

Research Article

Digital Twin-Enabled Online Battlefield Learning with Random Finite Sets

Peng Wang , Mei Yang, Jiancheng Zhu , Yong Peng, and Ge Li

College of Systems Engineering, National University of Defense Technology, Changsha 410073, China

Correspondence should be addressed to Jiancheng Zhu; zhujiancheng@nudt.edu.cn

Received 5 February 2021; Accepted 2 May 2021; Published 13 May 2021

Academic Editor: Qingshan Liu

Copyright © 2021 Peng Wang 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.

The digital twin is becoming the most promising emerging technology in the field of unmanned combat and has the potential to innovate future combat styles. Online battlefield learning is one of the key technologies for supporting the successful application of digital twin in unmanned combat. Since there is an urgent need for effective algorithms for online learning the battlefield states in real time, a new random finite set- (RFS-) based algorithm is proposed in the presence of detection uncertainty including clutters, missed detection, and noises. The system architecture and operational mode for implementing the digital twin-enabled online battlefield learning are provided. The unmanned ground vehicle (UGV) is employed as the experimental subject for systematically describing the proposed algorithm. The system architecture for implementing the digital twin-enabled online battlefield learning is firstly given, and its operational mode is also described in detail. The RFS-based digital twin models including the battlefield state model, UGV motion model, and sensor model are designed. The Bayesian inference is adopted, and the probability hypothesis density (PHD) filter is modified to implement the online learning process. At last, a group of experiments are conducted to verify the performance and effectiveness of the proposed algorithm. The research work in this paper will provide a good demonstration of the application of digital twin in unmanned combat.

1. Introduction

The adoption of unmanned vehicles brings both great autonomy and new technical challenges to modern warfare. Unmanned vehicles such as unmanned ground vehicles (UGVs) hold great promise for future combat operations and have already been used in several recent military conflicts in Syria and Afghanistan [1–3]. UGVs are the vehicles that operate while in contact with the ground and without a human presence on board. How to feed back the effective information collected from the real battlefield to the simulation space and how to enable the benefits of future paradigms, such as the Cyber-Physical Systems (CPSs) and digital twin, are big challenges for unmanned combat [4–7]. In this paper, we employ the UGV as the experimental subject to specify our contributions in implementing digital twins in unmanned combat.

Due to the data separation between the real battlefield and its models, it is difficult to achieve the automatic flow of

information in a closed loop. Digital twin provides a new and effective way to solve this problem. It can enable the real-time bidirectional interoperability between the real world and virtual simulation space and is also an effective way to enable efficient real-time data sharing throughout the entire operational process including intelligent monitoring, prediction, digital representation, evaluation, decision support, and battlefield learning [8–10].

Battlefield refers to the environment constituted by all the objective factors in the battlespace except the combatants and weapons. All kinds of combat operations are inseparable from the specific battlefield. Battlefield has an important influence on the course and outcomes of combat operations. Combat entities can receive inputs from and provide outputs to the battlefield. The combat intention of the combat entity is realized through its interaction with the battlefield.

Battlefield learning means sensing the entities on the battlefield rapidly, understanding the current situation comprehensively, and predicting future status accurately

before decision-making [11]. Battlefield learning is important for predicting future situations and evaluating the operational effectiveness of different actions. Battlefield learning helps to improve the commander's understanding of the situation as a whole and form a basis for decision-making. It is also very important for the commanders' real-time monitoring and perception of the dynamic situation [12].

Based on the classical definition of battlefield learning, online battlefield learning is the process of perceiving an existing battlefield and anticipating how it may evolve in the future. It is useful for obtaining knowledge of the previously unknown battlefield while the real combat process is proceeding simultaneously [13]. Online battlefield learning is also extremely important for generating plans and online decision support for security patrol [14].

In the military simulation, computer-generated force (CGF) is the virtual combat force object which is created by a computer and can control or guide all or part of its action and behavior [15]. The core task of constructing CGF is to model the behavior of combat entities on the battlefield. Online battlefield learning is one of the key technologies of CGF and has a broad application prospect. CGF depends on the online learning battlefield to fuse the data generated by the sensors in the battlefield and generate the real-time battlefield states online.

In recent years, the digital twin has become a hot topic, as well as the representative intelligence in all fields from military to people's livelihood [16–19]. Digital twin emphasizes that the virtual object evolves in real time by receiving data from the physical object, thereby keeping consistent with the physical object throughout its entire simulation cycle [20, 21]. In a broad sense, the digital twin is a system composed of physical objects, simulation models, and the real-time dynamic interaction between them. It requires building the simulation models for real entities and simulating their behaviors [22]. It is regarded as the core link between the real and virtual spaces. With the help of various high-performance sensors and high-speed communication technologies, the digital twin can present and predict the actual situation of physical entities in near real time by integrating the data of physical entities. It enhances the ability of analysis and simulation and controls the physical entities through the virtual-real interactive interfaces and data fusion algorithms [23]. Key to enable digital twin in unmanned combat is understanding the evolving situations in the battlefield accurately and timely.

In this paper, we focus on learning the battlefield states that consist of significant environmental cues and the UGV states. In order to explore how to implement the digital twin-enabled online battlefield learning, we propose a random finite set- (RFS-) based algorithm which can support real-time interaction, as well as the deep integration and mutually beneficial symbiosis between the virtual and real battlefield. It is the necessary foundation for the successful application of the digital twin in unmanned combat. Our main contribution is designing and implementing a new online battlefield learning algorithm by using the RFS-based Bayesian theory and modifying the probability hypothesis density

(PHD) filter [24]. The most important value of the proposed algorithm is to break through the data boundary between the real and virtual battlefield and enable the application of digital twin in unmanned combat. This algorithm can eliminate information islands and realize the tight integration and equal interaction of real and virtual battlefield.

The rest of the paper is structured as follows. A literature review on the recent digital twin and random finite set (RFS) is given in Section 2. We present the system architecture and operational mode of the proposed online battlefield learning algorithm in Section 3. The RFS-based battlefield model, UGV motion model, and sensor model are introduced in Section 4. The design and implementation of the learning process are given in Section 5. Experimental results are detailed in Section 6, and conclusions are given in Section 7.

2. Related Works

The digital twin has important research and application value in every stage of online battlefield learning. In the design and demonstration stage, the digital twin can help to improve the evaluation capability of system performance by enabling the equal two-way interaction between the simulation system and the real system. Through the semiphysical simulation, digital twin enhances the ability to quickly locate the design defect, optimize system design, and test the practicability of an online battlefield learning algorithm in execution.

In order to apply the digital twin-enabled online battlefield learning in the operation stage, it is important to realize the bidirectional interaction between the simulation space and the real space. Tao gives the five-dimensional structure models of digital twin and presents six application principles [25, 26]. The digital twin is the best way to realize the interactive integration of real space and simulation space and is highly concerned by many academics and enterprises. Its most important breakthrough is that it is not only a mirror image of the physical world but also accepts real-time data from the physical world and in turn acts on the physical world in real time [22, 27]. Digital twin brings new development opportunities to the combat simulation area, because it can allow commanders to have a complete digital footprint of the battlefield from beginning to end [28, 29]. The real-time dynamic interaction between the virtual world and the physical world is the foundation of the digital twin, as well as the main challenge of modeling and simulation. Some researchers present a digital twin-driven manufacturing cyber-physical system for parallel controlling of the smart workshop. By using the decentralized digital twin models, they successfully connect cyberspace and physical space.

Online battlefield learning needs autonomy in the operation stage. A decentralized multiagent system is also a new approach for implementing online battlefield learning, such as blockchain and CGF. Some researchers have discussed how to use blockchain to overcome the cybersecurity barriers for achieving intelligence in Industry 4.0 and introduced eight cybersecurity issues in manufacturing systems. Some researchers have surveyed the ability of blockchain for overcoming the barriers and examined the

literature on the manufacturing system perspective and the product lifecycle management perspective. Ali et al. provided a survey of all aspects of multiagent systems, starting from definitions, features, applications, challenges, and communications to evaluation. They also gave a classification on multiagent system applications and challenges along with references for further studies [30].

RFS provides a novel unified probabilistic way for fusing real-time battlefield data [31]. The conventional battlefield learning algorithms usually depend on the vector-based data representation and fail to support the digital twin in real time. The vector-based representation requires the dimension and elements' order in each vector to be equal and fixed. It also needs necessary operations outside of the Bayesian recursion to ensure the consistency of the vectors. The determination of newly observed measurements and missed measurements is implementing through vector augmentation and truncation which are very computationally intensive and irreversible. In this paper, we propose employing the random set theory to overcome these disadvantages. The proposed RFS-based algorithm can overcome the limitations of conventional algorithms very well, because it takes into account a more realistic situation where the randomly varying number of targets and measurements, detection uncertainty, false alarms, and association uncertainty are all taken into consideration.

3. System Architecture

With the rapid development of emerging information technologies, such as artificial intelligence (AI), cloud computing, edge computing, digital twin, and Internet of Things (IoT), the combat style has also been undergoing profound changes. New information technologies have facilitated the birth, development, and application of unmanned combat. Just as it is shown in Figure 1, new information technologies provide more diverse data sources, more powerful computing power, and more efficient computing methods for the key activities of unmanned combat including description, diagnosis, prediction, and decision.

