entropy

MDPI

Article

A Kalman Filter-Based Localization Calibration Method Optimized by Reinforcement Learning and Information Matrix Fusion

Zijia Huang [1], Qiushi Xu [2], Menghao Sun [2] and Xuzhen Zhu [2,*]

[1] National Key Laboratory of Multi-Domain Data Collaborative Processing and Control, Xi'an 710068, China; beiyouhzj@163.com
[2] State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications, Beijing 100876, China; 1518668454@bupt.edu.cn (Q.X.); 2024110214@bupt.cn (M.S.)
* Correspondence: zhuxuzhen@bupt.edu.cn

Abstract

To address the degradation in localization accuracy caused by insufficient robustness of filter parameters and inefficient multi-trajectory data fusion in dynamic environments, this paper proposes a Kalman filter-based localization calibration method optimized by reinforcement learning and information matrix fusion (RL-IMKF). An actor–critic reinforcement learning network is designed to adaptively adjust the state covariance matrix, enhancing the Kalman filter's adaptability to environmental changes. Meanwhile, a multi-trajectory information matrix fusion strategy is introduced, which aggregates multiple trajectories in the information domain via weighted inverse covariance matrices to suppress error propagation and improve system consistency. Experiments using both simulated and real-world sensor data demonstrate that the proposed method outperforms traditional extended Kalman filter approaches in terms of localization accuracy and stability, providing a novel solution for cooperative localization calibration of unmanned aerial vehicle (UAV) swarms in dynamic environments.

Keywords: Kalman filtering; reinforcement learning; collaborative correction

check for updates

Academic Editor: Adam Lipowski

Received: 28 June 2025
Revised: 26 July 2025
Accepted: 30 July 2025
Published: 1 August 2025

Citation: Huang, Z.; Xu, Q.; Sun, M.; Zhu, X. A Kalman Filter-Based Localization Calibration Method Optimized by Reinforcement Learning and Information Matrix Fusion. *Entropy* **2025**, *27*, 821. https://doi.org/10.3390/e27080821

Copyright: © 2025 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

With the rapid development of intelligent unmanned systems, the application of high-precision positioning technology for unmanned clusters has become a significant challenge. In dynamic scenarios, single sensors like Global Positioning System (GPS) often suffer from signal blockage and multipath effects, leading to degraded accuracy [1,2], while inertial measurement units (IMUs) accumulate errors over time and cannot support long-term localization independently [3]. Therefore, Kalman filter (KF)-based [4] multi-sensor fusion has become a mainstream solution, offering optimal estimation through a state-space model and theoretically enabling error minimization and calibration. However, the traditional Kalman filter relies on fixed parameters [5], leading to model mismatch in scenarios with motion state transitions (e.g., sudden acceleration, complex terrains) or time-varying sensor noise, which degrades localization accuracy—particularly in multi-trajectory cooperative scenarios.

In recent years, the localization calibration field has witnessed significant developments in both filtering models and learning-based strategies. Traditional Kalman filter variants, such as the extended Kalman filter (EKF) [6], unscented Kalman filter (UKF) [7],

and particle filter (PF) [8], have been widely applied in multi-sensor fusion scenarios. However, these methods still face challenges in coping with non-Gaussian noise and dynamic model mismatch, particularly in rapidly changing environments. On the other hand, adaptive filtering approaches like the adaptive Kalman filter (AKF) [9] and adaptive noise-adjusted Kalman filter (ANKF) [10] attempt to adjust filter parameters online but often rely on fixed rule-based heuristics that limit their generalizability. More recently, reinforcement learning (RL) [11] and deep neural networks have been introduced to enhance filter adaptability and learn optimal correction policies from data. Representative examples include the reinforcement learning-based adaptive Kalman filtering algorithm (RL-AKF) [12] methods, which estimate noise covariances using policy networks. Nevertheless, most of these methods focus solely on single-trajectory filtering, lacking a cooperative perspective. Meanwhile, existing multi-platform fusion approaches tend to apply heuristic weighting without fully leveraging the structural properties of uncertainty [13].

Therefore, this study proposes a Kalman filter localization calibration method based on reinforcement learning optimization and information matrix fusion. An actor–critic network architecture [14] is constructed, where a gated recurrent unit (GRU) [15] captures temporal dependencies of sensor data, and an attention mechanism [16] focuses on key state features to adaptively generate KF parameter adjustment factors. These factors dynamically regulate the state covariance matrix. Since the state covariance matrix determines the relative confidence between prediction and observation, inaccurate covariance estimation may cause the filter to under- or over-trust measurements, leading to degraded accuracy or even divergence. Therefore, adaptive optimization of the covariance matrix is essential for improving filter robustness in dynamic and uncertain environments. Additionally, we introduce a multi-trajectory information matrix fusion algorithm that aggregates inverse covariance matrices to form a layered uncertainty calibration mechanism. The main contributions of this paper are as follows: (1) Adaptive model uncertainty adjustment via reinforcement learning; (2) covariance-aware fusion for multi-trajectory cooperative localization; (3) significant accuracy improvements in both simulated and real-world UAV swarm data.

This paper is organized as follows: Section 2 reviews existing calibration methods; Section 3 details the algorithm, including the network architecture, dynamic adjustment mechanism for state covariance, and the derivation of the fusion algorithm based on information matrix inversion. Section 4 presents the experimental design and results. Section 5 concludes the study and discusses future work and applications.

2. Related Work

At present, multi-sensor fusion has become the mainstream approach for localization calibration in UAV swarms. Among them, GPS/IMU-integrated navigation systems are widely used due to their complementary advantages [17,18]. The KF, as a classical algorithm, performs optimal recursive estimation for linear systems. To address the significant errors introduced when using the KF in nonlinear scenarios [19], several nonlinear filtering variants have been developed, including the EKF, UKF, and PF. Moreover, to improve filter stability in high-dynamic environments, adaptive variants such as the AKF and federated Kalman filter (FKF) [20] have been proposed. These evolutionary filtering methods provide theoretical support for achieving high-precision localization in UAV swarms operating under complex environments, aggressive maneuvers, and sensor failures. In addition to classical adaptive Kalman filtering techniques, several advanced approaches have been proposed to address model uncertainty. These include interacting multiple-model (IMM) filters that handle mode switching [21], set-membership-based hybrid filters that enforce bounded state uncertainty [22], and robust cubature Kalman filters for tightly coupled

GNSS/INS systems [23]. Compared to these methods, our approach employs reinforcement learning to adaptively scale the state covariance based on temporal observation patterns, without relying on predefined model structures.

