



OPEN

Adaptive filtering and smoothing algorithm based on variable structure interactive multiple model

Kai-Yu Hu^{1,2✉}, Jiaming Wang¹, Yuqing Cheng¹ & Chunxia Yang¹

For maneuvering target tracking, a novel adaptive variable structure interactive multiple model filtering and smoothing (AVSIMMFS) algorithm is proposed in this paper. Firstly, an accurate model of the variable structure interactive multiple model algorithm is established. Secondly, by constructing a new model subset based on the original model subsets, the matching accuracy between the model subset and the actual maneuvering mode of the target is improved. Then, the AVSIMMFS algorithm is obtained by smoothing the filtered data of the new model subset. Because of the combination of forward filtering and backward smoothing, the target tracking accuracy is further improved. Finally, in order to verify the effectiveness of the algorithm, the simulation is carried out on two cases. The simulation results show that the tracking performance of AVSIMMFS algorithm is better than other methods and has lower calculation cost.

In the past few years, filters have been extensively employed in target tracking, parameter estimation, and state prediction^{1–6}. When the target maneuvering is extremely complicated and the filtering model does not align with the target maneuvering model, the accuracy of the filtering methods based on a single model will significantly decrease. To address this problem, the interactive multiple model (IMM) algorithm exhibits superior performance^{7–9}.

The IMM algorithm has received a great deal of attention in recent years, scholars have continuously developed and improved it in different aspects. In¹⁰, an IMM algorithm was designed based on the Constant Velocity (CV) and Current Statistics (CS) models, in which the average velocity of the CS model was first estimated by the least square method, and then the CS model was applied to the IMM algorithm, which improved the model accuracy. In¹¹, IMM algorithm had been improved in many aspects, including the adoption of improved Kalman filter as a sub-filter, and an entropy-based model probability update formula. An alternative method to IMM had been proposed in^{8,12}, the models in the model set were all composed of Constant accelerated (CA) models, which reduces the complexity of the model set. On the basis of this alternative method, the¹³ proposed an adaptive IMM algorithm, which firstly used the filter to estimate the acceleration of the target, and then selected the value near the estimated acceleration to build the model set, this method could reduce the number of models in the model set and improved the model accuracy while reducing the computational cost. In¹⁴, a second-order IMM algorithm was proposed based on the second-order Markov chain, which further improved the filtering accuracy due to the use of more prior information. Since the type and quantity of models in the model set of the above-mentioned IMM are unchanged, it is also called a Fixed Structure IMM (FSIMM)¹⁵.

In order to avoid accuracy errors due to model mismatch, when using the IMM algorithm, as many models as possible should be used to cover the target maneuvering model. However, it is worth noting that too many models in a single model set will also reduce the filtering accuracy^{16,17}. To overcome this defect, a variety of Variable Structure IMM (VSIMM) algorithms have been presented. After continuous development and improvement, VSIMM can be divided into four types: Model-Group-Switching (MGS), Likely-Mode-Set (LMS), Expected-Mode-Augmentation (EMA) and Adaptive Grid (AG)^{15,18,19}. Among them, MGS divides the model set into model subsets, only one model subset is selected for estimation at a time, and the switching between model subsets is selected according to the transition probability of model subsets²⁰. The LMS divides the models into three types at each moment: impossible, important, and dominant models, and the model subset used for estimation

¹The 304 Institute, China Aerospace Science and Industry Corporation, Beijing 100074, People's Republic of China. ²Center for Applied Mathematics, China Aerospace Science and Industry Corporation, Beijing 100074, People's Republic of China. ✉email: hkywuyue@163.com

at each moment consists of dominant and near-dominant models^{21,22}. Similar to MGS, two typical estimators divide a large model set into small model subsets, then calculates the probabilities of all model subsets at the next moment, and selects the model subset with the highest probability for estimation²³. Combined with graph theory, the AG algorithm forms a grid of all models, and uses prior information and current data to get a local refined grid, which forms a candidate model subset, and then selects models according to some rules to form a model subset for estimation at the next moment²⁴.

It is worth noting that the above-mentioned EMA algorithm approximates the formula when calculating the likelihood function and model subset probability. To solve this problem, a normal VSIMM algorithm with accurate mathematical model is designed in this paper. In addition, considering that when EMA divides a large model set into small model subsets, in order to reduce the computational cost, although all model subsets include all models, they are not all permutations and combinations. Therefore, in order to ensure that there is a model subset that can best match the target maneuver mode in all model subsets, on the basis of the existing model subset, a new model subset is constructed according to the rules, thus obtaining the adaptive VSIMM (AVSIMM) algorithm. Because the model in the new model subset may match the target maneuver model better, the target tracking accuracy can be improved. Finally, based on the forward AVSIMM estimation, the data are further smoothed backward, so we get an AVSIMM Filtering and Smoothing (AVSIMMFS) algorithm, which can further improve the tracking accuracy. The main contributions of this study can be summarized as follows:

- (1) A model of normal VSIMM algorithm is established and applied in the design of filter, it has obvious effect on eliminating fast random clutter. The variable structure accurate model in the algorithm avoids EMA's model approximation error and makes the subsequent filtering smooth more accurate.
- (2) Based on the VSIMM algorithm, the AVSIMM algorithm is presented and improves the matching degree between the model subset and the target maneuvering model, thus improving the tracking accuracy. The calculation time is greatly reduced by extracting the existing high probability models to form a new model subset and matching them directly.
- (3) The adaptive variable structure smoothing ensures that the peak buffeting of the tracking signal is eliminated when the target frequently switches flight modes, making the multi-target characteristics of the AVSIMMFS scheme more adaptive than the existing algorithms.

The rest of the paper is organized as follows. In Section “[The IMM filtering and IMM smoothing algorithm](#)”, the filtering and smoothing problems are described, and the mathematical models of the IMM filtering algorithm and the IMM smoothing algorithm are given. In Section “[The AVSIMMFS algorithm](#)”, the AVSIMMFS algorithm is designed based on normal VSIMM, including forward filtering and backward smoothing. The numerical simulation results are shown in Section “[Simulation and discussion](#)”. Eventually, the conclusion is summarized in Section “[Conclusion](#)”.

The IMM filtering and IMM smoothing algorithm

Filtering and smoothing problems. Assuming that the target may have r motion models, the model set is denoted as $\Omega = \{M^1, \dots, M^r\}$, and the transition probability matrix between the models is:

$$P = \begin{pmatrix} p^{11} & \dots & p^{1r} \\ \vdots & \ddots & \vdots \\ p^{r1} & \dots & p^{rr} \end{pmatrix} \quad (1)$$

where, p^{ij} ($1 \leq i \leq r, 1 \leq j \leq r$) is the transition probability from model i to model j .

The state equation for the discretization of the system is as follows:

$$x_k^j = F_k^j x_{k-1}^j + w_{k-1}^j, j = 1, 2, \dots, r \quad (2)$$

where, x_k^j is the state vector of the system, F_k^j is the state transition matrix of the model j , w_{k-1}^j is Gaussian white noise with mean 0, and its covariance matrix is Q_{k-1}^j .

The measurement equation of model j is:

$$z_k = H_k^j x_k^j + v_k^j \quad (3)$$

where, z_k is the measurement vector, H_k^j is the measurement matrix of model j , v_k^j is Gaussian white noise with mean 0, and its covariance matrix is R_k^j .

Based on the above state equation, measurement equation and Bayesian theory, the IMM filtering algorithm estimates the system state x_k at time k according to the measurement set $Z^k = \{z_1, \dots, z_k\}$ and the model set Ω . The purpose of smoothing is to estimate the system state $\hat{x}_{t|k}$ at the time t according to all the current observations $Z^k = \{z_1, z_2, \dots, z_k\}$, where $1 \leq t \leq k - 1$. Combining forward IMM filtering and backward IMM smoothing, the IMM Filtering and Smoothing algorithm (IMMFS) can be obtained.

