDGEMENTS**

This work was supported in part by the Ministry of Science and Technology, Taiwan (R.O.C.), under Grant numbers of MOST 110-2121-M-033-001 and MOST 111-2121-M-033-001, and in part by the National Science and Technology Council, Taiwan (R.O.C.), under Grant number of NSTC 112-2121-M-033-001, and by Chung Yuan Christian University, Taiwan, R.O.C., under Grant CYCU-10701228.

**REFERENCES:**

Al-Fuqaha, A., Guizani, M., Mohammadi, M., Aledhari, M. and Ayyash, M., 2015. Internet of Things: A Survey on Enabling Technologies, Protocols, and Applications. *IEEE Communications Surveys & Tutorials*, 17(4), pp. 2347-2376.

Cengiz, K., 2021. Comprehensive Analysis on Least-Squares Lateration for Indoor Positioning Systems. *IEEE Internet of Things Journal*, 8(4), pp. 2842-2856.

Bosch Pressure Sensor BMP384 datasheet, 2020. pp. 14. (Available: https://www.bosch-sensortec.com/media/boschsensortec/downloads/datasheets/bst-bmp384-ds003.pdf, Accessed: September 2023)

Chen, G. and Guo, L., 2005. The FPGA implementation of Kalman Filter. In *Proc. 5th WSEAS international Conference on Signal Processing, Computational Geometry and Artificial Vision(ISCGAV '05),* Sep. 2005, pp.61-65.

Chen, W.-T., Chen, S.-L., Chiou, Y.-S., Lin, T.-L., Wen, F.-J. and Lin, Y.-K., 2019. FPGA Based Implementation of Reduced-Complexity Filtering Algorithm for Real-Time Location Tracking. In *Proc. 2019 IEEE Intl Conf on Dependable, Autonomic and Secure Computing, Intl Conf on Pervasive Intelligence and Computing, Intl Conf on Cloud and Big Data Computing, Intl Conf on Cyber Science and Technology Congress (DASC/PiCom/CBDCom/CyberSciTech)*, pp. 721-726.

Chiou, Y.-S., Tsai, F. and Yeh, S. C., 2012. A Low-Complexity Data-Fusion Algorithm Based on Adaptive Weighting for Location Estimation. In *Proc. 2012 International Conference on Information Security and Intelligent Control*, pp. 294-297.

Chiou, Y.-S., Liu, Y.-H., Chen, Y.-J., Chen, S.-L., Lin, T.-L., Chen, W.-T., Zhang, Y.-S., Chen, T.-H., 2020. Design and Implementation of 3D Real-Time Positioning and Tracking Algorithms in FPGA for Location Estimation. In *Proc. 2020 IEEE Eurasia Conference on IOT, Communication and Engineering (ECICE), pp. 40-43*.

Doiphode, S. R., Bakal, J. W. and Gedam, M., 2016. A Hybrid Indoor Positioning System Based on Wi-Fi Hotspot and Wi-Fi Fixed Nodes. In *Proc. 2016 IEEE International Conference on Engineering and Technology (ICETECH)*, pp. 56-60.

He, Y., Martin, R. and Bilgic, A. M., 2010. Approximate Iterative Least Squares Algorithms for GPS Positioning. In *Proc. The 10th IEEE International Symposium on Signal Processing and Information Technology*, pp. 231-236.

Hou, L., Zhao, S., Xiong, X., Zheng, K., Chatzimision, P., Hossain, M. S. and Xiang, W., 2016. Internet of Things Cloud: Architecture and Implementation. *IEEE Communications Magazine*, 54(12), pp. 32-39.

Huang, J.-H., 2014. Study on Indoor Positioning with Bluetooth Low Energy. Available: https://hdl.handle.net/11296/8pe8t9

Kalman, R. E., 1960. A New Approach to Linear Filtering and Prediction Problems. *Journal of Basic Engineering, 82(1),* pp. 35-45.