In the development of integrated navigation technologies, machine learning-based adaptive calibration methods have gradually emerged in recent years with the advancement of deep learning and reinforcement learning [24,25]. These methods leverage historical data to learn complex error patterns and achieve high-precision calibration. For example, Wei et al. [26] proposed a navigation method that integrates random forest regression with adaptive Kalman filtering to improve accuracy under limited data availability. Liu et al. [27] proposed an integrated navigation filtering algorithm assisted by a radial basis function (RBF) neural network, aiming to improve the navigation accuracy of the system in the case of GPS signal loss. Other studies [28] have combined deep learning with extended state Kalman filtering (ES-EKF), using Long Short-Term Memory (LSTM) [29] networks to capture temporal dependencies in sensor data and convolutional neural networks (CNNs) [30] to extract spatial features. Furthermore, the RL-AKF has been proposed, which adaptively estimates the process noise covariance using reinforcement learning techniques. In the Kalman filter framework, the state covariance matrix plays a central role in quantifying uncertainty and directly affects the computation of the Kalman gain. However, current methods still largely rely on traditional techniques for its adaptive adjustment. Notably, sliding window techniques have been widely used to enhance temporal adaptability in dynamic filtering. For instance, the moving-window-based adaptive fitting $H\infty$ filter [31] leverages windowed data to suppress nonlinear system disturbances by focusing on recent state transitions, while the Sage windowing and random weighting adaptive filtering method [32,33] mitigates kinematic model errors through discounted weighting of older data within a sliding window. These methods demonstrate the effectiveness of temporal local processing in dynamic uncertainty quantification. Therefore, this paper proposes a reinforcement learning-based approach using an actor–critic architecture to dynamically adjust the state covariance matrix.

Meanwhile, in the area of multi-trajectory cooperative calibration, inter-platform information exchange is a critical component for achieving high-precision collaboration. Existing mainstream methods focus on fusion strategies for heterogeneous multi-source data. For example, in cooperative localization of UAV swarms, platforms can share position and inertial data; ultra-wideband (UWB) ranging technology is used to acquire relative distance constraints [34]; angle-of-arrival (AOA) methods employ antenna arrays to determine signal directions [35], forming joint constraints in distance and angle; and vision sensors can be used to capture environmental features or visual markers of other UAVs, with visual SLAM algorithms employed to extract feature correspondences and construct relative pose estimates [36]. Furthermore, cutting-edge fusion techniques such as random weighting estimation have been applied to multi-dimensional data fusion, dynamically adjusting weights based on real-time data credibility to enhance robustness under varying noise conditions [37]. The random weighting method for multi-sensor data fusion, known for its flexibility in handling unknown noise distributions, has also been adopted to improve fusion accuracy in heterogeneous sensor networks [38]. For complex integrated systems like INS/GNSS/CNS, matrix-weighted multi-sensor fusion strategies optimize performance by leveraging adaptive weight matrices that account for cross-correlations between sensors, ensuring more reliable state estimation [39]. Notably, several works (e.g., Li et al. [40]) have addressed uncertainty quantification in multi-sensor fusion. However, relatively fewer studies address real-time, decentralized fusion of state covariance across multiple moving platforms under Kalman filtering frameworks. Therefore, this study proposes a cooperative calibration algorithm based on multi-trajectory information matrix fusion.

This approach enables a more precise characterization of uncertainty distribution across trajectories in dynamic environments and provides a novel solution for high-precision cooperative localization of UAV swarms in complex scenarios.

3. Methods

To address the challenges of trajectory parameter uncertainty quantification and dynamic calibration in cooperative localization of UAV swarms, this study proposes a collaborative localization framework that integrates reinforcement learning with information matrix fusion. A dual closed-loop architecture—"single-trajectory dynamic filtering and multi-trajectory cooperative correction"—is adopted (as shown in Figure 1). First, an actor–critic reinforcement learning network is used to adaptively optimize the state covariance matrix in the Kalman filter for each individual trajectory. Then, an information matrix fusion strategy is applied, in which each trajectory's covariance matrix is transformed into its corresponding information matrix. By aggregating these matrices, the framework fuses the credibility of multi-source trajectory estimates and suppresses error propagation from individual platforms. The overall algorithm consists of two key components: reinforcement learning-driven filter parameter optimization and multi-trajectory information matrix fusion.

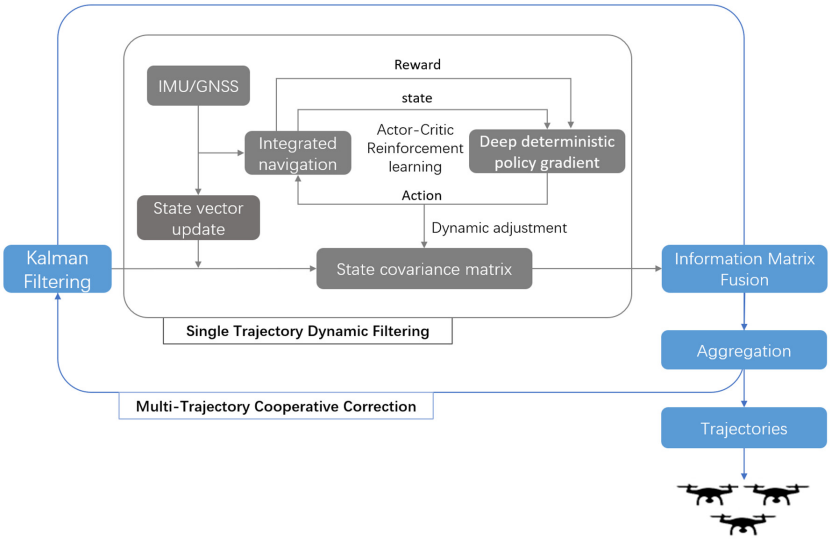


Figure 1. The framework adopts a "single-trajectory dynamic filtering–multi-trajectory collaborative correction" architecture. On the left is the single-trajectory Kalman filter parameter optimization process driven by the actor–critic reinforcement learning network, while on the right is the collaborative correction mechanism based on multi-trajectory information matrix fusion. By dynamically adjusting the state covariance matrix and aggregating information matrices, the framework achieves hierarchical calibration of positioning errors.

3.1. Reinforcement Learning-Driven Filter Parameter Optimization

In the state update process of the Kalman filter, this study employs a reinforcement learning mechanism to dynamically adjust the state covariance matrix.

