Data Engineering and Applications

Research Paper

Comparative Study of Kalman Filtering and Particle Filtering in Sensor Localization Systems

Jolanta Gromek[1], and Renata Dąbrowska[1, *]

[1] Faculty of Computer Science, University of Technology and Life Sciences, Bydgoszcz, 85-796, Poland
*Corresponding author: renata.d@utp.edu.pl

Abstract. In many data-driven applications in engineering and scientific fields, accurately locating sensor nodes is crucial for the reliability and performance of wireless sensor networks. This paper systematically compares the performance of two excellent state estimation algorithms and studies their application in sensor localization. This study used real datasets and synthetic datasets to evaluate various metrics. These metrics include convergence speed, root mean square error, mean absolute error, robustness to noise and node failures, computational cost, and scalability. Experimental results show that both methods achieved sub-meter accuracy in low-noise environments. In high-noise, multipath, and irregular sensor data conditions, particle filters are generally more effective than Kalman filters. Compared to the Kalman filter, particle filters can reduce localization errors by 15% to 30% in high-noise environments. Even with more than 40% of the sensor nodes failing, they can still operate normally. The aforementioned advantages come with higher computational costs and memory usage, making them unsuitable for all-weather system design. This study conducted an in-depth analysis of the advantages and disadvantages of each algorithm and demonstrated the application scenarios of complex sensor networks. This study provides a reference point for selecting suitable decentralized estimation methods for industrial and scientific applications.

Keywords: *Computer Localization, State Estimation, Wireless Sensor Networks, Kalman Filter, Particle Filter, Robustness, Data Fusion, Error Analysis*

Received on 28 October 2024, Accepted on 12 April 2025, Published on 20 April 2025

Introduction

With the expansion of wireless sensor networks (WSN), many new data-driven system operation modes have begun to appear in various fields, such as environmental monitoring, smart factories, and smart city construction [1]. Sensor nodes are distributed within these networks to collect and transmit spatial and temporal data, enabling advanced situational awareness and analysis [2]. The effectiveness of the sensor system depends on the precise location of each sensor node. If this is not done, the sensor system will not be able to process the collected data and use it for routing, data fusion, etc. [3] Due to various propagation conditions, unstable communication links, and severe energy and computational constraints, localization becomes very difficult [4]. Traditional methods cannot solve difficult problems, such as non-line-of-sight, dynamic interference, and multipath effects [5]. With the increasing demand in the fields of engineering and science for large-scale, real-time, and robust sensor networks, high-performance sensor localization algorithms are now being used to locate sensors [6]. By accurately setting the location of the collected data, an intelligent, independent, and distributed system can be established [7]. Therefore, further research in this field will provide theoretical and practical support to advance the development of next-generation sensor technology [8].

Recently, a large number of positioning methods using geometric and probabilistic frameworks have been studied. These methods include complex filtering algorithms, multilateration, and anchor-free schemes [9]. In dynamic systems with data uncertainty and noisy measurements, the Kalman filter (KF) and its nonlinear

https://doi.org/10.64972/dea.2025.v4i2.1983d:30-42

extensions are very suitable for state estimation [10]. In a linear Gaussian environment, the Kalman filter (KF) is the most effective. However, particle filters (PF) are considered a better choice because wireless sensor networks (WSN) in the real world exhibit a significant amount of nonlinearity and non-Gaussian noise [11]. Although Monte Carlo sampling is used to simulate various distributions, PFs generally require more computational cost and resources [12]. The lack of accuracy and generalization to different environments are currently unresolved issues [13]. Despite significant progress, direct and in-depth comparisons of these filtering schemes are still relatively uncommon due to the limitations imposed by sensor networks [14]. In addition, other issues need to be considered, such as how to handle outliers, the convergence of the filters, and the impact of filter design choices on scalability and real-time performance [15]. Therefore, more research is needed on all types of WSNs to determine the best algorithms and deployment modes through benchmarking and comprehensive evaluation [16].

In this paper, we will compare in detail the methods of particle filters and Kalman filters in sensor localization. This study accurately tested how these algorithms handle real and simulated data in typical modern sensor networks. We studied the convergence speed of performance metrics, localization error, and robustness, as well as their practical applications, computational efficiency, scalability, and integration feasibility. We identified the main advantages and disadvantages of all these methods through specific experiments and analyzes. Finally, this study provides researchers and practitioners with some basic information and relevant recommendations regarding high-quality, high-precision, and high-efficiency positioning services in the practical application of sensor networks.

Fundamental Principles and Algorithms of Filtering

Overview of State Estimation Methods

State estimators are the foundation of modern sensor localization, capable of measuring data to infer the hidden states of a system and making measurements based on noise in complex environments [17]. At the same time, in many engineering fields, scholars and practitioners are researching efficient methods to process data in order to address the issues and uncertainties that sensor networks face in the real world [18]. Deterministic attempts, such as least squares fitting and geometric triangulation, are theoretically feasible, but they are very sensitive to noise and environmental changes [19]. As research progresses, the probabilistic methods that explicitly model the random evolution of system states have been replaced by probabilistic methods that account for the uncertainties introduced by sensing and execution limitations.

Using Bayesian filtering is a significant advancement, as it provides a formal foundation for recursively estimating the state of stochastic processes [20]. The Bayesian method maintains a belief distribution over possible scenarios and continuously updates it based on recent discoveries. The changes in the aforementioned paradigm can be addressed by using real-time data in the filtering algorithm to cope with both linear and nonlinear system behaviors as well as various observation noises. Over time, Bayesian filters have gradually become the foundation of sensor localization research. Various methods have been developed to achieve results that are easy to use, accurate, and robust to model defects. Therefore, almost all modern sensor network state estimation methods (including hybrid methods, Kalman filtering, and particle filtering) are derived from probability theory.

Kalman Filter: Mathematical Foundations

Linear systems with Gaussian noise characteristics typically use Kalman filters (KFs) as state estimation algorithms [21]. For recursive and real-time operation, the KF uses a mathematical model to predict changes in the system state, and then modifies the prediction based on noisy sensor data [22]. Through prediction and update iterations, both flexibility and rapid convergence are achieved; therefore, by adding new data points, the uncertainty of the estimator is reduced. Under operating conditions, the Kalman Filter (KF) has advantages as an MMSE estimator and is computationally simple.

For the above reasons, the Kalman filter has been widely used in real-time positioning for devices with limited time and resources [23]. In the field of mobile robotics, vehicle localization and infrastructure monitoring are common applications. These applications utilize the fast convergence and low memory usage of the Kalman filter for real-time state estimation. When the underlying physical process is only slightly nonlinear, extended algorithms such as the Extended Kalman Filter (EKF) and the Unscented Kalman Filter (UKF) can be used to linearize around the operating point or use sigma point sampling. However, these methods are still primarily based on the assumption of unimodal, approximately Gaussian uncertainties [24]. Despite the aforementioned advancements, the efficiency of Kalman-based methods may significantly decrease when the system and noise models deviate markedly from linearity or when outliers and non-Gaussian effects are prominent. Therefore, there is an urgent need for a general framework to handle these situations.

