Research Article

Robust Interacting Multiple Model Unscented Particle Filter for Navigation

Li Xue , Yulan Han, and Chunning Na

School of Physics and Electronic-Electrical Engineering, Ningxia University, Yinchuan 750021, China

Correspondence should be addressed to Li Xue; xuelixuelixueli201@163.com

Received 27 August 2020; Revised 9 October 2020; Accepted 16 October 2020; Published 10 November 2020

Academic Editor: Guo Wei

Copyright © 2020 Li Xue et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

In order to solve the problems of particle degradation and difficulty in selecting importance density function in particle filter algorithm, a robust interacting multiple model unscented particle filter algorithm is presented, which is based on the advantages of interacting multiple model and particle filter algorithms. This algorithm can use the unscented transformation to get the particles that contain the latest measurement information of each model and calculate the robust equivalent weight function. This robust factor is designed to adjust the estimation and variance, and the important distribution function adaptively obtained is closer to the true distribution. Then, the particles weights can be flexibly adjusted in real time by using Euclidean distance to improve the computational efficiency during the resampling process. In addition, this filter process can comprehensively describe the uncertainty of the statistics characteristic of observation noise between different models. The diversity of available particles is increased, and the filter precision is improved. The proposed algorithm is applied to the SINS/GPS integrated navigation system, and the simulation analysis results demonstrate that the algorithm can effectively improve the filter performance and the calculation precision in positioning of integrated navigation system; thus, it provides a new method for nonlinear model filter.

1. Introduction

The problem of nonlinear filter has its origins in the areas of target tracking and signal processing; nevertheless, the underlying setting is extremely general and is ubiquitous in many applied areas such as integrated navigation, geodetic positioning, automatic control, finance, and artificial intelligence, where random processes are used to model the complex dynamical phenomenon. At present, the extended Kalman filter (EKF), unscented Kalman filter (UKF), central difference filter (CDKF), the particle filter (PF), and so on are widely used nonlinear filter algorithms. So, these methods are applied in many fields. Among them, the PF has also received much attention in recent years [1, 2]. In essence, the EKF, UKF, and CDKF algorithms are used to estimate the state of a nonlinear and non-Gaussian stochastic system from observation data. The UKF and CDKF are generally superior to the EKF in terms of accuracy, and the computational complexities of the three algorithms are roughly in the same order. However, this process inevitably involves

a gross error when there is a deviation between the theoretical model and the practical model, leading to the biased or even divergent filter solution [3]. The PF algorithm [4, 5] is the realization of Bayesian estimation under the nonlinear conditions. It can obtain highly efficient density function that can fully reflect the measurement information. How to sample high quality particles, reduce degradation, and improve estimation accuracy is the research focus of current particle filter algorithm. On this basis, the unscented particle filter (UPF) algorithm [6–8] is proposed, which makes a good optimization and supplement in the calculation process. However, when the number of particles is not distributed properly and the models involve errors, the UPF algorithm still suffers from the phenomenon of particle degradation, and it can affect the estimation accuracy.

In addition, the multiple model algorithm is suitable for the complex system, including interacting multiple model (IMM), autonomous multiple model (AMM), and cooperating multiple model (CMM) estimations. Among them, the IMM algorithm uses Markov transition probability to switch

multiple models through reasonable assumption management, which has received extensive attention and application research [9, 10]. This IMM is one of the several approaches to handle the maneuvering problem, and multiple models are used to describe the motion of the maneuvering target [11]. However, in many cases, the precise system model sets design of the IMM algorithm is complicated. When the prior information is uncertain, the filter accuracy will be reduced by using fixed Markov transition probability matrix. The KF or EKF is usually used in each model filter process in nonlinear and non-Gaussian system model, and its filter performance will be greatly reduced [12, 13]. In the IMM algorithm, with the unscented particle algorithm, the multimodel structures can adapt to high mobility target, and the particle filter can handle nonlinear non-Gaussian problems with unscented transition. The robust estimation is based on the statistical analysis method, and the robust factors are calculated according to the regional segmentation that can better reflect the latest measurement information. Therefore, a simple and effective robust interacting multiple models unscented particle filter (RIMMUPF) algorithm is proposed to obtain more accurate state variable estimation in the system, which could ensure the diversity of particles and improve the accuracy of filter calculation. In addition to this, the new algorithm should not only utilize the richness of all available information as much as possible, but also satisfy the high-precision requirements of complex systems due to the combination of different observation modes.

2. Interacting Multiple Model Unscented Particle Filter

2.1. Nonlinear System Model. Consider the following nonlinear system:

$$\begin{cases} \mathbf{x}_k = f(\mathbf{x}_{k-1}, \mathbf{w}_{k-1}), \\ \mathbf{y}_k = h(\mathbf{x}_k, \mathbf{v}_k), \end{cases} \quad (1)$$

where $\mathbf{x}_k \in R^n$ is the state vector at epoch k , $\mathbf{y}_k \in R^n$ is the system observation, $\mathbf{w}_{k-1} \in R^n$ is the system state noise with variance Q_k , $\mathbf{v}_k \in R^n$ is the measurement noise with variance R_k , $f(\cdot)$ and $h(\cdot)$ are nonlinear functions, and $k = 0, 1, \dots, N$ is the sampling time. The initial state obeys any distribution, $E(\mathbf{x}_0) = \bar{\mathbf{x}}_0$ denotes the initial mean, and $\text{cov}(\mathbf{x}_0) = \mathbf{P}_0$ denotes the initial covariance matrix [14, 15].

