

Research Repository

An Adaptive Radar Surface Vessel Tracking Method with Interacting Multiple Model and Gated Memory

Accepted for publication in Measurement.

Research Repository link: <https://repository.essex.ac.uk/41047/>

Please note:

Changes made as a result of publishing processes such as copy-editing, formatting and page numbers may not be reflected in this version. For the definitive version of this publication, please refer to the published source. You are advised to consult the published version if you wish to cite this paper.

<https://doi.org/10.1016/j.measurement.2025.117798>

An Adaptive Radar Surface Vessel Tracking Method with Interacting Multiple Model and Gated Memory

Xiaoqian Wang^b, Hui Feng^{a,b,*}, Haixiang Xu^{a,b} and Jianhua He^c

^aKey Laboratory of High Performance Ship Technology (Wuhan University of Technology), Ministry of Education, Wuhan, Hubei, 430063, China

^bSchool of Naval Architecture, Ocean and Energy Power Engineering, Wuhan University of Technology, Wuhan, Hubei, 430063, China

^cSchool of Computer Science and Electronic Engineering, University of Essex, CO4 3SQ, United Kingdom

ARTICLE INFO

Keywords:

Radar target tracking
Surface vessel
Interacting multiple model
Kalman filter
Unknown noise

ABSTRACT

Radar is a widely used state measurement unit in intelligent vessel systems and maritime surveillance. However, in new radar tracking scenarios, the statistical characteristics of noise are usually unknown. As a key target tracking method, the Interacting multiple model (IMM) filter tends to experience significantly degraded performance in surface vessel tracking due to the inefficiency of model switching under noise interference. To solve the problem, this paper proposes an adaptive method that can improve the accuracy and robustness of target tracking, by correcting the noise matrix of the rank Kalman filter (RKF) and the transition probability matrix (TPM) with exploitation of historical information in real-time. Then a gated memory mechanism is introduced in the transition probability correction function (TPCF) to accelerate the model jump response by updating and resetting memory adjustments. Numerical simulation and experiment show that the proposed algorithm can effectively reduce the negative impact of unknown noise on IMM and improve target tracking accuracy. Compared with the IMM-RKF algorithm, the proposed algorithm reduces the state estimation errors in position and velocity by 15.1% and 26.3%, respectively.

1. Introduction

Radar is a common sensor for obtaining target position information and widely used in surface vessel tracking scenarios [1, 2]. Surface vessel tracking is an important task for many applications such as perception of navigation environments for intelligent vessels, maritime surveillance and regulation, vessel collision avoidance and efficient navigation[3]. A radar tracking scenario in the Wuhan section of the Yangtze River is illustrated in Fig.1, which includes the target vessel and radar images. The target vessel in Fig.1 is a ferry, which frequently travels between the riverbanks and needs to avoid collisions with other vessels. However, for the tracking scenarios of the maneuvering targets (including ferries, speedboats, tugboats, etc.), there are two issues with radar-based surface vessel tracking methods. 1) The vessels are affected by environmental factors like wind, waves, and currents, while radars can be influenced by rain, snow, electromagnetic interference, and other elements. This leads to unknown statistical noise characteristics in the tracking system at new scenarios, differing from the assumption of known statistical characteristics of Gaussian white noise [4, 5]. 2) Target tracking methods based on Kalman filtering require accurate target motion model, such as constant velocity (CV) model, constant acceleration model, constant turn model [6], etc. In practical surface vessel tracking scenarios, determining the maneuvering intentions and new maneuvers of target vessels is very difficult. Furthermore, as a single model has limited capability of representing the maneuvers of target vessels, it is challenging to estimate future maneuvering situations of the targets. Therefore, using a single model to describe the states of the maneuvering target with unknown intentions will significantly lower tracking accuracy [7].

For the maneuvering target tracking, the multi-model state estimation methods use various models to consider different maneuvering scenarios of the target, thereby maintaining high tracking accuracy even when the target maneuvering intentions are unknown. The multi-model state estimation method, IMM, is widely used due to its high cost-effectiveness [8, 9]. As an important parameter in the IMM, the TPM affects the calculation of model probability and the response time of model jump during the iteration process. The TPM selected based on prior experience is relatively conservative in numerical selection to balance response time and model matching. The adaptive TPM

*corresponding author

 feng@whut.edu.cn (H. Feng)

ORCID(s): 0009-0001-4923-1622 (X. Wang); 0000-0001-6696-3094 (H. Feng)

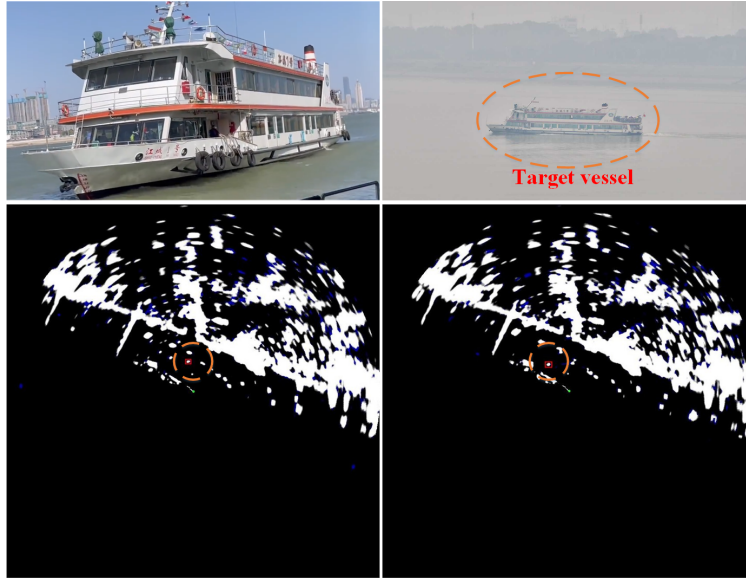


Fig. 1. A radar tracking scenario including the target vessel and radar images (the vessel position marked by the red rectangles).

method adjusts matrix values based on the target maneuvering, enhancing the matching model probability and tracking accuracy. However, the existing adaptive TPM methods in literature are susceptible to noise interference, which reduces the model switching efficiency and tracking accuracy of the IMM tracking algorithm.

To address the above problems of existing radar target tracking methods, this paper proposed a gated memory IMM adaptive RKF (GMIMM-ARKF) surface maneuvering vessel tracking algorithm, in order to accurately track target state under the interference of noise with unknown statistical characteristics in new environments. The contributions of this paper lie in:

- 1) To reduce the adverse effects of process and measurement noise with unknown statistical characteristics, an adaptive RKF method is proposed. This method dynamically adjusts process and measurement noise based on residual sequences in real-time, and ensures algorithm robustness by constraining the interval of the adaptive factor.
- 2) To effectively determine the model jump states under noise interference, a robust jump discriminator based on the model probability historical information is defined. It calculates the mean of the historical model probabilities to restore each model probabilities, and then determines whether the current matching model undergoes switching by the ratio of the model probability means.
- 3) By incorporating a gated memory mechanism into the TPCF, memory is updated during non-model jumping phases to maintain higher probabilities for the matching models. During model jumping phases, memory is quickly reset to expedite the model jumps, facilitating the rapid elevation of new matching model probabilities. This mechanism can enhance the tracking accuracy and stability.

The paper is structured as follows. Section 2 introduces existing research on the noise adaptation and the IMM methods. Section 3 introduces the tracking system model and reviews the RKF and the IMM algorithms. An adaptive IMM RKF with gated memory is proposed in Section 4. In Section 5, numerical simulation and experiment results are presented, validating the effectiveness of the proposed algorithm. The research conclusions of this paper are given in Section 6.

2. Related Work

In terms of unknown noise, the Sage-Husa adaptive Kalman filter is a commonly filtering method for noise adaptation [10]. It recursively updates noise covariance through maximum likelihood estimation, enabling adaptive adjustment of both process noise and measurement noise. This approach features simple structure and low computational load, but exhibits high sensitivity to initial noise settings. The innovation-based noise adaptation method performs rapid noise tracking using residual statistics, demonstrating excellent performance in dynamic environments, though

remaining highly sensitive to outliers [11]. The variational Bayesian-based noise adaptation method jointly estimates states and noise under non-Gaussian assumptions, providing relatively accurate noise computation. But, it suffers from structural complexity and high computational cost [12]. However, in practical radar tracking scenarios, the system is often nonlinear, presenting a challenge for traditional linear Kalman filter. In [13], the data-driven estimation results were used for the discrete time Kalman fusion to reduce estimation bias caused by insufficient model accuracy. The Extended Kalman Filter (EKF) linearizes nonlinear functions through first-order Taylor expansion and calculates the Jacobian matrix at the mean point to approximate state transitions and observation equations. This method has a simple structure and is easy to implement [14]. In [15], the positioning performance and robustness of the unmanned aerial vehicle are enhanced by adapting the noise matrix of the EKF algorithm using historical measurements. To address measurement anomalies and state disturbances caused by ship oscillation, Fan et al. introduced a strong tracking factor based on the EKF algorithm and employed a combination of unbiased and biased estimations of noise for noise adaptation [4]. Compared to the EKF algorithm, which incurs first-order truncation errors, nonlinear filtering methods based on deterministic sampling can avoid computing Jacobian matrices. They approximate the probability density function using deterministic sample points, thus achieving higher filtering accuracy [16]. The filtering methods based on deterministic sampling include the unscented Kalman filter (UKF) [17], cubature Kalman filter (CKF) [18], RKF [19] et al. UKF employs the unscented transform to propagate statistical properties through a deterministic set of $2n+1$ sigma points (where n is the state dimension). This derivative-free approach eliminates matrix inversion operations, significantly enhancing numerical stability compared to traditional linearization-based methods. Zhang et al. designed an exponential attenuation factor by prioritizing the latest data and used this factor to adaptively adjust the noise matrix of the UKF for vehicle state estimation [20]. CKF is a deterministic sampling method based on the Spherical Radial Cubature Rule, demonstrating superior filtering accuracy over UKF when handling high-dimensional state estimation problems. In [21], to address the information with random disturbances, the Sage-Husa noise estimator is incorporated into the CKF algorithm, and the state variables are augmented to estimate the multiplicative noise in the system. Zhang et al. use the Chi-square distribution statistical functions to detect the abnormal noise and adjust the process and measurement noise of the square-root CKF through a weighted adaptive approach based on a moving window [22]. Fang et al. designed a noise statistical estimator to detect time-varying and unknown noise information, and combined it with the CKF algorithm to estimate the state of maneuvering targets [23]. The RKF algorithm approximates the probability distribution of the system state through rank sampling. Although structurally similar to UKF and CKF, its main difference lies in the sampling process and the presence of more sampling points. This method achieves high filtering accuracy for nonlinear state estimation problems by Gaussian and non Gaussian distributions, and exhibits better performance compared to UKF and CKF [24, 25].

