

Article

Adaptive UAV Navigation Method Based on AHRS

Yin Lu ^{1,2,*} , Zhipeng Li ³, Jun Xiong ¹ and Ke Lv ³

¹ School of Internet of Things, Nanjing University of Posts and Telecommunications, Nanjing 210003, China; xiongjun@njupt.edu.cn

² Key Lab of Broadband Wireless Communication and Sensor Network Technology, Nanjing University of Posts and Telecommunications, Ministry of Education, Nanjing 210003, China

³ School of Communication and Information Engineering, Nanjing University of Posts and Telecommunications, Nanjing 210003, China; 1221014505@njupt.edu.cn (Z.L.); lwtg2009@163.com (K.L.)

* Correspondence: luyin@njupt.edu.cn; Tel.: +86-25-8548-3090

Abstract: To address the inaccuracy of the Constant Acceleration/Constant Velocity (CA/CV) model as the state equation in describing the relative motion state in UAV relative navigation, an adaptive UAV relative navigation method is proposed, which is based on the UAV attitude information provided by Attitude and Heading Reference System (AHRS). The proposed method utilizes the AHRS output attitude parameters as the benchmark for dead reckoning and derives a relative navigation state equation with attitude error as process noise. By integrating the extended Kalman filter output for relative state estimation and employing an adaptive decision rule designed using the innovation of the filter update phase, the proposed method recalculates motion states deviating from the actual motion using the Tasmanian Devil Optimization (TDO) algorithm. The simulation results show that, compared with the CA/CV model, the proposed method reduces the relative position errors by 12%, 23%, and 32% in the X, Y, and Z directions, respectively, and that it reduces the relative velocity errors by 350%, 330%, and 300%, respectively. There is a significant improvement in the relative navigation accuracy.

Keywords: Unmanned Aerial Vehicle relative navigation; Attitude and heading reference system; Kalman filter; Tasmanian devil optimization



Citation: Lu, Y.; Li, Z.; Xiong, J.; Lv, K. Adaptive UAV Navigation Method Based on AHRS. *Sensors* **2024**, *24*, 2518. <https://doi.org/10.3390/s24082518>

Academic Editors: Carlos Tavares Calafate, Jesús Ureña and Boyang Li

Received: 30 January 2024

Revised: 1 April 2024

Accepted: 5 April 2024

Published: 14 April 2024



Copyright: © 2024 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

Unmanned Aerial Vehicle (UAV) relative navigation is achieved by measuring elements to obtain data such as orientation and distance of UAV formations, determining the relative attitude, position, and velocity of UAVs within the formation. This technology is widely applied in various scenarios, including formation flying, aerial refueling, and autonomous rendezvous and docking of spacecraft. Currently, the more common method for relative navigation involves the combination of the Global Navigation Satellite System (GNSS) and Inertial Navigation System (INS), which provides high precision in relative positioning by obtaining the UAV's location information. However, with the continuous advancement in UAV technology, higher demands are placed on the precision of relative navigation.

In terms of the relative motion model, many studies have proposed different designs for relative motion state equations. Traditional UAV relative motion estimation usually adopts the CA/CV model [1], which neglects the issue of relative attitudes between UAVs, further affecting the calculation precision of other relative navigation states like position and velocity. Adam and John proposed relative attitude motion equations and relative barycenter motion equations based on attitude quaternions, relative velocity, and relative position as state variables to describe UAV relative motion. However, the complex parameter calculations of this method have some impact on algorithm efficiency [2]. Cheng, J et al. considered the characteristics of close-range accompanying flight of UAV swarms. By

omitting the elevation dimension of the navigation coordinate system and assuming that all UAVs are in the same local navigation coordinate system, the H coordinate system is proposed [3]. Based on this system, UAV relative motion equations are derived, simplifying the traditional relative barycenter equation's parameter calculations and improving algorithm efficiency. However, as it replaces the coordinate system transformation's three rotations with a single rotation, further improvements in system accuracy are needed. Aiming at the problem of UAV landing on the ship's body, Zewei Zheng, Zhenghao Jin, et al. proposed a relative model, which was based on the six-degrees-of-freedom (6-DOF) UAV and carrier model to establish a coupled six-degrees-of-freedom nonlinear relative motion model. However, due to the slow change in the motion of the ship's body relative to the UAV, the study neglected the effect of yaw angle in the relative motion [4].

As UAV clusters are widely used, UAV cooperative navigation methods are required to be able to further improve their solving performance. Reference [1] proposed a relative navigation method based on ultra-wideband/relative difference, utilizing measurements from multiple sensors such as GNSS/INS tight combination positioning data, ultra-wideband (UWB) ranging, and INS outputs. Subsequently, extended Kalman filtering is used for data fusion, reducing the error in relative navigation state quantities. However, due to the high maneuverability of UAVs, the traditional Kalman filtering method cannot adapt to the sudden changes in the motion state of UAVs. Under extreme conditions, the method even causes its output to diverge [5–8]. To solve this problem, there are two main methods: (1) The system state equations are derived by improving the accuracy and adaptability of the motion model to describe the motion state. Alternatively, the accuracy of the observation equation modeling is improved by comprehensively considering the observation noise of the sensors. (2) Redesigning the filtering method to enhance the filtering method's ability to deal with the problem when it occurs in the system, and to improve the overall performance of the system. The previous introduction of relative states of motion belongs to Method 1. Reference [9] devises adaptive filtering centered on innovation during the filtering process. The preferred method within this category is the strong tracking filter (STF) proposed in Reference [10]. The STF enhances the system's adaptability by computing the asymptotic cancellation factor. This factor ensures that the sequence of residuals becomes orthogonal, thereby compensating for significant prediction errors [11–13]. The theoretical derivation of constructing STF is more complicated, and there are problems such as large computational volume for solving the asymptotic cancellation factor and arbitrary insertion position [14]. Reference [15] proposed a novel adaptive Sage-Husa, which solves divergence caused by wrong selection values and has strong robustness. Reference [16] studies a joint filtering algorithm based on multi-source sensors. The algorithmic framework consists of a main filter and multiple sub-filters. These sub-filters estimate and output the main filter interdependently to obtain a globally optimal solution. However, all these methods use the extended Kalman filter (EKF). This filter is linearized using a Taylor series expansion. It omits the higher-order terms of the nonlinearization and is less stable in the case of strongly nonlinear equations of state. Arasaratnam proposed the Cubature Kalman Filter (CKF) algorithm, which uses the spherical radial volume criterion to approximate the optimal estimation of the state's posterior distribution [17,18]. Based on CKF, Reference [19] proposed an efficient adaptive filtering algorithm. The algorithm calculates innovative values and sets decision rules by observing equations and sensor observations. Compensation is performed when the state equations do not accurately describe the system, thus improving navigation accuracy. However, the setting of compensation coefficients in this method depends on empirical values and cannot be applied to different navigation scenarios.

Despite the simplicity of Kalman filter-based collaborative navigation methods, there are two significant drawbacks of such methods: (1) they do not apply to large-scale clusters, and the excessive number of collaborative nodes will increase the computation amount of matrix multiplication and inverse computation without any limitation; and (2) There is a certain performance loss, and they cannot make full use of geometric constraints between nodes to further improve the estimation performance. Collaborative navigation

techniques based on optimization algorithms can solve the above problems. Combined with GNSS/UWB systems through optimization algorithms, Günther Retscher et al. propose the fusion of GNSS pseudoranges with UWB ranges based on clustering and Weighted Least Squares (WLS) [20]. Recently, nature-inspired methods have become increasingly popular in UAV navigation because of their ability to efficiently deal with dynamic constraints due to their effectiveness in handling UAVs and searching for dynamic constraints of UAVs. These include Cuckoo Search, Genetic Algorithm (GA), Differential Evolutionary Algorithm (DE), Artificial Bee Colony Algorithm (ABC), Ant Colony Optimization Algorithm (ACO), Ant Colony Optimization (ACO), and Particle Swarm Optimization (PSO) [21–27].

In response to the challenges and research deficiencies mentioned above, this paper considers the attitude changes during UAV relative motion. To enhance the precision and stability of the UAV relative navigation system, an adaptive UAV navigation method based on AHRS is proposed. The main research work and contributions of this paper are as follows:

1. An AHRS-based novel UAV relative motion model is proposed. The model describes the relative motion using the specific force equation with the UAV attitude output from the AHRS. The model fully takes into account the effect of relative attitude on relative velocity. Therefore, compared with the CA/CV model, this model can improve the overall positioning accuracy of the relative positioning method. In addition, the model uses AHRS as the solution system for the UAV attitude. While ensuring the accuracy of parameter precision, it can realize the correction of attitude error. Compared with the existing relative model, this innovative model reduces the computational complexity of the model for modeling the relative error, while improving the accuracy.
2. A TDO-based adaptive filtering method is proposed. The method utilizes the innovation in the EKF process to design the adaptive judgment rule. This method solves the problem of increased error in Kalman filter state estimation due to abrupt changes during relative motion. Meanwhile, the objective function of the optimization algorithm is constructed using multi-sensor conditions such as AHRS, UWB, and GNSS. A new TDO algorithm is also used to correct the estimates that deviate from the true motion state during the filtering process. Finally, the optimization algorithm is combined with the filtering algorithm to form the TDO adaptive filtering algorithm. The algorithm has better convergence and accuracy than traditional optimization algorithms for high-dimensional optimization problems such as UAV positioning.
3. The performance of the TDO algorithm is compared with other traditional optimization algorithms by performing simulation verification. According to the results, the TDO algorithm has good stability and accuracy when dealing with the problem of UAV scenarios. Meanwhile, by comparing the method proposed in this paper with the traditional relative localization methods, it can be obtained that the new relative localization model can better deal with the relative motion state. Moreover, the TDO adaptive filtering algorithm can improve the accuracy of the method by correcting the deviation from the real motion trajectory.

The rest of this paper is organized as follows. Section 2 describes the system model. Section 3 provides a detailed description of the relative positioning method. Sections 4 and 5 present the simulation results and conclude the paper.

2. System Model

The traditional CA/CV model does not sufficiently consider the relative attitude changes between UAVs, resulting in the system model's imprecise description of the relative motion state. This inadequacy leads to increased solution errors or even divergence in practical applications. Considering the need to use UAV attitude parameters to construct the relative navigation state equation, the AHRS is introduced. The AHRS employs a triaxial magnetometer, a triaxial gyroscope, and a triaxial accelerometer. It describes the attitude of the moving body using quaternions, measures angular velocity through the

gyroscope, and integrates accelerometer and magnetometer data using the Kalman filter algorithm to correct the attitude quaternions, thereby enhancing the accuracy of attitude determination [28]. The structure of the AHRS attitude determination system is shown in Figure 1. Here, the quaternion q and gyroscope errors are taken as state variables, and the magnetometer output m^b and accelerometer output a^b serve as observational values in the Kalman filter measurement update. Finally, after the measurement update, gyroscope error e is used for real-time correction of the gyroscope, and the relationship between quaternion q and the direction cosine matrix is utilized to obtain the UAV's attitude information.