2.2. The Main Steps of UPF Algorithm

(1) Initialization

For $k = 0$, draw N sampling points according to the initial mean and variance, $\mathbf{x}_0^i \sim p(\mathbf{x}_0)$, where $i = 1, 2, \dots, N$. Assume the initial weights are $W_0^{(i)} = 1/N$.

(2) Importance Sampling

For $k = 1, 2, \dots, N$, $j = 0, \dots, 2L$, for each sample point $\mathbf{x}_{k-1}^i$ use unscented transformation and can further calculate update particle get $\hat{\mathbf{x}}_k^i$ and $\mathbf{P}_k^i$ [16, 17].

First, calculate the Sigma point χ_k^i and that of the particle weight ω_k^i . In the process of time update, $\chi_{j,k-1}^i$ is calculated according to the sigma point sampling strategy, and $\chi_{j,k/k-1}^i$ is calculated by the nonlinear state transfer function $f(\cdot)$, so as to obtain one-step state prediction $\hat{\mathbf{x}}_{k/k-1}$ and covariance matrix $\mathbf{P}_{k/k-1}$,

$$\begin{aligned} \chi_{j,k/k-1}^i &= f(\chi_{j,k-1}^i, \mathbf{w}_{k-1}); \quad j = 0, \dots, 2N, \\ \hat{\mathbf{x}}_{k/k-1}^i &= \sum_{j=0}^{2N} \omega_j^m \chi_{j,k/k-1}^i, \\ \mathbf{P}_{k/k-1}^i &= \sum_{j=0}^{2N} \omega_j [\chi_{j,k/k-1}^i - \hat{\mathbf{x}}_{k/k-1}^i] [\chi_{j,k/k-1}^i - \hat{\mathbf{x}}_{k/k-1}^i]^T. \end{aligned} \quad (2)$$

Similarly, $\gamma_{j,k/k-1}^i$ is calculated by the nonlinear measurement function $h(\cdot)$, and the output prediction $\hat{\mathbf{y}}_{k/k-1}$, $\mathbf{P}_{yy}$ can be obtained as follows:

$$\gamma_{j,k/k-1}^i = h(\chi_{j,k/k-1}^i), \quad (3)$$

$$\begin{aligned} \hat{\mathbf{y}}_{k/k-1}^i &= \sum_{j=0}^{2N} \omega_j \gamma_{j,k/k-1}^i \\ &= \sum_{j=0}^{2N} \omega_j h(\chi_{j,k/k-1}^i), \end{aligned} \quad (4)$$

$$\begin{aligned} \mathbf{P}_{yy}^i &= \sum_{j=0}^{2N} \omega_j [\gamma_{j,k/k-1}^i - \hat{\mathbf{y}}_{k/k-1}^i] \\ &\quad [\gamma_{j,k/k-1}^i - \hat{\mathbf{y}}_{k/k-1}^i]^T, \end{aligned} \quad (5)$$

$$\begin{aligned} \mathbf{P}_{xy}^i &= \sum_{j=0}^{2N} \omega_j [\chi_{j,k/k-1}^i - \hat{\mathbf{x}}_{k/k-1}^i] \\ &\quad [\gamma_{j,k/k-1}^i - \hat{\mathbf{y}}_{k/k-1}^i]^T. \end{aligned} \quad (6)$$

Then, the state estimation and covariance of the analysis can be computed by [18, 19]

$$\hat{\mathbf{x}}_k^i = \hat{\mathbf{x}}_{k/k-1}^i + \mathbf{K}_k^i (\mathbf{y}_k - \hat{\mathbf{y}}_{k/k-1}^i), \quad (7)$$

$$\mathbf{K}_k^i = \mathbf{P}_{xy}^i (\mathbf{P}_{yy}^i)^{-1}, \quad (8)$$

$$\mathbf{P}_k^i = \mathbf{P}_{k/k-1}^i - \mathbf{K}_k^i \mathbf{P}_{yy}^i (\mathbf{K}_k^i)^T, \quad (9)$$

where $\mathbf{K}_k^i$ denotes filter gain matrix.

(3) Calculate the particle weights

$$\omega_k^i = \omega_{k-1}^i \frac{p(\mathbf{y}_k | \mathbf{x}_k^i) p(\mathbf{x}_k^i | \mathbf{x}_{k-1}^i)}{q(\mathbf{x}_k^i | \mathbf{x}_{k-1}^i, \mathbf{y}_k)}. \quad (10)$$

And normalize them as

$$\tilde{\omega}_k^i = \frac{\omega_k^i}{\sum_{i=1}^N \omega_k^i}, \quad i = 1, \dots, N. \quad (11)$$

- (4) Calculate the state estimate and return to step (2).

2.3. The Main Steps of IMMUPF Algorithm. The IMM estimation algorithm is an iterative recursive projection method. Combined with the UPF algorithm above, the interacting multiple model unscented particle filter (IMMUPF) algorithm [20, 21] is obtained as follows:

- (1) Initialization

For $k = 0$, draw N sampling points according to the initial mean and variance $x_0^i \sim p(x_0)$, where $i = 1, 2, \dots, N$. Assume the initial weights are $W_0^{(i)} = 1/N$, and randomly draw particles at time k . Assume the particle state value of each model is $\tilde{x}_{k|k}^j$, and variance is $\tilde{P}_{k|k}^j$.

