

Manifold-Optimized Error-State Kalman Filter for Robust Pose Estimation in Unmanned Aerial Vehicles

Bolin Jia¹, Zongwen Bai^{1*}, Yiqun Gao¹, Dong Wang¹, Meili Zhou¹, Peiqi Gao¹, Pei Zhang¹, Zhang Yang²

¹School of Physics and Electronic Information, Shaanxi Key Laboratory of Intelligent Processing for Big Energy Data, Yan'an University, Yan'an 716000, Shaanxi, China

²Zichang Vegetable Center, Yan'an 716000, Shaanxi, China

*Corresponding author: Zongwen Bai, ydbzw@yau.edu.cn

Copyright: © 2025 Author(s). This is an open-access article distributed under the terms of the Creative Commons Attribution License (CC BY 4.0), permitting distribution and reproduction in any medium, provided the original work is cited.

Abstract: This paper presents a manifold-optimized Error-State Kalman Filter (ESKF) framework for unmanned aerial vehicle (UAV) pose estimation, integrating Inertial Measurement Unit (IMU) data with GPS or LiDAR to enhance estimation accuracy and robustness. We employ a manifold-based optimization approach, leveraging exponential and logarithmic mappings to transform rotation vectors into rotation matrices. The proposed ESKF framework ensures state variables remain near the origin, effectively mitigating singularity issues and enhancing numerical stability. Additionally, due to the small magnitude of state variables, second-order terms can be neglected, simplifying Jacobian matrix computation and improving computational efficiency. Furthermore, we introduce a novel Kalman filter gain computation strategy that dynamically adapts to low-dimensional and high-dimensional observation equations, enabling efficient processing across different sensor modalities. Specifically, for resource-constrained UAV platforms, this method significantly reduces computational cost, making it highly suitable for real-time UAV applications.

Keywords: UAV pose estimation; Error-State Kalman Filter; Manifold; GPS; LiDAR

Online publication: April 3, 2025

1. Introduction

Accurate pose estimation, which determines both position and orientation, is essential for unmanned aerial vehicles (UAVs) in tasks such as localization, navigation, path planning^[1], and other autonomous operations^[2]. In UAV applications, precise pose estimation is particularly critical for autonomous flight control, remote sensing target detection, and precise positioning of detected objects.

Traditional pose estimation methods rely on a single sensor, such as Global Positioning System (GPS), inertial measurement units (IMUs)^[3], or cameras, each with inherent advantages and limitations. GPS

provides absolute positioning but lacks orientation information and performs poorly in indoor or GPS-denied environments. IMUs, while capable of high-frequency motion sensing, suffer from drift errors due to cumulative integration noise. Vision-based methods, such as camera-based Simultaneous Localization and Mapping (SLAM), depend on good lighting conditions and visual feature availability, which are not always guaranteed in UAV operations.

To overcome these limitations, sensor fusion techniques have been widely explored to integrate IMU data with GPS or LiDAR, thereby improving pose estimation accuracy and robustness. This paper presents a manifold-optimized Error-State Kalman Filter (ESKF) framework designed specifically for real-time UAV applications, fusing IMU data with GPS or LiDAR to achieve accurate state estimation. Unlike traditional approaches that use rotation matrices or quaternions, which introduce redundant degrees of freedom, this work adopts a manifold-based optimization approach that leverages exponential and logarithmic mappings to transform rotation vectors into rotation matrices, thereby achieving a minimal three-degree-of-freedom representation.

The main contributions of this work are as follows: (1) Development of a real-time sensor fusion framework based on ESKF^[4], enabling accurate and robust UAV pose estimation by integrating IMU with GPS or LiDAR data. (2) Adoption of a manifold-based representation and error-state for rotational increments, addressing gimbal lock issues in Euler angles and eliminating redundancy in rotation matrices and quaternions, thereby improving computational stability. (3) Introduction of a novel Kalman gain computation strategy, which dynamically adjusts for low-dimensional and high-dimensional observation equations, ensuring computational efficiency. Specifically, for resource-constrained UAV platforms, this method significantly reduces computational cost, making it highly suitable for real-time UAV applications.