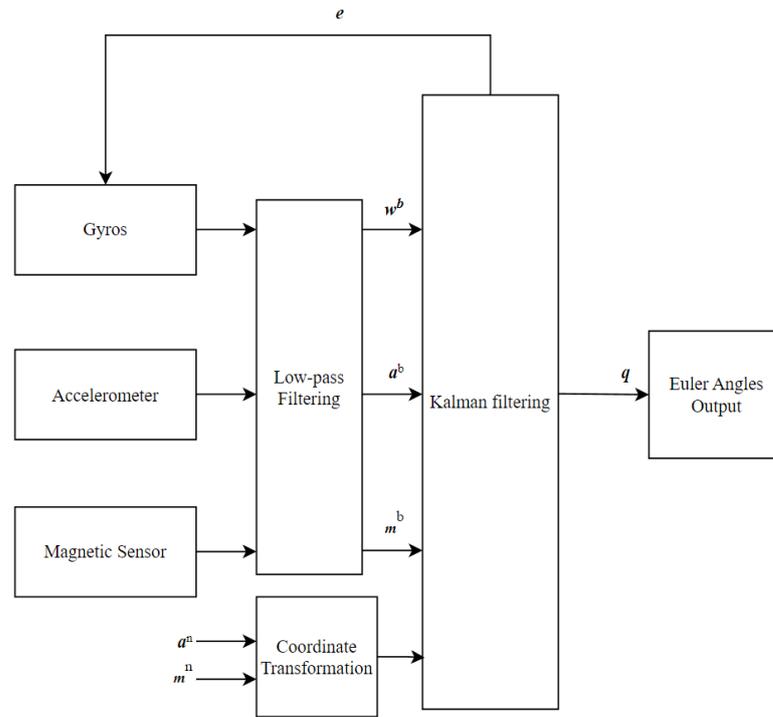


Figure 1. Attitude computing system architecture for AHRS.

The UAV utilizes the output from the AHRS system as the real-time body attitude angles and employs these angles to obtain the coordinate transformation matrix. The transformation relationship is as follows:

$$C_n^b = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \varphi & \sin \varphi \\ 0 & -\sin \varphi & \cos \varphi \end{bmatrix} \begin{bmatrix} \cos \theta & 0 & -\sin \theta \\ 0 & 1 & 0 \\ \sin \theta & 0 & \cos \theta \end{bmatrix} \begin{bmatrix} \cos \psi & \sin \psi & 0 \\ -\sin \psi & \cos \psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (1)$$

where φ is the roll angle, θ is the pitch angle, and ψ is the yaw angle. This transformation achieves the conversion between the body coordinate system and the navigation coordinate system.

The specific force f_b is the theoretical output of the accelerometer. Its output is affected by gravity. Based on the definition of specific force and neglecting minor errors, the output of the UAV accelerometer is expressed as [29,30]:

$$f_b = C_n^b(a_n + g_n) + \delta f \quad (2)$$

where a_n is the acceleration in the navigation coordinate system, g_n is the projection of the gravitational vector in the navigation coordinate system, and δf is the acceleration error. Because of the derivative relationship between acceleration and velocity, the UAV velocity differential equation is obtained according to Equation (2). \dot{v}_n is expressed as:

$$\dot{v}_n = a_n = C_b^n f_b - g_n + C_b^n \delta f_e \quad (3)$$

Considering that the AHRS system output attitude angle error can affect the whole system, the transfer of the b coordinate system to the N-coordinate system is viewed as two coordinate system rotations. $C_b^{b^e}$ consists of the attitude error of the AHRS.

$$C_b^{b^e} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \delta\varphi & \sin \delta\varphi \\ 0 & -\sin \delta\varphi & \cos \delta\varphi \end{bmatrix} C_b^n = C_{b^e}^n C_b^{b^e} \begin{bmatrix} \cos \delta\theta & 0 & -\sin \delta\theta \\ 0 & 1 & 0 \\ \sin \delta\theta & 0 & \cos \delta\theta \end{bmatrix} \begin{bmatrix} \cos \delta\psi & \sin \delta\psi & 0 \\ -\sin \delta\psi & \cos \delta\psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (4)$$

considering the small attitude error due to AHRS. Therefore, the limit is taken for Equation (4). C_b^n can be expressed as follows:

$$C_b^n = C_{b^e}^n \begin{bmatrix} 1 & -\delta\psi & \delta\theta \\ \delta\psi & 1 & -\delta\varphi \\ -\delta\theta & \delta\varphi & 1 \end{bmatrix} \quad (5)$$

Let $C_{b^e}^n$ be written as \hat{C}_b^n , and substitute Equation (4) into Equation (5).

$$\begin{aligned} \dot{v}_n &= \hat{C}_b^n \begin{bmatrix} 1 & -\delta\psi & \delta\theta \\ \delta\psi & 1 & -\delta\varphi \\ -\delta\theta & \delta\varphi & 1 \end{bmatrix} f_b - g_n + C_b^n \delta f \\ &= \hat{C}_b^n f_b + \hat{C}_b^n \begin{bmatrix} 0 & -\delta\psi & \delta\theta \\ \delta\psi & 0 & -\delta\varphi \\ -\delta\theta & \delta\varphi & 0 \end{bmatrix} f_b - g_n + C_b^n \delta f \\ &= \hat{C}_b^n f_b + \hat{C}_b^n [-f_b, x] \begin{bmatrix} \delta\varphi \\ \delta\theta \\ \delta\psi \end{bmatrix} - g_n + C_b^n \delta f \end{aligned} \quad (6)$$

Equation (6) finally establishes the UAV's velocity differential equation in the navigation coordinate system based on AHRS.

$$\dot{v}_n = \hat{C}_b^n [-f_b, x] \begin{bmatrix} \delta\varphi \\ \delta\theta \\ \delta\psi \end{bmatrix} + \hat{C}_b^n f_b - g_n + C_b^n \delta f \quad (7)$$

Here \hat{C}_b^n is obtained from the output of the AHRS system, and $\delta\varphi$, $\delta\psi$, and $\delta\theta$ represent the attitude measurement errors of the AHRS.

The differential equation for the relative velocity of the UAV can be obtained by the differential equation for the velocity of a single UAV. Assuming the differential equation for the velocity of a single lead UAV is equal to:

$$\dot{v}_l = \hat{C}_{bl}^l [-f_{bl}, x] \begin{bmatrix} \delta\varphi_l \\ \delta\theta_l \\ \delta\psi_l \end{bmatrix} + \hat{C}_{bl}^l f_{bl} - g_n + C_b^l \delta f_l \quad (8)$$

Similarly, the differential equation for the velocity of the follow UAV is equal to:

$$\dot{v}_f = \hat{C}_{bf}^f [-f_{bf}, x] \begin{bmatrix} \delta\varphi_f \\ \delta\theta_f \\ \delta\psi_f \end{bmatrix} + \hat{C}_{bf}^f f_{bf} - g_n + C_b^f \delta f_f \quad (9)$$

In the study of UAV formation flight, the dynamic changes in the relative positions of the two aircraft lead to corresponding changes in their respective navigation coordinate systems. Simply using the velocity differential equations of the leader and the wingman UAVs would overlook the differences between the two navigation coordinate systems, potentially introducing computational errors. To accurately describe the acceleration

states in the relative velocity differential equation, this research constructs a coordinate transformation matrix from the navigation system to the Earth-Centered, Earth-Fixed (ECEF) system based on the position information provided by the single aircraft navigation system. Through this coordinate transformation matrix, the accelerations of the leader and the wingman UAVs, denoted as \dot{v}_l and \dot{v}_f , are converted to the ECEF system. Finally, the relative velocity differential equation is obtained through calculation.

$$\begin{aligned} \dot{v}_r &= C_f^e \dot{v}_f - C_l^e \dot{v}_l \\ &= C_{bf}^e f_{bf} + C_{bf}^e [-f_{bf} x] \begin{bmatrix} \delta\varphi_f \\ \delta\theta_f \\ \delta\psi_f \end{bmatrix} - C_f^e g_n + C_{bf}^e \delta f_f - C_{bl}^e f_{bl} - C_{bl}^e [-f_{bl} x] \begin{bmatrix} \delta\varphi_l \\ \delta\theta_l \\ \delta\psi_l \end{bmatrix} + C_l^e g_n - C_{bl}^e \delta f_l \end{aligned} \quad (10)$$

Continuing from the previous discussion, since $C_l^e g_n - C_f^e g_n$ is a relatively small value, Equation (10) can be simplified as follows:

$$\dot{v}_r = C_{bf}^e f_{bf} + C_{bf}^e [-f_{bf} x] \begin{bmatrix} \delta\varphi_f \\ \delta\theta_f \\ \delta\psi_f \end{bmatrix} + C_{bf}^e \delta f_f - C_{bl}^e f_{bl} - C_{bl}^e [-f_{bl} x] \begin{bmatrix} \delta\varphi_l \\ \delta\theta_l \\ \delta\psi_l \end{bmatrix} - C_{bl}^e \delta f_l \quad (11)$$

where C_{bf}^e and C_{bl}^e are obtained from the longitude and latitude information provided by the individual navigation systems of the drones. f_{bf} and f_{bl} represent the acceleration outputs of each drone, while $\delta\psi_f$, $\delta\theta_f$, $\delta\varphi_f$, $\delta\psi_l$, $\delta\theta_l$, and $\delta\varphi_l$ denote the attitude output errors of each AHRS system. Additionally, δf_f and δf_l are the output errors of each accelerometer.

In the modeling of the differential equation for relative position, considering the complexity of the system and the brief duration of the filtering cycle, this equation is simplified to a constant velocity motion model. Additionally, addressing the issue of random offset in accelerometers, this study describes it as a first-order Markov process. Based on this assumption, and considering that the error models for the three axes have similar characteristics, it can be expressed as:

$$\begin{bmatrix} \dot{\delta f}_x \\ \dot{\delta f}_y \\ \dot{\delta f}_z \end{bmatrix} = \begin{bmatrix} -1/T_{ax} & & \\ & -1/T_{ay} & \\ & & -1/T_{az} \end{bmatrix} \begin{bmatrix} \delta f_x \\ \delta f_y \\ \delta f_z \end{bmatrix} + w(t) \quad (12)$$

where T_{ax} , T_{ay} , and T_{az} are the correlation times, and $w(t)$ is a Gaussian white noise with a mean of zero.

3. Methods

This paper introduces an adaptive filtering method. It is based on system innovation and formulates the adaptive judgment rule. It uses measurement equations and values to design the optimal objective function. Additionally, it incorporates the TDO algorithm into the filtering method. The method can improve the system's stability and accuracy, and at the same time, it is easy to realize.