Multiple model filtering is highly effective for maneuvering target tracking [7]. The generalized pseudo-Bayesian algorithm uses branching and pruning to manage models, providing theoretical rigor but with exponential computational growth. The variable structure multiple model techniques activate/deactivate models based on system states, offering high efficiency and strong anomaly handling, though with some detection delay. In contrast, the IMM method uses Markov transition probabilities for model fusion, ensuring linear computational growth and robust adaptation. Ebrahimi et al. introduced a second-order Markov model into the IMM algorithm to enhance the accuracy of bearing-only maneuvering target tracking system [26]. Samuel et al. handled the mixed noise maneuvering target tracking problem under the influence of both white and colored noises, proposing an IMM Filtering algorithm with mixed system noises [27]. In [28], the tracking capability of the system under uncertainty is enhanced by weighting the output results of filters based on adaptive fading factor and maximum correntropy. To enhance the positioning accuracy of the autonomous underwater vehicle suffering from sensor outliers, Zhang et al. use an outlier detector to determine whether the IMM is affected [29]. When the outliers are present, it generates a pseudo value as the state estimation result to enhance the algorithm robustness. Nevertheless, the TPM is a priori known and fixed in the traditional IMM. During the tracking process, conservative and inaccurate the TPM can lead to significant state estimation errors. Some IMM methods based on the adaptive TPM have attracted attention. Cosme et al. introduced the retention model concept into the IMM, and the adaptive updating of the TPM is determined by Chi-square statistics [30]. Lee et al. defined the polarizing TPCF and the activating TPCF for the TPM, which are respectively used to enhance matching model probability and expedite model probability updates, thereby improving the tracking accuracy [31]. As the IMM algorithm primarily updates model probabilities based on measurement residuals, both the calculation of model probabilities and the adaptation of the TPM are susceptible to interference from measurement noise, thereby diminishing the algorithm performance. Xie et al. proposed a parallel IMM (PIMM) method with the adaptive TPM, which uses the model historical information to construct the correction functions aimed at enhancing its noise

resilience. Additionally, the PIMM algorithm defines model jump thresholds to determine model jump states, thereby switching between the adaptive IMM and the common IMM to reduce model jump response time and estimation errors [32]. Li et al. similarly, based on the principle of the PIMM, proposed an adaptive IMM method using the EKF and the UKF frameworks to track underwater maneuvering targets with one-step randomly delayed measurements [33]. Although the PIMM algorithm demonstrates excellent performance, the model jumping threshold calculated based on the likelihood ratio is susceptible to measurement noise. Therefore, in the case of significant noise, the PIMM method is difficult to accurately determine whether the model jumps, which results in the algorithm frequently switching to the common IMM. Second, due to the PIMM method requiring the concurrent maintenance of two IMM operations, its calculation cost is higher compared to the traditional IMM. Furthermore, whether the TPM of the common IMM is used depend on the selection of this threshold, which notably affect the model jump response time and tracking accuracy.

In the above mentioned research works, most noise adaptive methods assume that the process and measurement noise are Gaussian distributions. However, the real working environment of the surface vessel tracking system is dynamic, and the system is often subject to external environmental interference. Therefore, the statistical characteristics of noise in real environments are usually unknown, requiring algorithms to handle both Gaussian and non-Gaussian noise. Regarding adaptive TPM, existing research does not accurately determine the model jump under noise interference, and the response speed after the model jump is relatively slow, which negatively impacts the target tracking. Therefore, an IMM-based target tracking approach capable of operating in unknown noise environments while ensuring rapid and accurate model switching is required.

3. Preliminaries

3.1. Tracking system model

Assuming the target is maneuvering on a two-dimensional coordinate system with the x-axis as due east and the y-axis as due north. The target motion state is represented by position and velocity, and the state vector at time-step k is denoted as $\mathbf{x}_k = [x_k, \dot{x}_k, y_k, \dot{y}_k]^T$. The measurements obtained by the tracking system are distance and orientation, and the measurement vector is $\mathbf{z}_k = [d_k, \theta_k]^T$. Consider the nonlinear target tracking system, the state-space model and the measurement model are given by [33]

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

$$\mathbf{z}_k = h(\mathbf{x}_k, \rho_k) + \mathbf{v}_k(\rho_k) \quad (2)$$

where ρ_k is real target model, and $\rho_k \in \{1, 2, \dots, L\}$. $f(\cdot)$ and $h(\cdot)$ are the dynamic equation and the nonlinear measurement equation, respectively. The process noise $\mathbf{w}_k(\rho_k)$ satisfies $N(0, \mathbf{Q}_k)$, the measurement noise $\mathbf{v}_k(\rho_k)$ satisfies $N(0, \mathbf{R}_k)$. Furthermore, it is assumed that $\mathbf{w}_k(\rho_k)$ and $\mathbf{v}_k(\rho_k)$ are uncorrelated.

3.2. Review of the RKF algorithm

The key of the RKF lies in approximating nonlinear functions using rank sigma points and propagating them during the filtering process to estimate the state and the error covariance. Assuming the state vector mean $\bar{\mathbf{x}}$ is and the error covariance is $\mathbf{P}$, the rank sigma points can be given by

$$\chi_s = \bar{\mathbf{x}} + r_t \eta_{p_t} (\sqrt{\mathbf{P}})_l \quad (3)$$

where χ_s is the state of the s -th rank sigma point. r_t is the correction coefficient of the rank sigma point, and when the sampling point is symmetrically distributed, $r_t = 1$. The value of t satisfies $t = 1, 2, \dots, 2m + 1$ and $t \neq m + 1$, m is the number of sampling layers. p_t is the distribution probability corresponding to the rank sigma point, usually calculated using median rank or average rank. η_{p_t} is the one-dimensional standard distribution quantile for the state $\mathbf{x}$ and it satisfies the probability p_t . This distribution can take various forms, such as a normal distribution, t-distribution, and so on. $(\sqrt{\mathbf{P}})_l$ is a vector of the l -th column of $\sqrt{\mathbf{P}}$, where the range of l is $\{1, 2, \dots, n\}$, and n is the dimension of the state $\mathbf{x}$. The number of rank sigma points is $2mn$, and the values of s can be defined as

$$\begin{cases} s = (t - 1)n + l, t = 1, 2, \dots, m \\ s = (t - 2)n + l, t = m + 2, m + 3, \dots, 2m + 1 \end{cases} \quad (4)$$

Considering the computation cost of the filtering algorithm, the sampling level is assumed to be $m = 2$. On this condition, the sampling strategy of the symmetrically distributed RKF will be discussed. The rank sigma point set at time-step $k - 1$ is given by

$$\chi_{k-1,s} = \begin{cases} \hat{\mathbf{x}}_{k-1} + r_1 \eta_{p_1} (\sqrt{\mathbf{P}_{k-1}})_s, s = 1, \dots, n \\ \hat{\mathbf{x}}_{k-1} - r_1 \eta_{p_1} (\sqrt{\mathbf{P}_{k-1}})_{s-n}, s = n+1, \dots, 2n \\ \hat{\mathbf{x}}_{k-1} + r_2 \eta_{p_2} (\sqrt{\mathbf{P}_{k-1}})_{s-2n}, s = 2n+1, \dots, 3n \\ \hat{\mathbf{x}}_{k-1} - r_2 \eta_{p_2} (\sqrt{\mathbf{P}_{k-1}})_{s-3n}, s = 3n+1, \dots, 4n \end{cases} \quad (5)$$

The probability p_i takes the median rank as $p_i = (t+2.7)/(2m+1.4)$ and takes the average rank as $p_i = j/(2m+2)$. Therefore, when the number of the sampling layers $m = 2$, the values of p_i and η_{p_i} solved by the median rank calculation method are $p_1 = 0.6852$, $p_2 = 0.8704$, $\eta_{p_1} = 0.4823$ and $\eta_{p_2} = 1.1281$.

The implementation process of the RKF can be briefly summarized by the following two steps.

Step 1) Time update

- Calculating the square-root of the posteriori covariance $\mathbf{P}_{k-1}$ at time-step $k - 1$,

$$\mathbf{P}_{k-1} = \sqrt{\mathbf{P}_{k-1}} \left(\sqrt{\mathbf{P}_{k-1}} \right)^T \quad (6)$$

- Construct the rank sigma points by (5),
- Propagating the rank sigma points,

$$\chi_{k|k-1,s}^* = \mathbf{f} \left(\chi_{k-1,s} \right), s = 1, 2, \dots, 4n \quad (7)$$

- The prior state estimation $\hat{\mathbf{x}}_{k|k-1}$ and the prior covariance $\mathbf{P}_{k|k-1}$ are given by

$$\hat{\mathbf{x}}_{k|k-1} = \frac{1}{4n} \sum_{s=1}^{4n} \chi_{k|k-1,s}^* \quad (8)$$

$$\mathbf{P}_{k|k-1} = \frac{1}{\omega_r} \sum_{s=1}^{4n} \left(\chi_{k|k-1,s}^* - \hat{\mathbf{x}}_{k|k-1} \right) \left(\chi_{k|k-1,s}^* - \hat{\mathbf{x}}_{k|k-1} \right)^T + \mathbf{Q}_k \quad (9)$$

where the calculation method for the covariance weighting coefficients ω_r is described as follows

$$\omega_r = 2r^2(\eta_{p_1}^2 + \eta_{p_2}^2) \quad (10)$$

Step 2) Measurement update

- Resampling the rank sigma points,

$$\chi_{k|k-1,s} = \begin{cases} \hat{\mathbf{x}}_{k|k-1} + r_1 \eta_{p_1} (\sqrt{\mathbf{P}_{k|k-1}})_s, s = 1, \dots, n \\ \hat{\mathbf{x}}_{k|k-1} - r_1 \eta_{p_1} (\sqrt{\mathbf{P}_{k|k-1}})_{s-n}, s = n+1, \dots, 2n \\ \hat{\mathbf{x}}_{k|k-1} + r_2 \eta_{p_2} (\sqrt{\mathbf{P}_{k|k-1}})_{s-2n}, s = 2n+1, \dots, 3n \\ \hat{\mathbf{x}}_{k|k-1} - r_2 \eta_{p_2} (\sqrt{\mathbf{P}_{k|k-1}})_{s-3n}, s = 3n+1, \dots, 4n \end{cases} \quad (11)$$

- Calculating the rank sigma points of the measurement,

$$\mathbf{Z}_{k,s} = \mathbf{h} \left(\chi_{k|k-1,s} \right) \quad (12)$$

- The prior measurement estimation $\mathbf{Z}_{k,s}$, the innovation covariance $\mathbf{P}_{zz,k}$ and the cross-covariance $\mathbf{P}_{xz,k}$ are calculated by

$$\hat{\mathbf{z}}_{k|k-1} = \frac{1}{4n} \sum_{s=1}^{4n} \mathbf{Z}_{k,s} \quad (13)$$

$$\mathbf{P}_{zz,k} = \frac{1}{\omega_r} \sum_{s=1}^{4n} \left(\mathbf{Z}_{k,s} - \hat{\mathbf{z}}_{k|k-1} \right) \left(\mathbf{Z}_{k,s} - \hat{\mathbf{z}}_{k|k-1} \right)^T + \mathbf{R}_k \quad (14)$$

$$\mathbf{P}_{xz,k} = \frac{1}{\omega_r} \sum_{s=1}^{4n} \left(\chi_{k|k-1,s} - \hat{\mathbf{x}}_{k|k-1} \right) \left(\mathbf{Z}_{k,s} - \hat{\mathbf{z}}_{k|k-1} \right)^T \quad (15)$$

- Calculating the gain matrix,

$$\mathbf{K}_k = \mathbf{P}_{xz,k} \mathbf{P}_{zz,k}^{-1} \quad (16)$$

- Calculating the posteriori state estimation $\hat{\mathbf{x}}_k$ and the posteriori covariance $\mathbf{P}_k$,

$$\hat{\mathbf{x}}_k = \hat{\mathbf{x}}_{k|k-1} + \mathbf{K}_k \left(\mathbf{z}_k - \hat{\mathbf{z}}_{k|k-1} \right) \quad (17)$$

$$\mathbf{P}_k = \mathbf{P}_{k|k-1} + \mathbf{K}_k \mathbf{P}_{zz,k} \mathbf{K}_k^T \quad (18)$$

3.3. Stochastic Stability Analysis of the Rank Kalman Filter

3.3.1. Preliminaries and Assumptions

Consider the nonlinear tracking system defined in Section 3.1. The RKF has the same structure as the UKF, differing only in the sampling strategy where RKF employs rank-based deterministic sampling points instead of sigma points. Building upon the established convergence proof framework for UKF [34], we extend the stochastic stability analysis to RKF by demonstrating that its rank-sampling scheme preserves all essential properties required for bounded estimation error and covariance convergence. To analyze the stochastic stability of the RKF algorithm, we make the following technical assumptions:

Assumptions 1.

