Celal Bayar University Journal of Science

Noise Reduction Techniques for Sensor Data: Comparative Analysis of Kalman, Butterworth, Savitzky-Golay, Median, and Moving Average Filters for UWB-Based Position Estimation

Levent Türkler[1]*, Lütfiye Özlem Akkan[2]

[1] Bergama Vocational Training School, Ege University, Izmir, Türkiye
[2] İzmir Vocational School, Dokuz Eylul University, Izmir, Turkey
* levent.turkler@ege.edu.tr
* Orcid No: 0000-0001-5957-8278

Received: April 25, 2025
Accepted: September 23, 2025
DOI: 10.18466/cbayarfbe.1682594

Abstract

Accurate localization of autonomous mobile systems has become a critical requirement in modern engineering applications. However, field environments often lead to erroneous position data due to signal interference or unexpected behaviors of signals in the presence of obstacles. In this study, raw data obtained from an Ultra-Wideband (UWB) positioning system was intentionally degraded by amplification and the addition of artificial noise to simulate realistic signal corruption. Subsequently, five different filters were evaluated to denoise this highly contaminated data. The performance of each filter was tested using custom-developed software by comparing its unoptimized and optimized configurations. As a key outcome, the optimal parameter set of the most effective filter for noise reduction was identified and reported.

Keywords: Kalman, Butterworth, Savitzky-Golay, Median, MAF, Filters.

1. Introduction

As the number of robots increases, it becomes increasingly important to know their exact locations and to ensure that they reach their intended destinations accurately [1]. While GPS-based methods are commonly used for outdoor localization, indoor environments require alternative techniques such as UWB, Wi-Fi, or infrared due to signal obstruction. However, a key challenge arises in ensuring positional accuracy, particularly due to increasing errors caused by environmental factors. Therefore, this study discusses methods used to eliminate such errors and achieve accurate localization and experimentally evaluates their effectiveness.

Regardless of whether the system operates indoors or outdoors, the positioning methods used are typically based on tracking fundamental signal sources. To ensure precise positioning, the selected signals must be resilient against both everyday signal interference and magnetic disturbances. This isolation is essential for maintaining accurate location data. In other words, minimizing the susceptibility of these signals to environmental interference or signal similarity is of critical importance. Nevertheless, unexpected disturbances remain a constant possibility. In some cases, inaccuracies may arise due to faulty data from positioning sensors, insufficient information during computations, or the presence of physical obstacles that interfere with signal transmission in the field.

In this context, cleaning the incoming data and retrieving the true values is a critical process. To achieve this, various filters are employed to denoise the signals, correct erroneous values, and optimize the results [1]. Some of these filters are more effective when processing dynamic data, while others perform better with static measurements. Certain filters exhibit strong noise suppression capabilities, whereas others excel at responding to sudden signal changes. In some cases, a filter that is advantageous under specific conditions may become less effective in different scenarios [2,3].

Hurtado-Perez et al. [4] conducted a comprehensive review of motion detection systems and examined the filters used in motion analysis. Some filtering techniques can be advantageous depending on the type of application, while others may have disadvantages.

Prior to the comparative analysis, reviewing the mathematical formulations of the selected filters—

Kalman, Butterworth, Savitzky-Golay, Median, and Moving Average—provides essential insights into their signal processing mechanisms.

Using filters for UWB-based position estimation is a current research topic. In a recent study by Borhan et al. [5], four filtering approaches were applied for indoor localization, and a comparative analysis was conducted.

In this study, five distinct filters were deliberately selected to represent different classes of signal processing techniques, each with unique mathematical structures and noise suppression characteristics. These filters are frequently applied in engineering applications and collectively offer a broad evaluation spectrum for denoising UWB-based localization data.

The Kalman Filter is a model-based (state-space) predictive technique commonly used in dynamic systems. Its ability to estimate system states through recursive updates makes it effective for tracking time-varying signals. Furthermore, it can be extended to nonlinear systems through variants such as the Extended Kalman Filter (EKF) and the Unscented Kalman Filter (UKF), which enhances its adaptability in various environments.

The Butterworth Filter is a classical digital low-pass filter that operates in the frequency domain. It provides a maximally flat magnitude response in the passband, making it effective in eliminating high-frequency continuous noise while maintaining signal smoothness.

The Savitzky-Golay Filter performs polynomial regression within a moving window in the time domain. It is particularly useful when preserving the local shape and curvature of the signal is essential, making it well-suited for applications requiring minimal signal distortion alongside noise suppression.

The Median Filter is a nonlinear, rank-based filter that excels in rejecting impulsive disturbances such as spike or salt-and-pepper noise. By replacing outliers with the median of the surrounding values, it preserves edge sharpness and is highly robust against isolated signal anomalies.

The Moving Average Filter is a computationally simple linear filter that smooths signals by averaging values over a fixed-length window. Due to its low computational cost and ease of implementation, it is widely used in real-time embedded systems.

Together, these five filters cover a wide range of noise conditions:

Gaussian and continuous noise: Kalman, Butterworth, Moving Average

Impulsive noise: Median

Shape-preserving smoothing: Savitzky-Golay

Several commonly known alternatives were excluded based on methodological considerations:

The Wiener filter requires prior knowledge of the signal-to-noise ratio (SNR) and assumes stationary noise conditions, which do not align with the assumptions of this study.

The Hampel Filter, though robust, is computationally intensive and thus less suitable for embedded or low-resource environments.

The Gaussian Filter, while effective in smoothing, may degrade signal fidelity in nonlinear signal trajectories due to its kernel-based nature.

This study introduces several key distinctions from prior filter comparison research, both in terms of methodology and practical relevance.

Five different filters—Kalman, Butterworth, Savitzky-Golay, Median, and Moving Average—were evaluated within a unified experimental framework using the same synthetic UWB-based dataset and motion scenario. Crucially, each filter was applied with individually optimized parameter settings rather than default or fixed values, enabling a fair and balanced comparison under ideal conditions.

The dataset was intentionally corrupted with a realistic mixture of noise types: approximately 95% Gaussian noise, 4% high-frequency jitter, and 1% impulsive spike noise. This hybrid noise model better approximates real-world conditions than conventional single-noise simulations, providing a more rigorous test environment for each filtering method.

Performance assessment was carried out using a diverse set of evaluation metrics, including RMSE, R^2, average error, maximum error, standard deviation, and a composite success score. This comprehensive approach allows for the analysis of not only filtering accuracy but also robustness and consistency across dimensions.

Moreover, the study focuses specifically on Ultra-Wideband (UWB)-based localization, which involves unique signal characteristics due to its reliance on Time-of-Flight measurements. This domain-specific focus enhances the real-world applicability of the findings, particularly for indoor positioning scenarios.

Although the filtering was performed offline, the controlled nature of the experiment allows the outcomes to inform future real-time implementations. The results provide a practical foundation for selecting appropriate filter configurations in embedded or constrained environments where noise characteristics vary over time.

All filtering operations performed in this study were carried out offline using pre-recorded UWB data. This controlled setup was designed to isolate filter behavior under consistent noise conditions, without the constraints of real-time system latency or processing limitations.

1.1 Kalman Filter

The Kalman Filter is a prediction-based filtering method that models measurement data using system dynamics [6]. Its predictive nature enables it to compensate for missing values and reduce errors in dynamic systems. As a result, it is widely used in fields such as robotic navigation, image processing, financial modeling, machine learning, and sensor fusion [7].

The filter operates through two main steps: prediction of the system state using prior data and a system model, followed by an update based on new measurements to reduce noise. This cyclical process allows the system to approach an ideal trajectory and is suitable for real-time applications.

The standard Kalman Filter is designed for linear system estimation. For nonlinear systems, extended versions such as the Extended Kalman Filter (EKF), Unscented Kalman Filter (UKF), and Cubature Kalman Filter (CKF) provide improved accuracy, though they require more processing power [8]. For highly variable or high-dimensional systems, adaptive or ensemble methods such as the Ensemble Kalman Filter (EnKF) are often used [9,10]. Sub-models of the Kalman Filter are shown in Figure 1. These sub-models are categorized and adapted to provide solutions for different types of problems.