2. Related work

Pose estimation is a fundamental task in autonomous UAV cruising, enabling precise positioning, flight control, and target tracking. However, compared to ground-based systems, UAVs face a series of unique challenges, including limited onboard computational resources, GPS signal degradation, and rapid dynamic motion, making real-time state estimation more challenging than ground systems. To enhance the accuracy and robustness of UAV pose estimation, multi-sensor fusion techniques have been widely adopted. These fusion approaches effectively compensate for the limitations of individual sensors and improve UAV adaptability in complex mission scenarios. IMU-GPS fusion provides global positioning information, while IMU-LiDAR^[5] fusion performs well in environments where GPS signals are limited or unavailable. Additionally, IMU-vision^[6] fusion methods have played a significant role in autonomous UAV navigation, especially in complex environments with low illumination or no GPS availability.

Optimization-based methods have been widely adopted in SLAM systems, such as VINS-Mono^[7] and VINS-Motion^[8]. These methods employ IMU pre-integration^[6] to fuse IMU data with other sensor inputs, resulting in high-precision pose estimation. However, due to their high computational demands, these methods are challenging to implement in real-time applications, especially in resource-constrained environments such as UAVs or small robots.

Filtering-based methods offer a computationally efficient alternative to optimization-based approaches. However, handling rotational states in UAV pose estimation remains a key challenge. Classical Kalman filters

assume a Gaussian distribution of system states, requiring the state space to be Euclidean. However, rotation matrices and quaternions do not satisfy these conditions due to their nonlinear nature. To address this issue, Euler angles or axis-angle representations are sometimes used, but both suffer from gimbal lock or singularities. When using rotation matrices or quaternions, nonlinear extensions such as Extended Kalman Filters (EKF) ^[9] or Unscented Kalman Filters (UKF) ^[10] are typically employed. However, rotation matrices require 9 degrees of freedom, while quaternions require 4 degrees of freedom, introducing redundancy that negatively affects computational efficiency. This issue is particularly relevant for real-time UAV applications, where computational resources are limited, and efficient state estimation is required.

Given these challenges, this paper proposes a fusion method that integrates IMU, GPS, or LiDAR data, using the ESKF as the fusion algorithm for UAV pose estimation. This method optimizes the representation of rotational increments, allowing them to be expressed using a minimal three-degree-of-freedom parameter set. By utilizing the error-state formulation, it avoids gimbal lock and singularity issues that arise when describing rotation with three degrees of freedom, thereby improving the accuracy of pose estimation. Additionally, a new Kalman gain calculation formula is introduced, enabling different formulas to be applied when dealing with low-dimensional and high-dimensional observation signals, ensuring computational efficiency.

3. Error-State Kalman Filter algorithm composition

3.1. Problems faced in rotation calculations

In classical Kalman filter algorithms, it is generally assumed that the initial system values, system noise, and observation noise are mutually independent and follow a Gaussian distribution. This assumption requires the vector space defined by the state variables to be closed under addition and scalar multiplication. However, in the process of IMU pose estimation, the rotation quantities are typically defined as rotation matrices or quaternions, with quaternions subject to the constraint: $\|q\| = 1$ and rotation matrices subject to the constraint ^[11]: $RR^T = I$. Obviously, neither of these representations is closed under addition.

When using Euler angles to solve for IMU attitudes, although there are no constraints and both addition and multiplication operations are closed, the classical Kalman filter algorithm can be directly applied. However, as the number of iterations increases, singularities and gimbal lock problems can occur. It can be easily verified that with increasing iterations, the rotation quantities quickly move away from the origin, leading to these issues.

To address these problems, the manifold space error-state Kalman filter algorithm is introduced. Its advantages are: In handling rotation quantities, the error-state Kalman filter can use rotation vectors to directly describe rotations. The error state is a small quantity that always remains near the origin, avoiding singularities and gimbal lock issues caused by an increasing number of iterations. Additionally, its higher-order terms (second order and above) converge easily.

3.2. Manifold space

Let M be the manifold of dimension n in consideration (e.g. $M = SO(3)$). Since manifolds are locally homeomorphic to R^n , we can establish a bijective mapping from a local neighborhood on M to its tangent space R^n via two encapsulation operators (and ‘ ^[12]:

$$(\cdot) : M \times \mathbb{R}^n \rightarrow M; \quad ' : M \times M \rightarrow \mathbb{R}^n$$

$$M = SO(3) : \mathbf{R}(\mathbf{r}) = \mathbf{R} \text{Exp}(\mathbf{r}); \quad \mathbf{R}_1 ' \mathbf{R}_2 = \text{Log}(\mathbf{R}_2^T \mathbf{R}_1)$$

$$M = \mathbb{R}^n : \mathbf{a}(\mathbf{b}) = \mathbf{a} + \mathbf{b}; \quad \mathbf{a}'\mathbf{b} = \mathbf{a} - \mathbf{b}$$

