Multisensor Distributed Dynamic Programming Method for Collaborative Warning and Tracking

Peng Ni, Bo Zhang, Yafei Song, and Mingliang Zhang

Air and Missile Defense College, Air Force Engineering University, Xi'an 710051, China

Correspondence should be addressed to Bo Zhang; afeu_zhb@163.com

Received 17 March 2020; Accepted 28 April 2020; Published 15 May 2020

Academic Editor: Mauro Gaggero

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

Multisensor distributed dynamic programming for collaborative warning and tracking during antimissile combat serves to meet the tracking accuracy requirements of all ballistic targets in the battlefield under the circumstance of a limited total amount of sensor resources. This paper proposes a method of multisensor distributed dynamic programming for collaborative warning and tracking based on game theory. First, starting from the target tracking algorithm, according to the characteristics of antimissile multisensor combat planning, the box particle filter (BPF) theory capable of distributed filtering and inaccurate measurement is introduced. Using the flight phase characteristics of ballistic targets, a variable structure adaptive multimodel box-based particle filter tracking method is constructed. A box particle filter with the variable structure adaptive interacting multiple model (VSAIMM-BPF) is proposed. The method solves the continuous real-time tracking problem of the ballistic target in all the phases and achieves high tracking accuracy while reducing computational complexity. Then, the motion state of each ballistic target in combat is recursively evaluated by the filtering algorithm, and the calculated sensor information gain is used as a measure to obtain more or better sensor resources for the community of interest to track the corresponding ballistic target through the game. Ultimately, the method achieves distributed dynamic programming.

1. Introduction

Multisensor task planning is a hot topic in information fusion, which was proposed in 1970s, and firstly applied in military command and control [1]. Dressler et al. [2] realized the implementation of dynamic task and optimized control of data collection based on closed-loop control strategy. This can be regarded as the earlier application of multisensor task planning. Nash [3] studied the problem of sensor allocation in multitarget tracking based on multisensor. But the work of Nash [3] was carried out on single platform. Manyika [4] and Schmaedeke [5] pointed out that the detection of sensors can be described by posterior entropy, which is helpful for sensors’ optimized control. Then, the method of information-driven sensor task planning gets rapid development. The necessity and advantage of the information-driven method were claimed by Hintz and McIntyre [6].

Due to its sensitivity, there is little public report on the sensor allocation in antiballistic battle. Research on such topics mainly focuses on two aspects: (1) the allocation and optimization of detection and tracking ability, based on information theory, operation theory, and filter and intelligent optimization method; (2) target recognition and guidance technology, containing intelligent recognition, the choice of guidance method, and target indication.

Above analysis indicates that most researchers pay much attention to the application of task planning based on a specified sensor platform or a specified stage of the task. This is helpful for the research on multisensor task planning. But the battle field of antiballistic missile is full of high complexity, strong antagonism, and increasing dynamics. These characteristics may bring great challenge to the research of multisensor task planning. This motivates us to develop a new method to solve the problem of target tracking, especially for the ballistic missiles with great threat. The main contribution of this paper is the presentation of the new box-particle filter with variable structure adaptive interacting multiple model. The proposed method can improve the track
accuracy of ballistic missiles. Moreover, the time complexity of the tracking is also reduced, which is helpful for the timely requirement of antimissile.

1.1. Basic Process Analysis. Distributed dynamic programming for collaborative warning and tracking is divided into warning and coordination [1–12]. The main concerns of both are the stability and optimization of target tracking performance. Both can use the optimal metric of certain specific characteristics, such as detection probability, intercept probability, and tracking accuracy, as the objective function, and achieve the dynamic target sensor adjustment with the distributed task planning method. Therefore, they can be incorporated into a unified framework for processing.

Regarding the combat operational process, the warning and tracking process of antimissile operations is mainly completed by space-based infrared satellites, P-band long-range warning radars, and ground/sea-based X-band multifunction phased-array radars. Early-stage warning is mainly completed by space-based infrared satellites. The secondary warning is completed by space-based infrared satellites and P-band long-range warning radar. P-band long-range warning radars and ground/sea-based X-band multifunction phased-array radars are responsible for detection and tracking and involve the entire ballistic flight. The specific tracking time-space windows are constrained by the radar deployment locations. The warning and tracking processes are integrated and involve coordination and cooperation between multiple sensors. At the same time, for the defendants, the threat of each ballistic target is deadly. If the warning is not made in time or the target is missed, the consequences are extremely serious. The warning targets have certain randomness, and the time, location, quantity, and type of occurrence are difficult to predict; therefore, all the warning tasks are randomly and dynamically generated. The countermeasures taken by the ballistic target, and the performance limitations of the system’s sensor resources, result in certain uncertainty in tracking the target. This requires the improvement of the target detection probability (warning capability) of emergent/lost targets to ensure continuous and stable tracking of the tracked ballistic target. In summary, warning and tracking, the two different target detection processes, are closely related. It is necessary to consider both the target tracking accuracy and target detection probability according to the detection state (warning or tracking) of the target and to combine them into the objective function and adaptively allocate the limited sensor resources using the adaptive optimization algorithm.

1.2. Analysis of Key Factors. Distributed warning dynamic programming and distributed tracking dynamic programming are two closely related stages of a common problem (the target is in the warning stage or tracking stage; these are the two different stages of the target detection process). This is a complex optimal combination problem. In order to achieve timely warning of new/lost targets while ensuring continuous tracking of tracked targets, it is necessary to analyze and solve all targets together; namely, the purpose of multisensor distributed dynamic programming for collaborative warning and tracking in an antimissile combat is to dynamically adjust the sensors under the constraints of limited sensor resources and the required tracking accuracy when multiple ballistic targets appear dynamically. Thus, we can complete proper detection (warning and tracking) behavior to optimize overall warning and tracking performance. Specifically, the following need to be addressed.

1.2.1. Filtering Algorithm Planning Requirements. Tracking filtering is the most basic element of a detection and tracking system [1]. In antimissile combat operations, the measured values (radial distance, pitch angle, and azimuth) by the sensors are nonlinearly related to the target state, and the tracking system is often in a situation of unknown synchronous deviation or system delay in a complex combat environment. The state noise and the measurement noise are also non-Gaussian. Therefore, this is a nonlinear filtering problem. Currently, the commonly used filtering algorithms are the extended Kalman filter (EKF) algorithm [13], the unscented Kalman filter (UKF) algorithm [14, 15], and the related improved algorithms. The improved algorithms include the point estimation algorithm, represented by the organic combination of the Kalman filter and the nonlinear system linear-approximation method, and the density estimation algorithm, represented by the particle filter (PF) algorithm and ensemble Kalman filter (EnKF) algorithm. The former can handle multivariable, time-varying systems and nonstationary random signals. However, for the problem of non-Gaussian noise, its tracking performance will be significantly deteriorated and even lead to divergence. The latter is not limited by linearization error or Gaussian noise assumption and can be used in nonlinear non-Gaussian systems; however, due to the increasing amount of computing with increasing performance, its real-time performance is poor. Therefore, it is necessary to balance the accuracy, measurement efficiency, and robustness of the algorithm in order to meet the specific requirements of the multisensor task planning system for antimissile combat and to develop an appropriate filtering algorithm.