The operational mode of the digital twin-enhanced online battlefield learning consists of five elements, i.e., computing services, physical entities, simulation models, connected data, and the connection between them. As shown in Figure 2, digital twin enables the bidirectional real-time mapping and interaction between real battlefield and its simulation model. Simulation models of the real combat entities are employed to reflect and predict their behaviors in real space. On the other hand, through the RFS-based battlefield states generated by the online battlefield learning algorithm, the combat simulation systems could guide the military commanders to respond to situation changes and choose the optimal courses of action (COA). Digital twin realizes the closed-loop optimization in the entire process from observing, orienting, and deciding to act. The simulation aspect of digital twin means building digital models of weapons, soldiers, or battlefield and executing all the models in an integrated way. The RFS-based simulation models are

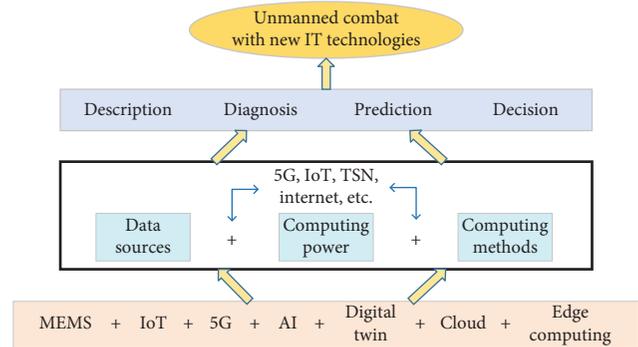


FIGURE 1: The relationship between new information technologies and unmanned combat.

executed in parallel with the real battlefield and provide useful knowledge to the commanders.

The battlefield considered in this paper consists of all the significant environmental cues and the states of UGVs. Since GPS and topographic map in actual combat are most likely be disabled, location and mapping for unmanned vehicles can only be obtained with the help of the equipped sensors. The RFS-based online battlefield learning algorithm plays a central role in the virtual space. It provides simulated battlefield information to the decision support system to train the deep learning network system. It can also generate real-time battlefield information to the unmanned combat simulation system and helps to evaluate the possible outputs of available COAs.

For combat simulation, the battlefield provides spatial-temporal constraints for all participating actors. The simulated combat objects are deployed and controlled in the virtual space. They learn the battlefield that consists of other combat objects and significant environmental cues by using the proposed algorithm. The combat simulation system in the virtual space is used as a decision-making aid tool that assists the commanders to evaluate all the available COAs. It is in charge of choosing the optimal COA. The proposed online battlefield learning algorithm aims at analyzing and understanding operational activity in the real space at a given time. It can help to make the right decision and predict the future situation. It is the key technology for enabling and implementing digital twin-enabled online battlefield learning in unmanned combat.

Corresponding to the operational mode, the system architecture of digital twin-enabled online battlefield learning in unmanned combat is shown in Figure 3. The runtime infrastructure (RTI) is adopted to provide the simulation services to support the interconnection and interoperation for the entities in the real space and the simulation models in the virtual space. This system architecture employs digital twin and RTI to support real-time interaction between the virtual and real battlefield. By this means, it can realize the deep integration and mutually beneficial symbiosis between the virtual and real battlefield. The proposed algorithm can synchronously learn the number and position of the significant environmental cues (or landmarks) in the battlefield that exist in the sensor's

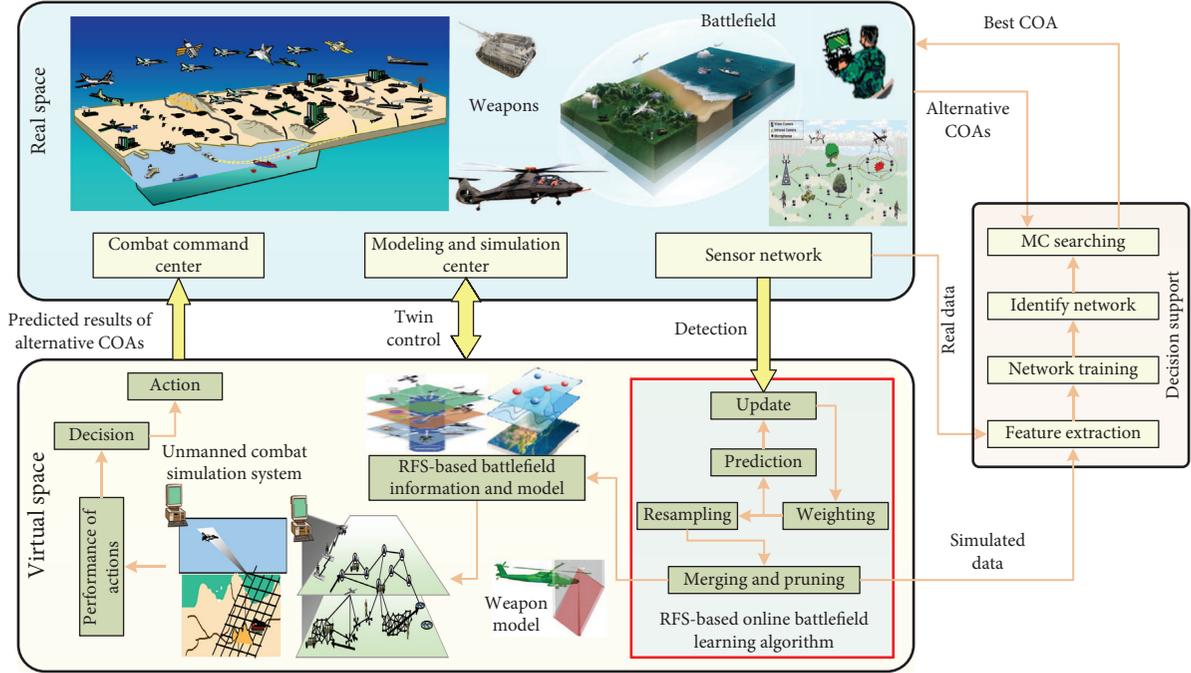


FIGURE 2: The operational mode of digital twin-enabled online battlefield learning in unmanned combat.

field of view (FoV). It also has the advantages of precise mapping, virtual-real interaction, stereoperception, intelligent intervention, and other characteristics.

4. RFS-Based Simulation Models

The digital twin-enabled battlefield modeling consists of three aspects. The first one is modeling the battlefield states including cues (or landmarks). The second one is modeling the UGV movement. The third one is modeling the sensors equipped on the UGV. In order to overcome the data association uncertainty problem under high clutters and measurement noises, the RFS-based modeling method is employed to fully integrate data association uncertainty into battlefield learning. The key of the proposed algorithm is to represent the battlefield states by using RFS. The derivation of the simulation models depends on RFS. RFS is the theory proposed by Mahler for implementing RFS in engineering applications [32]. The RFS-based models are the twinning models that are executed in parallel with the real entities and provide new knowledge about the real battlefield [8, 27].

The vector-based representation of the battlefield has been demonstrated to have some mathematical consequences, such as the ordering of significant environmental cues, data association problems, and element management problems. In addition, for the dynamic random scene, how to quantify the errors of the learned results generated by the vector-based Bayesian inference is also a great challenge. The abovementioned problems are usually solved by augmenting or truncating vectors outside of the Bayesian inference process. This will lead to the problem that the Bayesian optimality can only be achieved on the subset of the battlefield that is defined in advance. In this section, we give the

RFS-based models which can solve these problems systematically.

The difficulty of RFS-based Bayesian inference is its computational complexity. To solve this problem, Mahler proposed the PHD (probability hypothesis density) filter. The PHD of the posterior probability density $f_{k|k}(\mathbf{X}_k|\mathbf{Z}_k)$ is denoted by $v_{k|k}(\mathbf{x}|\mathbf{Z}_k)$ and is a density function defined on the single object state $\mathbf{x} \in \mathbf{X}_0$ as follows:

$$v_{k|k}(\mathbf{x}|\mathbf{Z}_k) = \int f_{k|k}(\{\mathbf{x}\} \cup \mathbf{X}_k|\mathbf{Z}_k) \delta \mathbf{X}_k. \quad (1)$$

Here, \mathbf{Z}_k denotes the RFS of detection received at time k , and \mathbf{X}_k denotes the RFS of states at time k . We use the abbreviation $v_{k|k}(\mathbf{x}) = v_{k|k}(\mathbf{x}|\mathbf{Z}_k)$. In point theory, $v_{k|k}(\mathbf{x})$ is defined as the intensity density. It is not a probability density and represents the density of the expected number of points at \mathbf{x} . Given any subspace S of single object state space \mathbf{X}_0 , the integral $\int_S v_{k|k}(\mathbf{x}) d\mathbf{x}$ is the expected number of objects in S .

4.1. RFS-Based Battlefield Representation. We adopt the RFS-based battlefield representation; here, S denotes the RFS that represents the entire unknown battlefield. In addition, in order to assist in operational decision-making, we also relate the battlefield S to the UGV state X . RFS S_{k-1} , which is based on the UGV state $X_{0:k-1} = [X_0, X_1, \dots, X_{k-1}]$ at time $k-1$, is used to denote the battlefield that has been explored. S_{k-1} is the RFS of the battlefield states which consists of significant environmental cues and is the intersection of the union of all FoVs and the entire battlefield state. Thus, S_{k-1} can be represented as follows:

$$S_{k-1} = S \cap \text{FoV}(X_{0:k-1}). \quad (2)$$

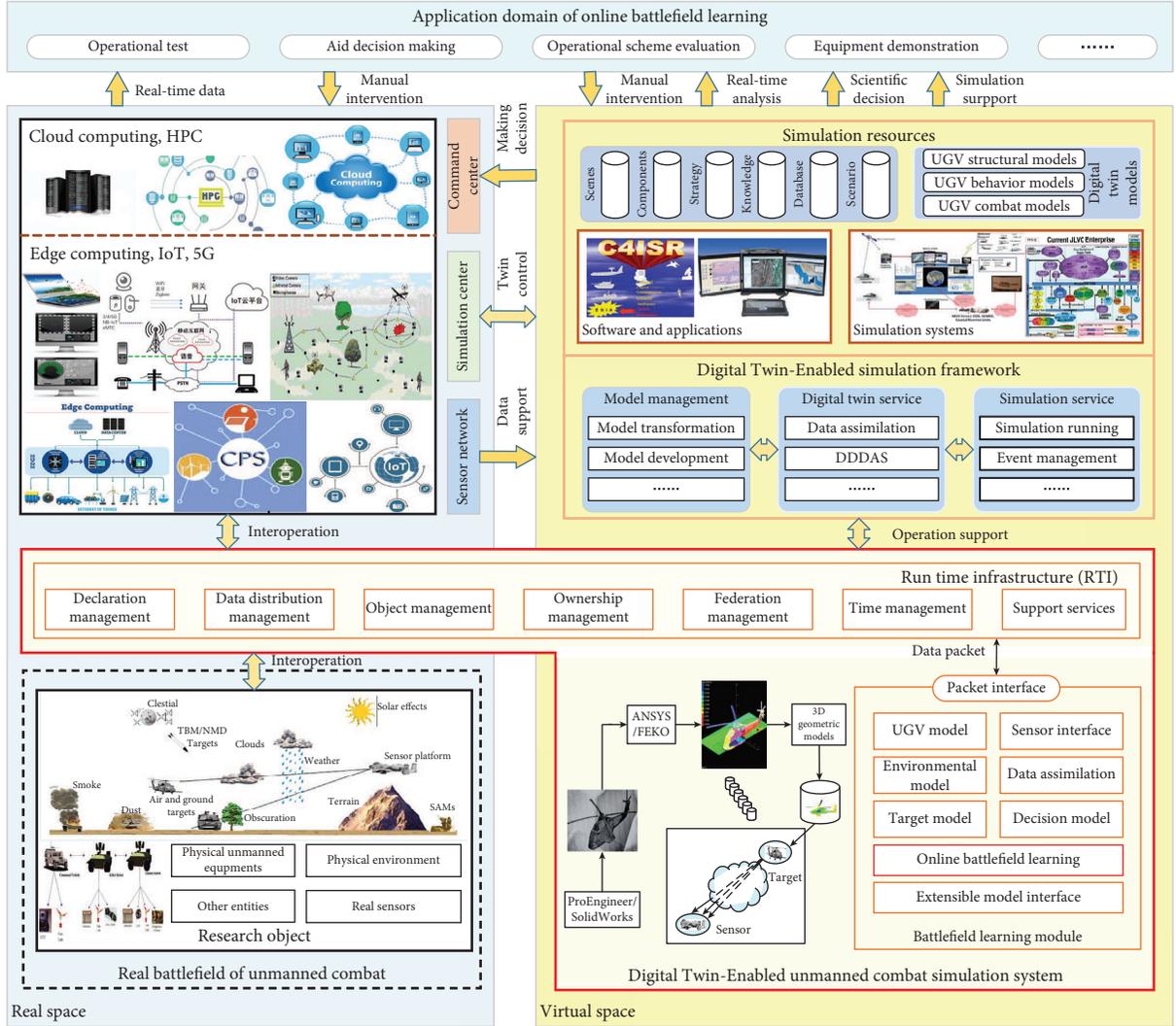


FIGURE 3: The system architecture of digital twin-enabled unmanned combat simulation.

Here, $\text{FoV}(X_{0:k-1}) = \text{FoV}(X_0) \cup \text{FoV}(X_1) \cup \dots \cup \text{FoV}(X_{k-1})$. FoV depends on the UGV states at time $k-1$. The learned battlefield at time k can be obtained based on S_{k-1} in the following way:

$$S_k = S_{k-1} \cup (\text{FoV}(X_k) \cap \bar{S}_{k-1}), \quad (3)$$

where $\bar{S}_{k-1} = S - S_{k-1}$ represents the unexplored battlefield namely, the set of significant environmental cues that are not in S_{k-1} . RFS $B_k(X_k)$ denotes the learned battlefield which has appeared in the FoV for the first time. Therefore, the battlefield transition process can be modeled as

$$f_S(S_k | S_{k-1}, X_k) = \sum_{W \in S_k} f_S(W | S_{k-1}) f_B(S_k - W | X_k), \quad (4)$$

where $f_S(W | S_{k-1})$ denotes the state transition density of battlefield from time $k-1$ to time k , and $f_B(S_k - W | X_k)$ denotes the density of the RFS $B(X_k)$.

4.2. RFS-Based UGV Motion Model. The location of UGV can be represented by the state vector $X = [x, y, \theta]^T$. The UGV motion model characterizes the state transition between $X_{k-1} = [x_{k-1}, y_{k-1}, \theta_{k-1}]^T$ and $X_k = [x_k, y_k, \theta_k]^T$ after inputting the control command \mathbf{u}_{k-1} . In this paper, we adopt the following two-dimensional motion model with translational and rotational displacement:

$$\begin{aligned} X_k &= \begin{bmatrix} x \\ y \\ \theta \end{bmatrix}_k \\ &= g(X_{k-1}, \mathbf{u}_{k-1}) + \delta \\ &= g\left(\begin{bmatrix} x \\ y \\ \theta \end{bmatrix}_k, \begin{bmatrix} \delta x \\ \delta y \\ \delta \theta \end{bmatrix}_k\right) + \omega_k, \quad \omega_k \sim (0, \mathbf{Q}). \end{aligned} \quad (5)$$

In this paper, the specific mathematical expression of g is employed as follows:

$$\begin{bmatrix} x_k \\ y_k \\ \theta_k \end{bmatrix} = \begin{bmatrix} x_{k-1} + \frac{t_{k-1}}{\gamma_{k-1}} (\cos(\theta_{k-1} + \gamma_{k-1}) - \cos(\theta_{k-1})) \\ y_{k-1} + \frac{t_{k-1}}{\gamma_{k-1}} (\sin(\theta_{k-1} + \gamma_{k-1}) - \sin(\theta_{k-1})) \\ \theta_{k-1} + \gamma_{k-1} \end{bmatrix} + \boldsymbol{\omega}_k. \quad (6)$$

Here, $\boldsymbol{\omega}_k$ is used to represent the uncertainty and noise, and $\mathbf{u}_{k-1} = [t_{k-1}, \gamma_{k-1}]^T$ is the control command that UGV received at time $k-1$.

4.3. RFS-Based Sensor Model. Given the current UGV state RFS X_k and the battlefield RFS S_k , the detection RFS can be described as follows:

$$Z_k = \bigcup_{s \in S_k} D_k(s, X_k) \cup C_k(X_k). \quad (7)$$

Here, $D_k(s, X_k)$ denotes the detection RFS related to the significant environmental cue with state s , and $C_k(X_k)$ denotes the clutters RFS, which is related to the UGV state X_k . Due to the uncertainty and randomness in the detection process, the number of elements in Z_k is random and may be different from the number of states in S_k .

The detection RFS $D_k(s, X_k)$ generated by battlefield state s is modeled by Bernoulli RFS. Therefore, there are two forms of $D_k(s, X_k)$. The first one is $D_k(s, X_k) = \emptyset$ and the probability is $1 - p_D(s, X_k)$. The other one is $D_k(s, X_k) = z$ and the probability is $p_D(s, X_k)g_k(z|s, X_k)$. X_k denotes the UGV state at time k , and $p_D(s, X_k)$ denotes the probability of generating detection from s . $g_k(z|s, X_k)$ models the likelihood that s generates detection z . In this paper, $p_D(s, X_k) = p_D$ if the significant environmental cue exists in the sensor's FoV, and $p_D(s, X_k) = 0$, otherwise.

Depending on X_k and S_k , the sensor's likelihood function for generating Z_k is represented as follows:

$$g_k(Z_k|X_k, S_k) = \sum_{W \subseteq Z_k} g_D(W|S_k, X_k)g_C(Z_k - W). \quad (8)$$

Here, $g_D(W|S_k, X_k)$ is the likelihood function of generating detection RFS D_k for RFS S_k , and $g_C(Z_k - W)$ is the probability density of the clutter RFS C_k .

In this paper, the range and bearing sensor is used. The detection generated by the two-dimensional environmental cues at location s can be modeled as follows:

$$\begin{aligned} \mathbf{z}_k &= \begin{bmatrix} r_k \\ b_k \end{bmatrix} \\ &= h(\mathbf{x}_k, s) + \mathbf{e}_k \\ &= \begin{bmatrix} \sqrt{(x_s - x)^2 + (y_s - y)^2} \\ \arctan\left(\frac{y_s - y}{x_s - x}\right) - \theta \end{bmatrix} + \mathbf{e}_k, \quad \mathbf{e}_k \sim (\mathbf{0}, \mathbf{R}). \end{aligned} \quad (9)$$

Here, $\mathbf{z}_k = [r_k \ b_k]^T$ is the range and bearing detection, $s = [x_s \ y_s]^T$ is the cue's position, and \mathbf{e}_k is the noise with covariance \mathbf{R} .

5. Learning Process and Its Implementation

In this section, we give the basic principles, design, and implementation of the proposed algorithm. The process of the proposed algorithm relies on sequentially propagating the joint posterior probability density of the RFS-based battlefield and the UGV state as detection arrives.