Particle Filter: Probabilistic Framework

Using Bayesian state estimation sampling methods, particle filtering techniques directly address the limitations of the linear-Gaussian assumption [25]. Particle filters approximate the complete posterior distribution of the system state using a set of discrete samples called "particles," rather than using a set of parametric statistics to represent the belief state. The system dynamic model propagates each particle separately and then reweights them based on their interpretation of the recently observed values to identify multimodal uncertainty and perform nonlinear inference.

For the above reasons, particle filters have been chosen to address various issues in sensor localization, such as motion uncertainty and non-Gaussian noise measurements [26]. Due to their good performance in the presence of uncertainty, particle filters have recently been applied to many problems, such as autonomous driving and object tracking in complex environments. In practical applications, particle filters also face some issues. Sampling the posterior distribution in high-dimensional space requires a large number of particles, which can lead to high computational costs. Additionally, algorithmic design is needed to address the problem of sample impoverishment, such as system resampling and particle rejuvenation. As the scale and complexity of sensor networks continue to increase, particle filters have theoretically been proven to be a universal solution for recursive estimation. It is necessary to strictly manage computational constraints and enable the algorithm to adapt to embedded and distributed systems.

Implementation, Complexity, and System Integration

Computational Efficiency and Algorithm Process

Computational efficiency and predictable runtime are the requirements for deploying embedded or edge sensor positioning platforms. Prediction and correction are the two typical stages of a Kalman filter, both of which achieve the aforementioned goals.

In the prediction phase, the state at time k can be obtained by propagating the previous state through a linear system model or a nonlinear system model:

$$\boldsymbol{x}_{\boldsymbol{k}|\boldsymbol{k}-\boldsymbol{1}} = \boldsymbol{f}(\boldsymbol{x}_{\boldsymbol{k}-\boldsymbol{1}|\boldsymbol{k}-\boldsymbol{1}}, \boldsymbol{u}_{\boldsymbol{k}-\boldsymbol{1}}) + \boldsymbol{\nu}_{\boldsymbol{k}-\boldsymbol{1}} \quad \text{Eq.(1)}$$

Where $\boldsymbol{\nu}_{k-1}$ is the process noise, and $f(\cdot)$ may be a strictly linear function or a nonlinear dynamic operator. The covariance update simultaneously considers the accumulated prediction error and the dynamic error of the system:

$$\boldsymbol{P}_{\boldsymbol{k}|\boldsymbol{k}-\boldsymbol{1}} = \boldsymbol{F}_{\boldsymbol{k}}\boldsymbol{P}_{\boldsymbol{k}-\boldsymbol{1}|\boldsymbol{k}-\boldsymbol{1}}\boldsymbol{F}_{\boldsymbol{k}}^{\top} + \boldsymbol{L}_{\boldsymbol{k}}\boldsymbol{Q}_{\boldsymbol{k}-\boldsymbol{1}}\boldsymbol{L}_{\boldsymbol{k}}^{\top} \quad \text{Eq.(2)}$$

At this state, $\mathbf{F}_k$ is the Jacobian matrix of the process model, and $\mathbf{L}_k$ explicitly projects the process noise into the estimation subspace to ensure the retention of all independent and cross terms. This improves convergence and robustness to model uncertainty.

In the update (correction) phase, the Kalman gain integrates the new sensor measurements:

$$\boldsymbol{K}_{\boldsymbol{k}} = \boldsymbol{P}_{\boldsymbol{k}|\boldsymbol{k}-\boldsymbol{1}}\boldsymbol{H}_{\boldsymbol{k}}^{\top}(\boldsymbol{H}_{\boldsymbol{k}}\boldsymbol{P}_{\boldsymbol{k}|\boldsymbol{k}-\boldsymbol{1}}\boldsymbol{H}_{\boldsymbol{k}}^{\top} + \boldsymbol{R}_{\boldsymbol{k}})^{-\boldsymbol{1}} \quad \text{Eq.(3)}$$

According to the innovation's signal-to-noise ratio and the prediction uncertainty, the weight matrix is used to determine how much to give to the innovation. Since the matrix $\mathbf{H}_k$ is the linearization of the measurement model, and $\mathbf{R}_k$ is the time-varying noise model of the sensor, its performance in the actual system will change over time.

https://doi.org/10.64972/dea.2025.v4i2.1983d:30-42

In contrast, particle filters (sequential Monte Carlo methods) are used for high-dimensional, nonlinear, or multimodal posteriors. The belief state is expressed in the form of weighted empirical distribution samples. The state evolution of each randomly sampled particle is as follows:

$$x_k^{(i)} \sim p\left(x_k \mid x_{k-1}^{(i)}, u_{k-1}\right) \qquad \text{Eq.(4)}$$

Propagate all particles, and then each receives a new weight according to how well it explains the new data:

$$w_k^{(i)} \propto w_{k-1}^{(i)} \cdot p\left(z_k \mid x_k^{(i)}\right) \qquad \text{Eq.(5)}$$

The resampling process avoids particle deprivation by duplicating high-probability particles and discarding less likely ones, thereby ensuring an effective sample size and preventing degradation.

Algorithm complexity comparison: For an n -state system, the complexity of the Kalman filter during each update is $O(n^3)$, due to the matrix inversion. However, the complexity can be further reduced by utilizing structural properties such as sparsity or block diagonal forms. The complexity of the particle filter is $O(Nn)$, where N is the number of particles, linearly related to the set size, but for high fidelity, it may be exponentially related to the state space. Therefore, embedded applications require GPU/FPGA acceleration or clever scheme design. Figure 1 shows the above process. Particle filters use random sampling, individual weight calculation, and periodic resampling, while Kalman filters use deterministic matrix operations. Based on the system's computational capacity and other constraints, choose the appropriate method.

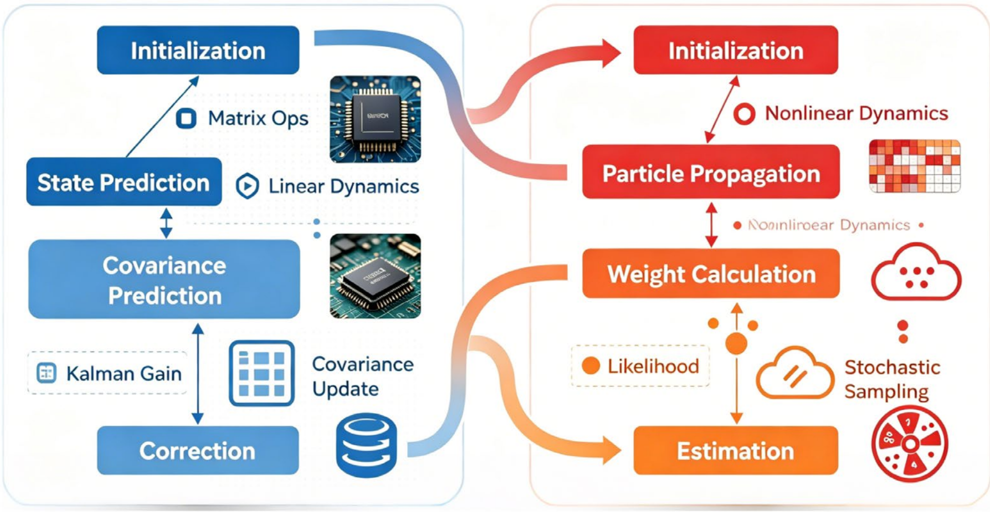

Figure 1. Process flow comparison of Kalman and particle filters.