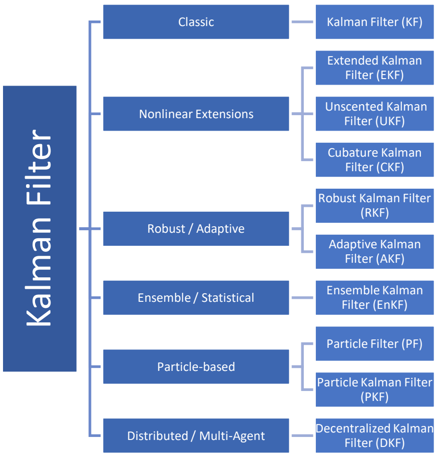

Kalman Filter
Classic
Kalman Filter (KF)
Nonlinear Extensions
Extended Kalman Filter (EKF)
Unscented Kalman Filter (UKF)
Cubature Kalman Filter (CKF)
Robust / Adaptive
Robust Kalman Filter (RKF)
Adaptive Kalman Filter (AKF)
Ensemble / Statistical
Ensemble Kalman Filter (EnKF)
Particle-based
Particle Filter (PF)
Particle Kalman Filter (PKF)
Distributed / Multi-Agent
Decentralized Kalman Filter (DKF)

Figure 1. Models of Kalman Filter.

Choosing the appropriate variant depends on system complexity and computational capacity; the most advanced filter is not always the optimal choice. The mathematical model of the basic Kalman Filter is given in Equations 1.1–1.3 [11], while the EKF formulation is presented in Equations 1.4–1.10 [12].

$$\hat{x}_k^- = F_k \hat{x}_{k-1} + B_k u_k \tag{1.1}$$

$$K_k = P_k^- H_k^T (H_k P_k^- H_k^T + R_k)^{-1} \tag{1.2}$$

$$\hat{x}_k = \hat{x}_k^- + K_k (z_k - H_k \hat{x}_k^-) \tag{1.3}$$

Although the EKF and standard Kalman Filter share similar structures, the EKF is more complex and not preferred for linear systems.

$$\hat{x}_k^- = f(\hat{x}_{k-1}, u_{k-1}) \tag{1.4}$$

$$P_k^- = F_k P_{k-1} F_k^T + Q_k \tag{1.5}$$

$$F_k = \left.\frac{\partial f}{\partial x}\right|_{x=\hat{x}_{k-1}, u=u_{k-1}} \tag{1.6}$$

$$K_k = P_k^- H_k^T (H_k P_k^- H_k^T + R_k)^{-1} \tag{1.7}$$

$$H_k = \left.\frac{\partial h}{\partial x}\right|_{x=\hat{x}_k^-} \tag{1.8}$$

$$\hat{x}_k = \hat{x}_k^- + K_k \big(z_k - h(\hat{x}_k^-)\big) \tag{1.9}$$

$$P_k = (I - K_k H_k) P_k^- \tag{1.10}$$

In the basic Kalman Filter, proper balance between the input data and system dynamics (prediction) must be maintained for the model to operate effectively. This balance is achieved through the calibration of the Kalman Filter using two key matrices: the process noise covariance matrix (Q), which reflects system uncertainties, and the measurement noise covariance matrix (R), which represents errors in sensor measurements and determines how much confidence is placed in sensor data. A small value for Q assigns more weight to the system model, thereby reducing the influence of sensor measurements. In contrast, a large Q reflects greater process uncertainty, increasing the filter's reliance on incoming sensor data. Similarly, a small R value suggests high confidence in sensor data, while a large R means greater reliance on predicted values.

By reducing measurement noise, the Kalman Filter improves position estimation, aligning predicted values with actual positions. Its practical application on raw data is shown in Figure 2.

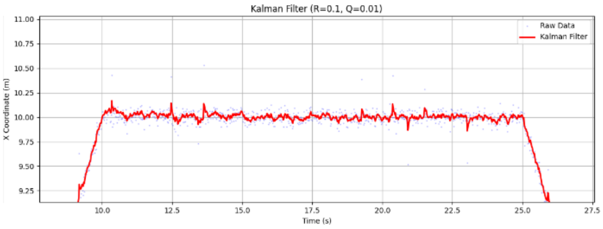


Figure 2. Basic Kalman Filter Application.

1.2 Butterworth Filter

The Butterworth Filter operates with a distinct approach: it suppresses high-frequency noise in the frequency domain, resulting in a smoother signal profile. Fundamentally, it is a low-pass filter characterized by a maximally flat frequency response in the passband, which ensures the absence of ripple [13].

High-frequency noise is eliminated by attenuating frequency components above the cutoff frequency. The sharpness of the transition band depends on the filter order; as the order increases, the transition becomes steeper. This filter is typically implemented as an Infinite Impulse Response (IIR) filter, meaning it operates based on both past inputs and past outputs. The difference equation in the time domain is generally expressed as shown in Equation 1.11 [14]:

$$y[n] = b_0 x[n] + b_1 x[n-1] + \cdots + b_M x[n-M] - a_1 y[n-1] - \cdots - a_N y[n-N] \tag{1.11}$$

$x[n]$	: current input signal
$y[n]$	: filtered output
$x[n-1], x[n-M]$	: previous input values
$y[n-1], y[n-N]$	: previous output (filtered) values
b_i, a_i	: filter coefficients determined through the Butterworth Filter design

As seen in this equation, the filter operates not only on the current input but also on past input and output values. As a result, it does not respond instantly to sudden changes; instead, the Butterworth Filter smooths the signal and introduces a phase delay. This phase delay shifts the output in time relative to the input, which may impact real-time responsiveness. However, in low-order filters, this phase delay is generally negligible. Therefore, it does not pose a significant issue in offline signal processing. On the other hand, in real-time systems, the delay can affect the system’s response time and may introduce minor inaccuracies [15].

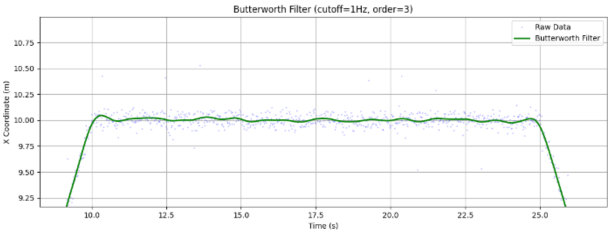


Figure 3. Basic Butterworth Filter Application.

In this study, a second-order Butterworth Filter—characterized by a two-pole configuration—was employed to balance noise attenuation and computational simplicity. The filter’s order influences both frequency selectivity and phase delay. While lower-order Butterworth Filters introduce less phase delay, they still achieve effective noise suppression in applications where moderate filtering is sufficient. The second-order Butterworth Filter offers a smooth frequency response in the transition band and effectively attenuates high-frequency components, resulting in a smoothed signal. Due to these characteristics, it is often preferred as a simple and stabilizing filtering method that does not significantly distort the phase of the signal [16]. Figure 3 shows the application of the Butterworth Filter to standard raw data.

1.3 Savitzky-Golay Filter

Another filter included in the comparison prior to the start of the analysis is the Savitzky-Golay (S-Golay) Filter. This digital filtering technique is particularly used in spectral data to reduce noise while preserving the essential features of the signal. The method is based on fitting a polynomial to the data within a defined window using the least squares approach and estimating either the central value or the derivative of the polynomial. In this way, the signal can be smoothed, and its derivatives can also be computed [17].