Forward IMM filtering. The forward IMM filtering algorithm is described in Fig. 1, which can be divided into the following four steps^{7,8}:

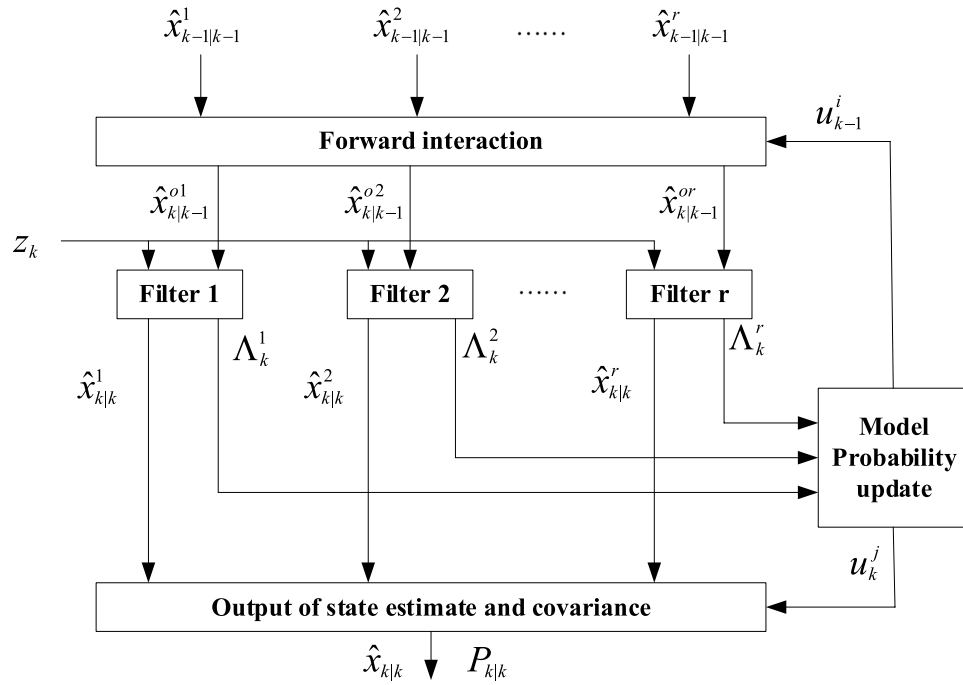


Figure 1. The forward IMM filtering algorithm.

A_Step1: forward interaction.

$$\hat{x}_{k-1|k-1}^{oj} = \sum_{i=1}^r \hat{x}_{k-1|k-1}^i u_{k-1|k-1}^{ij} \tag{4}$$

$$P_{k-1|k-1}^{oj} = \sum_{i=1}^r u_{k-1|k-1}^{ij} \{ P_{k-1|k-1}^i + [\hat{x}_{k-1|k-1}^i - \hat{x}_{k-1|k-1}^{oj}] [\hat{x}_{k-1|k-1}^i - \hat{x}_{k-1|k-1}^{oj}]^T \} \tag{5}$$

where, $\hat{x}_{k-1|k-1}^{oj}$ and $P_{k-1|k-1}^{oj}$ are the mixed state estimation of model j and the corresponding covariance matrix, respectively, $\hat{x}_{k-1|k-1}^i$ and $P_{k-1|k-1}^i$ are the Kalman estimate of model i and the corresponding covariance matrix, and $u_{k-1|k-1}^{ij}$ is the mixing probability, and its calculation formula is:

$$u_{k-1|k-1}^{ij} = \frac{p^{ij} u_{k-1}^i}{c_k^j} \tag{6}$$

where, u_{k-1}^i is the probability of model i , and the predicted probability $c_k^j = \sum_{i=1}^r p^{ij} u_{k-1}^i$ is the normalization factor.

A_Step2: Kalman filtering. For each model, Kalman filtering is performed according to the following formulas:

$$\hat{x}_{k|k-1}^{oj} = F_{k|k-1}^j \hat{x}_{k-1|k-1}^{oj} \tag{7}$$

$$P_{k|k-1}^{oj} = F_{k|k-1}^j P_{k-1|k-1}^{oj} F_{k|k-1}^{jT} + Q_{k-1}^j \tag{8}$$

$$K_k^{oj} = P_{k|k-1}^{oj} H_k^{jT} [H_k^j P_{k|k-1}^{oj} H_k^j + R_k^j]^{-1} \tag{9}$$

$$\hat{x}_{k|k}^j = \hat{x}_{k|k-1}^{oj} + K_k^{oj} [z_k - H_k^j \hat{x}_{k|k-1}^{oj}] \tag{10}$$

$$P_{k|k}^j = [I - K_k^{oj} H_k^j] P_{k|k-1}^{oj} \tag{11}$$

A_Step3: model probability update. The probability of model j at time k is:

$$u_k^j = \frac{\Lambda_k^j c_k^j}{\sum_{i=1}^r \Lambda_k^i c_k^i}$$

where, Λ_k^j obeys the Gaussian distribution, and its calculation formula is:

$$\Lambda_k^j = \frac{1}{(2\pi)^{n/2} |S_k^j|^{1/2}} \exp \left\{ -\frac{1}{2} (\gamma_k^j)^T (S_k^j)^{-1} \gamma_k^j \right\} \tag{12}$$

where,

$$\gamma_k^j = z_k - H_k^j \hat{x}_{k-1|k-1}^{oj} \tag{13}$$

$$S_k^j = H_k^j P_{k|k-1}^{oj} H_k^{jT} + R_k^j \tag{14}$$

A_Step4: output. The state estimate and covariance of the forward IMM filtering algorithm are:

$$\hat{x}_{k|k} = \sum_{j=1}^r \hat{x}_{k|k}^j u_k^j \tag{15}$$

$$P_{k|k} = \sum_{j=1}^r u_k^j \{ P_{k|k}^j + [\hat{x}_{k|k}^j - \hat{x}_{k|k}] [\hat{x}_{k|k}^j - \hat{x}_{k|k}]^T \} \tag{16}$$

Backward IMM smoothing. On the basis of the forward IMM filtering, the data is further smoothed to obtain the backward IMM smoothing algorithm. The block diagram of the IMM smoothing algorithm is shown in Fig. 2, including the following steps²⁵:

B_step1: backward interaction. The backward state interaction and corresponding covariance are:

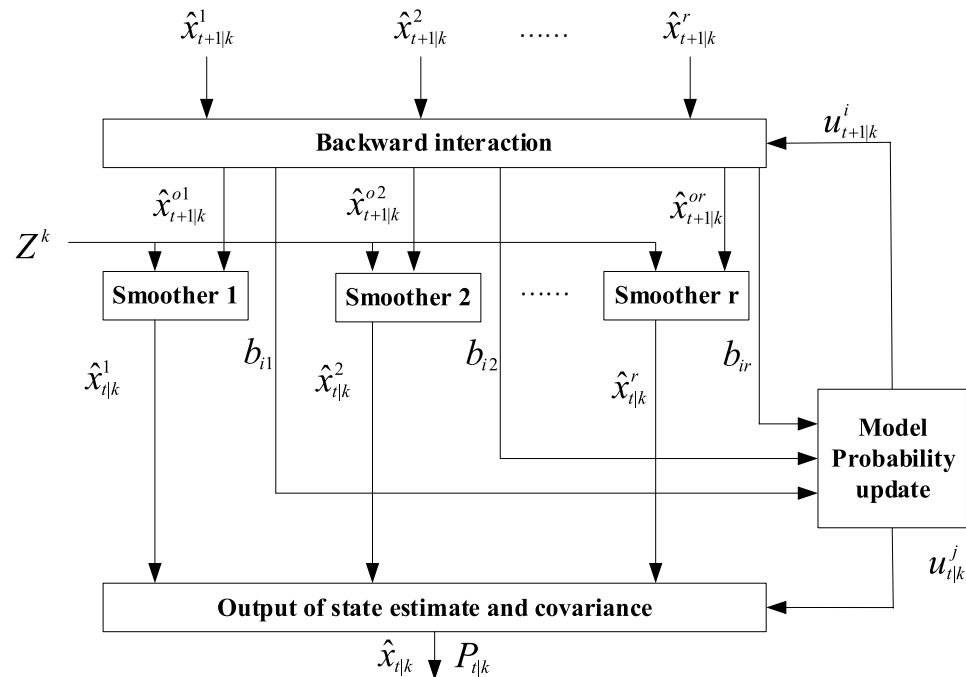


Figure 2. The backward IMM smoothing algorithm.

$$\hat{x}_{t+1|k}^{oj} = \sum_{i=1}^N u_{t+1|k}^{ij} \hat{x}_{t+1|k}^i \tag{17}$$

$$P_{t+1|k}^{oj} = \sum_{i=1}^N u_{t+1|k}^{ij} \{ P_{t+1|k}^i + [\hat{x}_{t+1|k}^i - \hat{x}_{t+1|k}^{oj}] [\hat{x}_{t+1|k}^i - \hat{x}_{t+1|k}^{oj}]^T \} \tag{18}$$

where, $u_{t+1|k}^{ij}$ is the backward mixing probability, and its calculation formula is:

$$u_{t+1|k}^{ij} = \frac{1}{d_j} b_{ij} u_{t+1|k}^i \tag{19}$$

where, $d_j = \sum_{l=1}^N b_{lj} u_{t+1|k}^l$ is the normalization factor, b_{ij} is the backward model transition probability, and its calculation formula is:

$$b_{ij} = \frac{1}{e_i} p_{ji} u_{t|k}^j \tag{20}$$

where, $e_i = \sum_{l=1}^N p_{li} u_{t|k}^l$ is the normalization factor.