Scalability and Real-Time Performance

As sensor networks expand, the issues of system design scalability and strict real-time performance are becoming increasingly severe. A paradigm of distributed Kalman filters assigns local estimators to each network node and uses local data to independently propagate and correct state estimates:

$$x_{k|k}^{(j)} = x_{k|k-1}^{(j)} + K_k^{(j)}\left(z_k^{(j)} - H^{(j)} x_{k|k-1}^{(j)}\right) \qquad \text{Eq.(6)}$$

This local autonomy significantly reduces the bandwidth and latency of communication between nodes, which is particularly evident in sensor networks. Regularly merge local results to ensure overall system consistency and appropriate state inference. The transition from local to global is as follows:

$$x_{k|k}^{global} = \sum_{j=1}^{M} \beta_j x_{k|k}^{(j)} \qquad \text{Eq.(7)}$$

The fusion weight β_j adjusts inaccurate results because it reflects trust in local estimators, recent measurement quality, and previous system reliability. In the case of redundant observations or field of view overlap, the uncertainty cross-correlation between distributed nodes propagates as follows:

$$P_{k|k-1}^{(j,l)} = F_k^{(j)} P_{k-1|k-1}^{(j,l)} F_k^{(l)\top} \qquad \text{Eq.(8)}$$

Including the dynamics of the coupled system and the changes at each node.

The size of the particle filter depends on the number of available particles and parallelism. The optimal sample size after sensor fusion:

$$N_{eff} = \frac{1}{\sum_{i=1}^{N} \left(w_k^{(i)}\right)^2} \quad \text{Eq.(9)}$$

Determine the resampling time. To prevent the filter from collapsing, a relatively large N_{eff} should be maintained. In addition, partitioning the particle set can be used to handle high-throughput distributed inference in network nodes or clusters.

To handle high-frequency, high-bandwidth sensor data, hardware-assisted parallelization (such as SIMD, multi-core CPUs, FPGAs, and GPUs) is used to meet the demand for short response times in dynamic, bandwidth-constrained environments.

As shown in Figure 2, the scalable distributed localization architecture includes hierarchical fusion, local computation, and network flows of raw and synthesized information.

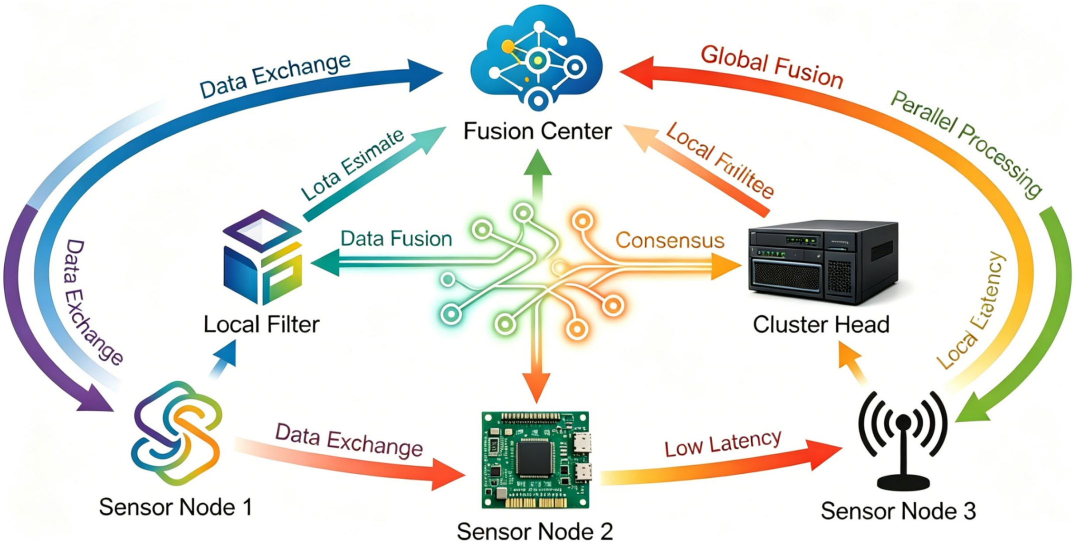

Figure 2. Distributed and hierarchical estimation system architecture.

Data Fusion and Application Scenarios

Multi-sensor localization combines data collected by sensors in different locations and tasks to improve the reliability and accuracy of system position estimation. Under the assumption of Gaussian noise, the weighted least squares estimator is the fundamental structure for centralized fusion, as it most effectively combines sensor measurements:

$$\hat{x}_k = \left(\sum_{m=1}^{M} H^{(m)\top} R_k^{(m)-1} H^{(m)}\right)^{-1} \left(\sum_{m=1}^{M} H^{(m)\top} R_k^{(m)-1} z_k^{(m)}\right) \quad \text{Eq.(10)}$$

Here, the global state estimate $\hat{\mathbf{x}}_k$ is normalized by the measurement matrix $\mathbf{H}^{(m)}$ of each sensor and the corresponding observation noise covariance $\mathbf{R}_k^{(m)}$. This combines all available sensor measurements $\mathbf{z}_k^{(m)}$. Due to its reliable low-noise data source, the first structure receives a higher weight.

In distributed and consensus-based fusion architectures, each node maintains a local state estimate and continuously improves this estimate by collecting data from communications with neighboring nodes.

$$x_k^{(j)} = \sum_{l \in \mathcal{N}_j} \alpha_{jl} x_k^{(l)} \quad \text{Eq.(11)}$$

At this point, the convex combination weights α_{jl} update the local estimates to facilitate the dissemination of information in large-scale networks while ensuring robustness against node loss and communication failures.

https://doi.org/10.64972/dea.2025.v4i2.1983d:30-42

The fusion at the covariance level aids in the estimation of independent sensors and improves accuracy. The optimal linear fusion of multiple Gaussian estimates can be expressed as an information filter, with means $\mathbf{x}_k^{(m)}$ and covariance matrices $\mathbf{P}_k^{(m)}$:

$$\boldsymbol{x}_{\boldsymbol{k}}^{\boldsymbol{fused}} = \left(\sum_{m=1}^{M} \boldsymbol{P}_{\boldsymbol{k}}^{(m)-1} \right)^{-1} \left(\sum_{m=1}^{M} \boldsymbol{P}_{\boldsymbol{k}}^{(m)-1} \boldsymbol{x}_{\boldsymbol{k}}^{(m)} \right) \quad \text{Eq.(12)}$$

The corresponding fused covariance is:

$$\boldsymbol{P}_{\boldsymbol{k}}^{\boldsymbol{fused}} = \left(\sum_{m=1}^{M} \boldsymbol{P}_{\boldsymbol{k}}^{(m)-1} \right)^{-1} \quad \text{Eq.(13)}$$

Therefore, the confidence level of each estimator is clearly visible. In addition, under the assumptions of linearity and Gaussianity, the minimum mean square error of the fusion results is guaranteed.

