IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 4 March 2023, accepted 11 May 2023, date of publication 22 May 2023, date of current version 30 May 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3278597

RESEARCH ARTICLE

Confrontation and Obstacle-Avoidance of Unmanned Vehicles Based on Progressive Reinforcement Learning

CHENGDONG MA[1], JIANAN LIU[1,2], SAICHAO HE[1,2], WENJING HONG[1,2], AND JIA SHI[1,2]
[1]Department of Chemical and Biochemical Engineering, College of Chemistry and Chemical Engineering, Xiamen University, Xiamen 361005, China
[2]Institute of Artificial Intelligence, Xiamen University, Xiamen 361005, China

Corresponding author: Jia Shi (jshi@xmu.edu.cn)

ABSTRACT The core technique of unmanned vehicle systems is the autonomous maneuvering decision, which not only determines the applications of unmanned vehicles but also is the critical technique many countries are competing to develop. Reinforcement Learning (RL) is the potential design method for autonomous maneuvering decision-making systems. Nevertheless, in the face of complex decision-making tasks, it is still challenging to master the optimal policy due to the low learning efficiency caused by the complex environment, high dimensional state, and sparse reward. Inspired by the human learning process from simple to complex, we propose a novel progressive deep RL algorithm for policy optimization in unmanned autonomous decision-making systems in this paper. The proposed algorithm divides the training of the autonomous maneuvering decision into a sequence of curricula with learning tasks from simple to complex. Finally, through the self-play stage, the iterative optimization of the policy is realized. Furthermore, the confrontation environment with two unmanned vehicles with obstacles is analyzed and modeled. Finally, the simulation leads to the one-to-one adversarial tasks demonstrate the effectiveness and applicability of the proposed design algorithm.

INDEX TERMS Unmanned systems, reinforcement learning, autonomous maneuvering decision-making, obstacle-avoidance.

I. INTRODUCTION

With the development of sensors, computers, and communication technology, the performance of unmanned vehicles have been significantly improved. Compared with manned vehicles, unmanned vehicles are used to complete more difficult and complex tasks in the military and civilian fields. Thus, research papers on unmanned vehicles are emerging in an endless stream, ushering in a spurt of innovation [1], [2], [3], [4]. In civilian applications, unmanned vehicles give more advantages in safety, economy, and applicability [5], [6], [7]. In addition, they are far superior to manned vehicles regardless of applicability or performance. Nevertheless, there are multiple practical hurdles to deploying unmanned vehicles in real-world robotics problems. For example, most unmanned vehicles are still controlled by the remote manual operation system. This control mode makes the applications of unmanned vehicles depend on the maneuvering decision ability of the remote operators, which is often not applicable to complex and fast-changing scenarios. Therefore, improving the maneuvering decision capabilities for complex tasks is still a key problem in the current unmanned vehicle systems, such as automatic polite of unmanned vehicles [8], [9], [10], [11], autonomous obstacle-avoidance [12], and the autonomous confrontation of unmanned fighters [13], etc.

For the design of autonomous maneuvering decision systems of unmanned vehicles, optimization principles and artificial intelligence (AI) algorithms are widely used in current research. Theoretically, the design methods of autonomous maneuvering decisions are divided into three categories: the game theory [14], [15], [16], [17], the optimization

The associate editor coordinating the review of this manuscript and approving it for publication was Hiram Ponce.

This work is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 License.
For more information, see https://creativecommons.org/licenses/by-nc-nd/4.0/