*B_Step2: Kalman smoothing*²⁶. The smoothing value and the corresponding covariance are:

$$\hat{x}_{t|k}^j = \hat{x}_{t|t}^j + A_{t|k}^j (\hat{x}_{t+1|k}^j - \hat{x}_{t+1|k}^{oj}) \tag{21}$$

$$P_{t|k}^j = P_{t|t}^j + A_{t|k}^j (P_{t+1|k}^j - P_{t+1|k}^{oj}) A_{t|k}^{jT} \tag{22}$$

where, $A_{t|k}^j = P_{t|t}^j F_t^{jT} (P_{t+1|k}^{oj})^{-1}$.

B_step3: model probability update. The model probability calculation formula after smoothing is:

$$u_{t|k}^j = \sum_{i=1}^N b_{ij} u_{t+1|k}^i \tag{23}$$

B_step4: output. The state estimate and covariance of the backward IMM smoothing algorithm are:

$$\hat{x}_{t|k} = \sum_{j=1}^N u_{t|k}^j \hat{x}_{t|k}^j \tag{24}$$

$$P_{t|k} = \sum_{j=1}^N u_{t|k}^j \{ P_{t|k}^j + [\hat{x}_{t|k}^j - \hat{x}_{t|k}] [\hat{x}_{t|k}^j - \hat{x}_{t|k}]^T \} \tag{25}$$

The AVSIMMFS algorithm

The normal VSIMM algorithm. In order to avoid too many models in the model set and reduce model errors, the normal VSIMM algorithm is designed in this section. The normal VSIMM filtering algorithm includes multiple model subsets, each model subset is independently used in parallel, and the estimation result of the model subset with the highest probability is selected as the final estimated state output. The block diagram of the normal VSIMM algorithm is shown in Fig. 3, which includes the following steps:

C_step1: parallel independent IMM filtering. For different model subsets, run the IMM algorithm independently to obtain the estimated state value $\hat{x}_{k|k}(n)$ and the corresponding covariance $P_{k|k}(n)$, and to obtain the model probability $u_k^i(n)$, likelihood function $\Lambda_k^i(n)$ and predicted probability $c_k^i(n)$ in each model subset, where n is the number of the model subset, $1 \leq n \leq N$, i is the number of the models in the model subset.

C_step2: calculation of the likelihood function for model subsets.

$$p(z_k | Z^{k-1}, \Pi_k(n)) = \sum_i p(z_k | Z^{k-1}, M_k^i, \Pi_k(n)) p(M_k^i | Z^{k-1}, \Pi_k(n)) = \sum_i \Lambda_k^i(n) c_k^i(n) \tag{26}$$

where, $\Pi_k(n)$ represents the n th model subset, $\Lambda_k^i(n)$ is calculated by A_Step3, and $c_k^i(n)$ is calculated by A_Step1. Let $C_k(n) = p(z_k | Z^{k-1}, \Pi_k(n))$, then we have:

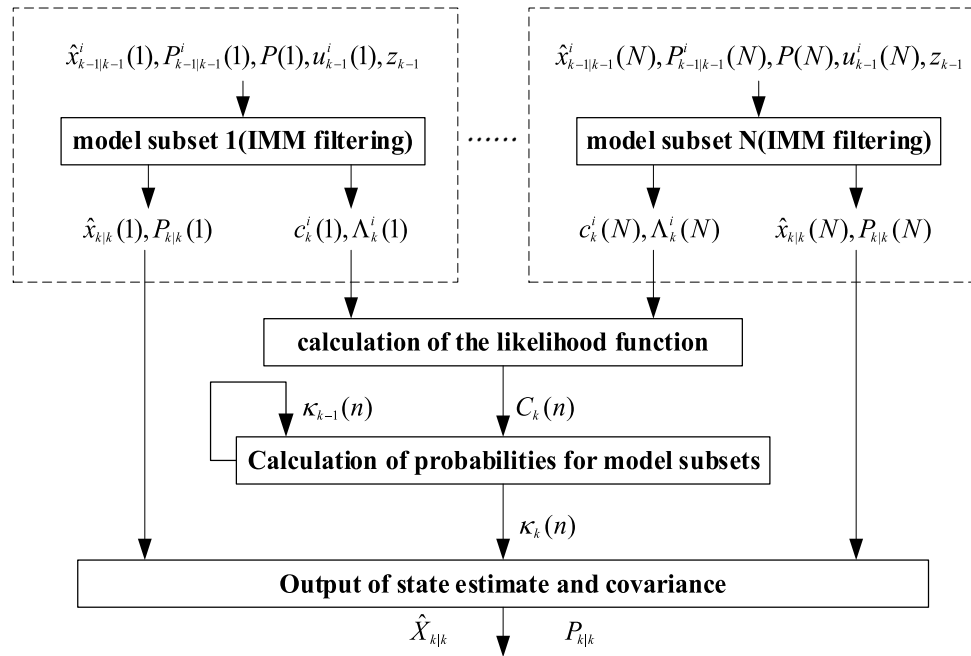


Figure 3. The NVSIMM algorithm.

$$C_k(n) = \sum_i \Lambda_k^i(n) c_k^i(n) \tag{27}$$

C_step3: calculation of probabilities for model subsets. Denote the model subset probability $p(\Pi_k(m)|Z^k) = \kappa_k(n)$, then we have:

$$\kappa_k(n) = \frac{p(z_k|Z^{k-1}, \Pi_k(n))p(\Pi_k(n)|Z^{k-1})}{\sum_n p(z_k|Z^{k-1}, \Pi_k(n))p(\Pi_k(n)|Z^{k-1})} = \frac{p(z_k|Z^{k-1}, \Pi_k(n))p(\Pi_k(n)|Z^{k-1})}{p(z_k|Z^{k-1})} \tag{28}$$

where, $p(z_k|Z^{k-1}, \Pi_k(n))$ has been calculated in *C_step2*, the denominator is the normalization factor, and $p(\Pi_k(n)|Z^{k-1})$ according to the full probability formula is:

$$p(\Pi_k(n)|Z^{k-1}) = \sum_m p(\Pi_k(n)|\Pi_{k-1}(m), Z^{k-1})p(\Pi_{k-1}(m)|Z^{k-1}) \tag{29}$$

where $\kappa_{k-1}(m) = p(\Pi_{k-1}(m)|Z^{k-1})$ is the model subset probability at the previous moment. Applying the Markov property, it can be obtained that the model subset transition probability is independent of the observed value, namely:

$$p(\Pi_k(n)|\Pi_{k-1}(m), Z^{k-1}) = p(\Pi_{k+1}(n)|\Pi_k(m)) = \sum_j \prod_i p_{ij} \tag{30}$$

C_step4: output. The IMM estimation result of the model subset with the largest probability is selected as the final state estimation output. Firstly, the number of the model subset with the highest probability is calculated as follows:

$$n_m = \max_n \{\kappa_k(n)\} \tag{31}$$

Thus, the final state estimation and corresponding covariance can be obtained as follows:

$$\hat{x}_{k|k} = \hat{x}_{k|k}(n_m) \tag{32}$$

$$P_{k|k} = P_{k|k}(n_m) \tag{33}$$

The AVSIMM algorithm. Assuming that the number of models in the model subset in the normal VSIMM algorithm is L , so C_r^L model subsets can be obtained by permutation and combination. However, in order to

reduce the computational cost, only $r - 1$ model subsets are usually constructed to cover all models (denoted as the original model subset). These $r - 1$ model subsets may not be the model subsets that best match the target maneuvering model. Therefore, in order to ensure that the calculation amount is not excessively increased, and to improve the matching degree between the model subsets and the target actual maneuvering model, a new model subset is constructed based on all the original model subsets, so we get the AVSIMM algorithm. The AVSIMM algorithm consists of the following steps:

D_step1: parallel independent IMM filtering. Operate on all original model subsets in the same way as C_step1.

D_step2: calculation of the likelihood function for model subsets. Operate on all original model subsets in the same way as C_step2.

D_step3: calculation of probabilities for model subsets. Operate on all original model subsets in the same way as C_step3.

D_step4: build a new model subset. Find the L model subsets with the highest probability according to the following formula:

$$\{\Pi_{k+1}(m_1), \dots, \Pi_{k+1}(m_L)\} = \max_L \{p(\Pi_{k+1}(1)|Z^k), \dots, p(\Pi_{k+1}(N)|Z^k)\} \quad (34)$$

where, the symbol \max_L is used to find the model subsets with the 1st, 2nd, ..., L th largest probability according to the set $\{p(\Pi_{k+1}(1)|Z^k), \dots, p(\Pi_{k+1}(N)|Z^k)\}$, which means $p(\Pi_{k+1}(m_L)|Z^k) \leq \dots \leq p(\Pi_{k+1}(m_1)|Z^k)$. Then, the model with the highest probability is selected from each model subset to form a new model subset. If the new model subset has the same model, or is the same as the original model subset, skip to D_step8; otherwise, continue to the next step.