1.2.2. Appropriate Criteria. On the basis of the filtering algorithm, it is necessary to judge, according to the merits of the current tracking, and then decide whether to carry out dynamic programming. The first step is to resolve what criteria are used to guide the process of dynamic programming:

\[
C_i = \arg \max_y f(x_i), \\
\ell_j(x_i) \leq 0, \\
s.t. \quad y \in S, \\
j = 1, 2, \ldots, n.
\] (1)

As shown in equation (1), in terms of the mathematical model, the process of dynamic programming is as follows: for the target \( i \), under the constraint condition find the
1.2.3. Mathematical Methods for Effectively Solving Conflict and Cooperation Problems. From the perspective of target tracking, it is always the case that more sensors used to track targets will lead to the capture of greater amounts of information and more benefits to target tracking by enhancing the accuracy and stability of the target tracking. Meanwhile, from the perspective of the whole sensor network, the total resources for the warning and detection are limited. The more the sensor warning and detection resources are taken up by a certain target, the more limited are the sensor warning and detection resources available for other targets. Therefore, in future scenarios of large-scale, multibatch, multilevel, antisaturation attack antimissile operations, the improvement of target tracking accuracy is not the most crucial issue of task planning. How to balance the conflict between resources and objectives, under the premise of ensuring a certain degree of tracking accuracy, and the dynamic planning of limited sensor resources is the research focus of distributed dynamic programming for collaborative warning and tracking.

1.3. Planning Guidelines and Assumptions. The guidelines for multisensor distributed dynamic programming for collaborative warning and tracking mainly include the following:

(1) In terms of the defender, it is necessary to ensure that each ballistic target can be detected, the warning capability (detection probability) must not be lower than the set threshold, and priority is given to tracking targets with a high threat level

(2) In terms of the ballistic target, the performance of each target’s tracking must at least meet the requirements of task planning, and the time duration to track the target should be as long (stable and continuous) as possible

(3) In terms of sensor tracking performance, it is preferred to use sensors with high detection accuracy to detect targets

(4) In terms of the reliability and technical complexity of antimissile combat operations, it is necessary to minimize the number of sensor handovers to ensure continuous target tracking

The study of multisensor distributed dynamic programming for collaborative warning and tracking is based on the following assumptions:

(1) It is assumed that the deployment location, performance parameters, and corresponding detection and tracking coverage are known for each sensor.

(2) The ballistic target is predictable; that is, the initial information of the target can be predicted, and the target only has limited maneuver during the flight.

(3) Based on (1) and (2), when warning for a target occurs, the visualization time window of the sensor for the target (time and spatial information of the target’s coverage by the sensor) can be approximated.
(4) The handover problem limits the target transition between two sensors. During the handover, the sensor devices must coordinate the transition so that the ballistic target trajectory can be smoothly connected. This involves specific radar systems and sampling intervals. This study assumes that the sensor handover process is smooth; that is, the success rate of handover is 100%.

(5) It is assumed that the information in antimissile combat can be shared in real time; that is, the communication network can support coordinated operation of multiple sensors, ignoring the problem of communication energy consumption.

2. Game Theory and Distributed Dynamic Programming for Collaborative Warning and Tracking in Antimissile Combat

A game is an interdependent decision-making situation in which more than one party participates. In general, the elements of the game model mainly include participants, information, strategy, payout (or compensation), rationality, goals, sequence of action, results, and equilibrium. Among them, participants, strategies, and payout (or profits) are the most basic elements. Participants refer to individuals who can make decisions in the game. Participants must have certain autonomy (otherwise, they are passive participants and can only act as an environmental parameter in the game process). They achieve minimizing cost (or maximizing payout) through employing strategies in the game. In this paper, the warning/tracking community of interest corresponding to each ballistic target is considered a virtual participant that participates in the game; they seek as many sensor resources as possible for each ballistic target that is part of the cooperative combat. A strategy is a participant’s possible action or a plan, and the participant acts according to their strategy. In this paper, the strategy under the collaborative warning and tracking dynamic programming problem is that each warning/tracking community of interest provides a series of possible sensor sets according to the tracking requirements of the system for the corresponding ballistic targets. Payoff/return is a common unit of measure in game theory, describing the participants’ preferences for different game outcomes. In this paper, the improvement of the tracking accuracy of the warning/tracking community of interest under certain sensor sets in the game is used as the payoff. In addition, without loss of generality, this paper assumes that each sensor can only observe one target at a time (a sensor with the ability to observe multiple targets can be considered as consisting of multiple virtual sensors, each observing a single target).

2.1. Game Payoff in Distributed Dynamic Programming for Collaborative Warning and Tracking. According to the three elements of game theory, the determination of game payoff is the key point of the whole problem. For warning and tracking, the game payoff is the ability to improve the task quality (tracking accuracy and detection probability) obtained by using the sensor sequence scheme as the target warning and tracking tasks of the ballistic target represented by the consumer. Therefore, it is first necessary to find reasonable evaluation criteria to measure the quality of the task. Obviously, this starts with the target tracking algorithm.

In the nonlinear and non-Gaussian environment, the continuous and stable tracking of ballistic targets is particularly important, and it has been a hot subject of study. In terms of the filtering algorithm, since the motion equation and radar measurement equation of ballistic targets are usually nonlinear, tracking is a nonlinear filtering problem [17] and thus requires high precision and a stable nonlinear filtering method. Currently, the filtering algorithms used for ballistic target tracking mainly include EKF, UKF, PF, and related improved filtering algorithms. The literature shows that EKF has good tracking performance but has poor adaptability to nonlinearity and diverges easily [18]. UKF has better tracking performance, but it is more complicated in high-dimensional cases. PF can achieve a higher tracking accuracy, but it poses problems such as a large amount of computing, poor real-time performance, and particle degradation [19–21]. In summary, a ballistic target as a non-cooperative target is of high speed, high threat, and with strict real-time requirement, thus requiring an algorithm with low computational complexity and high accuracy.

Therefore, this paper uses a generalized particle filter, that is, a box particle filter (BPF) for target tracking [14–17]. Based on the phased characteristics of ballistic target flight, an interacting multimodel box particle filter is designed to solve the problem of full-phase, continuous, and stable tracking of the ballistic target. By using the BPF algorithm to achieve the tracking of the ballistic target with less cost and algorithm complexity, we strive to meet the real-time requirements of the antimissile multi-sensor task planning, in addition to ensuring desired tracking accuracy.

2.1.1. Ballistic Target Tracking Based on Variable Structure Adaptive Interacting Multimodel Box Particle Filter (VSAIMM-BPF). Ballistic targets are usually divided into three basic phases based on their flight characteristics: the active phase, free phase, and reentry phase. Because ballistic targets in different phases are subject to significantly different forces, the dynamic characteristics are different. Therefore, most researchers have conducted a lot of research on the tracking problem of a specific stage of the ballistic target and have obtained remarkable results. However, there are few studies on the tracking ballistic targets throughout all phases. The specific limitations are as follows: (1) with strong nonlinearity, it is difficult to describe the flight process of the entire ballistic trajectory with a single mathematical model. (2) It is difficult to accurately determine the specific transition time between flight phases, and the error is significantly increased when the common tracking method is used in the transition between two different motion phases, resulting in the target being easily lost. Therefore, at present,
the interacting multimodel method is mainly used to realize the tracking problem of ballistic targets at different phases \[22\]. Yu et al. \[23\] proposed that the continuous tracking algorithm can solve such problems fundamentally, and the continuous stable tracking of ballistic targets is a key problem that must be solved in task planning.

Given the above situation, this paper proposes a new IMM-BPF tracking method by integrating interacting multimodel and box particle filtering algorithms.

Variable Structure Adaptive Interacting Multiple Model Tracking Method (VSAIMM).