- (2) Input interaction: Input interaction for the corresponding particles of each model.

$$\tilde{x}_{k|k}^j = \sum_{i=1}^m \mathbf{u}_{k|k}^{ij} \tilde{x}_{k|k}^i, \quad (12)$$

$$\tilde{P}_{k|k}^j = \sum_{i=1}^m \mathbf{u}_{k|k}^{ij} \left\{ \tilde{P}_{k|k}^i + [\tilde{x}_{k|k}^i - \tilde{x}_{k|k}^j][\tilde{x}_{k|k}^i - \tilde{x}_{k|k}^j]^T \right\}, \quad (13)$$

where the matching model probability is $\mathbf{u}_{k|k}^{ij} = (1/\bar{C}_j^n) \pi_{ij} u_{i,k}^n$, $\bar{C}_j^n = \sum_{i=1}^m \pi_{ij} u_{i,k}^n$. In the model transformation process, π_{ij} is the transition probability under unconstrained conditions, then $\sum_{i=1}^m \pi_{ij} = 1$, $i = 1, 2, \dots, m$, $j = 1, 2, \dots, m$, $n = 1, 2, \dots, N$.

- (3) The model matches UPF filter: $\{\tilde{x}_{k|k}^j, \tilde{P}_{k|k}^j, n = 1, 2, \dots, N\}$ can be brought into the UPF filter, and the likelihood function under the nonlinear model is obtained as

$$\Lambda_{j,k+1}^n = \frac{1}{n} \sum_{j=1}^n N[\mathbf{y}_{j,k+1} - \hat{\mathbf{y}}_{j,k+1/k}^n; 0; \mathbf{S}_{j,k+1}], \quad (14)$$

where $\mathbf{S}_{j,k+1}$ is the covariance of the residual.

- (4) Resampling: To evaluate the weight of particles, the procedure is the same as that of the previous particle filter.

- (5) Model probability update

$$\mathbf{u}_{k+1}^j = \frac{1}{C^n} \Lambda_{j,k+1}^n \bar{C}_j^n, \quad (15)$$

where $C^n = \sum_{j=1}^m \Lambda_{j,k+1}^n \bar{C}_j^n$, $\Lambda_{j,k+1}^n$ is the likelihood function of the n th particle at time $k + 1$.

- (6) Combined output: The weighted sum calculation is completed for the m models corresponding particle sets.

In the filter process, the IMMUPF algorithm can be to autonomously integrate the multiple models through Markov parameters to switch between different models [22].

3. Robust Interacting Multiple Model Unscented Particle Filter

The proposed robust interacting multiple model unscented particle filter algorithm improves the robust adjustment capability of the system model by modifying the measurement and state covariance in real time, so as to ensure the accuracy of the calculation. The IMM process can take into consideration the probability of each model by Markov transition probability. The basic design idea of the algorithm is as follows: firstly, these groups of corresponding particles are interacted as inputs to each model, and this process can reinitialize the filter input value; secondly, the unscented transformation is carried out for each model, and at the time k the Sigma point is calculated to obtain the estimations with the latest measurement information. It can calculate the equivalent weight functions at different time periods to obtain the robust factor. This process can not only adjust state estimations and variances, but also take into account the latest measurements information. So, the model is matched with the robust unscented particle filter; thirdly, in the resampling process, the calculated weights of the particles can be adjusted by the Euclidean distance to increase the weights of the useful particles; then, it can update the corresponding model probability and calculate the output interaction for the particle sets of each model; finally, the state estimation is realized by constantly cyclic updating of the particles. So, this filter process can also make efficient use of the latest measurement information and can be widely employed in the positioning applications to describe the uncertainty of the system model and statistics characteristic of observation noise. The RIMMUPF algorithm includes the following steps:

Step 1: Assume that the Markov model transition probability matrix is $\mathbf{T} = \{\pi_{lj}; l, j = 1, 2, \dots, m\}$, and each model probability matrix is $\mathbf{U} = \{u_{l,k}^n; l = 1, 2, \dots, m; n = 1, 2, \dots, N\}$, where m is

the number of models, and N is the number of particles.

Step 2: Initialization. For $k = 0$, draw N particle sampling points according to the initial mean and variance, $x_0^i \sim p(x_0)$, where $i = 1, 2, \dots, N$. Assume that the initial weights are $W_0^{(i)} = 1/N$, where $\bar{x}_0^i = E(x_0^i)$ and $\mathbf{P}_0^i = E[(x_0^i - \bar{x}_0^i)(x_0^i - \bar{x}_0^i)^T]$.

Step 3: Input interaction. It can complete input interaction for the corresponding particles of each model by using the IMMUPF algorithm formulae (12) and (13). This process can reinitialize the filter input value, and the initial value of each filter obtained from the previous filter process. The state estimation and the conditional transition probability of the model are calculated using the system prior information.