algorithm [18], [19], [20], [21], [22], [23], [24] and artificial intelligence methods [25], [26], [27], [28], [29], [30]. Among them, the methods based on the game theory address the confrontation tasks as a dynamical game and then make the optimal decision through the differential game [14], [15] or the influence diagram algorithm [17]. For confrontation tasks with a highly dynamic characteristic, it is often difficult to obtain the optimal real-time policy due to the complexity of computation [17]. For the methods based on the optimization algorithms, such as genetic algorithm [19], Bayesian inference [20], and statistical theory [24], they transform the maneuvering decision problem into an optimization problem and solve it mathematically to obtain an autonomous optimal policy. However, for large-scale complex non-convex optimization, it is also difficult to ensure the optimal solution of the solution. Furthermore, the above methods are mostly offline [19]. AI-based methods include expert systems [26], neural networks [27], and RL methods [28], [29], [30], [31]. The expert system models the maneuvering decision system as a rule inference system. On the one hand, it is hard to transform expert experiences into a rule inference system. On the other hand, the fixed expert experience usually difficult to guarantee the optimal decision for complicated dynamics. The neural network-based methods represent the decision as an artificial neural network (ANN); theoretically, the training algorithms will obtain the optimal decision. However, it is difficult to obtain effective training data in practical applications, and the resulting performance of the autonomous maneuvering decision is usually limited. Compared with the above methods, the RL algorithm requires no model or prior knowledge of the processes, but only through the interaction between the agent and the environment. The policy of the agent will be constantly optimized until the optimal or suboptimal policy is obtained [32]. In addition, the policy of RL is represented by a deep neural network that not only has the capability for nonlinear approximation but also has good generalization [33], which possibly leads to optimal decision-making and better robustness. Therefore, deep RL algorithms are a powerful and advantageous method for complex environments.

In the autonomous maneuvering decisions of unmanned vehicles, the autonomous navigation and confrontation decision-making of Unmanned Aerial Vehicle (UAV) have received extensive attention. Currently, most of the RL algorithms applied to UAVs are based on the deep Q-learning (DQN) algorithm [29], [33], [34], [35], [36]. By describing the maneuvering decision of the UAVs as a sequence of some simple fixed actions, the complexity of the design problem is reduced significantly. However, it leads to a significant difference from reality, and the confrontation performance is difficult to be guaranteed. Furthermore, the efficiency is very low in the process of learning the optimal policy for complex scenarios. In the practical system, the action and the state of the UAV are considered in the continuous time and high-dimensional space, which causes a dimensional disaster and sparse reward problem. Although the deep deterministic policy gradient (DDPG) [37], a kind of RL algorithm is used for policy optimization problems with continuous state and action, many hyperparameters need to be appropriately determined. Although Soft Actor-Critic (SAC) [38], [39] algorithm is a way to deal with simple scenarios with continuous action space, its low learning efficiency leads to low adaptability to difficult scenarios.

This paper proposes a Progressive RL algorithm to overcome low learning efficiency for the complex decision-making adversarial tasks of unmanned vehicle systems. It is a straightforward development inspired by the human learning process from simple to complex. To illustrate the specific training algorithm and demonstrate its feasibility and effectiveness, the one-to-one autonomous confrontation and obstacle-avoidance of two unmanned vehicles is considered as the practical scenario in this paper. Firstly, the confrontation environment is modeled, and the corresponding performance indexes are presented for confrontation and obstacle-avoidance. Then, a progressive RL algorithm is proposed based on the SAC framework with the reward function, state information, and progressive learning curricula designed properly. All the simulation results demonstrate that the proposed design algorithm not only gives a feasible policy for confrontation and obstacle-avoidance but also realize the iterative optimization by the training course of the self-play. Furthermore, compared with the conventional reinforcement learning algorithm, the proposed design algorithm has superior learning efficiency and better performance in autonomous confrontation and obstacle-avoidance. The experimental results demonstrate that the proposed algorithm is feasible for the autonomous maneuvering decision system.

A. CONTRIBUTIONS

In summary, compared with previous studies in confrontation and obstacle-avoidance of unmanned vehicles, our contributions are two-fold.

- We propose a progressive RL algorithm based on the soft actor-critic (SAC) RL framework to improve learning efficiency for the complex decision-making adversarial tasks of unmanned vehicle systems, especially imitating the process of human learning, and designed a learning curricula with increasing difficulty in the algorithm.
- All the simulation results show that our algorithm not only has higher learning efficiency but also provides feasible strategies for confrontation and obstacle-avoidance.