5.1. RFS-Based Learning Process. With the RFS-based battlefield modeling, the RFS-based Bayesian inference is used to jointly learn the environmental cues' locations and UGV state at every time step. The battlefield RFS can be characterized as follows:

$$p_{k|k}(S = \{s^1, s^2, \dots, s^{\hat{m}_k}\} | Z_{0:k}, X_{0:k}). \quad (10)$$

In this paper, we use $p_{k|k-1}(X_{0:k}, S_k | Z_{0:k-1}, U_{0:k-1}, X_0)$ to denote the predicted distribution of the battlefield state and $p_{k|k}(X_{0:k}, S_k | Z_{0:k}, U_{0:k-1}, X_0)$ to denote the a posteriori distribution of the battlefield state. The knowledge of the battlefield can be propagated by the following prediction and update process:

- (i) Predict the battlefield state by using the previous battlefield states and input parameters:

$$\begin{aligned} &p_{k|k-1}(X_{0:k}, S_k | Z_{0:k-1}, U_{0:k-1}, X_0) \\ &= \int f(X_{0:k}, S_k | X_{0:k-1}, S_{k-1}, U_{k-1}) \\ &\times p_{k-1|k-1}(X_{0:k-1}, S_{k-1} | Z_{0:k-1}, U_{0:k-2}, X_0) dX_{k-1}. \end{aligned} \quad (11)$$

- (ii) Update the battlefield state depending on the received detection RFS Z_k :

$$\begin{aligned}
& p_{k|k}(X_{0:k}, S_k | Z_{0:k}, U_{0:k-1}, X_0) \\
&= \frac{g_k(Z_k | S_k, X_k) p_{k|k-1}(X_{0:k}, S_k | Z_{0:k-1}, U_{0:k-1}, X_0)}{\iint g_k(Z_k | S_k, X_k) p_{k|k-1}(X_{0:k}, S_k | Z_{0:k-1}, U_{0:k-1}, X_0) dX_k \delta S_k}.
\end{aligned} \quad (12)$$

Here, δ implies set integration.

In this paper, the PHD filter is employed to implement the RFS-based Bayesian recursion [24, 33–35]. We modify and extend the Gaussian mixture-based PHD filter with a particle filter. The Gaussian mixture PHD filter is applied to learn the number and locations of the environmental cues, and the particle filter is applied to learn the UGV state at the same time. The computing process of online battlefield learning by modifying PHD filter is shown in Figure 4. The Bayesian recursion encapsulates the inherent uncertainty of the number of significant environmental cues that may be caused by detection uncertainty, clutters, UGV maneuvers, and the uncertainty related to detection noises.

The main challenge of online battlefield learning is how to learn the number and location of environmental cues while estimating the UGV state at the same time. In this paper, we partition the battlefield state into two kinds: s for environmental cues and $X_k^{(i)}$ for UGV movement. We can analytically integrate out s provided that we know $X_{0:k}^{(i)}$. This means that even though we only have the sample sets $X_{0:k}^{(i)}$, we can also represent $p(s|X_{0:k}^{(i)})$ successfully. Thus, each particle $X_k^{(i)}$ represents a value for s . The advantage of this approach is that we can reduce the dimensionality of state space in which we are sampling and reduce the error of the learned battlefield.

Here, the Gaussian mixture PHD filter is applied to propagate each PHD that depends on the UGV state. The location of environmental cues in the battlefield is characterized by the Gaussian components of the mixture, and the number of cues in the battlefield is characterized by masses of all the Gaussian components. In this paper, the PHD at time $k-1$ is characterized by the following N particles:

$$\{w_{k-1|k-1}^{(i)}, X_{0:k-1}^{(i)}, v_{k-1|k-1}^{(i)}(s|X_{0:k-1}^{(i)})\}_{i=1}^N, \quad (13)$$

where $X_{0:k-1}^{(i)} = [X_0^{(i)}, X_1^{(i)}, X_2^{(i)}, \dots, X_{k-1}^{(i)}]$ is the i th hypothesized UGV state set, $w_{k-1|k-1}^{(i)}$ denotes the weight, and $v_{k-1|k-1}^{(i)}(s|X_{0:k-1}^{(i)})$ is the related PHD. The posterior distribution is approximated by the following set of weighted particles:

$$\{w_{k|k}^{(i)}, X_{0:k}^{(i)}, v_{k|k}^{(i)}(s|X_{0:k}^{(i)})\}_{i=1}^N. \quad (14)$$

In this paper, $v_{k-1|k-1}^{(i)}(s|X_{k-1}^{(i)})$ is the prior PHD of the battlefield states for the i th particle related to the i th UGV trajectory. $v_{k-1|k-1}^{(i)}(s|X_{k-1}^{(i)})$ can be represented by the following Gaussian mixture:

$$v_{k-1|k-1}^{(i)}(s|X_{k-1}^{(i)}) = \sum_{j=1}^{J_{k-1|k-1}^{(i)}} \eta_{k-1|k-1}^{(i,j)} N(s; \mu_{k-1|k-1}^{(i,j)}, P_{k-1|k-1}^{(i,j)}), \quad (15)$$

which consists of $J_{k-1|k-1}^{(i)}$ Gaussian components. For j th Gaussian component, $\eta_{k-1|k-1}^{(i,j)}$ is predicted weight, $\mu_{k-1|k-1}^{(i,j)}$ is mean, and $P_{k-1|k-1}^{(i,j)}$ is covariance. The PHD of the new environmental cue for the sampled state $X_k^{(i)}$ at time k is represented by $b(s|Z_{k-1}, X_k^{(i)})$. $b(s|Z_{k-1}, X_k^{(i)})$ is also a Gaussian mixture and can be represented as follows:

$$b(s|Z_{k-1}, X_k^{(i)}) = \sum_{j=1}^{J_{b,k}^{(i)}} \eta_{b,k}^{(i,j)} N(s; \mu_{b,k}^{(i,j)}, P_{b,k}^{(i,j)}), \quad (16)$$

where $J_{b,k}^{(i)}$ is the number of the Gaussian components of the new PHD at time k , and $\eta_{b,k}^{(i,j)}$, $\mu_{b,k}^{(i,j)}$, and $P_{b,k}^{(i,j)}$ are the corresponding Gaussian parameters. The predicted PHD is therefore also a Gaussian mixture and can be represented as follows:

$$v_{k|k-1}(s|X_k^{(i)}) = \sum_{j=1}^{J_{k|k-1}^{(i)}} \eta_{k|k-1}^{(i,j)} N(s; \mu_{k|k-1}^{(i,j)}, P_{k|k-1}^{(i,j)}). \quad (17)$$

Here, $v_{k|k-1}(s|X_k^{(i)})$ is composed of $J_{k|k-1}^{(i)} = J_{k-1|k-1}^{(i)} + J_{b,k}^{(i)}$ Gaussian components that represent the union of the prior PHD $v_{k-1|k-1}(s|X_{k-1}^{(i)})$ and the PHD of new environmental cues. Since the detection function can also be represented by a Gaussian mixture, the posterior PHD $v_{k|k}(s|X_k^{(i)})$ can be represented by a Gaussian mixture as follows:

$$\begin{aligned}
v_{k|k}(s|X_k^{(i)}) &= v_{k|k-1}(s|X_k^{(i)}) \\
&\cdot \left[1 - p_D(s|X_k^{(i)}) + \sum_{z \in Z_k} \sum_{j=1}^{J_{k|k-1}^{(i)}} v_{G,k}^{(i,j)}(z, s|X_k^{(i)}) \right].
\end{aligned} \quad (18)$$

The components of equation (18) are given as follows:

$$v_{G,k}^{(i,j)}(z, s|X_k^{(i)}) = \eta_{k|k}^{(i,j)}(z|X_k^{(i)}) N(s; \mu_{k|k}^{(i,j)}, P_{k|k}^{(i,j)}),$$

$$\eta_{k|k}^{(i,j)}(z|X_k^{(i)}) = \frac{p_D(s|X_k^{(i)}) \eta_{k|k-1}^{(i,j)} q^{(i,j)}(z, X_k^{(i)})}{c(z) + \sum_{l=1}^{J_{k|k-1}^{(i)}} p_D(s|X_k^{(i)}) \eta_{k|k-1}^{(i,l)} q^{(i,l)}(z, X_k^{(i)})}. \quad (19)$$

Here, $q^{(i,j)}(z, X_k^{(i)}) = N(z; H_k \mu_{k|k}^{(i,j)}, S_{k|k}^{(i,j)})$. The terms $\mu_{k|k}^{(i,j)}$, $P_{k|k}^{(i,j)}$, and $S_{k|k}^{(i,j)}$ can be got through standard Kalman filters; here, we adopt the unscented Kalman filter [36].

We assume that the number of clutters in C_k complies with the Poisson distribution, and the elements comply with uniform distribution over the battlefield state space. Then, the clutter PHD can be represented by $c(z) = \lambda_c U(z)$; here, λ_c denotes the averaged number of clutters, and $U(z)$ complies with a uniform distribution. In order to reduce the amount of calculation, we use pruning and merging methods to reduce the number of Gaussian components of the updated distribution [37].

