sensors

MDPI

Article

Sensor Fusion of GNSS and IMU Data for Robust Localization via Smoothed Error State Kalman Filter

Yuming Yin [1], Jinhong Zhang [1], Mengqi Guo [2], Xiaobin Ning [1,*], Yuan Wang [1] and Jianshan Lu [1]

[1] College of Mechanical Engineering, Zhejiang University of Technology, Hangzhou 310014, China
[2] College of Engineering, Beijing Forestry University, Beijing 100083, China
* Correspondence: nxb@zjut.edu.cn

Abstract: High−precision and robust localization is critical for intelligent vehicle and transportation systems, while the sensor signal loss or variance could dramatically affect the localization performance. The vehicle localization problem in an environment with Global Navigation Satellite System (GNSS) signal errors is investigated in this study. The error state Kalman filtering (ESKF) and Rauch–Tung–Striebel (RTS) smoother are integrated using the data from Inertial Measurement Unit (IMU) and GNSS sensors. A segmented RTS smoothing algorithm is proposed in order to estimate the error state, which is typically close to zero and mostly linear, which allows more accurate linearization and improved state estimation accuracy. The proposed algorithm is evaluated using simulated GNSS signals with and without signal errors. The simulation results demonstrate its superior accuracy and stability for state estimation. The designed ESKF algorithm yielded an approximate 3% improvement in long straight line and turning scenarios compared to classical EKF algorithm. Additionally, the ESKF−RTS algorithm exhibited a 10% increase in the localization accuracy compared to the ESKF algorithm. In the double turning scenarios, the ESKF algorithm resulted in an improvement of about 50% in comparison to the EKF algorithm, while the ESKF−RTS algorithm improved by about 50% compared to the ESKF algorithm. These results indicated that the proposed ESKF−RTS algorithm is more robust and provides more accurate localization.

Keywords: error state Kalman; RTS smoothing; robust and accurate localization

check for updates

Citation: Yin, Y.; Zhang, J.; Guo, M.; Ning, X.; Wang, Y.; Lu, J. Sensor Fusion of GNSS and IMU Data for Robust Localization via Smoothed Error State Kalman Filter. *Sensors* **2023**, *23*, 3676. https://doi.org/10.3390/s23073676

Academic Editors: Shichang Du, Yiping Shao, Delin Huang and Robert Odolinski

Received: 23 February 2023
Revised: 16 March 2023
Accepted: 30 March 2023
Published: 1 April 2023

1. Introduction

The Global Navigation Satellite System (GNSS) has found widespread usage in vehicle measurement, intelligent driving, and robot navigation due to its stable long−term localization performance [1–3]. However, the susceptibility of satellite signals to external interference would lead to the GNSS having unreliable localization accuracy. Therefore, the combination of GNSS and Inertial Navigation System (INS) has been widely employed, in which both systems could compensate for each other's drawbacks and leverage their respective strengths to achieve continuous localization [4–6]. The reported studies have investigated the combined filtering algorithm of GNSS and INS, GNSS/IMU can provide position, velocity, and attitude information for vehicle control. Liu [7,8] presents a novel pitch and roll feedback mechanism that utilizes intrinsic information of the vehicle such as steering angle and wheel speed. To compensate for the cumulative velocity errors that occur during low sampling intervals of GNSS, the integration of reverse smoothing and grey prediction is employed. Shin and Naser [9] reported that the algorithm yielded better navigation and localization results, even in cases of short−term GNSS loss. Han [10] compared the combined satellite/inertial guidance parameter estimation results with those of a single satellite system, with the former yielding superior results. Li and Zhang [11] studied the combined GNSS/INS navigation algorithm and highlighted that the filtering algorithm still struggled to provide accurate parameter estimation in situations where few satellite data were available. Erfianti [12] researched a combined GNSS/INS navigation algorithm and discovered that the INS could only provide high−accuracy parameter estimation results for a short duration. During the satellite signal occlusion, the forward filtering algorithm alone thus cannot guarantee a robust localization.

Applying the extended Kalman filter (EKF) to estimate the motion of vehicle systems is well desirable due to the system nonlinearity [13–16]. The EKF linearizes the nonlinear model by approximating it with a first−order Taylor series around the state estimate and then estimates the state using the Kalman filter. M. M. Atia [17] proposed utilizing the extended Kalman filter to merge data from inertial sensors with GNSS data in a loosely coupled mode to improve the accuracy of road network maps in determining lanes. The resulting lane determination success rate was an impressive 97.14%. However, the local linearization operation of this approach can introduce significant estimation errors. Julier [18] introduced an Unscented Kalman filter algorithm that improves the estimation results, while its high complexity makes it unsuitable for most inertial guidance devices. Arasaratnam [19] et al. pointed out through theoretical analysis and simulation verification that the UKF has poor filtering performance or even divergence in solving the high−dimensional ($\geq$20) nonlinear state estimation problem. For this reason, they used the spherical radial rule to approximate the state posterior distribution in the optimal framework, and then proposed the Cubature Kalman Filter (CKF), but the CKF has higher computational complexity and requires more sampling and operations, resulting in poor real−time performance, which is not suitable for some applications with high real−time performance requirements. The extended Kalman filter thus remains the mainstream state estimation algorithm, and developing a low−complexity filter with high accuracy is still challenging [20,21].

To address this challenge, S. S Kourabbaslou [22] presents a flexible design framework utilizing symbolic engines to represent and linearize system and measurement models. A robust fixed−lag smoothing approach is proposed in case there is a mismatch between the nominal model and the actual model [23,24]. To improve the accuracy of vehicle stand−alone localization in highly dynamic driving conditions during GNSS outages, Gao [25] proposed a vehicle localization system based on vehicle chassis sensors considering vehicle lateral velocity. The Kalman filter combines vehicle states obtained from vehicle kinematics and dynamics to improve the reliability and accuracy of autonomous driving. A consensus−based and vehicle kinematics/dynamics integrated autonomous vehicle sideslip angle estimation algorithm based on GNSS/INS was proposed [26,27]. Madyastha [28] proposed a Kalman filtering method based on attitude error states. The error state Kalman filter (ESKF) is designed for covariance estimation, which serves as a weight for the inertial odometry optimization process [29–31]. These results validate the ESKF's effectiveness for error−state estimation, although a full−state estimation based on ESKF is not available yet. Further exploration of state estimation based on ESKF is thus favorable.

The major challenge in designing a sensor fusion algorithm for state estimation is to address GNSS signal errors and low−cost hardware limitations with a smoother approach. The errors of inertial navigation systems accumulate over time, while GNSS signals are heavily influenced by satellite signal quality and urban obstructions [32–34]. These would lead to significant localization errors. Using INS alone cannot provide accurate position estimation for a long term in signal−obscured environments [35–37]. In such cases, real−time high−accuracy localization is necessary, and data fusion from multiple sensors is often required [38,39]. In order to provide reliable localization information, simultaneous localization and map construction (SLAM) systems can be used, as well as multi−sensor integrated systems that combine LiDAR [40] and navigation systems to solve problems caused by GNSS antenna failures. To reduce GNSS and Inertial Measurement Unit (IMU) fusion costs and achieve desired accuracy, different navigation satellite [41–43] constellations need to be combined, especially in urban working environments.

In this paper, the low−cost sensor combination of only GNSS and IMU is considered, and effects of GNSS signal error and different fusion algorithm design are investigated. The extended Kalman filter and error state Kalman filter are designed in Section 2. The application of the error−extended Kalman filter and the RTS optimal smoothing algorithm is investigated in Section 3 for combined inertial guidance and GNSS localization in urban working environments. The insights and conclusions on the performance of the designed sensor fusion algorithms are further analyzed.

2. GNSS and IMU Integrated Filter Design

The overall sensor fusion framework integrating the GNSS and IMU sensor data with significant GNSS signal errors is illustrated in Figure 1. It mainly consists of four procedures, including data analysis, prediction process, update process and reverse smoothing, contributing to the developed ESKF−RTS smoothing localization algorithm. In particular, the algorithm defines a nominal state without considering the measurement noise of IMU and system disturbance, and an error state containing the noise and disturbance information is used for state estimation. The nominal state condition and error state prediction are updated simultaneously, and the error state is corrected using GNSS signal measurement and injected into the nominal state. The error state and its covariance matrix are subsequently reset. The acquisition frequency for GNSS data is 1 Hz, while the IMU data are acquired at a frequency of 100 Hz. The reverse time update and reverse segmentation smoothing are parallelly performed, which will be detailed below.

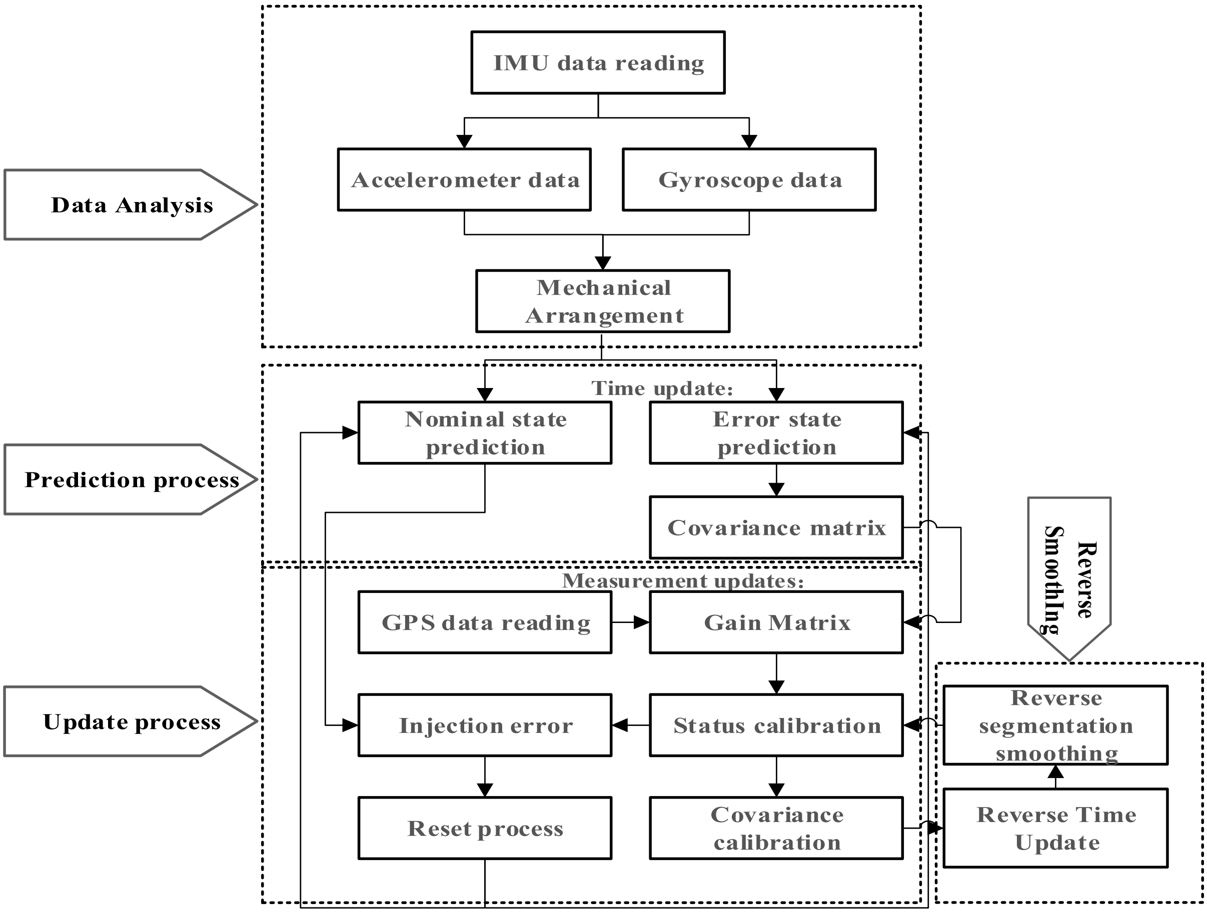

Figure 1. Flow chart of the designed ESKF−RTS algorithm.