3.1. Design of Adaptive UAV Navigation Method Based on AHRS

This paper is based on the system model proposed in Section 2. At the same time, the effect of partial sensor measurement noise is considered under the UAV equipped with multiple sensors. Combined with the EKF algorithm, an AHRS-based UAV adaptive navigation method is proposed. The example is the UAV l , and the design of the method is shown in Figure 2.

$$\mathbf{X}_k^e = [\Phi_E, \Phi_N, \Phi_U, \delta v_E, \delta v_N, \delta v_U, \delta L, \delta \lambda, \delta h, \varepsilon_{cgx}, \varepsilon_{cgy}, \varepsilon_{cgz}, \varepsilon_{rgx}, \varepsilon_{rgy}, \varepsilon_{rgz}, \varepsilon_{rax}, \varepsilon_{ray}, \varepsilon_{raz}, \delta s_u, \delta s_{ru}]^T \quad (14)$$

This \mathbf{X}_k^e consists of the error; the first 6 dimensions are the local filter output parameter error, which includes the 3-dimensional inertial navigation platform error angle, the 3-dimensional velocity error, and the 3-dimensional position error. The 10th to 15th state variables, respectively, are gyroscope three-axis constant drift error, and three-axis random drift error. The 16th to 18th dimensions are accelerometer three-axis random drift errors. The last two dimensions are the distance error due to the equivalent clock error, and the distance error due to the equivalent clock frequency error.

Based on the error analysis of the Strapdown inertial navigation system, several error equations can be obtained [31].

The platform error angle equation is expressed as:

$$\dot{\Phi} = \delta \bar{\omega}_{in}^n + \Phi \times \bar{\omega}_{in}^n - \varepsilon^n \quad (15)$$

where, ε^n denotes the gyroscope instrument error, and $\delta \bar{\omega}_{in}^n$ and $\Phi \times \bar{\omega}_{in}^n$ are the gyroscope drift caused by the rotation motion of the Earth and the UAV's motion.

The velocity error equation is given as:

$$\delta \dot{v} = \Phi \times f^n + \nabla^n - (2\delta \omega_{ie}^n + \delta \omega_{en}^n) \times v^n - (2\omega_{ie}^n + \omega_{en}^n) \times \delta v + \delta g \quad (16)$$

where δg is the accelerometer error, ω_{ie}^n is the Earth's rotational velocity, and ω_{en}^n is the projection of the UAV N-coordinate system onto the N-coordinate system relative to the E-coordinate system.

The position error equation is represented as:

$$\begin{cases} \delta \dot{L} = \frac{v_U}{(R_M+h)^2} \delta h + \frac{1}{R_M+h} \delta v_N \\ \delta \dot{\lambda} = \frac{v_E \tan L}{(R_N+h) \cos L} \delta L - \frac{v_E}{(R_N+h)^2 \cos L} \delta h + \frac{1}{(R_N+h) \cos L} \delta v_E \\ \delta \dot{h} = \delta v_U \end{cases} \quad (17)$$

where R is the radius of the Earth, about 6317 KM, $R_M = (1 - 2f + 3f \sin 2L)R_e$, $R_N = (1 + f \sin 2L)R_e$, and $f = 1/298.257$.

Accelerometers and gyroscopes typically include mounting errors, scale factor errors, and instrumentation random errors, which are usually colored noise. In this section, these physical errors are uniformly considered as comprehensive random errors.

Let the gyro drift error consist of scale factor error, zero-bias error, random constant, and white noise error.

$$\varepsilon = \varepsilon_{bias} + \varepsilon_g + w_g \quad (18)$$

Assuming that the zero-bias error and the scale factor error are modeled as a first-order Markov process, the following is obtained:

$$\begin{cases} \dot{\varepsilon}_{bias} = 0 \\ \dot{\varepsilon}_g = -\frac{1}{T_g} \varepsilon_g + w_r \end{cases} \quad (19)$$

where T_g is the gyroscope correlation time.

Let the accelerometer error model consist only of a first-order Markov process and assume that the error model is the same for the 3 axial directions of the accelerometer. The same reasoning is obtained:

$$\dot{\varepsilon}_a = -\frac{1}{T_a} \varepsilon_a + w_a \quad (20)$$

where T_a is the accelerometer correlation time.

The GNSS receiver correlation error is modeled as:

$$\begin{cases} \delta\dot{s}_u = \delta s_{ru} + w_{tu} \\ \delta\dot{s}_{ru} = -\beta_{tru}\delta\dot{s}_{ru} + w_{tru} \end{cases} \quad (21)$$

where $\delta\dot{s}_u$ and $\delta\dot{s}_{ru}$ are the distance error due to the receiver equivalent clock error and the distance error due to the equivalent clock frequency error, respectively.

By associating Equations (15)–(21), the state transfer matrix $F_{k/k-1}^e$ and the noise coefficient matrix G_{k-1}^e can be obtained.

In designing the measurement equations for combining the local filters, this section adopts a tight combination approach: i.e., instead of directly using the GNSS-solved positioning information, mathematical modeling is performed based on the pseudorange observed by the GNSS receiver.

Using the pseudorange value ρ_{li} obtained from the position calculation and the receiver measurement pseudorange value ρ_{gi} , the observation equation is obtained by difference processing [32]:

$$\Delta\rho_i = \rho_{li} - \rho_{gi} = \frac{\partial\rho_{li}}{\partial x}\delta x + \frac{\partial\rho_{li}}{\partial y}\delta y + \frac{\partial\rho_{li}}{\partial z}\delta z + \delta t_u + v_{\rho_i} \quad (22)$$

where ρ_{li} is the calculated value of the pseudorange and ρ_{gi} is the measured value of the pseudorange. δx , δy , δz are the position errors in the E-coordinate system.

According to the local filter state variables, the position error is output in the N-coordinate system. Therefore, it is necessary to convert the position error in the E-coordinate system to the N-coordinate system. The conversion equation is expressed as:

$$\begin{cases} \delta x = \delta h \cos L \cos \lambda - (R_N + h) \sin L \cos \lambda \delta L - (R_N + h) \cos L \sin \lambda \delta \lambda \\ \delta y = \delta h \cos L \sin \lambda - (R_N + h) \sin L \sin \lambda \delta L + (R_N + h) \cos L \cos \lambda \delta \lambda \\ \delta z = \delta h \sin L + [R_N(1 - f)^2 + h] \cos L \delta L \end{cases} \quad (23)$$

Bringing Equation (23) into Equation (22), the equation for the pseudorange differential measurement is given as:

$$\mathbf{Z}_k^e = \delta\rho = \mathbf{H}_k^e \mathbf{X}_k^e + \mathbf{V}_k^e \quad (24)$$

where \mathbf{H}_k^e is the pseudorange observation equation and \mathbf{V}_k^e is the pseudorange measurement noise.

Ultimately, the error data for each sensor can be obtained by EKF. This error data is then used to calibrate the accelerometer and gyroscope outputs in the AHRS. Also, using the accelerometer and gyroscope data, the local filter can provide the UAV with absolute positioning information. This information is valuable in improving the overall accuracy of the relative positioning method.

3.1.2. Relative Navigation State Filtering

Relative navigation state filtering is designed based on the model presented in Section 2. Set the state variables and state equations for this filtering method according to Equations (11) and (12).

$$\begin{aligned} \mathbf{X}^r &= [v_{r_x}, v_{r_y}, v_{r_z}, r_x, r_y, r_z, \delta f_{l_x}, \delta f_{l_y}, \delta f_{l_z}, \delta f_{f_x}, \delta f_{f_y}, \delta f_{f_z}] \\ \mathbf{X}^r(t) &= \mathbf{F}_r(t)\mathbf{X}^r + \mathbf{B}(t)\mathbf{U}_r + \mathbf{G}(t)\mathbf{W}(t) \end{aligned} \quad (25)$$

where v_r is the relative velocity vector, r is the relative position vector, and $\delta f_l, \delta f_f$ are the ratio error vectors of UAV l and UAV f , respectively.

Next, the state transfer matrix can be represented as:

$$\mathbf{F}_r(t) = \begin{bmatrix} 0_{3 \times 3} & 0_{3 \times 3} & -\mathbf{C}_{bl}^e & \mathbf{C}_{bf}^e \\ 0_{3 \times 3} & \mathbf{I}_{3 \times 3} & 0_{3 \times 3} & 0_{3 \times 3} \\ 0_{3 \times 3} & 0_{3 \times 3} & \frac{-1}{T_a} & 0_{3 \times 3} \\ 0_{3 \times 3} & 0_{3 \times 3} & 0_{3 \times 3} & \frac{-1}{T_a} \end{bmatrix}_{12 \times 12} \quad (26)$$

The system control variables are assumed to be:

$$\mathbf{U}_r = \mathbf{C}_{bf}^e \mathbf{f}_{bf} - \mathbf{C}_{bl}^e \mathbf{f}_{bl} \quad (27)$$

The control variable coefficient matrix is expressed as:

$$\mathbf{B}(t) = \begin{bmatrix} \mathbf{I}_{3 \times 3} \\ \mathbf{0}_{9 \times 3} \end{bmatrix} \quad (28)$$

Set the system process noise $\mathbf{W}(t)$ to:

$$[\delta\varphi_l \quad \delta\theta_l \quad \delta\psi_l \quad \delta\varphi_f \quad \delta\theta_f \quad \delta\psi_f \quad W_{al} \quad W_{al} \quad W_{al} \quad W_{af} \quad W_{af} \quad W_{af}] \quad (29)$$

The noise factor matrix $\mathbf{G}(t)$ is denoted as:

$$\begin{bmatrix} \mathbf{C}_{bl}^e [\mathbf{f}_{bl} x] & 0_{3 \times 3} & 0_{3 \times 6} \\ 0_{3 \times 3} & \mathbf{C}_{bf}^e [-\mathbf{f}_{bf} x] & 0_{3 \times 6} \\ 0_{3 \times 6} & \mathbf{I}_{3 \times 3} & 0_{3 \times 3} \\ 0_{3 \times 6} & 0_{3 \times 3} & \mathbf{I}_{3 \times 3} \end{bmatrix} \quad (30)$$

The discretized equation of state is obtained by discretizing Equation (25).

$$\mathbf{X}_k^r = \mathbf{F}_{k|k-1} \mathbf{X}_{k-1}^r + \mathbf{B} \mathbf{U}_k + \mathbf{G}_{k-1} \mathbf{W}_{k-1} \quad (31)$$

UWB is not only used as a communication module to provide data interaction for the navigation system, but it also provides ranging functions. Its observation equation and observation matrix are denoted as [33]:

$$\begin{aligned} r_{UWB} &= \|\mathbf{r}\|_2 + n_{UWB} \\ \mathbf{H}_{UWB} &= \begin{bmatrix} 0_{1 \times 3} & \frac{\partial r_{UWB}}{\partial \mathbf{r}} & 0_{1 \times 6} \end{bmatrix} \end{aligned} \quad (32)$$