The posterior UGV state $p_k(X_{1:k})$ is sampled by $\{\bar{w}_{k|k}^{(i)}, \bar{X}_k^{(i)}\}_{i=1}^N$ with

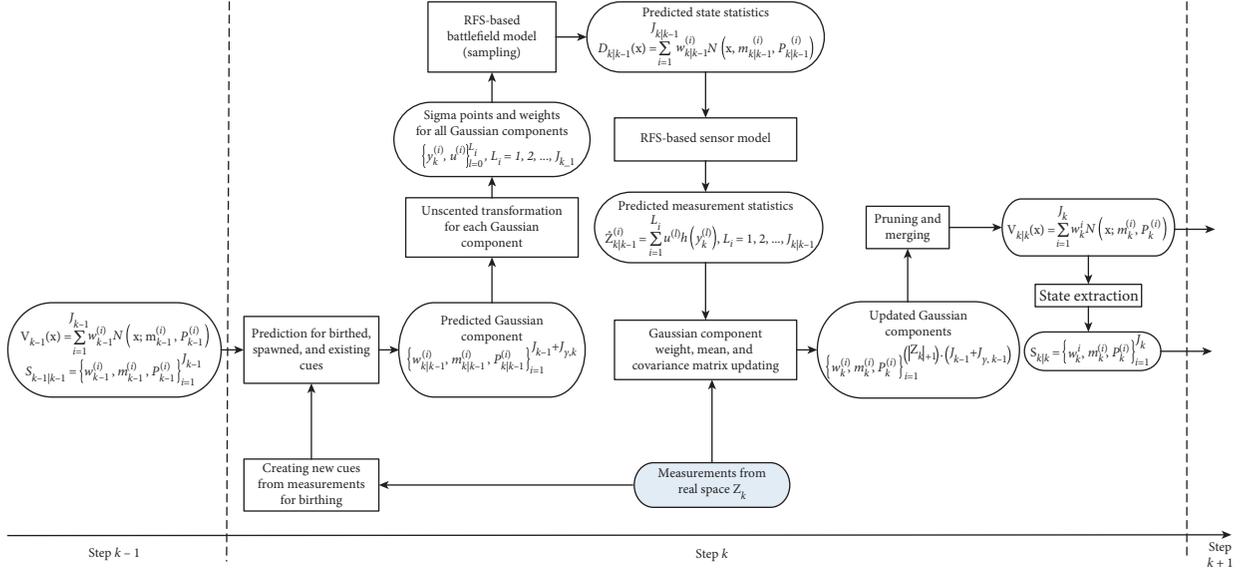


FIGURE 4: The online battlefield learning process based on Gaussian mixture-based PHD filter.

$$\begin{aligned} \tilde{X}_k^{(i)} &\sim f\left(\tilde{X}_k^{(i)} | X_{k-1}^{(i)}, U_{k-1}\right), \\ \tilde{w}_k^{(i)} &= \frac{g_k\left(Z_k | Z_{0:k-1}, \tilde{X}_{0:k}^{(i)}\right) f\left(\tilde{X}_k^{(i)} | X_{k-1}^{(i)}, U_{k-1}\right)}{f\left(\tilde{X}_k^{(i)} | X_{0:k-1}^{(i)}, U_{k-1}\right)} w_{k-1}^{(i)}. \end{aligned} \quad (20)$$

The weights should be normalized as $\sum_{i=1}^N \tilde{w}_k^{(i)} = 1$. With the resampling step [24], we can get the resampled particles

$$g_k(Z_k | Z_{0:k-1}, X_{0:k}) \approx \frac{1}{\Gamma} \left[\left((1 - p_D(\bar{s} | X_k)) \kappa_k^{Z_k} + p_D(\bar{s} | X_k) \sum_{z \in Z_k} \kappa_k^{Z_k \setminus \{z\}} g_k(z | \bar{s}, X_k) \right) v_{k|k-1}(\bar{s} | X_{0:k}) \right], \quad (22)$$

with

$$\Gamma = \exp\left(\hat{m}_{k|k-1} - \hat{m}_{k|k} + \int c_k(z) dz\right) v_{k|k}(\bar{s} | X_{0:k}). \quad (23)$$

$$\text{Here, } \hat{m}_{k|k-1} = \sum_{j=1}^{J_{k|k-1}} \eta_{k|k-1}^{(i,j)} \text{ and } \hat{m}_{k|k} = \sum_{j=1}^{J_{k|k}} \eta_{k|k}^{(i,j)}.$$

5.2. Implementation. According to the learning process given above, we give the concrete realization method of the proposed algorithm in this section. We use C++ to write the experimental program for this algorithm. The C++ library dependencies such as Eigen (version 3.0.0), Boost (version 1.5.3), and gtest are also used. In order to detail the implementation of the proposed algorithm, the flow diagram of the proposed algorithm is presented in Figure 5. The concrete steps are described in Algorithm 1.

The computational complexity of the proposed algorithm is $O(m_k \cdot |Z_k| \cdot N)$ and is linear in the number of landmarks (in the FoV), as well as the number of detections and number of particles for the UGV state.

$\left\{ \tilde{w}_k^{(i)}, \tilde{X}_k^{(i)} \right\}_{i=1}^N$. By choosing the UGV transition density as the proposal density, we get the weight as follows:

$$\tilde{w}_k^{(i)} = g_k\left(Z_k | Z_{0:k-1}, \tilde{X}_{0:k}^{(i)}\right) w_{k-1}^{(i)}. \quad (21)$$

By assuming that there is only one environmental cue \bar{s} in the battlefield, then, we can get

6. Experiments

In this section, a group of experiments are conducted to quantitatively verify the effectiveness and analyze the performance of the proposed algorithm. The virtual machine we used to run our experiments has 4 G of RAM and 6 3.40 GHz Intel CPUs and runs on Ubuntu 14.04 OS. The experimental data used to support the findings of this study are included within the article. The parameters used in this experiment are given in Section 6.1, and the models used in this experiment are given in Section 4.

6.1. Experimental Setup. As shown in Figure 6, the UGV patrols in a simulated two-dimensional space. The known ground truth (including the UGV states and locations of landmarks) is generated by the simulation models. The black dots represent the real locations of landmarks, and the black dashed line represents the real UGV states. The number of clutters complies with the Poisson distribution, and the

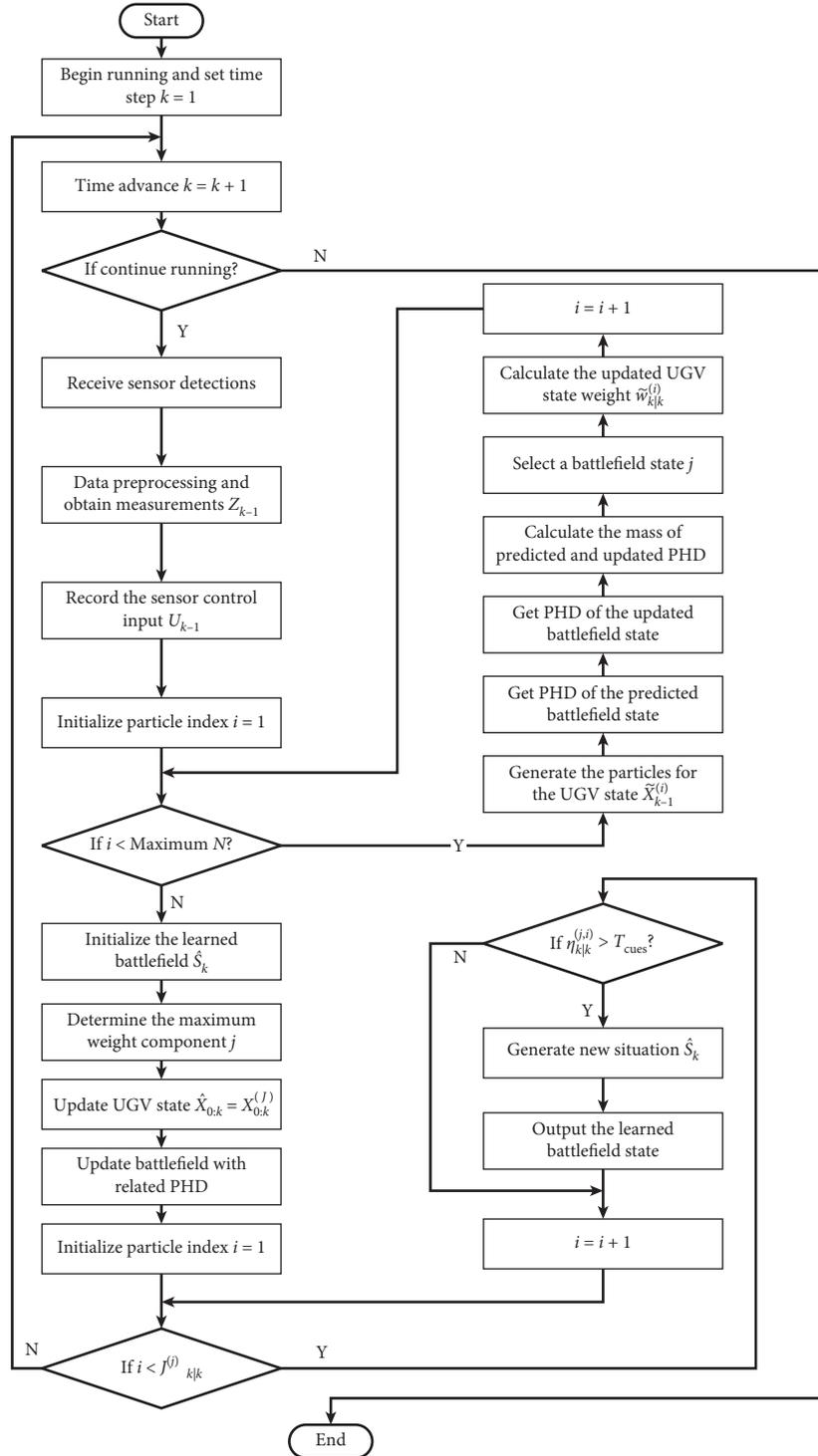


FIGURE 5: The flow diagram of the RFS-based online battlefield learning algorithm.

clutter PHD is uniformly distributed. Table 1 shows some important parameters for the simulation models to generate the ground truth.

The sensor used in this experiment is the range-bearing sensor that can detect landmarks with distances of 5 m to 30 m in any direction. The range measurement standard deviation (std) is 1 m and the bearing measurement std is

2 deg. The maximum FoV of the sensor used by the UGV is 10 m and 360 deg.