2.1. Extended Kalman Filter

The extended Kalman filter (EKF) is also designed for comparison, using the sensor measurements directly. The EKF procedure is described briefly.

The developed discrete nonlinear vehicle system equations are:

$$\begin{cases} \boldsymbol{X}_k = f(\boldsymbol{X}_{k-1}) + \boldsymbol{W}_{k-1} \\ \boldsymbol{Z}_K = h(\boldsymbol{X}_k) + \boldsymbol{V}_k \end{cases}, \tag{1}$$

where $f(\cdot)$ and $h(\cdot)$ represent the state function and measurement function of nonlinear vehicle systems, $\boldsymbol{W}$ and $\boldsymbol{V}$ are Gaussian white noise of state $\boldsymbol{X}$ and measurement $\boldsymbol{Z}$, respectively, and k represents the discrete time step.

The state function of the nonlinear vehicle system in Equation (1) is approximated through local linearization. The nonlinear equation is expanded using a Taylor series, and only the first−order term is retained while ignoring the second and higher−order terms. This yields the following expression:

$$\boldsymbol{X}_k = f(\breve{\boldsymbol{X}}_{k-1}) + \boldsymbol{F}_{k-1}(\boldsymbol{X}_{k-1} - \breve{\boldsymbol{X}}_{k-1}) + \boldsymbol{W}_{k-1}, \tag{2}$$

where $\boldsymbol{F}_{k-1}$ is the Jacobian matrix of the function of the function $f(\boldsymbol{X}_{k-1})$.

The statistical characteristics of the predicted state can be obtained based on the results of the linearization in the form of

$$\overset{\frown}{\boldsymbol{X}}_{k,k-1} = E[f(\breve{\boldsymbol{X}}_{k-1}) + \boldsymbol{F}_{k-1}(\boldsymbol{X}_{k-1} - \breve{\boldsymbol{X}}_{k-1}) + \boldsymbol{W}_{k-1}] = f(\breve{\boldsymbol{X}}_{k-1}). \tag{3}$$

It is possible to further predict the variance matrix as

$$\begin{aligned} \boldsymbol{P}_{k,k-1} &= E[(\boldsymbol{X}_k - \breve{\boldsymbol{X}}_{k,k-1})(\boldsymbol{X}_k - \breve{\boldsymbol{X}}_{k,k-1})^T] \\ &= E\left\{[\boldsymbol{F}_{k-1}(\boldsymbol{X}_{k-1} - \breve{\boldsymbol{X}}_{k-1}) + \boldsymbol{W}_{k-1}][\boldsymbol{F}_{k-1}(\boldsymbol{X}_{k-1} - \breve{\boldsymbol{X}}_{k-1}) + \boldsymbol{W}_{k-1}]^T\right\}. \\ &= \boldsymbol{F}_{k-1}\boldsymbol{P}_{k-1}\boldsymbol{F}_{k-1}^T + \boldsymbol{Q}_{k-1} \end{aligned} \tag{4}$$

The same method is used for the measurement function $h(\boldsymbol{x}_k)$ at the point $\hat{\boldsymbol{x}}_k$, which can be obtained as

$$\boldsymbol{Z}_K = h(\breve{\boldsymbol{X}}_k) + \boldsymbol{H}_k(\boldsymbol{X}_k - \breve{\boldsymbol{X}}_k) + \boldsymbol{V}_k, \tag{5}$$

where $\boldsymbol{H}_k$ is the Jacobian matrix of the function of the function $h(\boldsymbol{X}_{k-1})$. The prediction of measurement is calculated as

$$\overset{\frown}{\boldsymbol{Z}}_{k,k-1} = E[h(\breve{\boldsymbol{X}}_{k-1}) + H_{k-1}(\boldsymbol{X}_{k-1} - \breve{\boldsymbol{X}}_{k-1}) + \boldsymbol{V}_{k-1}] = h(\breve{\boldsymbol{X}}_{k-1}). \tag{6}$$

Then, the collaborative variance can be further predicted as

$$\begin{aligned} \boldsymbol{P}_{k,k-1} &= E[(\boldsymbol{Z}_k - \breve{\boldsymbol{Z}}_{k,k-1})(\boldsymbol{Z}_k - \breve{\boldsymbol{Z}}_{k,k-1})^T] \\ &= E\left\{[H_{k-1}(\boldsymbol{X}_{k-1} - \breve{\boldsymbol{X}}_{k-1}) + \boldsymbol{V}_{k-1}][H_{k-1}(\boldsymbol{X}_{k-1} - \breve{\boldsymbol{X}}_{k-1}) + \boldsymbol{V}_{k-1}]^T\right\}. \\ &= \boldsymbol{H}_{k-1}\boldsymbol{P}_{k-1}\boldsymbol{H}_{k-1}^T + \boldsymbol{R}_{k-1} \end{aligned} \tag{7}$$

Using the information above, the iterative process of the EKF algorithm for the sensor fused localization can be summarized as

$$\begin{aligned} &\breve{\boldsymbol{X}}_k = f(\breve{\boldsymbol{X}}_{k-1}) \\ &\breve{\boldsymbol{P}}_k = \boldsymbol{F}_{k-1}\boldsymbol{P}_{k-1}\boldsymbol{F}_{k-1}^T + \boldsymbol{Q}_{k-1} \\ &\boldsymbol{K}_{k+1} = \breve{\boldsymbol{P}}^{-}_{k+1}\boldsymbol{H}_{k+1}^T(\boldsymbol{H}_{k+1}\breve{\boldsymbol{P}}^{-}_{k+1}\boldsymbol{H}_{k+1}^T + \boldsymbol{V}_{k+1})^{-1}. \\ &\boldsymbol{X}_{k+1} = \boldsymbol{K}_{k+1}(\boldsymbol{Z}_{k+1} - \boldsymbol{h}(\breve{\boldsymbol{X}}_{k+1})) \\ &\boldsymbol{P}_{k+1} = (\boldsymbol{I} - \boldsymbol{K}_{k+1}\boldsymbol{H}_{k+1})\breve{\boldsymbol{P}}^{-}_{k+1} \end{aligned} \tag{8}$$

2.2. Error State Kalman filter

For the formulation of the error state filter, the system states are defined as the true, nominal, and error state values. The true state is expressed as a combination of the nominal and error states. The approach is to treat the nominal state as a dominant signal, which is highly nonlinear, and the error state as a small signal, which is linearly effective and suitable for linear Gaussian filtering. The nominal state vector of $[\boldsymbol{p}, \boldsymbol{v}, \boldsymbol{q}, \boldsymbol{a_b}, \boldsymbol{\omega_b}]^T$ is used in this study, where the additional dimension appears due to the quaternion representation used for rotation, the quaternion method is widely used in attitude update because it only requires the calculation of matrix differential equations, which is relatively small in computation and can avoid the singular value problem of Euler angles. In this paper, the quaternion method is used to solve the attitude of the carrier. The operators $\boldsymbol{q}$ and $\boldsymbol{R}$ represent, respectively, the quaternion corresponding to the axial angular vector $\boldsymbol{\theta}$ and its rotation matrix. In addition, the error state vector is $[\delta\boldsymbol{p}, \delta\boldsymbol{v}, \delta\boldsymbol{q}, \delta\boldsymbol{a_b}, \delta\boldsymbol{\omega_b}]^T$. The relevant symbol definitions of true, nominal, and error state are listed and compared in Table 1.

Table 1. The definition of variable symbols.

Name	True State	Nominal State	Error States
All states	$\boldsymbol{x_t}$	$\boldsymbol{x}$	$\delta\boldsymbol{x}$
Location	$\boldsymbol{p_t}$	$\boldsymbol{p}$	$\delta\boldsymbol{p}$
Speed	$\boldsymbol{v_t}$	$\boldsymbol{v}$	$\delta\boldsymbol{v}$
Quaternion	$\boldsymbol{q_t}$	$\boldsymbol{q}$	$\delta\boldsymbol{q}$
Rotation matrix	$\boldsymbol{R_t}$	$\boldsymbol{R}$	$\delta\boldsymbol{R}$
Angular vectors	$\boldsymbol{\theta_t}$	$\boldsymbol{\theta}$	$\delta\boldsymbol{\theta}$
Acceleration bias	$\boldsymbol{a_{bt}}$	$\boldsymbol{a_b}$	$\delta\boldsymbol{a_b}$
Angular velocity bias	$\boldsymbol{\omega_{bt}}$	$\boldsymbol{\omega_b}$	$\delta\boldsymbol{\omega_b}$

2.2.1. Continuous Time Kinetic Model

The nominal state kinematics refers to the system modelled with system noise or external perturbations in the form of

$$\begin{aligned} &\dot{\boldsymbol{p}}_t = \boldsymbol{v}_t \\ &\dot{\boldsymbol{v}}_t = \boldsymbol{R}_t(\boldsymbol{a}_m - \boldsymbol{a}_{bt} - \boldsymbol{a}_n) + \boldsymbol{g}_t \\ &\dot{\boldsymbol{q}}_t = \tfrac{1}{2}\boldsymbol{q}_t \otimes (\boldsymbol{\omega}_m - \boldsymbol{\omega}_{bt} - \boldsymbol{\omega}_n). \\ &\dot{\boldsymbol{a}}_{bt} = \boldsymbol{a}_\omega \\ &\dot{\boldsymbol{\omega}}_{bt} = \boldsymbol{\omega}_\omega \end{aligned} \tag{9}$$

The acceleration and angular velocity measurements are represented by $\boldsymbol{a}_m$ and $\boldsymbol{\omega}_m$, respectively, while $\boldsymbol{a}_n$ and $\boldsymbol{\omega}_n$ represent the corresponding noise vectors, and $\boldsymbol{a}_\omega$ and $\boldsymbol{\omega}_\omega$ represent the bias vectors for acceleration and angular velocity, respectively. The equations driving the system dynamics in the nominal state are as follows, where the nominal state refers to the modeled system without any noise or perturbations:

$$\begin{aligned} &\dot{\boldsymbol{p}} = \boldsymbol{v} \\ &\dot{\boldsymbol{v}} = \boldsymbol{R}(\boldsymbol{a}_m - \boldsymbol{a}_b) + \boldsymbol{g} \\ &\dot{\boldsymbol{q}} = \tfrac{1}{2}\boldsymbol{q} \otimes (\boldsymbol{\omega}_m - \boldsymbol{\omega}_b). \\ &\dot{\boldsymbol{a}}_b = 0 \\ &\dot{\boldsymbol{\omega}}_b = 0 \end{aligned} \tag{10}$$

Solving for the error state and simplifying all second−order infinitesimals. The kinetic equation for the error state can be obtained from Equations (9) and (10).

$$\begin{aligned} &\mathit{ffi}\dot{\boldsymbol{p}} = \mathit{ffi}\boldsymbol{v} \\ &\mathit{ffi}\dot{\boldsymbol{v}} = \boldsymbol{R}(\boldsymbol{a}_m - \boldsymbol{a}_b)\mathit{ffi}\boldsymbol{\theta} - R\mathit{ffi}\boldsymbol{a}_b - R\boldsymbol{a}_n \\ &\mathit{ffi}\dot{\boldsymbol{\theta}} = -(\boldsymbol{\omega}_m - \boldsymbol{\omega}_b)\mathit{ffi}\text{`} - \mathit{ffi}\boldsymbol{\omega}_b - \boldsymbol{\omega}_n. \\ &\mathit{ffi}\dot{\boldsymbol{a}}_b = \boldsymbol{a}_\omega \\ &\mathit{ffi}\dot{\boldsymbol{\omega}}_b = \boldsymbol{\omega}_\omega \end{aligned} \tag{11}$$

2.2.2. Discrete Time Kinetic Model

In order to apply the derived state and measurement equations to the sensor fusion filter, they must be discretized based on the sampling time. The original continuous time equations are transformed into their discrete time form.