We first define the system state vector $x_k = \begin{bmatrix} p_k & v_k & q_k & a_k \end{bmatrix}^T$, including the platform's position p_k, velocity v_k, quaternion q_k (which belongs to the unit quaternion group isomorphic to SO(3)) [41], and acceleration a_k. Here, $q_k = [q_w,\ q_x,\ q_y,\ q_z]^T$ denotes the unit quaternion representing the platform's orientation at time step k. It can be derived from the Euler angles (ϕ, θ, ψ), corresponding to roll, pitch, and yaw, respectively, as

$$q_k = \begin{bmatrix} q_w \\ q_x \\ q_y \\ q_z \end{bmatrix} = \begin{bmatrix} \cos(\phi/2)\cos(\theta/2)\cos(\psi/2) + \sin(\phi/2)\sin(\theta/2)\sin(\psi/2) \\ \sin(\phi/2)\cos(\theta/2)\cos(\psi/2) - \cos(\phi/2)\sin(\theta/2)\sin(\psi/2) \\ \cos(\phi/2)\sin(\theta/2)\cos(\psi/2) + \sin(\phi/2)\cos(\theta/2)\sin(\psi/2) \\ \cos(\phi/2)\cos(\theta/2)\sin(\psi/2) - \sin(\phi/2)\sin(\theta/2)\cos(\psi/2) \end{bmatrix} \tag{1}$$

Note that gyroscope measurements are not included in the state vector, as they are treated as control inputs that drive the quaternion update through the angular velocity term in the state transition function. The state update is as follows:

$$x_{k+1} = f(x_k, u_k) + w_k \tag{2}$$

$$P_{k+1} = F_k P_k {F_k}^T + Q_k \tag{3}$$

Here, x_{k+1} and P_{k+1} are the system state vector and covariance matrix at time step $k+1$; w_k is the process noise, which is assumed to be zero-mean Gaussian white noise with covariance matrices Q_k; and F_k is the Jacobian of the nonlinear state transition function $f(\cdot)$. In our approach, the process noise covariance Q_k is initially set as a prior based on sensor characteristics, $Q_k = \text{diag}(Q_{pp}, Q_{vv}, Q_{qq}, Q_{aa})$, where $Q_{pp} = \sigma_p^2 I_3$,$Q_{vv} = \frac{dt^2}{2} Q_{aa}$, $Q_{qq} = (\frac{ARW \cdot \pi}{180 \cdot \sqrt{3600}} \cdot dt)^2 I_4$(ARW: angular random walk), $Q_{aa} = (\frac{VRW \cdot 9.81}{\sqrt{3600}} \cdot dt)^2 I_3$(VRW: velocity random walk) (the specific sensor parameters are introduced in Section 4.1), but it may not fully reflect real-time dynamics or noise variations. While Q_k affects Equation (2), state covariance P better captures real-time uncertainties that Q cannot. Optimizing P via reinforcement learning adapts to these time-varying factors, complementing a fixed Q for more accurate uncertainty quantification. The function $f(x_k, u_k)$ defines the motion dynamics of the UAV platform and is given as

$$f(x_k, u_k) = \begin{bmatrix} p_k + v_k \cdot dt + 0.5 \cdot a_k \cdot dt^2 \\ v_k + a_k \cdot dt \\ \text{QuaternionUpdate}(q_k, \omega_k, dt) \\ a_k \end{bmatrix} \tag{4}$$

where ω_k is the angular velocity measured by the gyroscope, and the quaternion is updated based on standard quaternion kinematics driven by ω_k. The quaternion update follows the exponential map on $SO(3)$: $q_{k+1} = q_k \otimes \exp\left(\frac{dt}{2}\omega_k\right)$, where $(\exp : \mathbb{R}^3 \to S^3)$ is the quaternion exponential map, and $\otimes$ denotes quaternion multiplication [42].

The Jacobian matrix F_k of the state transition function $f(\cdot)$ includes the dynamic coupling terms among position, velocity, and acceleration, as well as quaternion updates, where $\Omega(\omega) = \begin{bmatrix} 0 & -\omega_x & -\omega_y & -\omega_z \\ \omega_x & 0 & \omega_z & -\omega_y \\ \omega_y & -\omega_z & 0 & \omega_x \\ \omega_z & \omega_y & -\omega_x & 0 \end{bmatrix}$ is an antisymmetric matrix constructed from angular velocity:

$$F_k = \begin{bmatrix} I_3 & dt \cdot I_3 & 0_{3\times4} & 0.5 \cdot dt^2 \cdot I_3 \\ 0_{3\times3} & I_3 & 0_{3\times4} & dt \cdot I_3 \\ 0_{4\times3} & 0_{4\times3} & F_q & 0_{4\times3} \\ 0_{3\times3} & 0_{3\times3} & 0_{3\times4} & I_3 \end{bmatrix} \tag{5}$$

$$F_q = \begin{cases} I_4 + \frac{dt}{2} \cdot \Omega(\omega), ||\omega|| \ll 1 \\ \cos(\frac{||\omega|| \cdot dt}{2}) I_4 + \frac{\sin(\frac{||\omega|| \cdot dt}{2})}{||\omega||} \Omega(\omega), others \end{cases} \tag{6}$$

Specifically, the block structure of F is constructed by linearizing the nonlinear state transition function around the current state, considering the temporal derivatives (with time step dt) of position with respect to velocity and acceleration, and the quaternion update law using the antisymmetric matrix $\Omega(\omega)$ of angular velocity ω. The quaternion update block F_q in Equation (4) is derived from the kinematic equation of quaternions under rotational motion: for small angular velocities ($\|\omega\| \ll 1$), it approximates the first-order Taylor expansion of the quaternion exponential map; for general cases, it uses the exact form of quaternion rotation based on trigonometric functions, consistent with standard quaternion update rules in inertial navigation [43].

The observation model is given by

$$z_k = Hx_k + v_k \tag{7}$$

where z_k is the GPS measurement vector (3D position), explicitly given by $z_k = [x_{\text{GPS}}, y_{\text{GPS}}, z_{\text{GPS}}]^T$; $H = \begin{bmatrix} I_{3\times3} & O_{3\times3} & O_{3\times4} & O_{3\times3} \end{bmatrix}$ is the observation matrix; and v_k is measurement noise, which is assumed to be zero-mean Gaussian white noise with covariance matrices R. H is defined as the observation Jacobian, a linear mapping that selects the position components from the full state vector. Observations are position measurements from GPS, while IMU data (accelerometer, gyroscope) serve as inputs to the dynamic motion model in Equation (1) to drive state transitions. The innovation vector is then defined as $y_k = z_k - Hx_k$.

Then, we calculate the Kalman gain K_k, which represents the degree of influence of the observed values on the state update:

$$K_k = P_k H_k^T (HP_k H^T + R)^{-1} \tag{8}$$

where R represents the measurement noise covariance matrix, quantifying the uncertainty of GPS position measurements. It is initialized as a diagonal matrix based on the known noise characteristics of the GPS sensor. Specifically, we set $R = \text{diag}([\sigma_x^2, \sigma_y^2, \sigma_z^2])$, where $\sigma_x = \sigma_y = \sigma_z \approx 2.5\,\text{m}$ (converted from CEP50 to standard deviation for Gaussian noise modeling).