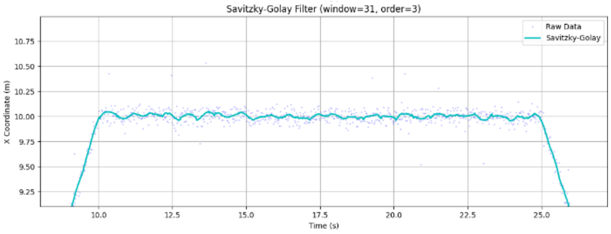


Figure 4. Basic Savitzky-Golay Filter Application.

The Savitzky-Golay Filter reduces noise while maintaining the signal's local trends and curvature, making it particularly suitable for smoothing and derivative estimation in time-series data. It is particularly suitable for derivative calculations and operates in the time domain, which results in minimal phase distortion.

As shown in Table 1 and Table 2, selecting an appropriate window size and polynomial order is crucial. Otherwise, the filter may introduce signal smoothing artifacts or fail to suppress noise effectively. This method is limited by its reduced ability to suppress high-frequency noise and its dependency on large data windows, which can degrade performance when sample size is insufficient [18]. An example of this filter, applied to raw data without optimization, is presented in Figure 4.

Table 1. Effect of Window Size Selection.

Small Window (5 samples)	**Large Window (25 samples)**
Preserves fine details	Reduces more noise
Limited noise reduction	May lose subtle features
Fast response	Slower response, potential lag

Table 2. Effect of Polynomial Order.

Low Order (1–2)	High Order (≥3)
Smooth curves	Captures subtle trends
Poor tracking of complex patterns	Risk of overfitting
Low computational cost	High comp. cost

The Savitzky-Golay Filter fits a polynomial to the data points within a window. The mathematical formulation of this process is expressed in Equation 1.12 [19]:

$$\hat{y}[n] = \sum_{k=-M}^{M} c_k \cdot x[n+k] \qquad (1.12)$$

Where:

$\hat{y}[n]$: filtered output at the n^{th} point
$x[n+k]$: original data points within the window
c_k : filter coefficients (precomputed weights)
M : window half-width (Window size = 2M + 1)

Here, the coefficients c_k are calculated using least squares regression, depending on the window size and polynomial order, as shown in Equation 1.13:

$$c = (A^T A)^{-1} A^T e \qquad (1.13)$$

1.4 Median Filter

Another filter used in this study is the Median Filter. In terms of its window-based operation, it shares certain similarities with the Savitzky-Golay (S-Golay) filter. Like S-Golay, the Median Filter employs a sliding window; however, instead of fitting a polynomial or computing an arithmetic mean, it sorts the values within the window and selects the median, as shown in Equation 1.15. This allows it to ignore sudden spikes or outliers (spike noise) [20], making it particularly effective against impulsive noise.

The Median Filter, when configured with appropriate parameters, maintains the general characteristics of the signal. However, the use of large window sizes can result in excessive smoothing and loss of fine signal details. Employing a fixed window size on signals with dynamic variations may lead to suboptimal performance, manifesting as either inadequate noise suppression or distortion of significant signal components. Consequently, window size selection requires careful consideration. With proper configuration, the Median Filter offers a simple yet effective noise reduction method. Nevertheless, its performance remains limited when dealing with broadband, low-amplitude noise types such as white noise [21].

Given a signal $x[n]$, the output of the Median Filter $y[n]$ is defined as in Equation 1.14:

$$y[n] = median\,\{x[n-k], \dots, x[n], \dots, x[n+k]\} \qquad (1.14)$$

Where:

n : current sampling point
k : half of the window width (The full window size is $2k+1$; for example, if the window size is 33, then W/2=k =16)

$$\text{median}(S) = S_{\left(\frac{|S|+1}{2}\right)} \qquad (1.15)$$

When S, represents the sorted data set.

In this model, data points are selected according to the defined window size, sorted, and the central median value is assigned as the output. This process is repeated across the entire dataset by sliding the window. It is important to note that selecting the middle value after sorting may resemble an arithmetic mean; however, this is not the case. Unlike the arithmetic mean, which is highly sensitive to outliers, the Median Filter provides robust performance by selecting the middle value from a sorted window, effectively suppressing impulsive noise. This characteristic makes the Median Filter particularly advantageous in datasets with non-consecutive, abrupt spikes. This behavior of the Median Filter is illustrated in Figure 5, where it has been applied to raw data.

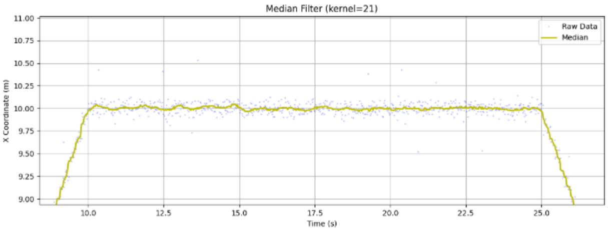


Figure 5. Basic Median Filter Application.

1.5 Basic Moving Average Filter

Another filter used in this study is the Moving Average Filter. It is one of the simplest and most commonly used low-pass filters. It operates by averaging the signal values within a defined window, thereby reducing high-frequency noise and smoothing abrupt changes. The filter has a linear structure and can be applied in either a causal or non-causal manner depending on the time reference [22].

Given a signal $x[n]$, the Moving Average Filter with a window size of N is defined as in Equation 1.16 [23]:

$$y[n] = \frac{1}{N} \sum_{k=0}^{N-1} x[n-k] \qquad (1.16)$$

Where:

N : window size (number of samples to be considered)
$x[n-k]$: signal values within the window
$y[n]$: filtered output signal

At each step, the filter computes the average of the most recent N values, shifts the window, and calculates a new average.

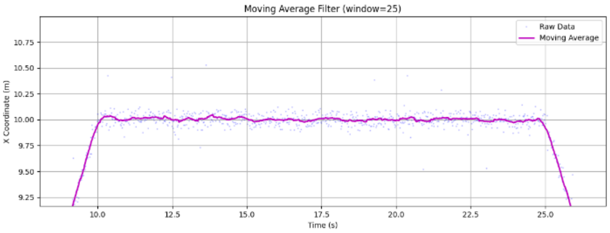


Figure 6. Basic Moving Average Filter Application.

The Moving Average Filter attenuates high-frequency random noise by averaging successive values, while preserving general low-frequency trends in the signal. Its simple and computationally efficient structure makes it well-suited for real-time applications, particularly in resource-constrained environments such as embedded systems. The filter passes low-frequency components and attenuates high-frequency components depending on the window length. The frequency response ($T\omega$) depends on the window duration as shown in Equation 1.17 [22]; increasing the window length improves low-pass characteristics but slows the system response. At the same time, frequency components at $f = 1/T_w$ and its harmonics are fully suppressed.

$$G_{MAF}(j\omega) = \left|\frac{sin(\omega Tw/2)}{\omega Tw/2}\right| \tag{1.17}$$

Delay Analysis: A delay also occurs in the output, corresponding to half the window length in sampling intervals, as defined in Equation 1.18 [24].

$$Tdelay = \frac{Tw}{2} \tag{1.18}$$

Increasing the window length enhances the suppression of high-frequency noise in the frequency domain, but also increases phase delay and slows the filter’s response time in the time domain.

Application Scenario: The Moving Average Filter is particularly preferred in applications where high-frequency noise suppression is required. Due to its simplicity, it is widely used in resource-constrained environments such as microcontrollers and embedded systems. For instance, it is effective in attenuating high-frequency noise in IMU data, particularly from accelerometer and gyroscope outputs. However, in the presence of sharp signal transitions, the Moving Average Filter tends to smooth out critical signal features and performs poorly against impulsive noise patterns characterized by sudden, isolated deviations. Furthermore, the filter's performance is highly dependent on the window size, which must be carefully optimized according to the specific application [22].

In summary, the Moving Average Filter offers advantages such as low computational cost, ease of implementation, and acceptable frequency filtering capabilities. However, it also has limitations, including response delay and edge blurring. Therefore, the window size must be chosen carefully depending on the intended use case. Figure 6 presents the Basic Moving Average Filter applied to raw data.