(1) Method Description. The basic steps of the interacting multiple model algorithm are as follows: input interaction, filtering, model probability update, and output fusion. Based on the filtering algorithm, interacting multiple model algorithms can be divided into the interacting multiple model Kalman filter (IMM-KF), interacting multiple model unscented Kalman filter (IMM-UKF), and interacting multiple model particle filter (IMM-PF).

To use the interacting multimodel algorithm, determining the target’s motion model set is the key. The evaluation performance of an interacting multimodel method depends on a large extent on the set of models used. In theory, the established motion model needs to cover all the motion patterns of the target. This causes contradictions: to improve the tracking accuracy, many models are needed to fit the motion of the target, but too many models will drastically increase the computing requirements, which leads to decreased estimation performance \[24\]. Therefore, this section uses a variable multimodel filtering method based on the irreversible phase characteristics of ballistic target flight. The method is based on the three-phase model set of ballistic target flight; the set includes an active phase model, free phase model, and reentry phase model. A time-varying model set is adopted, and the box particle filter algorithm is adopted in each filter.

(2) Transition Probability between Model Sets. According to the definition of the flight phase of the ballistic target, the process has certain irreversibility and measurability; that is, if the ballistic target is in the free phase, it cannot transfer to the active phase; also, if the ballistic target is in the active phase, it cannot transfer to the reentry phase, as shown in Figure 1. In Figure 1, \(\pi_{11}\) is the probability of remaining in the active phase, \(\pi_{12}\) is the probability of transitioning from the active phase to the free phase, \(\pi_{22}\) is the probability of remaining in the free phase, \(\pi_{23}\) is the probability of maintaining the free phase to the reentry phase, and \(\pi_{33}\) is the probability of remaining in the reentry phase. Therefore, the Markov transition probability matrix based on the flight phase characteristics is obtained as follows:

\[
\pi = \begin{bmatrix}
\pi_{11} & \pi_{12} & 0 \\
0 & \pi_{22} & \pi_{23} \\
0 & 0 & \pi_{33}
\end{bmatrix}.
\]

(3) Variable Structure Model Set. The key point of the full-phase continuous tracking algorithm is the stable and continuous tracking performance of the ballistic target at the transition of two different flight phases. At the transition, it is difficult to determine the current flight state of the ballistic target. Different from the traditional method of judging the flight phase based on motion characteristics such as target position and speed, this section uses the current model probability to determine the current flight phase:

\[
\max\{p_i\} \geq a, \quad i = 1, 2, 3, \quad (3)
\]

As shown in equation (3), \(p_i\) represents the probability that the \(i\)th model is correctly described, and \(a\) represents a parameter setting that is generally set to 0.8. In this way, the corresponding model set at each flight phase can be obtained, as shown in Table 1.

2.1.2. Interacting Multimodel Box Particle Filter Algorithm. After determining the flight phase and the corresponding set of models, the models interact through an interacting multimodel algorithm structure. For each model, a group of box particles is generated. This group of boxes is resampled after interacting input and box particle filtering and is finally subject to interacting output. This cyclic recursive propagation updates these box particles to complete the estimation of target states.

As shown in Figure 2, the steps of the interacting multimodel box particle filter algorithm are as follows:

(1) Initialization (Newborn Box Particles). When there is a large error between the target prediction and the measurement, contracted box particles after constraint propagation will degrade rapidly. Therefore, it is necessary to initialize before the prediction. The newborn box particles are replenished according to the current measurement. The specific steps are described below.

According to the target measurement \[z_{k-1}\] obtained at the time \(k-1\), the model generates a group of newborn box particles, \(x_{i,k-1|k-1}^{(n)}, m = 1, \ldots, M_0\), where \(M_0\) is the number of newborn box particles. At the same time, the box particles of the model resampled at time \(k - 2\) of retention are \(x_{i,k-1|k-1}^{(n)}, n = 1, \ldots, N_0\), where \(N_0\) is the number of persistent box particles. Then, the box particle state set of the \(i\)th model consists of the newborn box particles and the persistent box particles:

\[
\{x_{i,k-1|k-1}^{(l)}\}_{l=1}^{N} = \left\{x_{i,k-1|k-1}^{(m)}\right\}_{m=1}^{M_0} \cup \left\{x_{i,k-1|k-1}^{(n)}\right\}_{n=1}^{N_0}. \quad (4)
\]

It is easy to know that the total number of box particles in the \(i\)th model is \(N = M_0 + N_0\).

(2) Input Interaction. We first calculate the transition mixture probability of the model:

\[
\mu_{ij,k|k-1} = \frac{\pi_{ij} \cdot \mu_{ij,k-1}}{\sum_{i=1}^{M} \pi_{ij} \cdot \mu_{ij,k-1}}, \quad i, j = 1, \ldots, M. \quad (5)
\]

Then, we calculate the mixture estimate, which is the model’s interacting box particle input:
(3) **Box Particle Filtering for Each Model.** Predicted status:

$$
\begin{align*}
\mathbf{x}_{j,k|k-1}^{(l)} &= \Phi_{j,k}(\mathbf{x}_{j,k|k-1}^{(l-1)}) + \mathbf{w}_k. \\
\mathbf{y}_{j,k|k-1} &= \mathbf{z}_k - h_{j,k}(\mathbf{x}_{j,k|k-1}^{(l-1)})
\end{align*}
$$

(4) **Model Probability Update.** Model likelihood calculation:

$$
\Lambda_{j,k} = \frac{\sum_{l=1}^{N} w_{j,k|k-1}^{(l)}}{N}. 
$$

**Table 1:** Model set for each flight phase.

| Flight phase                      | Model set                                      |
|----------------------------------|------------------------------------------------|
| Active phase                     | Active phase model and free phase model        |
| Free phase                       | Free phase model and reentry phase model       |
| Reentry phase                    | Reentry phase model                            |
| Unknown phase                    | Active phase model, free phase model, and reentry phase model |

**Predicted measurement:**

$$
\begin{align*}
\hat{z}_{j,k|k-1} &= \hat{h}_{j,k}(\mathbf{x}_{j,k|k-1}^{(l-1)}) + \mathbf{v}_k. \\
\mathbf{r}_{j,k} &= \mathbf{z}_k - \hat{h}_{j,k}(\mathbf{x}_{j,k|k-1}^{(l-1)}).
\end{align*}
$$

**Interval measurement innovation:**

$$
\begin{align*}
\mathbf{r}_{j,k} &= \mathbf{z}_k - \hat{h}_{j,k}(\mathbf{x}_{j,k|k-1}^{(l-1)}).
\end{align*}
$$

**Box particle weight calculation:**

$$
\begin{align*}
\mathbf{w}_{j,k|k-1}^{(l)} &= \left| \mathbf{r}_{j,k}^{(l)} \right| \\
\overline{\mathbf{w}}_{j,k|k-1} &= \sum_{l=1}^{N} \mathbf{w}_{j,k|k-1}^{(l)}.
\end{align*}
$$

**Contracted box particles:**

For any box particles \(\mathbf{x}_{j,k|k-1}^{(l)}\) that satisfy \(\left\{ \mathbf{x}_{j,k|k-1}^{(l)}, \mathbf{y}_{j,k|k-1}^{(l)} \right\} \cap \mathcal{O} \neq \emptyset\) (box particles that intersect with the observation area), a newborn box particle \(\mathbf{x}_{j,k|k-1}^{\text{new}}\) is obtained by using the CP algorithm constraint. The box particles are contracted according to the constraint algorithm described previously. Resampling: resampling \(\left\{ \mathbf{x}_{j,k|k-1}^{\text{new}}, \overline{\mathbf{w}}_{j,k|k-1} \right\}_{j=1}^{N} \) yields a newborn box particle set, \(\left\{ \mathbf{x}_{j,k|k}^{(l)}, \mathbf{w}_{j,k|k}^{(l)} = 1/N_{0} \right\}_{l=1}^{N_{0}}\).