The recursive expression for the motion model of the nominal state is

$$\begin{aligned} &\boldsymbol{p}_{(k+1)} = \boldsymbol{p}_k + \boldsymbol{v}_k\Delta t + \tfrac{1}{2}[\boldsymbol{R}_k(\boldsymbol{a}_{mk} - \boldsymbol{a}_{bk}) + \boldsymbol{g}]\Delta t^2 \\ &\boldsymbol{v}_{(k+1)} = \boldsymbol{v}_k + [\boldsymbol{R}_k(\boldsymbol{a}_{mk} - \boldsymbol{a}_{bk}) + \boldsymbol{g}_k]\Delta t \\ &\boldsymbol{q}_{(k+1)} = \boldsymbol{q}_k \otimes \boldsymbol{q}_k((\boldsymbol{\omega}_{mk} - \boldsymbol{\omega}_{bk})\Delta t) \\ &\boldsymbol{a}_b(k+1) = \boldsymbol{a}_{bk} \\ &\boldsymbol{\omega}_b(k+1) = \boldsymbol{\omega}_{bk} \end{aligned} \tag{12}$$

The kinematic model of the error state is expressed recursively as a function of the non−negative integer k representing the kth time step:

$$\begin{aligned} &\mathit{ffi}\boldsymbol{p}_{k+1} = \mathit{ffi}\boldsymbol{p}_k + \mathit{ffi}\boldsymbol{v}_k\Delta t \\ &\mathit{ffi}\boldsymbol{v}_{k+1} = [-\boldsymbol{R}_k(a_{mk} - a_{bk})\mathit{ffi}\boldsymbol{\theta}_k - \boldsymbol{R}_k\mathit{ffi}\boldsymbol{a}_{bk}]\Delta t + \mathit{ffi}\boldsymbol{v}_k - \boldsymbol{w}_{vk} \\ &\mathit{ffi}\boldsymbol{\theta}_{k+1} = \boldsymbol{R}_k^T((\boldsymbol{\omega}_{mk} - \boldsymbol{\omega}_{bk})\Delta t)\mathit{ffi}\boldsymbol{\theta}_k - \mathit{ffi}\boldsymbol{\omega}_{bk}\Delta t + \boldsymbol{w}_{\text{`}k} \\ &\mathit{ffi}\boldsymbol{a}_{b(k+1)} = \mathit{ffi}\boldsymbol{a}_{bk} + \boldsymbol{w}_{ak} \\ &\mathit{ffi}\boldsymbol{\omega}_{b(k+1)} = \mathit{ffi}\boldsymbol{\omega}_{bk} + \boldsymbol{w}_{wk} \end{aligned} \tag{13}$$

The Gauss random noise for velocity, attitude, acceleration bias, and angular velocity bias are denoted as $\boldsymbol{w}_{vk}, \boldsymbol{w}_{\text{`}k}, \boldsymbol{w}_{ak}, , \boldsymbol{w}_{!k}$, their mean is zero, and their covariance matrices are obtained by integrating the covariances of $\sigma_{\boldsymbol{a}_n}$, $\sigma_{\boldsymbol{\omega}_n}$, $\sigma_{\boldsymbol{a}_\omega}$ and $\sigma_{\boldsymbol{\omega}_\omega}$ over the step time Δt.

$$\begin{aligned} &\boldsymbol{W}_v = \sigma^2_{\boldsymbol{a}_n}\Delta t^2\boldsymbol{I} \\ &\boldsymbol{W}_{\text{`}} = \sigma^2_{\boldsymbol{\omega}_n}\Delta t^2\boldsymbol{I} \\ &\boldsymbol{W}_A = \sigma^2_{\boldsymbol{a}_!}\Delta t\boldsymbol{I} \\ &\boldsymbol{W}_\Omega = \sigma^2_{\boldsymbol{\omega}_!\Delta t}\boldsymbol{I} \end{aligned} \tag{14}$$

The standard deviation of Gaussian white noise for acceleration and angular velocity are denoted by $\sigma_{\boldsymbol{a}_n}$ and $\sigma_{\boldsymbol{\omega}_n}$, respectively. Similarly, $\sigma_{\boldsymbol{a}_\omega}$ and $\sigma_{\boldsymbol{\omega}_\omega}$ are used to represent the standard deviation of Gaussian white noise for acceleration and angular velocity bias, respectively.

2.2.3. Development of the Error State Model

The discrete vector forms for all states, error states, IMU measurements, and noise are defined as follows:

$$\begin{aligned} &\boldsymbol{x}_k = [\boldsymbol{x}_k, \boldsymbol{v}_k, \boldsymbol{q}_k, \boldsymbol{a}_{bk}, \boldsymbol{\omega}_{bk}]^T, \mathit{ffi}\boldsymbol{x}_k = [\mathit{ffi}\boldsymbol{p}_k, \mathit{ffi}\boldsymbol{v}_k, \mathit{ffi}\boldsymbol{q}_k, \mathit{ffi}\boldsymbol{a}_{bk}, \mathit{ffi}\boldsymbol{\omega}_{bk}]^T, \\ &\boldsymbol{u}_{mk} = [\boldsymbol{a}_{mk}, \boldsymbol{\omega}_{mk}]^T, \boldsymbol{w}_k = [\boldsymbol{w}_{vk}, \boldsymbol{w}_{\text{`}k}, \boldsymbol{w}_{ak}, \boldsymbol{w}_{!k}]^T \end{aligned} \tag{15}$$

The recursive equation for all the error state is derived by combining Equations (5) and (7) in the form of

$$\mathit{ffi}\boldsymbol{x}_{k+1} = \boldsymbol{f}_{\mathit{ffi}}(\boldsymbol{x}_k, \mathit{ffi}\boldsymbol{x}_k, \boldsymbol{u}_{mk}, \boldsymbol{w}_k) = \boldsymbol{F}_{xk}(\boldsymbol{x}_k, \boldsymbol{u}_{mk}) + \boldsymbol{F}_{wk}\boldsymbol{w}_k, \tag{16}$$

where $f_\delta(\cdot)$ represents the recursive function of the error state, the F_{xk} and F_{wk} are the Jacobian matrices corresponding to the error and noise states, respectively, which can be derived as

$$\boldsymbol{F}_{xk} = \begin{bmatrix} \boldsymbol{I} & \boldsymbol{I}\Delta t & 0 & 0 & 0 \\ 0 & \boldsymbol{I} & -\boldsymbol{R}_k^T(\boldsymbol{\omega}_{mk} - \boldsymbol{\omega}_{bk})\Delta t & -\boldsymbol{R}_k\Delta t & 0 \\ 0 & 0 & \boldsymbol{R}_k^T(\boldsymbol{\omega}_{mk} - \boldsymbol{\omega}_{bk})\Delta t & 0 & -\boldsymbol{I}\Delta t \\ 0 & 0 & 0 & \boldsymbol{I} & 0 \\ 0 & 0 & 0 & 0 & \boldsymbol{I} \end{bmatrix}, \boldsymbol{F}_{wk} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ \boldsymbol{I} & 0 & 0 & 0 \\ 0 & \boldsymbol{I} & 0 & 0 \\ 0 & 0 & \boldsymbol{I} & 0 \\ 0 & 0 & 0 & \boldsymbol{I} \end{bmatrix}. \tag{17}$$

2.2.4. The ESKF Prediction Process

The error states and the covariance prediction process can be obtained as

$$\begin{aligned} \mathit{ffi}\breve{\boldsymbol{x}}^{-}_{k+1} &= \boldsymbol{F}_{xk}(\overset{\frown}{\boldsymbol{x}}_k, \boldsymbol{u}_{mk})\mathit{ffi}\boldsymbol{x}_k \\ \breve{\boldsymbol{P}}^{-}_{(k+1)} &= \boldsymbol{F}_{xk}\breve{\boldsymbol{P}}_k\boldsymbol{F}_{xk}^T + \boldsymbol{F}_{wk}\boldsymbol{Q}_w\boldsymbol{F}_{wk}^T \end{aligned} \tag{18}$$

where $\boldsymbol{Q}_w$ represents the noise covariance matrix of the form:

$$\boldsymbol{Q}_w = \begin{bmatrix} \boldsymbol{W}_v & 0 & 0 & 0 \\ 0 & \boldsymbol{W}_\theta & 0 & 0 \\ 0 & 0 & \boldsymbol{W}_A & 0 \\ 0 & 0 & 0 & \boldsymbol{W}_\Omega \end{bmatrix}. \tag{19}$$

2.2.5. The ESKF Observation Process

Once the GNSS information becomes available, the observation is incorporated to continuously update the filter. This process also involves calibrating for the accelerometer and gyroscope biases.

The observation equation is typically expressed in a more uniform form within the filter in the form of

$$z_k = \boldsymbol{h}(\boldsymbol{x}_{tk}) + \boldsymbol{w}_{mk}, \tag{20}$$

where z_k is the measurement signal vector, $\boldsymbol{w}_{mk}$ represents the Gaussian white noise of the measurement signal, and its covariance is $\boldsymbol{V}$. The error calibration update equation is derived as

$$\begin{aligned} \boldsymbol{K}_{k+1} &= \breve{\boldsymbol{P}}^{-}_{k+1}\boldsymbol{H}^T_{k+1}(\boldsymbol{H}_{k+1}\breve{\boldsymbol{P}}^{-}_{k+1}\boldsymbol{H}^T_{k+1} + \boldsymbol{V}_{k+1})^{-1} \\ \mathit{ffi}\overset{\frown}{\boldsymbol{X}}_{k+1} &= \boldsymbol{K}_{k+1}(\boldsymbol{Z}_{k+1} - \boldsymbol{H}(\breve{\boldsymbol{X}}_{k+1})) \\ \overset{\frown}{\boldsymbol{P}}_{k+1} &= (\boldsymbol{I} - \boldsymbol{K}_{k+1}\boldsymbol{H}_{k+1})\breve{\boldsymbol{P}}^{-}_{k+1} \end{aligned} \tag{21}$$