When there are different types or non-Gaussian distributions in the sensor network, the individual likelihoods of these observations are combined. This is done to determine the joint likelihood of each particle in the particle filter.

$$\boldsymbol{p}\left(\boldsymbol{z}_{\boldsymbol{k}} \mid \boldsymbol{x}_{\boldsymbol{k}}^{(i)}\right) = \prod_{m=1}^{M} \boldsymbol{p}\left(\boldsymbol{z}_{\boldsymbol{k}}^{(m)} \mid \boldsymbol{x}_{\boldsymbol{k}}^{(i)}\right) \quad \text{Eq.(14)}$$

By using the Bayesian update process and importance weighting and resampling, the particle set can get very close to the true posterior distribution. A nonlinear, multimodal, and even non-parametric measurement model can be represented by each term $p\left(\mathbf{z}_k^{(m)} \mid \mathbf{x}_k^{(i)}\right)$. This model can integrate visual, radio frequency, inertial data, and other sensor data.

Comparative Performance Evaluation

Evaluation Metrics and Benchmark Scenarios

Testing the accuracy, repeatability, and stability of the positioning algorithm. To ensure the fairness and scientific validity of the comparison, this study will use familiar performance metrics and typical benchmark scenarios to account for the differences between real-world environments [27].

Root Mean Square Error (RMSE) is a general method for measuring positioning accuracy, and it is very sensitive to large errors; therefore, cases that deviate significantly from the expected position will be severely penalized [28]. Mean Absolute Error (MAE) is also used to assess the degree of bias in estimates; it is more robust to operational stability and more sensitive to rare events [29]. The aforementioned quantitative indicators can be used to conduct comparative analysis of the results obtained by various estimators at different time points.

The analysis of the cumulative error distribution can also be used to show the frequency and likelihood of different sizes of errors appearing in the evaluation set. Using this method can reveal potential flaws in the algorithm, such as occasional large estimation errors or the inability to maintain a high confidence interval around the median error [30]. In addition, box plot statistics can show the range of error distribution, median, and outliers at different time points in the experiment. The data obtained using this statistical method indicate whether the estimates are stable during the positioning process and to what extent they are stable [31].

The benchmark scenarios for evaluation come from real-world datasets and controlled synthetic testing platforms. Parameter synthesis can generate various noise settings, such as sensor faults, dynamic range changes, noise level variations, and counter-interference. Therefore, the tested algorithms can be evaluated far beyond their expected operational range and still produce valid results under harsh conditions [32]. However, multipath propagation, non-line-of-sight conditions, temporal variations, and hardware-induced uncertainties are common complexities in evaluating datasets in the real world. The aforementioned environments cover many issues related to compliance in field tests and actual operations, which frequently occur in real-time deployments.

https://doi.org/10.64972/dea.2025.v4i2.1983d:30-42

The aforementioned metrics and reference scenarios will be used to evaluate performance. They can be used to study error characteristics, convergence behavior, and fault immunity. The open and reproducible basis for comparing results and substantive analysis in the subsequent chapters is the aforementioned protocol.

Simulation and Experimental Results

This section will present a comparison between the simulation results and the actual experimental deployment to ensure the stability and feasibility of the evaluated algorithm. Based on the above reasons, the benchmark scenarios and evaluation results will be presented in this manner. These charts are designed to help us quickly and reasonably determine the quality of the estimates, as well as whether they converge, are stable, and are suitable for various situations [33].

Figure 3 shows the main positioning error characteristics of the candidate estimators. To measure the RMSE curves under different noise levels, as shown in Figure 3(a), under high noise conditions, the steady-state error of the particle filter is lower, while the Kalman filter performs excellently under moderate noise conditions. For example, when the noise level increases from 0.01 to 1.0, the root mean square error (RMSE) of the Kalman filter rises from approximately 0.3 meters to over 0.5 meters, while the particle filter remains below 0.45 meters throughout the process. Therefore, in high noise conditions, its error is approximately 15% to 30%.

Figure 3(b) shows the comparison of MAE values for key deployment scenarios, including dynamic outdoor environments and controlled laboratory tests. For example, in factory and multipath environments, the MAE can reach 0.35 meters; however, in laboratory environments, the accuracy of both algorithms is below 0.15 meters. In addition, the particle filter is always 10-20% lower than the Kalman filter. The above analysis indicates that under conditions of rich multipath and non-line-of-sight (NLOS), the MAE is generally larger. On the other hand, the method using particle filters is more robust to environmental changes. The histogram of cumulative errors is shown in Figure 3(c). As shown above, the tail probability of the particle filtering scheme is relatively low. The advanced filter effectively suppresses the probability of rare but extreme estimation slip errors, keeping over 80% of the errors within 0.4 meters. Figure 3(d) shows the main differences between the algorithms, displaying the variance and frequency of outliers under environmental changes. The factory and multipath environments have more outliers and a larger interquartile range, indicating that their estimation becomes more challenging.

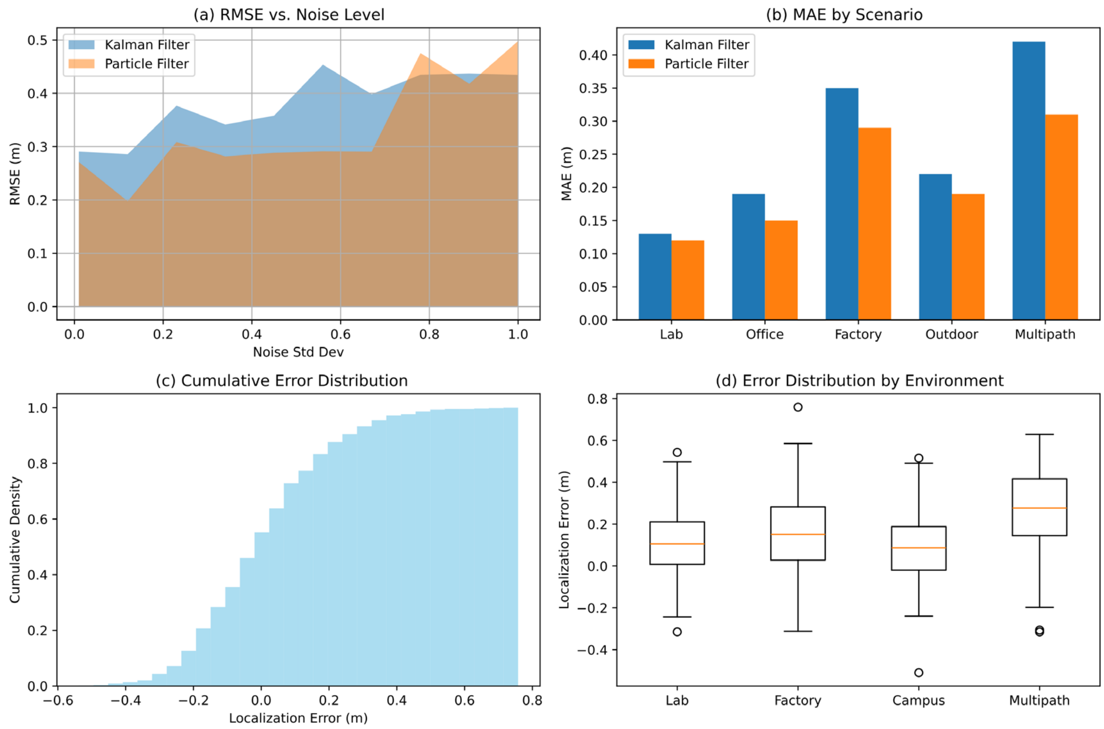

Figure 3. Localization error analysis. (a) RMSE curve under different noise levels. (b) MAE bar chart under varying scenarios. (c) Cumulative error distribution histogram. (d) Box plot for different environments.