**Output:**

$$
\mathbf{x}_{j,k|k} = \frac{1}{N_{0}} \sum_{l=1}^{N_{0}} \mathbf{x}_{j,k|k}^{(l)}.
$$
Different from the standard IMM Bayesian model likelihood function calculation, as shown in equation (13), the likelihood function of model $j$ is the ratio between the sum of the box particle weights after the update and the number of box particles before sampling. Obviously, it can be seen from equation (13) that when a model conforms to the motion state of the current target, the corresponding box particle weight of the model will be larger than the box particle weight of other models; otherwise, the model cannot correctly describe the current target motion state, and the prediction bias will increase, which will cause the likelihood function to decrease. Thus, the updated model probability can be correctly calculated by equation (14) to achieve the model selection.

Model probability update:

$$
\mu_{j,k} = \frac{\Lambda_j \sum_{i=1}^{M} \pi_{ij} \mu_{i,k-1-jk-1}}{\sum_{j=1}^{M} \Lambda_j \sum_{i=1}^{M} \pi_{ij} \mu_{i,k-1-jk-1}}.
$$

(14)

(5) Interacting Multimodel Box Particle Filtering Output:

$$
x_{k,j} = \sum_{j=1}^{M} \mu_{j,k} \cdot \text{mid}(x_{j,k}),
$$

(15)

$$
P_{k,j} = \sum_{j=1}^{M} \mu_{j,k} \left\{ \sum_{i=1}^{N} w_{i,k,m}^{j} (x_{k,i,m} - \text{mid}(x_{i,k,m})) \right\}$$

$$
\cdot \left( \begin{array}{c}
x_{k,i,m} - \text{mid}(x_{i,k,m}) \\
\sum_{i=1}^{N} w_{i,k,m}^{j} \text{mid}(x_{i,k,m}) \\
\sum_{i=1}^{N} w_{i,k,m}^{j} \text{mid}(x_{i,k,m})
\end{array} \right)^{T}
$$

$$
+ \left( \begin{array}{c}
x_{k,i,m} - \text{mid}(x_{i,k,m}) \\
\sum_{i=1}^{N} w_{i,k,m}^{j} \text{mid}(x_{i,k,m}) \\
\sum_{i=1}^{N} w_{i,k,m}^{j} \text{mid}(x_{i,k,m})
\end{array} \right)^{T}
$$

(16)

2.2. Calculation of Game Payoff

2.2.1. Rényi Information Gain. Liu [1] pointed out that the Rényi information gain has no Gaussian limitation on the distributions of the prior probability density function $p(x)$ and the posterior probability density function $P(x | z_j)$. Rényi information gain can be used to emphasize a certain local information. Therefore, it is more flexible and effective than the mutual information-based method. The Rényi information gain can thus be used to characterize the tracking performance of a sensor and be used as the game payoff. The Rényi information gain is defined as follows:

$$
I_{\alpha}(P(x | z_j) \| p(x)) = \frac{1}{\alpha - 1} \log \left[ \int p^{\alpha}(x | z_j) \cdot p^{1-\alpha}(x) dx \right] - \log \left[ \frac{p(z_j | x) p(x) dx}{p^\alpha(z_j)} \right],
$$

(17)

where $P(x | z_j)$ and $P(z_j | x)$ represent the posterior probability density function and likelihood function of the sensor $j$ to the target, respectively, and $p^\alpha(z_j)$ is the edge distribution function observed by the sensor $j$.

The prior probability density function $P(x)$ in box particle filtering can be calculated according to

$$
P(x) = \sum_{i=1}^{N_p} \omega_{i} U[x^{(i)}](x).
$$

(18)

Here, $N_p$ is the number of box particles, $\omega_{i}$ represents the normalized weight, and $U[x^{(i)}](x)$ represents the box $[x^{(i)}]$ as a uniform probability distribution function of the support set.

For the box measurement $[z_j]$, there are

$$
P(z_j | x) = U[z_j] \cdot g(x).
$$

(19)

Substituting equations (18) and (19) into equation (17) yields

$$
I_{\alpha}(P(x | z_j) \| p(x)) = \frac{1}{\alpha - 1} \log \left[ \sum_{i=1}^{N_p} \omega_{i} \int [x^{(i)}] \cdot \left[ g^{-1}(z_j) \right]^u dx \right] - \log \left[ \sum_{i=1}^{N_p} \omega_{i} \right]
$$

$$
\frac{\int [x^{(i)}] \cdot \left[ g^{-1}(z_j) \right]^u dx}{\left[ g^{-1}(z_j) \right]^u}.
$$

(20)

where $[x^{(i)}] \cdot \left[ g^{-1}(z_j) \right]^u$ indicates that the box particles are contracted by measurement, thus eliminating the excessive box particles.

Similarly, according to the law of total probability and equation (18), $p^\alpha(z_j)$ can be discretized as follows:

$$
p^\alpha(z_j) = \left[ \int p(z_j | x) p(x) dx \right]^u
$$

$$
= \sum_{i=1}^{N_p} \omega_{i} \left[ [x^{(i)}] \cdot \left[ g^{-1}(z_j) \right]^u \right] \right].
$$

(21)
Thus, according to law of total probability and equation (21), equation (20) can be discretized as follows:

\[
I_a\left(P\left(x \mid z_j\right) \| p(x)\right) = \frac{1}{\alpha - 1} \log \frac{\sum_{i=1}^{N_j} \omega_i \left(\left\lvert [x'] \cap \left\lvert g^{-1}(z_j)\right\rvert\right\rvert + \left\lvert g^{-1}(z_j)\right\rvert\right)^{\alpha}}{\sum_{i=1}^{N_j} \omega_i \left(\left\lvert [x'] \cap \left\lvert g^{-1}(z_j)\right\rvert\right\rvert + \left\lvert g^{-1}(z_j)\right\rvert\right)^{\alpha}} \quad (22)
\]

It can be seen from equation (22) that the Rényi information gain can be approximated by the sum of a set of finite samples (box particles) with weights through the box particle filtering, which avoids complicated integral operations. Liu [1] analyzed the adjustment parameter \( \alpha \) and pointed out that when \( \alpha \) is set to 0.5, the two probability densities can be better identified and have better tracking performance.

2.2.2. Game Payoff. Both sides of the game always want to maximize their own payoff in the game; that is, they expect to select the sensor set that brings the most Rényi information gains to track the target. Hence, the game payoff is defined as below.

Define the gain \( G^i_S(k) \) of sensor set \( S \) at the time \( k \) to the target \( i \) as

\[
G^i_S(k) = \sum_{j \in S} E_j[I_a], \quad (23)
\]

where \( E_j[I_a] \) represents the expected value of the Rényi information gain at the current time and is obtained according to

\[
E_j[I_a] = \int I_a p(z_j) dz_j = \sum_{m=1}^{N_j} p(z_{jm}) I_a \cdot \Delta z_j, \quad (24)
\]

where \( N_j \) is the number measured by the sensors.

Obviously, by increasing the number of sensors in the sensor set \( S \), the payoff can be increased. Moreover, by replacing a sensor with a sensor with good tracking performance, both \( E_j[I_a] \) and the payoff \( G^i_S(k) \) are increased.

3. Game Theory-Based Collaborative Warning and Tracking Dynamic Programming Method

3.1. Execution Conditions. According to the target detection status, that is, new/lost target and tracked target, dynamic programming is performed under the following two execution conditions.