The Jacobian matrix $\boldsymbol{H}$ is required to be defined with respect to the error state $\mathit{ffi}\boldsymbol{x}$, and evaluated at the best true state estimate $\hat{\boldsymbol{x}}_t = \boldsymbol{x} \oplus \mathit{ffi}\hat{\boldsymbol{x}}$. As the error state mean is zero at this stage (we have not observed it yet), we have $\boldsymbol{x}_t = \boldsymbol{x}$, and we can use the nominal error x as the evaluation point, leading to

$$\boldsymbol{H} = \left.\frac{\partial \boldsymbol{h}}{\partial \boldsymbol{x}_t}\right|_x \left.\frac{\partial \boldsymbol{x}_t}{\partial \mathit{ffi}\boldsymbol{x}}\right|_x = \boldsymbol{H}_x\boldsymbol{X}_{\mathit{ffi}x}, \tag{22}$$

in which

$$\boldsymbol{X}_{\mathit{ffi}x} \triangleq \left.\frac{\partial \boldsymbol{x}_t}{\partial \mathit{ffi}\boldsymbol{x}}\right|_x = \begin{bmatrix} \boldsymbol{I}_6 & 0 & 0 \\ 0 & \boldsymbol{Q}_{\mathit{ffi}\boldsymbol{\theta}} & 0 \\ 0 & 0 & \boldsymbol{I}_6 \end{bmatrix}, \tag{23}$$

$$\boldsymbol{Q}_{\mathit{ffi}\boldsymbol{\theta}} = \frac{1}{2}\begin{bmatrix} -q_x & -q_y & -q_z \\ q_w & -q_z & q_y \\ q_z & q_w & -q_x \\ -q_y & q_x & q_w \end{bmatrix}. \tag{24}$$

2.2.6. Combination of ESKF Error State and Nominal State Process

Based on the recurrence of the a priori system state and the calibration of the error state, the updated form is obtained as follows:

$$\overset{\frown}{\boldsymbol{x}}_k = \overset{\frown}{\boldsymbol{x}}_k^- \oplus ffi\overset{\frown}{\boldsymbol{x}}_k. \tag{25}$$

Each of these system states corresponds to the following:

$$\begin{aligned} \overset{\frown}{\boldsymbol{p}}_k &= \overset{\frown}{\boldsymbol{p}}_k^- + ffi\overset{\frown}{\boldsymbol{p}}_k \\ \overset{\frown}{\boldsymbol{v}}_k &= \overset{\frown}{\boldsymbol{v}}_k^- + ffi\overset{\frown}{\boldsymbol{v}}_k \\ \overset{\frown}{\boldsymbol{q}}_k &= \overset{\frown}{\boldsymbol{q}}_k^- \otimes \overset{\frown}{\boldsymbol{q}} ffi\overset{\frown}{\boldsymbol{\theta}}_k. \\ \overset{\frown}{\boldsymbol{a}}_{bk} &= \overset{\frown}{\boldsymbol{a}}_{bk}^- + ffi\overset{\frown}{\boldsymbol{a}}_{bk} \\ \overset{\frown}{\boldsymbol{\omega}}_{bk} &= \overset{\frown}{\boldsymbol{\omega}}_{bk}^- + ffi\overset{\frown}{\boldsymbol{\omega}}_{bk} \end{aligned} \tag{26}$$

2.2.7. ESKF Error State Reset

After injecting the error state into the nominal state, the a priori error state and the corresponding covariance need to be reset, as

$$\begin{aligned} & ffi\overset{\frown}{\boldsymbol{x}} = 0 \\ & \boldsymbol{P} = \boldsymbol{G}_k \boldsymbol{P}_k \boldsymbol{G}_k^T \end{aligned} \tag{27}$$

where $\boldsymbol{G}$ is the reset function $\boldsymbol{g}(ffi\boldsymbol{x}) = ffi\boldsymbol{x} \ominus ffi\hat{\boldsymbol{x}}$ of the Jacobi matrix, which is defined in the form of

$$\boldsymbol{G} \triangleq \left.\frac{\partial g}{\partial ffi\boldsymbol{x}}\right|_{ffi\overset{\frown}{x}} = \begin{pmatrix} \boldsymbol{I}_6 & 0 & 0 \\ 0 & \boldsymbol{I} - (\frac{1}{2} ffi\overset{\frown}{\boldsymbol{\theta}}) & 0 \\ 0 & 0 & \boldsymbol{I}_9 \end{pmatrix}. \tag{28}$$

3. Robust Localization via RTS Smoothing

3.1. Reliability of Measurement Information

In the case of a GNSS signal loss, when there are no available GNSS measurements to update, the covariance matrix in Equation (21) tends to be infinite. The Kalman gain K tends to zero, so that when the measurement information is zero or differs too much from the predicted information, the update equation changes to the following form:

$$\begin{aligned} & ffi\overset{\frown}{\boldsymbol{x}}_{k+1} = ffi\overset{\frown}{\boldsymbol{x}}_k \\ & \overset{\frown}{\boldsymbol{P}}_{k+1} = \overset{\smile}{\boldsymbol{P}}_k^- \end{aligned} \tag{29}$$

3.2. RTS Fundamental Design

The RTS smoothing can be regarded as a technique for obtaining an optimal state estimate when observations are available from moment 1 to moment N; it involves using previous estimates obtained through Kalman filtering in order to perform backward smoothing from moment k + 1 to moment k resulting in a more precise estimate. This method falls under the category of fixed interval smoothing and is considered as a fixed interval smoother, which is convenient for implementation.

The RTS smoother involves a two−step process: a forward filter followed by a backward smoothing. The forward filter is a standard Kalman filter described by Equation (18), which maintains all the predicted and updated estimates as well as their corresponding covariances for each epoch during the entire mission. The backward smoothing procedure begins at the end of the forward filter at time t_N, with an initial condition $ffix_{N,N}$ and $t_{N,N}$. Therefore, the backward smoothing can be seen as an update to the forward filter for obtaining an enhanced estimate. The equation for the RTS algorithm can be represented in the form of

$$\begin{aligned} \mathit{ffi}\widehat{\boldsymbol{x}}_{k,N} &= \mathit{ffi}\widehat{\boldsymbol{x}}_{k,k} + \boldsymbol{U}_k(\widehat{\boldsymbol{x}}_{k+1,N} - \widehat{\boldsymbol{x}}_{k+1,k}) \\ \widehat{\boldsymbol{P}}_{k,N} &= \widehat{\boldsymbol{P}}_{k,k} + \boldsymbol{U}_k(\widehat{\boldsymbol{P}}_{k+1,N} - \widehat{\boldsymbol{P}}_{k+1,k})\boldsymbol{U}_k^T \end{aligned} \tag{30}$$

where delta $\mathit{ffi}x_{k,N}$ is the smoothed estimate of the state vector, and U_k is the smoothing gain matrix, which is calculated by the following equation:

$$\boldsymbol{U}_k = \widehat{\boldsymbol{P}}_{k,k}\boldsymbol{F}_{k+1,k}^T\widehat{\boldsymbol{P}}_{k+1,k}^{-1}, \tag{31}$$

where $k = N-1, N-2, \ldots\ldots, 0$. The RTS smoothing algorithm uses the results of the forward filter and backward smoothing to obtain an improved estimate. The inverse smoothing starts from the last filter result and proceeds forward one by one, so it must obtain N filter results to smooth the observations within the period [0, N]. However, if N is too large, the inverse smoothing process can be obviously lagged, which would limit the algorithm's application in real time. Therefore, a segmented RTS smoothing method is designed in this study.

3.3. Segmented RTS Smoothing Algorithm

Assuming the duration of a dynamic system is k = 1, 2 . . . . . . , N, the segmented RTS smoothing algorithm involves forward filtering and inverse smoothing of N observations in segments. The segment length is L, where 1 < L < N. In other words, the RTS inverse smoothing process is performed immediately after obtaining the filtering results for the segmented period, without waiting for subsequent filtering results. This approach significantly reduces the lag time of the inverse smoothing process. Additionally, the segmentation length can be varied, avoiding the issue of poor real−time performance associated with RTS smoothing. In this study, the segmented RTS smoothing algorithm is applied to the potential probability assumption density filtering algorithm, effectively addressing the problem of poor real−time performance resulting from the use of RTS smoothing.

Step 1 Output variables are cleared: $\mathit{ffi}\hat{x}_k = [\,]$, $\hat{p}_k = [\,]$.

Step 2 Local filtering result variable clearing $\mathit{ffi}\check{x}_k = [\,]$,$\check{p}_k = [\,]$. The filtered result variable after local association is cleared $x_k = [\,]$, $Q_k = [\,]$, the segmented smoothing counter *count* is cleared to zero.

Step 3 If the tracking of the target is not finished, action continues down the track, otherwise the smoothing result $\mathit{ffi}\hat{x}_k$, $\hat{p}_k$ is output; then, the entire segmented RTS smoothing algorithm ends.

Step 4 Upon receiving the current filtering result, it is stored in the variables $\mathit{ffi}\check{x}_k$ and $\check{p}_k$. the Hungarian algorithm is used to correlate the trajectories and estimates, obtaining the filter value for each target. The correlated filter result is saved in variables x_k and Q_k, and they are counted using a counter called count.

Step 5 If $count == L$, after performing local correlation using x_k and Q_k for each target's filtering results, the inverse smoothing process uses the RTS smoothing algorithm. The results of the smoothing process are stored in variables delta $\mathit{ffi}\check{x}_k$ and $\check{p}_k$ before being transferred to Step 2. In order to significantly reduce the lag time of the inverse smoothing process, it is carried out immediately following the acquisition of the filtered data without waiting for subsequent data. Real−time performance of the fusion algorithm can be improved by decreasing the segment length L.

To correct the position, velocity, and attitude values that have been calculated in the forward filtering, it is recommended to use the state error smoothing values. These smoothing values can be utilized to derive the final optimal smoothed values for these states using the following equations.

4. Results and Discussions

The proposed sensor fusion algorithm is demonstrated in a relatively open environment, which allows for uninterrupted satellite signal and individualized GNSS localization. The acquisition frequency for GNSS data is 1 Hz, while the IMU data are acquired at a frequency of 100 Hz; the smooth dimension L is selected as 10. These parameters provide acceptable conditions for analyzing the accuracy of combined GNSS/IMU localization, even in the event of GNSS signal loss, as demonstrated by the trajectory shown in Figure 2. In order to evaluate the localization accuracy of the GNSS/IMU combined localization process, the simulation involves artificially inducing the loss of lock by introducing exaggerated errors to the GNSS satellite observation information based on raw GNSS data.

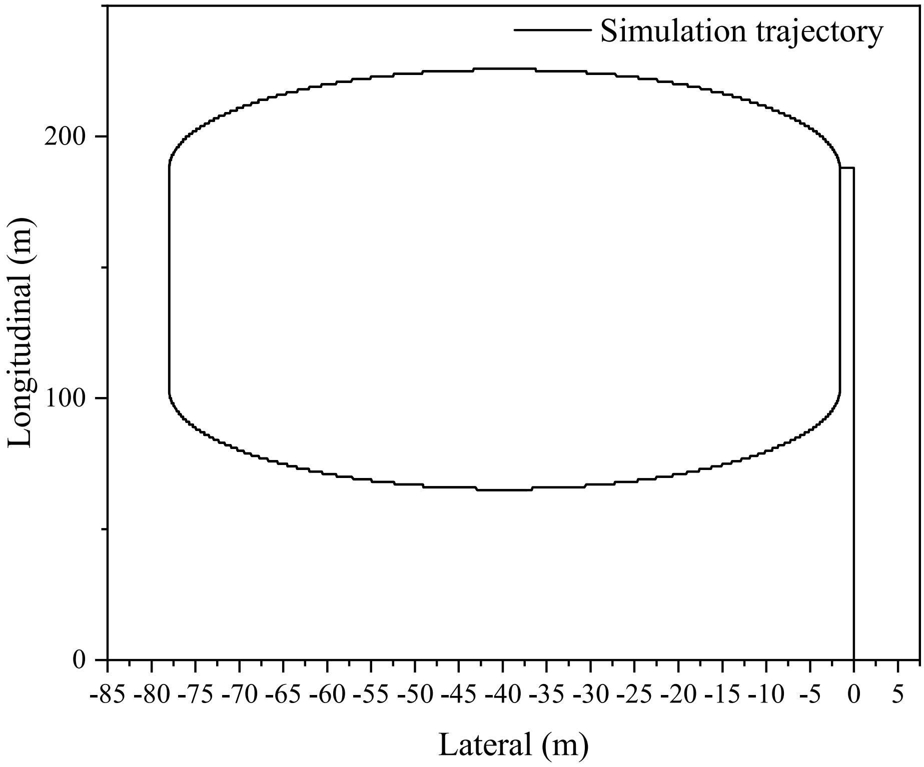

Figure 2. Oval shape motion trajectory.