Figure 4 shows the robustness and convergence of the study. Comparison of convergence rates: Figure 4(a) shows the relationship between the normalized error reduction of all algorithms and the iteration time. It is worth noting that the distributed particle filter exhibits a faster initial convergence speed under high noise or non-stationary conditions. It achieves a normalized error of less than 0.25 in fewer than ten iterations and consistently outperforms the Kalman filter baseline. Figure 4(b) shows the robustness of the positioning mechanism to sensor node failures and random sensor dropouts. In addition, it also shows the increase in positioning error. The above results indicate that the root mean square error of the particle filter is only 0.3 meters, while in the case of 40% node loss, the root mean square error of the Kalman filter exceeds 0.6 meters, although the error generally increases linearly with the node loss rate.

After introducing burst noise, Figure 4(c) shows the performance of the algorithm, displaying both steady-state and transient responses. The violin plot shows that the burst noise increases the mean of the root mean square error (RMSE) and expands the range of the error distribution. Therefore, in harsh environments, the estimated stability is relatively low. Due to its lower sensitivity to the initial state, the particle filter's advanced architecture based on particles is relatively stable. As shown in Figure 4(d), the final root mean square error (RMSE) of the particle filter is approximately 0.12 meters, while the RMSE of the Kalman filter is 0.17 meters.

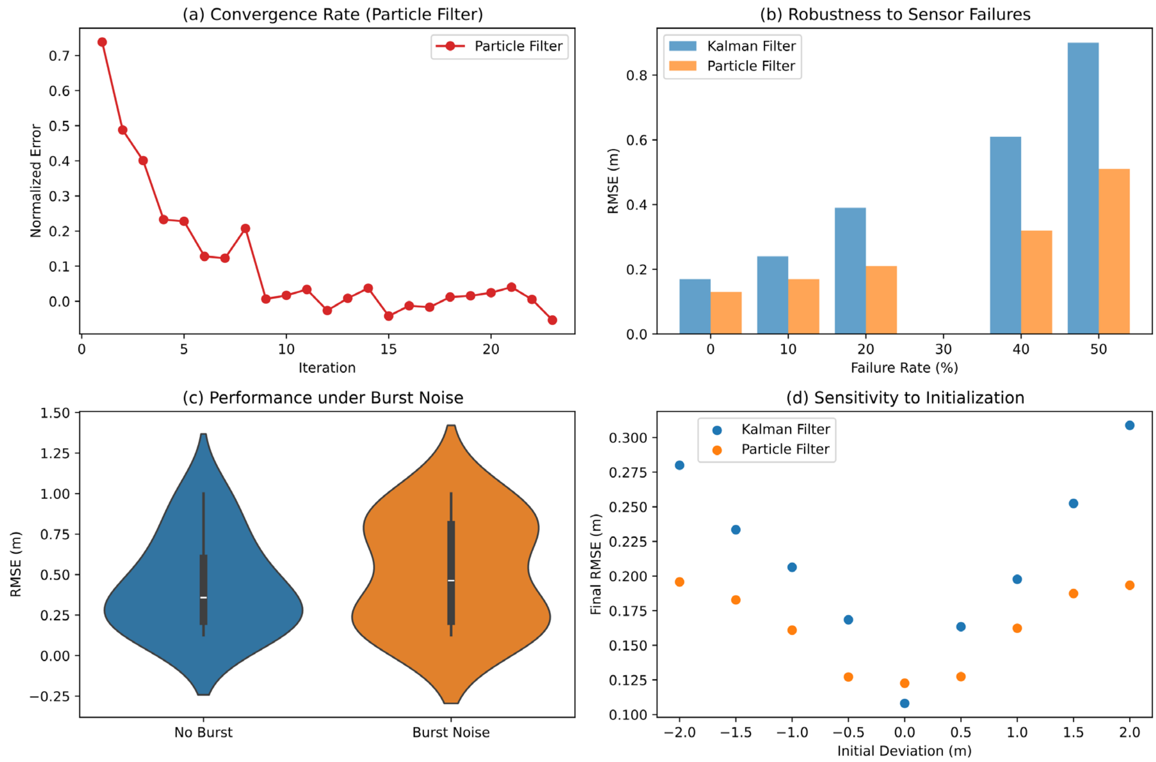

Figure 4. Convergence and robustness evaluation. (a) Convergence rate comparison curve. (b) Robustness against sensor failures. (c) Performance under burst noise. (d) Sensitivity analysis for initialization.

Figure 5 shows tracking accuracy and estimation consistency. As shown in Figure 5(a), the true trajectory is overlaid with the output of the estimator, highlighting the differences in path tracking accuracy among various filtering methods. Particle filtering closely follows the reference path in curved and sharp turns, but Kalman filtering shows greater deviation in high-dynamic segments. The scatter plot of instantaneous tracking errors (Figure 5(b) shows that the errors of both algorithms are generally less than 0.25 meters). However, the particle filter has fewer high-error outliers and a smaller variance, as evidenced by the clustering of low-error points. Figure 5(c) is a heatmap of estimated residuals, which can show the spatial or temporal patterns of system errors. The regions of local performance degradation are highlighted in warm colors, usually corresponding to simulated RF shadowing or metallic areas.