GNSS not only forms a filter with AHRS but also forms DGNSS by differential means, eliminating common errors present in the pseudorange. The observation matrix and observation equation for the double difference between pseudorange and pseudorange rates are expressed as [1]:

$$\begin{aligned} \Delta \nabla \rho_{lf}^{S_1 S_i} &= (\mathbf{e}^{S_i} - \mathbf{e}^{S_1}) \mathbf{r} + n_{lf}^{S_1 S_i}, \quad i = 2, \dots, 8 \\ \Delta \nabla \dot{\rho}_{lf}^{S_1 S_i} &= (\mathbf{e}^{S_i} - \mathbf{e}^{S_1}) \mathbf{v} + (\dot{\mathbf{e}}^{S_i} - \dot{\mathbf{e}}^{S_1}) \mathbf{r} + n_{lf_rate}^{S_1 S_i}, \quad i = 2, \dots, 8 \\ \mathbf{H}_{dd} &= \begin{bmatrix} e^{S_j} - e^{S_i} & \dot{e}^{S_j} - \dot{e}^{S_i} & 0_{1 \times 6} \\ \vdots & \vdots & \vdots \\ 0_{1 \times 3} & e^{S_j} - e^{S_i} & 0_{1 \times 6} \\ \vdots & \vdots & \vdots \end{bmatrix} \end{aligned} \quad (33)$$

where \mathbf{e}^{S_i} is the direction cosine vector from the midpoint to the satellite S_i between the two UAVs, $\dot{\mathbf{e}}^{S_i}$ is the change rate of the direction cosine vector from the midpoint to the satellite S_i between the two UAVs, \mathbf{r} is the relative position vector between the UAVs, \mathbf{v} is the relative velocity vector between the UAVs, $n_{lf}^{S_1 S_i}$ is the noise that exists in the

pseudorange bi-differential computation, $n_{lf_rate}^{S_1 S_i}$ is the noise that exists in the pseudorange bi-differential rate computation, and both kinds of noise belong to Gaussian white noise.

The AHRS/GNSS filtering method outputs the absolute navigation information of the UAV along with the closed-loop correction of the sensor. Based on this, the relative velocity and relative position of the UAV can be obtained. The measurement matrix of this value is expressed as:

$$\begin{aligned} \mathbf{r}_{abs} &= \mathbf{r} + \mathbf{n}_{abs} \\ \mathbf{v}_{abs} &= \mathbf{v} + \mathbf{n}_{abs_rate} \\ \mathbf{H}_{abs} &= \begin{bmatrix} I_3 & 0_{1 \times 3} & 0_{1 \times 3} & 0_{1 \times 3} \\ 0_{1 \times 3} & I_3 & 0_{1 \times 3} & 0_{1 \times 3} \end{bmatrix} \end{aligned} \quad (34)$$

Associating Equation (31) to Equation (33), the measurement model for relative navigation state filtering is obtained as:

$$\begin{aligned} \mathbf{Z}_k^r &= \mathbf{H}_k^r \mathbf{X}_k^{r-} + \mathbf{V}^r \\ \mathbf{H}_k^r &= \begin{bmatrix} \mathbf{H}_{dd} & \mathbf{H}_{abs} & \mathbf{H}_{UWB} \end{bmatrix}^T \end{aligned} \quad (35)$$

where \mathbf{V}^r is the measurement noise vector.

After outputting the relative navigation state \mathbf{X}_k^{r+} using the EKF, the value is subjected to an adaptive judgment. If it deviates from the true motion then it is corrected using the TDO algorithm.

3.2. TDO Algorithm Principle and Flow

TDO is an optimization algorithm inspired by nature [34]. This algorithm mimics the behavioral patterns of the pouched Tasmanian devil when searching for food and is used to solve complex optimization problems. There are two strategies within the TDO algorithm: attacking live prey or feeding on the carrion of the animal.

The TDO algorithm is a population-based stochastic algorithm whose search subject is the Tasmanian devil. So, in conjunction with the scenario of this paper, the initialization of the pouched Tasmanian devil population is obtained from the relative navigated system a posteriori state with the system state covariance matrix, and each member of the pouched Tasmanian devil represents a set of navigated state quantities. Each member of the TDO population is a searcher on the solution space, so that the TDO navigated state member population can be described as a matrix:

$$\mathbf{X} = \begin{bmatrix} \mathbf{X}_1 \\ \vdots \\ \mathbf{X}_i \\ \vdots \\ \mathbf{X}_N \end{bmatrix}_{N \times m} = \begin{bmatrix} x_{1,1} & \cdots & x_{1,j} & \cdots & x_{1,m} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ x_{i,1} & \cdots & x_{i,j} & \cdots & x_{i,m} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ x_{N,1} & \cdots & x_{N,j} & \cdots & x_{N,m} \end{bmatrix}_{N \times m} \quad (36)$$

where \mathbf{X} represents the whole navigation state population, \mathbf{X}_i represents the i_{th} population member, and $X_{i,j}$ represents the j_{th} state quantity in member i . Through this matrix, it can be seen that N is the number of navigation state population members, and m is the dimension of the system to solve the state quantity.

Bringing \mathbf{X} in Equation (36) into the objective function $F(x)$ yields a vector \mathbf{F} consisting of function values for each navigation state of the candidate.

$$\mathbf{F} = \begin{bmatrix} F_1 \\ \vdots \\ F_i \\ \vdots \\ F_N \end{bmatrix}_{N \times 1} = \begin{bmatrix} F(\mathbf{X}_1) \\ \vdots \\ F(\mathbf{X}_i) \\ \vdots \\ F(\mathbf{X}_N) \end{bmatrix}_{N \times 1} \quad (37)$$

where F is the vector composed of objective function values and F_i is the objective function value obtained for the i_{th} navigation state. Each objective function value demonstrates the quality of the corresponding candidate solution. The candidate solution that can compute the optimal value of the objective function is considered as the optimal member of the population. In each iteration, the optimal member of the population is updated based on the new value.

3.2.1. Strategy 1: Carrion Eater Strategy

The strategy of navigating the state as a “carrion eater” is mathematically modeled by Equations (38)–(40). In the TDO design, for each member it is assumed that the positions of other population members in the search space are “carrion” positions. Random selection of carrion locations is modeled in Equation (38).

$$C_i = X_k, i = 1, 2, \dots, N, k \in \{1, 2, \dots, N \mid k \neq i\} \quad (38)$$

where C_i represents the carrion location chosen by the i_{th} member of the population, and X_k is the location of all members except the i_{th} member.

The new position of the member in the search space is calculated based on the selection of carrion; in this strategy, the member will move towards the carrion if the carrion objective function value is better and vice versa, then the member will move away from the carrion, which is modeled by Equation (39).

$$x_{i,j}^{new,S1} = \begin{cases} x_{i,j} + r \cdot (c_{i,j} - I \cdot x_{i,j}), & F_{C_i} < F_i \\ x_{i,j} + r \cdot (x_{i,j} - c_{i,j}), & \text{else} \end{cases} \quad (39)$$

Equation (39) will calculate the new position of the member of the navigation state; if this objective function value is better in this new position then X_i will be updated, otherwise X_i will remain in its original position.

$$X_i = \begin{cases} X_i^{new,S1}, & F_i^{new,S1} < F_i \\ X_i, & \text{else} \end{cases} \quad (40)$$

where $X_i^{new,S1}$ is the new value calculated by the “carrion strategy”, $X_{i,j}^{new,S1}$ is the j_{th} element in $X_i^{new,S1}$, F_i is the objective function value, $F_i^{new,S1}$ is the objective function value of the carrion position, r is a random value between 0 and 1, and I is 1 or 2.

3.2.2. Strategy 2: Predator Strategy

The second strategy of the TDO algorithm is the predator strategy, where the member’s behavior during predation has two phases: in the first phase, the prey is selected by searching the area; then, the second phase will keep approaching the selected prey to attack. The selection of prey in the first phase is similar to the modeling of Carrion Eater Strategy

$$P_i = X_k, i = 1, 2, \dots, N, k \in \{1, 2, \dots, N \mid k \neq i\} \quad (41)$$

where P_i is the prey chosen by the i_{th} member, X_k is any other population member except i , and k is a random number between 1 and N .

After determining the position of the prey, the new position of the member is calculated, and if the objective function value of the chosen prey is better then the member moves towards it, and vice versa it moves away from that position

$$x_{i,j}^{new,S2} = \begin{cases} x_{i,j} + r \cdot (p_{i,j} - I \cdot x_{i,j}), & F_{P_i} < F_i \\ x_{i,j} + r \cdot (x_{i,j} - p_{i,j}), & \text{else} \end{cases} \quad (42)$$

$$X_i = \begin{cases} X_i^{\text{new},S2}, & F_i^{\text{new},S2} < F_i \\ X_i, & \text{else} \end{cases} \quad (43)$$

where $X_i^{\text{new},S2}$ is the new value calculated in the first phase of the predator strategy, $X_{i,j}^{\text{new},S2}$ is the j^{th} element of $X_i^{\text{new},S2}$, $F_i^{\text{new},S2}$ is the objective function value of the new value, F_i is the objective function value of the position of the prey, r is a random value between 0 and 1, and I is 1 or 2.

In order to simulate the process of members chasing the prey near the prey location, the prey location obtained in the first stage is used as the center of the chase and a search radius is set to represent the chase range, with the search radius computed as:

$$R_{TDO} = 0.01 \left(1 - \frac{t}{T} \right) \quad (44)$$

where t is the current iteration number and T is the maximum iteration number of the algorithm. The mathematical modeling of the members pursuing the prey in the region is as follows:

$$x_{i,j}^{\text{new}} = x_{i,j} + (2r - 1) \cdot R_{TDO} \cdot x_{i,j} \quad (45)$$

$$X_i = \begin{cases} X_i^{\text{new}}, & F_i^{\text{new}} < F_i \\ X_i, & \text{else} \end{cases} \quad (46)$$

where R_{TDO} is the radius of prey determined through Equation (44), X_i^{new} is the location of prey determined in the second phase of the predator strategy, F_i^{new} is the value of X_i^{new} after it is brought into the objective function, and $X_{i,j}^{\text{new}}$ is the j^{th} element of X_i^{new} .

The TDO algorithm flowchart is shown in Figure 4.