4.1. Oval Track Simulation Analysis

Figure 2 illustrates the simulated oval shape trajectory, which comprises a long straight line with small angle turns in two dimensions.

The localization results for the various filtering algorithms are shown in Figures 3–5 and Table 2. It can be seen that the ESKF algorithm exhibits better target tracking accuracy compared to the EKF algorithm, as evidenced by its RMS values of 1.633 m, 1.782 m, and 1.476 m for Lateral, Longitudinal, and Vertical directions, respectively. By reducing the error state parameter, the accuracy of the three poses is improved by 2.8%, 2.1%, and 52.0%, respectively. After performing the backward smoothing filtering process, the RMS values of the three attitudes are further optimized to 1.463 m, 1.588 m, and 1.393 m, respectively. Moreover, the target tracking accuracy of ESKF–RTS is superior to ESKF, with the attitude accuracy in the three directions improving by 10.4%, 10.9%, and 5.6%, respectively. The trajectory also reveals that the curve of ESKF–RTS is smoother, further highlighting the advantages of ESKF–RTS over the other two algorithms in addressing the problem of smooth estimation of nonlinear states.

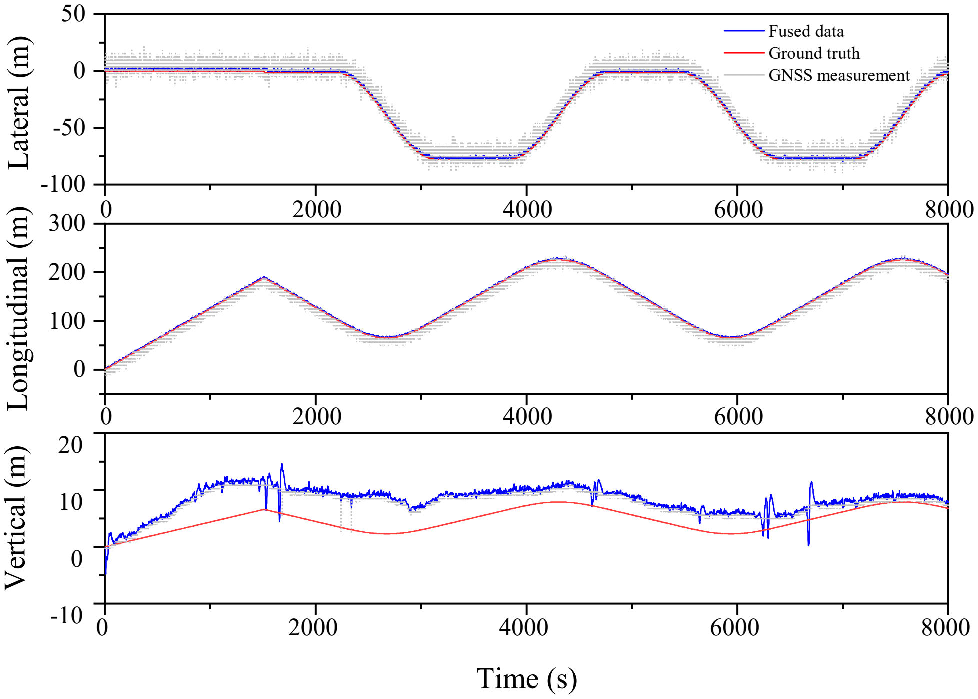

Figure 3. Lateral, Longitudinal and Vertical positions of EKF (Oval).

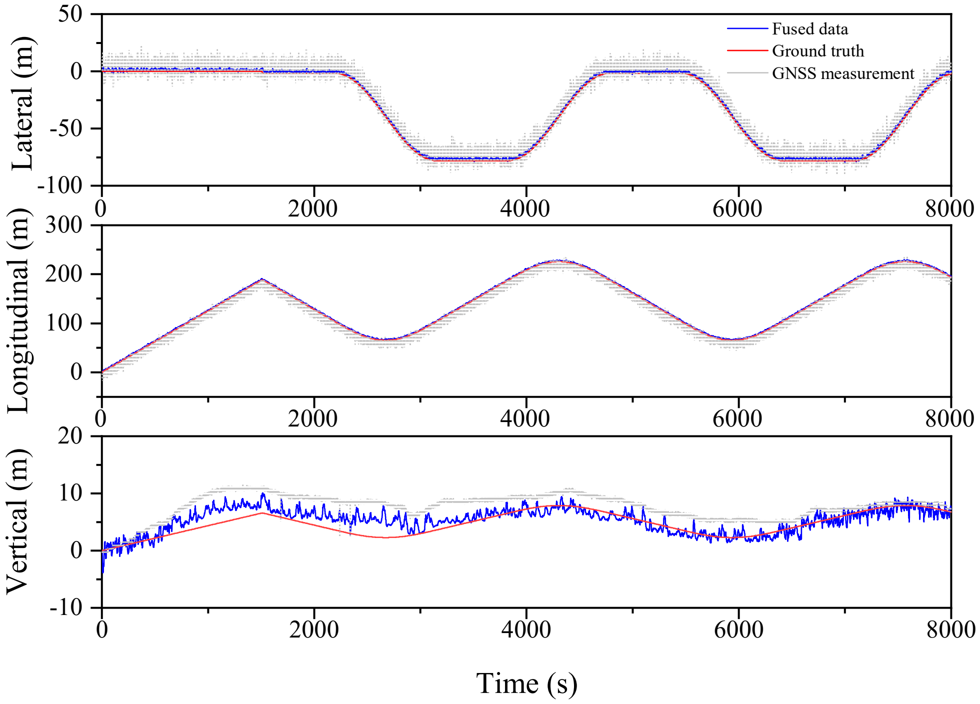

Figure 4. Lateral, Longitudinal and Vertical positions of ESKF (Oval).

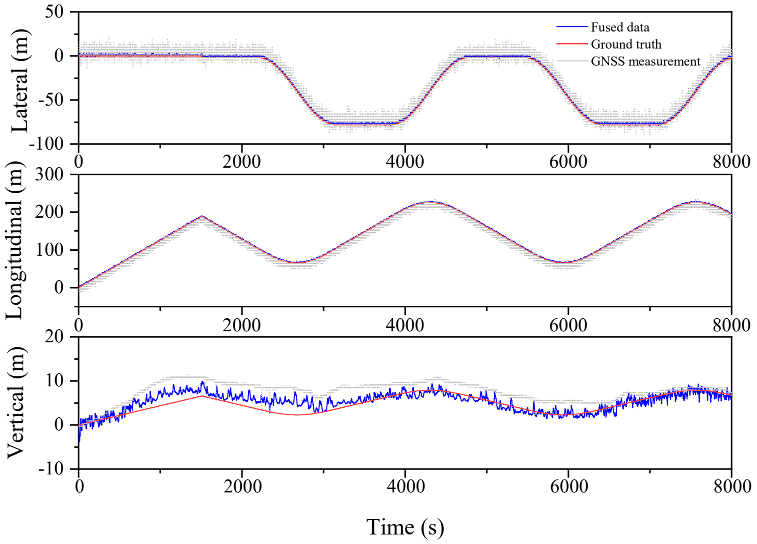

Figure 5. Lateral, Longitudinal and Vertical positions of ESKF–RTS (Oval).

Table 2. RMS values of localization errors for different algorithms (Oval).

Position Error RMS (m)	Lateral	Longitudinal	Vertical
GNSS	2.049	2.598	2.619
EKF	1.680	1.820	3.075
ESKF	1.633	1.782	1.476
ESKF–RTS	1.463	1.588	1.393

4.2. Serpentine Track Simulation Analysis

Further verification of the designed algorithms is performed in a serpentine trajectory, as shown in Figure 6, and the localization results are compared as shown in Figures 7–9 and Table 3.

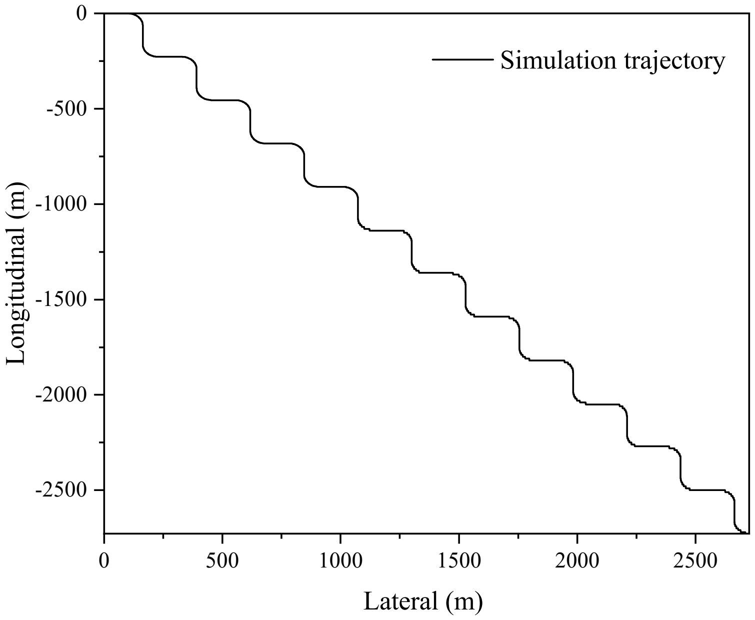

Figure 6. Serpentine shape motion trajectory.

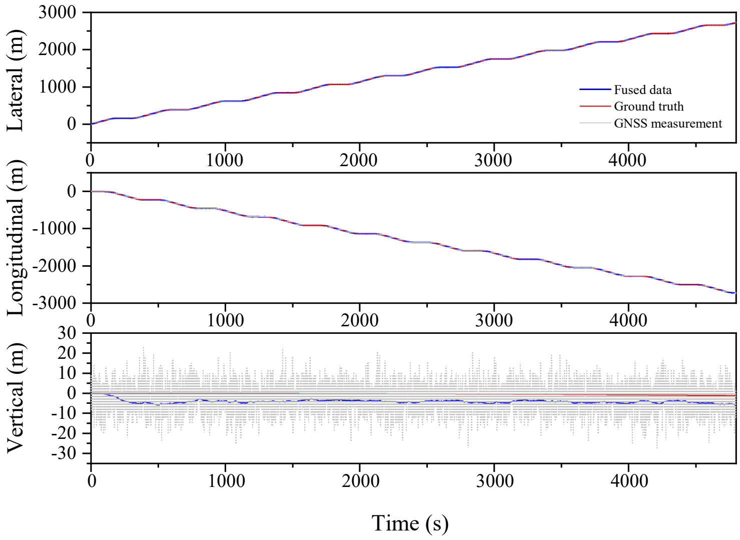

Figure 7. Lateral, Longitudinal and Vertical positions of EKF (Serpentine 1).