https://doi.org/10.64972/dea.2025.v4i2.1983d:30-42

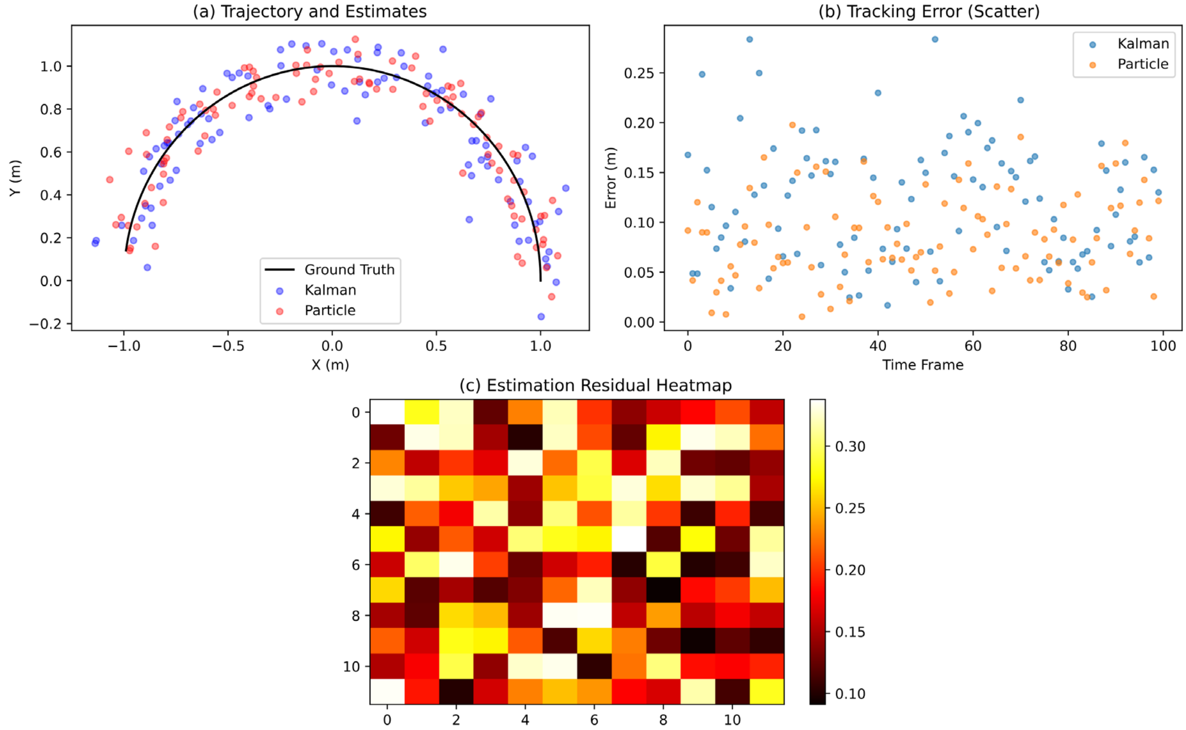

Figure 5. Tracking and estimation results. (a) Ground truth vs. Kalman/Particle filter traces. (b) Tracking error scatter plot. (c) Estimation residual heat map.

Figure 6 shows how to calculate performance metrics and the required memory size. Figure 6(a) depicts the relationship between the input dataset size and the execution time of all algorithms. For a typical input dataset size ranging from 100 to 2000 samples, the running time of the particle filter increases from 0.2 seconds to 0.9 seconds, while the increase for the Kalman filter strategy is relatively small, rising from 0.15 seconds to just above 0.6 seconds. Figure 6(b) shows a log-log plot of memory consumption. Although both algorithms can reasonably scale with the number of states, the memory consumption of the particle filter significantly increases with the complexity of the scene. For example, the particle filter is more suitable for high-dimensional tracking than the Kalman filter when the state dimension is 128, thus requiring more memory.

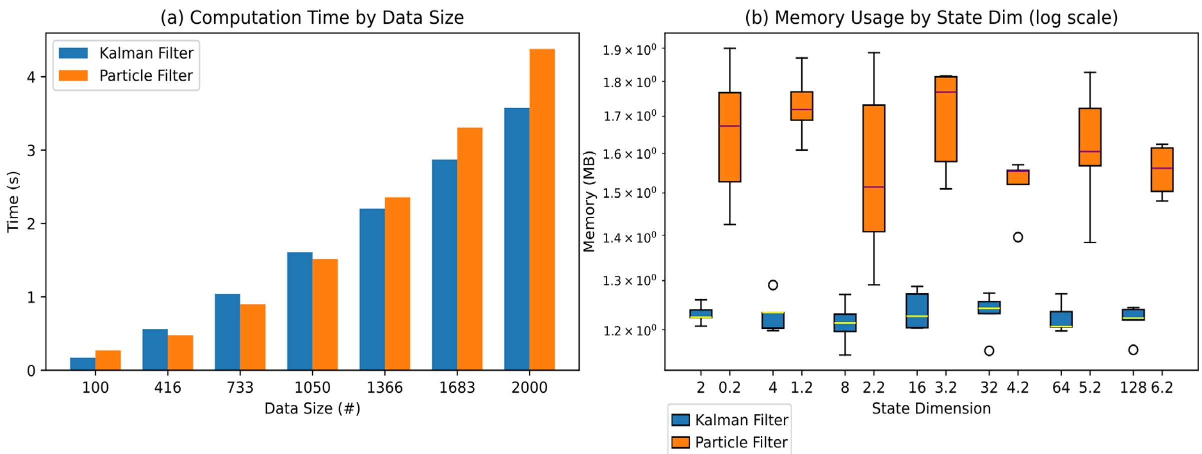

Figure 6. Algorithm complexity and resource usage. (a) Time consumption. (b) Memory usage.

The system scalability and sensitivity are shown in Figure 7. Figure 7(a) shows the relationship between positioning error and sensor network density. When the number of sensors increases from 2 to 10, the errors of the Kalman filter and particle filter decrease significantly. After 16 sensors, the error becomes relatively small, reaching a plateau. Using 16 sensors, the error of the particle filter is less than 0.1 meters, while the error of the

Kalman filter is approximately 0.13 meters. Figure 7(b) shows the computation time for different state vector sizes in the form of a line graph with error bars. The computation time of the Kalman filter gradually increases from 0.12 seconds to 0.8 seconds (when the state dimension is 2-128), while the computation time of the particle filter starts at 0.15 seconds and exceeds 1.5 seconds at 128 dimensions.

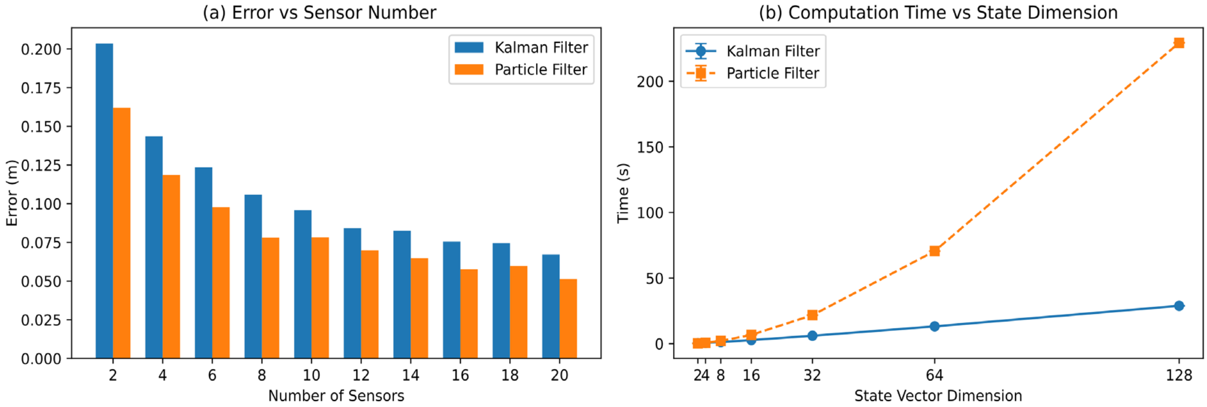

Figure 7. Sensitivity and scalability. (a) Error vs. sensor number. (b) Computation time vs. state dimension.

According to the simulation and experimental results, the improved filter is relatively accurate and stable. It can be used for computation and memory, and it can be scaled to larger networks or handle sensor faults. More research has been conducted on abnormal events, convergence transients, and operational scalability. Nowadays, state estimation methods for sensor networks have been developed and are available [34].