Following these evaluations, a comparative summary of the main characteristics of the filters discussed can be found in Table 3.

Table 3. Comparative Analysis of Filters.

Feature	Kalman	Butterworth	Savitzky-Golay	Median	Moving Average
Domain of Operation	Model-based, time domain	Frequency domain	Time domain	Time domain	Time domain
Mathematical Structure	State estimation using measurement and process noise models	IIR (Infinite Impulse Response), smooth passband	Sliding window with polynomial fitting	Sliding window with sorting (non-linear)	Sliding window with arithmetic mean
Linearity	Linear / Non-linear in extended forms	Linear	Linear	Non-linear	Linear
Noise Type Addressed	Gaussian noise, measurement noise	High-frequency continuous noise (jitter)	Gaussian noise, curve-preserving smoothing	Impulsive noise (spike noise), salt-and-pepper noise	Low-level continuous noise, Gaussian noise
Phase Delay	Minimal (optimal prediction)	Noticeable phase shift, more prominent with sharp transitions	Low, depending on window size	Present but minimal (depends on window size)	Low, depending on window size
Performance with Heavy Noise	High, dynamically adaptive	Good, especially in strong noisy environments	Good with low-to-medium noise, preserves shape	Very effective for sudden high-intensity noise	Limited effectiveness in strong noise
Computational Complexity	Medium-High (state prediction)	Low-Medium	Medium (polynomial coefficients calculation)	Low-Medium (sorting algorithm)	Low (simple addition and division)

Latency (Data Delay)	Very low	Medium	Medium, depending on window size	Medium-low, window size dependent	Low, window size dependent
Signal Shape Preservation	High (model-based prediction)	Low (frequency suppression)	High (maintains signal shape)	Medium (focused on outlier removal)	Low to medium
Typical Applications	Robotics navigation, positioning estimation, dynamic systems	Audio processing, radar, control systems	Biomedical signals, spectral data smoothing	Image processing, sensor data denoising	Basic data smoothing, time series analysis
Advantages	Dynamic, adaptive, optimal estimation	Effective noise suppression, easy design	Smoothing while preserving shape	Excellent for outlier and spike noise removal	Simple, fast, computationally light
Disadvantages	Requires system model, computationally intensive	Phase shift may affect low frequencies	Critical window and polynomial degree selection	Risk of losing details if window too large	Weak performance with high-frequency or complex noise

While filters are commonly applied in dynamic and real-time systems, this study deliberately employs a static model to isolate and evaluate filtering performance under controlled conditions. This means that the data will be pre-recorded, and the filters will be applied afterward to achieve the most realistic modeling possible. The primary objective is to evaluate the effectiveness of basic filtering methods in the context of known position tracking. Therefore, external factors such as signal impairments and non-line-of-sight (NLoS) conditions, which typically affect signal behavior, are excluded from the scope and not considered in the performance evaluation. Additionally, the study seeks to mitigate the impact of faulty sensor data on signal-based localization accuracy. In summary, this work tests the performance of various filters in suppressing noise by processing ideal, error-free Ultra-Wideband (UWB) signals collected from a vehicle with a known trajectory.

2. Materials and Methods

Ultra-Wideband (UWB) technology is a wireless communication method operating in the 3.1 GHz to 10.6 GHz frequency range, characterized by its wide spectrum and high temporal resolution. Thanks to its large bandwidth, it enables highly accurate Time of Flight (ToF) measurements in positioning tasks, offering precision at the millimeter level. In this study, positioning data will be collected using UWB devices, and during the data acquisition phase, multipath effects and environmental noise will also be considered. To simulate realistic noise conditions, raw UWB signals were artificially degraded, and the resulting dataset was used to evaluate the performance of the selected filters.

To generate the dataset and implement the system, ToF measurements will be carried out using anchors and tags, as position estimation will be required by the UWB device. The UWB module used in the study, shown in Figure 7, is based on an ESP32 microcontroller and features a DW1000 UWB chip along with a small integrated display [25].

According to the principles of triangulation, UWB systems require at least four anchors for accurate 3D localization. However, in a 2D environment where height variation is negligible, three anchors are sufficient to determine the horizontal position of the tag, as illustrated in Figure 8(a) [26]. In this study, position estimation was carried out using only two anchors, assuming that their positions were precisely known and remained static throughout the experiment. This configuration, shown in Figure 8(b), was designed to minimize anchor-related uncertainty in the 2D plane [27].

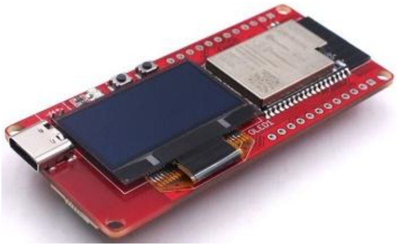
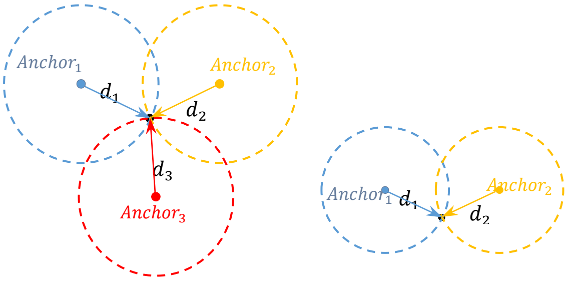


Figure 7. The UWB module to be used.



(a) (b)

Figure 8. Position estimation using 3 anchors.

In 2D positioning using UWB-based distance measurements, the position is calculated using Equation 1.19 [28].

$$d_i = c \cdot \frac{(t_{rx,i} - t_{tx,i})}{2} \tag{1.19}$$

Where:
d_i → distance to the i^{th} anchor
c → speed of light
t_{rx}, t_{tx} → reception and transmission times

In this way, the distance is calculated based on the delay between the transmission and reception of the signal from the anchor. Assuming that the signal propagates at the speed of light is a widely accepted approach. Accordingly, position estimation using at least three anchors (trilateration) is calculated as shown in Equation 1.20:

$$\begin{cases}(x-x_1)^2+(y-y_1)^2=d_1^2\\(x-x_2)^2+(y-y_2)^2=d_2^2\\(x-x_3)^2+(y-y_3)^2=d_3^2\end{cases} \tag{20}$$

By solving this system of equations using distance values d_1, d_2, d_3 obtained from the UWB devices, the instantaneous position (x, y) can be determined. In the initial stage of the application, it is assumed that the movement area is clearly defined, obstacle-free, and within a line-of-sight (LoS) condition. Under these assumptions, designing the system structure becomes more straightforward.

In the system design, two anchors and one tag were used. Additionally, a Raspberry Pi 5 (8 GB) was employed as a Single Board Computer (SBC) for data collection. The UWB system was deployed in a fixed 10 × 10m square area. The planned motion trajectory of the tag, including its position and velocity profiles across different time intervals, is detailed in Table 4. This kinematic scenario was designed to include both linear segments and sharp turns to evaluate filter response under varying dynamic conditions.

To emulate real-world signal imperfections, a synthetic dataset of 2500 entries was created by introducing random noise, occasional missing values, and abrupt deviations. This dataset served as a controlled testbed for evaluating the performance of the filtering algorithms.

Table 4. UWB motion dynamics summary.