Step 4: For k time, calculate the robust equivalent weight function and adjust particles completed unscented transformation, then get $\hat{x}_k^{i*}$ and $\mathbf{P}_k^{i*}$, as follows,

- (1) For robust factor, namely, equivalent weight function, it can select the equivalent weight matrix constructed by using the IGG descending weight function [23, 24]. The measured value can be controlled by robust factor. If the reciprocal function is taken, it is defined as variance expansion factor function. Assume that $\bar{P} = \text{diag}(\bar{P}_1, \bar{P}_2, \dots, \bar{P}_k)$, and the IGG weight function followed:

$$\bar{P}_i = \begin{cases} P_i, & |V_i| \leq k_0, \\ P_i \frac{k_0}{|V_i|}, & k_0 < |V_i| \leq k_1, \\ 0, & |V_i| > k_1, \end{cases} \quad (16)$$

or

$$\bar{P}_i = \begin{cases} P_i, & |V_i| \leq k_0, \\ P_i \frac{k_0}{|V_i|} \frac{(k_1 - |V_i|)^2}{(k_1 - k_0)^2}, & k_0 < |V_i| < k_1, \\ 0, & k_1 \leq |V_i|, \end{cases} \quad (17)$$

where k is a constant, $k_0 \in (1, 1.5)$ and $k_1 \in (3, 8)$, $\mathbf{P}_i$ is obtained by formula (5), and V_k is the residual vector of the observed value y_k .

And calculate Sigma points and update particles $\{x_{k-1}^i, p_{k-1}^i\}$ by unscented transformation; χ_k^i and ω_k^i are calculated as follows,

$$\begin{cases} \chi_{0,k-1}^i = \hat{x}_{k-1}^i, \\ \chi_{j,k-1}^i = \hat{x}_{k-1}^i + \sqrt{(N+\lambda)(\mathbf{P}_{k-1}^i)_j}; & j = 1, \dots, N, \\ \chi_{j,k-1}^i = \hat{x}_{k-1}^i - \sqrt{(N+\lambda)(\mathbf{P}_{k-1}^i)_j}; & j = n+1, \dots, 2N, \\ \omega_0^m = \frac{\lambda}{(N+\lambda)}, \\ \omega_0^c = \frac{\lambda}{(N+\lambda)} + (1 - \alpha^2 + \beta), \\ \omega_j^m = \omega_j^c = \frac{\lambda}{2(N+\lambda)}; & j = 1, \dots, 2N, \end{cases} \quad (18)$$

where $\lambda = \alpha^2(N + \kappa)$ denotes the scale factor, κ is zero, and α at the range $10^{-3} < \alpha \leq 1$, then the value is 1, and β is 2; ω_j is the weight of the j th Sigma point, which satisfies $j = 0, 1, \dots, 2N$.

The particles are used to predict and update, and $\{\hat{x}_k^i, p_k^i\}$ can be obtained by formulae (7) to (9). The robust equivalent weight factor is used to calculate the state estimation and variance $\hat{x}_k^{i*}, p_k^{i*}$ of the sampled particles.

$$\begin{aligned} \hat{x}_k^{i*} &= \hat{x}_{k/k-1}^i + \mathbf{K}_k^{i*} (y_k - \hat{y}_{k/k-1}), \\ \mathbf{K}_k^{i*} &= \mathbf{P}_{xy}^i (\bar{\mathbf{P}}_{yy}^i)^{-1}, \\ \mathbf{P}_k^{i*} &= \mathbf{P}_{k/k-1}^i - \mathbf{K}_k^{i*} \bar{\mathbf{P}}_{yy}^i (\mathbf{K}_k^i)^T. \end{aligned} \quad (19)$$

The above formula shows that factors $\mathbf{P}_k^{i*}$ can be influenced and regulated by $\bar{\mathbf{P}}_{yy}^i$. This estimation scheme fully takes into account the actual situation of the measurement data and can classify gross errors. It contains three sections: When observation error is small, equivalent weight is selected the original; when the observation error is greater, but not significant, the observations should fall weight; when the observations are notably abnormal, it should be eliminated; that is, take the weight zero. If $\bar{\mathbf{P}}_{yy}^i = \mathbf{P}_{yy}^i$, the $\hat{x}_k^{i*}$ and p_k^{i*} are obtained by the UKF algorithm.

- (2) The time and measurement are updated according to UPF algorithm in Section 2.2; then the particle

state values and covariance of the m models are calculated.

- (3) Calculate the particle weights and normalize.
- (4) The state variables and covariance at the k time can be calculated as follows,

$$\begin{aligned}\hat{\mathbf{x}}_{i,k|k}^{n*} &= \sum_{n=1}^N \tilde{\omega}_k^i \mathbf{x}_k^{i*}, \quad i = 1, 2, \dots, m, \\ \hat{\mathbf{P}}_{i,k|k}^{n*} &= \sum_{n=1}^N \tilde{\omega}_k^i [\mathbf{x}_k^{i*} - \hat{\mathbf{x}}_k^{i*}] [\mathbf{x}_k^{i*} - \hat{\mathbf{x}}_k^{i*}]^T,\end{aligned}\quad (20)$$

where $i = 1, 2, \dots, m; n = 1, 2, \dots, N$.

Step 5: The model matches robust UPF filter.

$\{\hat{\mathbf{x}}_{k|k}^{j*}, \hat{\mathbf{P}}_{k|k}^{j*}, n = 1, 2, \dots, N\}$ is applied to the j th model by the robust UPF filter, then $\hat{\mathbf{x}}_{k+1|k+1}^{j*}$ and $\hat{\mathbf{P}}_{k+1|k+1}^{j*}$ can be calculated at time $k + 1$. The other parameters are also calculated such as $\mathbf{S}_{k+1|k+1}^{j*}$ and $\Lambda_{k+1|k+1}^{j*}$ etc.

Step 6: Residual resampling.