Execution condition 1: the target is a new target or a lost target. If there is a new target or if the target has been lost, the dynamic programming should be performed immediately, and the sensors in the current monitored area should be selected for detection and interception. Under this condition, the real-time performance requirements for the dynamic programming are high.

Execution condition 2: the tracking performance requirement for the target cannot be met. During the target tracking, the sensor may not be able to stably track the target due to interference, target penetration, or sensor performance limitations (line of sight, azimuth, and pitch angle). Once the target tracking accuracy is lower than the set threshold, dynamic programming should be immediately performed, and more suitable sensor nodes should be selected to increase tracking accuracy.

3.2. Game Theory-Based Collaborative Warning and Tracking Dynamic Programming. According to the analysis of the key factors presented in Section 1.2, a new/lost ballistic target is often more threatening than the currently tracked ballistic target (an unknown and understood target is often more threatening than a currently known target), and thus, sensors need to be timely assigned for detection and interception; namely, when there is a new target or the target is lost, the sensor in the vicinity of the area needs to be adjusted, and the corresponding sensors are assigned to the target to achieve timely detection. Then, the sensors that are tracking the target are adjusted to ensure that the target is continuously being tracked.

3.2.1. Determination of Warning/Tracking Community of Interest Based on New/Lost Target Detection Probability Model

1) Detection Probability Model [1]. It is assumed that the normal velocity of the new/lost target at the boundary of the surveillance area is uniformly distributed \([0, V_{\text{max}}]\), and \( m \) uniformly distributed particles are used to characterize the locations that may occur within the surveillance area. Thus, the distance \( D \) between the position where the new target appears and the particle satisfies the distribution is calculated in the following equation:

\[
P(D) = \int_0^{V_{\text{max}}} p(D \mid v) p(v) dv = \int_{D_t}^{V_{\text{max}}} \frac{1}{\sqrt{T_s} v T_s} \frac{1}{V_{\text{max}}} dv = \frac{1}{V_{\text{max}} T_s} \cdot \ln\left(\frac{V_{\text{max}} T_s}{D}\right). \quad (25)
\]

Here, \( T_s \) is the sampling time of the sensor, and \( v \) is the velocity of a particle. The detection probability \( P_{ij}(j, i) \) of the sensor \( j \) at the location \( i \) represented by the particle can be calculated according to

\[
P_{ij}(j, i) = P_f^{R_i(j, i)} [R_i(j, i + S_j R_i)]. \quad (26)
\]

(1) Detection Probability Model [1].
Here, \( P_{j} \) is the false alarm probability of the sensor, \( R(j,i) \) is the distance between the sensor \( j \) and the particle \( i \), and \( S_{0} \) is the detection signal-to-noise ratio of the sensor at \( R_{0} \). Then, at time \( K \), the target detection probability, \( P_{d}^{k}(COI_{i}) \), of the community of interest for the target \( i \) can be calculated according to the following equation:

\[
P_{d}^{k}(COI_{i}) = 1 - \frac{1}{m} \sum_{j=1}^{m} \prod_{j=1}^{N} (1 - S_{j} \cdot C_{j} \cdot P_{d}(j,i)),
\]  

(27)

where \( S_{j} \) is a binary function indicating whether the sensor in the monitoring area detects the target, and there are \( N \) in total. \( S_{j} = 1 \) indicates that the sensor detect the target; otherwise, it does not detect the target. Obviously, the sensor set inside \( COI_{i} \) must be a certain combination scheme \( COI_{i} = S = \{S_{1}, S_{2}, \ldots, S_{n}\} \) for \( S_{n} \). \( C_{j} \) is the sensor detection constraint. \( C_{j} = 1 \) indicates that the target is within the observation coverage of sensor \( j \); otherwise, the target is not within the observation coverage of sensor \( j \).


(2) Determination of Community of Interest for Warning/Tracking. According to equation (27), it can be determined whether the detection probability \( S \) of a certain sensor combination of target \( i \) reaches an agreed threshold. If it does, the community of interest \( COI_{i} \) for target \( i \) is constructed by the combination of sensors corresponding to \( S \). Otherwise, community of interest \( COI_{i} \) for target \( i \) is constructed by the sensor combination corresponding to the highest detection probability.

3.2.2. Dynamic Programming of the Target under Tracking. The major idea of the dynamic programming method based on game theory is to complete the game between different targets with the community of interest (COI) for the targets as the object. For the tracking \( COI \) of target \( i \), when the \( COI \) is unable to meet the tracking accuracy requirements of the system for target \( i \), the \( COI \) of other targets is negotiated to adjust the network sensor combination to form a new \( COI \) and thus achieve stable tracking of the target.

The game between the two targets is taken as an example to discuss the dynamic programming method based on game theory. Assume that before the dynamic programming target \( i \) and target \( j \) warning/tracking communities of interest are \( (COI_{i}, COI_{j}) \). After the dynamic programming is performed at the moment \( K \), target \( i \) and target \( j \) warning/tracking communities of interest are adjusted to \( (COI'_{i}, COI'_{j}) \). Then, the average gains of both parties in the game is calculated according to the following equations:

\[
U(COI'_{i}, k) = \frac{\Delta k \cdot G_{COI'_{i}}(k) + G_{COI_{i}}(k)}{\Delta k + 1},
\]  

(28)

\[
U(COI'_{j}, k) = \frac{\Delta k \cdot G_{COI'_{j}}(k) + G_{COI_{j}}(k)}{\Delta k + 1}.
\]  

(29)

Here, \( \Delta k \) is the time consumed by the game. According to the definition of game payoff, before the game, the Rényi information gain of a single sensor to the target is stable, and therefore, it is reasonable and feasible to use the average payoff over time \( \Delta k \) to play the game.

If it is necessary to improve the tracking accuracy of target \( j \) at moment \( K \), \( COI_{j} \) will request negotiation with \( COI_{i} \). If \( COI_{j} \) wants to meet the demand of \( COI_{i} \), it must sacrifice its own gain (the tracking accuracy of target \( i \)). If the negotiation is successful, we can get the following equation:

\[
U(COI'_{i}, k + 1) \leq U(COI_{i}, k) \quad \text{and} \quad U(COI'_{j}, k + 1) > U(COI_{j}, k).
\]

Clearly, for the negotiation initiator \( COI_{i} \), the sooner the outcome is reached, the better the outcome will be. The opposite is true for the recipient \( COI_{j} \). This forms a game. Three important definitions of game theory are now introduced [1]:

\[
P_{\text{pos}}(k) = \{S = (COI'_{i}, COI'_{j}) \mid U(COI'_{i}, k) > U(COI_{i}, k)\},
\]  

(30)

\[
U(COI'_{i}, S_{0}(k), k) = \max_{S_{b}(k) \in P_{\text{pos}}(k)} U(COI'_{i}, S, k), S_{b}(k) \in P_{\text{pos}}(k),
\]  

(31)

\[
C(k) = \{S^{*} \in P_{\text{pos}}(k) \mid U(COI'_{i}, S^{*}, k) \geq U(COI'_{j}, S_{b}(k + 1), k + 1)\}.
\]  

(32)

The game between the two participants in the case of successful negotiation proceeds as follows. From the gaming perspective of the negotiator (\( COI_{i} \)), equation (30) represents at the moment \( k \) all the game scenarios \( P_{\text{pos}}(k) \) that are better than the failure of the negotiation. From the gaming perspective of the negotiation’s recipient \( COI_{j} \), equation (31) shows the game plan \( S_{b}(k) \) in \( P_{\text{pos}}(k) \) that results in the maximized \( COI_{i} \) gain/minimized \( COI_{j} \) gains if the negotiation is successful.