The nonlinear functions $f()$ and $h()$ are Lipschitz continuous with constants $L_f, L_h > 0$. Jacobian matrices $\mathbf{F}_k = \frac{\partial \mathbf{f}}{\partial \mathbf{x}} \Big|_{\hat{\mathbf{x}}_k}$ and $\mathbf{H}_k = \frac{\partial \mathbf{h}}{\partial \mathbf{x}} \Big|_{\hat{\mathbf{x}}_{k+1|k}}$ satisfy $\|\mathbf{F}_k\| \leq \bar{\mathbf{f}}$ and $\|\mathbf{H}_k\| \leq \bar{\mathbf{h}}$. Covariance bounds satisfy $q\mathbf{I} \leq \mathbf{Q}_k \leq \bar{q}\mathbf{I}$, $r_c\mathbf{I} \leq \mathbf{R}_k \leq \bar{r}_c\mathbf{I}$.

The $4n$ rank-sampling points $\{\chi_{k,i}\}_{i=1}^{4n}$ satisfy:

$$\frac{1}{4n} \sum_{i=1}^{4n} \chi_{k,i} = \hat{\mathbf{x}}_k, \quad \frac{r_c}{\omega_r} \sum_{i=1}^{4n} (\chi_{k,i} - \hat{\mathbf{x}}_k)(\chi_{k,i} - \hat{\mathbf{x}}_k)^T = \mathbf{P}_k \quad (19)$$

3.3.2. Mean-Square Boundedness Analysis

Theorem 1 (Error Boundedness).

If there exist constants $\alpha_c, \beta_c > 0$ such that $\alpha_c\mathbf{I} \leq \mathbf{P}_k \leq \beta_c\mathbf{I}$, then the estimation error $\tilde{\mathbf{x}}_k = \mathbf{x}_k - \hat{\mathbf{x}}_k$ satisfies:

$$\limsup_{k \rightarrow \infty} E \left[\|\tilde{\mathbf{x}}_k\|^2 \right] \leq C \quad (20)$$

where $C = \frac{\mu_c}{\lambda_c}$ with:

$$\lambda_c = \gamma \kappa \cdot \frac{q}{\beta_c L_f^2 + \bar{q}} \quad (21)$$

$$\mu_c = \text{tr}(\mathbf{R}_k \mathbf{H}_k^T \mathbf{H}_k) \quad (22)$$

$$\kappa = \frac{\lambda_{\min}(\mathbf{H}_k^T \mathbf{R}_k^{-1} \mathbf{H}_k)}{\lambda_{\max}(\mathbf{P}_{k+1|k}^{-1})} \quad (23)$$

Proof. Boundedness of the estimation error for the RKF.

1) Define Lyapunov function. Then $\mathbf{V}_k = \tilde{\mathbf{x}}_k^T \mathbf{P}_k^{-1} \tilde{\mathbf{x}}_k$. From Assumptions 1, $\mathbf{P}_k$ is positive definite.

2) Prediction Step.

The prediction error covariance satisfies:

$$\mathbf{P}_{k+1|k} \leq L_f^2 \mathbf{P}_k + \mathbf{Q}_k \leq (\beta_c L_f^2 + \bar{q}) \mathbf{I} \quad (24)$$

3) Update Step.

When $\gamma_k = 1$, the Kalman gain $\mathbf{K}_{k+1}$ reduces the covariance:

$$\mathbf{P}_{k+1} = \mathbf{P}_{k+1|k} - \mathbf{K}_{k+1} \mathbf{P}_{k+1|k} \mathbf{K}_{k+1}^T \leq (1 - \gamma_k) \mathbf{P}_{k+1|k} \quad (25)$$

4) Stochastic Stability.

Taking expectations over γ_k :

$$E[\mathbf{V}_{k+1}|F_k] \leq (1 - \lambda_c) \mathbf{V}_k + \mu_c \quad (26)$$

By the stochastic Lyapunov lemma [34], $\tilde{\mathbf{x}}_k$ is mean-square bounded.

3.3.3. Covariance Convergence

Theorem 2 (Bounded Covariance).

If Assumptions 1 hold and the observation probability γ exceeds:

$$\gamma > \gamma^* = \frac{L_f^2 \omega_r^{-1} - 1}{L_f^2 \phi_c^2} \quad (27)$$

where ϕ_c is the rank-sampling linearization error constant, then:

$$\limsup_{k \rightarrow \infty} E[\mathbf{P}_k] \leq \bar{\mathbf{P}} \quad (28)$$

where $\bar{\mathbf{P}}$ being the solution to:

$$\bar{\mathbf{P}} = (1 - \gamma \phi_c^2) L_f^2 \bar{\mathbf{P}} L_f^T + \gamma \phi_c^2 \cdot \frac{\bar{r}_c}{h^2 \beta_c^2} \mathbf{I} + \bar{q} \mathbf{I} \quad (29)$$

Proof. Boundedness of the error covariance matrices.

1) Induction Basis.

Assume $E[\mathbf{P}_k] \leq \bar{\rho} \mathbf{I}$. Then:

$$E[\mathbf{P}_{k+1|k}] \leq L_f^2 \bar{\rho} \mathbf{I} + \bar{q} \mathbf{I} \quad (30)$$

2) Critical Probability Condition.

For $\gamma > \gamma^*$, the matrix series converges to $\bar{\mathbf{P}}$:

$$\bar{\mathbf{P}} = (\mathbf{I} - (1 - \gamma \phi_c^2) L_f^2)^{-1} (\gamma \phi_c^2 \cdot \frac{\bar{r}_c}{h^2 \beta_c^2} \mathbf{I} + \bar{q} \mathbf{I}) \quad (31)$$

Thus, $E[\mathbf{P}_k]$ is uniformly bounded.

3.4. Review of the IMM method

The IMM is composed of the multiple sub-filters, which fuse the state estimation results of the sub-filters using the model probabilities to obtain the current target state. Assuming the IMM method incorporates L models, with each model utilized as a filter for specific types of motion. The probability of model j at time-step $k - 1$ can be defined as μ_{k-1}^j ($j = 1, 2, \dots, L$), and $\boldsymbol{\mu}_k$ contains all the model probabilities at time-step k . The transition probability $\pi_k^{i,j}$ and the TPM are described as follows:

$$\pi_k^{i,j} = p\{\rho_k = j | \rho_{k-1} = i\} \quad (32)$$

$$\boldsymbol{\Pi}_k = [\pi_k^{i,j}]_{M \times M} \quad (33)$$

where $\pi_k^{i,j}$ represents the transition probability from model i at time-step $k-1$ to model j at time-step k , and it satisfies

$$\sum_{j=1}^M \pi_k^{i,j} = 1, 0 \leq \pi_k^{i,j} \leq 1 \quad (34)$$

The iterative steps of the IMM method are summarized by the following four steps.

Step 1) Model interaction

- Calculate the mixing model probability $\mu_{k-1}^{i|j}$ based on the TPM,

$$u_{k-1}^{i|j} = \frac{\pi_k^{i,j} u_{k-1}^i}{\bar{c}_{k-1}^j} \quad (35)$$

where the calculation method for the normalization constant $\bar{c}_{k-1}^j$ is given by

$$\bar{c}_{k-1}^j = \sum_{i=1}^L \pi_k^{i,j} u_{k-1}^i \quad (36)$$

- The state $\bar{\mathbf{x}}_{k-1}^j$ and the error covariance $\bar{\mathbf{P}}_{k-1}^j$ of the sub-filter j are calculated by

$$\bar{\mathbf{x}}_{k-1}^j = \sum_{i=1}^L \hat{\mathbf{x}}_{k-1}^i u_{k-1}^{i|j} \quad (37)$$

$$\bar{\mathbf{P}}_{k-1}^j = \sum_{i=1}^L u_{k-1}^{i|j} \{ \mathbf{P}_{k-1}^i + (\hat{\mathbf{x}}_{k-1}^i - \bar{\mathbf{x}}_{k-1}^j)(\hat{\mathbf{x}}_{k-1}^i - \bar{\mathbf{x}}_{k-1}^j)^T \} \quad (38)$$

Step 2) State estimation

- Estimate the target state separately using the sub-filters based on L models. The target state vector $\hat{\mathbf{x}}_k^j$, the error covariance $\mathbf{P}_k^j$, and the innovation covariance $\mathbf{P}_{zz,k}^j$ based on model j are calculated.

Step 3) Model probability update

- The likelihood of model j at time-step k is given by

$$\begin{aligned} \Lambda_k^j &= N[\boldsymbol{\epsilon}_k^j; 0, \mathbf{P}_{zz,k}^j] \\ &= \frac{1}{\sqrt{2\pi} |\mathbf{P}_{zz,k}^j|} \exp \left\{ -\frac{1}{2} \boldsymbol{\epsilon}_k^{j,T} (\mathbf{P}_{zz,k}^j)^{-1} \boldsymbol{\epsilon}_k^j \right\} \end{aligned} \quad (39)$$

where the measuring residuals $\boldsymbol{\epsilon}_k^j$ is calculated by

$$\boldsymbol{\epsilon}_k^j = \mathbf{z}_k - \hat{\mathbf{z}}_{k|k-1}^j \quad (40)$$

- Subsequently, update the model probabilities u_k^j ,

$$u_k^j = \frac{\Lambda_k^j \bar{c}_{k-1}^j}{\sum_{j=1}^L \Lambda_k^j \bar{c}_{k-1}^j} \quad (41)$$

Step 4) Estimation Fusion The outputs of L filters are weighted by the updated model probabilities. The target state $\hat{\mathbf{x}}_k^*$ and the error covariance $\mathbf{P}_k^*$ are calculated as

$$\hat{\mathbf{x}}_k^* = \sum_{j=1}^L \hat{\mathbf{x}}_k^j u_k^j \quad (42)$$

$$\mathbf{P}_k^* = \sum_{j=1}^L u_k^j \{ \mathbf{P}_k^j + (\hat{\mathbf{x}}_k^j - \hat{\mathbf{x}}_k^*)(\hat{\mathbf{x}}_k^j - \hat{\mathbf{x}}_k^*)^T \} \quad (43)$$

4. Proposed Method

4.1. Noise adaptive RKF algorithm

To overcome the problem of unknown noise in nonlinear target tracking systems, this section presents a novel noise adaptive RKF (ARKF) method based on a moving window of measurement residuals. Adaptive correction during the tracking process is an effective method for unknown process noise and measurement noise. By considering residual sequences over multiple time-steps using a moving window approach, adaptive adjustment of process noise $\hat{\mathbf{Q}}_k$ and measurement noise $\hat{\mathbf{R}}_k$ is achieved, thus enhancing the accuracy and reliability of the target tracking.

Firstly, the measurement residual is calculated by (27).

According to the moving window method, the innovation covariance base on the historical information $\mathbf{P}_{ee,k}$ can be defined as

$$\mathbf{P}_{ee,k} = \frac{1}{M} \sum_{c=1}^{M-1} \boldsymbol{\epsilon}_{k-c} \boldsymbol{\epsilon}_{k-c}^T = \mathbf{R}_k - \tilde{\mathbf{P}}_{zz,k} \quad (44)$$

where the measurement covariance $\tilde{\mathbf{P}}_{zz,k}$ is calculated as

$$\tilde{\mathbf{P}}_{zz,k} = \frac{1}{\omega_r} \sum_{s=1}^{4n} (\mathbf{Z}_{k,s} - \hat{\mathbf{z}}_{k|k-1}) (\mathbf{Z}_{k,s} - \hat{\mathbf{z}}_{k|k-1})^T \quad (45)$$

Substituting (32) into (31)

$$\mathbf{R}_k = \mathbf{P}_{ee,k} + \tilde{\mathbf{P}}_{zz,k} \quad (46)$$

Combining the measurement noise $\hat{\mathbf{R}}_{k-1}$ at time-step $k-1$, the adaptive measurement noise $\hat{\mathbf{R}}_k$ is given by

$$\hat{\mathbf{R}}_k = (1 - \beta_k) \hat{\mathbf{R}}_{k-1} + \beta_k (\mathbf{P}_{ee,k} + \tilde{\mathbf{P}}_{zz,k}) \quad (47)$$

where $\beta_k = (1 - b)/(1 - b^k)$; the forgetting factor b satisfy $0 < b < 1$, which is usually 0.95 [10].