All particle weights in each model are evaluated separately to generate new samples with similar weights. This process sometimes reduces the diversity of particles, which can be selected according to the specific situation.

According to formula (10), the weight is improved as follows:

$$w_k^{i*} = w_k^{i*} + \left(\frac{w_{k-\max}^{i*}}{N} \right) \cdot \sin\left(\frac{L^i}{L_{\max} \cdot (\pi/2)} \right) \cdot \beta, \quad (21)$$

where $w_{k-\max}^{i*}$ denotes the maximum weight value, and $L_{\max}$ and L^i denote the Euclidean distance of the measurement residual $y_{k-\max}^i$ and $y_{k-\min}^i$, as follows:

$$\begin{aligned}L_{\max} &= (y_{k-\min}^i - y_{k-\max}^i)^T \cdot (y_{k-\min}^i - y_{k-\max}^i), \\ L^i &= (y_k^i - y_{k-\max}^i)^T \cdot (y_k^i - y_{k-\max}^i),\end{aligned}\quad (22)$$

where β is the adaptive coefficient, which is determined by the statistical characteristics of the measurement noise. When the equivalent measurement noise is low, take $\beta = 0$, and it does not adjust the likelihood distribution; on the contrary, it can artificially make the likelihood distribution wider [25]. Then, particle weight normalization is done as formula (11).

Step 7: Model probability update. At the current moment, the model probability is obtained according to the previous step results. Update the probability of the model by using the IMMUPF algorithm formula (15).

Step 8: Output interaction. The corresponding particles of the m model can be output interaction, and the formula superscript $*$ is omitted here.

$$\begin{aligned}\mathbf{x}_{k+1|k+1}^n &= \sum_{j=1}^m \hat{\mathbf{x}}_{j,k+1|k+1}^n \mathbf{u}_{j,k+1}^n, \\ \hat{\mathbf{P}}_{k+1|k+1}^n &= \sum_{j=1}^m \mathbf{u}_{j,k+1}^n \left\{ \hat{\mathbf{P}}_{j,k+1|k+1}^n + [\hat{\mathbf{x}}_{j,k+1|k+1}^n - \hat{\mathbf{x}}_{k+1|k+1}^n] [\hat{\mathbf{x}}_{j,k+1|k+1}^n - \hat{\mathbf{x}}_{k+1|k+1}^n]^T \right\}.\end{aligned}\quad (23)$$

This algorithm can recursively calculate the iteration sample particles, to accomplish the robust adaptive adjustment of the state estimation and variance. The robust factor has a good adaptive ability to the make use of the measurement information of system noise, and the important distribution function obtained is closer to the true distribution. The diversity of particles is increased during the sampling process, and the filter precision is improved.

4. Experimental Results and Analysis

Experiments about vehicle testing have been conducted to comprehensively evaluate and analyze the performance of the proposed method for SINS/GPS integrated navigation system [26, 27]. The comparison with the IMMUPF algorithms is also discussed in this section. The testing requires the vehicle to

run steadily. The actual navigation position information of the testing car was obtained by using the high-precision vehicle-mounted differential GNSS receiver UT-206 (< 10 cm position accuracy) and was used as the reference value of the experimental results. Test instruments and equipment include SINS/GPS integrated navigation system, industrial control computer, testing power supply, current voltmeter, and equipment fixing board. In the testing simulation conditions, the vehicle was carried out on a highway, with total mileage about 15 km and the average running speed at 40 km/h. The vehicle testing and loading equipment is shown in Figure 1.

The sensor main parameters of SINS/GPS integrated navigation system followed: SINS replacement rate 100 Hz, GPS replacement 5 Hz; Gyroscope measurement range $\pm 200^\circ/\text{s}$, zero-bias stability $0.5^\circ/\text{hr}$ (normal temperature), resolution ratio $0.12^\circ/\text{h}$, bandwidth 100 Hz, random walk

FIGURE 1: Vehicle testing and loading equipment.

$0.15^\circ/\text{hr}^{1/2}$; the accelerometer measurement range $\pm 12 \text{ g}$, zero-bias stability 0.05 mg (normal temperature), resolution ratio 1.9 ug , bandwidth 100 Hz , random walk $0.06 \text{ m/s/hr}^{1/2}$; during the testing process, the high-precision wide area difference system was used as the position reference system, and the test data was obtained by data storage. In the post-processing, it was used to solve the filtering and verify the performance of filter algorithms. The GPS has a good observation environment, and velocity measurement accuracy was 0.02 m/s (RMS) . At the beginning, the vehicle trajectory position coordinates in SINS/GPS integrated navigation system were initialized and shown in Figure 2.

In SINS/GPS integrated navigation system, the state variables $\mathbf{x}(t)$ are selected as

$$\mathbf{x}(t) = [(\delta\mathbf{q})^T, (\delta\mathbf{V})^T, (\delta\mathbf{P})^T, \varepsilon^T, \nabla^T]^T, \quad (24)$$

where $\delta\mathbf{q} = [\delta q_0, \delta q_1, \delta q_2, \delta q_3]^T$ denotes attitude error quaternion of SINS, $\delta\mathbf{V} = [\delta V_E, \delta V_N, \delta V_U]^T$ denotes velocity error in three direction of SINS, $\delta\mathbf{P} = [\delta L, \delta \lambda, \delta H]^T$ denotes the errors of latitude, longitude and height, $\varepsilon = [\varepsilon_x, \varepsilon_y, \varepsilon_z]^T$ denotes random drift of the gyro, and $\nabla = [\nabla_x, \nabla_y, \nabla_z]^T$ denotes constant bias of the accelerometer. So, the system state equation can be expressed as