Time (s)	**Position**	**Velocity (m/s)**		**Description**
		X	**Y**	
0	(0.0 0.0)	0	0	Starting Point
0-10	(0.0 0.0) → (10.0 0.0)	+1.0	0	Moving at constant speed in X axis
10	(10.0 0.0)	0	0	First Corner
10-25	(10.0 0.0) → (10.0 10.0)	0	+0.667	Moving at constant speed in Y axis
25	(10.0 10.0)	0	0	Second Corner
25-35	(10.0 10.0) → (0.0 10.0)	-1.0	0	Moving at constant speed in -X axis
35	(0.0 10.0)	0	0	Third Corner
35-50	(0.0 10.0) → (0.0 0.0)	0	-0.667	Moving at constant speed in -Y axis
50	(0.0 0.0)	0	0	End Point

The UWB-based position data were recorded in a CSV file, as shown in Table 5. When the trajectories and velocities of these movements are analyzed in the time domain, the noise distribution depicted in Figure 9 is observed. The error distribution approximates a Gaussian profile, characterized by dense small-amplitude variations along both the UWB_X and UWB_Y axes, with an average noise level of approximately ±0.2 m. For example, several distinct peaks are observed around ±0.5 m [29,30]. This indicates that the dataset is predominantly affected by low-amplitude Gaussian noise, with occasional occurrences of impulsive spike noise. The noise composition is estimated to be approximately 95% Gaussian, 4% high-frequency jitter, and 1% impulsive noise.

Table 5. Sample UWB-Tag data.

Time	**UWB_X**	**UWB_Y**
0.0	0.05325032774257202	0.10040631139887034
0.020008003201280513	0.0830952712004122	0.06051834721755446
0.040016006402561026	0.18017345033699717	-0.09376326227337284
0.060024009603841535	0.14679581203675318	-0.011623728512373806
0.08003201280512205	0.015216670328084658	-0.044687552804533265

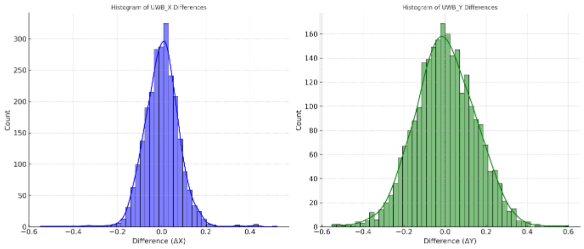


Figure 9. Histogram of position differences in the X and Y axes of the UWB data.

As shown in Figure 9, there is a difference in histogram counts. This discrepancy arises from the different distribution characteristics of the data along the X and Y axes.

Although the total number of data points is the same, the variations along the X-axis are concentrated within a narrower range, resulting in a higher count at the histogram peak.

In this study, five different filtering methods were initially compared using non-optimized parameters, as illustrated in Figure 10. The purpose of this preliminary comparison was to establish a baseline and identify the potential of each filter. Based on the results, the main goal was to tune each filter's parameters to their optimal settings in order to maximize noise suppression within the defined motion model and determine the most effective filtering method. Although the graph in Figure 10 was generated using non-optimized parameters, visual inspection indicates that several of the peak noise artifacts present in the raw data are no longer visible in the filtered outputs. Nevertheless, for improved precision and valid comparative evaluation, parameter optimization is required for each filter

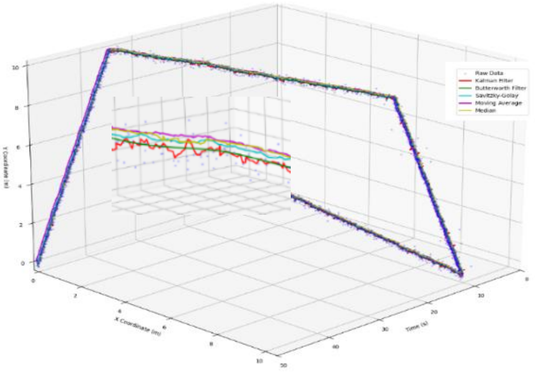


Figure 10. Preliminary Visual Analysis of UWB Positioning Data Filtered with Non-Optimized Parameters.

Building upon the observations in Figure 10, a preliminary assessment was performed to analyze the time-series error trends and statistical distributions of the filters under non-optimized parameter conditions. The graph presented in Figure 11 reflects these initial configurations and should therefore be interpreted as illustrative rather than definitive. Further improvements are expected through the optimization of each filtering method's parameters. At this stage, the definition and application of performance metrics become critically important. To evaluate filtering effectiveness, an ideal (noise-free) motion profile was used as the reference, and the deviation between the filtered signals and this reference was computed. The success rate was obtained by normalizing the absolute error and expressing the result as a percentage [31]. This approach enables a direct and quantitative comparison of filtering accuracy against the ground truth during parameter tuning. Accordingly, the filter type and parameter combination that best aligns with the reference trajectory can be objectively determined. In addition, regression analysis and RMSE were used to compute the overall performance score [32], from which the success rates of the filtering methods were derived.

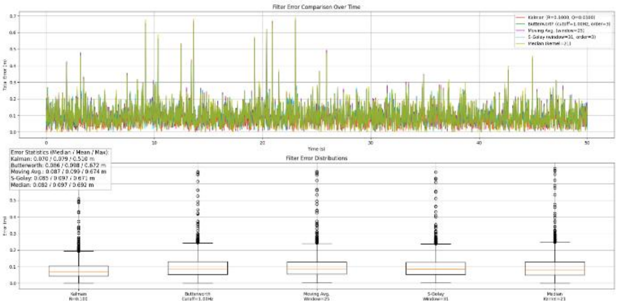
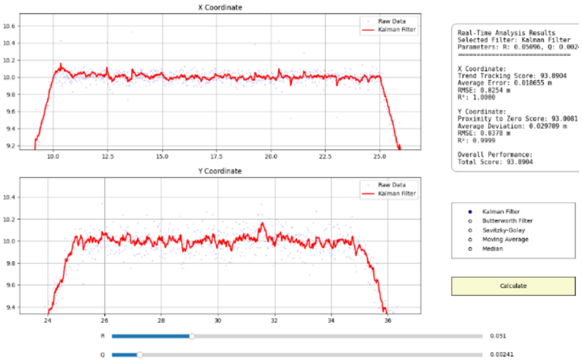


Figure 11. Error Trends and Statistical Distribution Comparison of Filtering Methods (Preliminary Parameter Evaluation).

According to the designed processing model, when the necessary parameter optimization is applied to the raw data, the resulting graphs exhibit significantly improved performance, as illustrated in Figures 12 to 16. Naturally, each filtering method has its own specific tuning parameters. By adjusting these parameters, the goal was to bring the total score as close as possible to 100%. The comparison of filters and performance evaluations are presented graphically in Figures 12 to 16 and are also supported by the Real-Time Analysis Results.

The Kalman Filter exhibited strong performance in mitigating low-frequency noise, primarily due to its dynamic state estimation based on prior measurements. This balance between prediction and observation resulted in a notable improvement in positional accuracy, particularly in slowly varying motion segments.



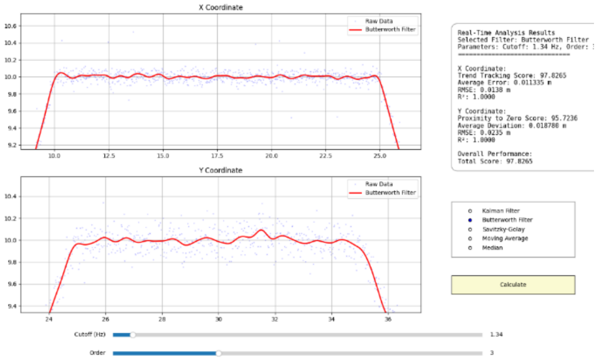
```
Real-Time Analysis Results
Selected Filter: Kalman Filter
Parameters: R: 0.05096, Q: 0.002
================================

X Coordinate:
Trend Tracking Score: 93.8904
Average Error: 0.018655 m
RMSE: 0.0254 m
R²: 1.0000

Y Coordinate:
Proximity to Zero Score: 93.0081
Average Deviation: 0.029709 m
RMSE: 0.0378 m
R²: 0.9999

Overall Performance:
Total Score: 93.8904
```

● Kalman Filter
○ Butterworth Filter
○ Savitzky-Golay
○ Moving Average
○ Median

R 0.051
Q 0.00241

Figure 12. Optimized output of the Kalman Filter following experimental parameter tuning.