Assuming α_k is the adaptive factor for process noise, its calculation method is given by [35]

$$\alpha_k = \frac{|\text{tr}(\mathbf{N}_k)|}{\text{tr}(\mathbf{M}_k)} \quad (48)$$

where

$$\begin{aligned} \text{tr}(\mathbf{N}_k) = & \text{tr}(\hat{\mathbf{R}}_k) + \text{tr}(\mathbf{P}_{xz,k}^T \mathbf{P}_{k|k-1}^{-1} \tilde{\mathbf{P}}_{k|k-1} \mathbf{P}_{xz,k} \mathbf{P}_{k|k-1}^{-1}) \\ & - \text{tr}(\boldsymbol{\epsilon}_k \boldsymbol{\epsilon}_k^T) - \text{tr}(\mathbf{P}_{xz,k}^T \mathbf{P}_{k|k-1}^{-1} \mathbf{K}_k \mathbf{P}_{zz,k} \mathbf{K}_k^T \mathbf{P}_{xz,k} \mathbf{P}_{k|k-1}^{-1}) \end{aligned} \quad (49)$$

$$\text{tr}(\mathbf{M}_k) = \text{tr}(\mathbf{P}_{xz,k}^T \mathbf{P}_{k|k-1}^{-1} \hat{\mathbf{Q}}_{k-1} \mathbf{P}_{xz,k} \mathbf{P}_{k|k-1}^{-1}) \quad (50)$$

Since (36) includes the trace of the measurement residual, when noise causes dramatic change in the measurement residual, the value of α_k will also undergo significant change. In order to bolster the robustness of the adaptive filter, the size of is constrained by upper and lower bounds. Hence, the adaptive process noise $\hat{\mathbf{Q}}_k$ is calculated as

$$\hat{\mathbf{Q}}_k = \bar{\alpha}_k \hat{\mathbf{Q}}_{k-1} \quad (51)$$

$$\begin{cases} \bar{\alpha}_k = \alpha_k, & \text{if } \psi_{\min} \leq \alpha_k \leq \psi_{\max} \\ \bar{\alpha}_k = \psi_{\max}, & \text{if } \alpha_k < \psi_{\min} \text{ or } \alpha_k > \psi_{\max} \end{cases} \quad (52)$$

where $\psi_{\max}$ and $\psi_{\min}$ are upper bound and lower bound of the adaptive factor α_k .

The brief calculation process of the ARKF algorithm comprises the following five steps.

Step 1) Initialization: The initial values of the target state $\hat{\mathbf{x}}_0$, the error covariance $\mathbf{P}_0$, the process noise $\mathbf{Q}_0$, and the measurement noise $\mathbf{R}_0$ are set.

Step 2) Time update: Calculating the square-root of the error covariance matrix is solved by (6). Calculate and propagate the rank sigma points using (5) and (7). Subsequently, the prior state estimation $\hat{\mathbf{x}}_{k|k-1}$ and the prior covariance $\mathbf{P}_{k|k-1}$ are calculated according to (8) and (9), respectively.

Step 3) Measurement update: Rank sigma points are resampled and propagated by (11) and (12). The prior measurement estimation $\mathbf{Z}_{k,s}$, the innovation covariance $\mathbf{P}_{zz,k}$ and the cross-covariance $\mathbf{P}_{xz,k}$ are calculated according to (13)-(15) sequentially. The gain matrix $\mathbf{K}_k$, the posteriori state estimation $\hat{\mathbf{x}}_k$ and the posteriori covariance $\mathbf{P}_k$ are updated using (16)-(18).

Step 4) Measurement noise adaption: Calculate the measurement residual ϵ_k using (27). The adaptive measurement noise $\hat{\mathbf{R}}_k$ updated by (31)-(34), and $\hat{\mathbf{R}}_k$ is substituted into (14) as the new measurement noise.

Step 5) Process noise adaption: The adaptive factor α_k and the adaptive process noise $\hat{\mathbf{Q}}_k$ are calculated according to (35)-(39), $\hat{\mathbf{Q}}_k$ is substituted into (9) to update the process noise.

4.2. Adaptive TPM method based on historical information

In the process of tracking surface target, using the IMM algorithm with a fixed TPM to track maneuvering target is relatively conservative. The adaptive TPM method is able to maintain a high probability for the matching model during non-jumping phases, thus enhancing tracking accuracy. During jumping phases, the algorithm switches to new matching models and adjusts the model probabilities.

Constructing a TPCF in the IMM algorithm is an efficient adaptive TPM method. According to (26)-(28), the IMM adjusts model probabilities by introducing the measurement information. This approach ensures that the probabilities of each model are consistent with the actual maneuvering situation. Nevertheless, the measurement information contains some measurement noise. Although the methods of the adaptive measurement noise reduce the negative impact of unknown noise, it still interferes with the calculation of the model probabilities. Due to the inertia of vessels during maneuvering, their historical model probabilities include the target maneuvering trends. This section will extract this trend from the model probabilities affected by noise, and then introduce it into the TPCF to guide the adaptive TPM. The TPCF is defined as follows [31]

$$g^j = \frac{1}{1 - \Delta\mu_k^{j,N}} \quad (53)$$

where N is the step number that retain historical increments, $\Delta\mu_k^{j,N}$ is the historical increment of model j , which includes the model probability historical information from time-step $k - N$ to time-step k , and its calculation method is given by

$$\Delta\mu_k^{j,N} = \frac{1}{N} \sum_{c=0}^{N-1} (\mu_{k-c}^j - \mu_{k-c-1}^j) \quad (54)$$

Update the TPM and normalize by the TPCF,

$$\pi_k^{*i,j} = g^j * \pi_{k-1}^{i,j} \quad (55)$$

$$\pi_k^{i,j} = \frac{\pi_k^{*i,j}}{\sum_{j=1}^L \pi_k^{*i,j}} \quad (56)$$

After the TPM correction and normalization, it still satisfies (21). It can be inferred from (41) that when $\Delta\mu_k^{j,N} > 0$, the historical probability of model presents an increasing trend. If for the model, the probability of the non-matching models will increase until a model jump occurs. The relationship between g^j and $\pi_k^{i,j}$ is as follows:

$$\begin{cases} \Delta\mu_k^{j,N} > 0, g^j > 1, \pi_k^{i,j} > \pi_{k-1}^{i,j} \\ \Delta\mu_k^{j,N} = 0, g^j = 1, \pi_k^{i,j} = \pi_{k-1}^{i,j} \\ \Delta\mu_k^{j,N} < 0, g^j < 1, \pi_k^{i,j} < \pi_{k-1}^{i,j} \end{cases} \quad (57)$$

The TPCF adjusts the TPM through the above process, which enhances the matching model probability based on the model historical information and improves the anti-noise capability of the target tracking algorithm.

The AIMM based on the model probability historical information can enhance the stability of the model probabilities when models remain unchanged, thereby enabling the matching model j to dominate. However, during target maneuvering process, the system model may experience instantaneous jumps, and this jump information could be obscured by noise, making it difficult to switch matching models quickly. This issue results in a period after the model switch where the model probabilities do not match the actual maneuver. To accurately determine whether the model jumps under the noise interference and promptly adjust the TPCF to better match the current maneuvering state, this section proposes an adaptive TPM method based on gated memory. Firstly, a robust jump discriminator is designed to detect model jumps under the noise interference. Next, the gated memory mechanism is introduced into the adaptive TPM, controlling the memory of the model historical probabilities through gate updates and resets. This approach enables flexible adjustment of the historical information contained in the TPCF.

4.3. Robust jump discriminator

Due to the impact of noise, the model probabilities experience considerable fluctuations, causing instability in the matching model probabilities, which could potentially lead to misidentification of the matching model. Similarly, the PIMM method proposed in reference [32] and [33] to detect model jumps using the likelihood ratio will also be affected by noise. Based on the concept of the reverse gradient descent, this section uses the model probability historical information as the basis for detecting model jumps and designs a robust jump discriminator. The current matching model can be determined by the ratio of the model probability historical information:

$$\lambda_k^{j|i} = \frac{\phi_k^j}{\phi_k^i}, i = [1, 2, \dots, L] \text{ and } j \neq i \quad (58)$$

where the model probability historical information based on the reverse gradient, denoted as ϕ_k^j , is calculated:

$$\phi_k^j = \sum_{d=0}^{D-1} \frac{\mu_k^j}{\tau^d} \quad (59)$$

where the ranges of the decay factor τ is $(1, +\infty)$. A higher value of τ corresponds to less historical information contained in the jump discriminator; D represents the number of time-steps for retaining model historical probabilities.

The model label ς_k is defined as follows:

$$\varsigma_k = \begin{cases} 1, \min[\lambda_k^{1|i} > 1] \text{ and model non-jumping} \\ 2, \min[\lambda_k^{2|i} > 1] \text{ and model non-jumping} \\ \vdots \\ L, \min[\lambda_k^{L|i} > 1] \text{ and model non-jumping} \\ 0, \text{model jumping} \end{cases} \quad (60)$$

where $\min[\lambda_k^{L|i} > 1]$ represents that the model probability historical information ratio of model L to other models is greater than 1.

From (45)-(47), it can be observed that when model j consistently remains the matching model, its model probability μ_k^j surpasses that of other non-matching models. ϕ_k^j will also be higher than that of other non-matching model historical information. At this point, the historical information ratio $\lambda_k^{j|i} > 1$, and the model label $\varsigma_k = j$. Assuming the tracked target changes its maneuvering, when the matching model jumps from model j to model i , the probability of model j will significantly decrease and become lower than the newly matched model after the jump. Meanwhile, the historical information ratio $\lambda_k^{j|i} < 1$, and the model label $\varsigma_k = 0$. In the next time-step following a model jump, if the matching model remains consistent with the previous time-steps, it is considered sustained, indicating that the dominance of the matching model is stable. Therefore, the historical information ratio $\lambda_k^{j|i} > 1$ and the value of the model label ς_k is the identifier of the matching model.

4.4. Transition probability correction function based on gated memory

When the target model does not jump, maintaining the probability of the matching model at a high level is beneficial for tracking accuracy. To address the interference of noise on the model probability stability for the matching model,

the historical increments of the model probabilities are introduced in (40), and the TPM is corrected by computing the average increment. The parameter N , which retains the historical increments, is an important parameter that regulates the number of the historical increments to be memorized. If some historical increments are memorized, the correction function can more stably maintain the current matching model probability. Therefore, this section introduces a gated mechanism to update and memorize the historical increments of each model probability. Within one time-step, if the matching model remains unchanged, i.e., when the matching model probability is higher than that of other models, the number of saved historical increments n_r will increase until it reaches the set upper limit for historical increments N_r . The TPCF based on updated memory is defined as:

$$g_{non-jumping}^j = g^j = \frac{1}{1 - \Delta\mu_k^{j,n_r}} \quad (61)$$

$$n_r = \begin{cases} n_r + 1, n_r < N_r \\ N_r, n_r \geq N_r \end{cases} \quad (62)$$

However, during the model jumping phases, due to the inclusion of different matching model information, the historical increments from the previous period (when the model did not jump) may conflict with the historical increments after jumping to a new matching model. At this time, an excess of the historical increments stored in memory will slow down the switching speed of the matching model, and these memories should be deleted. Additionally, after the model has completed the jump, it is difficult to rapidly increase the matching model probability through the adaptive TMP. To address this issue, we refer to the gated mechanism to reset the model probability increment by a higher initial increment or prior increment value. And set $n_r=0$ to delete the historical increments from the previous period, thereby increasing the model jumping speed.