Finally, we update the state vector based on innovation y_k and Kalman gain:

$$\hat{x}_k = x_k \oplus K_k y_k \tag{9}$$

To enable the Kalman filter to adaptively adjust its covariance in response to dynamic environmental changes, we introduce an RL module between the prediction and update steps. Specifically, the RL agent observes a sliding window of past states, extracts temporal features through a GRU-attention mechanism, and outputs a scaling factor that adjusts the state covariance matrix before computing the Kalman gain. This adjustment loop forms a closed process where the RL policy is iteratively optimized based on the positioning error after each update, enabling real-time adaptation of the filter to varying motion conditions.

For reinforcement learning modeling, the actor network employs a two-layer GRU to extract historical state sequences within a sliding window $S_k = \begin{bmatrix} x_{k-n}, & \ldots, & x_k \end{bmatrix} (n = 10)$. The sliding window is used to capture temporal local dependencies in dynamic UAV motion: recent states (within the window) retain critical information about short-term motion continuity (e.g., acceleration trends, attitude transitions) while excluding overly distant data that may introduce noise from outdated motion patterns. An attention mechanism is applied to assign weights to key frame features:

$$\alpha_k = softmax(W_2 \tanh(W_1 \cdot GRU(S_k))), c_k = \sum \alpha_k \cdot GRU(S_k). \tag{10}$$

In both the actor and critic networks, the attention mechanism assigns weights to key frame features of the state sequence extracted by the GRU. This enhances capture of important temporal dependencies in sensor data, improving the actor's dynamic adjustment of covariance matrices and the critic's precision in state-value estimation.

The policy network generates covariance adjustment parameters based on the contextual feature c_k. It outputs the mean μ and standard deviation σ through fully connected layers to define a Gaussian distribution $a_k(\mu,\sigma^2)$, from which an action a_k is sampled. This action is then used to update the state covariance matrix: $P_{k|k} = P_{k|k} \cdot abs(a_k)$. Here, abs(.) denotes the element-wise absolute value to ensure positive scaling of the covariance matrix. The critic also employs a GRU combined with an attention mechanism to extract temporal features and evaluate the state value V(s), and uses the negative Euclidean distance between the estimated and true positions $r = -||\hat{x}_k - x_k^{true}||$ as the reward. Where *V(s)* represents the estimated value of the current system state and this value estimation reflects the long-term cumulative reward expected from the current state under the optimal policy, guiding the actor network to adjust the state covariance matrix more effectively.

The policy update process optimizes the network parameters using the advantage function, where $\gamma = 0.99$ is the discount factor:

$$A_k = r_k + \gamma V(s_{k+1}) - V(s_k) \tag{11}$$

The parameters of the actor network are updated using the following gradient:

$$\nabla_\theta J(\theta) = E_k[\nabla_\theta \log \pi(a_k|s_k;\theta) \cdot A_k] \tag{12}$$

The parameters of the critic network are updated by minimizing the mean squared error (MSE):

$$L_{critic} = (r_k + \gamma V(s_{k+1}) - V(s_k))^2 \tag{13}$$

The overall network architecture is illustrated in Figure 2 below. Unlike simple scaling, this framework adjusts each dimension (adjustments to 13 dimensions) of P based on real-time motion dynamics. As detailed in Equations (11)–(13) and Figure 2, the agent learns dimension-specific adjustment strategies by analyzing motion characteristics, ensuring each sub-block of P accurately reflects the actual noise level of its corresponding state component, thereby enhancing the reliability of state estimation.

3.2. Multi-Trajectory Information Fusion

At the multi-platform level, to achieve cooperative correction of trajectory information among platforms, this paper proposes a fusion strategy based on the information matrix.

First, for trajectory *i*, its state covariance matrix is P_i and the corresponding information matrix is defined as $\Lambda_i = P_i^{-1}$ (in practice, a pseudo-inverse is used to replace the inverse for singular matrices). For the entire UAV swarm, the fused information matrix is given by

$$\Lambda_{fused} = \sum \Lambda_i \tag{14}$$

$$P_{fused} = \Lambda_{fused}^{-1} \tag{15}$$

Considering the differences in covariance among trajectories, the following method is used to compute the mean in order to avoid distortion:

$$P_{mean} = \frac{1}{N} \sum_{i=1}^{n} P_i \tag{16}$$

Using the norm $||P_{mean}||$ as a reference threshold, an asymmetric correction mechanism is adopted. For each trajectory i, if its covariance norm $||P_i|| > ||P_{mean}||$, the trajectory is considered to have high uncertainty. In this case, the fused result is fed back to the original trajectory and updated by weighted averaging:

$$P_i = \frac{1}{2}(P_{mean} + P_i), ||P_i|| > ||P_{mean}|| \tag{17}$$

This enables trajectory estimates with high uncertainty to have their covariance compressed using the fused result, thereby improving overall estimation consistency and system robustness while avoiding the degradation of high-quality estimates.

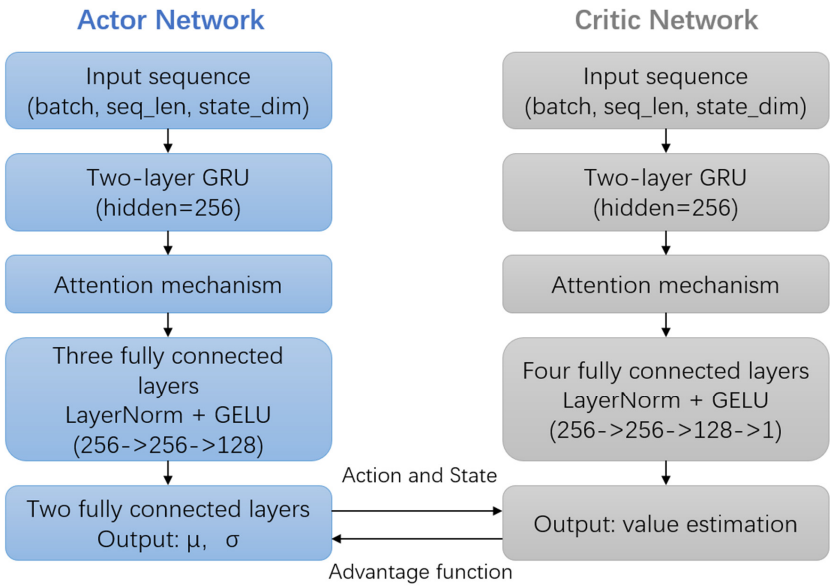


Figure 2. Both the actor and critic networks take the state sequence within a sliding window as input. Temporal features are extracted through two layers of a GRU, and an attention mechanism is applied to focus on key states. The actor network outputs covariance adjustment parameters (mean μ and standard deviation σ), while the critic network outputs the estimated state value. Policy optimization is achieved through the advantage function.