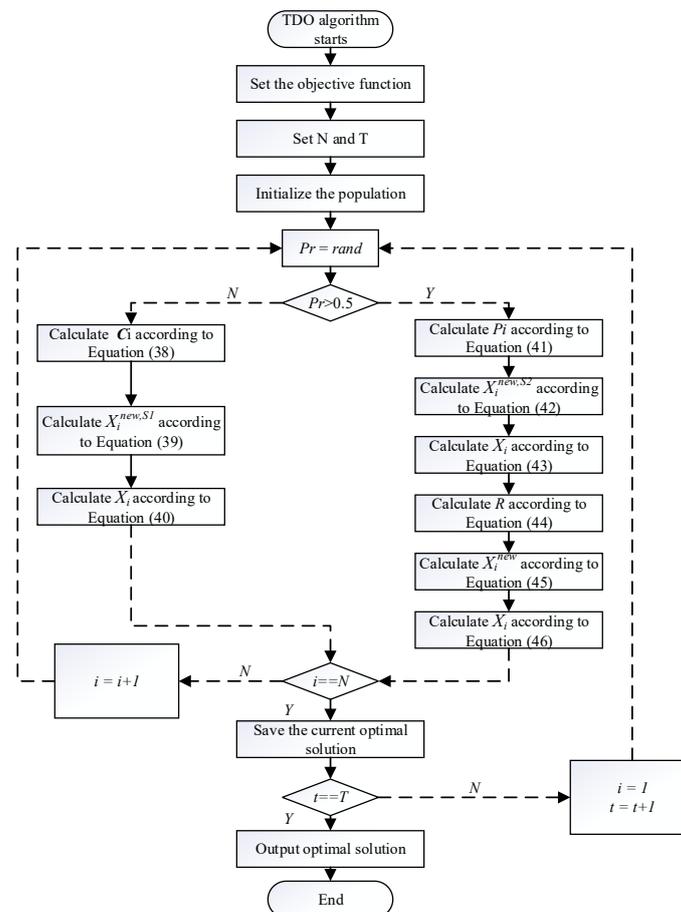


Figure 4. Flowchart of TDO algorithm.

3.3. Adaptive Judgment Rule Design

EKF is a commonly used algorithm for data fusion, and the principle of the algorithm will not be repeated in this paper; the algorithm calculates the optimal value of the state quantities need to use the innovation, which is denoted as:

$$\mathbf{e} = \mathbf{Z}_k^r - \mathbf{H}_k^r \mathbf{X}_k^{r-} \quad (47)$$

where \mathbf{Z}_k^r denotes the sensor observation value at the current moment, \mathbf{H}_k^r denotes the observation matrix, and \mathbf{X}_k^{r-} denotes the a priori state quantity calculated by the state equation. When the UAV motion model does not match the system state equation or the UAV has a sudden change in motion, the measurement prediction value obtained after the nonlinear measurement function introduced by the state equation will have a large deviation, which is propagated to the innovation \mathbf{e} by Equation (24).

Since the Kalman filtering algorithm utilizes the innovation for a posteriori estimation of state quantities, in order to make the filtering results more accurate, it is necessary to judge the innovation \mathbf{e} . The innovation \mathbf{e} can reflect the error between the current measurement prediction and the real measurement, and it can also show the “size” of the deviation of the current state value from the actual motion state. The innovation \mathbf{e} can reflect the error between the current measurement prediction and the real measurement, and it can also show the “size” of the current state value deviation from the actual motion state, so we can use the innovation to design the adaptive filter judgment threshold.

When designing the adaptive filtering judgment threshold, it is considered that the measurement noise of each sensor is Gaussian white noise. Therefore, when each sensor works normally, the measurement covariance matrix \mathbf{R} can be used to judge whether the innovation \mathbf{e} is out of the error range of the measurement. Assuming that the current measurement value \mathbf{Z}_i^r is the center of the judgment range, and because the components on the diagonal of the measurement covariance matrix represent the variance of the corresponding observation value, $\sqrt{\mathbf{R}_i}$ (the i component on the diagonal of the covariance matrix) is taken as the radius of the judgment range, and the error within the range belongs to the normal range. When the absolute value of the innovation exceeds this range, it is necessary to start correcting the state values by the algorithm.

Combined with the use of UAV multi-sensors, it can be seen that in the filtering algorithm process there will be more than one measurement value, and the measurement value accuracy of different sensors is not the same; in order to more reasonably use the measurement value to determine whether the state quantity needs to be corrected, it is necessary to carry out appropriate preprocessing for \mathbf{Z}_i^r and \mathbf{e} . Through the elements on the diagonal of \mathbf{R} we can know each kind of measured value error size, and through the preprocessing we hope that in the adaptive correction judgment, the sensor with high accuracy can play a greater role, so the proposed preprocessing formula is

$$a_j = \frac{\sum_{i=1}^k \sqrt{\mathbf{R}_i} - \sqrt{\mathbf{R}_j}}{\sum_{i=1}^k \sqrt{\mathbf{R}_i}} \quad (48)$$

$$\bar{e} = \sum_{j=1}^k a_j |e_j| \quad (49)$$

$$\lambda = \sum_{j=1}^k a_j \sqrt{\mathbf{R}_j} \quad (50)$$

where k is the number of UAV sensors, \bar{e} is the preprocessing value obtained from the current innovation, and λ is the adaptive judgment threshold. According to Equations (48)–(50),

it can be seen that when the accuracy of the sensors is higher the coefficient a_j in the calculation of the judgment threshold is larger.

In practical application, the following adaptive judgment rule is proposed: through \bar{e} with λ to recognize the abnormal situation of state quantity, when \bar{e} exceeds λ , the relative navigation state will be corrected by using the TDO algorithm, through which the process ensures that the algorithm can adapt to the relative motion state changes of the UAV, so as to improve its stability in the dynamic environment.

3.4. Objective Function Setting and Performance Analysis of TDO

With the development of multi-source sensors, the sensor measurement accuracy is getting higher and higher, so Equation (47) is set as the objective function. And because the measurement equations between the sensors are all linearly independent, it can be seen that the rank of the H matrix is greater than the dimension of the relative navigation state volume.

$$\text{Rank}(\mathbf{H}_k^r) > D(\mathbf{X}_r) \quad (51)$$

Therefore, the optimal solution can be found by setting Equation (51) as the objective function.

$$F_{obj} = e = \mathbf{Z}_k^r - \mathbf{H}_k^r \mathbf{X}_k^{r-} \quad (52)$$

According to the principle analysis of the TDO algorithm in the previous section, it can be known that the time complexity of TDO is mainly related to N , m , and T . The algorithm is processed in a loop and the time complexity is $O(N*m*T)$.

The performance of the TDO algorithm is further analyzed using Equation (52) as the objective function. Sensor measurements and navigation data are derived from analogue values. The advantages of TDO are verified by comparing GA and PSO. The optimization results are shown in Figure 5.

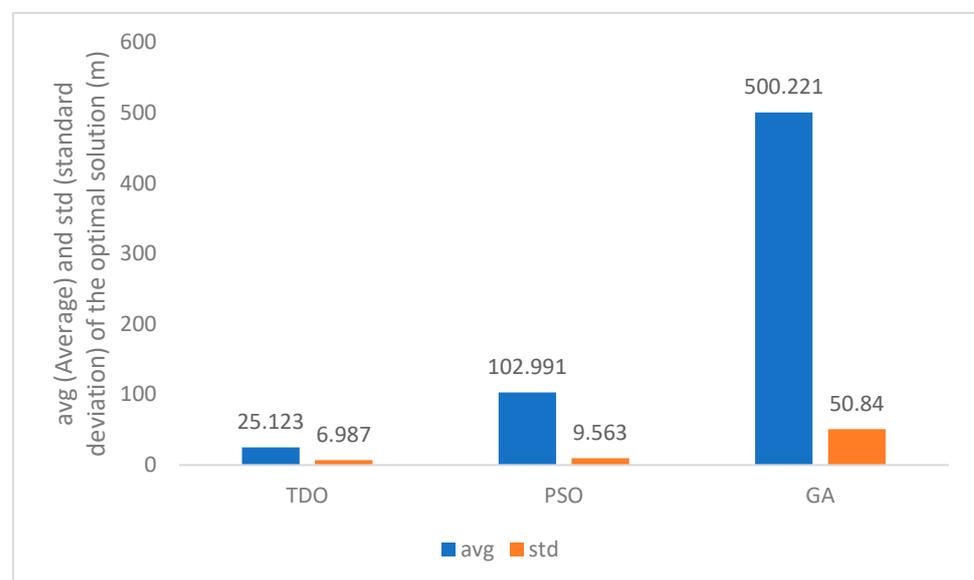


Figure 5. Optimization results of TDO and competitor algorithms.

The TDO algorithm has better performance than traditional optimization algorithms in high-dimensional optimization problems like UAV state solving.

3.5. TDO Adaptive Kalman Filtering Algorithm Flow

Based on the relative motion state equation, adaptive judgment rule, and TDO algorithm proposed above, combination with the extended Kalman filter algorithm can give the adaptive Kalman filter algorithm flow.

Initialize the number of TDO iterations t , the maximum number of TDO iterations T , the number of TDO populations N , the initial covariance matrix of the system P_0 , the initial state of the system X_0 , and the measurement noise covariance matrix R :

Step 1. The system state and covariance matrix are predicted in one step, i.e.,

$$\begin{cases} X_k^{r-} = F_{k,k-1} X_{k-1}^{r+} + B U_k \\ P_k^- = F_{k,k-1} P_{k-1}^+ F_{k,k-1}^T + G_{k-1} Q_{k-1} G_{k-1}^T \end{cases};$$

Step 2. The gain matrix K_k is computed from the measurement noise covariance matrix R , i.e.,

$$K_k = P_k^- H_k^{rT} [H_k^r P_k^- H_k^{rT} + R_k]^{-1};$$

Step 3. Calculate the innovation e and combine it with K_k using Equation (47) for optimal estimation of the relative navigation state, i.e.,

$$X_k^{r+} = X_k^{r-} + K_k e;$$

Step 4. Update the system state covariance matrix, i.e.,

$$P_k^+ = (I - K_k H_k^r) P_k^-;$$

Step 5. Use the new innovation e and R through Equation (48) to Equation (50) to obtain the adaptive filtering threshold λ and \bar{e} ;

Step 6. If $\bar{e} < \lambda$, then the algorithm ends outputting X_k^{r+} with P_k^+ , otherwise go to step 7;

Step 7. If $t < T$, generate N population members by arraying control system state upper and lower bounds in accordance with the objective function Equation (52) with the number of populations N . Then, use Equations (36)–(46) for TDO algorithm loop iteration solution, otherwise the algorithm ends outputting the optimal solution of the relative navigational state X_k^{r+} .

4. Results

4.1. Simulation Initial Conditions

The UAV l and UAV f adopt a close-range accompanying flight mode, and their flight trajectories are shown in Figure 6. It is assumed that each aircraft adopts the GPS/INS tight combination positioning method, and real-time relative navigation is carried out during the flight, and the flight duration is 600 s.

Table 1 lists the parameter settings in the relative navigation method proposed in this paper. The number of satellites, the maximum number of iterations of the TDO algorithm, and the number of TDO population members.

Table 1. Relative navigation method Parameters.