D_step5: parallel independent IMM filtering. Operate on all original model subsets and new model subset in the same way as C_step1, the sum of the number of original model subsets and new model subsets is denoted by $N' = N + L$.

D_step6: calculation of the likelihood function for model subsets. Operate on all original model subsets and new model subset in the same way as C_step2.

D_step7: calculation of probabilities for model subsets. Operate on all original model subsets and new model subset in the same way as C_step3.

D_step8: output. The procedure for this step is the same as C_step4.

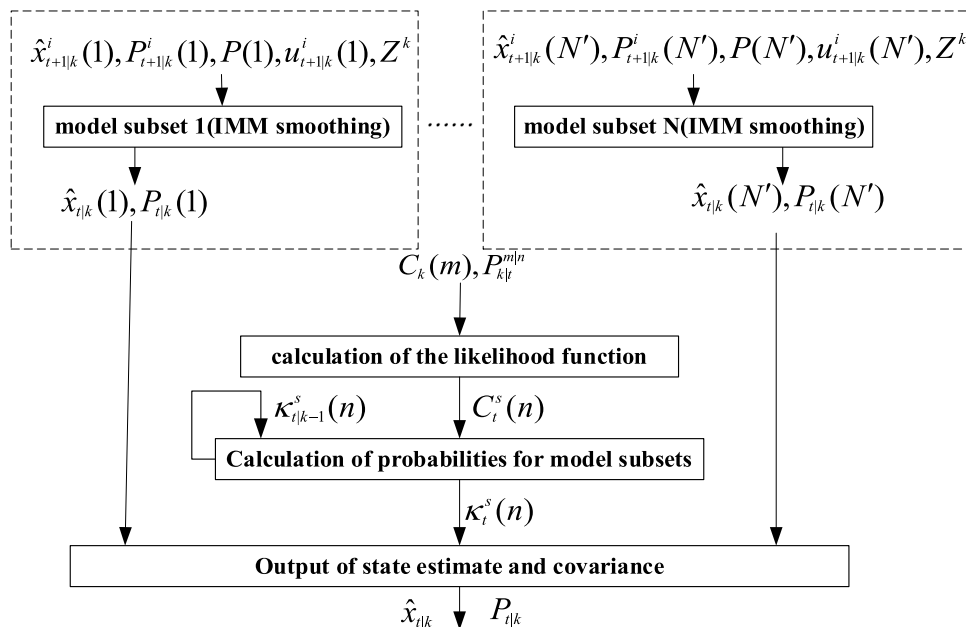


Figure 4. Backward smoothing process.

The AVSIMMFS algorithm. In this section, the AVSIMMFS algorithm can be obtained by further backward smoothing for the data obtained by forward AVSIMM filtering. The smoothing process is shown in Fig. 4, including the following steps:

E_step1: parallel independent IMM smoothing. N' original model subsets and new model subsets are operated independently, and the backward IMM smoothing algorithm was run respectively to obtain the smoothing estimate $\hat{x}_{t|k}(n)$ and the corresponding covariance matrix $P_{t|k}(n)$.

E_step2: Calculation of the likelihood function for model subsets. Applying the full probability formula and the Markov property, the likelihood function of the model subset can be obtained as:

$$\begin{aligned}
 p(z_k|Z^{k-1}, \Pi_t(n)) &= \sum_m p(z_k|Z^{k-1}, \Pi_k(m), \Pi_t(n))p(\Pi_k(m)|Z^{k-1}, \Pi_t(n)) \\
 &= \sum_m p(z_k|Z^{k-1}, \Pi_k(m))p(\Pi_k(m)|Z^{k-1}, \Pi_t(n)) \\
 &= \sum_m p(z_k|Z^{k-1}, \Pi_k(m))p(\Pi_k(m)|\Pi_t(n))
 \end{aligned}
 \tag{35}$$

where, $p(z_k|Z^{k-1}, \Pi_k(m))$ is the likelihood function of the model subset in the forward NVSIMM filtering process. Applying the full probability formula and the Markov property, $p(\Pi_k(m)|\Pi_t(n))$ can be transformed into:

$$\begin{aligned}
 p(\Pi_k(m)|\Pi_t(n)) &= \sum_{n_{t+1}} p(\Pi_k(m)|\Pi_{t+1}(n_{t+1}), \Pi_t(n))p(\Pi_{t+1}(n_{t+1})|\Pi_t(n)) \\
 &= \sum_{n_{t+1}} p(\Pi_k(m)|\Pi_{t+1}(n_{t+1}))p(\Pi_{t+1}(n_{t+1})|\Pi_t(n)) \\
 &= \sum_{n_{t+1}} \left(\sum_{n_{t+2}} p(\Pi_k(m)|\Pi_{t+2}(n_{t+2}))p(\Pi_{t+2}(n_{t+2})|\Pi_{t+1}(n_{t+1})) \right) p(\Pi_{t+1}(n_{t+1})|\Pi_t(n)) \\
 &= \dots \\
 &= \sum_{n_{t+1}} \left\{ \sum_{n_{t+2}} \left(\dots \sum_{n_{k-1}} p(\Pi_k(m)|\Pi_{k-1}(n_{k-1}))p(\Pi_{k-1}(n_{k-1})|\Pi_{k-2}(n_{k-2})) \dots \right) p(\Pi_{t+2}(n_{t+2})|\Pi_{t+1}(n_{t+1})) \right\} \\
 &\quad \times p(\Pi_{t+1}(n_{t+1})|\Pi_t(n))
 \end{aligned}
 \tag{36}$$

where, $p(\Pi_{t+1}(n_{t+1})|\Pi_t(n)), \dots, p(\Pi_k(m)|\Pi_{k-1}(n_{k-1}))$ is the transition probability of the model subset, which has been calculated in C_step3, and the model subset transition probability $p(\Pi_k(m)|\Pi_{k-1}(n_{k-1}))$ is denoted by $P_{k|k-1}^{m|n_{k-1}}$, then formula (36) becomes:

$$P_{k|t}^{m|n} = \sum_{n_{t+1}} \left\{ \sum_{n_{t+2}} \left(\dots \sum_{n_{k-1}} P_{k|k-1}^{m|n_{k-1}} P_{k-1|k-2}^{n_{k-1}|n_{k-2}} \dots \right) P_{t+2|t+1}^{n_{t+2}|n_{t+1}} \right\} P_{t+1|t}^{n_{t+1}|n_t}
 \tag{37}$$

Denote $p(z_k|Z^{k-1}, \Pi_t(n))$ as $C_t^s(n)$, then Eq. (35) becomes:

$$C_t^s(n) = \sum_m C_k(m)P_{k|t}^{m|n}
 \tag{38}$$

E_step3: calculation of probabilities for model subsets. According to Bayes theorem, the model subset probability can be obtained as:

$$p(\Pi_t(n)|Z^k) = \frac{p(z_k|Z^{k-1}, \Pi_t(n))p(\Pi_t(n)|Z^{k-1})}{\sum_n p(z_k|Z^{k-1}, \Pi_t(n))p(\Pi_t(n)|Z^{k-1})}
 \tag{39}$$

where, $p(\Pi_t(n)|Z^{k-1})$ has been calculated in the smoothing process at the previous moment and is a known value. $p(z_k|Z^{k-1}, \Pi_t(n))$ is obtained in E_step2. Denoting $p(\Pi_t(n)|Z^k)$ as $\kappa_{t|k}^s(n)$, Eq. (39) can be transformed into:

$$\kappa_{t|k}^s(n) = \frac{p(z_k|Z^{k-1}, \Pi_t(n))\kappa_{t|k-1}^s(n)}{\sum_n p(z_k|Z^{k-1}, \Pi_t(n))\kappa_{t|k-1}^s(n)}
 \tag{40}$$

E_step4: output. The model subset number with the highest probability is:

$$n_m^s = \max_n \{k_{t|k}^s(n)\} \quad (41)$$

The state estimate and the corresponding covariance of AVSIMMFS are:

$$\hat{x}_{t|k} = \hat{x}_{t|k}(n_m^s) \quad (42)$$

$$P_{t|k} = P_{t|k}(n_m^s) \quad (43)$$

It is worth noting that applying the smoothing process to normal VSIMM, the normal VSIMM filtering and smoothing algorithm can also be obtained. In the normal VSIMM filtering and smoothing algorithm, the smoothing process only operates on the N original model subsets; in AVSIMMFS algorithm, the smoothing process needs to operate on the N' original model subsets and the new model subsets.