The Butterworth Filter effectively attenuated high-frequency components due to its maximally flat response in the passband. With the optimized cutoff frequency, it maintained signal shape integrity while significantly reducing spectral noise, leading to smoother output without phase distortion.

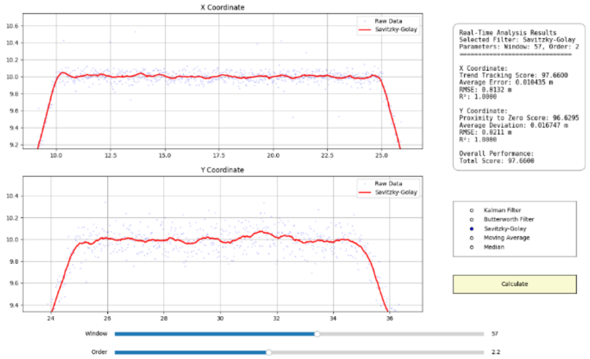


```
Real-Time Analysis Results
Selected Filter: Butterworth Filter
Parameters: Cutoff: 1.34 Hz, Order:
================================

X Coordinate:
Trend Tracking Score: 97.8265
Average Error: 0.011335 m
RMSE: 0.0138 m
R²: 1.0000

Y Coordinate:
Proximity to Zero Score: 95.7236
Average Deviation: 0.018788 m
RMSE: 0.0235 m
R²: 1.0000

Overall Performance:
Total Score: 97.8265
```

○ Kalman Filter
● Butterworth Filter
○ Savitzky-Golay
○ Moving Average
○ Median

Cutoff (Hz) 1.34
Order 3

Figure 13. Optimized output of the Butterworth Filter following experimental parameter tuning.



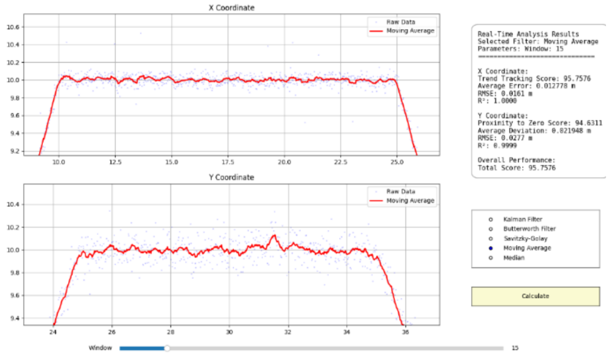
```
Real-Time Analysis Results
Selected Filter: Savitzky-Golay
Parameters: Window: 57, Order: 2
================================

X Coordinate:
Trend Tracking Score: 97.6600
Average Error: 0.010435 m
RMSE: 0.0132 m
R²: 1.0000

Y Coordinate:
Proximity to Zero Score: 96.6295
Average Deviation: 0.016747 m
RMSE: 0.0211 m
R²: 1.0000

Overall Performance:
Total Score: 97.6600
```

○ Kalman Filter
○ Butterworth Filter
● Savitzky-Golay
○ Moving Average
○ Median

Window 57
Order 2.2

Figure 14. Optimized output of the Savitzky-Golay Filter following experimental parameter tuning.

The Savitzky-Golay Filter provided a smooth output by fitting local polynomial functions, effectively preserving the structural characteristics of the original signal. It demonstrated strong performance in maintaining low- and mid-frequency components without distorting the overall waveform, making it particularly suitable for signals with gradual variations.



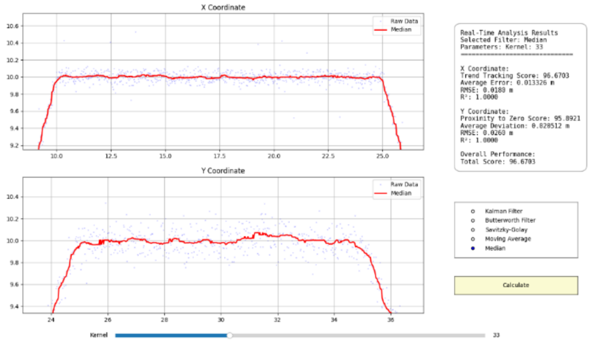
```
Real-Time Analysis Results
Selected Filter: Moving Average
Parameters: Window: 15
================================

X Coordinate:
Trend Tracking Score: 95.7576
Average Error: 0.012778 m
RMSE: 0.0161 m
R²: 1.0000

Y Coordinate:
Proximity to Zero Score: 94.6311
Average Deviation: 0.021948 m
RMSE: 0.0277 m
R²: 0.9999

Overall Performance:
Total Score: 95.7576
```

○ Kalman Filter
○ Butterworth Filter
○ Savitzky-Golay
● Moving Average
○ Median

Window 15

Figure 15. Optimized output of the Moving Average following experimental parameter tuning.

The Moving Average Filter effectively reduced random fluctuations and provided a smoothed signal trajectory, particularly along the X-axis. However, on the Y-axis, rapid transitions introduced noticeable output lag due to the inherent averaging mechanism, which incorporates a fixed-length window of prior samples. This temporal smoothing results in delayed responses during abrupt signal changes. While effective in suppressing noise, the filter is especially advantageous in removing low-frequency and continuous noise. However, it is limited in preserving signal details during sharp deviations.



```
Real-Time Analysis Results
Selected Filter: Median
Parameters: Kernel: 33
================================

X Coordinate:
Trend Tracking Score: 96.6703
Average Error: 0.013326 m
RMSE: 0.0183 m
R²: 1.0000

Y Coordinate:
Proximity to Zero Score: 95.8921
Average Deviation: 0.020512 m
RMSE: 0.0260 m
R²: 1.0000

Overall Performance:
Total Score: 96.6703
```

○ Kalman Filter
○ Butterworth Filter
○ Savitzky-Golay
○ Moving Average
● Median

Kernel 33

Figure 16. Optimized output of the Median Filter following experimental parameter tuning.

The Median filter effectively suppressed outliers, particularly in regions affected by impulsive spike noise, by replacing extreme values with the median of the surrounding window. This process yielded stable outputs across both X and Y coordinates. However, in segments with rapid signal transitions, the filter exhibited limited responsiveness due to its non-weighted, rank-based structure. While increasing the kernel size improved smoothness, it also caused a loss of fine-grained signal detail. These findings confirm that although the Median Filter is robust against sudden noise spikes, it is less suitable for applications requiring high sensitivity to subtle variations.

The total success score was calculated based on the deviation of the filtered points from the ideal motion coordinates, separately for the X and Y coordinate planes, and then combined to form an overall score. A major challenge in this evaluation is the absence of a standardized noise pattern, which prevents filter parameters from producing a universally valid solution set for all points. Furthermore, while errors in the Y-axis can be averaged and interpreted meaningfully, a clear reference point is often lacking along the X-axis. Therefore, it becomes essential to define a standardized motion path and to mathematically compute and verify the UWB position based on both time and trajectory. As a result, in some filters, certain parameter settings may yield highly accurate results in one coordinate plane but fail to achieve the same level of performance in the other.

3. Results and Discussion

After optimization, the performance of the filters was compared using metrics such as total success score, RMSE, and R^2. As a result of parameter optimization, the effectiveness of each filter was revealed, as presented in Table 6..

Table 6 provides a detailed performance comparison of the filtering methods by aggregating both overall accuracy indicators and error-specific statistical metrics. While the total score, RMSE, and R^2 values quantify the global effectiveness of each filter, the mean error, maximum error, and standard deviation offer insights into the variability and stability of the error patterns. This consolidated view allows for a direct and objective assessment of each filter's accuracy and robustness.

Table 6. Performance comparison of filtering methods based on Total Score, RMSE, and R^2 metrics.