A TPCF based on reset memory is defined as:

$$g_{jumping}^j = \frac{1}{1 - \Delta\mu^{j,0}} \quad (63)$$

where $\Delta\mu^{j,0}$ represents the initial increment or prior increment with matching model j .

Compared to the traditional TPCFs, the correction functions defined in (48) and (50) take into account both the model probability historical information and the model jump information, thereby enhancing the noise robustness and model switching speed of the adaptive TPM method. The first step is utilizing the jump discriminator to assess whether the tracking system is experiencing a model jumping phase. Subsequently, during model non-jumping phases, (48) is utilized to update and retain the model probability historical increments. This step enhances the memory of historical increments and maintains high probability and stability for the matching model. During model jumping phases, (50) is employed to clear the historical increments and reset them to their initial or prior values, thereby rapidly boosting the matching model probability to a high level after the model jump. Differing from the PIMM, this method does not require the establishment of a model jumping threshold. It accurately determines the model jumping state under the influence of noise by directly utilizing the model probability historical information. At the same time, it avoids the computational burden of maintaining two parallel IMM systems, thereby having advantages in terms of computational efficiency and robustness.

4.5. The GMIMM-ARKF algorithm

Based on the ARKF algorithm in Section 4.1, and the gated memory IMM method in Section 4.2-4.4, a novel GMIMM-ARKF algorithm is summarized. Fig.2 shows the schematic diagram of the GMIMM-ARKF, and the algorithm process comprises the following six steps.

Step 1) Initialization: Set the initial values of parameters $\hat{\mathbf{x}}_0$, $\mathbf{P}_0$, $\mathbf{Q}_0$, $\mathbf{R}_0$, $\mathbf{\Pi}_0$, μ_0^j based on the tracking system and target state.

Step 2) Model interaction: the mixing model probability is solved using (22) and (23) based on the previously updated adaptive TPM. The states and the error covariances of each model are calculated according to (24) and (25).

Step 3) State estimation: The target state is estimated using an ARKF based on model j , which obtains the state $\hat{\mathbf{x}}_k^j$, the error covariance $\mathbf{P}_k^j$, and the innovation covariance $\mathbf{P}_{zz,k}^j$.

Step 4) Model probability update: The likelihood of model j is solved by (26) and (27), and the model probabilities is updated according to (28).

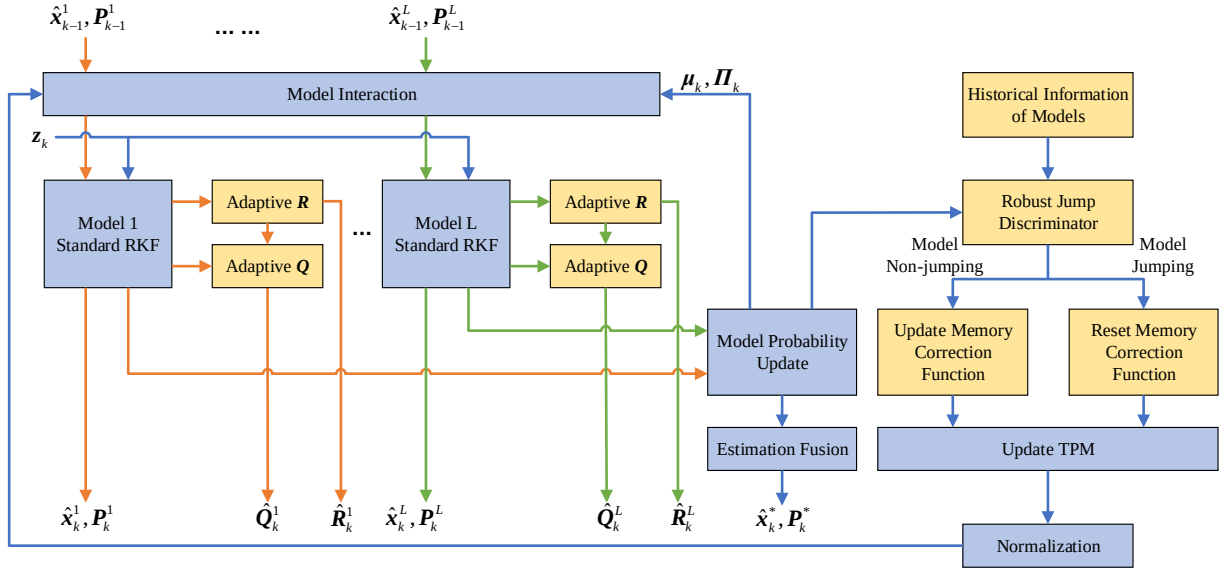


Fig. 2. Schematic diagram of the GMIMM-ARKF.

Step 5) The TPM adaptation based on gated memory: First, use (45)-(47) to determine whether the tracking system has entered the model jump phase. If the system is in the non-model jumping phase, update the memory of the TPCF. Select (48) as the correction function, and (42) and (43) are used to update the TPM. When the system is in the model jumping phase, the memory of the TPCF is reset. (50) is chosen as the correction function, followed by updating the TPM using (42) and (43).

Step 6) Estimation Fusion: (29) and (30) are used to fuse the output results of the sub-models, obtaining the state $\hat{x}_k^*$ and error covariance P_k^* of the target at time-step k .

5. Numerical simulation and experiment

Numerical simulation and experiment were conducted to compare the proposed algorithm with some conventional algorithms to verify the tracking accuracy of the proposed algorithm. The model jump threshold of the PIMM-ARKF algorithm is $Th = 1$. The root mean square errors (RMSE) and the averaged RMSE (ARMSE) are used to compare the estimation effects of the above filtering algorithms, which are calculated as follows:

$$RMSE = \sqrt{\sum_{i=1}^{Mc} (x_k^i - \hat{x}_k^i)^2 / Mc} \quad (64)$$

$$ARMSE = \sqrt{\sum_{k=1}^{T_N} \sum_{i=1}^{Mc} (x_k^i - \hat{x}_k^i)^2 / T_N Mc} \quad (65)$$

5.1. Numerical Simulation and Discussion

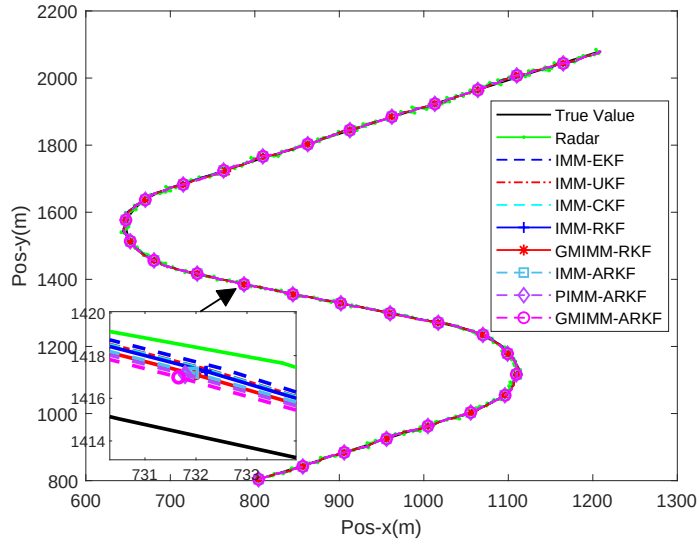
The sampling time for this numerical simulation is $T = 1s$, and Monte Carlo runs 500 times. The IMM model set consists of CV model, coordinated left turn (CLT) model and coordinated right turn (CRT) model, which is used to deal with the maneuvering actions of straight, left turn, and right turn of the tracking target, respectively [6]. The relationship between the real motion model of the tracked target and the time series is shown in Table 1. The coordinated turn rates of CLT and CRT models are $\omega_L = 0.04/0.06$ rad/s and $\omega_R = -0.04/0.06$ rad/s.

The target initial state is $\hat{x}_0 = [800m, 5m/s, 800m, 4m/s]$, The position of the radar is $[x_0, y_0] = [0m, 0m]$. Assuming that the radar can obtain the relative distance and azimuth angle of the target, the measurement equation can

Table 1

The real motion model of the tracked target.

Time(s)	1~50	51~100	101~150	151~200	201~300
Model	CV	CLT	CV	CRT	CV


Fig. 3. A radar tracking scenario including the target vessel and radar images (the vessel position marked by the red rectangles).

be given by

$$\mathbf{z}_k = \begin{pmatrix} \sqrt{(x_k - x_0)^2 + (y_k - y_0)^2} \\ \arctan(\frac{y_k - y_0}{x_k - x_0}) \end{pmatrix} + \mathbf{v}_k \quad (66)$$

The covariance of process noise and measurement noise are as follows:

$$\mathbf{Q}_k = \sigma_q \begin{bmatrix} T^3/3 & T^2/2 & 0 & 0 \\ T^2/2 & T & 0 & 0 \\ 0 & 0 & T^3/3 & T^2/2 \\ 0 & 0 & T^2/2 & T \end{bmatrix} \quad (67)$$

$$\mathbf{R}_k = \sigma_r \begin{bmatrix} \delta_r^2 & 0 \\ 0 & \delta_\theta^2 \end{bmatrix} \quad (68)$$

where δ_r^2 is the distance variance, and δ_θ^2 is the azimuth variance; σ_q and σ_r represent the intensity of process and measurement noise, respectively.

To validate the performance of the proposed algorithm under unknown noise conditions, the initial noise and real noise are set separately. Therefore, the noise intensity of the initial process noise of the algorithm is $\sigma_q^{\text{int}} = 0.01$, and the parameters for the initial measurement noise are $\sigma_r^{\text{int}} = 1$, $\delta_r^{\text{int}} = 3$ and $\delta_b^{\text{int}} = 0.2^\circ$. The noise intensity of the actual process noise is $\sigma_q^{\text{true}} = 0.01$. The true measurement noise in the numerical simulations are represented using a mixture of Gaussian distribution noise, as described in [4], which is given by

$$\mathbf{R}_k^{\text{true}} = 0.6 \begin{bmatrix} \delta_{r1}^2 & 0 \\ 0 & \delta_{b1}^2 \end{bmatrix} + 0.4 \begin{bmatrix} \delta_{r2}^2 & 0 \\ 0 & \delta_{b2}^2 \end{bmatrix} \quad (69)$$

where the parameters for the true measurement noise are $\delta_{r1}=2$, $\delta_{r1}=0.5$, $\delta_{\theta1}=0.2^\circ$ and $\delta_{\theta2}=0.1^\circ$.