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, t) + \mathbf{w}(t), \quad (25)$$

where $\mathbf{f}(\mathbf{x}, t)$ is the system nonlinear state function, and $\mathbf{w}(t)$ is the system process noise vector, where

FIGURE 2: Vehicle trajectory position coordinates.

$$\mathbf{f}(\mathbf{x}, t) = \begin{bmatrix} \mathbf{B}(\mathbf{I} - \mathbf{C}_n^c)\omega_{in}^n - \mathbf{B}\mathbf{C}_b^c \boldsymbol{\varepsilon}^b \\ (\mathbf{I} - \mathbf{C}_c^n)\mathbf{C}_b^c \mathbf{f}^b - (2\omega_{ie}^n + \omega_{en}^n) \times \delta\mathbf{V}^n \\ -(2\delta\omega_{ie}^n + \delta\omega_{en}^n) \times \mathbf{V}^n + \mathbf{C}_b^c \nabla^b \\ \mathbf{M}\delta\mathbf{V} + \mathbf{N}\delta\mathbf{L} \\ 0_{3 \times 1} \\ 0_{3 \times 1} \end{bmatrix}, \quad (26)$$

where $\mathbf{B} = (1/2)\mathbf{M}(\mathbf{Q}_p^n)$, and $\mathbf{M}(\mathbf{Q}_p^n)$ denotes matrix representation of quaternion multiplication; $\mathbf{C}_n^c$ and $\mathbf{C}_b^c$ are, respectively, the attitude transfer matrices of the navigation coordinate system n and the carrier coordinate system b to the computational coordinate system c ; ω_{in}^n denotes the projection of the angular velocity of the carrier navigation system into the navigation coordinate system, ω_{ie}^n and ω_{en}^n are the angular velocity of the Earth's rotation and the position angular velocity, $\mathbf{V}^n$ denotes vehicle speed, ∇^b denotes accelerometer error, $\mathbf{f}^b$ is the measured value of the accelerometer, and $\mathbf{M}$ and $\mathbf{N}$ are velocity error and position error coefficient matrix, respectively. $\mathbf{w}(t) = [(-\mathbf{B}\mathbf{C}_b^p \mathbf{w}_g)^T, (\mathbf{C}_b^p \mathbf{w}_a)^T, 0_{1 \times 9}]^T$, where w_g denotes gyro white noise, and w_a denotes accelerometer white noise. The dynamic model of SINS/GPS integration in discrete-time form can be expressed as

$$\mathbf{x}_k = \mathbf{f}(\mathbf{x}_{k-1}) + \mathbf{w}_k. \quad (27)$$

The position and speed output information of SINS is subtracted from the corresponding information of GPS as measurement, and the system measurement equation can be written as

$$\mathbf{z}(t) = \mathbf{h}\mathbf{x}(t) + \mathbf{v}(t), \quad (28)$$

FIGURE 3: The velocity estimation errors of IMMUPF.

where $\mathbf{h}$ is the measurement matrix, $\mathbf{v}(t)$ is the measurement of white noise, and the mean value is zero, then, in discrete-time form, it can be expressed as

$$\mathbf{z}_k = \mathbf{H}_k \mathbf{x}_k + \mathbf{v}_k. \quad (29)$$

And take measurement vector as

$$\mathbf{z}_k = \begin{bmatrix} V_{ES} - V_{EG} & V_{NS} - V_{NG} & V_{US} - V_{UG} \\ L_S - L_G & \lambda_S - \lambda_G & H_S - H_G \end{bmatrix}^T. \quad (30)$$

So,

$$\mathbf{z}_k = \mathbf{H}_k \mathbf{x}_k + \mathbf{v}_k = \begin{bmatrix} \delta \mathbf{V}_S - \delta \mathbf{V}_G \\ \delta \mathbf{P}_S - \delta \mathbf{P}_G \end{bmatrix} = \begin{bmatrix} \mathbf{H}_{V,k} \\ \mathbf{H}_{P,k} \end{bmatrix} \mathbf{x}_k + \begin{bmatrix} \mathbf{v}_{v,k} \\ \mathbf{v}_{p,k} \end{bmatrix}, \quad (31)$$

where

$$\mathbf{H}_k = \begin{bmatrix} 0_{3 \times 4} & I_{3 \times 3} & 0_{3 \times 3} & 0_{3 \times 3} & 0_{3 \times 3} \\ 0_{3 \times 4} & 0_{3 \times 3} & I_{3 \times 3} & 0_{3 \times 3} & 0_{3 \times 3} \end{bmatrix}. \quad (32)$$

And V_S and P_S , respectively, represent the output velocity and position of SINS, V_G and P_G represent the results output velocity and position of GPS, and $\mathbf{v}_{v,k}$ and $\mathbf{v}_{p,k}$ are the measurement noises corresponding to the velocity and position errors of GNSS receiver.