Parameters	Meaning	Value
N_s	Number of available satellites	8
N	Number of TDO stock members	30
T	Maximum number of TDO iterations	800

Table 2 lists the parameters of each sensor device used for the UAV with each measurement noise. Both UAVs are equipped with IMU, satellite navigation receiver, UWB sensors, pseudorange, and pseudorange rate, and the UWB measurement noise is Gaussian white noise.

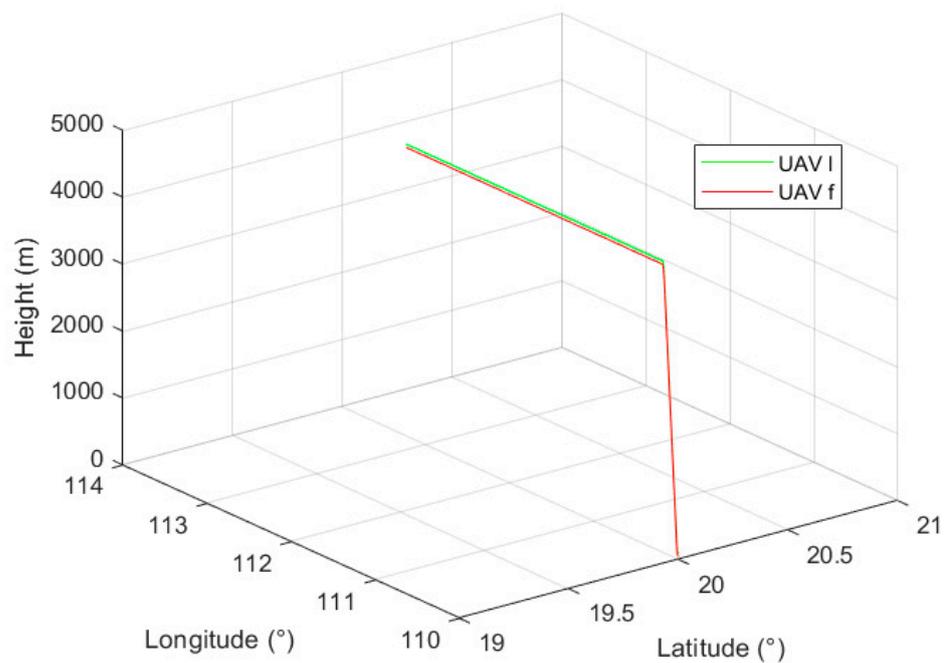


Figure 6. UAV *l* and UAV *f* flight trace.

Table 2. Sensor Configuration and Measurement Noise Settings.

Sensors	Parameters	Value
Gyros	Constant Drift	0.1 (°)/h
	White noise error	0.1 (°)/h
	First-order Markov random noise	0.1 (°)/h
	First-order Markov correlation time	3600 s
Accelerometer	First-order Markov random noise	0.01 g
	First-order Markov correlation time	0.03 m/s
GPS	Pseudorange error	3 m
	Pseudorange rate error	0.03 m/s
UWB	Ranging noise	0.15 m
	Crystal Error Scaling Factor	1×10^{-3}

4.2. Simulation Results and Analysis

We aim to verify that the relative motion state equation derived based on AHRS can more accurately describe the relative motion of the UAV compared to the CA/CV motion equation, and at the same time, adaptive Kalman filtering can reduce the system error in the case of mismatch between the state equation and the motion state; therefore, this paper compares the results of relative navigation in three configurations:

1. AHRS+TDO adaptive Kalman filtering method: the method proposed in this paper, using the relative navigation equation of state derived based on AHRS, using UWB, relative differential, and dual positioning differential data as observation data;
2. CA/CV equation of state: traditional relative motion equation of state, using observation data consistent with method 1;
3. AHRS: only the relative motion equation of state derived based on AHRS is used, using observation data consistent with method 1;

The cumulative distribution function of the relative position errors of the three methods is first compared. As shown in Figure 7, according to the distribution and slope of the CDF curve, the error of method 1 is concentrated near 0 and has the largest slope. The relative position error–CDF curves for Method 2 and Method 3 are similar.

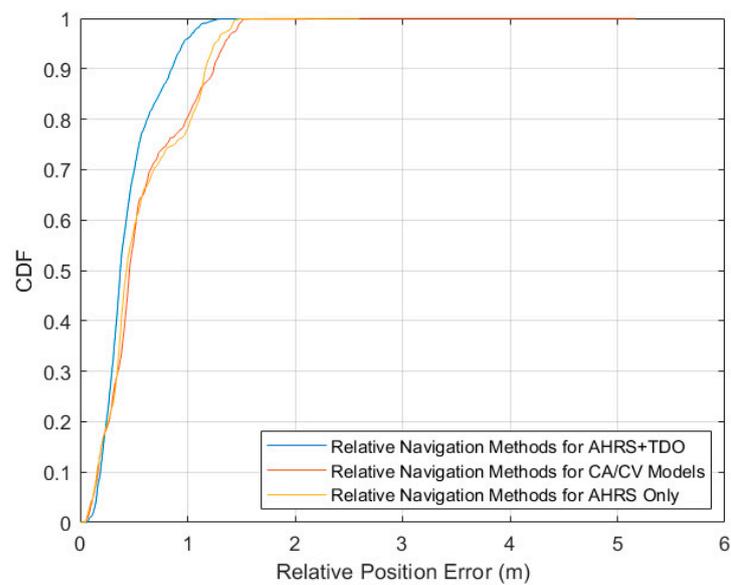


Figure 7. Relative position error versus CDF.

The following text analyses the stability of three methods based on the standard deviation of the position errors in the E-coordinate system's X, Y, and Z directions. Figure 8 shows that Method 1 is the most stable method due to its adaptive method based on the AHRS motion state equation. Method 2, on the other hand, is unable to correct the state value that deviates from the actual motion, resulting in less stability than Method 1. Finally, Method 3, which employs the traditional positioning technique, has the lowest stability among the three methods.

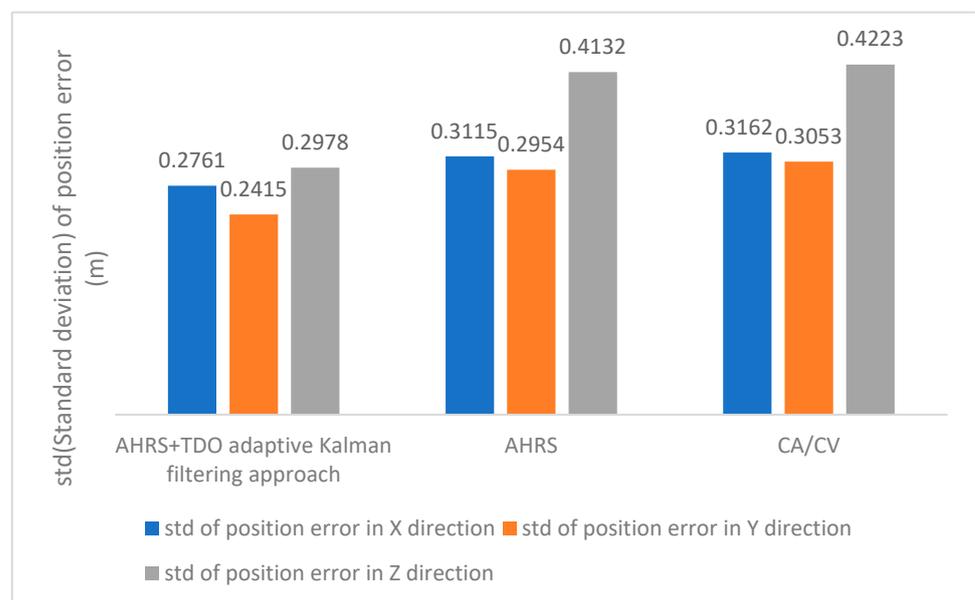
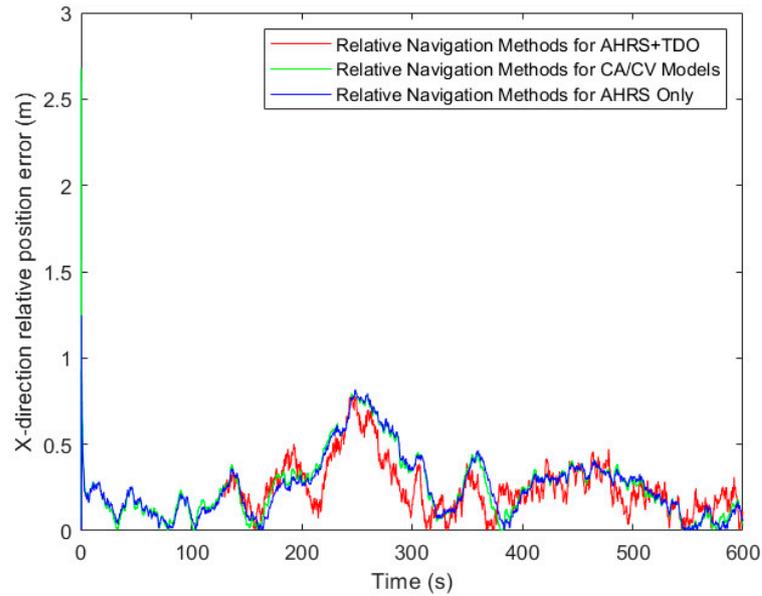
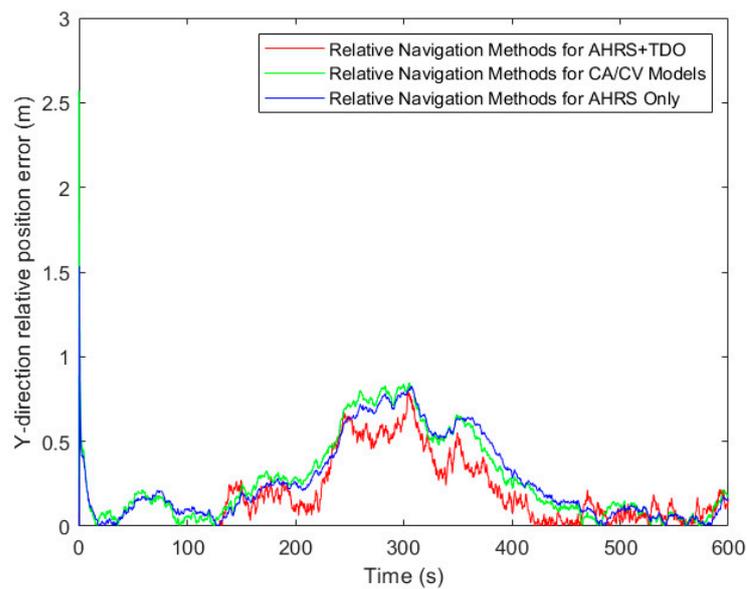


Figure 8. Relative position error in the direction of X, Y, and Z axes std.