With the development of semiconductor technology, the computing speed of multi-core heterogeneous chips on aircraft will become faster and faster^{27–29}. Through the distributed computing technology of the on-board computer, different algorithms can be injected into different chip cores or kernel computing units, and then parallel computing. Therefore, complex algorithm is naturally fast. Based on the advanced hardware resources, we complete the design process in steps: Step 1: forward IMM and a backward IMM. Step 2: a model of normal VSIMM algorithm is established and applied. Step 3: based on the VSIMM algorithm, the AVSIMM algorithm is presented and improved the matching degree between the model subset and the target model. When we have completed the first step of IMM, we first conduct validity verification experiment, rather than rushing to design Step 2. Under the condition of clear verification criteria, the Step 1 of successful verification is used to design the VSIMM in Step 2: the process of connecting multiple IMM combinations in parallel and calculating the largest possible subset. The Step 2 of the design also needs to be verified first rather than rushed into the third step of the content. Similarly, the Step 2 of verifying success is used as the basis for completing the Step 3, which adds D_step4 to the Step 2. This improvement avoids a lot of repeated calculation of multiple subsets, and only needs to calculate each subset once and get a new subset, then use the new subset to perform filtering and smoothing task, which theoretically ensures the rapidity of the algorithm. Finally, the experiment is verified again to ensure the whole scheme feasibility. By independently verifying/designing each step, adding D_step4, this study ensures the complexity, feasibility and fast real-time of the algorithm scheme from both experimental and method design.

Simulation and discussion. In order to verify the effectiveness of AVSIMMFS, this section first analyzes three cases. The first and third cases are simulation data, and the second case is real data. Suppose the target maneuvers in a two-dimensional plane.

In the first case, the initial position, velocity and acceleration of the target are (300 m, 100 m), (5 m/s, 0) and (0, 0), respectively. The maneuvering parameters of the target are as follows:

- (1) CV motion in 20 s.
- (2) CT motion in 20 s, the turning angle rate is 0.157 rad/s.
- (3) CA motion in 10 s, its acceleration is $(-2 \text{ m/s}^2, 0 \text{ m/s}^2)$.
- (4) CT motion in 10 s, the turning angle rate is 0.157 rad/s.

In the second case, the initial position, velocity and acceleration of the target are (30 km, 40 km), (300 m/s, 0), (0, 0), respectively. The maneuvering parameters of the target are as follows:

- (1) Singer motion³⁰ in 30 s, its acceleration is $(-20 \text{ m/s}^2, -20 \text{ m/s}^2)$, maneuvering constant $\alpha = 1/60$.
- (2) Both CV motions in 30 s.
- (3) CA motion in 30 s, its acceleration is $(20 \text{ m/s}, 10)$.

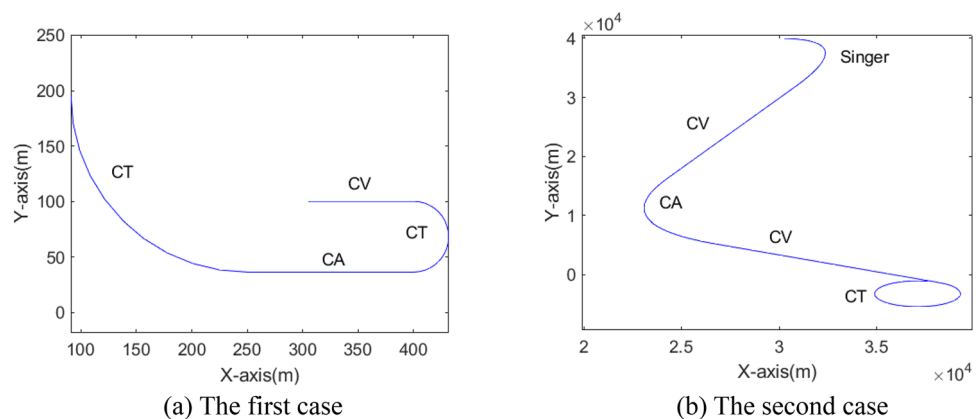


Figure 5. The target trajectories in two cases.

(4) CT motion in 30 s, the turning angle rate is 0.2 rad/s.

In the two cases, the target’s trajectory is shown in Fig. 5.

The performance of IMM, normal VSIMM, AVSIMM and AVSIMMFS is analyzed and compared by Monte Carlo method. For the first and second cases, the model set of all algorithms is {CV, CA, CT}, normal VSIMM, AVSIMM and AVSIMMFS algorithms include two model subsets, {CV, CA} and {CV, CT}, respectively. For the second case, the model set of all algorithms is {CV, CA, CT, Singer}, normal VSIMM, AVSIMM and AVSIMMFS algorithms include three model subsets, {CV, CA}, {CV, CT} and {CV, Singer}. The model transition probability matrixes of three models and four models are:

$$P_3 = \begin{pmatrix} 0.95 & 0.025 & 0.025 \\ 0.025 & 0.95 & 0.025 \\ 0.025 & 0.025 & 0.95 \end{pmatrix}, \quad P_4 = \begin{pmatrix} 0.94 & 0.02 & 0.02 & 0.02 \\ 0.02 & 0.94 & 0.02 & 0.02 \\ 0.02 & 0.02 & 0.94 & 0.02 \\ 0.02 & 0.02 & 0.02 & 0.94 \end{pmatrix}$$

In the first case and the third case, it is assumed that the standard deviations of the system process noise and the measurement noise are 1 m/s² and 50 m, respectively. The values of the covariance Q_{k-1}^j are between 0 and 0.03. It is randomly designed and calculated without affecting the result. The measured data of the second case are obtained by GPS measurement. The simulation results are shown in Figs. 6, 7, 8, 9, 10, 11, 12, and 13.

Figure 6 shows the model probability curve in the IMM algorithm. For these two cases, the model corresponding to the curve with the highest probability is the same as the target actual maneuvering model. Figure 7 shows the probability curve of model subsets in normal VSIMM, in which the model subset with the largest model probability always includes the model that matches the actual maneuvering model of the target.

It can be seen from Fig. 8 that the model subset probability curve in AVSIMM is roughly the same as that in Fig. 7, and the main difference between AVSIMM and normal VSIMM lies in the AVSIMM algorithm builds a new model subset. From Fig. 8a, it can be seen that AVSIMM constructs a new model subset {CA, CT} when the target performs CT maneuver around 50–60 s. It can be seen from Fig. 8b that a new model subset {CA, Singer} is constructed when the target performs Singer maneuver in 0–30 s and CA maneuver in 60–90 s. For all of the above cases, the probability of the newly constructed model subset is greater than that of the original model

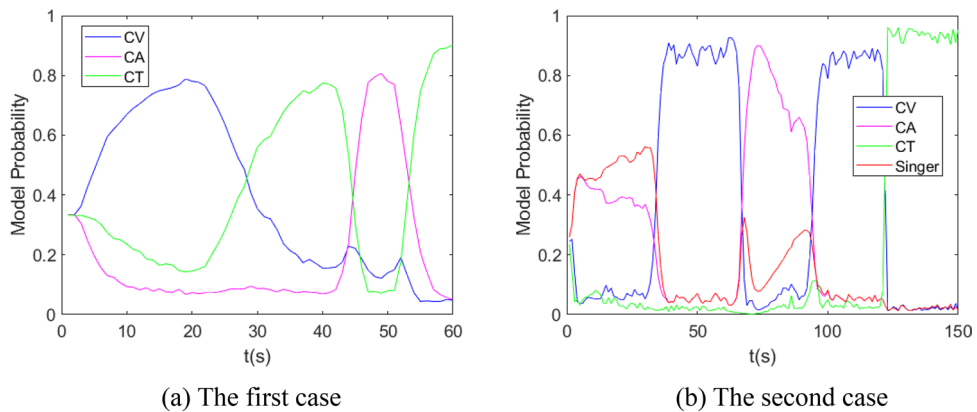


Figure 6. Model probability in IMM algorithm.