Assume model sets $M = (m_1, m_2, m_3)$, m_1 at constant velocity $Q_1 = Q_0$, $R_1 = R_0$; m_2 at constant acceleration $Q_2 = 5Q_0$, $R_2 = 10R_0$; m_3 at coordinated turn, the noise is high $Q_3 = 25Q_0$, $R_3 = 100R_0$. Consider that the model initial Markov transition probability matrix π_{ij} and initial transfer probability vector $\mathbf{u}$ are chosen as follows,

$$\pi_{ij} = \begin{bmatrix} 0.8 & 0.1 & 0.1 \\ 0.1 & 0.8 & 0.1 \\ 0.1 & 0.1 & 0.8 \end{bmatrix}, \quad \mathbf{u}_{ij} = \begin{bmatrix} 0.8 & 0.1 & 0.1 \end{bmatrix}. \quad (33)$$

The root mean squared errors (RMSE) of the estimation error are computed to show the navigation performance of the comparison methods from a statistical perspective. The RMSE is defined as

$$\text{RMSE} = \sqrt{\frac{1}{M} \sum_{i=1}^M (\hat{\mathbf{x}}_i - \mathbf{x}_{\text{true}})^2}, \quad (34)$$

where M is the Monte Carlo runs. Take the estimated velocity and position error as an example, and the simulation results obtained by the two algorithms are shown in the Figures 3–6.

It can be seen from the velocity error estimation results that the longitude and latitude errors obtained by the IMMUPF algorithm are controlled within $(-0.121, 0.110)$ and $(-0.011, 0.110)$ range, while the height errors are controlled within the $(-0.071, 0.080)$ range after 200 s. The longitude error obtained by the RIMMUPF algorithm is basically controlled within the $(-0.082, 0.071)$ range, the latitude error is basically controlled within the $(-0.071, 0.101)$ range, and the height error is basically controlled within the $(-0.051, 0.053)$ range except for the fluctuation closed to 1200 s. Table 1 describes the RMSEs of the velocity errors by the two algorithms and demonstrates that the RMSE and error range of the proposed RIMMUPF method are much smaller than those of the IMMUPF method.

FIGURE 4: The velocity estimation errors of RIMMUPF.

FIGURE 5: The position estimation errors of IMMUPF.

The position estimation errors simulation results show that the longitude, latitude, and height estimation errors are basically controlled within the $(-20.51, 19.70)$, $(-24.09, 24.01)$, and $(-16.37, 0.88)$ ranges by the IMMUPF algorithm, while the estimation errors are controlled within the $(-18.31, 16.10)$, $(-14.11, 15.01)$, and $(-15.07, 14.85)$

ranges by the RIMMUPF algorithm. Table 2 provides the error comparison for these two methods, showing that the proposed robust method has much smaller RMSE and position estimation error. These simulation results also verify that the proposed RIMMUPF has a stronger ability to resist the unknown dynamic model errors, thus leading to

FIGURE 6: The position estimation errors of RIMMUPF.

TABLE 1: Velocity error comparison for the simulation case (s/m).

Filtering methods	RMSE of δV_E	RMSE of δV_N	RMSE of δV_U	Velocity error range
IMMUPF	0.0587	0.0541	0.0354	$(-0.121, 0.110)$, $(-0.011, 0.110)$, $(-0.071, 0.080)$
RIMMUPF	0.0430	0.0403	0.0227	$(-0.082, 0.071)$, $(-0.071, 0.101)$, $(-0.051, 0.053)$

TABLE 2: Position error comparison for the simulation case (m).

Filtering methods	RMSE of δL	RMSE of $\delta \lambda$	RMSE of δH	Position error range
IMMUPF	7.926	7.847	5.165	$(-20.51, 19.70)$, $(-24.09, 24.01)$, $(-16.37, 0.88)$
RIMMUPF	6.286	4.587	4.097	$(-18.31, 16.10)$, $(-14.11, 15.01)$, $(-15.07, 14.85)$

improved navigation accuracy for the vehicle navigation with SINS/GPS integration.

5. Conclusion

The state estimation and variance calculated by the proposed RIMMUPF can better describe the characteristics of the system model. It can increase the robust adaptive adjustment capability of the system model and restrain the influence of abnormal interference on the state and measurement. By making full use of the effective information, a better particle importance density function is obtained, and the precision of particle solution in multiple model is improved. In the resampling process, the Euclidean distance can be used to adaptively adjust the particles weight in real time according to the specific situation, so as to improve the calculation efficiency. The experiments and comparison analysis demonstrate that the proposed algorithm can effectively improve the filter performance and the calculation precision in

positioning of SINS/GPS integrated navigation system; thus, it provides a new method for nonlinear model filter.

Future work will focus on the improvement of the proposed algorithm to making the performance of algorithm more perfect, and how to reduce the complexity of calculation and improve the speed of the filter.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

The work of this paper was supported by the Natural Science Foundation (China) (Project Number: 61863030) and the

Ningxia Key Research and Development Program (Special Talents) (China) (Project Numbers: 2018BEB04003 and 2019BDE03007).