Filter Type	Total Score (%)	RMSE (m)	R^2	Average Error (m)	Maximum Error (m)	Standard Deviation (m)
Kalman Filter	93.89039	0.04558	0.99996	0.038603	0.193185	0.024229
Butterworth Filter	97.82530	0.02723	0.99999	0.023799	0.109672	0.013232
S-Golay Filter	97.66003	0.02491	0.99999	0.021668	0.081425	0.012284
Moving Avg. Filter	95.75757	0.03204	0.99999	0.027707	0.131092	0.016097
Median Filter	96.67033	0.03166	0.99998	0.027286	0.095715	0.016066

In this study, the performance of several filtering techniques was systematically compared based on their ability to process UWB positioning data contaminated with mixed noise types. The dataset was composed of approximately 95% Gaussian noise, 4% high-frequency jitter, and 1% impulsive spike noise. Each filtering method was assessed in light of this composition to determine its effectiveness under real-world signal distortion conditions.

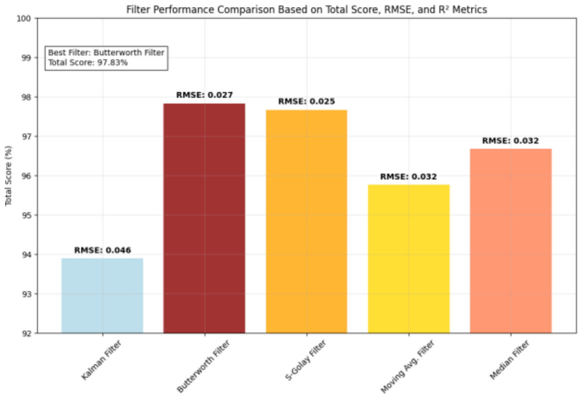


Figure 17. Comparative filter performance showing Total Score, RMSE, and R^2 values.

As summarized in Table 6 and Figure 17, the Butterworth Filter achieved the highest overall success rate of 97.8253%, demonstrating superior performance in suppressing high-frequency jitter. It also produced the lowest RMSE (0.016 m) and a perfect R^2 value (1.0000). The Savitzky-Golay Filter preserved the signal shape effectively, with a success rate of 97.66%. The Median Filter was particularly successful against impulsive spike noise, reaching 97.6603%. The Moving Average Filter, effective in reducing low-level continuous noise, recorded a success rate of 95.7575%. Although the Kalman Filter performed well against Gaussian noise, its overall success rate of 93.8903% remained the lowest among the evaluated methods.

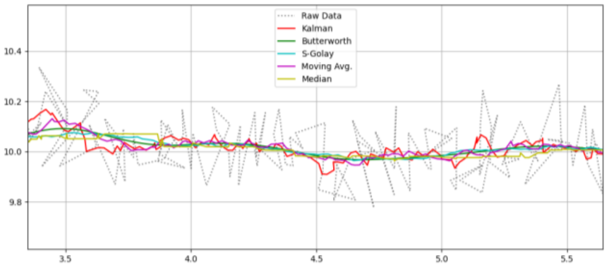


Figure 18. All filter responses compared over a limited data window.

To complement the numerical results, Figure 18 presents all filter responses over a limited data window, enabling a direct visual comparison of their smoothing behaviors under identical input conditions.

The results indicate that different filtering methods offer distinct advantages depending on the type of noise. Therefore, careful consideration must be given when selecting a filter, based on the noise characteristics and the intended application. The mathematical structure of the Butterworth and Savitzky-Golay filters appears to naturally align with the noise profile of the dataset, enabling them to deliver superior performance. Specifically, these filters demonstrated strong capabilities in suppressing dominant Gaussian and jitter components. Among them, the Butterworth Filter stood out due to its effectiveness in attenuating high-frequency jitter noise.

4. Conclusions and Future Works

This study proposed a comparative framework for evaluating multiple filtering techniques applied to UWB-based positioning data affected by a composite noise profile. The dataset incorporated approximately 95% Gaussian noise, 4% high-frequency jitter, and 1% impulsive spike noise. Each filter was assessed based on key performance indicators, including Total Score, RMSE, and R², in order to quantify their effectiveness under noisy signal conditions.

When evaluated under optimal parameter configurations, the Butterworth Filter exhibited the highest overall performance, achieving a 97.8253% success rate, an RMSE of 0.02723 m, and an R² value of 0.99999. This outcome is attributed to its strong capability in attenuating high-frequency jitter noise while maintaining signal fidelity. The Savitzky-Golay Filter followed with a 97.6600% success rate, primarily due to its ability to preserve the signal curvature. The Median Filter achieved 96.6703%, effectively suppressing impulsive noise components. The Moving Average Filter performed well in reducing low-amplitude continuous noise, reaching a success rate of 95.7575%. The Kalman Filter, while effective against Gaussian noise, resulted in the lowest overall success rate of 93.8904%, likely influenced by parameter sensitivity and the specific noise composition of the dataset.

The findings highlight that the performance of filtering techniques is directly influenced by the noise characteristics present in the data. Therefore, the proper selection of filters and optimization of their parameters is critical for enhancing the accuracy and reliability of UWB-based positioning systems. These results provide a valuable roadmap for researchers and practitioners seeking to improve positioning accuracy under various noise conditions.

Despite offering a controlled and reproducible evaluation framework, this study has several limitations that should be acknowledged. All filtering operations were conducted offline using pre-recorded UWB data; therefore, latency, processing time, and system load were not considered. The dataset was synthetically degraded by injecting Gaussian, jitter, and spike noise, which, while useful for testing in controlled scenarios, may not fully capture the complexity and temporal variability of real-world sensor noise. The motion scenario was limited to a single 10×10 m environment with constant-speed linear trajectories and only two anchor nodes; more dynamic behaviors such as rotations or velocity shifts were not included. In addition, filter parameters were optimized manually for this specific scenario, which may reduce generalizability to other systems where such fine-tuning is impractical. The analysis was also limited to UWB signals in two-dimensional (X–Y) space, without incorporating other sensor modalities such as IMU, GNSS, or LiDAR, nor accounting for 3D positioning or height variations. As such, further research is needed to extend the findings to multi-sensor, real-time, and three-dimensional localization systems.

Future research will focus on further optimizing the evaluated filters and exploring hybrid filtering strategies that combine the strengths of multiple methods. In addition, the proposed framework will be extended to real-time applications using live UWB data and dynamically evolving motion trajectories.

Acknowledgement

The authors declare that there is no funding or external support for this research.

Author’s Contributions

Levent TÜRKLER: Was responsible for the design and planning of the study, conducted the experiments, performed the analysis of the data and results, and drafted the manuscript.

Lütfiye Özlem AKKAN: Assisted in the design of the study, carried out the literature review, contributed to result interpretation, and edited the manuscript for final submission.

Ethics

There are no ethical issues for the publication of this manuscript.

References

[1]. Wang, W., Guo, Y., Huang, B., Zhao, G., Liu, B., Wang, L. 2011. Analysis of Filtering Methods for 3D Acceleration Signals in Body Sensor Network. *Proceedings of the International Symposium on Bioelectronics and Bioinformations*. Suzhou, China, pp. 263–266. https://doi.org/10.1109/ISBB.2011.6107697.

[2]. Ullah, I., Shen, Y., Su, X., Esposito, C., Choi, C. 2020. A localization based on unscented Kalman Filter and particle filter localization algorithms. *IEEE Access*, 8: 2233–2246. https://doi.org/10.1109/ACCESS.2019.2961740.

[3]. Xu, Y., Shmaliy, Y.S., Li, Y., Chen, X. 2017. UWB-based indoor human localization with time-delayed data using EFIR filtering. *IEEE Access*, 5: 16676–16683. https://doi.org/10.1109/ACCESS.2017.2743213.

[4]. Hurtado-Perez, A.E., Toledano-Ayala, M., Cruz-Albarran, I.A., Lopez-Zúñiga, A.; Moreno-Perez, J.A., Álvarez-López, A., Rodriguez-Resendiz, J., Perez-Ramirez, C.A. 2025. Use of technologies for the acquisition and processing strategies for motion data analysis. *Biomimetics*, 10: 339. https://doi.org/10.3390/biomimetics10050339.