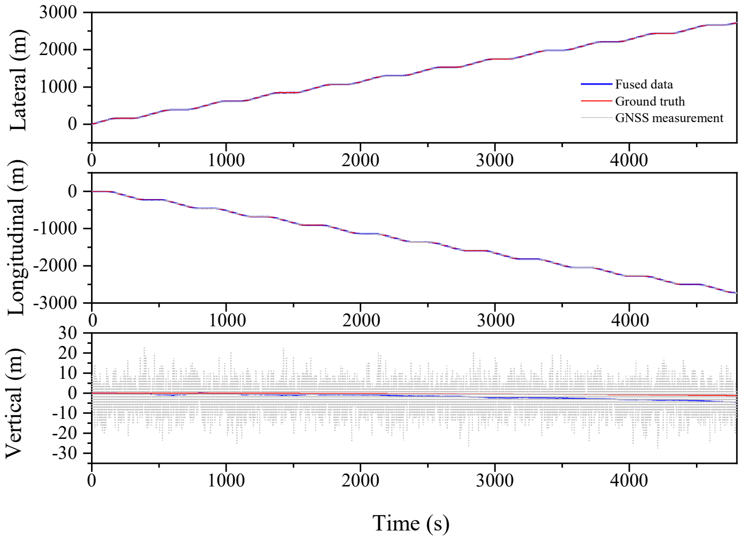

Figure 8. Lateral, Longitudinal and Vertical positions of ESKF (Serpentine 1).

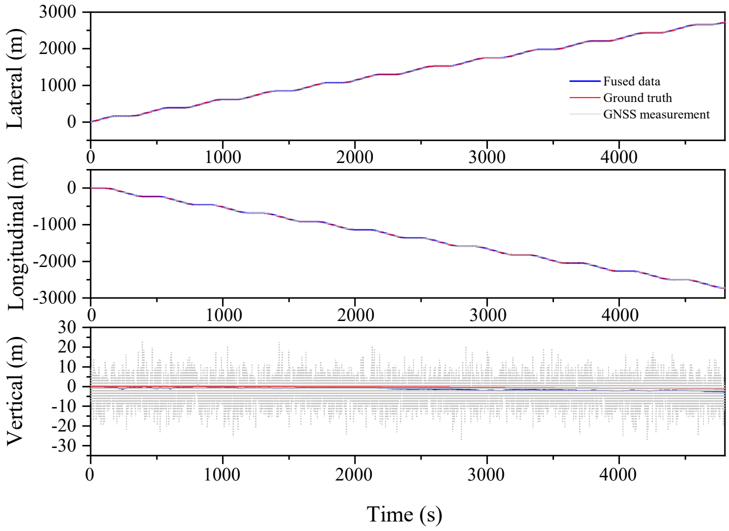

Figure 9. Lateral, Longitudinal and Vertical positions of ESKF−RTS (Serpentine 1).

Table 3. RMS values of localization errors for different algorithms (Serpentine 1).

Position Error RMS (m)	Lateral	Longitudinal	Vertical
GNSS	1.912	1.793	5.680
EKF	0.955	1.585	5.823
ESKF	0.464	0.641	1.700
ESKF−RTS	0.206	0.243	0.912

Table 3 illustrates the improved performance of the ESKF and ESKF−RTS algorithms compared to the EKF algorithm in the serpentine condition. The RMS of Lateral, Longitudinal, and Vertical in EKF are 0.955 m, 1.585 m, and 5.823 m, respectively. After reducing the state value, the RMS of the three positions for the ESKF algorithm are 0.464 m, 0.641 m, and 1.700 m, which are 51.4%, 59.6%, and 70.8% better than the EKF algorithm. The ESKF−RTS algorithm further improves the position accuracy, with RMS values of 0.206 m, 0.243 m, and 0.912 m for the three directions, respectively, which are 55.6%, 62.1%, and 46.4% higher than those of the ESKF algorithm. Figures 7–9 show the comparison of the root mean square errors, which further demonstrates the superior performance and stability of the ESKF−RTS algorithm. However, since the simulation was conducted with a relatively good GNSS signal, the robustness of the algorithm could not be well evaluated. Subsequently, the GNSS data for the serpentine condition are partitioned into eight segments, and the error magnitude is amplified by a factor of two at intervals of 600 s in order to evaluate the algorithm's robustness.

The serpentine working condition with further exaggerated GNSS signal error is analyzed, as shown in Figures 10–12 and Table 4. The RMS values for the Lateral, Longitudinal, and Vertical directions are 1.231 m, 1.735 m, and 1.453 m, respectively, for the EKF algorithm. Meanwhile, the ESKF algorithm improves these values by 48.4%, 48.7%, and 34.1%, respectively, resulting in the RMS values of 0.635 m, 0.890 m, and 0.957 m. Moreover, the ESKF−RTS algorithm improves the position accuracy in three directions by 42.1%, 52.6%, and 52.1%, respectively, compared to the ESKF algorithm, with RMS values of 0.368 m, 0.422 m, and 1.456 m. Even though the accuracy of ESKF−RTS decreases on the Vertical axis, the higher degree of smoothing in Figure 12 suggests that the ESKF−RTS smoothing algorithm can significantly enhance robustness and achieve precise localization, even with lower accuracy of GNSS sensors.

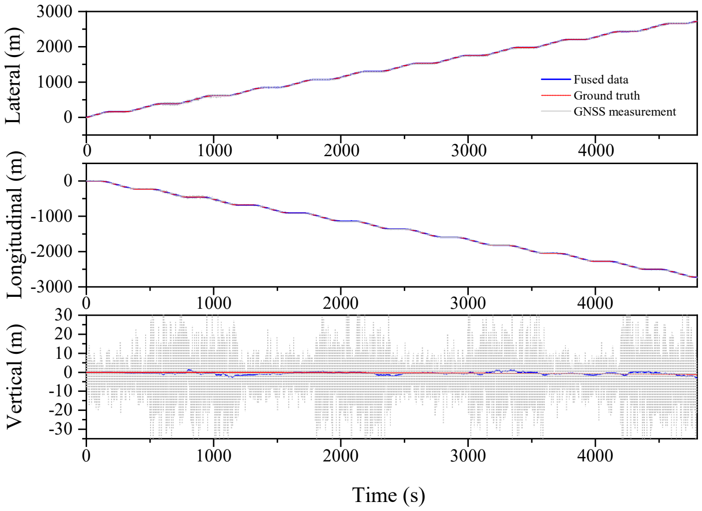

Figure 10. Lateral, Longitudinal and Vertical positions of EKF (Serpentine 2).

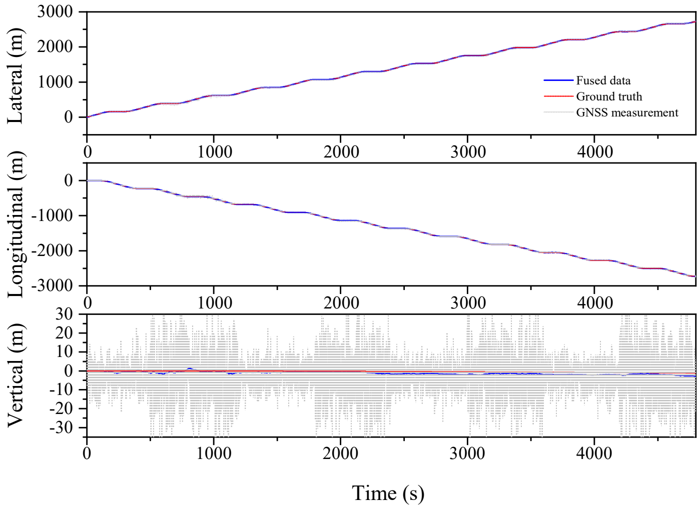

Figure 11. Lateral, Longitudinal and Vertical positions of ESKF (Serpentine 2).

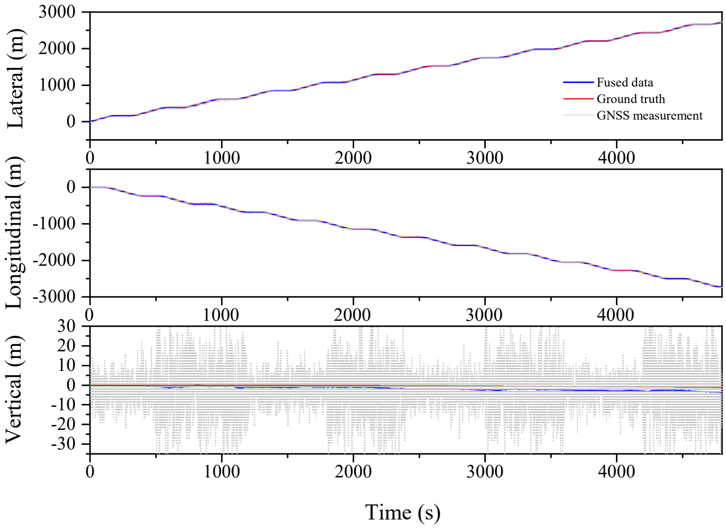

Figure 12. Lateral, Longitudinal and Vertical positions of ESKF−RTS (Serpentine 2).

Table 4. RMS values of localization errors for different algorithms (Serpentine 2).

Position Error RMS (m)	Lateral	Longitudinal	Vertical
GNSS	3.370	3.308	8.988
EKF	1.231	1.735	1.453
ESKF	0.635	0.890	0.957
ESKF−RTS	0.368	0.422	1.456

Next, we continued the simulation of circular operating conditions and divided the entire simulation process into four sections to validate the feasibility of the algorithm by doubling the GNSS signal error, as shown in Figures 13–16. The experimental results shown in Table 5 once again demonstrated the feasibility and robustness of the error state Kalman filtering algorithm, indicating that the algorithm can achieve stable and accurate integrated navigation under various operating conditions.

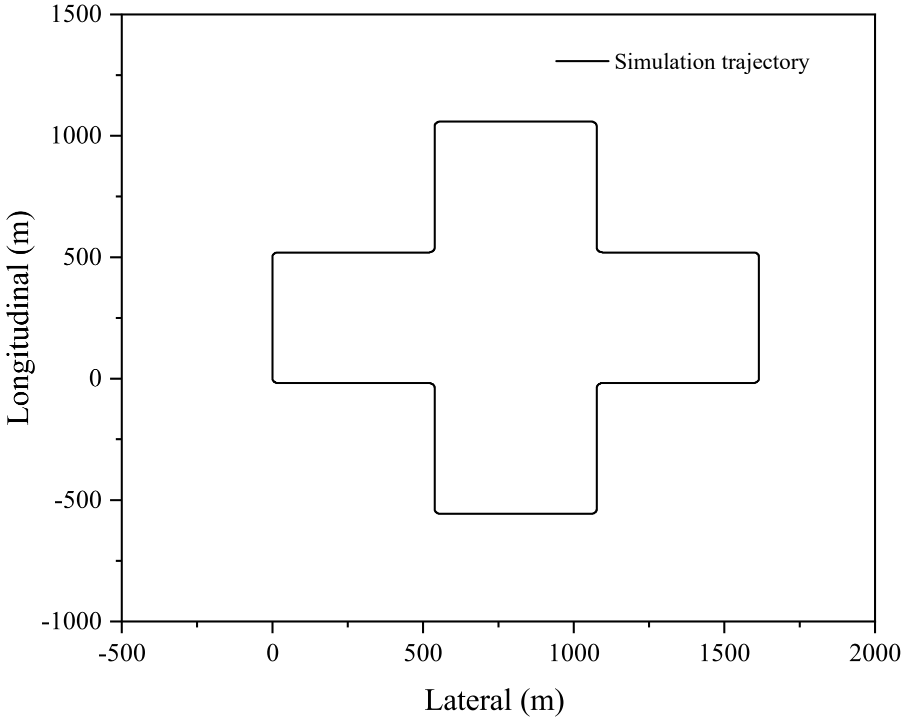

Figure 13. Motion trajectory of polygon.