In the case of negotiation failure, from the gaming perspective of the negotiator \( COI_{i} \), the sensors involved in \( COI_{i} \) could be taken over to improve the tracking accuracy of target \( j \). From the gaming perspective of the recipient \( COI_{j} \), the greatest concession by \( COI_{i} \) is that as long as it satisfies its own expected tracking accuracy for target \( i \), it can provide as many sensor resources as possible. Equation (32) shows that, for \( COI_{i} \) at the moment \( k \), there may be a game scheme \( C(k) \in P_{\text{pos}}(k) \) where the game payoff is greater than that in the optimal scheme \( S_{b}(k + 1) \in P_{\text{pos}}(k + 1) \) of \( COI_{i} \) at the moment \( k + 1 \). \( C(k) \) may be an empty set (representing the absence of such a scheme).

To summarize this, the game process at moment \( K \) between \( COI_{i} \) and \( COI_{j} \) is shown in Algorithm 1.

3.3. Algorithm Implementation Steps. As shown in Figure 3, the game-based collaborative warning and tracking dynamic programming method is performed as follows.
**Step 1:** according to the current sensor-target status, \( COI_j \) provides \( COI_i \) with \( P_{\text{poss}}(k) \) and requests the relevant sensors in a coordinated combat operation. Under the condition of maximizing the game’s payoff with regard to the target \( i \), \( COI_i \) chooses \( S_h(k) \) from \( P_{\text{poss}}(k) \). At this point, \( COI_j \) has the least payoff in the game.

**Step 2:** \( COI_j \) evaluates \( S_h(k) \) provided by \( COI_i \). If \( COI_j \) is not satisfied (the requirement for tracking accuracy is not met), go to **Step 3**; otherwise, go to **Step 5**.

**Step 3:** if \( C(k) \) is an empty set, go to **Step 4**; otherwise, \( COI_i \) provides \( COI_j \) with a plan to maximize the game payoff for \( COI_j \), and we proceed to **Step 5**.

**Step 4:** \( COI_j \) accepts the plan \( S_h(k) \) provided by \( COI_i \).

**Step 5:** The game ends.

**Algorithm 1:** Game between the communities of interest.

**Step 1.** Interacting multimodel box particle filtering.

Perform box particle filtering by taking the sensor observation value and multimodel interaction as input values and obtain the state estimation value of the relevant sensor regarding the target at time \( K \): \( X_{k/k} \), \( P_{k/k} \).

![Figure 3: Game theory-based collaborative warning and tracking dynamic programming algorithm flow chart.](image-url)
Step 2. Execution condition decision.

Decide the execution condition. If it is execution condition 1, go to Step 3. If it is execution condition 2, go to Step 4.

Step 3. New/lost target sensor assignment.

Generate $m$ detection particles in the region where the new/lost target may occur. Acquire a set of sensors $U_k$ available in the region and calculate the target detection probability $P^k_d(S_k)$ of the different sensor combinations $S_k (S_k \subseteq U_k)$ in the set. Once there is a $P^k_d(S_k)$ that is greater than the set detection threshold $T_d$, output the corresponding $S_k$ to form the warning community of interest for the new target, and go to Step 4. If all $S_k$ do not meet the threshold conditions, output $S_k$ corresponding to the largest detection probability to form the warning community of interest for the new target and proceed to Step 4.

Step 4. Dynamic adjustment of sensors for tracked targets.

1. Calculate the difference $E_k(COI_j)$ between the actual variance level for the community of interest for each target and the system’s expected variance level $||P_i^{-1}||_2$ at the moment $K$ and sort the results in descending order (in execution condition 1, it is necessary to update the sensor set in the corresponding tracking community of interest according to the sensor set $S_k$ of the warning community of interest and then calculate $||P_i^{-1}||_2$ where $i \in I$, and $I$ is the total number of targets). $E_k(COI_j)$ is calculated as follows:

$$E_k(COI_j) = ||P_i^{-1}||_2 - \left(\sum_{j \in COI_j} (P_i^{-1})^{-1}\right)^{-1}$$

(33)

2. Select $COI_{j_1}$ in $\max_{i \in I} E_k(COI_j)$ and play the game with $COI_{j_2}$ in $\min_{i \in I} E_k(COI_j)$, where the negotiation proceeds according to Algorithm 1 to obtain a new community of interest $COI_j'$. $COI_j'$

3. Update and sort $E_k(COI_j)$ corresponding to the target, and iteratively loop Step 2 until

$$\max_{i \in I} E_k(COI_j) = \min_{i \in I} E_k(COI_j)$$

or the value of $\max_{i \in I} E_k(COI_j)$ approaches 0.

4. At the end of the game, output the COI set for the current target: $\{COI_1, COI_2, \ldots, COI_l\}$.

Step 5. Perform detection and tracking of the target according to the adjusted allocation scheme.

4. Simulation Analysis of Distributed Dynamic Programming Method for Collaborative Warning and Tracking Based on Game Theory

4.1. Experiment Scenario Setting. In order to verify the rationality and effectiveness of the method in this chapter and to achieve distributed dynamic programming for collaborative warning and tracking, the simulation scenario is set as follows. Twelve sensors are deployed. Sometime after launch, three batches of ballistic targets enter the radar surveillance area at the simulation times of 10 s, 20 s, and 30 s. The relative positions of the sensors and the ballistic targets are shown in Figure 4. The corresponding sensor position coordinates and the coordinates of launch points and landing points of the ballistic targets are shown in Tables 2–4. When the VSAIMM-BPF algorithm is executed, the noise interval takes the confidence interval of 99%, the interval length of the measurement $\Delta = [\Delta t, \Delta E] = [150m, 0.4'\), $\lambda_1 = 0.6$, and $\lambda_2 = 0.4$; the number of persistent box particles is 50, the number of newborn box particles is 10, and the measurement sampling interval $T = 0.5$ s.

4.2. Analysis of Simulation Results. Assuming that, at the initial time of tracking (10 s), the sensor set is $\{S_1, S_3, S_5, S_2, S_4, S_10, S_{11}\}$ in the community of interest $COI_1$ for target 1 in the network, the simulations use the method described in this chapter and a random assignment method (subjectively selecting the available radars to track the target). Special attention is paid to the target tracking performance of the sensors at the moment when the new target appears and the transition time of each target flight phase. The obtained simulation results are shown in Figure 5.

Figure 5 shows the adjusted results of the game-based distributed dynamic programming for collaborative warning and tracking used in different simulation moments ($t_0 \sim t_k$). In Figure 5, the sensor nodes within the black solid line loop denote $COI_1$ at the current time planned for system target 1, the sensor nodes within the blue solid line loop denote $COI_2$ at the current time planned for system target 2, and the sensor nodes within the red solid line loop denote $COI_3$ at the current time planned for system target 3. Tables 5–7 show the changes in the sensor set responsible for tracking each target at the different simulation moments under the same simulation conditions and using either the method described in this chapter or the random assignment method.

Accordingly, Figures 6 and 7 show the tracking trajectory of each ballistic target in the simulation using the two methods. It can be seen that, in the time interval $[t_0, t_2]$, when the new target appears, the motion state estimation of target 1 using the random assignment method disagrees on multiple occasions with the actual target motion state of the target. For the method described in this chapter, although the motion state estimation of target 1 disagrees with the actual motion state, better tracking accuracy is obtained after adjustment. The two methods are more stable for the tracking of target 2, and the estimations of their motion state are more accurate. For target 3, when the random assignment method is used, a large estimation error occurs after the target has flown for a certain period of time. Throughout the entire tracking process of each target, the method described in this chapter is significantly better than the random assignment method.