Table 3, along with Figures 9–11, presents the simulation results of three methods under the RMSE performance index. Firstly, in terms of relative position results, using the RMSE in X, Y, and Z directions as the comparison term, the AHRS+TDO adaptive Kalman filtering method demonstrated the highest accuracy. The second highest accuracy was obtained by using only the relative motion equation based on AHRS. The traditional CA/CV motion model method showed the worst relative navigation performance.

Table 3. Relative position error comparison.

Relative Position Error	RMSE(m)		
	AHRS+TDO Adaptive Kalman Filtering Approach	AHRS	CA/CV
X-direction	0.27857	0.31351	0.31852
Y-direction	0.28461	0.35852	0.36996
Z-direction	0.34116	0.49446	0.50831

**Figure 9.** Comparison of relative position error in X-direction.**Figure 10.** Comparison of relative position error in Y-direction.

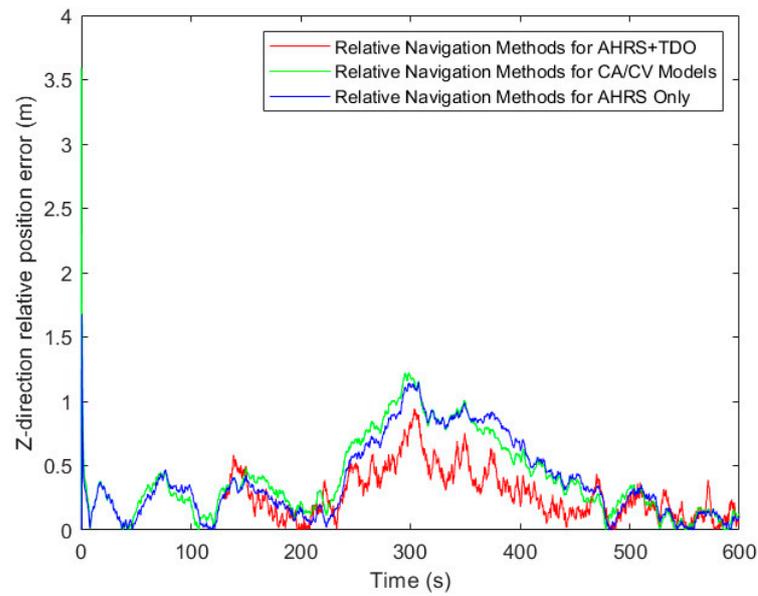


Figure 11. Comparison of relative position error in Z-direction.

The relative velocity error CDF curves for Method 1 and Method 3, are shown in Figure 12. The error distribution of method 1 is significantly better than that of method 3. This is due to the accurate description of the relative motion by the AHRS-based motion model proposed in this paper.

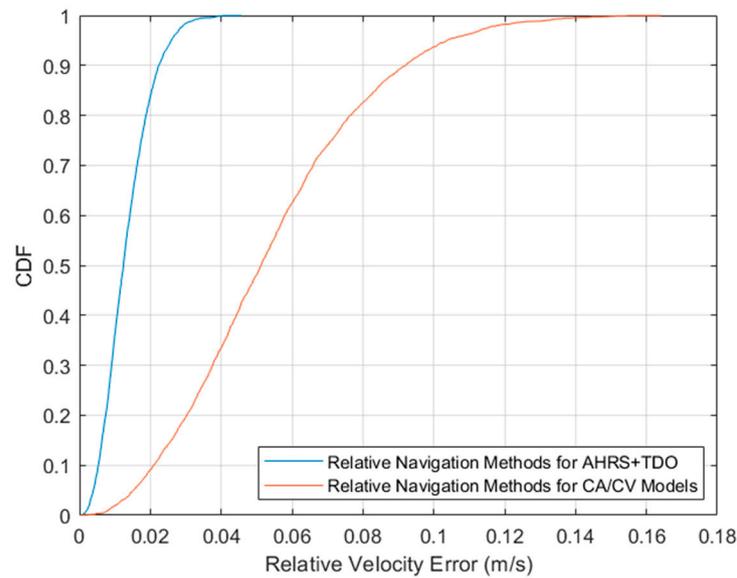


Figure 12. Relative velocity error CDF.

Method 1 and method 3's relative velocity error STDs are shown in Figure 13. Simulation results demonstrate that the AHRS-based adaptive filtering method has better stability.

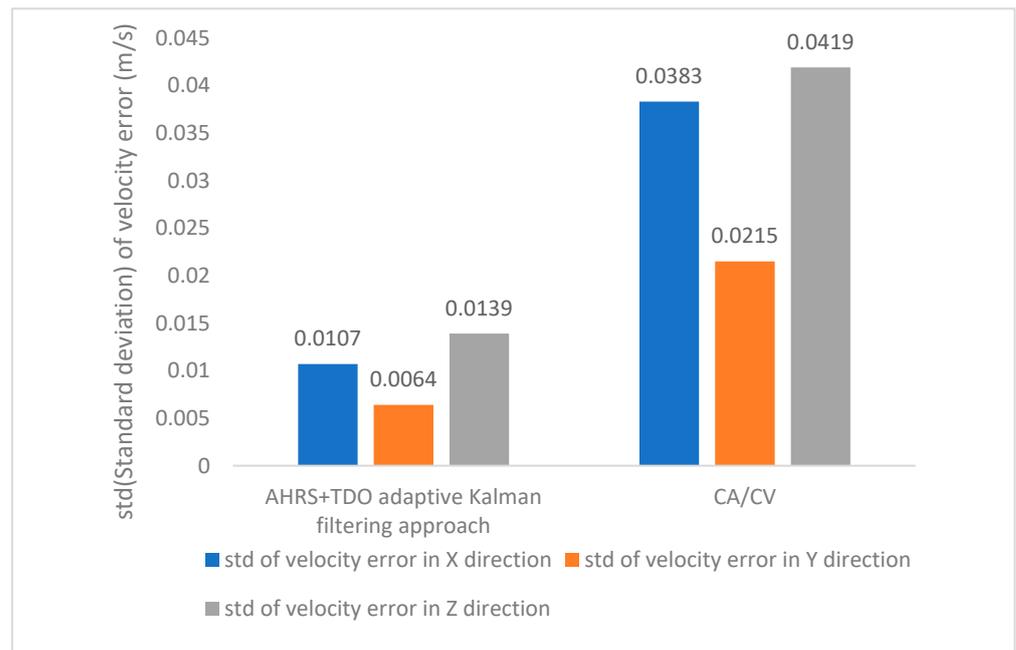


Figure 13. Relative velocity error in the direction of X, Y, and Z axes std.

Upon comparing the relative velocity error, the algorithm proposed in this paper has significantly enhanced the relative velocity navigation accuracy in comparison to the traditional CA/CV model. This can be observed in Figures 14–16. The accuracy of relative navigation in the X, Y, and Z directions has improved by 3.5, 3.3, and 3 times, respectively. For more specific data, please refer to Table 4.

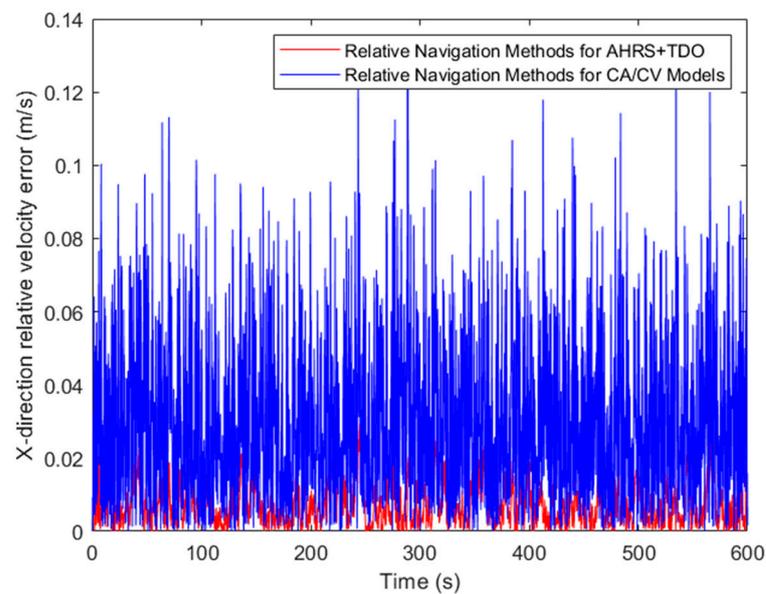


Figure 14. Comparison of relative velocity error in X-direction.

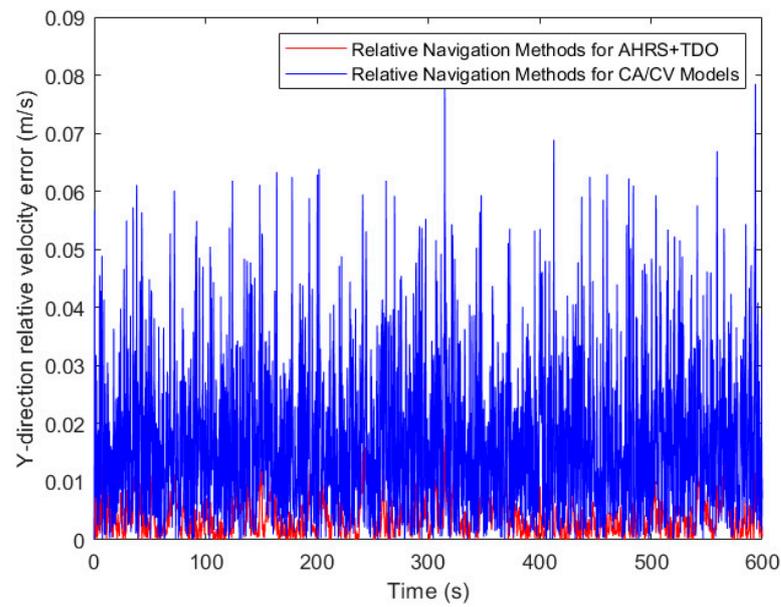


Figure 15. Comparison of relative velocity error in Y-direction.

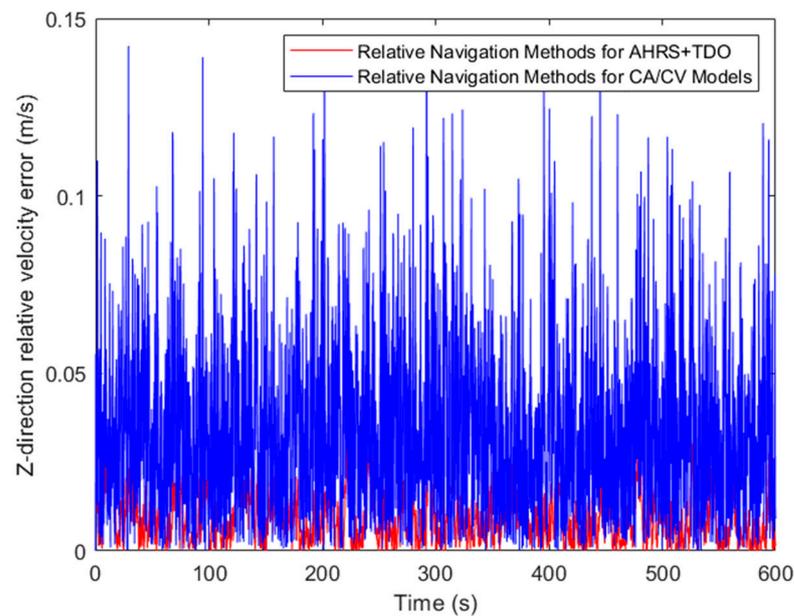


Figure 16. Comparison of relative velocity error in Z-direction.