Where $\text{Exp}(\mathbf{r}) = \mathbf{I} + \frac{\mathbf{r}}{\|\mathbf{r}\|} \sin(\|\mathbf{r}\|) + \frac{\mathbf{r}^2}{\|\mathbf{r}\|^2} (1 - \cos(\|\mathbf{r}\|))$ is the exponential map and $\text{Log}(\cdot)$ is its inverse map. For a compound manifold $M = SO(3) \times \mathbb{R}^n$, we have:

$$\begin{bmatrix} \mathbf{R} \\ \mathbf{a} \end{bmatrix} (\begin{bmatrix} \mathbf{r} \\ \mathbf{b} \end{bmatrix}) = \begin{bmatrix} \mathbf{R}(\mathbf{r}) \\ \mathbf{a} + \mathbf{b} \end{bmatrix}; \quad \begin{bmatrix} \mathbf{R}_1 \\ \mathbf{a} \end{bmatrix} ' \begin{bmatrix} \mathbf{R}_2 \\ \mathbf{b} \end{bmatrix} = \begin{bmatrix} \mathbf{R}_1 ' \mathbf{R}_2 \\ \mathbf{a} - \mathbf{b} \end{bmatrix}$$

From the above definition, it is easy to verify that:

$$(\mathbf{x}(\delta))' \mathbf{x} = \delta; \quad \mathbf{x}(\mathbf{y}'\mathbf{x}) = \mathbf{y}; \quad \forall \mathbf{x}, \mathbf{y} \in M, \forall \delta \in \mathbb{R}^n$$

3.3. IMU state model

Generally, the state variables of an IMU are typically represented as:

$$M = SO(3) \times \mathbb{R}^{15}, \dim(M) = 18$$

$$\mathbf{x}(t) \doteq \begin{bmatrix} {}^G \mathbf{R}_I^T & {}^G \mathbf{p}_I^T & {}^G \mathbf{v}_I^T & \mathbf{b}_\omega^T & \mathbf{b}_a^T & {}^G \mathbf{g}^T \end{bmatrix}$$

Where $\mathbf{x}(t)$ is a function of time t .

The kinematic model can be expressed as^[13]:

$$\begin{aligned} {}^G \dot{\mathbf{p}}_I &= {}^G \mathbf{v}_I \\ {}^G \dot{\mathbf{v}}_I &= {}^G \mathbf{R}_I (a_m - \mathbf{b}_a - \mathbf{n}_a) + {}^G \mathbf{g}, {}^G \dot{\mathbf{g}} = 0 \\ {}^G \dot{\mathbf{R}}_I &= {}^G \mathbf{R}_I [\omega_m - \mathbf{b}_\omega - \mathbf{n}_\omega]^\wedge \\ \dot{\mathbf{b}}_\omega &= \mathbf{n}_{b\omega}, \dot{\mathbf{b}}_a = \mathbf{n}_{ba} \end{aligned}$$

In (1) and (2), ${}^G \mathbf{p}_I$ and ${}^G \mathbf{R}_I$ represent the position and attitude of the IMU in the world frame (typically, the first frame of the IMU is defined as the world frame), ${}^G \mathbf{v}_I$ represents the velocity of the IMU in the world frame, \mathbf{b}_ω and \mathbf{b}_a represent the biases of the angular velocity and acceleration, which are modeled as the random walk process with Gaussian noises $\mathbf{n}_{b\omega}$ and \mathbf{n}_{ba} , \mathbf{n}_a and \mathbf{n}_ω represent the Gaussian white noise in the IMU measurement process, ${}^G \mathbf{g}$ represents the gravitational acceleration in the world frame, a_m and ω_m represent the IMU measurements of angular velocity and acceleration, respectively. The notation $[\theta]^\wedge$ denotes the skew-symmetric matrix of vector $\theta \in \mathbb{R}^3$ that maps the cross product operation^[14].