B. PAPER ORGANIZATION

The rest of the paper is organized as follows: In Section II, we present the environments with the one-to-one confrontation and obstacle-avoidance of the unmanned vehicles, including the dynamic description of the vehicle and the performance indexes of confrontation and obstacle-avoidance.

In Section III, the SAC is briefly introduced, then based on SAC algorithm, the progressive training algorithm for autonomous confrontation and obstacle-avoidance policy is developed and discussed; In Section IV, the numerical simulation of the proposed algorithm is conducted to the one-to-one autonomous confrontation and obstacle-avoidance, and the simulation results are discussed. Conclusions are drawn in Section V

II. MODELING OF VEHICLE CONFRONTATION AND OBSTACLE-AVOIDANCE

In this section, the one-to-one autonomous confrontation and obstacle-avoidance of unmanned vehicles on a two-dimensional plane are considered practical scenarios. The proposed design method is directly extended to more complex scenarios, such as the autonomous driving of ships, the air combat of UAVs, etc.

A. DYNAMICS OF THE UNMANNED VEHICLE SYSTEM

For the simplicity of description, the unmanned vehicle is considered a two-wheeled vehicle with self-balancing ability. Therefore, the structure of the vehicle is similar to a bicycle, and its motion space is a two-dimensional plane. Figure 1 shows the structure schematic of the unmanned vehicle.

As shown in Figure 1, the dynamics of the vehicle is described by the following differential equation:

$$\dot{x} = v\cos(\varphi + \beta) \tag{1}$$

$$\dot{y} = v\sin(\varphi + \beta) \tag{2}$$

$$\dot{\varphi} = v\frac{\sin\beta}{l_r} \tag{3}$$

$$\dot{v} = a \tag{4}$$

$$\beta = \tan^{-1}\left(\frac{l_r}{l_r + l_f}\tan(\delta)\right) \tag{5}$$

where (x, y) indicates the position of the vehicle; v denotes the velocity scale of the vehicle; $\dot{x}$ and $\dot{y}$ are the velocity scales on the ox-axis and oy-axis, respectively; φ is the angle between the body direction and the ox-axis; l_r represents the distance between the rear of the vehicle and the steering center; l_f represents the distance between the head of the vehicle and the steering center, and β represents the angle between the direction of the steering center and the body.

Assumption 1: Assume that the control angle of the front wheel of the vehicle relative to the direction of the body is δ, and the driving force is described by the acceleration a. In the above model, a and δ are the manipulated variables that control the motion of the vehicle.

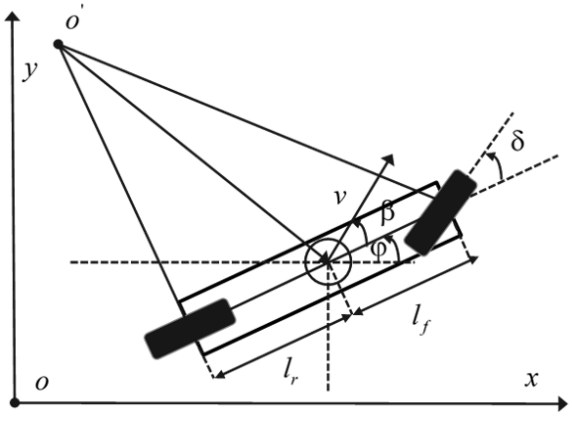


FIGURE 1. Structure schematic of the two-wheeled vehicle.

B. MODELING AND EVALUATION OF ONE-TO-ONE CONFRONTATION AND OBSTACLE-AVOIDANCE

The scenario considered in this paper is the autonomous maneuvering decision system with two vehicles in a two-dimensional environment with obstacles [40]. It is assumed that vehicle A represents the tracking vehicle, and vehicle B is the tracked vehicle.

Design Objective: Based on the RL algorithm, the optimal policy of vehicle A is learned to avoid dangerous collisions with obstacles and maintain the best tracking advantage as much as possible during the confrontation.