4. Experiments and Results

4.1. Experimental Setup

The experiments use the IMU data and GNSS signals generated by the simulation software GNSS-INS-SIM (Version 2.1), which produces IMU data and GPS signals with configurable noise characteristics (consistent with typical low-cost sensors: gyroscope (bias drift 10°/h, ARW 0.75°/$\sqrt{h}$) and accelerometer (bias drift 2×10^{-4} g, VRW 0.05 g/$\sqrt{h}$). The dataset is divided into training and testing sets, covering motion scenarios including static, uniform motion, and accelerated turns. The training dataset contains 20 trajectories covering multiple dynamic motion patterns; the testing dataset contains 5 independent trajectories not included in training, used to evaluate the algorithm's generalization capability. Additionally, a set of real data collected by the WTGAHRS3-232 (sourced from WitMotion Shenzhen Co., Ltd., Shenzhen, China) inertial measurement module is introduced for validation, whose main parameters are accelerometer range: ±16 g; RMS noise: 0.75 1mg-rms; gyroscope range: ±2000°/s; RMS noise: 0.05°/s-rms; attitude accuracy (pitch/roll): 0.2°; GPS positioning accuracy: 2.5 m CEP50. During algorithm training, the actor–critic network uses a batch size of 20, learning rates of 1×10^{-5} for the actor and 1×10^{-4} for the critic, and a temporal window length where sequence length = 10. The Adam optimizer is used for 20 training epochs. The evaluation metric is the three-dimensional root mean square error

(RMSE) of position. The effectiveness of the proposed algorithm is validated by comparing its results with those of the EKF and other methods. Figure 3 below shows the 20 training trajectories and 5 testing trajectories. Trajectories 1–4 are simulated data, while trajectory 5 is the ground truth data. Red scatter points represent GPS observations, the green curve denotes the error-free ground truth, and the blue curve indicates the estimated trajectory.

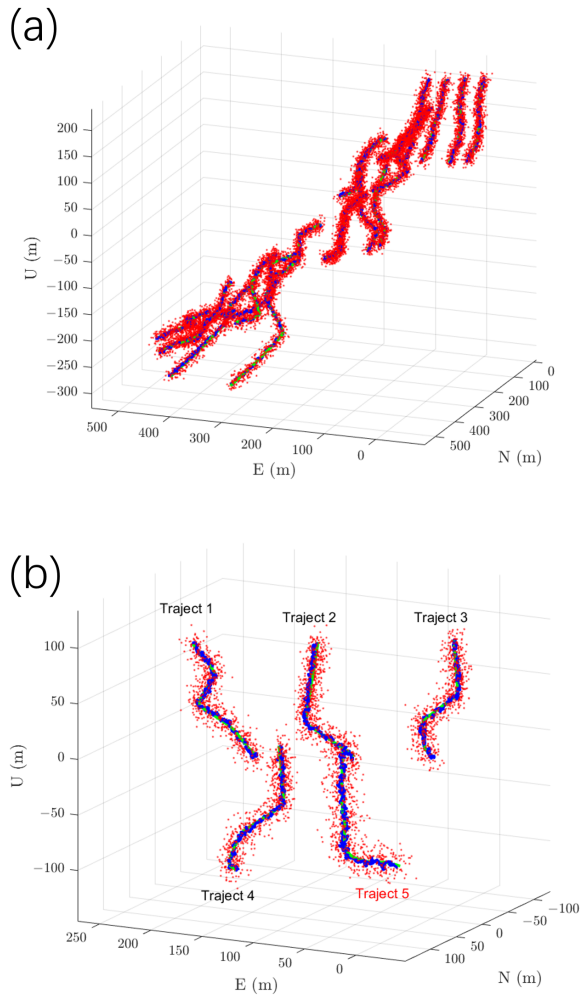


Figure 3. Visualization of training and testing trajectories. (**a**) shows 20 sets of training trajectories, and (**b**) shows 5 sets of testing trajectories (trajectory 5 represents real-world data collected by the WTGAHRS3-232 module). Red dots indicate GPS observations, the green curve represents the ground truth without error, and the blue curve shows the RL-IMKF estimated trajectories, illustrating motion under various dynamic scenarios.

4.2. Experimental Methods

The experiment was conducted through the following steps for model training and testing analysis. During the training phase, simulated data was used to construct state sequences (13 dimensions including position, velocity, quaternion, and acceleration). The actor–critic network optimized the dynamic adjustment parameters of the Kalman filter covariance matrix, using the negative position error as the reward function. Training proceeded iteratively until the loss function converged. During testing, the trained model was applied to both simulated and real data to calculate the error between the filtered trajectory positions and the ground truth, and to compare the performance of the proposed method against other filtering methods. Figure 4a below shows the actor loss trend during training, which exhibits three distinct stages: initially, due to random parameters and exploratory strategies, the loss fluctuated sharply; subsequently, as the network learned

effective strategies, the loss rapidly increased above zero and the parameter adjustment direction became clearer; finally, the loss stabilized around zero with minor fluctuations, indicating ongoing fine-tuning. In Figure 4b, the critic loss decreased rapidly at first due to large value estimation errors, then fluctuated under data and scenario disturbances, and eventually converged to a low level with small oscillations, reflecting progressively more accurate value fitting.

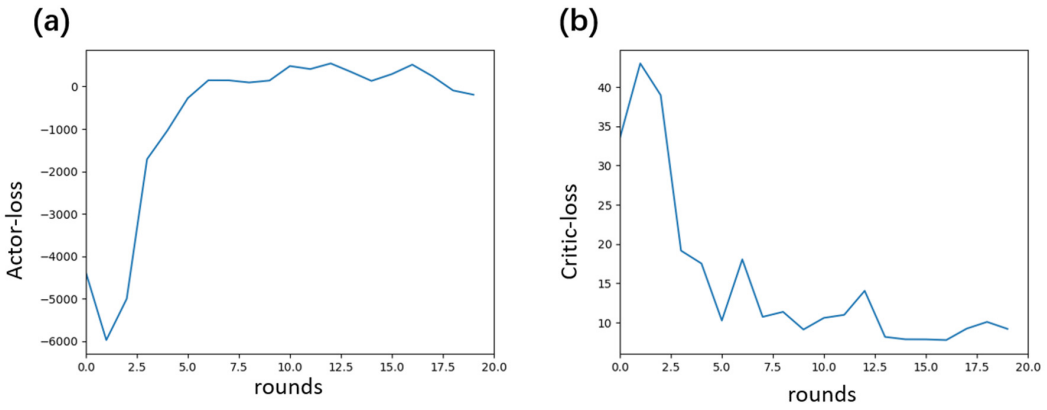


Figure 4. Loss curves during the training process. (**a**) The actor loss exhibits a three-phase convergence pattern: significant fluctuations during the initial random exploration phase, a rapid increase during the policy optimization phase, and eventual stabilization near zero. (**b**) The critic loss decreases from a high level to low-amplitude fluctuations as the value function fitting accuracy improves, reflecting the progressively improving accuracy of state-value estimation.