3.4. Error-State Kalman Filter

Define the above states as follows:

$${}^G \mathbf{R}_I^T \doteq \mathbf{R}, \quad {}^G \mathbf{p}_I^T \doteq \mathbf{p}, \quad {}^G \mathbf{v}_I^T \doteq \mathbf{v}, \quad \mathbf{b}_\omega^T \doteq \mathbf{b}_\omega, \quad \mathbf{b}_a^T \doteq \mathbf{b}_a, \quad {}^G \mathbf{g}^T \doteq \mathbf{g}$$

The ESKF algorithm treats the true state as a combination of the nominal state and the error state, expressed as:

$$\mathbf{x}(t) = \mathbf{x}(t) (\delta \mathbf{x}(t))$$

The true state is represented by $\mathbf{x}(t)$, $\mathbf{x}(t)$ and $\delta\mathbf{x}(t)$ are the noise-free nominal state and the error state including noise, respectively. Here $\delta\mathbf{x}(t) \sim N\{0, \mathbf{P}\}$, \mathbf{P} is the covariance matrix of the error state, which can be specifically expanded as:

$$\mathbf{x}(t)_t = \begin{bmatrix} \mathbf{R}_t \\ \mathbf{p}_t \\ \mathbf{v}_t \\ \mathbf{b}_{\omega t} \\ \mathbf{b}_{at} \\ \mathbf{g}_t \end{bmatrix} = \begin{bmatrix} \mathbf{R}(\delta\mathbf{r}) \\ \mathbf{p} + \delta\mathbf{p} \\ \mathbf{v} + \delta\mathbf{v} \\ \mathbf{b}_{\omega} + \delta\mathbf{b}_{\omega} \\ \mathbf{b}_a + \delta\mathbf{b}_a \\ \mathbf{g} + \delta\mathbf{g} \end{bmatrix} \sim (\mathbf{R}(\delta\mathbf{r}) = \mathbf{R} \text{Exp}(\delta\mathbf{r}))$$

In (3) and (4), the nominal state does not include noise terms; the noise terms are included in the error state. Together, they add up to form the true state. During the recursive operations, compared to the nominal state, the error state can be considered as a small quantity containing noise. Thus, we can derive the kinematic models for the nominal states $\dot{\mathbf{x}}(t)$ and $\delta\dot{\mathbf{x}}(t)$ as follows:

$$\dot{\mathbf{x}}(t) = \begin{bmatrix} \dot{\mathbf{R}} \\ \dot{\mathbf{p}} \\ \dot{\mathbf{v}} \\ \dot{\mathbf{b}}_{\omega} \\ \dot{\mathbf{b}}_a \\ \dot{\mathbf{g}} \end{bmatrix} = \begin{bmatrix} \mathbf{R}(\omega_m - \mathbf{b}_a)^\wedge \\ \mathbf{v} \\ \mathbf{R}(a_m - \mathbf{b}_{\omega}) + \mathbf{g} \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

$$\delta\dot{\mathbf{x}}(t) = \begin{bmatrix} \delta\dot{r} \\ \delta\dot{\mathbf{p}} \\ \delta\dot{\mathbf{v}} \\ \delta\dot{\mathbf{b}}_{\omega} \\ \delta\dot{\mathbf{b}}_a \\ \delta\dot{\mathbf{g}} \end{bmatrix} = \begin{bmatrix} -(\omega_m - \mathbf{b}_{\omega})^\wedge \delta r - \delta\mathbf{b}_{\omega} - \mathbf{n}_{\omega} \\ \delta\mathbf{v} \\ -\mathbf{R}(a_m - \mathbf{b}_a)^\wedge \delta r - \mathbf{R}\delta\mathbf{b}_a - \mathbf{n}_a + \delta\mathbf{g} \\ \mathbf{n}_{b\omega} \\ \mathbf{n}_{ba} \\ 0 \end{bmatrix}$$

3.4.1. Discrete-time Error-State Kalman Filter kinematic equations

Based on (5) and (6), we can respectively establish the discrete-time nominal state and error state Kalman kinematic equations:

For the nominal state, we have:

$$\mathbf{x}(t + \Delta t) = \begin{bmatrix} \mathbf{R}(t + \Delta t) \\ \mathbf{p}(t + \Delta t) \\ \mathbf{v}(t + \Delta t) \\ \mathbf{b}_{\omega}(t + \Delta t) \\ \mathbf{b}_a(t + \Delta t) \\ \mathbf{g}(t + \Delta t) \end{bmatrix} = \begin{bmatrix} \mathbf{R}(t) \text{Exp}((\omega_m - \mathbf{b}_a)\Delta t) \\ \mathbf{p}(t) + \mathbf{v}\Delta t + \frac{1}{2}(\mathbf{R}(a_m - \mathbf{b}_{\omega}) + \mathbf{g})\Delta t^2 \\ \mathbf{v}(t) + (\mathbf{R}(a_m - \mathbf{b}_{\omega}) + \mathbf{g})\Delta t \\ \mathbf{b}_{\omega}(t) \\ \mathbf{b}_a(t) \\ \mathbf{g}(t) \end{bmatrix}$$

For the error state, we have:

$$\delta\mathbf{x}(t + \Delta t) = \begin{bmatrix} \delta r(t + \Delta t) \\ \delta\mathbf{p}(t + \Delta t) \\ \delta\mathbf{v}(t + \Delta t) \\ \delta\mathbf{b}_{\omega}(t + \Delta t) \\ \delta\mathbf{b}_a(t + \Delta t) \\ \delta\mathbf{g}(t + \Delta t) \end{bmatrix} = \begin{bmatrix} \text{Exp}(-(\omega_m - \mathbf{b}_{\omega})\Delta t)\delta r - \delta\mathbf{b}_{\omega}\Delta t - r_i \\ \delta\mathbf{p} + \delta\mathbf{v}\Delta t \\ \delta\mathbf{v} + (-\mathbf{R}(a_m - \mathbf{b}_a)^\wedge \delta r - \mathbf{R}\delta\mathbf{b}_a + \delta\mathbf{g})\Delta t + \mathbf{v}_i \\ \delta\mathbf{b}_{\omega} + \omega_i \\ \delta\mathbf{b}_a + a_i \\ \delta\mathbf{g} \end{bmatrix}$$

Where $\text{Var}(r_i) = \sigma_{n_r}^2 \Delta t^2 \mathbf{I}$, $\text{Var}(\mathbf{v}_i) = \sigma_{n_v}^2 \Delta t^2 \mathbf{I}$, $\text{Var}(\omega_i) = \sigma_{b_\omega}^2 \Delta t \mathbf{I}$, $\text{Var}(a_i) = \sigma_{b_a}^2 \Delta t \mathbf{I}$.

3.4.2. Error-State Kalman Filter propagation

To obtain a compact form of expression, we define as follows:

Let k denote the k -th iteration, Δt the time step, \mathbf{x}_k the nominal state vector, $\delta \mathbf{x}_k$ the error state vector, \mathbf{u}_{mk} the input vector, and \mathbf{i} the disturbance pulse vector. Specifically, this can be expanded as follows:

$$\mathbf{x}_k = \begin{bmatrix} \mathbf{R}_k \\ \mathbf{p}_k \\ \mathbf{v}_k \\ \mathbf{b}_{(\omega)k} \\ \mathbf{b}_{(a)k} \\ \mathbf{g}_k \end{bmatrix}, \delta \mathbf{x}_k = \begin{bmatrix} \delta r_k \\ \delta \mathbf{p}_k \\ \delta \mathbf{v}_k \\ \delta \mathbf{b}_{(\omega)k} \\ \delta \mathbf{b}_{(a)k} \\ \delta \mathbf{g}_k \end{bmatrix}, \mathbf{u}_{(m)k} = \begin{bmatrix} \omega_{(m)k} \\ a_{(m)k} \end{bmatrix}, \mathbf{i} = \begin{bmatrix} r_i \\ \mathbf{v}_i \\ \omega_i \\ a_i \end{bmatrix}$$

The error state propagation process is:

$$\delta \mathbf{x}_{k+1} \leftarrow f(\mathbf{x}_k, \delta \mathbf{x}_k, \mathbf{u}_{(m)k}, \mathbf{i}) = \mathbf{F}_x(\mathbf{x}_k, \mathbf{u}_{(m)k}) \cdot \delta \mathbf{x}_k + \mathbf{F}_i \cdot \mathbf{i}$$

The prediction process of the Error-State Kalman Filter is:

$$\widehat{\delta \mathbf{x}}_{k+1}^- \leftarrow \mathbf{F}_x(\widehat{\mathbf{x}}_k, \mathbf{u}_{(m)k}) \cdot \widehat{\delta \mathbf{x}}_k$$

$$\mathbf{P}_{k+1}^- \leftarrow \mathbf{F}_x \mathbf{P}_k \mathbf{F}_x^T + \mathbf{F}_i \mathbf{Q}_i \mathbf{F}_i^T$$