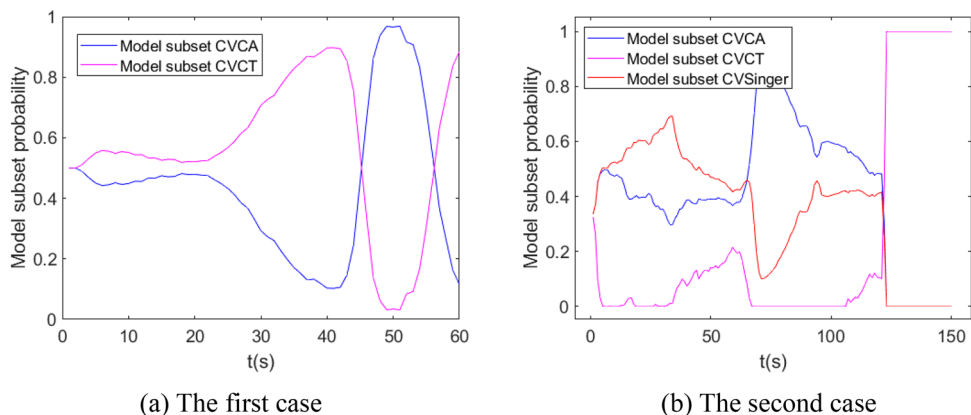


Figure 7. Model subset probability in normal VSIMM algorithm.

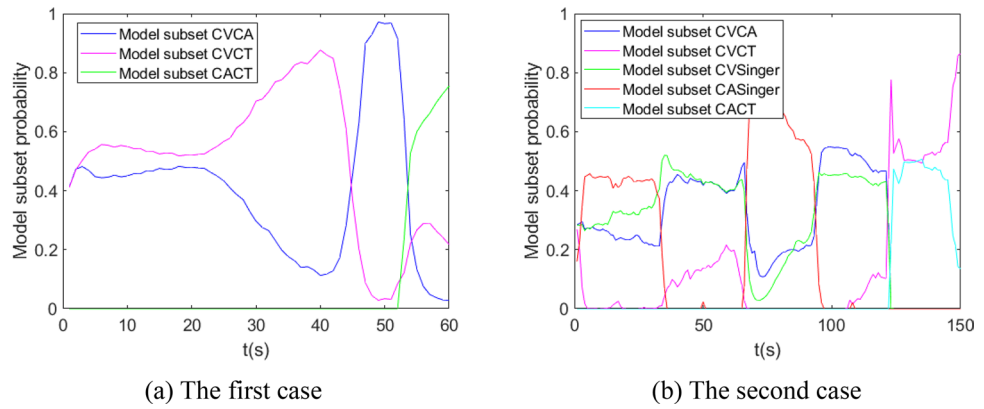


Figure 8. Model subset probability in AVSIMM algorithm.

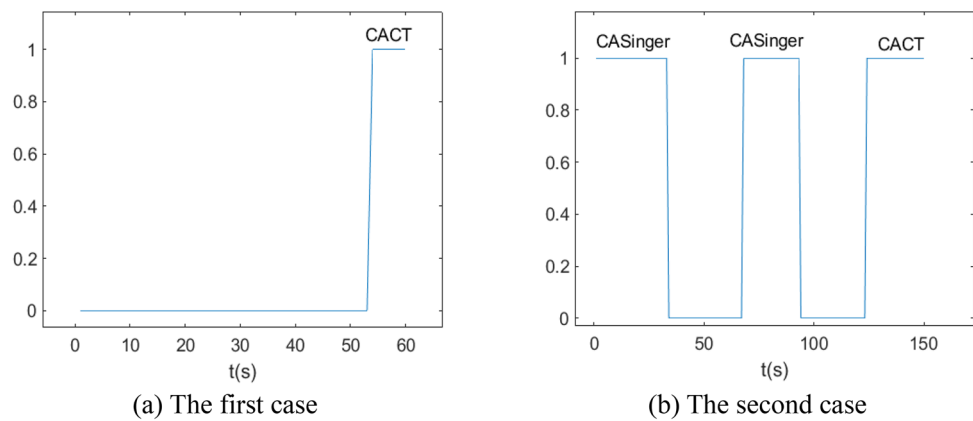


Figure 9. Whether a new model subset CAVT is built in AVSIMM.

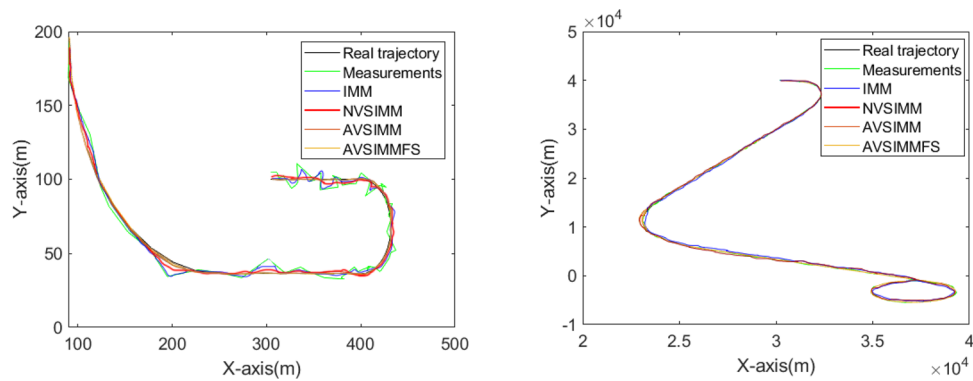


Figure 10. Target trajectory tracking effect.

subset, because the newly constructed model subset matches the target actual maneuvering model more closely. However, for the third case, when the target performs CT maneuver around 120–150 s, a new model subset {CA, CT} is constructed. At this time, the probability of the new model subset {CA, CT} is less than the probability of the original model subset {CV, CT}, so the estimation of the new model subset will not be the final estimation of the AVSIMM algorithm. Figure 9 depicts the moment when the new model subset is constructed. Analyzing this figure leads to the same conclusion as Fig. 8.

As can be seen from Fig. 10, IMM, normal VSIMM, AVSIMM and AVSIMMFS all have good target tracking capabilities, and the differences among them are reflected in Figs. 11, 12, and 13.

Even considering only the results of Fig. 10b, Table 1 still shows the difference in experimental results. In fact, due to the larger scale, the true errors are even larger than that of Fig. 10a. Table 1 is as follows:

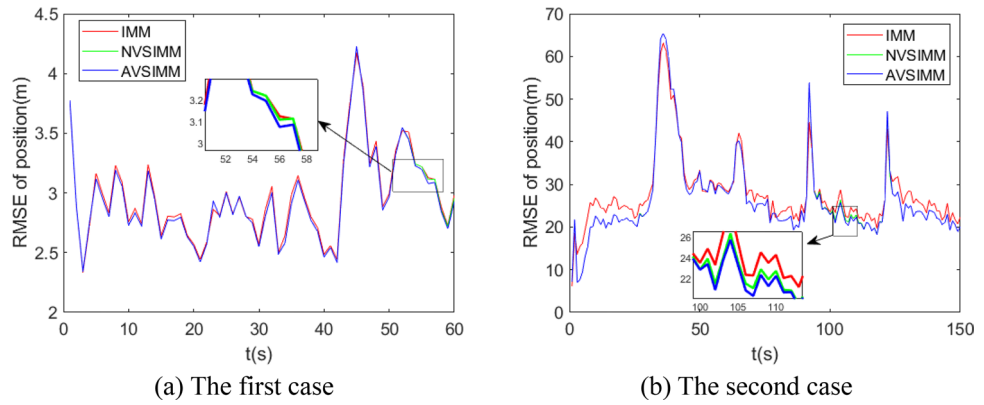


Figure 11. Root mean square error (RMSE) of position tracking.

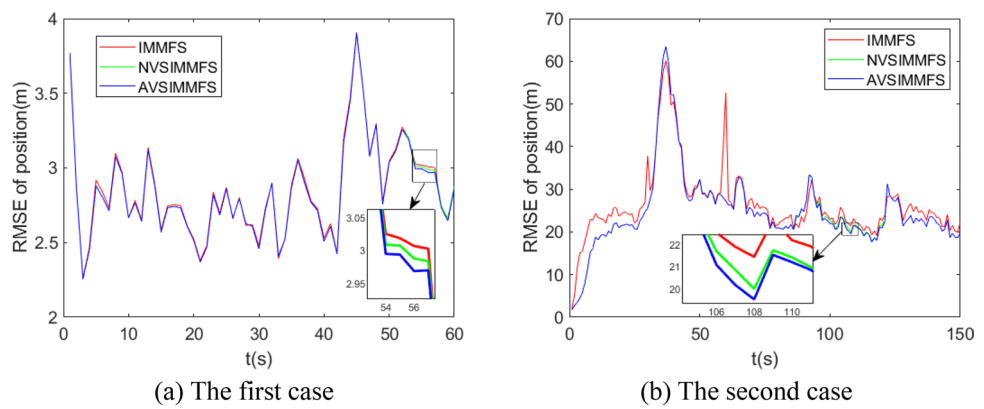


Figure 12. RMSE of position for IMMFS, normal VSIMMFS and AVSIMMFS.