4.3. Evaluation of Experimental Results

To evaluate the performance of the different filtering methods, this experiment compared the standard EKF, adaptive noise-adjusted Kalman filter (ANKF) [44], bidirectional extended Kalman filter (BEKF) [45], LSTM-based trajectory estimation algorithm [46], adaptive Kalman filter algorithm based on reinforcement learning, and the proposed RL-IMKF, where a number of trajectories M = 5 was used for the information matrix fusion. The primary comparison metric was the average position error across each trajectory. Table 1 lists the average position errors for the different filtering methods. The RL-IMKF method demonstrated the lowest error, achieving a 17.4% reduction compared to EKF. This validates that RL-IMKF improves system stability by dynamically generating covariance scaling factors through the actor–critic network. Figure 5 illustrates the positional error trends for the same trajectory under different algorithms. The RL-IMKF shows a smoother overall error curve, with both peak values and fluctuation amplitudes smaller than the methods, demonstrating a clear advantage in stability.

Table 1. Comparison of average positioning errors across different filtering methods.

Filtering Method	Average Error on Training Data (m)	Average Error on Test Data (m)
Raw GNSS	9.0915	9.8605
EKF	2.7469	2.9337
ANKF	2.6414	2.8140
BEKF	2.4125	2.7037
LSTM-based	2.4767	2.5836
RL-AKF	2.5756	2.6932
RL-IMKF	2.3141	2.4221

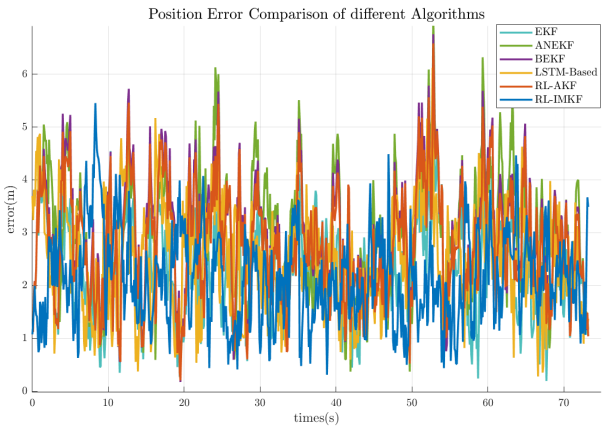


Figure 5. Temporal comparison of positioning errors across different algorithms. This compares the positioning errors of EKF, ANKF, BEKF, LSTM-based method, RL-AKF, and RL-IMKF in dynamic scenarios. The RL-IMKF error curve (blue) exhibits the highest overall smoothness, with peak errors significantly lower than those of the other algorithms, demonstrating the stability advantages of dynamic covariance optimization and information matrix fusion.

The experimental results indicate that the standard EKF, due to its use of fixed filtering parameters, is prone to model mismatch in dynamic motion scenarios, resulting in the largest errors. Although ANKF introduces an adaptive noise adjustment mechanism, it relies on predefined noise variation models, limiting its adaptability to complex motion patterns. BEKF improves trajectory smoothness through bidirectional filtering but does not consider the credibility differences among multiple trajectories. The LSTM-based trajectory estimation method leverages recurrent neural networks to model temporal dependencies in sequential positioning data, but the deterministic nature of LSTM predictions lacks explicit uncertainty quantification. RL-AKF uses reinforcement learning to adjust process noise but does not integrate multi-trajectory fusion, restricting its ability to suppress individual trajectory errors. In contrast, RL-IMKF dynamically optimizes the state covariance matrix of the Kalman filter through reinforcement learning and combines information matrix fusion to leverage multi-trajectory redundancy. This approach not only enables real-time filter parameter optimization but also suppresses error propagation from abnormal trajectories via the cumulative information matrix mechanism, demonstrating superior performance in both training data fitting and test data generalization.

Additionally, to investigate the impact of the number of trajectories on positioning accuracy in multi-trajectory fusion, an experiment selecting trajectory 1 compared the single-trajectory position errors of IMKF with trajectory counts, M, ranging from 1 to 5. The results are shown in Table 2.

Table 2. Single-trajectory positioning errors under different numbers of fused trajectories M.

M	Position Error (m)
1	2.6679
2	2.6373
3	2.6655
4	2.1202
5	2.0767

It can be seen that when only a small number of trajectories are fused, the improvement in positioning accuracy is not significant, and error fluctuations may occur due to information redundancy; however, as the number of trajectories increases, the complemen-

tary information among multiple trajectories gradually dominates, resulting in a significant decrease in positioning error with the increase in the number of trajectories.

5. Conclusions

This study proposes a Kalman filter localization calibration method based on RL-IMKF, effectively addressing the robustness issue of filter model parameters in dynamic positioning of unmanned clusters. The constructed actor–critic network captures temporal features using a GRU and attention mechanisms to dynamically generate covariance adjustment factors. Simultaneously, a multi-trajectory information matrix fusion algorithm is designed to aggregate the inverse of the trajectory covariance matrices, quantify uncertainty, suppress error propagation, and improve the utilization of multi-source data. The experimental results demonstrate that RL-IMKF achieves the lowest average position error on both training and testing datasets, significantly reducing errors by 17.4% compared to traditional methods such as EKF. Regarding error variation trends, the RL-IMKF error curve is smoother, with peak values and fluctuation amplitudes noticeably smaller than those of other comparative algorithms. Furthermore, a dedicated experiment on the impact of the number of fused trajectories on positioning accuracy shows that reasonably increasing the number of fused trajectories leverages complementary information to reduce errors, providing valuable insights for practical engineering applications.

In practical applications, the RL-IMKF method demonstrates significant potential for adaptation to diverse scenarios. Whether deployed in intelligent unmanned system clusters with stringent environmental perception requirements or in complex environments affected by signal interference and data noise, this method achieves high-precision localization calibration through dynamic optimization of filter parameters and fusion of multi-source trajectory information. From transportation and logistics to monitoring and reconnaissance, RL-IMKF can enhance the reliability and cooperation of unmanned device positioning, facilitating broader and more effective deployment of unmanned systems across multiple fields.