According to equation (33), the average position errors for the three targets under the two methods are calculated, as shown in Figure 8. It can be seen that, in the early and late stages of tracking the three targets of the sensor network, the overall tracking errors are greater than those in the middle stage. In the middle stage of tracking, the tracking error is small, and the target tracking effectiveness is stable. The
The target tracking accuracy of the method presented in this chapter is better than that of the random assignment method in general, especially in the early and late stages of target tracking. This is explained as follows. In the early stage of target tracking, the emergence of new targets takes up a certain amount of sensor resources, and the tracking error is sharply increased due to the inappropriate sensor distribution scheme with the random assignment method; moreover, as each target moves from the active phase to the free phase and although there is an adaptive adjustment of the VSAIMM-BPF algorithm, errors are unavoidable. Using the random assignment method, it is not possible to quickly and efficiently select suitable sensors for target tracking, thus making it difficult to effectively reduce local error for each target. At the same time, these local errors further affect the decision of the next random assignment, which leads to an increase and greater variation in the overall target tracking error. When there are new targets and the ballistic targets transition between flight phases and by using the real-time and accurate Rényi information gain obtained by the VSAIMM-BPF algorithm, the game theory-based method

![Figure 4: Two-dimensional distribution chart of sensor positions and ballistic target motion trajectories.](image1)

| Table 2: Sensor deployment location. |
|-------------------------------------|
| **Location** | **Sensor identification number** |
| North (km) | S_1 | S_2 | S_3 | S_4 | S_5 | S_6 |
| 750 | −690 | −600 | −495 | −455 | −400 |
| East (km) | 290 | 150 | 595 | 120 | 200 | 500 |

| Table 3: Sensor deployment location. |
|-------------------------------------|
| **Location** | **Sensor identification number** |
| North (km) | S_7 | S_8 | S_9 | S_10 | S_11 | S_12 |
| −354 | −328 | −242 | −193 | −134 | −98.4 |
| East (km) | 310 | 93 | 258 | 3.8 | 103 | 193 |

| Table 4: Ballistic target initial information and performance parameters. |
|---------------------------------|
| **Ballistic target** | **Time at entering the radar surveillance area (s)** | **Location of launch (km)** | **Location of landing (km)** | **Range (km)** | **Time at shutdown (s)** |
| Target 1 | 10 | (−688.7, 380.4) | (−78.7, 3.0) | 700 | 65 |
| Target 2 | 20 | (−744.3, 255.0) | (−78.1, −10.7) | 700 | 78 |
| Target 3 | 30 | (−389.5, 642.3) | (−76.2, 59.4) | 650 | 83 |
Figure 5: Continued.
used in this chapter balances the target tracking capability and demand of the sensor nodes in the entire sensor network. The method also adaptively adjusts to the current suitable sensor set for target tracking, thereby obtaining a better overall tracking effectiveness.

Table 8 shows the statistical results of the two methods from 100 Monte Carlo simulation experiments. It can be seen that the detection probability yield of the random assignment method is slightly higher than that of the proposed method. Combining Table 7 with Figures 5 and 8 and

| Ballistic target | Method | Sensor sets for respective targets at different times |
|------------------|--------|-----------------------------------------------------|
| Target 2         | Random assignment | $\emptyset$ |
| Target 2         | The proposed method | $\emptyset$ |
| Target 2         | Random assignment | $\emptyset$ |
| Target 2         | The proposed method | $\emptyset$ |
| Target 3         | Random assignment | $\emptyset$ |
| Target 3         | The proposed method | $\emptyset$ |

Figure 5: Sensor assignment results of the method proposed in this paper at different times: (a) $t_0 = 10$ s; (b) $t_1 = 20$ s; (c) $t_2 = 30$ s; (d) $t_3 = 66$ s; (e) $t_4 = 78$ s; (f) $t_5 = 83$ s; (g) $t_6 = 325$ s; (h) $t_7 = 327$ s; (i) $t_8 = 364$ s.

Table 5: Distribution results of sensors obtained when using the proposed method at different times.
Table 6: Distribution results of sensors obtained when using the proposed method at different times.

| Ballistic target | Method                  | $t_3$                     | $t_4$                     | $t_5$                     |
|------------------|-------------------------|---------------------------|---------------------------|---------------------------|
| Target 1         | Random assignment       | $\{S_1, S_2, S_3, S_5, S_{11}\}$ | $\{S_1, S_3, S_5, S_{11}\}$ | $\{S_1, S_3, S_5, S_{11}\}$ |
|                  | The proposed method     | $\{S_1, S_3, S_5, S_{11}\}$ | $\{S_5, S_5, S_7, S_{11}\}$ | $\{S_3, S_5, S_{11}\}$   |
| Target 2         | Random assignment       | $\{S_4, S_5, S_{10}\}$    | $\{S_3, S_5, S_{11}\}$    | $\{S_2, S_4, S_{10}\}$   |
|                  | The proposed method     | $\{S_2, S_4, S_{10}\}$    | $\{S_1, S_2, S_4, S_{10}\}$ | $\{S_1, S_2, S_4, S_{10}\}$ |
| Target 3         | Random assignment       | $\{S_6, S_7, S_{12}\}$    | $\{S_6, S_7, S_{12}\}$    | $\{S_6, S_7, S_{12}\}$   |
|                  | The proposed method     | $\{S_6, S_7, S_{12}\}$    | $\{S_6, S_7, S_{12}\}$    | $\{S_6, S_7, S_{12}\}$   |

Table 7: Distribution results of sensors obtained when using the proposed method at different times.

| Ballistic target | Method                  | $t_6$                     | $t_7$                     | $t_8$                     |
|------------------|-------------------------|---------------------------|---------------------------|---------------------------|
| Target 1         | Random assignment       | $\{S_3, S_5, S_7, S_9, S_{11}\}$ | $\{S_3, S_5, S_7, S_9, S_{11}\}$ | $\{S_3, S_5, S_7\}$   |
|                  | The proposed method     | $\{S_3, S_5, S_7, S_9, S_{11}\}$ | $\{S_3, S_4, S_5, S_{11}\}$ | $\{S_3, S_4, S_5, S_{11}\}$ |
| Target 2         | Random assignment       | $\{S_1, S_2, S_4, S_{10}\}$ | $\{S_1, S_2, S_4, S_{10}\}$ | $\{S_1, S_1, S_2, S_4, S_{10}\}$ |
|                  | The proposed method     | $\{S_1, S_2, S_4, S_{10}\}$ | $\{S_1, S_2, S_4, S_{10}\}$ | $\{S_1, S_2, S_4, S_{10}\}$ |
| Target 3         | Random assignment       | $\{S_6, S_{12}\}$        | $\{S_6, S_{12}\}$        | $\{S_6, S_{12}\}$ |
|                  | The proposed method     | $\{S_6, S_{12}\}$        | $\{S_6, S_{12}\}$        | $\{S_6, S_{12}\}$ |

Figure 6: Continued.
Figure 6: Tracking effectiveness with the random assignment method: (a) overall tracking trajectory of each target with the random assignment method; (b) local zoomed-in tracking trajectory of each target with the random assignment method.