Table 4. Relative speed error comparison.

Relative Speed Error	RMSE (m/s)	
	AHRS+TDO Adaptive Kalman Filtering Approach	CA/CV
X-direction	0.01076	0.03826
Y-direction	0.00644	0.02151
Z-direction	0.01393	0.04194

This paper proposes a UAV navigation method based on AHRS. The system model uses the relative motion equation to enhance the method's accuracy. The multi-stage filter

structure corrects the sensor error. Additionally, the method utilizes adaptive judgment rules to correct the state that deviates from the real motion, which further improves the method's stability and accuracy.

5. Conclusions

In this paper, relative navigation methods for UAVs are investigated. Aiming to solve the problem of inaccurate description of the relative navigation state by the traditional CA/CV motion model, the state equation for relative navigation is derived based on the aerial position information provided by AHRS, and a state equation more suitable for relative navigation scenarios is proposed. Considering the navigation accuracy degradation problem caused by the mismatch between the state equation and the motion state during the navigation process, the adaptive judgment rule based on innovation judgment is proposed, the TDO objective function is designed by using the measurement information and the measurement equation, and, finally, the state quantities that deviate from the real motion are solved by the TDO algorithm. The simulation results show that the relative navigation method proposed in this paper can effectively improve the accuracy of the navigation system, effectively reduce the relative position error under the same sensor conditions compared with the traditional CA/CV model, and simultaneously reduce the relative speed error substantially, and it is suitable for the application scenarios where the UAVs have high requirements for relative navigation accuracy and robustness such as in formation.

Author Contributions: Conceptualization, Y.L. and J.X.; methodology, Y.L.; software, Y.L.; writing—original draft preparation, Z.L.; validation and visualization, Y.L.; writing—review and editing, Y.L. and K.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Natural Science Foundation of China, grant number 62203228, and the open research fund of the Key Lab of Broadband Wireless Communication and Sensor Network Technology (Nanjing University of Posts and Telecommunications), Ministry of Education, grant number JZNY202114.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data are contained within the article.

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

References

1. Xiong, J.; Xiong, Z.; Yu, Y. Ultra-wideband ranging-assisted UAV proximity relative navigation method. *Chin. J. Inert. Technol.* **2018**, *26*, 346–351.
2. Fosbury, A.M.; Crassidis, J.L. Relative navigation of air vehicles. *J. Guid. Control Dyn.* **2008**, *31*, 824–834. [[CrossRef](#)]
3. Cheng, J.; Ren, P.; Deng, T. A Novel Ranging and IMU-Based Method for Relative Positioning of Two-MAV Formation in GNSS-Denied Environments. *Sensors* **2023**, *23*, 4366. [[CrossRef](#)] [[PubMed](#)]
4. Zheng, Z.; Jin, Z.; Sun, L.; Zhu, M. Adaptive Sliding Mode Relative Motion Control for Autonomous Carrier Landing of Fixed-Wing Unmanned Aerial Vehicles. *IEEE Access* **2017**, *5*, 5556–5565. [[CrossRef](#)]
5. Yu, H.; Zhu, W.P.; Champagne, B. Speech enhancement using a DNN-augmented colored-noise Kalman filter. *Speech Commun.* **2020**, *125*, 142–151. [[CrossRef](#)]
6. Jin, K.; Chai, H.Z.; Su, C. Adaptive filtering of fiber optic gyro random noise considering colored noise. *J. Surv. Mapp.* **2022**, *51*, 80–89.
7. Ramezani, A.; Safarinejadian, B.; Zarei, J. Fractional order chaotic cryptography in colored noise environment by using fractional order interpolatory cubature Kalman filter. *Trans. Inst. Meas. Control* **2019**, *41*, 3206–3222. [[CrossRef](#)]
8. Liu, C.; Wei, J.; Li, W. Adaptive UKF in colored observation noise for multipath error reduction in BeiDou. *J. Electron. Meas. Instrum.* **2023**, *34*, 101–107.
9. Xu, S.; Lin, X. Robust CKF-based multi-sensor full information fusion algorithm. *J. Electr. Mach. Control* **2013**, *17*, 90–97.
10. Zhou, D.H.; Xi, Y.G.; Zhang, Z.J. A suboptimal multiple fading extended Kalman filter. *Acta Autom. Sin.* **1991**, *17*, 689–695.
11. Sun, Y.; Lu, T.; Chen, Q. An improved volumetric Kalman filter based on strong tracking. *J. Huazhong Univ. Sci. Technol.* **2013**, *S1*, 451–454.

12. Li, N.; Zhu, R.; Zhang, Y. A strong tracking square root CKF algorithm based on multiple fading factors for target tracking. In Proceedings of the 2014 Seventh International Joint Conference on Computational Sciences and Optimization, Beijing, China, 4–6 July 2014; IEEE: Piscataway, NJ, USA, 2014; pp. 16–20.
13. Zhang, A.; Bao, S.; Bi, W.; Yuan, Y. Low-cost adaptive square-root cubature Kalman filter for systems with process model uncertainty. *J. Syst. Eng. Electron.* **2016**, *27*, 945–953. [[CrossRef](#)]
14. Zhang, H.; Junwei, X.; Ge, J. Strongly tracked square-root volumetric Kalman filtering algorithm for adaptive CS models. *Syst. Eng. Electron.* **2019**, *41*, 9.
15. Wang, Z.; Liu, Z.; Tian, K.; Zhang, H. Frequency-scanning interferometry for dynamic measurement using adaptive Sage-Husa Kalman filter. *Opt. Lasers Eng.* **2023**, *165*, 107545. [[CrossRef](#)]
16. Yuan, G.; Yuan, K.; Zhang, H. A variable proportion adaptive federal kalman filter for INS/ESGM/GPS/DVL integrated navigation system. In Proceedings of the Fourth International Joint Conference on Computational Sciences and Optimization, Kunming, China, 15–19 April 2021; IEEE: Piscataway, NJ, USA, 2011; pp. 978–981.
17. Arasaratnam, I.; Haykin, S. Cubature Kalman filters. *IEEE Trans. Autom. Control* **2009**, *54*, 1254–1269. [[CrossRef](#)]
18. Arasaratnam, I.; Haykin, S.; Hurd, T.R. Cubature Kalman filtering for continuous-discrete systems: Theory and simulations. *IEEE Trans. Signal Process.* **2010**, *58*, 4977–4993. [[CrossRef](#)]
19. Li, C.; Ma, J.; Yang, Y. Low complexity adaptive volumetric Kalman filtering algorithm. *J. Beijing Univ. Aero-Naut. Astronaut.* **2022**, *48*, 716–724.
20. Retscher, G.; Kiss, D.; Gabela, J. Fusion of GNSS Pseudoranges with UWB Ranges Based on Clustering and Weighted Least Squares. *Sensors* **2023**, *23*, 3303. [[CrossRef](#)] [[PubMed](#)]
21. Song, P.C.; Pan, J.S.; Chu, S.C. A parallel compact cuckoo search algorithm for three-dimensional path planning. *Appl. Soft Comput.* **2020**, *94*, 106443. [[CrossRef](#)]
22. Roberge, V.; Tarbouchi, M.; Labonté, G. Fast genetic algorithm path planner for fixed-wing military UAV using GPU. *IEEE Trans. Aerosp. Electron. Syst.* **2018**, *54*, 2105–2117. [[CrossRef](#)]
23. Roberge, V.; Tarbouchi, M.; Labonté, G. Comparison of parallel genetic algorithm and particle swarm optimization for real-time UAV path planning. *IEEE Trans. Ind. Inform.* **2012**, *9*, 132–141. [[CrossRef](#)]
24. Fu, Y.; Ding, M.; Zhou, C.; Hu, H. Route planning for unmanned aerial vehicle (UAV) on the sea using hybrid differential evolution and quantum-behaved particle swarm optimization. *IEEE Trans. Syst. Man Cybern. Syst.* **2013**, *43*, 1451–1465. [[CrossRef](#)]
25. Sun, Z.; Wu, J.; Yang, J.; Huang, Y.; Li, C.; Li, D. Path planning for GEO-UAV bistatic SAR using constrained adaptive multiobjective differential evolution. *IEEE Trans. Geosci. Remote Sens.* **2016**, *54*, 6444–6457. [[CrossRef](#)]
26. Phung, M.D.; Ha, Q.P. Safety-enhanced UAV path planning with spherical vector-based particle swarm optimization. *Appl. Soft Comput.* **2021**, *107*, 107376. [[CrossRef](#)]
27. Shi, D.S.; Tian, F. Three-dimensional deployment and optimization of UAV communication based on multi-objective particle swarm algorithm. *J. Nanjing Univ. Posts Telecommun.* **2022**, *42*, 11.
28. Wu, T.; Bai, R.; Zhu, L. Design of a navigation posture reference system based on Kalman filtering. *J. Sens. Technol.* **2016**, *29*, 531–535.
29. Peng, F.Q.; Wen, Y.; Jin, Q.F. Effects of perturbed gravity on inertial navigation. *Surv. Mapp. Sci. Eng.* **2014**, *34*, 6.
30. Wu, Q.; Wu, R.; Han, F.; Zhang, R. A Three-Stage Accelerometer Self-Calibration Technique for Space-Stable Inertial Navigation Systems. *Sensors* **2018**, *18*, 2888. [[CrossRef](#)] [[PubMed](#)]
31. Su, X.L.; Wang, H.; Jian, X. Drift stability analysis of platform inertial navigation gyro. *Mod. Transp. Technol. Res.* **2024**, *6*, 101–105.
32. Qin, Y.Y. *Kalman Filtering and Combinatorial Navigation Principles*; Northwestern Polytechnical University Press: Xi'an, China, 2021.
33. Dong, M. A low-cost NLOS identification and mitigation method for UWB ranging in static and dynamic environments. *IEEE Commun. Lett.* **2021**, *25*, 2420–2424. [[CrossRef](#)]
34. Dehghani, M.; Hubálovský, Š.; Trojovský, P. Tasmanian devil optimization: A new bio-inspired optimization algorithm for solving optimization algorithm. *IEEE Access* **2022**, *10*, 19599–19620. [[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.