6.2. *Results and Analysis.* The experimental results are shown in Figure 6, the red dashed line represents the learned UGV states, and the red points represent the learned

Input: $(\{w_{k-1}^{(i)}, X_{0:k-1}^{(i)}, v_{k-1}^{(i)}(s|Z_{k-1}, X_{k-1}^{(i)})\}_{i=1}^N, Z_{k-1}, U_{k-1})$
Output: $(\hat{X}_{0:k}, \hat{S}_k)$

- (1) for $i = 1$ to N do
- (2) Generate the particles for the UGV state, $\tilde{X}_{k-1}^{(i)} \sim f(\tilde{X}_{k-1}^{(i)}|X_{k-1}^{(i)}, U_{k-1})$;
- (3) Get predicted battlefield PHD through the predict step of PHD filter;
- (4) Get updated battlefield PHD through the update step of PHD filter;
- (5) Get predicted PHD mass $\hat{m}_{k|k-1} = \sum_{j=1}^{J_{k|k-1}^{(i)}} \eta_{k|k-1}^{(i,j)}$;
- (6) Get updated PHD mass $\hat{m}_{k|k} = \sum_{j=1}^{J_{k|k}^{(i)}} \eta_{k|k}^{(i,j)}$;
- (7) Select a given battlefield state $j = \{i = 1, \dots, J_k^{(i)} | m^{(i,j)} = \bar{m}\}$;
- (8) $a = (1 - p_D)c(z)^{|Z_k|} + p_D \eta_{k|k-1}^{(i,j)} \times \sum_{z \in Z_k} (c(z)^{|Z_k|-1})N(z; z_{k|k-1}, S_k^{(i,j)})$;
- (9) $b = \exp^{\hat{m}_{k|k-1} - \hat{m}_{k|k} + \lambda_c} \eta_{k|k}^{(i,j)}$;
- (10) Get updated UGV state weight $\tilde{w}_{k|k}^{(i)} = (a/b)\tilde{w}_{k|k-1}^{(i)}$;
- (11) end for
- (12) Initialize the learned battlefield state $\hat{S}_k = \emptyset, I = \{1, \dots, N\}$;
- (13) Determine the maximum weight component $j = \operatorname{argmax}_{j \in I} w_{k|k}^{(j)}$;
- (14) Update UGV state with $\hat{X}_{0:k} = X_{0:k}^{(j)}$;
- (15) Update battlefield state according to the related PHD:
- (16) for $i = 1$ to $J_{k|k}^{(j)}$ do
- (17) if $\eta_{k|k}^{(j,i)} > T_{\text{cues}}$, here T_{cues} is the landmark existence threshold
- (18) Generate new battlefield state by $\hat{S}_k = [\hat{S}_k \mu_{k|k}^{(j,i)}]$;
- (19) end if
- (20) end for
- (21) Return $(\hat{X}_{0:k}, \hat{S}_k)$.

ALGORITHM 1: RFS-based online battlefield learning algorithm.

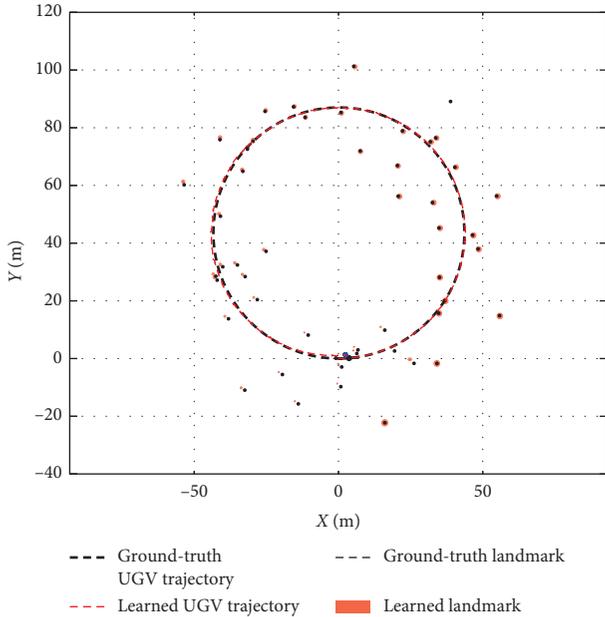


FIGURE 6: The real and learned battlefield states.

locations of landmarks. The collection of red dashed lines and points represents the battlefield states. The results successfully confirm that the proposed algorithm can learn the battlefield states by using sensor detection at runtime.

In order to quantitatively evaluate the performance of the proposed algorithm, we give the errors of the learned battlefield states in Figures 7 and 8. Figures 7(a) and 7(b) give the errors of the learned number and locations of landmarks. The errors of

locations are represented by optimal subpattern assignment (OSPA) distance [38]. We can find out that the performance of the proposed algorithm can satisfy the requirements of simulation and evaluation in unmanned combat.

Consider two sets $\mathbf{X} = \{\mathbf{x}_1, \dots, \mathbf{x}_m\}$ and $\mathbf{Y} = \{\mathbf{y}_1, \dots, \mathbf{y}_n\}$, where $m, n \in \mathbb{N}_0 = \{0, 1, \dots\}$. Vectors $\mathbf{x} \in \mathbf{X}$ and $\mathbf{y} \in \mathbf{Y}$ are taking values from the battlefield state space. The OSPA metric is defined as a distance between sets \mathbf{X} and \mathbf{Y} . The OSPA distance of order $1 \leq p \leq \infty$, with the cut-off parameter c , is defined for $m \leq n$ as follows:

$$d_{p,c}(\mathbf{X}, \mathbf{Y}) = \left[\frac{1}{n} \left(\min_{\pi \in \Pi_n} \sum_{i=1}^m (d_c(\mathbf{x}_i, \mathbf{y}_{\pi(i)}))^p + (n-m) \cdot c^p \right) \right]^{(1/p)}, \quad (24)$$

where Π_n represents the permutations set of length m with elements taken from $\{1, 2, \dots, n\}$.

The errors of learned UGV states are shown in Figures 8(a) and 8(b). We can find out that the proposed algorithm can generate the learned UGV states with acceptable accuracy. But the errors are increased as time advance. This is caused by the cumulative errors of the UGV states.

In order to analyze how detection parameters affect the proposed algorithm, the averaged errors of the UGV states and landmarks are generated with different probabilities of detection p_D from 0.1 to 0.99 and clutter intensity λ_c from 0.0001 to 1. For each pair of parameters, 10 simulation runs were carried out. Here, cardinalized optimal linear assignment (COLA) is used to evaluate the errors for the learned landmarks. From Figures 9(a) and 9(b), we can find out that the errors of the learned UGV states increase as p_D

TABLE 1: Partial experimental parameters.

Parameter	Velocity input std. (m/s)	Steering input std. (deg)	Detection probability p_D	Clutter rate λ_c	Particle number N	Landmark existence threshold T_{cue}
Value	2	2	0.90	0.0001	100	0.5

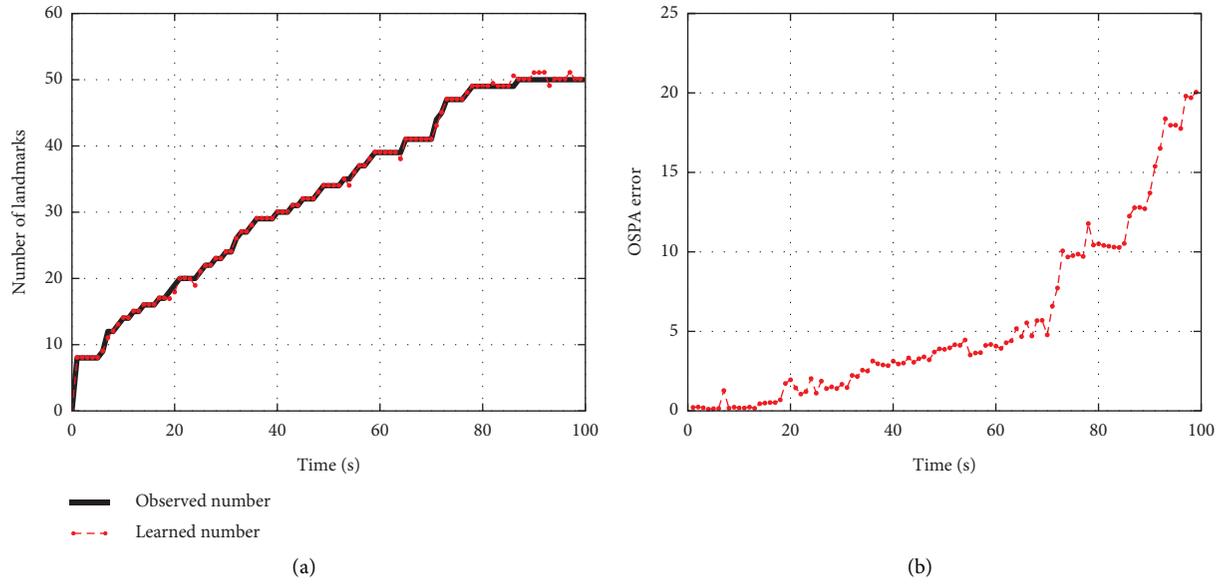


FIGURE 7: The errors of the learned number and locations of landmarks. (a) The learned and observed number of landmarks. (b) The OSPA errors of the learned landmarks.