The initial values of the probabilities for each model are respectively $\mu_{CV}=0.4$, $\mu_{CLT}=0.3$ and $\mu_{CRT}=0.3$. The number of time-steps for retaining historical model probabilities is $D=4$, and the upper limit for historical increments is $N_r=4$. The prior increment $\Delta\mu^0$ and the TPM initial values are as follows:

$$\begin{cases} \Delta\mu_{CV}^0 = [0.09, 0.005, 0.005] \\ \Delta\mu_{CLT}^0 = [0.005, 0.09, 0.005] \\ \Delta\mu_{CRT}^0 = [0.005, 0.005, 0.09] \end{cases} \quad (70)$$

$$\pi_0 = \begin{bmatrix} 0.9 & 0.05 & 0.05 \\ 0.05 & 0.9 & 0.05 \\ 0.05 & 0.05 & 0.9 \end{bmatrix} \quad (71)$$

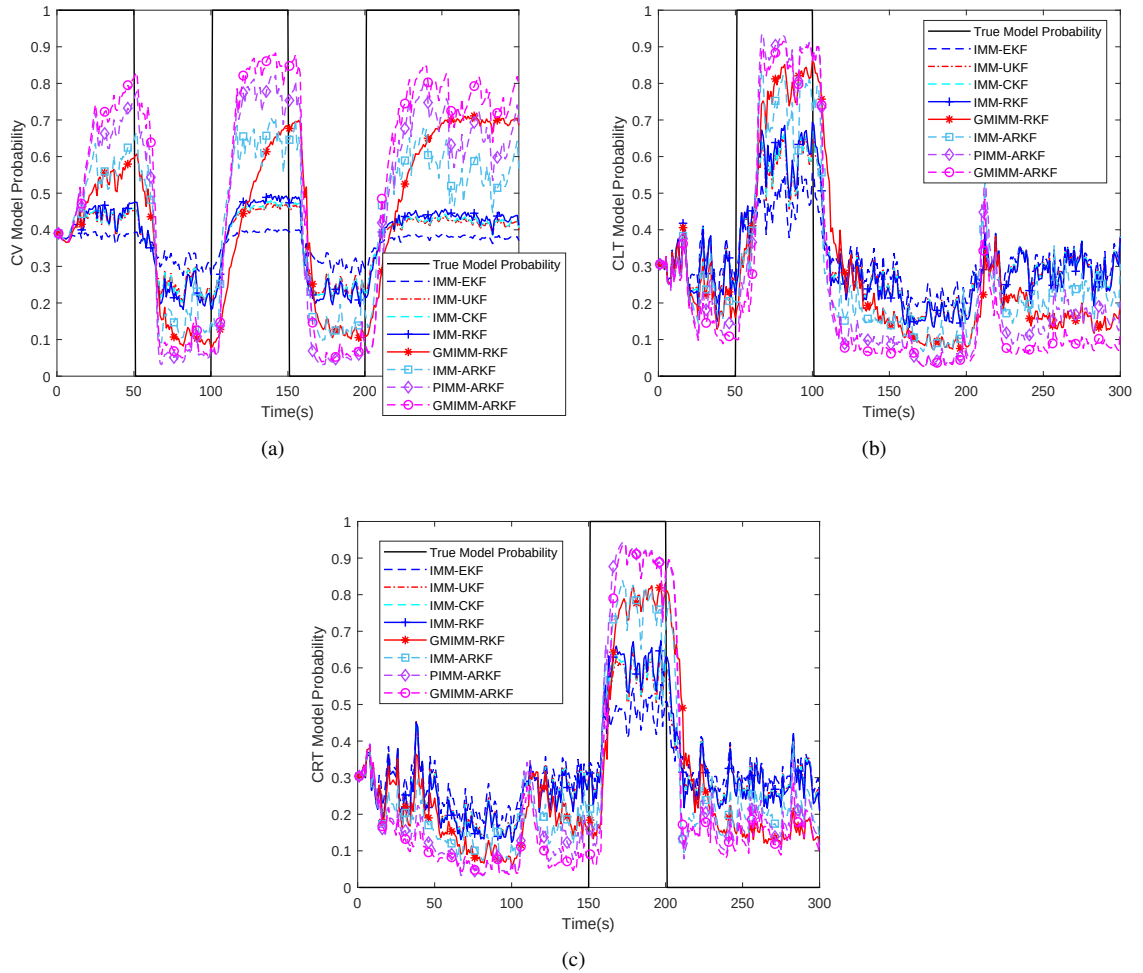


Fig. 4. The model probabilities of each algorithm and the true model probabilities. (a) Probabilities of CV. (b) Probabilities of CLT. (c) Probabilities of CRT.

Fig.3 shows the target motion trajectory and the tracking results of each algorithm. Assuming the model probability of the true model is 1, Fig.4 shows the comparison with the model probabilities of each algorithms and the true model

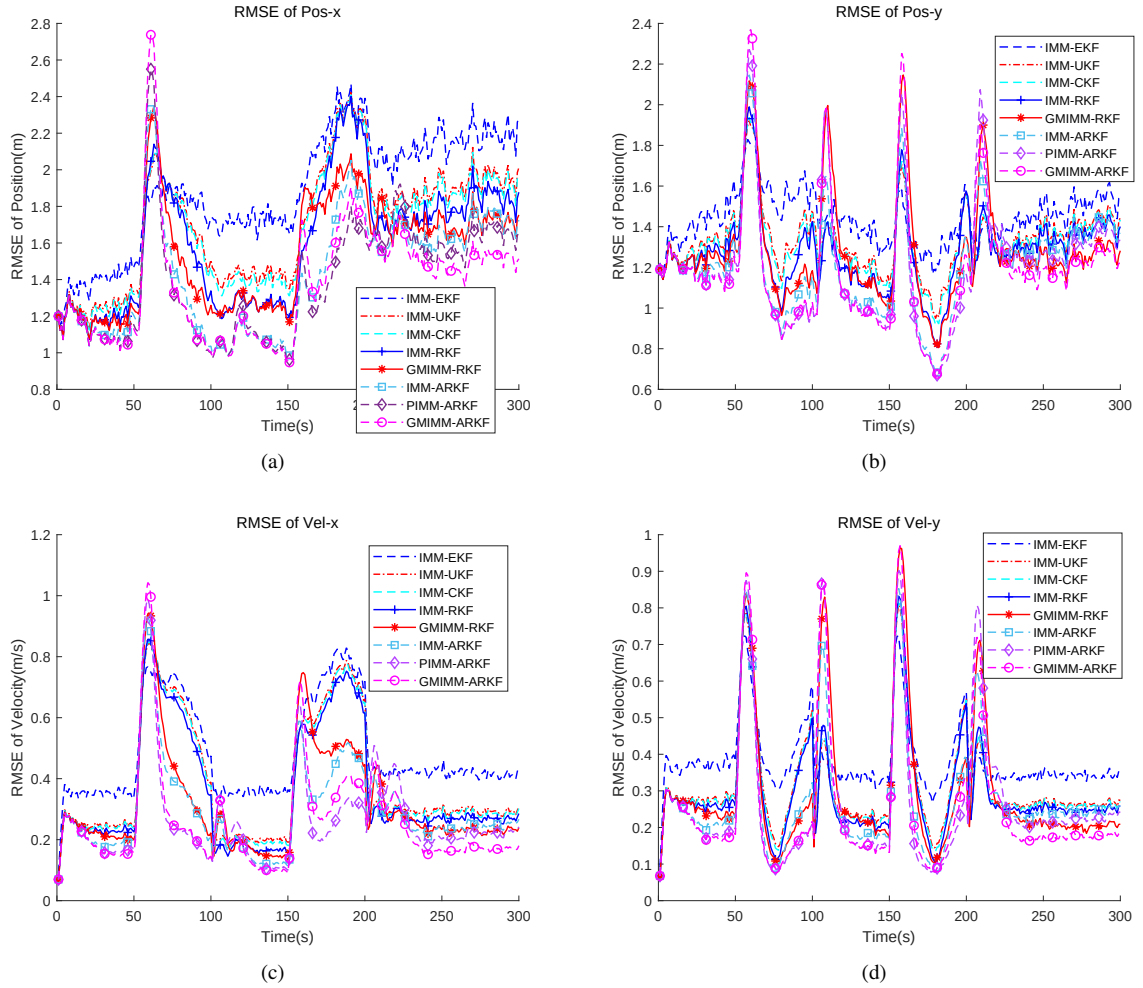


Fig. 5. The position and velocity RMSE of each algorithm on the x-axis and y-axis. (a) RMSE of position on the x-axis. (a) RMSE of position on the y-axis. (b) RMSE of velocity on the x-axis. (b) RMSE of velocity on the y-axis.

probabilities in a single Monte Carlo simulation. From Fig.4, it can be seen that both the adaptive TPM method based on gated memory and the noise adaptive method can improve the matching model probability. The matching model probability of the proposed algorithm can rapidly increase after the model jumps, and in most cases, it is closest to the true model.

Fig.5 shows a comparison of the position and velocity RMSE of each algorithm. It can be observed from Fig.5 that the RMSE of the proposed algorithm is lower than that of other algorithms during most model non-jumping phase. At time-steps 160-200, the RMSE of the PIMM-ARKF algorithm is the smallest, with the proposed algorithm following closely behind. During the model jumping phase in the tracking process, the RMSE of all five algorithms shows a noticeable increase. The reason is that the model probabilities, the adaptive TPM, the adaptive process and measurement noise remain in their states prior to the model jump, which do not correspond to the true maneuvering and noise after the jump. Since IMM, adaptive TPM, and noise adaptation are all soft-switching methods, they require a short period to adapt to the new maneuvering state. Thus, the proposed algorithm can swiftly accomplish model switching and maintain lower RMSE during model non-jumping phases.

Table 2 and Table 3 lists the position and velocity ARMSE of each algorithm, indicating that the proposed algorithm achieves the best tracking accuracy among the five algorithms. Meanwhile, the ARMSE of the model probabilities for the proposed algorithm is also the best among the five algorithms [32]. Compared to the IMM-RKF in Table 2, the

Table 2

Comparison of the position and velocity ARMSE for simulation (The coordinated turn rates are 0.04 and -0.04).

Algorithm	Pos-x (m)	Pos-y (m)	Vel-x (m/s)	Vel-y (m/s)	Model probability
IMM-EKF	1.8918	1.4465	0.4822	0.3811	0.3992
IMM-UKF	1.7071	1.3345	0.3858	0.3121	0.3676
IMM-CKF	1.6832	1.3215	0.3785	0.3082	0.3649
IMM-RKF	1.5167	1.1176	0.3339	0.2676	0.3475
GMIMM-RKF	1.4525	1.1057	0.2908	0.2604	0.2778
IMM-ARKF	1.2971	1.0261	0.2584	0.2378	0.2702
PIMM-ARKF [32]	1.2632	1.0137	0.2288	0.2294	0.2217
GMIMM-ARKF	1.2461	1.0016	0.2251	0.2209	0.2039

Table 3

Comparison of the position and velocity ARMSE for simulation (The coordinated turn rates are 0.06 and -0.06).

Algorithm	Pos-x (m)	Pos-y (m)	Vel-x (m/s)	Vel-y (m/s)	Model probability
IMM-EKF	1.6412	1.4293	0.4765	0.4588	0.3652
IMM-UKF	1.4794	1.3431	0.3819	0.3963	0.3306
IMM-CKF	1.4602	1.3332	0.3757	0.3923	0.3279
IMM-RKF	1.4054	1.3057	0.3605	0.3825	0.3209
GMIMM-RKF	1.3563	1.2688	0.3154	0.3441	0.2422
IMM-ARKF	1.2898	1.2317	0.2931	0.3221	0.2502
PIMM-ARKF [32]	1.2646	1.2182	0.2646	0.3016	0.1997
GMIMM-ARKF	1.2503	1.2051	0.2563	0.2893	0.1840

Table 4

Average running time for single simulation.

Algorithm	IMM-EKF	IMM-UKF	IMM-CKF	IMM-RKF
Running time (s)	0.0329	0.0911	0.0905	0.0915
Algorithm	GMIMM-RKF	IMM-ARKF	PIMM-ARKF [32]	GMIMM-ARKF
Running time (s)	0.1016	0.1245	0.2479	0.1352

proposed algorithm reduces the position and velocity ARMSE by 15.1% and 26.3%, respectively, and decreases the model probability ARMSE by 41.3%.