[5]. Borhan, N.; Saleh, I.; Rahiman, W. 2024. Comparative analysis of filtering techniques for AGV indoor localization with ultra-wideband technology. *Pertanika Journal of Science & Technology*, 32: 2151–2164, https://doi.org/10.47836/pjst.32.5.13.

[6]. Kim, Y., Bang, H. 2019. Introduction to Kalman Filter and Its Applications. In: Govaers, F. (ed.) *Introduction and Implementations of the Kalman Filter*; IntechOpen, London, UK. ISBN 978-1-83880-536-4.

[7]. You, W., Li, F., Liao, L., Huang, M. 2020. Data fusion of UWB and IMU based on unscented Kalman Filter for indoor localization of quadrotor UAV. *IEEE Access*, 8: 64971–64981. https://doi.org/10.1109/ACCESS.2020.2985053.

[8]. Bai, Y., Yan, B., Zhou, C., Su, T., Jin, X. 2023. State of art on state estimation: Kalman Filter driven by machine learning. *Annual Reviews in Control*, 56: 100909. https://doi.org/10.1016/j.arcontrol.2023.100909.

[9]. Garcia, R.V., Pardal, P.C.P.M., Kuga, H.K., Zanardi, M.C. 2019. Nonlinear filtering for sequential spacecraft attitude estimation with real data: Cubature Kalman Filter, Unscented Kalman Filter and Extended Kalman Filter. *Advances in Space Research*, 63: 1038–1050. https://doi.org/10.1016/j.asr.2018.10.003.

[10]. Ma, W., Wang, C., Dang, L., Zhang, X., Chen, B. 2023. Robust dynamic state estimation for DFIG via the generalized maximum correntropy criterion Ensemble Kalman Filter. *IEEE Transactions on Instrumentation and Measurement*, 72: 1–13. https://doi.org/10.1109/TIM.2023.3328095.

[11]. Li, Q., Li, R., Ji, K., Dai, W. 2015. Kalman Filter and its application. *Proceedings of the 8th International Conference on Intelligent Networks and Intelligent Systems (ICINIS),* Tianjin, China, pp. 74–77. https://doi.org/10.1109/ICINIS.2015.35.

[12]. Romanenko, A., Castro, J.A.A.M. 2004. The unscented filter as an alternative to the EKF for nonlinear state estimation: a simulation case study. *Computers & Chemical Engineering*, 28: 347–355. https://doi.org/10.1016/S0098-1354(03)00193-5.

[13]. Piskorowski, J. 2006. Phase-compensated time-varying Butterworth filters. *Analog Integrated Circuits and Signal Processing,* 47: 233–241. https://doi.org/10.1007/s10470-006-5255-9.

[14]. Manal, K., Rose, W. 2007. A general solution for the time delay introduced by a low-pass Butterworth digital filter: an application to musculoskeletal modeling. *Journal of Biomechanics,* 40: 678–681. https://doi.org/10.1016/j.jbiomech.2006.02.001.

[15]. Robertson, D.G.E., Dowling, J.J. 2003. Design and responses of Butterworth and critically damped digital filters. *Journal of Electromyography and Kinesiology,* 13: 569–573. https://doi.org/10.1016/S1050-6411(03)00080-4.

[16]. Bharathi, M., Akshitha, K., Divyashree, M., Nagakrupa, B., Sindhu, J. 2021. OTA based 2nd order Butterworth filter for mobile communication using CMOS technology. *Proceedings of the First International Conference on Advanced Scientific Innovation in Science, Engineering and Technology (ICASISET),* Chennai, India, pp. 1–7. http://doi.org/10.4108/eai.16-5-2020.2304039.

[17]. Schmid, M., Rath, D., Diebold, U. 2022. Why and how Savitzky–Golay filters should be replaced. *ACS Measurement Science Au,* 2: 185–196. https://doi.org/10.1021/acsmeasuresciau.1c00054.

[18]. Sadeghi, M., Behnia, F., Amiri, R. 2020. Window selection of the Savitzky–Golay filters for signal recovery from noisy measurements. *IEEE Transactions on Instrumentation and Measurement*, 69: 5418–5427. https://doi.org/10.1109/TIM.2020.2966310.

[19]. Savitzky, A., Golay, M.J.E. 1964. Smoothing and differentiation of data by simplified least squares procedures. *Analytical Chemistry*, 36: 1627–1639. https://doi.org/10.1021/ac60214a047.

[20]. Huang, T., Yang, G., Tang, G. 1979. A fast two-dimensional median filtering algorithm. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 27: 13–18. https://doi:10.1109/TASSP. 1979.1163188.

[21]. Klenk, J., Becker, C., Lieken, F., Nicolai, S., Maetzler, W., Alt, W., et al. 2011. Comparison of acceleration signals of simulated and real-world backward falls. *Medical Engineering & Physics*, 33: 368–373. https://doi.org/10.1016/j.medengphy.2010.11.003.

[22]. Golestan, S., Ramezani, M., Guerrero, J.M., Freijedo, F.D., Monfared, M. 2014. Moving average filter based phase-locked loops: performance analysis and design guidelines. *IEEE Transactions on Power Electronics,* 29: 2750–2763. https://doi.org/10.1109/TPEL.2013.2273461.

[23]. Wang, J., Liang, J., Gao, F., Zhang, L., Wang, Z. 2015. A method to improve the dynamic performance of moving average filter-based PLL. *IEEE Transactions on Power Electronics*, 30: 5978–5990. https://doi.org/10.1109/TPEL.2014.2381673.

[24]. Freijedo, F.D., Doval-Gandoy, J., López, Ó., Acha, E. 2009. A generic open-loop algorithm for three-phase grid voltage/current synchronization with particular reference to phase, frequency, and amplitude estimation. *IEEE Transactions on Power Electronics,* 24: 94–107. https://doi.org/10.1109/TPEL.2008.2005580.

[25]. Van Herbruggen, B., Luchie, S., Wilssens, R., De Poorter, E. 2024. Single anchor localization by combining UWB angle-of-arrival and two-way-ranging: an experimental evaluation of the DW3000. *Proceedings of the International Conference on Localization and GNSS (ICL-GNSS)*, Antwerp, Belgium, pp. 1–7. https://doi.org/10.1109/ICL-GNSS60721.2024.10578498.

[26]. Niculescu, V., Palossi, D., Magno, M., Benini, L. 2023. Energy-efficient, precise UWB-based 3-D localization of sensor nodes with a nano-UAV. *IEEE Internet of Things Journal*, 10: 5760–5777. https://doi.org/10.1109/JIOT.2022.3166651.

[27]. Makerfabs. ESP32 UWB Pro with Display. https://www.makerfabs.com/esp32-uwb-pro-with-display.html (accessed at 23.02.2025).

[28]. Silva, B., Pang, Z., Akerberg, J., Neander, J., Hancke, G. 2014. Experimental study of UWB-based high precision localization for industrial applications. *Proceedings of the IEEE International Conference on Ultra-WideBand (ICUWB),* Paris, France, pp. 280–285. https://doi.org/10.1109/ICUWB.2014.6958993

[29]. Gonzalez, R.C., Woods, R.E. 2018. *Digital Image Processing*; 4th ed., Pearson Education: New York, USA: Pearson Education.

[30]. Hagedorn, J., Alicke, F., Verma, A. 2017. How to measure total jitter. *Texas Instruments Application Note,* SCAA120B.

[31]. Alt, H., Godau, M. 1995. Computing the Fréchet distance between two polygonal curves. *International Journal of Computational Geometry & Applications,* 5: 75–91. https://doi.org/10.1142/S0218195995000064.

[32]. Chicco, D., Warrens, M.J., Jurman, G. 2021. The coefficient of determination R-Squared is More informative than SMAPE, MAE, MAPE, MSE and RMSE in regression analysis evaluation. *PeerJ Computer Science,* 7: e623. https://doi.org/10.7717/peerj-cs.623.