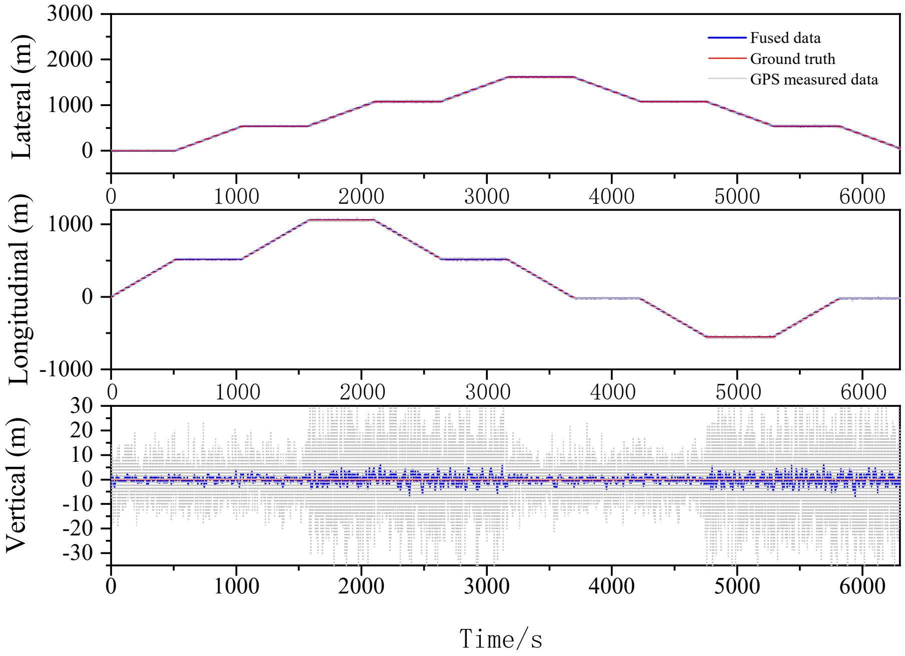

Figure 14. Lateral, Longitudinal and Vertical positions of EKF (Polygonal).

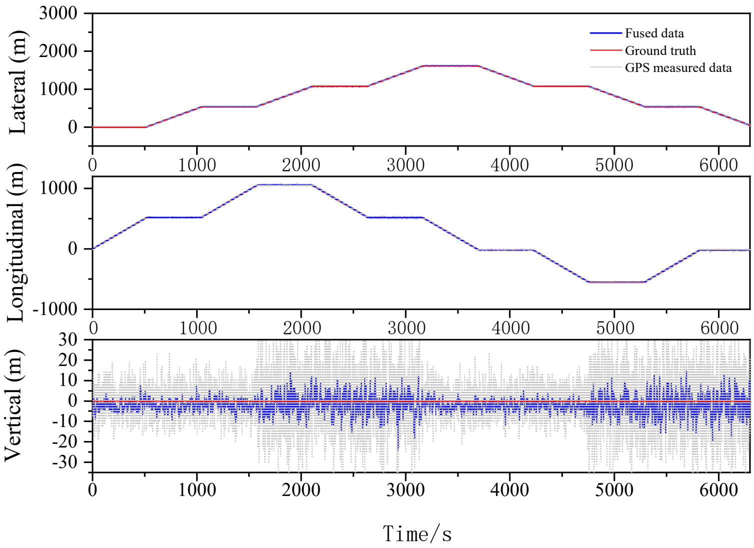

Figure 15. Lateral, Longitudinal and Vertical positions of ESKF (Polygonal).

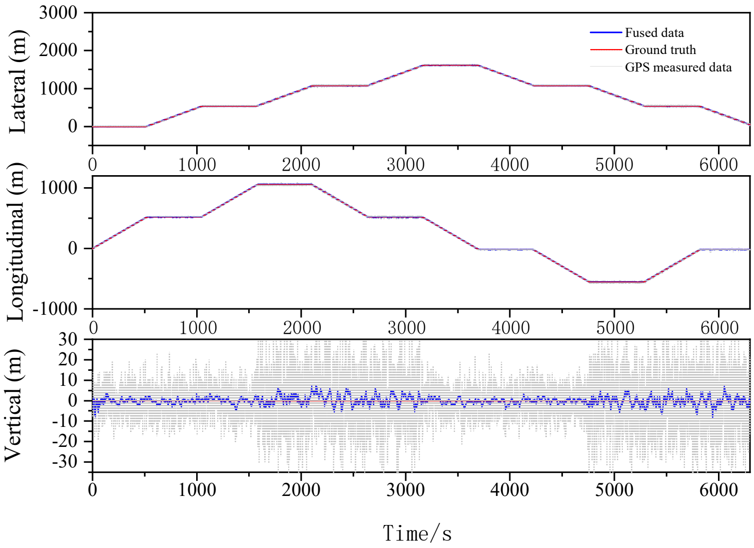

Figure 16. Lateral, Longitudinal and Vertical positions of ESKF−RTS (Polygonal).

Table 5. RMS values of localization errors for different algorithms (Polygonal).

Position Error RMS (m)	Lateral	Longitudinal	Vertical
GNSS	2.601	2.677	8.516
EKF	1.243	1.289	4.328
ESKF	1.253	1.320	1.237
ESKF−RTS	0.900	0.896	1.999

The above data show the results of testing the navigation system using EKF, ESKF, and ESKF−RTS algorithms. By analyzing these results, we can draw the following conclusions. The above data demonstrate the performance of three different filtering algorithms in GNSS navigation. First, we can see that the performance of the EKF and ESKF algorithms is relatively similar, with position errors around 1.3 m. This is because both algorithms use Kalman filtering, but the ESKF algorithm introduces error states while considering errors, which can more accurately estimate errors and improve accuracy. At the same time, the ESKF algorithm can also reduce the impact of errors in future time by predicting error states, thus improving the stability of the algorithm.

On the other hand, the ESKF−RTS algorithm performs even better, with position errors even lower than 1 m. This is because the ESKF−RTS algorithm uses segmental smoothing to optimize the filtering results, which can more accurately estimate and correct errors. In the ESKF−RTS algorithm, RTS stands for "Recursive Least Squares Smoothing" which can combine the prediction results of the ESKF algorithm with the observation results to obtain more accurate state estimation results.

Overall, the above data indicate that the ESKF−RTS algorithm performs well in GNSS navigation. The advantage of this algorithm is that it not only considers error states, but also further optimizes the filtering results through smoothing algorithms. Therefore, in practical applications, selecting the ESKF−RTS algorithm for navigation filtering can achieve more accurate and stable results, improving the reliability and accuracy of the navigation system.

5. Conclusions and Future Work

This paper introduces an error state extended Kalman filter algorithm and segmented Rauch–Tung–Striebel (RTS) smoothing algorithm to enhance the localization accuracy and robustness of GNSS and IMU sensors. The cumulative error of INS over time are overcome when GNSS signal is disturbed. The simulation results show that the proposed method is more linear and has higher localization accuracy than the traditional EKF algorithm. It also demonstrates good robustness in achieving better accuracy with low−quality GNSS signals. The algorithm can serve as a foundation for low−cost sensor fusion processing and is a valuable reference for further research.

Over the next few years, combined navigation systems will see increased usage and development across a range of applications. As sensor and communication technologies progress, and the demands for navigation safety and reliability continue to rise, there will be a greater need for more accurate and dependable navigation systems.

Furthermore, the RTS smoothing algorithm−assisted combined navigation algorithm presented in this paper was simulated for offline computation. For practical applications in the future, real−world testing will be necessary to verify the hardware feasibility of the algorithm, and to consider real−time data transmission and computation to achieve unmanned driving. Overall, positioning is a crucial component of intelligent driving, and its potential impact on society and the economy is vast. Combined navigation will be more and more widely used in production and practical life.

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

Funding: This research was funded by the Zhejiang Provincial Natural Science Foundation of China, grant number LY22E050018; This research was funded by the Basic Public Welfare Research Program of Zhejiang Province, grant number LGG22E050019; the National Natural Science Foundation of China (NSFC), grant number 51905483.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data that support the findings of this study are available from the corresponding author, Xiaobin Ning, upon reasonable request.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

GNSS	Global Navigation Satellite System
IMU	Inertial Measurement Unit
INS	Inertial Navigation System
RTS	Rauch–Tung–Striebel
EKF	Extended Kalman filter
CKF	Cubature Kalman Filter
ESKF	Error state Kalman filter
RMS	Root mean square
Symbols	
X	System state
$\overset{\smile}{X}$	Prediction of the system state
$\overset{\frown}{\boldsymbol{X}}$	Update of the system state
$\delta\overset{\frown}{x}_{k,N}$	The smoothed estimate of the state vector
U_k	The smoothing gain matrix
$\overset{\frown}{\boldsymbol{P}}_{k,N}$	The covariance of smoothing estimates
W	Gaussian white noise
Z	System measurement
V_k	Measurement of Gaussian white noise
$f(\cdot)$	State functions of nonlinear systems
$h(\cdot)$	Measurement functions for nonlinear systems
F_k	The Jacobi matrix of $f(\cdot)$ at x_k
H_k	The Jacobi matrix of $h(\cdot)$ at x_k
p_k	Covariance matrix of states
q_k	Covariance matrix of noise
δ	Error character
x_t	True state
x	Nominal state
p_t	Position at time t
v_t	Velocity at time t
q_t	Quaternion at time t
R_t	Rotation matrix at time t
θ_t	The angular vector at time t
a_{bt}	Acceleration bias at time t
ω_{bt}	Angular velocity bias at time t
a_m	Acceleration measurement
ω_m	Angular velocity measurement
a_n	Acceleration noise vector
ω_n	Angular velocity noise vector
a_ω	Acceleration bias vector
ω_ω	Angular velocity bias vector
Δt	Time interval from k to k+1
w_{vk}	Velocity Gaussian random noise
$w_{\theta k}$	Angular Gaussian random noise
w_{ak}	Acceleration bias Gaussian random noise
$w_{\omega k}$	Velocity biased Gaussian random noise
W_v	Corresponding covariance matrix of $\boldsymbol{w}_{vk}$
v	Corresponding covariance matrix of $\boldsymbol{w}_{mk}$
σ_{a_n}	Acceleration Gaussian white noise
F_{xk}	Error state Jacobi matrix
F_{wk}	Noise state Jacobi matrix
q_w	Noise covariance matrix
U_k	Smoothing gain matrix
$count$	Smooth counter