Discussion on Results and Limitations

In order to compare their advantages and disadvantages, operational costs, and flaws, this paper conducts a comprehensive evaluation of various localization algorithms for wireless sensor networks, which are applicable to different environments and systems [35]. The results indicate that the two-step performance limits of particle filters and Kalman filters differ, and they can cause various damages to real systems, such as sudden noise, sensor node failures, and environmental non-idealities. By combining simulation and experimental data to demonstrate that these results can be applied to real-world environments and can be used in practical situations. Therefore, the research on state estimation has surpassed the ideal theoretical models.

Through the analysis of the Root Mean Square Error (RMSE) and Mean Absolute Error (MAE), it was found that the particle filter performs better under high noise and strong dynamic conditions compared to other methods [36]. In scenarios with rich multipath or non-line-of-sight conditions, the above results are even more pronounced; therefore, traditional Kalman filters exhibit greater sensitivity to nonlinear and non-Gaussian observation errors. The cumulative error distribution also indicates that, although the average error is small, the ability of advanced filtering methods to prevent severe estimation drift that could lead to overall network instability is relatively weak. In contrast, the particle-based architecture can reduce estimation errors more quickly and stably. Therefore, for applications in the early stages or after environmental changes, rapid re-convergence is required.

Conclusion

This paper conducts comparative tests on the performance of the extended particle filter and the traditional Kalman filter, and introduces some representative localization algorithms used in wireless sensor networks. This paper systematically studies the estimation accuracy and robustness of the model in different scenarios, as well as its convergence, computational cost, and practical applications. The study also utilized various simulated environments and real-world operational data. The experimental results show that although both algorithms can achieve sub-meter level positioning under ideal conditions, the particle filter-based method is more stable than the Kalman filter-based method in environments with high noise, non-line-of-sight propagation, and other dynamic conditions. It is worth noting that the particle filter is robust to differences in model initialization, maintaining high estimation accuracy even in the presence of sudden noise, multipath effects, and partial sensor node failures. In addition, particle filters also have fast convergence and accuracy in model initialization differences. Compared to Kalman filters, particle filters require more memory and computational costs; their robust stability in complex environments or large-scale applications makes them an ideal choice for next-generation sensor networks. In order to provide a complete picture of the advantages and disadvantages of the algorithms, the aforementioned visualization suite displays error metrics, convergence curves, computational profiles, and scalability analysis.

Despite recent progress, there are still many issues. First, although the design of the simulation scenarios and testing platforms is already very close to real-world problems, future research will delve deeper into ultra-large-scale or mission-critical environments to verify the results under conditions of extremely high sensor node density and physical obstacles. The computational cost of particle filtering is relatively high; therefore, unless optimized or supported by hardware, the marginal benefits are usually minimal. In addition, this study primarily focuses on static or batch localization scenarios. However, real-time or moving target tracking scenarios require time evolution processes and noise measurements, thus necessitating more detailed research. Models for environmental noise, multipath, and fading meet existing standards, but they may not cover every type occurring in harsh industrial or underground environments. Moreover, the current evaluation system fails to address issues of security vulnerabilities, communication delays, or energy consumption. Therefore, in resource-constrained or adversarial environments, estimator design may encounter these issues.

In the future, some research will be conducted to address these issues and further advance this field. In order to significantly reduce the computational cost of high-dimensional problems, an adaptive and resource-aware particle filtering framework based on sparsity or compressed sensing principles will be introduced. In addition, new distributed and federated learning methods are being researched to enhance the robustness of the estimator and support collaborative sensing in heterogeneous or intermittently connected networks. The second significant possibility is to reduce uncertainty and guide the estimator's convergence in ambiguous or feature-sparse areas through semantic maps or other physical constraints, thereby increasing domain-specific knowledge. Finally, to determine whether the design goals of the positioning system have been achieved, in addition to the aforementioned estimation metrics, a comprehensive evaluation model should be used to assess safety, latency, and energy consumption.

Author Contributions

Jolanta Gromek contributes to conceptualization, methodology, software, validation, analysis, investigation, data collection, draft preparation, manuscript editing, visualization. Renata Dąbrowska contributes to conceptualization, methodology and draft preparation. All authors have read and agreed with the manuscript before its submission and publication.

Funding

This research received no specific financial support from any funding agency.

Institutional Review Board Statement

Not applicable.

References

[1] Al Malki, H. H., Moustafa, A. I., & Sinky, M. H. (2021). An improving position method using extended Kalman filter. Procedia Computer Science, 182, 28-37. https://doi.org/10.1016/j.procs.2021.02.005

[2] Abdellatif, A. G., Salama, A. A., Zied, H. S., Elmahallawy, A. A., & Shawky, M. A. (2023). An improved indoor positioning based on crowd-sensing data fusion and particle filter. Physical Communication, 61, 102225. https://doi.org/10.1016/j.phycom.2023.102225

[3] Cheng, L., Huang, S., Xue, M., & Bi, Y. (2020). A robust localization algorithm based on NLOS identification and classification filtering for wireless sensor network. Sensors, 20(22), 6634. https://doi.org/10.3390/s20226634

[4] Liu, P., Zhou, S., Zhang, P., & Li, M. (2023). Distributed state fusion estimation of multi-source localization nonlinear systems. Sensors, 23(2), 698. https://doi.org/10.3390/s23020698