Figure 7: Continued.
comparing the results of the two methods at the moment $t_1 \sim t_2$ when the new target appears, it can be seen that this is because the random assignment method allocates all the sensor resources in the network for the detection of the new target when the target appears. In contrast, the tracking resources reserved for the tracked targets are reduced, and the overall tracking accuracy of the system is reduced. Our method can reasonably allocate the remaining sensor resources to the target with a detection probability greater than 98%, thus improving the overall tracking performance of the system. In terms of tracking accuracy, the method described in this chapter is better than the random assignment method. It can balance the tracking demand of the system in the monitored area to dynamically adjust the sensor resources so that the system maintains continuous and stable tracking of all targets.

5. Conclusions

In this paper, we first apply the tracking filter algorithm because the existing ballistic target tracking methods in antimissile warfare are mostly based on the single-motion model and thus cannot be applied to the entire duration of ballistic target tracking. Moreover, we propose a variable
structure adaptive multimodel box particle filtering tracking method. This method allows for continuous and persistent tracking of ballistic targets and reduces the computational complexity while achieving higher tracking accuracy. Then, with the Rényi information gain as the game payoff for the target detection (warning) and target tracking states, we propose a game theory-based dynamic programming method. Through the game design, it is possible to adaptively obtain more or better sensor resources for the communities of interest that need to be coordinated to detect and track the corresponding ballistic targets. Simulation experiments show that this method achieves good overall tracking performance while ensuring an adequate detection probability (warning capability).

### Data Availability

The data used to support the findings of this study are included within the article. The code of the proposed method was supplied by the author Peng Ni under license and so cannot be made freely available. Requests for access to these data should be made to him via nipeng198509@163.com.

### Conflicts of Interest

The authors declare that there are no conflicts of interest.

### Acknowledgments

This work was supported by the National Natural Science Foundation of China (Grant no. 61703426), by Young Talent fund of University Association for Science and Technology in Shaanxi, China (Grant no. 20190108).

### References

[1] Q. Liu, *Study on Multisensor Collaborative Tracking in Sensor Networks*, Xidian University, Xi’an, China, 2013.

[2] R. M. Dressler, L. I. Meier, and J. Peschon, “An approach to model-referenced adaptive control systems,” *IEEE Transactions on Automatic Control*, vol. 12, no. 5, pp. 528–536, 1967.

[3] J. M. Nash, “Optimal allocation of tracking resources,” in *Proceedings of the Decision and Control Including the 16th Symposium on Adaptive Processes and A Special Symposium on Fuzzy Set Theory and Applications*, New Orleans, Mexico, December 1977.

[4] A. Manyika, *Data Fusion And Sensor Management: A Decentralized Information-Theoretic Approach*, Prentice Hall PTR, Upper Saddle River, NJ, USA, 1995.

[5] W. Schmaedeke, “Information-based sensor management,” *Proceedings of SPIE the International Society for Optical Engineering*, vol. 1955, no. 1955, pp. 156–164, 1993.

[6] K. J. Hintz and G. A. McIntyre, “Information instantiation in sensor management,” in *Proceedings of the SPIE International Symposium on Aerospace/Defense Sensing & Control*, vol. 3374, pp. 38–47, Orlando, FL, USA, 1998.

[7] Y. Song, Q. Fu, Y.-F. Wang, and X. Wang, “Divergence-based cross entropy and uncertainty measures of Atnassov’s intuitionistic fuzzy sets with their application in decision making,” *Applied Soft Computing*, vol. 84, p. 105703, 2019.

[8] B. Zhang, M. Zhang, Y. Song, and L. Zhang, “Combining evidence sources in time domain with decision maker’s preference on time sequence,” *IEEE Access*, vol. 7, no. 12, pp. 174210–174218, 2019.

[9] Y. Song, X. Wang, W. Quan, and W. Huang, “A new approach to construct similarity measure for intuitionistic fuzzy sets,” *Soft Computing*, vol. 23, no. 6, pp. 1985–1998, 2019.

[10] Y. Song, X. Wang, J. Zhu, and L. Lei, “Sensor dynamic reliability evaluation based on evidence theory and intuitionistic fuzzy sets,” *Applied Intelligence*, vol. 48, no. 11, pp. 3950–3962, 2018.

[11] Y. Song, X. Wang, L. Lei, and A. Xue, “A novel similarity measure on intuitionistic fuzzy sets with its applications,” *Applied Intelligence*, vol. 42, no. 2, pp. 252–261, 2015.

[12] L. Lei, Y. Song, and X. Luo, “A new re-encoding ecoc using a reject option,” *Applied Intelligence*, 2020.

[13] B.-Shalom and T. E. Fortmann, *Tracking and Data Association*, Academic Press, Cambridge, MA, USA, 1988.

[14] D. R. Umesh and B. Fok, “Performance-based sensor selection for optimal target tracking,” in *Proceedings of the 12th International Conference on Information Fusion*, pp. 1687–1694, Seattle, WA, USA, July 2009.

[15] G. Y. Kulikov and M. V. Kulikova, “Accurate continuous-discrete unscented Kalman filtering for estimation of nonlinear continuous-time stochastic models in radar tracking,” *Signal Processing*, vol. 139, pp. 25–35, 2017.

[16] K. Luo, W. Jiang, and Y. Li, “Review of sensor management,” *Acta Electronica Sinica*, vol. 38, no. 8, pp. 1900–1907, 2010.

[17] C. Wu and C. Han, “New nonlinear filtering algorithm for ballistic target tracking,” *Computer Engineering and Applications*, vol. 47, no. 20, pp. 20–23, 2011.

[18] Y. Chen, Z. Cheng, and S. Wen, “Comparative study on methods for tracking ballistic reentry targets,” *Systems Engineering and Electronics*, vol. 33, no. 3, pp. 495–499, 2011.

[19] Z. L. Zhao, H. M. Chen, G. S. Chen et al., “Comparison of several ballistic target tracking filters,” in *Proceedings of the American Control Conference*, pp. 2197–2202, Minneapolis, MN, USA, June 2006.

[20] M. Joaquin, I. P. Maríno, and M. A. Vázquez, “Analysis of a nonlinear importance sampling scheme for Bayesian parameter estimation in state-space models,” *Signal Processing*, vol. 142, pp. 281–291, 2018.

### Table 8: Detection probability and RMSE of target tracking obtained with different methods.

| Algorithm                  | Target 1 Detection probability (%) | Location RMSE (m) | Target 2 Detection probability (%) | Location RMSE (m) | Target 3 Detection probability (%) | Location RMSE (m) |
|----------------------------|------------------------------------|-------------------|------------------------------------|-------------------|------------------------------------|-------------------|
| Random assignment          | —                                  | 82.25             | 99.98                              | 77.59             | 99.52                              | 88.06             |
| The proposed method        | —                                  | 53.10             | 99.90                              | 44.16             | 98.96                              | 58.75             |
[21] Z. CHEN, Y. QU, Z. XI, Y. Bo, and B. Liu, "Efficient particle swarm optimized particle filter based improved multiple model tracking algorithm," Computational Intelligence, vol. 33, no. 2, pp. 262–279, 2017.

[22] F. Zhang, M. Xi, and K. Tian, "Ballistic missile tracking based on interacting multiple model," Missiles and Space Vehicles, vol. 34, no. 6, pp. 34–39, 2011.

[23] J. Yu, M. Liu, Y. Wu, and H. Gao, "Ballistic missile tracking based on asymmetric interacting multiple model," Systems Engineering and Electronics, vol. 32, no. 12, pp. 2570–2575, 2010.

[24] J. Niu, Q. Shan, Q. Ren, and J. Liu, "A new algorithm with variable-structure multiple model for tracking ballistic missile at entire flight stages," Radar Science and Technology, vol. 9, no. 3, pp. 224–231, 2011.