References

1. Jiang, H.; Shi, C.; Li, T.; Dong, Y.; Li, Y.; Jing, G. Low-cost GNSS/INS integration with accurate measurement modeling using an extended state observer. *GNSS Solut.* **2020**, *25*, 17.
2. Jiang, W.; Liu, D.; Cai, B.; Rizos, C.; Wang, J.; Shangguan, W. A Fault-Tolerant Tightly Coupled GNSS/INS/OVS Integration Vehicle Navigation System Based on an FDP Algorithm. *IEEE Trans. Veh. Technol.* **2019**, *68*, 6365–6378. [CrossRef]
3. Shu, Y.; Xu, P.; Niu, X.; Chen, Q.; Qiao, L.; Liu, J. High-Rate Attitude Determination of Moving Vehicles With GNSS: GPS, BDS, GLONASS, and Galileo. *IEEE Trans. Instrum. Meas.* **2022**, *71*, 5501813. [CrossRef]
4. Li, T.; Zhang, H.; Niu, X.; Gao, Z. Tightly-Coupled Integration of Multi-GNSS Single-Frequency RTK and MEMS-IMU for Enhanced Positioning Performance. *Sensors* **2017**, *17*, 2462. [CrossRef]
5. Miller, I.; Campbell, M. Sensitivity Analysis of a Tightly-Coupled GNSS/INS System for Autonomous Navigation. *IEEE Trans. Aerosp. Electron. Syst.* **2012**, *48*, 1115–1135. [CrossRef]
6. Shi, B.; Wang, M.; Wang, Y.; Bai, Y.; Lin, K.; Yang, F. Effect Analysis of GNSS/INS Processing Strategy for Sufficient Utilization of Urban Environment Observations. *Sensors* **2021**, *21*, 620. [CrossRef] [PubMed]
7. Liu, W.; Xiong, L.; Xia, X.; Lu, Y.; Gao, L.; Song, S. Vision-aided intelligent vehicle sideslip angle estimation based on a dynamic model. *IET Intell. Transp. Syst.* **2020**, *14*, 1183–1189. [CrossRef]
8. Liu, W.; Xia, X.; Xiong, L.; Lu, Y.; Gao, L.; Yu, Z. Automated Vehicle Sideslip Angle Estimation Considering Signal Measurement Characteristic. *IEEE Sens. J.* **2021**, *21*, 21675–21687. [CrossRef]
9. Shin, E.H.; El-Sheimy, N. Accuracy Improvement of Low Cost INS/GNSS for Land Applications. In Proceedings of the 2002 National Technical Meeting of the Institute of Navigation, San Diego, CA, USA, 28–30 January 2002.
10. Han, H.; Wang, J.; Wang, J.; Tan, X. Performance Analysis on Carrier Phase-Based Tightly-Coupled GNSS/BDS/INS Integration in GNSS Degraded and Denied Environments. *Sensors* **2015**, *15*, 8685–8711. [CrossRef]
11. Li, T.; Zhang, H.; Niu, X.; Zhang, Q. Performance Analysis of Tightly Coupled RTK/INS Algorithm in Case of Insufficient Number of Satellites. *Geomat. Inf. Sci. Wuhan Univ.* **2018**, *43*, 478–484.
12. Erfianti, R.; Asfihani, T.; Suhandri, H.F. GNSS/IMU Sensor Fusion Performance Comparison of a Car Localization in Urban Environment Using Extended Kalman Filter. *IOP Conf. Ser. Earth Environ. Sci.* **2023**, *1127*, 012006. [CrossRef]
13. Sabatelli, S.; Galgani, M.; Fanucci, L.; Rocchi, A. A Double-Stage Kalman Filter for Orientation Tracking With an Integrated Processor in 9-D IMU. *IEEE Trans. Instrum. Meas.* **2012**, *62*, 590–598. [CrossRef]
14. Shao, Y.; Xu, F.; Chen, J.; Lu, J.; Du, S. Engineering surface topography analysis using an extended discrete modal decomposition. *J. Manuf. Process.* **2023**, *90*, 367–390. [CrossRef]
15. Zhao, C.; Lui, C.F.; Du, S.; Wang, D.; Shao, Y. An Earth Mover's Distance based Multivariate Generalized Likelihood Ratio Control Chart for Effective Monitoring of 3D Point Cloud Surface. *Comput. Ind. Eng. Comput. Ind. Eng.* **2023**, *175*, 108911. [CrossRef]
16. Zhao, C.; Du, S.; Lv, J.; Deng, Y.; Li, G. A novel parallel classification network for classifying three-dimensional surface with point cloud data. *J. Intell. Manuf.* **2023**, *34*, 515–527. [CrossRef]
17. Atia, M.M.; Hilal, A.R.; Stellings, C.; Hartwell, E.; Toonstra, J.; Miners, W.B.; Basir, O.A. A Low-Cost Lane-Determination System Using GNSS/IMU Fusion and HMM-Based Multistage Map Matching. *IEEE Trans. Intell. Transp. Syst.* **2017**, *18*, 3027–3037. [CrossRef]
18. Julier, S.J.; Uhlmann, J.K. New Extension of the Kalman Filter to Nonlinear Systems. *Signal Processing Sens. Fusion Target Recognit. VI. SPIE* **1997**, *3068*, 182–193.
19. Arasaratnam, I.; Haykin, S. A numerical-integration perspective on Gaussian filters. *IEEE Trans Autom. Control.* **2009**, *54*, 1254–1269. [CrossRef]
20. Jeon, H.; Min, J.; Bang, H.; Youn, W. Quaternion-Based Iterative Extended Kalman Filter for Sensor Fusion of Vision Sensor and IMU in 6-DOF Displacement Monitoring. *IEEE Sens. J.* **2022**, *22*, 23188–23199. [CrossRef]
21. Lee, J.-C.; Chen, C.-C.; Shen, C.-T.; Lai, Y.-C. Landmark-Based Scale Estimation and Correction of Visual Inertial Odometry for VTOL UAVs in a GPS-Denied Environment. *Sensors* **2022**, *22*, 9654. [CrossRef]
22. Kourabbaslou, S.S.; Zhang, A.; Atia, M.M. A Novel Design Framework for Tightly Coupled IMU/GNSS Sensor Fusion Using Inverse-Kinematics, Symbolic Engines, and Genetic Algorithms. *IEEE Sens. J.* **2019**, *19*, 11424–11436. [CrossRef]
23. Yi, S.; Zorzi, M. Robust Kalman Filtering Under Model Uncertainty: The Case of Degenerate Densities. *IEEE Trans. Autom. Control* **2021**, *67*, 3458–3471. [CrossRef]
24. Yi, S.; Zorzi, M. Robust fixed-lag smoothing under model perturbations. *J. Frankl. Inst.* **2023**, *360*, 458–483. [CrossRef]
25. Gao, L.; Xiong, L.; Xia, X.; Lu, Y.; Yu, Z.; Khajepour, A. Improved Vehicle Localization Using On-Board Sensors and Vehicle Lateral Velocity. *IEEE Sens. J.* **2022**, *22*, 6818–6831. [CrossRef]
26. Xia, X.; Hashemi, E.; Xiong, L.; Khajepour, A. Autonomous Vehicle Kinematics and Dynamics Synthesis for Sideslip Angle Estimation Based on Consensus Kalman Filter. *IEEE Trans. Control. Syst. Technol.* **2022**, *31*, 179–192. [CrossRef]
27. Xia, X.; Xiong, L.; Huang, Y.; Lu, Y.; Gao, L.; Xu, N.; Yu, Z. Estimation on IMU yaw misalignment by fusing information of automotive onboard sensors. *Mech. Syst. Signal Process.* **2022**, *162*, 107993. [CrossRef]
28. Madyastha, V.K.; Ravindra, V.C.; Mallikarjunan, S.; Goyal, A. Extended Kalman filter vs. error state Kalman filter for aircraft attitude estimation. In Proceedings of the AIAA Guidance, Navigation, and Control Conference and Exhibit, Portland, OR, USA, 8–11 August 2011.

29. He, J.; Sun, C.; Zhang, B.; Wang, P. Adaptive Error-State Kalman Filter for Attitude Determination on a Moving Platform. *IEEE Trans. Instrum. Meas.* **2021**, *70*, 9513110. [CrossRef]
30. Critchley-Marrows, J.J.R.; Wu, X.; Cairns, I.H. Treatment of Extended Kalman Filter Implementations for the Gyroless Star Tracker. *Sensors* **2022**, *22*, 9002. [CrossRef]
31. Xia, M.; Sun, P.; Guan, L.; Zhang, Z. Research on Algorithm of Airborne Dual-Antenna GNSS/MINS Integrated Navigation System. *Sensors* **2023**, *23*, 1691. [CrossRef]
32. Ke, Y.; Lv, Z.; Zhang, C.; Deng, X.; Zhou, W.; Song, D. Tightly Coupled GNSS/INS Integration Spoofing Detection Algorithm Based on Innovation Rate Optimization and Robust Estimation. *IEEE Access* **2022**, *10*, 72444–72457. [CrossRef]
33. Chen, Q.; Zhang, Q.; Niu, X. Estimate the Pitch and Heading Mounting Angles of the IMU for Land Vehicular GNSS/INS Integrated System. *IEEE Trans. Intell. Transp. Syst.* **2020**, *22*, 6503–6515. [CrossRef]
34. 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]
35. 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]
36. Yang, C.; Cheng, Z.; Jia, X.; Zhang, L.; Li, L.; Zhao, D. A Novel Deep Learning Approach to 5G CSI/Geomagnetism/VIO Fused Indoor Localization. *Sensors* **2023**, *23*, 1311. [CrossRef]
37. Akeila, E.; Salcic, Z.; Swain, A. Reducing Low-Cost INS Error Accumulation in Distance Estimation Using Self-Resetting. *IEEE Trans. Instrum. Meas.* **2013**, *63*, 177–184. [CrossRef]
38. Chen, C.-L.; He, R.; Peng, C.-C. Development of an Online Adaptive Parameter Tuning vSLAM Algorithm for UAVs in GPS-Denied Environments. *Sensors* **2022**, *22*, 8067. [CrossRef]
39. López, E.; García, S.; Barea, R.; Bergasa, L.M.; Molinos, E.J.; Arroyo, R.; Romera, E.; Pardo, S. A Multi-Sensorial Simultaneous Localization and Mapping (SLAM) System for Low-Cost Micro Aerial Vehicles in GPS-Denied Environments. *Sensors* **2017**, *17*, 802. [CrossRef]
40. Elamin, A.; Abdelaziz, N.; El-Rabbany, A. A GNSS/INS/LiDAR Integration Scheme for UAV-Based Navigation in GNSS-Challenging Environments. *Sensors* **2022**, *22*, 9908. [CrossRef]
41. Revert Calabuig, N.; Laarossi, I.; Álvarez González, A.; Pérez Nuñez, A.; González Pérez, L.; García-Minguillán, A.C. Development of a Low-Cost Smart Sensor GNSS System for Real-Time Localization and Orientation for Floating Offshore Wind Platform. *Sensors* **2023**, *23*, 925. [CrossRef]
42. Liu, Y.; Liu, F.; Gao, Y.; Zhao, L. Implementation and Analysis of Tightly Coupled Global Navigation Satellite System Precise Point Localization/Inertial Navigation System (GNSS PPP/INS) with Insufficient Satellites for Land Vehicle Navigation. *Sensors* **2018**, *18*, 4305. [CrossRef]
43. Li, Z.; Zhang, Y.; Shi, Y.; Yuan, S.; Zhu, S. Performance Enhancement of INS and UWB Fusion Localization Method Based on Two-Level Error Model. *Sensors* **2023**, *23*, 557. [CrossRef] [PubMed]

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.