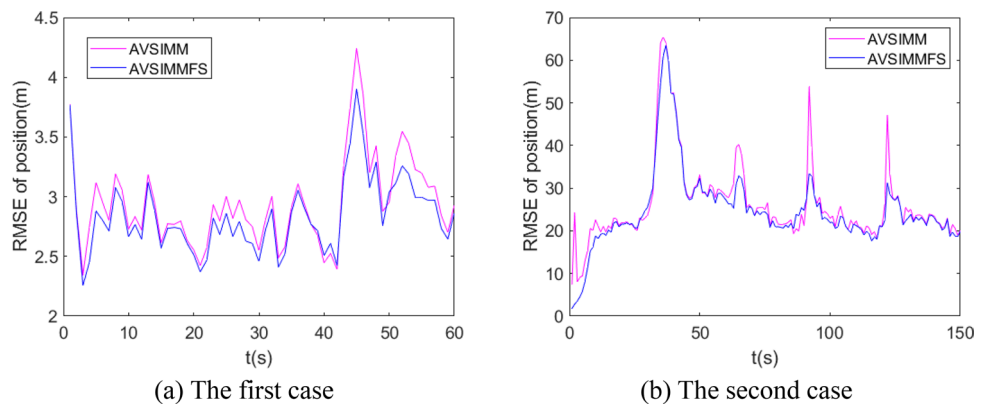


Figure 13. RMSE of position for AVSIMM and AVSIMMFS.

In Table 1, e_{av} and e_{max} are the average error and maximum error respectively. It can be seen from the table that the error differentiation in different algorithm design stages in Fig. 10 indicates that the AVSIMMFS designed at last has the best performance based on the previous design. This ensures that the readers can distinguish between the filtering and smoothing capabilities in Fig. 10.

Figures 11, 12, and 13 shows the RMSE of different algorithms for target position tracking. From Fig. 11, it can be seen that the RMSE of normal VSIMM and AVSIMM is smaller than that of IMM, and only for a short time when the target maneuver mode is changed, the performance of normal VSIMM and AVSIMM will decrease. The performance of normal VSIMM and AVSIMM is very close. Main difference between the two is in the time

	IMM	NVSIMM	AVSIMM	AVSIMMFS
e_{av}/m				
Case 1	5.7	1.4	0.83	0.42
Case 2	49	17	13	3
e_{max}/m				
Case 1	9.8	2.9	1.5	0.51
Case 2	241	58	20	7

Table 1. Comparative analysis of tracking errors in different algorithm design stages.

period when the new model subset is built. As can be seen from the enlarged area in Fig. 11, the performance of AVSIMM is slightly better than that of normal VSIMM, because the new constructed model subset better matches the actual maneuvering model of the target. ‘NVSIMM’ means normal VSIMM.

Figure 12 compares the tracking performance of IMMFS, normal VSIMMFS and AVSIMMFS. It can be seen from the figure that the RMSE of AVSIMMFS and normal VSIMMFS for target tracking is less than the RMSE of IMMFS. Similar to the results in Fig. 11, the performance of AVSIMMFS and normal VSIMMFS is similar, but the main difference between them is that AVSIMMFS performs slightly better than normal VSIMMFS in the time period when the new model subset is built. Figure 13 analyzes and compares AVSIMM and AVSIMMFS. It can be seen from the figure that the RMSE of AVSIMMFS for position estimation is smaller than that of AVSIMM, so the AVSIMMFS has better performance.

To further demonstrate the superiority of AVSIMMFS algorithm. VSIMMFS is compared with Maneuvering Target Tracking based on Deep Reinforcement Learning (MTDRL)³¹, Particle Filter (PF), Particle Swarm Optimization algorithm-based PF (PSO-PF)³², Chaos PSO-PF (CPSO-PF)^{33,34}. In order to ensure the validity of the comparison, the simulation parameters are designed according to^{31,34}, and the trajectories of the target is shown in Fig. 14. Figure 15 analyzes and compares the tracking performance of the above algorithm. It can be

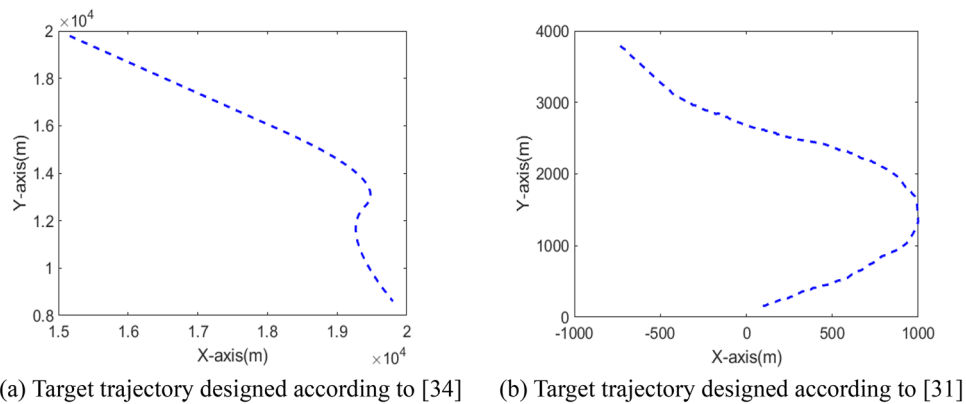


Figure 14. Target trajectories.

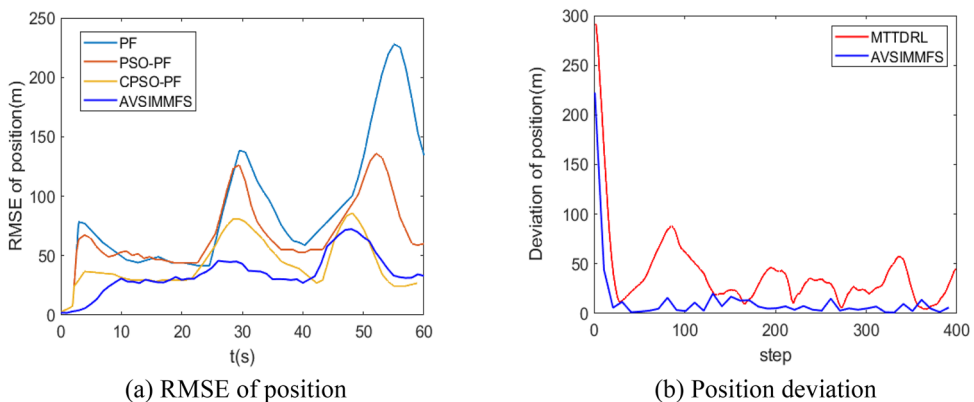


Figure 15. Performance comparison of algorithms.

	AVSIMMFS	MTTDRL	PF	PSO-PF	CPSO-PF
Average time consuming	0.060 s	0.180 s	0.493 s	0.236 s	0.221 s
Memory footprint	320 MB	431 MB	94 MB	249 MB	397 MB

Table 2. Comparative analysis of algorithm resource consumption.

seen from Fig. 15a that the RMSE of AVSIMMFS is smaller than that of PF, PSO-PF and CPSO-PF. As can be seen from Fig. 15b, compared with MTTDRL algorithm, AVSIMMFS has smaller position deviation when tracking the target. In conclusion, AVSIMMFS has better tracking performance than other algorithms mentioned above.

The simulations of AVSIMMFS are implemented in a MATLAB environment, and the main configuration of the computer is 2.1 GHz CPU and 16 GByte RAM. When the smoothing interval is 150 steps, the time required for the smoothing process is about 0.06 s. Therefore, computers with better hardware performance will greatly improve the real-time performance of the algorithm. Under the same hardware configuration and smoothing step size, the computational cost comparison between the proposed algorithm and the existing algorithms is shown in Table 2. In terms of computation time and memory footprint, AVSIMMFS has the lowest computation cost: the memory footprint is in the median of existing algorithms, but the computation time is significantly less. Combined with the significant performance advantages, AVSIMMFS is clearly the best overall performance.

Conclusion

In order to improve the target tracking performance, an AVSIMMFS algorithm is presented in this paper. This method builds a new model subset on the basis of the original model subset, which can improve the matching degree between the model subset and the actual maneuvering model of the target without excessively increasing the computational cost, so as to improve the tracking accuracy of the target. Finally, the filtering data of the AVSIMM algorithm is smoothed to obtain the AVSIMMFS algorithm, which further improves the target tracking accuracy. By using real data and simulation data to compare with IMM, normal VSIMM, AVSIMM, PF, PSO-PF, CPSO-PF and MTTDRL algorithms, AVSIMMFS has the best tracking performance. Compared with IMM and normal VSIMM, AVSIMMFS can construct a new model subset based on the original model subsets, which improves the matching accuracy of the model subset, thus improving the tracking accuracy. Due to the smoothing of the filtered data, the root mean square error of AVSIMMFS for position tracking is less than that of AVSIMM. In addition, compared with other types of filtering algorithms such as PF, PSO-PF, CPSO-PF and MTTDRL, AVSIMMFS also has the smallest tracking error. In terms of calculation cost, AVSIMMFS only needs about 0.06 s to obtain the results, showing good real-time performance.