References

- [1] G. Katalin, K. András, and D. László, "Unscented kalman filters and particle filter methods for nonlinear state estimation," *Procedia Technology*, vol. 12, pp. 65–74, 2014.
- [2] G. Hu, S. Gao, and Y. Zhong, "A derivative UKF for tightly coupled INS/GPS integrated navigation," *ISA Transactions*, vol. 56, pp. 135–144, 2015.
- [3] B. Gao, S. Gao, G. Hu, Y. Zhong, and C. Gu, "Maximum likelihood principle and moving horizon estimation based adaptive unscented Kalman filter," *Aerospace Science and Technology*, vol. 73, pp. 184–196, 2018.
- [4] P. Fearnhead and B. M. Taylor, "An adaptive sequential Monte Carlo sampler," *Bayesian Analysis*, vol. 8, no. 2, pp. 411–438, 2013.
- [5] M. Speekenbrink, "A tutorial on particle filters," *Journal of Mathematical Psychology*, vol. 73, pp. 140–152, 2016.
- [6] J. Candy, "Bootstrap particle filtering," *IEEE Signal Processing Magazine*, vol. 24, no. 4, pp. 73–85, 2007.
- [7] J. Lim, "Particle filtering for nonlinear dynamic state systems with unknown noise statistics," *Nonlinear Dynamics*, vol. 78, no. 2, pp. 1369–1388, 2014.
- [8] R. Vander Merwe and E. Wan, "Gaussian mixture sigma-point particle filters for sequential probabilistic inference in dynamic state-space models," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, vol. 6, pp. 701–704, Brighton, UK, May 2003.
- [9] Y. Wang, D. Zhou, J. Liu et al., "Adaptive Interacting multiple model algorithm for maneuvering target tracking," *Fire Control and Command Control*, vol. 42, no. 3, pp. 137–140, 2017, in Chinese.
- [10] W. Zhou, T. Sun, and M. Chu, "Interacting multiple model particle filter optimization resampling algorithm," *Journal of Bei-Jing University of Aeronautics and Astronautics*, vol. 43, no. 5, pp. 865–871, 2017, in Chinese.
- [11] S. Vasuhi and V. Vaidehi, "Target tracking using interactive multiple model for wireless sensor network," *Information Fusion*, vol. 27, pp. 41–53, 2016.
- [12] D. Xu, S. Cheng, and S. Bao, "Interacting multiple model algorithm based on adaptive transition probability," *Acta Electronica Sinica*, vol. 45, no. 9, pp. 2113–2120, 2017, in Chinese.
- [13] Y. Liu and F. Zhang, "An improved interactive multi model particle filter algorithm," *Machinery & Electronics*, vol. 36, no. 9, pp. 3–6, 2018, in Chinese.
- [14] P. Del Moral, A. Doucet, and A. Jasra, "On adaptive resampling strategies for sequential Monte Carlo methods," *Bernoulli*, vol. 18, no. 1, pp. 252–278, 2012.
- [15] G. Hu, B. Gao, Y. Zhong, and C. Gu, "Unscented kalman filter with process noise covariance estimation for vehicular INS/GPS integration system," *Information Fusion*, vol. 64, pp. 194–204, 2020.
- [16] M. Ades and P. J. Van Leeuwen, "The equivalent-weights particle filter in a high-dimensional system," *Quarterly Journal of the Royal Meteorological Society*, vol. 141, no. 687, pp. 484–503, 2015.
- [17] P. Closas and A. Guillaumon, "Sequential estimation of intrinsic activity and synaptic input in single neurons by particle filtering with optimal importance density," *Journal of Advances in Signal Processing*, vol. 65, pp. 1–22, 2017.
- [18] B. Gao, G. Hu, S. Gao, Y. Zhong, and C. Gu, "Multi-sensor optimal data fusion for INS/GNSS/CNS integration based on unscented kalman filter," *International Journal of Control, Automation and Systems*, vol. 16, no. 1, pp. 129–140, 2018.
- [19] G. Hu, L. Ni, B. Gao, X. Zhu, W. Wang, and Y. Zhong, "Model predictive based unscented kalman filter for hypersonic vehicle navigation with INS/GNSS integration," *IEEE Access*, vol. 8, pp. 4814–4823, 2020.
- [20] Di Yi, Gu Xiao-Hui, L. Fei, and Li Jun-xiang, "An improved UPF-based interacting multiple model target tracking method for dynamic acoustic array," *Control and Decision*, vol. 33, no. 2, pp. 249–255, 2018, in Chinese.
- [21] N. Zheng, Q. Zhang, S. Zhang, and L. Wang, "Improved algorithm based on interacting multiple model and H filtering for IMU/GPS navigation," *Journal of China University of Mining Technology*, vol. 43, no. 5, pp. 920–926, 2014.
- [22] Y. Yao, X. Xu, Li Yao, and T. Zhang, "A hybrid IMM based INS/DVL Integration solution for underwater vehicles," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 6, pp. 1–11, 2019.
- [23] Y. Yang, H. B. He, and G. Xu, "Adaptively robust filtering for kinematic geodetic positioning," *Journal of Geodesy*, vol. 75, no. 2-3, pp. 109–116, 2001.
- [24] J.-W. Zhou, "Classical error theory and robust estimation," *Acta Geoda et Cartographica Sinica*, vol. 18, no. 2, pp. 115–120, 1989.
- [25] Li Xue, S. Gao, and Y. Zhao, "Unscented particle filtering with adaptive adjusted weight and its application in integrated navigation," *Journal of Chinese Inertial Technology*, vol. 20, no. 4, pp. 459–463, 2012.
- [26] G. Hu, W. Wang, Y. Zhong, B. Gao, and C. Gu, "A new direct filtering approach to INS/GNSS integration," *Aerospace Science and Technology*, vol. 77, pp. 755–764, 2018.
- [27] J. C. Fang and X. L. Gong, "Predictive iterated Kalman filter for INS/GPS integration and its application to SAR motion compensation," *IEEE Transactions on Instrumentation and Measurement*, vol. 59, no. 4, pp. 909–915, 2015.