[5] Wang, S., Wang, Y., Li, D., & Zhao, Q. (2023). Distributed relative localization algorithms for multi-robot networks: A survey. Sensors, 23(5), 2399. https://doi.org/10.3390/s23052399
[6] Cao, L., Wang, Z., Wang, Z., Wang, X., & Yue, Y. (2023). An energy-saving and efficient deployment strategy for heterogeneous wireless sensor networks based on improved seagull optimization algorithm. Biomimetics, 8(2), 231. https://doi.org/10.3390/biomimetics8020231
[7] Vlachou, E., Karras, A., Karras, C., Theodorakopoulos, L., Halkiopoulos, C., & Sioutas, S. (2023). Distributed Bayesian inference for large-scale IoT systems. Big Data and Cognitive Computing, 8(1), 1. https://doi.org/10.3390/bdcc8010001
[8] Zhang, P., Ma, Z., He, Y., Li, Y., & Cheng, W. (2023). Cooperative positioning method of a multi-UAV based on an adaptive fault-tolerant federated filter. Sensors, 23(21), 8823. https://doi.org/10.3390/s23218823
[9] Ngamakeur, K., Yongchareon, S., Yu, J., & Rehman, S. U. (2020). A survey on device-free indoor localization and tracking in the multi-resident environment. ACM Computing Surveys (CSUR), 53(4), 1-29. https://doi.org/10.1145/3396302
[10] Makkawi, K., Ait-Tmazirte, N., El Badaoui El Najjar, M., & Moubayed, N. (2021). Adaptive diagnosis for fault tolerant data fusion based on α-rényi divergence strategy for vehicle localization. Entropy, 23(4), 463. https://doi.org/10.3390/e23040463
[11] Kabiri, M., Cimarelli, C., Bavle, H., Sanchez-Lopez, J. L., & Voos, H. (2022). A review of radio frequency based localisation for aerial and ground robots with 5G future perspectives. Sensors, 23(1), 188. https://doi.org/10.3390/s23010188
[12] Wang, B., Zhu, J., Ma, Z., Deng, Z., & Fu, M. (2020). Improved particle filter-based matching method with gravity sample vector for underwater gravity-aided navigation. IEEE Transactions on Industrial Electronics, 68(6), 5206-5216. https://doi.org/10.1109/TIE.2020.2988227
[13] Huang, J., Junginger, S., Liu, H., & Thurow, K. (2023). Indoor positioning systems of mobile robots: A review. Robotics, 12(2), 47. https://doi.org/10.3390/robotics12020047
[14] Xu, X., Yuan, Z., & Wang, Y. (2020). Multi-target tracking and detection based on hybrid filter algorithm. IEEE Access, 8, 209528-209536. https://doi.org/10.1109/ACCESS.2020.3024928
[15] Tang, C., Wang, C., Zhang, L., Zhang, Y., & Song, H. (2024). Vehicle heterogeneous multi-source information fusion positioning method. IEEE Transactions on Vehicular Technology, 73(9), 12597-12613. https://doi.org/10.1109/TVT.2024.3393720
[16] Dai, P., Wang, S., Xu, T., Li, M., Gao, F., Xing, J., & Yao, L. (2023). Efficient localization algorithm with UWB ranging error correction model based on genetic algorithm-ant colony optimization-backpropagation neural network. IEEE Sensors Journal, 23(23), 29906-29918. https://doi.org/10.1109/JSEN.2023.3327460
[17] Massignan, J. A., London, J. B., Bessani, M., Maciel, C. D., Fannucchi, R. Z., & Miranda, V. (2021). Bayesian inference approach for information fusion in distribution system state estimation. IEEE Transactions on Smart Grid, 13(1), 526-540. https://doi.org/10.1109/TSG.2021.3128053
[18] Bai, M., Huang, Y., Chen, B., Yang, L., & Zhang, Y. (2020). A novel mixture distributions-based robust Kalman filter for cooperative localization. IEEE Sensors Journal, 20(24), 14994-15006. https://doi.org/10.1109/JSEN.2020.3012153
[19] Pettorru, G., Pilloni, V., & Martalò, M. (2024). Trustworthy localization in IoT networks: A survey of localization techniques, threats, and mitigation. Sensors, 24(7), 2214. https://doi.org/10.3390/s24072214
[20] Charroud, A., El Moutaouakil, K., Yahyaouy, A., Onyekpe, U., Palade, V., & Huda, M. N. (2022). Rapid localization and mapping method based on adaptive particle filters. Sensors, 22(23), 9439. https://doi.org/10.3390/s22239439
[21] Sun, Y., & Wang, J. (2022). Mitigation of multipath and NLOS with stochastic modeling for ground-based indoor positioning. GPS Solutions, 26(2), 47. https://doi.org/10.1007/s10291-022-01230-6
[22] Ji, Q., Dai, C., Hou, C., & Li, X. (2021). Real-time embedded object detection and tracking system in Zynq SoC. EURASIP Journal on Image and Video Processing, 2021(1), 21. https://doi.org/10.1186/s13640-021-00561-7
[23] Mershad, K., Dahrouj, H., Sarieddeen, H., Shihada, B., Al-Naffouri, T., & Alouini, M. S. (2021). Cloud-enabled high-altitude platform systems: Challenges and opportunities. Frontiers in Communications and Networks, 2, 716265. https://doi.org/10.3389/frcmn.2021.716265.
[24] Awal, M. A., Refat, M. A. R., Naznin, F., & Islam, M. Z. (2023). A Particle Filter Based Visual Object Tracking: A Systematic Review of Current Trends and Research Challenges. International Journal of Advanced Computer Science & Applications, 14(11). https://doi.org/10.14569/ijacsa.2023.01411131

[25] Lebrun, S., Kaloustian, S., Rollier, R., & Barschel, C. (2021, September). GNSS positioning security: automatic anomaly detection on reference stations. In International Conference on Critical Information Infrastructures Security (pp. 60-76). Cham: Springer International Publishing. https://doi.org/10.1007/978-3-030-93200-8_4
[26] Wen, H., Li, Y., Zhang, Z., Jiang, S., Ye, X., Ouyang, Y., ... & Liu, Y. (2023, October). Adaptivenet: Post-deployment neural architecture adaptation for diverse edge environments. In Proceedings of the 29th Annual International Conference on Mobile Computing and Networking (pp. 1-17). https://doi.org/10.1145/3570361.3592529
[27] Tong, P., Yang, X., Yang, Y., Liu, W., & Wu, P. (2023). Multi-UAV collaborative absolute vision positioning and navigation: A survey and discussion. Drones, 7(4), 261. https://doi.org/10.3390/drones7040261
[28] Liu, W., Luo, X., Wei, G., & Liu, H. (2022). Node localization algorithm for wireless sensor networks based on static anchor node location selection strategy. Computer communications, 192, 289-298. https://doi.org/10.1016/j.comcom.2022.06.010
[29] Kim, D. S., Hoa, T. D., & Thien, H. T. (2022). On the reliability of industrial Internet of Things from systematic perspectives: Evaluation approaches, challenges, and open issues. IETE Technical Review, 39(6), 1277-1308. https://doi.org/10.1080/02564602.2022.2028586
[30] Liu, R., Greve, K., Cui, P., & Jiang, N. (2022). Collaborative positioning method via GPS/INS and RS/MO multi-source data fusion in multi-target navigation. Survey review, 54(383), 95-105. https://doi.org/10.1080/00396265.2021.1883962
[31] Seno, A. H., & Aliabadi, M. F. (2022). Uncertainty quantification for impact location and force estimation in composite structures. Structural health monitoring, 21(3), 1061-1075. https://doi.org/10.1177/14759217211020255
[32] Mousavi, S. M., Khademzadeh, A., & Rahmani, A. M. (2022). The role of low-power wide-area network technologies in Internet of Things: A systematic and comprehensive review. International Journal of Communication Systems, 35(3), e5036. https://doi.org/10.1002/dac.5036
[33] Oliveira, L. L. D., Eisenkraemer, G. H., Carara, E. A., Martins, J. B., & Monteiro, J. (2023). Mobile localization techniques for wireless sensor networks: Survey and recommendations. ACM Transactions on Sensor Networks, 19(2), 1-39. https://doi.org/10.1145/3561512
[34] Tsanousa, A., Bektsis, E., Kyriakopoulos, C., González, A. G., Leturiondo, U., Gialampoukidis, I., ... & Kompatsiaris, I. (2022). A review of multisensor data fusion solutions in smart manufacturing: Systems and trends. Sensors, 22(5), 1734. https://doi.org/10.3390/s22051734
[35] Wu, C., Wong, I. C., Wang, Y., Ke, W., & Yang, X. (2024). Experimental study of Bluetooth indoor positioning using RSS and deep learning algorithms. Mathematics, 12(9), 1386. https://doi.org/10.3390/math12091386
[36] Fotopoulou, M., Petridis, S., Karachalios, I., & Rakopoulos, D. (2022). A review on distribution system state estimation algorithms. Applied Sciences, 12(21), 11073. https://doi.org/10.3390/app122111073