Table 4 lists the average running time for a single simulation. In this table, the computation time of the proposed algorithm is slightly longer than that of the traditional IMM-RKF algorithm. The reasons are as follows: 1) The proposed algorithm requires computation of adaptive measurement noise and process noise; 2) It needs to utilize a robust jump discriminator and gated memory mechanism to adapt the TPM. As the PIMM method necessitates the continuous maintenance of two IMM filters for target tracking, it incurs a substantial computational burden. In contrast, the proposed algorithm reduces computation time by 45.5% compared to the PIMM-ARKF algorithm.

5.2. Experiment and Discussion

To verify the effectiveness of the proposed algorithm in practical target tracking systems, experiments were conducted by tracking ships navigating the Wuhan section of the Yangtze River using a radar tracking system. The target tracking scenario is shown in Fig.1. The radar tracking system integrates multiple devices, including radar units, computers, and surveillance cameras. The radar is employed to acquire target information, which is subsequently transmitted to a computer for data processing and target tracking. The radar is employed to acquire target information, which is subsequently transmitted to a computer for data processing and target tracking. The radar employs a SIMRAD

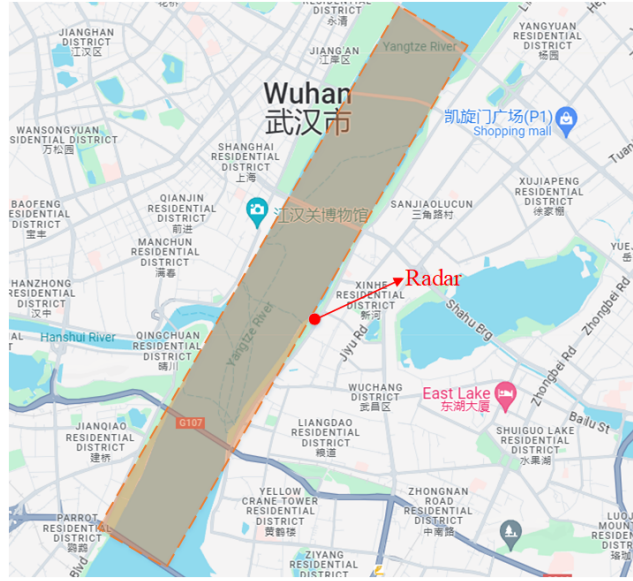


Fig. 6. Experimental area and the position of radar.

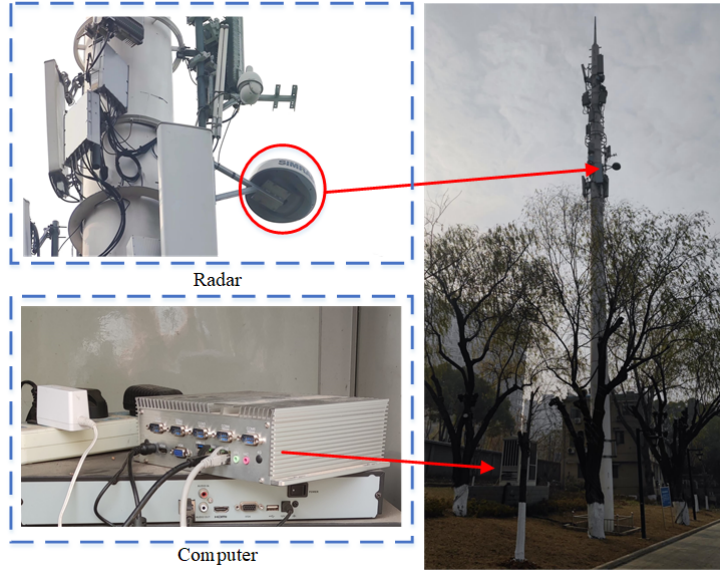


Fig. 7. The radar tracking system.

HALO20+ unit operating in the X-band frequency range of 9.4-9.5 GHz, while the CPU of computational platform is Intel Core i5-10400 2.90 GHz. Fig.6 shows the experimental area. Fig.7 illustrates the radar tracking system. The radar is mounted on the signal tower to detect and track passing vessels. In this experiment, the ferry operating on the Yangtze River will be used as the target vessel. Real-time target positioning data were acquired by deploying a centimeter-accuracy DGPS system aboard the target vessel. Due to the high positioning accuracy of DGPS, in this experiment, the ferry trajectory output by DGPS is assumed to be the true trajectory, and this data is used to validate the performance of the proposed algorithm. For this experiment, error and mean absolute error (MAE) are employed as metrics to evaluate the tracking accuracy of the proposed algorithm, and the formulas are given by:

$$Error = |x_k - \hat{x}_k| \quad (72)$$

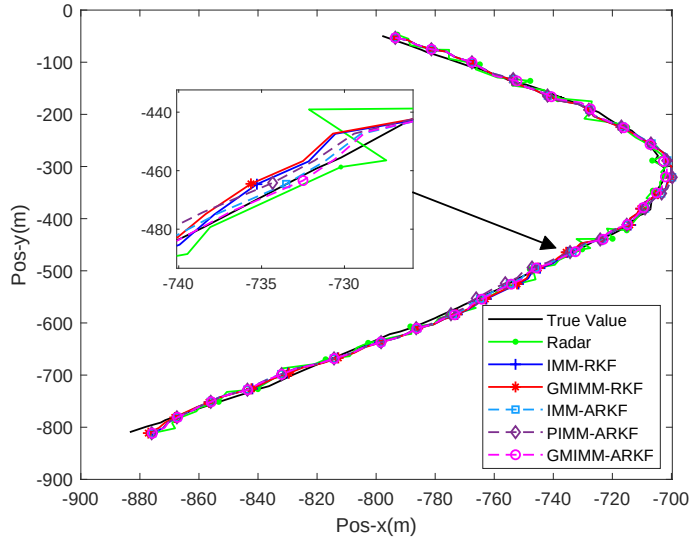


Fig. 8. The target trajectory and the tracking results of the experiment.

$$MAE = \sum_{k=1}^{T_M} |x_k - \hat{x}_k| / T_M \quad (73)$$

In this experiment, the radar sampling time is 2.5s. The North-East coordinate system is established with the radar coordinates (N30°34'0.4260", E114°18'1.8072") as the origin, and the radar position is $[0m, 0m]$. We set the target initial state relative to the radar as $[-797.92m, 1.80m/s, -49.62m, 3.55m/s]$. Furthermore, the target motion trajectory and the tracking results of the five algorithms are shown in Fig.8. The target undergoes three motion during the tracking process: straight, right turn and straight.

Table 5

Comparison of position and velocity MAE for experiment.

Algorithm	IMM-RKF	GMIMM-RKF	IMM-ARKF	PIMM-ARKF [32]	GMIMM-ARKF
Pos-x (m)	2.6439	2.5759	2.3885	2.3921	2.2362
Pos-y (m)	2.7751	2.7597	2.7512	2.7706	2.4185

Table 6

Comparison of running time for experiment.

Algorithm	IMM-RKF	GMIMM-RKF	IMM-ARKF	PIMM-ARKF [32]	GMIMM-ARKF
Running time (s)	0.0910	0.0925	0.1077	0.1767	0.1155

Fig.9 shows the model probabilities of each algorithm in the experiment. As shown in Fig.9 (a) and (c), significant improvements are observed in the model probabilities of the CV and CRT models during straight and right turn movements, respectively. Among them, the model probabilities of the proposed algorithm best match the target maneuvers, and it has a fast model switching speed. Since the target did not perform a left turn, the probability of the CLT model should remain relatively low. In Fig.9 (b), the proposed algorithm exhibits consistently lower probabilities for the CLT model in most periods.

Fig.10 shows the comparison of position errors. It can be observed that the proposed algorithm exhibits better target tracking accuracy than other algorithms. Table 5 shows the MAE of each algorithm, indicating that the proposed

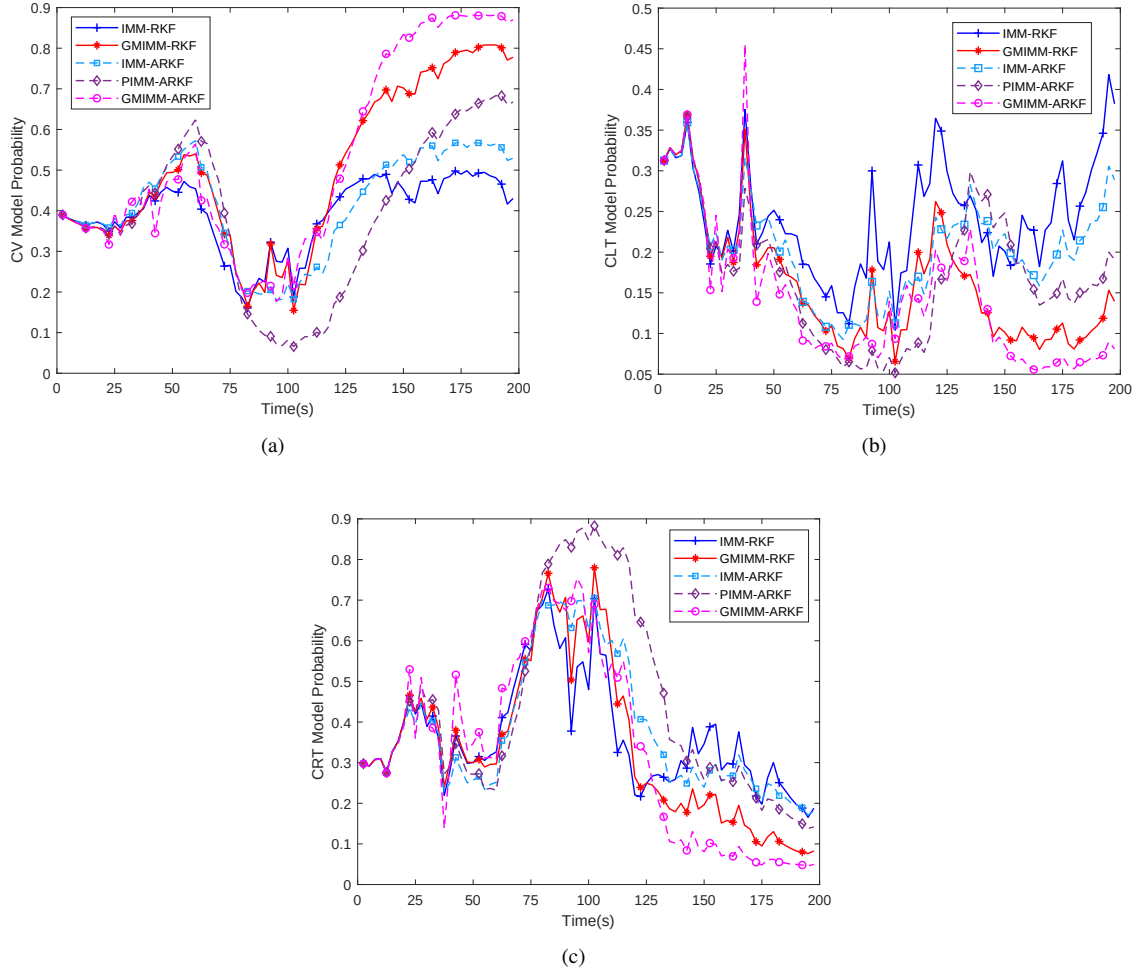


Fig. 9. The model probabilities of each algorithm in the experiment. (a) Probabilities of CV. (b) Probabilities of CLT. (c) Probabilities of CRT.