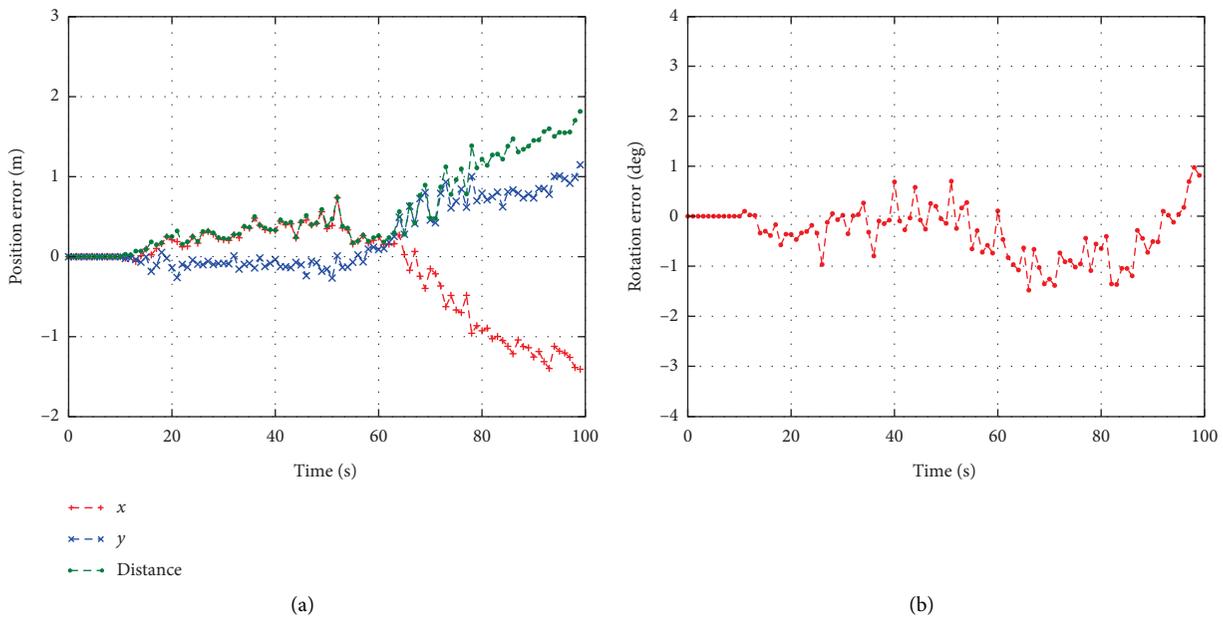
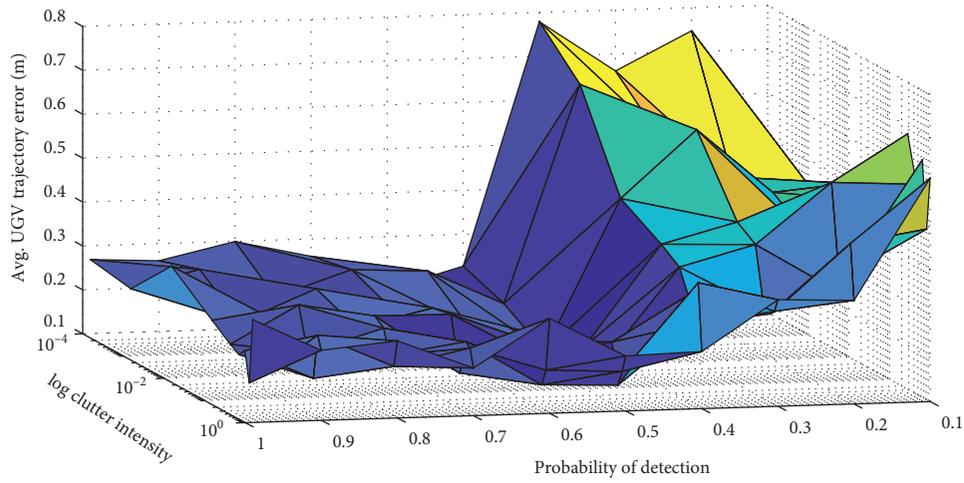


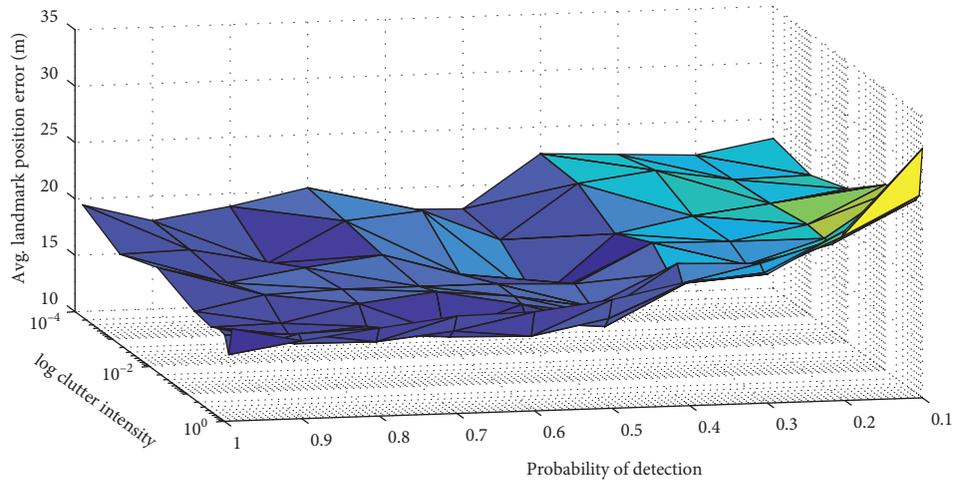
FIGURE 8: The errors of the learned UGV states. (a) Euclidean errors of the learned UGV states. (b) Orientation errors of the learned UGV states.

decreases. The errors of the learned landmarks only increase slightly as P_D decreases. The increase of λ_c will increase the errors of the learned locations of landmarks, but the effect on the errors of learned UGV states is quite small.

In order to apply the proposed algorithm in real unmanned combat applications, the time cost should be fully evaluated. As shown in Figure 10, we record 10 simulation runs for each pair of detection probability and clutter

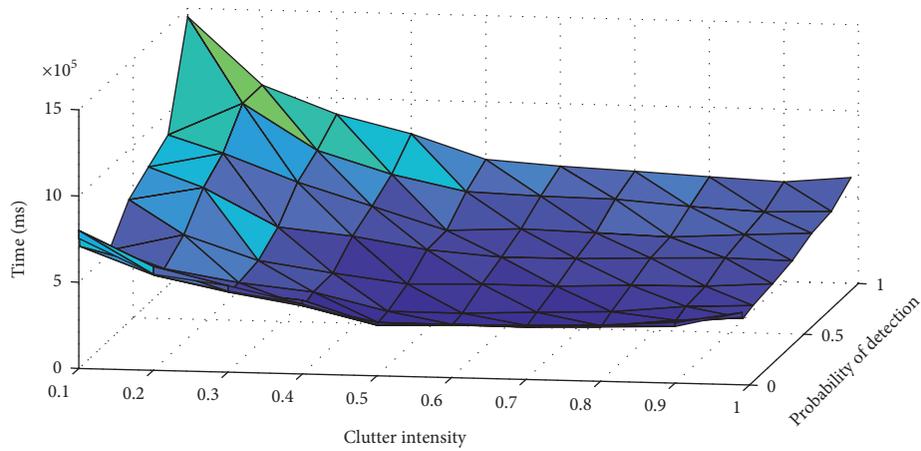


(a)



(b)

FIGURE 9: The performance for varying values of probability of detection and clutter intensity. (a) Averaged errors for the learned UGV states. (b) Averaged COLA errors for learned landmarks.



(a)

FIGURE 10: Continued.

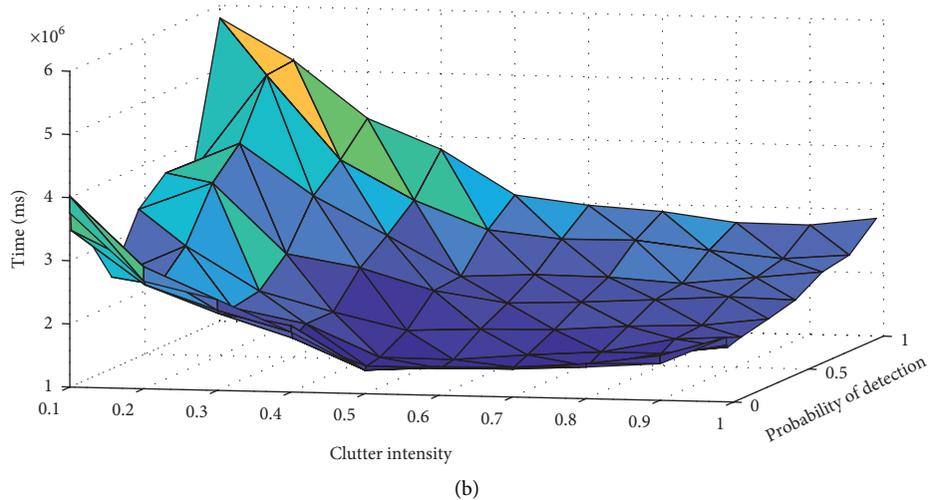


FIGURE 10: Time cost for varying values of detection probability and clutter intensity. (a) Averaged time cost of the algorithm. (b) Averaged time cost of CPU while the algorithm running.

intensity, and each simulation run consists of 1000 time steps. The averaged time costs of the proposed algorithm and CPU are shown in Figures 10(a) and 10(b). We can find that the increase of detection probability will increase the time cost, and the decrease of clutter intensity will also increase the time cost. The average time cost for each time step is about 500 ms, and it can satisfy many unmanned combat applications very well.

7. Conclusions

Digital twin technology enables real-time dynamic interaction between the real battlefield and the simulation system. Our main contribution is proposing a new online battlefield learning algorithm based on RFS to enable the application of the digital twin in unmanned combat. The digital twin has a broad application prospect in unmanned combat and greatly promotes the innovation of unmanned combat mode. Since the implementation of the digital twin in unmanned combat depends on battlefield understanding, an effective battlefield learning algorithm is quite important. By adopting the RFS-based representation of the battlefield, the proposed algorithm can overcome the limitations of the traditional vector-based representation. The performance of the proposed algorithm is verified by using two groups of experiments. This paper is the first attempt for applying the digital twin to the unmanned combat area and has practical significance for implementing the digital twin in many other areas.

Abbreviations

CGF: Computer-generated force
 COA: Courses of action
 COLA: Cardinalized optimal linear assignment
 EKF: Extended Kalman filter
 FoV: Field of view
 IoT: Internet of things
 OSPA: Optimal subpattern assignment

PHD: Probability hypothesis density
 RFS: Random finite set
 UGV: Unmanned ground vehicle.