In summary, the RL-IMKF method offers a new approach for high-precision positioning of unmanned clusters. Its strong performance in both theoretical validation and experimental testing provides technical support for the widespread application of intelligent unmanned systems in military, civilian, and other fields. Although this study has achieved significant progress, there remains room for optimization and further development. Future work could explore integrating multi-source heterogeneous sensor data such as UWB, vision, and LiDAR into the algorithm framework, and improve the design of the reinforcement learning state-space and information matrix fusion strategies to enable multimodal data collaboration in more complex environments. Additionally, considering the dynamic changes in communication topology during unmanned cluster movement, researching algorithms based on graph neural networks for trajectory association and dynamic fusion will also be a crucial direction to enhance algorithm adaptability.

Author Contributions: Conceptualization, Z.H. and Q.X.; methodology, Q.X. and M.S.; software, Q.X. and M.S.; validation, Z.H. and X.Z.; formal analysis, Z.H.; investigation, M.S. and X.Z.; resources, X.Z.; data curation, X.Z.; writing—original draft preparation, Z.H. and Q.X.; writing—review and editing, X.Z.; visualization, M.S.; supervision, X.Z.; project administration, X.Z.; funding acquisition, Z.H. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Key Laboratory of Multi-domain Data Collaborative Processing and Control under the Open Fund project MDPC20240103.

Institutional Review Board Statement: Not applicable.

Data Availability Statement: The data used in this study are available from the corresponding author upon reasonable request.

Acknowledgments: The authors would like to express their sincere gratitude to the National Key Laboratory of Multi-domain Data Collaborative Processing and Control for its support and funding. This support has been instrumental in facilitating this research.

Conflicts of Interest: The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

1. Pereira, P.M.C.; da Silva, H.D.M.; Lima, C.M.G.S. Advancements in multipath mitigation for gnss receivers: Review of channel estimation techniques. *Space Sci. Technol.* **2025**, *5*, 0278. [CrossRef]
2. Chen, W.; Zhang, C.; Peng, Y.; Yao, Y.; Cai, M.; Dong, D. Enhancing gnss positioning in urban canyon areas via a modified design matrix approach. *IEEE Internet Things J.* **2024**, *11*, 10252–10265. [CrossRef]
3. Pai, K.R.; Marakala, N. A review on inertial navigational systems. In Proceedings of the 2016 International Conference on Electrical, Electronics, and Optimization Techniques (ICEEOT), Chennai, India, 3–5 March 2016; pp. 1682–1686. [CrossRef]
4. Kalman, R.E. A new approach to linear filtering and prediction problems. *Trans. ASME-Basic Eng.* **1960**, *82*, 35–45. [CrossRef]
5. Alsaggaf, A.U.; Saberi, M.; Berry, T.; Ebeigbe, D. Nonlinear kalman filtering in the absence of direct functional relationships between measurement and state. *IEEE Control Syst. Lett.* **2024**, *8*, 2865–2870. [CrossRef]
6. Kou, E.; Haggenmiller, A. Extended kalman filter state estimation for autonomous competition robots. *J. Stud. Res.* **2023**, *12*, 1. [CrossRef]
7. Wan, E.; Van Der Merwe, R. The unscented kalman filter for nonlinear estimation. In Proceedings of the IEEE 2000 Adaptive Systems for Signal Processing, Communications, and Control Symposium (Cat. No.00EX373), Lake Louise, AB, Canada, 1–4 October 2000; pp. 153–158. [CrossRef]
8. Luo, J.; Wang, Z.; Chen, Y.; Wu, M.; Yang, Y. An improved unscented particle filter approach for multi-sensor fusion target tracking. *Sensors* **2020**, *20*, 6842. [CrossRef]
9. Huang, Y.; Zhang, Y.; Xu, B.; Wu, Z.; Chambers, J.A. A new adaptive extended kalman filter for cooperative localization. *IEEE Trans. Aerosp. Electron. Syst.* **2018**, *54*, 353–368. [CrossRef]
10. Akhlaghi, S.; Zhou, N.; Huang, Z. Adaptive adjustment of noise covariance in Kalman filter for dynamic state estimation. In Proceedings of the 2017 IEEE Power & Energy Society General Meeting, Chicago, IL, USA, 16–20 July 2017; pp. 1–5. [CrossRef]
11. Lillicrap, T.P.; Hunt, J.J.; Pritzel, A.; Heess, N.; Erez, T.; Tassa, Y.; Silver, D.; Wierstra, D. Continuous control with deep reinforcement learning. *arXiv* **2019**, arXiv:1509.02971. [PubMed]
12. Gao, X.; Luo, H.; Ning, B.; Zhao, F.; Bao, L.; Gong, Y.; Xiao, Y.; Jiang, J. Rl-akf: An adaptive kalman filter navigation algorithm based on reinforcement learning for ground vehicles. *Remote Sens.* **2020**, *12*, 1704. [CrossRef]
13. Julier, S.; Uhlmann, J.K. General Decentralized Data Fusion with Covariance Intersection (CI). In *Handbook of Multisensor Data Fusion, Theroy and Practice*; CRC Press: Boca Raton, FL, USA, 2017. [CrossRef]
14. Sutton, R.S.; McAllester, D.; Singh, S.; Mansour, Y. Policy gradient methods for reinforcement learning with function approximation. In Proceedings of the 13th International Conference on Neural Information Processing Systems (NIPS'99), Denver, CO, USA, 29 November–4 December 1999; MIT Press: Cambridge, MA, USA, 1999; pp. 1057–1063.
15. Chung, J.; Gulcehre, C.; Cho, K.; Bengio, Y. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv* **2014**, arXiv:1412.3555. [CrossRef]
16. Niu, Z.; Zhong, G.; Yu, H. A review on the attention mechanism of deep learning. *Neurocomputing* **2021**, *452*, 48–62. [CrossRef]
17. Jaradat, M.A.K.; Abdel-Hafez, M.F. Enhanced, delay dependent, intelligent fusion for ins/gps navigation system. *IEEE Sens. J.* **2014**, *14*, 1545–1554. [CrossRef]
18. Ni, S.; Li, S.; Xie, Y.; Deng, D. Overview of gnss/ins ultra-tight integrated navigation. *J. Natl. Univ. Def. Technol.* **2023**, *45*, 48–59. [CrossRef]
19. Zhao, H.; Wang, Z. Motion measurement using inertial sensors, ultrasonic sensors, and magnetometers with extended kalman filter for data fusion. *IEEE Sens. J.* **2012**, *12*, 943–953. [CrossRef]
20. Zhang, Z.; Zhou, X.; Wang, C.; Zhao, W.; Wu, G.; Jiang, T.; Wang, M. Adaptive robust federal kalman filter for multisensor fusion positioning systems of intelligent vehicles. *IEEE Sens. J.* **2024**, *24*, 17269–17281. [CrossRef]
21. Gao, B.; Gao, S.; Zhong, Y.; Hu, G.; Gu, C. Interacting multiple model estimation-based adaptive robust unscented Kalman filter. *Int. J. Control Autom. Syst.* **2017**, *15*, 2013–2025. [CrossRef]
22. Zhao, Y.; Zhang, J.; Hu, G.; Zhong, Y. Set-Membership Based Hybrid Kalman Filter for Nonlinear State Estimation under Systematic Uncertainty. *Sensors* **2020**, *20*, 627. [CrossRef] [PubMed] [PubMed Central]