In (9), (10), and (11), $\widehat{\delta \mathbf{x}}_{k+1}^-$ is the prior estimate of the error state, \mathbf{F}_x and \mathbf{F}_i are the Jacobian matrices with respect to the error state and the disturbance, respectively, and \mathbf{Q}_i is the covariance matrix of the disturbance pulse.

$$\mathbf{F}_x = \left. \frac{\partial f}{\partial \delta \mathbf{x}} \right|_{\mathbf{x}, \mathbf{u}_m} = \begin{bmatrix} \text{Exp}(-(\omega_m - \mathbf{b}_g)\Delta t) & 0 & 0 & -\mathbf{I}\Delta t & 0 & 0 \\ 0 & \mathbf{I} & \mathbf{I}\Delta t & 0 & 0 & 0 \\ -\mathbf{R}(a_m - \mathbf{b}_a)\Delta t & 0 & \mathbf{I} & 0 & -\mathbf{R}\Delta t & 0 \\ 0 & 0 & 0 & \mathbf{I} & 0 & 0 \\ 0 & 0 & 0 & 0 & \mathbf{I} & 0 \\ 0 & 0 & 0 & 0 & 0 & \mathbf{I} \end{bmatrix}$$

$$\mathbf{F}_i = \left. \frac{\partial f}{\partial \mathbf{i}} \right|_{\mathbf{x}, \mathbf{u}_m} = \begin{bmatrix} \mathbf{I} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \mathbf{I} & 0 & 0 & 0 \\ 0 & 0 & 0 & \mathbf{I} & 0 & 0 \\ 0 & 0 & 0 & 0 & \mathbf{I} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}, \mathbf{Q}_i = \begin{bmatrix} \sigma_{n_r}^2 \Delta t^2 \mathbf{I} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \sigma_{n_v}^2 \Delta t^2 \mathbf{I} & 0 & 0 & 0 \\ 0 & 0 & 0 & \sigma_{b_\omega}^2 \Delta t \mathbf{I} & 0 & 0 \\ 0 & 0 & 0 & 0 & \sigma_{b_a}^2 \Delta t \mathbf{I} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

3.4.3. Error-State Kalman Filter update process

When observation data is received, such as GPS, BDS, or related visual information, the error state can then be updated.

Suppose there is a set of sensors providing state observation information, we have:

It is evident that this new Kalman gain algorithm can significantly reduce the computational complexity of the inversion operation.

3.4.4. Error-State Kalman Filter reset

Let the reset function be $g()$, then we have:

$$\delta \mathbf{x}_k^{reset} \leftarrow g(\delta \mathbf{x}_k) = (\mathbf{x}_k \quad \delta \mathbf{x}_k)' \left(\mathbf{x}_k \quad \widehat{\delta \mathbf{x}}_k \right)$$

The reset operation for the Error-State Kalman Filter is as follows:

$$\widehat{\delta \mathbf{x}}_k \leftarrow 0$$

$$\mathbf{P}_k \leftarrow \mathbf{G} \mathbf{P}_k \mathbf{G}^T$$

$$\mathbf{G} \triangleq \frac{\partial g}{\partial \delta \mathbf{x}} \bigg|_{\widehat{\delta \mathbf{x}}} = \begin{bmatrix} \mathbf{I} - \frac{1}{2} \delta r_k^\wedge & 0 \\ 0 & \mathbf{I}_{15} \end{bmatrix}$$

4. Experiments and analysis

To evaluate the performance of the ESKF, we conducted a series of simulations using data generated by the GNSS-INS-SIM tool. The following details the simulation setup, data characteristics, and results.

4.1. Simulation results

In this subsection, we present the experimental results of the proposed Manifold Space ESKF algorithm. The experiment used IMU and GPS data for pose estimation.

Figure 1 shows the object's trajectory in the XYZ plane, including the true path, GPS observation data, and the fusion algorithm results. The results demonstrate that the proposed algorithm meets the accuracy requirements, with no significant drift, and closely follows the true path.