Data availability

The datasets generated and/or analysed during the current study are not publicly available due the Chinese military's strictest secrecy policy on hypersonic missile technology. but are available from the corresponding author on reasonable request.

Received: 8 March 2023; Accepted: 19 July 2023

Published online: 10 August 2023

References

1. Cao, W. *et al.* A novel adaptive state of charge estimation method of full life cycling lithiumion batteries based on the multiple parameter optimization. *Energy Sci. Eng.* 7(5), 1544–1556 (2019).
2. Wang, S. L. *et al.* An adaptive working state iterative calculation method of the power battery by using the improved Kalman filtering algorithm and considering the relaxation effect. *J. Power Sources* 428, 67–75 (2019).
3. Wang, S. *et al.* A novel safety assurance method based on the compound equivalent modeling and iterate reduce particle-adaptive Kalman filtering for the unmanned aerial vehicle lithium-ion batteries. *Energy Sci. Eng.* 8(5), 1484–1500 (2020).
4. Wang, S. *et al.* A critical review of improved deep learning methods for the remaining useful life prediction of lithium-ion batteries. *Energy Rep.* 7, 5562–5574 (2021).
5. Xu, C., Wang, X., Duan, S. & Wan, J. Spatial-temporal constrained particle filter for cooperative target tracking. *J. Netw. Comput. Appl.* 176, 102913 (2021).
6. Wang, S., Jin, S., Deng, D. & Fernandez, C. A critical review of online battery remaining useful lifetime prediction methods. *Front. Mech. Eng.* 7, 719718 (2021).
7. Song, R., Fang, Y. & Huang, H. Estimation of automotive states based on optimized neural networks and moving horizon estimator. *IEEE/ASME Trans. Mech.* 1, 1–12. <https://doi.org/10.1109/TMECH.2023.3262365> (2023).
8. Guo, X., Sun, L., Wen, T., Hei, X. & Qian, F. Adaptive transition probability matrix-based parallel IMM algorithm. *IEEE Trans. Syst. Man Cybern. Syst.* 51(5), 2980–2989 (2021).
9. Gao, H., Qin, Y., Hu, C., Liu, Y. & Li, K. An interacting multiple model for trajectory prediction of intelligent vehicles in typical road traffic scenario. in *IEEE Transactions on Neural Networks and Learning Systems* (2021).
10. Li, Y. F. & Bian, C. J. Object tracking in satellite videos: A spatial-temporal regularized correlation filter tracking method with interacting multiple model. *IEEE Geosci. Remote Sens. Lett.* 19, 5791–5802 (2022).
11. Fan, X. X., Wang, G., Han, J. C. & Wang, Y. H. Interacting multiple model based on maximum correntropy Kalman filter. *IEEE Trans. Circuits Syst. II* 68(8), 3017–3021 (2021).
12. Hu, D., Chen, Z. & Yin, F. L. Information weighted consensus with interacting multiple model over distributed networks. *IEEE Trans. Circuits Syst. II* 68(4), 1537–1541 (2021).
13. Qiu, J. *et al.* Centralized fusion based on interacting multiple model and adaptive Kalman filter for target tracking in underwater acoustic sensor networks. *IEEE Access* 7, 25948–25958 (2019).

14. Lan, J., Li, X. R., Jilkov, V. P. & Mu, C. Second-order Markov chain based multiple-model algorithm for maneuvering target tracking. *IEEE Trans. Aerosp. Electron. Syst.* **49**(1), 3–19 (2013).
15. Xu, L., Li, X. R. & Duan, Z. Hybrid grid multiple-model estimation with application to maneuvering target tracking. *IEEE Trans. Aerosp. Electron. Syst.* **52**(1), 122–136 (2016).
16. Wang, Q. H., Li, P. F. & Fan, E. Target classification aided variable-structure multiple-model algorithm. *IEEE Access* **8**, 147692–147702 (2020).
17. Cosme, L. B., Caminhas, W. M. & D'Angelo, M. F. A novel fault-prognostic approach based on interacting multiple model filters and fuzzy systems. *IEEE Trans. Ind. Electron.* **66**(1), 519–528 (2019).
18. Xu, L. F., Li, X. R. & Duan, Z. S. Hybrid grid multiple-model estimation with application to maneuvering target tracking. *IEEE Trans. Aerosp. Electron. Syst.* **52**(1), 126–136 (2016).
19. Zhang, J., Zhang, X. & Song, J. Maneuvering target tracking algorithm based on multiple models in radar networking. in *2019 International Conference on Control, Automation and Information Sciences (ICCAIS)*, 1–6. (IEEE, 2019).
20. Li, S., Fang, H. & Shi, B. Remaining useful life estimation of Lithium-ion battery based on interacting multiple model particle filter and support vector regression. *Reliab. Eng. Syst. Saf.* **210**, 107542 (2021).
21. Messing, M., Rahimifard, S., Shoa, T. & Habibi, S. Low temperature, current dependent battery state estimation using interacting multiple model strategy. *IEEE Access* **9**, 99876–99889 (2021).
22. Al-Nuaimi, M. N., Al Sawafi, O. S., Malik, S. I. & Al-Marouf, R. S. Extending the unified theory of acceptance and use of technology to investigate determinants of acceptance and adoption of learning management systems in the post-pandemic era: A structural equation modeling approach. *Interact. Learn. Environ.* **1**, 1–27 (2022).
23. Kong, X. Y. *et al.* Adaptive dynamic state estimation of distribution network based on interacting multiple model. *IEEE Trans. Sustain. Energy* **13**(2), 643–652 (2022).
24. Sun, L., Zhang, J., Yu, H., Fu, Z. & He, Z. Tracking of maneuvering extended target using modified variable structure multiple-model based on adaptive grid best model augmentation. *Remote Sens.* **14**(7), 1613 (2022).
25. Nadarajah, N., Tharmarasa, R., McDonald, M. & Kirubarajan, T. IMM forward filtering and backward smoothing for maneuvering target tracking. *IEEE Trans. Aerosp. Electron. Syst.* **48**(3), 2673–2678 (2012).
26. Sun, W. W., Wu, Y. & Lv, X. Y. Adaptive neural network control for full-state constrained robotic manipulator with actuator saturation and time-varying delays. *IEEE Trans. Neural Netw. Learn. Syst.* **33**(8), 3331–3342 (2022).
27. Li, X. R. & Jilkov, V. P. Survey of maneuvering target tracking. Part I. Dynamic models. *IEEE Trans. Aerosp. Electron. Syst.* **39**(4), 1333–1364 (2003).
28. Zhu, B. *et al.* Millimeter-wave radar in-the-loop testing for intelligent vehicles. *IEEE Trans. Intell. Transp. Syst.* **23**(8), 11126–11136 (2022).
29. Su, J., Li, Y. A. & Ali, W. Underwater passive manoeuvring target tracking with isogradient sound speed profile. *IET Radar Sonar Navig.* **16**(9), 1415–1433 (2022).
30. Dulek, B. Corrections and comments on “an efficient algorithm for maneuvering target tracking” [corrections and comments]. *IEEE Signal Process. Mag.* **39**(4), 138–139 (2022).
31. Wang, X., Cai, Y. L., Fang, Y. Z. & Deng, Y. F. Intercept strategy for maneuvering target based on deep reinforcement learning. in *40th Chinese Control Conference (CCC)*, 3547–3552. (IEEE, 2021).
32. Feng, Q. *et al.* Multiobjective particle swarm optimization algorithm based on adaptive angle division. *IEEE Access* **7**, 87916–87930 (2019).
33. Wang, E. *et al.* Fault detection and isolation in GPS receiver autonomous integrity monitoring based on chaos particle swarm optimization-particle filter algorithm. *Adv. Space Res.* **61**(5), 1260–1272 (2018).
34. Zhang, Z. *et al.* Hummingbirds optimization algorithm-based particle filter for maneuvering target tracking. *Nonlinear Dyn.* **97**(2), 1227–1243 (2019).

Acknowledgements

The project was supported by the project of quality technology of CASIC Third Research Institute under Grant JR202210.

Author contributions

K.-Y.H., Y.C., and C.Y. conceived the research, performed the research and wrote the manuscript. K.-Y.H., and J.W. performed the research, revised the manuscript and provided the fund support.

Competing interests

The authors declare no competing interests.

Additional information

Correspondence and requests for materials should be addressed to K.-Y.H.

Reprints and permissions information is available at www.nature.com/reprints.

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

© The Author(s) 2023