23. Gao, B.; Hu, G.; Zhong, Y.; Zhu, X. Cubature Kalman Filter With Both Adaptability and Robustness for Tightly-Coupled GNSS/INS Integration. *IEEE Sens. J.* **2021**, *21*, 14997–15011. [CrossRef]
24. Cohen, N.; Klein, I. Inertial navigation meets deep learning: A survey of current trends and future directions. *Results Eng.* **2024**, *24*, 103565. [CrossRef]
25. Jwo, D.-J.; Biswal, A.; Mir, I.A. Artificial neural networks for navigation systems: A review of recent research. *Appl. Sci.* **2023**, *13*, 4475. [CrossRef]
26. Wei, X.; Li, J.; Feng, K.; Zhang, D.; Li, P.; Zhao, L.; Jiao, Y. A mixed optimization method based on adaptive kalman filter and wavelet neural network for ins/gps during gps outages. *IEEE Access* **2021**, *9*, 47875–47886. [CrossRef]
27. Liu, H.; Li, K.; Fu, Q.; Yuan, L. Research on integrated navigation algorithm based on radial basis function neural network. *J. Phys. Conf. Ser.* **2021**, *1961*, 012031. [CrossRef]
28. Li, S.; Mikhaylov, M.; Mikhaylov, N.; Pany, T. Deep learning based kalman filter for gnss/ins integration: Neural network architecture and feature selection. In Proceedings of the 2023 International Conference on Localization and GNSS (ICL-GNSS), Castellón, Spain, 6–8 June 2023; pp. 1–7. [CrossRef]
29. Hochreiter, S.; Schmidhuber, J. Long short-term memory. *Neural Comput.* **1997**, *9*, 1735–1780. [CrossRef]
30. O'Shea, K.; Nash, R. An introduction to convolutional neural networks. *arXiv* **2015**, arXiv:1511.08458. [CrossRef]
31. Xia, J.; Gao, S.; Zhong, Y.; Qi, X.; Li, G.; Liu, Y. Moving-window-based adaptive fitting H-infinity filter for the nonlinear system disturbance. *IEEE Access* **2020**, *8*, 76143–76157. [CrossRef]
32. Gao, S.; Wei, W.; Zhong, Y.; Subic, A. Sage windowing and random weighting adaptive filtering method for kinematic model error. *IEEE Trans. Aerosp. Electron. Syst.* **2015**, *51*, 1488–1500. [CrossRef]
33. Song, I.Y.; Jeon, M.; Shin, V. Efficient multisensor fusion with sliding window Kalman filtering for discrete-time uncertain systems with delays. *IET Signal Process.* **2012**, *6*, 446–455. [CrossRef]
34. Han, Y.; Wei, C.; Li, R.; Wang, J.; Yu, H. A novel cooperative localization method based on imu and uwb. *Sensors* **2020**, *20*, 467. [CrossRef] [PubMed]
35. Kang, X.; Wang, D.; Shao, Y.; Ma, M.; Zhang, T. An efficient hybrid multi-station tdoa and single-station aoa localization method. *IEEE Trans. Wirel. Commun.* **2023**, *22*, 5657–5670. [CrossRef]
36. Qin, T.; Li, P.; Shen, S. Vins-mono: A robust and versatile monocular visual-inertial state estimator. *IEEE Trans. Robot.* **2018**, *34*, 1004–1020. [CrossRef]
37. Gao, S.; Zhong, Y.; Shirinzadeh, B. Random weighting estimation for fusion of multi-dimensional position data. *Inf. Sci.* **2010**, *180*, 4999–5007. [CrossRef]
38. Gao, S.; Zhong, Y.; Li, W. Random Weighting Method for Multisensor Data Fusion. *IEEE Sens. J.* **2011**, *11*, 1955–1961. [CrossRef]
39. Hu, G.; Gao, S.; Zhong, Y.; Gao, B.; Subic, A. Matrix weighted multisensor data fusion for INS/GNSS/CNS integration. *Proc. Inst. Mech. Eng. Part G J. Aerosp. Eng.* **2016**, *230*, 1011–1026. [CrossRef]
40. Li, G.; Knoop, V.L.; van Lint, H. How predictable are macroscopic traffic states: A perspective of uncertainty quantification. *Transp. B Transp. Dyn.* **2024**, *12*, 2314766. [CrossRef]
41. Solà, J. Quaternion kinematics for the error-state Kalman filter. *arXiv* **2017**, arXiv:1711.02508. [CrossRef]
42. Xu, W.; Cai, Y.; He, D.; Lin, J.; Zhang, F. FAST-LIO2: Fast Direct LiDAR-Inertial Odometry. *IEEE Trans. Robot.* **2022**, *38*, 2053–2073. [CrossRef]
43. Titterton, D.H.; Weston, J.L. *Strapdown Inertial Navigation Technology*, 2nd ed.; Institution of Engineering and Technology: Stevenage, UK, 2004.
44. Vettori, S.; Lorenzo, D.; Peeters, B.; Luczak, M.; Chatzi, E. An adaptive-noise augmented kalman filter approach for input-state estimation in structural dynamics. *Mech. Syst. Signal Process.* **2023**, *184*, 109654. [CrossRef]
45. Dong, Y.; Kwan, K.; Arslan, T. Enhanced pedestrian trajectory reconstruction using bidirectional extended kalman filter and automatic refinement. In Proceedings of the 2024 14th International Conference on Indoor Positioning and Indoor Navigation (IPIN), Kowloon, Hong Kong, 14–17 October 2024; pp. 1–6. [CrossRef]
46. Zhang, X.; He, F.; Zheng, T. An LSTM-based trajectory estimation algorithm for non-cooperative maneuvering flight vehicles. In Proceedings of the 2019 Chinese Control Conference (CCC), Guangzhou, China, 27–30 July 2019; pp. 8821–8826. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.