Data Availability

The experimental data used to support the findings of this study are included within the article.

Conflicts of Interest

The authors declare no conflicts of interest.

Authors' Contributions

Peng Wang conceived, designed, and performed the simulations and wrote the manuscript. Jiancheng Zhu and Yong Peng provided the basic ideas and analyzed the experimental results. Mei Yang helped to perform the experiments. Ge Li and Yong Peng reviewed the manuscript.

Acknowledgments

The authors would like to acknowledge the support of the Young Elite Scientists Sponsorship Program of China Association of Science and Technology.

References

- [1] E. P. Blasch, R. Breton, P. Valin, and E. Bosse, "User information fusion decision making analysis with the C-OODA model," in *Proceedings of 14th International Conference on Information Fusion*, pp. 1–8, Chicago, IL, USA, July 2011.
- [2] N. Wang, X. Jin, and M. J. Er, "A multilayer path planner for a USV under complex marine environments," *Ocean Engineering*, vol. 184, no. 15, pp. 1–10, 2019.
- [3] J. Lee, S. Shin, M. Park, and C. Kim, "Agent-based simulation and its application to analyze combat effectiveness in network-centric warfare considering communication failure

- environments,” *Mathematical Problems in Engineering*, vol. 2018, Article ID 2730671, 9 pages, 2018.
- [4] J. Luo, W. Zhang, W. Gao, Z. Liao, X. Ji, and X. Gu, “Opponent-aware planning with admissible privacy preserving for UGV security patrol under contested environment,” *Electronics*, vol. 9, no. 1, p. 5, 2020.
- [5] A. Bonci, M. Pirani, and S. Longhi, “Tiny cyber-physical systems for performance improvement in the factory of the future,” *IEEE Transactions on Industrial Informatics*, vol. 15, no. 3, pp. 1598–1608, 2019.
- [6] C. Thule, K. Lausdahl, C. Gomes, G. Meisl, and P. G. Larsen, “Maestro: the INTO-CPS co-simulation framework,” *Simulation Modelling Practice and Theory*, vol. 92, pp. 45–61, 2019.
- [7] Z. B. Rivera, M. C. De Simone, and D. Guida, “Unmanned ground vehicle modelling in gazebo/ROS-based environments,” *Machines*, vol. 7, no. 2, p. 42, 2019.
- [8] A. Francisco, N. Mohammadi, and J. E. Taylor, “Smart city digital twin-enabled energy management: toward real-time urban building energy benchmarking,” *Journal of Management in Engineering*, vol. 36, no. 2, Article ID 04019045, 2020.
- [9] M. Abramovici, J. C. Göbel, and H. B. Dang, “Semantic data management for the development and continuous reconfiguration of smart products and systems,” *CIRP Annals*, vol. 65, no. 1, pp. 185–188, 2016.
- [10] M. Schlus and J. Rossmann, “From simulation to experimentable digital twins—simulation based development and operation of complex technical systems,” vol. 2016, pp. 273–278, in *Proceedings of 2nd IEEE International Symposium on Systems Engineering*, vol. 2016, , IEEE, Edinburgh, UK, October 2016.
- [11] H. R. Faghihi, M. A. Fazli, and J. Habibi, “Hybrid-learning approach toward situation recognition and handling,” 2019, <https://arxiv.org/abs/1906.09816>.
- [12] C. Mundutéguy and I. Ragot-Court, “A contribution to situation awareness analysis,” *Human Factors: The Journal of the Human Factors and Ergonomics Society*, vol. 53, no. 6, pp. 687–702, 2011.
- [13] B. Handaoui, M. Alkalbani, T. Znati, and R. Ammar, “Unleashing the power of participatory iot with block chains for increased safety and situation awareness of smart cities,” *IEEE Network*, vol. 34, no. 2, pp. 202–209, 2020.
- [14] R. A. Bell, “Unmanned ground vehicles and EO-IR sensors for border patrol,” *Optics and Photonics in Global Homeland Security III*, International Society for Optics and Photonics, vol. 6540Bellingham, DC, USA, Article ID 65400B, 2007.
- [15] X. Xu, M. Yang, and G. Li, “Adaptive CGF commander behavior modeling through HTN guided Monte Carlo tree search,” *Journal of Systems Science and Systems Engineering*, vol. 27, no. 2, pp. 231–249, 2018.
- [16] S. Boschert and R. Rosen, “Digital twin—the simulation aspect,” in *Mechatronic Futures: Challenges and Solutions for Mechatronic Systems and Their Designers*, pp. 59–74, Springer International Publishing, Berlin, Germany, 2016.
- [17] J. Cheng, H. Zhang, F. Tao, and C.-F. Juang, “DT-II: digital twin enhanced industrial internet reference framework towards smart manufacturing,” *Robotics and Computer-Integrated Manufacturing*, vol. 62, no. 4, Article ID 101881, 2020.
- [18] J. Hochhalter, “Coupling damage-sensing particles to the digital twin concept,” 2018, <https://ntrs.nasa.gov/search.jsp?R=20140006408>.
- [19] S. Boschert, “Next generation digital twin,” in *Proceedings of TMCE 2018*, Las Palmas de Gran Canaria, Spain, May 2018.
- [20] Y. Lu, C. Liu, K. I.-K. Wang, H. Huang, and X. Xu, “Digital twin-driven smart manufacturing: connotation, reference model, applications and research issues,” *Robotics and Computer-Integrated Manufacturing*, vol. 61, Article ID 101837, 2020.
- [21] P. Wang, M. Yang, Y. Peng, J. Zhu, R. Ju, and Q. Yin, “Sensor control in anti-submarine warfare—a digital twin and random finite sets based approach,” *Entropy*, vol. 21, no. 8, p. 767, 2019.
- [22] P. M. Karve, Y. Guo, B. Kapusuzoglu, S. Mahadevan, and M. A. Haile, “Digital twin approach for damage-tolerant mission planning under uncertainty,” *Engineering Fracture Mechanics*, vol. 225, Article ID 106766, 2020.
- [23] L. Y. Yang, S. Y. Chen, X. Wang, J. Zhang, and C. H. Wang, “Digital twins and parallel systems: state of the art, comparisons and prospect,” *Acta Automatica Sinica*, vol. 45, no. 11, pp. 2001–2031, 2019.
- [24] R. P. S. Mahler, “Multitarget Bayes filtering via first-order multitarget moments,” *IEEE Transactions on Aerospace and Electronic Systems*, vol. 39, no. 4, pp. 1152–1178, 2003.
- [25] F. Tao, J. Cheng, Q. Qi, M. Zhang, H. Zhang, and F. Sui, “Digital twin-driven product design, manufacturing and service with big data,” *The International Journal of Advanced Manufacturing Technology*, vol. 94, pp. 3563–3576, 2018.
- [26] F. Tao and Q. Qi, “Make more digital twins,” *Nature*, vol. 573, no. 7775, pp. 490–491, 2019.
- [27] Y. Zheng, S. Yang, and H. Cheng, “An application framework of digital twin and its case study,” *Journal of Ambient Intelligence and Humanized Computing*, vol. 10, no. 3, pp. 1141–1153, 2019.
- [28] F. Tao, H. Zhang, A. Liu, and A. Y. C. Nee, “Digital twin in industry: state-of-the-art,” *IEEE Transactions on Industrial Informatics*, vol. 15, no. 4, pp. 2405–2415, 2019.
- [29] S. Weyer, T. Meyer, M. Ohmer, D. Gorecky, and D. Zühlke, “Future modeling and simulation of CPS-based factories: an example from the automotive industry,” *IFAC-PapersOnLine*, vol. 49, no. 31, pp. 97–102, 2016.
- [30] D. Ali, S. S. Kanhere, and K. Raja, “Multi-agent systems: a survey,” *IEEE Access*, vol. 6, pp. 28573–28593, 2018.
- [31] D. A. Kai, L. I. Tiancheng, Y. Zhu, H. Fan, and Q. Fu, “Recent advances in multisensor multitarget tracking using random finite set,” *Frontiers of Information Technology & Electronic Engineering*, vol. 22, no. 1, pp. 5–24, 2021.
- [32] P. Wang, G. Li, Y. Peng, and R. Ju, “Random finite set based parameter estimation algorithm for identifying stochastic systems,” *Entropy*, vol. 20, no. 8, 569 pages, 2018.
- [33] H. Sidenbladh, “Multi-target particle filtering for the probability hypothesis density,” in *Proceedings of the 6th International Conference of Information Fusion 2003*, p. 806, Cairns, Australia, July 2003.
- [34] T. Zajic and R. P. S. Mahler, “Particle-systems implementation of the PHD multitarget-tracking filter,” *Signal Processing, Sensor Fusion, and Target Recognition XII*, vol. 5096, no. 1, pp. 291–299, 2003.

- [35] C. Musso, N. Oudjane, and F. LeGland, "Improving regularised particle filters," *Sequential Monte Carlo Methods in Practice*, Springer-Verlag, New York, NY, USA, 2001.
- [36] P. Wang, G. Li, R. Ju, and Y. Peng, "Random finite set based data assimilation for dynamic data driven simulation of maritime pirate activity," *Mathematical Problems in Engineering*, vol. 2017, Article ID 5307219, 18 pages, 2017.
- [37] R. Mahler, "PHD filters of higher order in target number," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 43, no. 4, pp. 1523–1543, 2007.
- [38] D. Schuhmacher, B.-T. Vo, and B.-N. Vo, "A consistent metric for performance evaluation of multi-object filters," *IEEE Transactions on Signal Processing*, vol. 56, no. 8, pp. 3447–3457, 2008.