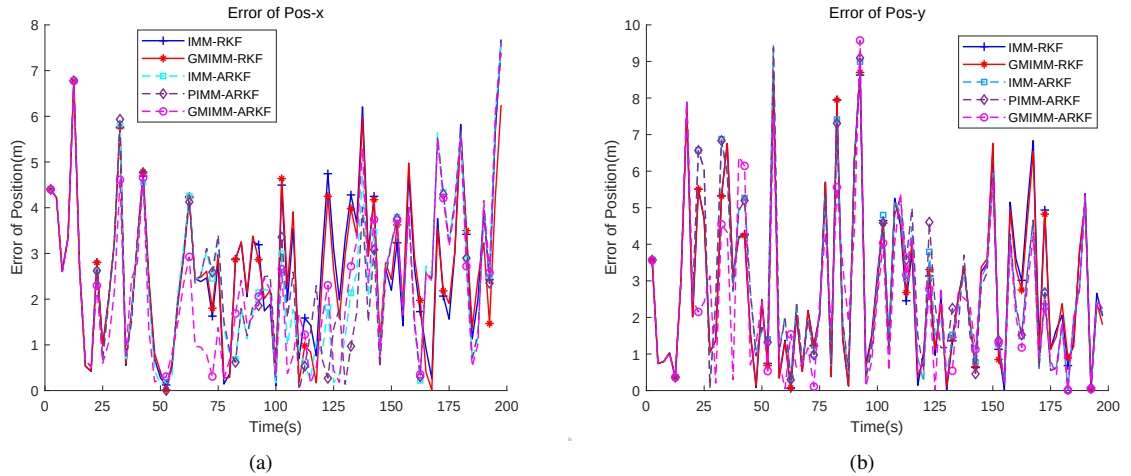


Fig. 10. The comparison of position errors. (a) Position errors on the x-axis. (b) Position errors on the y-axis.

algorithm has the lowest MAE in position. Table 6 presents the running time of each algorithm during the tracking process, with the proposed algorithm computation time notably lower than that of the PIMM-ARKF. Hence, the performance of the proposed algorithm has been further confirmed by this experiment.

6. Conclusion

In this paper, a novel method for target tracking based on radar measurement units was proposed, which can track surface maneuvering vessels under the interference of unknown noise. To reduce the negative impact of unknown process and measurement noise on target tracking accuracy, an adaptive RKF algorithm was proposed. It can calculate measurement noise and process noise in real-time based on residual sequences, and ensure algorithm stability by setting upper and lower bounds for the adaptive factor. To enhance the matching model probability and enable rapid switching during model jumps in the presence of noise, the robust jump discriminator and the TPCFs based on a gated memory mechanism are designed. Numerical simulation results showed that the GMIMM-ARKF algorithm reduces errors in position, velocity and model probability by 15.1%, 26.3% and 41.3%, compared to the IMM-RKF algorithm. Furthermore, it reduced the computation cost by 45.5% compared to the PIMM-ARKF algorithm. The surface vessel tracking experiments validated the feasibility and effectiveness of the GMIMM-ARKF algorithm.

In practical applications, the vessel maneuvers may not occur at ideal uniform speeds, and the model combinations used in this paper may not fully encompass all maneuvering situations. We will study the model combinations that better reflect real vessel maneuvers in future work to solve this problem. Additionally, this paper considers a surface vessel tracking method for shore-based radar. Future research will explore tracking methods for shipborne radar and use a ship-shore cooperative framework to fuse information from both shore-based and shipborne radars. The fusion results will provide more comprehensive and reliable information about the navigation environment to support the safe navigation of intelligent vessels.

CRedit authorship contribution statement

Xiaoqian Wang: Conceptualization, Methodology, Software, Formal analysis, Writing-original draft, Writing-review & editing. **Hui Feng:** Supervision, Funding acquisition, Project administration, Resources, Validation, Writing-review & editing. **Haixiang Xu:** Supervision, Project administration, Resources, Writing-review & editing. **Jianhua He:** Supervision, Validation, Formal analysis, Writing-review & editing.

Declaration of competing interest

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

Acknowledgement

The authors appreciate the constructive suggestions from reviewers and the Associate Editor. This work is supported by the National Natural Science Foundation of China under Grant No. 52371374, 51979210. This work is partly funded by EPSRC with RC Grant reference EP/Y027787/1, UKRI under grant number EP/Y028317/1, the Horizon Europe MSCA programme under grant agreement No 101086228.

References

- [1] O.-K. Helgesen, K. Vasstein, E.-F. Brekke, and A. Stahl. Heterogeneous multi-sensor tracking for an autonomous surface vehicle in a littoral environment. *Ocean Engineering*, 252:111168, 2022.
- [2] X. Shi, A. Taheri, T. Cecen, and N. Celik. Particle filtering-based low-elevation target tracking with multipath interference over the ocean surface. *IEEE Transactions on Aerospace and Electronic Systems*, 56(4):3044–3054, 2020.
- [3] Y. Wu, X. Chu, L. Deng, J. Lei, W. He, G. Królczyk, and Z. Li. A new multi-sensor fusion approach for integrated ship motion perception in inland waterways. *Measurement*, 200:111630, 2022.
- [4] Y. Fan, S. Qiao, G. Wang, S. Chen, and H. Zhang. A modified adaptive kalman filtering method for maneuvering target tracking of unmanned surface vehicles. *Ocean Engineering*, 266:112890, 2022.
- [5] H. Fu, Y. Cheng, and W. Huang. A novel variational robust filter with gaussian mixture model for unknown non-gaussian noises. *Measurement*, 221:113417, 2023.

- [6] X.-R. Li and V.-P. Jilkov. Survey of maneuvering target tracking. part i. dynamic models. *IEEE Transactions on Aerospace and Electronic Systems*, 39(4):1333–1364, 2003.
- [7] M. Kumar and S. Mondal. Recent developments on target tracking problems: A review. *Ocean Engineering*, 236:109558, 2021.
- [8] I. Hwang, C.-E. Seah, and S. Lee. A study on stability of the interacting multiple model algorithm. *IEEE Transactions on Automatic Control*, 62(2):901–906, 2016.
- [9] D. Zhang and M. Hao. Tracking magnetic target based on interactive multi-model square root unscented kalman filter. *IEEE Transactions on Magnetics*, 59(6):1–12, 2023.
- [10] Q. Cai, C. Yang, J. Li, G. Yang, N. Song, H. Yin, and Y. Liu. An el-shakf-based integration scheme for gyro thermal-magnetic coupling heading-effect drift compensation in ins. *IEEE Sensors Journal*, 20(19):11508–11517, 2020.
- [11] S. Akhlaghi, N. Zhou, and Z. Huang. Adaptive adjustment of noise covariance in kalman filter for dynamic state estimation. *2017 IEEE Power & Energy Society General Meeting*, pages 1–5, 2017.
- [12] X. Li, P. Ma, X. Wen, T. Chao, and M. Yang. A robust and efficient cubature kalman filter based on the variational bayesian method and its application in target tracking. *Measurement Science and Technology*, 36(1):016135, 2024.
- [13] Z. Zhou, H. Xu, H. Feng, and W. Li. A game theory-based fusion algorithm for autonomous navigation of smart ships. *Measurement*, 216:112897, 2023.
- [14] O.-J. Montanez, M.-J. Suarez, and E.-A. Fernandez. Application of data sensor fusion using extended kalman filter algorithm for identification and tracking of moving targets from lidar-radar data. *Remote Sensing*, 15(13):3396, 2023.
- [15] B. Yang, E. Yang, L. Yu, and C. Niu. Adaptive extended kalman filter-based fusion approach for high-precision uav positioning in extremely confined environments. *IEEE/ASME Transactions on Mechatronics*, 28(1):543–554, 2022.
- [16] Y. Meng, X. Zhang, X. Zhang, and Y. Duan. Maneuverability indices estimation of nonlinear response model based on full-scale trial data and square root unscented kalman filter. *Ocean Engineering*, 286:115699, 2023.
- [17] H. Liu, Q. Jiang, Y. Qin, R. Yin, and S. Zhao. Double layer weighted unscented kalman underwater target tracking algorithm based on sound speed profile. *Ocean Engineering*, 266:112982, 2022.
- [18] Z. Wang, Y. Huang, M. Wang, J. Wu, and Y. Zhang. A computationally efficient outlier-robust cubature kalman filter for underwater gravity matching navigation. *IEEE Transactions on Instrumentation and Measurement*, 71:1–18, 2022.
- [19] H. Fu, Q. Xiao, T. Lou, and M. Xiao. Nonlinear and non-gaussian rank filter method. *Journal of Aerospace Power*, 30(10):2318–2322, 2015.
- [20] Y. Zhang, M. Li, Y. Zhang, Z. Hu, Q. Sun, and B. Lu. An enhanced adaptive unscented kalman filter for vehicle state estimation. *IEEE Transactions on Instrumentation and Measurement*, 71:1–12, 2022.
- [21] Q. Luo, Y. Shao, J. Li, X. Yan, and C. Liu. A multi-auv cooperative navigation method based on the augmented adaptive embedded cubature kalman filter algorithm. *Neural Computing and Applications*, 34(21):18975–18992, 2022.
- [22] Z. Zhang, L. Zheng, Y. Li, H. Wu, Y. Liang, and X. Qiao. Correction adaptive square-root cubature kalman filter with application to autonomous vehicle target tracking. *Measurement Science and Technology*, 32(11):115101, 2021.
- [23] X. Fang and D. Huang. Robust adaptive cubature kalman filter for tracking manoeuvring target by wireless sensor network under noisy environment. *IET Radar, Sonar & Navigation*, 17(2):179–190, 2023.
- [24] T. Lou, D. Han, X. Yang, and S. Jiang. Robust state estimation of induction motor using desensitized rank kalman filter. *2020 Chinese Automation Congress (CAC)*, pages 450–456, 2020.
- [25] T. Lou, H. Ban, S. Zhao, Z. He, and Y. Wang. Rank kalman filter-slam for vehicle with non-gaussian noise. *2020 5th International Conference on Advanced Robotics and Mechatronics (ICARM)*, pages 12–15, 2020.
- [26] M. Ebrahimi, M. Ardeshtiri, and S.-A. Khanghah. Bearing-only 2d maneuvering target tracking using smart interacting multiple model filter. *Digital Signal Processing*, 126:103497, 2022.
- [27] K. Samuel and J.-W. Choi. Improved imm filter for tracking a highly maneuvering target with mixed system noises. *International Journal of Control, Automation and Systems*, 16:2763–2771, 2018.
- [28] B. Hou, J. Wang, Z. He, Y. Qin, H. Zhou, D. Wang, and D. Li. Novel interacting multiple model filter for uncertain target tracking systems based on weighted kullback-leibler divergence. *Journal of the Franklin Institute*, 357(17):13041–13084, 2020.
- [29] X. Zhang, B. He, S. Gao, P. Mu, J. Xu, and N. Zhai. Multiple model auv navigation methodology with adaptivity and robustness. *Ocean engineering*, 254:111258, 2022.
- [30] L.-B. Cosme, M.-F.-S.-V. D’Angelo, W.-M. Caminhas, M.-O. Camargos, and R.-M. Palhares. An adaptive approach for estimation of transition probability matrix in the interacting multiple model filter. *Journal of Intelligent Fuzzy Systems*, 41(1):155–166, 2021.
- [31] I.-H. Lee and C.-G. Park. An improved interacting multiple model algorithm with adaptive transition probability matrix based on the situation. *International Journal of Control, Automation and Systems*, 21(10):3299–3312, 2023.
- [32] G. Xie, L. Sun, T. Wen, X. Hei, and F. Qian. Adaptive transition probability matrix-based parallel imm algorithm. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 51(5):2980–2989, 2021.
- [33] X. Li, B. Lu, Y. Li, X. Lu, and H. Jin. Adaptive interacting multiple model for underwater maneuvering target tracking with one-step randomly delayed measurements. *Ocean Engineering*, 280:114933, 2023.
- [34] L. Li and Y. Xia. Stochastic stability of the unscented kalman filter with intermittent observations. *Automatica*, 48(5):978–981, 2012.
- [35] C. Hajiyev and H.-E. Soken. Robust adaptive unscented kalman filter for attitude estimation of pico satellites. *International Journal of Adaptive Control and Signal Processing*, 28(2):107–120, 2014.