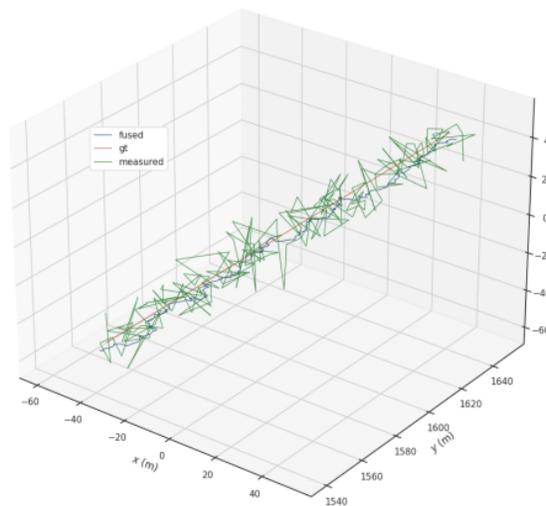


Figure 1. Object trajectory in the XYZ plane

Figure 2 displays the Euler angles corresponding to the fusion results. It can be observed that when the roll angle exceeds 90 degrees, both the pitch and yaw angles remain below 40 degrees, indicating that the proposed algorithm effectively avoids the gimbal lock issue. Figure 3 presents the XYZ three-axis position data corresponding to the trajectory.

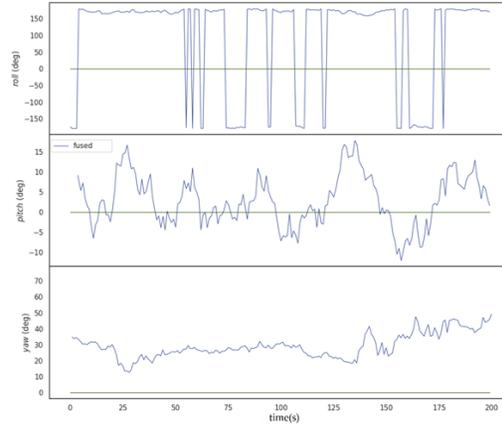


Figure 2. Orientation results for fused sensors

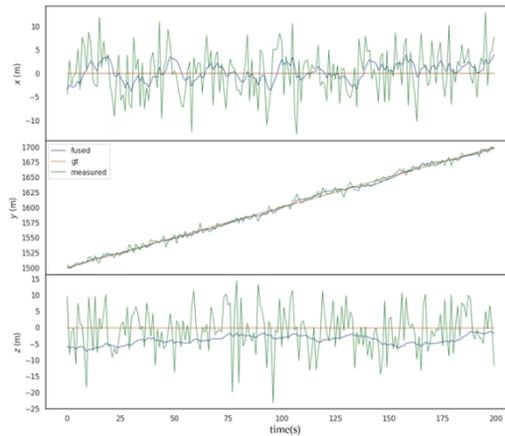


Figure 3. Position corresponding to the trajectory

4.2. Simulation analysis

The specific analysis of the experimental results is as follows: Given the low-dimensional nature of the GPS data, we utilized formula (15)(a) to compute the Kalman gain. In this comparison, we selected Quaternion-based ESKF and Multiplicative EKF for the experiment. The errors are represented by the root mean square error of the absolute pose. The computation times and errors for the three algorithms are summarized in Table 1.

Table 1. Errors and computation times for manifold space, quaternion, and multiplicative extended Kalman filter algorithms

Algorithms	Time (ms)	Error
Manifold space ESKF	79.75	3.546
Quaternion-based ESKF	95.77	4.716
Multiplicative EKF	69.26	4.710

From the results, the Manifold Space Error-State Kalman Filter algorithm demonstrates superior accuracy compared to the other two filtering algorithms. It also exhibits faster computation times than the quaternion-based Error-State Kalman Filter. Quaternions require four dimensions to describe rotation, while rotation matrices require nine dimensions. In contrast, the Manifold Space Error-State Kalman Filter only requires three dimensions to represent rotation, leading to significant memory savings.

For high-dimensional observation data, such as LiDAR image, we employed (15)(b) for Kalman gain calculation. Experimental results for different dimensionalities of images using the two algorithms are presented in Table 2.

Table 2. Computation time for different Kalman gain computation formulas

Dimension	247	618	1046	1532
Old formula (ms)	8.2	34.6	267	1974
New formula (ms)	0.08	0.19	0.49	1.54

5. Conclusion

This paper presents a pose estimation algorithm for UAVs, utilizing the ESKF to overcome the limitations of traditional methods in handling rotational states. By leveraging three-dimensional rotational vectors in the manifold space, the proposed approach achieves efficient and accurate UAV pose estimation while avoiding gimbal lock and singularity issues. Simulation results demonstrate that the ESKF with rotation vector representation significantly outperforms traditional quaternion-based methods and the multiplicative extended Kalman filter (MEKF) in terms of estimation accuracy, making it well-suited for UAV navigation and localization.

Furthermore, this paper introduces a new Kalman gain computation strategy for UAV multi-sensor fusion applications to enhance computational efficiency. For low-dimensional observation data (e.g., from GNSS), the classical Kalman gain formula offers low computational complexity. For high-dimensional observation data (e.g., visual information), the new Kalman gain formula significantly reduces computational complexity. This flexible gain calculation method effectively improves the algorithm's computational efficiency under different data conditions.

However, this study presents certain limitations in UAV sensor adaptability. The current Kalman gain computation method selects the gain formula based only on the dimensionality of the observation data, but it does not support real-time switching to LiDAR-based positioning when GPS signals suddenly become unreliable.

Funding

National Natural Science Foundation of China (Grant No. 62266045); National Science and Technology Major Project of China (No. 2022YFE0138600)

Disclosure statement

The authors declare no conflict of interest.

References

- [1] Alomari A, Phillips W, Aslam N, et al., 2017, Dynamic Fuzzy-Logic Based Path Planning for Mobility-Assisted Localization in Wireless Sensor Networks. *Sensors*, 17: 1904.
- [2] Yurtsever E, Lambert J, Carballo A, et al., 2020, A Survey of Autonomous Driving: Common Practices and Emerging Technologies. *IEEE Access*, 8: 58443–58469.
- [3] Sabatini AM, 2011, Estimating Three-Dimensional Orientation of Human Body Parts by Inertial/Magnetic Sensing. *Sensors*, 11: 1489–1525.
- [4] Sola J, 2017, Quaternion Kinematics for the Error-State Kalman Filter. *arXiv*, <https://doi.org/10.48550/arXiv.1711.02508>
- [5] Zhao S, Zhang H, Wang P, et al., 2021, Super Odometry: IMU-centric LiDAR-Visual-Inertial Estimator for Challenging Environments. *IEEE International Conference on Robotics and Automation (ICRA)*, 2021: 14193–14200.
- [6] Forster C, Carlone L, Dellaert F, et al., 2017, On-Manifold Preintegration for Real-Time Visual-Inertial Odometry. *IEEE Transactions on Robotics*, 33(1): 1–21.
- [7] Qin T, Li P, Shen S, 2018, Vins-Mono: A Robust and Versatile Monocular Visual-Inertial State Estimator. *IEEE Transactions on Robotics*, 34(4): 1004–1020.
- [8] Yu Z, Zhu L, Lu G, 2021, Vins-motion: Tightly-Coupled Fusion of Vins and Motion Constraint. *2021 IEEE International Conference on Robotics and Automation (ICRA)*, 71(6): 5799–5810.
- [9] Bloesch M, Burri M, Omari S, et al., 2017, Iterated Extended Kalman Filter Based Visual-Inertial Odometry Using Direct Photometric Feedback. *International Journal of Robotics Research*, 36(10): 1053–1072.
- [10] Xia R, Pei H, 2020, Ranging-Aided Aerobridge Navigation Using Dual Quaternion Based Multiplicative Extended Kalman Filter. *2020 International Symposium on Autonomous Systems (ISAS)*, 2020: 1–6.
- [11] Xiang G, 2019, *14 Lessons of Visual SLAM: From Theory to Practice* (2019), Electronic Industry Press, Beijing.
- [12] Hertzberg C, Wagner R, Frese U, et al., 2013, Integrating Generic Sensor Fusion Algorithms with Sound State Representations Through Encapsulation of Manifolds. *Information Fusion*, 14(1): 57–77.
- [13] Crassidis JL, Junkins JL, 2011, *Optimal Estimation of Dynamic Systems*. CRC Press, New York.
- [14] Xiang G, 2023, *SLAM in Autonomous Driving and Robotics: From Theory to Practice*, Electronic Industry Press, Beijing.
- [15] Xu W, Zhang F, 2021, Fast-Lio: A Fast, Robust Lidar-Inertial Odometry Package by Tightly-Coupled Iterated Kalman Filter. *IEEE Robotics and Automation Letters*, 6(2): 4675587.

Publisher's note

Bio-Byword Scientific Publishing remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.