To evaluate the performance of the autonomous maneuvering decision system, the confrontation advantage and obstacle-avoidance performance are defined and designed, respectively.

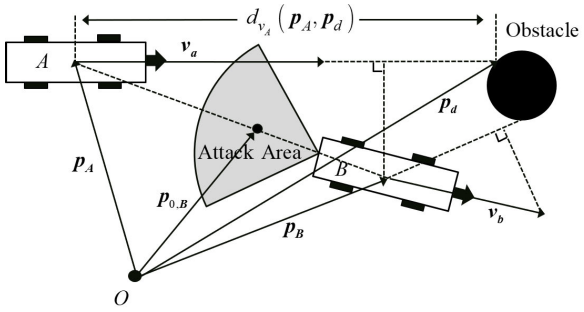


FIGURE 2. Modeling schematic of one-to-one confrontation and obstacle-avoidance.

Figure 2 shows the position status of two vehicles and the nearest obstacles at any time, where $\mathbf{p}_A = (x_A, y_A)$ represents the spatial position of vehicle A; $\mathbf{p}_B = (x_B, y_B)$ represents the spatial position of vehicle B; $\mathbf{v}_A$ and $\mathbf{v}_B$ represent the velocity vectors of vehicle A and vehicle B respectively. $\mathbf{p}_d$ represents the position of the nearest obstacle in the movement direction of vehicle A, and $d_{\mathbf{v}_A}(\mathbf{p}_A, \mathbf{p}_d)$ represents the distance between the vehicle A and the nearest obstacle in the movement direction. When there is no obstacle in the movement direction, the distance is assumed infinite. The gray fan-shaped area behind vehicle B represents the effective confrontation (attack) area of vehicle A. This area is moving with the movement of vehicle B. $\mathbf{p}_{0,B}$ denotes the center point of the confrontation area, indicating the best confrontation (attack) position of vehicle A.

1) ATTACK ADVANTAGE INDEX

Figure 2 shows that vehicle A needs to keep itself in the effective confrontation (attack) area and close to the best confrontation position as possible to maintain the dominance in confrontation. In addition, vehicle A must also have an attack advantage in the direction of movement. Considering these two factors, the attack advantage index of vehicle A at any time t is defined by:

$$I_{1,A} = \frac{w_1 \cos\langle \mathbf{v}_A, \mathbf{v}_B\rangle + w_2 \cos\langle \mathbf{v}_A, \mathbf{p}_B - \mathbf{p}_A\rangle}{\left|\mathbf{p}_A - \mathbf{p}_{0,B}\right| + d_0} \tag{6}$$

where $\langle\cdot,\cdot\rangle$ is the vector angle operator; $|\cdot|$ indicates the norm of vector; $\langle \mathbf{v}_A, \mathbf{v}_B\rangle$ describes the angle between the velocities of vehicle A and vehicle B; $\langle \mathbf{v}_A, \mathbf{p}_B - \mathbf{p}_A\rangle$ describes the angle between the moving direction of vehicle A and the position direction between vehicle B and vehicle A, which is the attack angle of vehicle A. If α is defined as the maximum angle within which vehicle A can attack possibly, then vehicle A has an attack advantage when $\langle \mathbf{v}_A, \mathbf{p}_B - \mathbf{p}_A\rangle \leq \alpha \cdot d_0$ is a positive constant to ensure the denominator higher than zero, w_1 and w_2 are the weighting factors determined by the importance of the two angles.

Definition (6) shows that the attack advantage of vehicle A is determined by three aspects:

- $\left|\mathbf{p}_A - \mathbf{p}_{0,B}\right|$. The distance between vehicle A and the best confrontation (attack) position. The closer the distance means the higher attack advantage.
- $\cos\langle \mathbf{v}_A, \mathbf{v}_B\rangle$. The consistency between the velocity directions of vehicle A and vehicle B, The higher consistency results the higher advantage.
- $\langle \mathbf{v}_A, \mathbf{p}_B - \mathbf{p}_A\rangle$. The attack angle of vehicle A. The lower the attack angle indicates the higher attack advantage.

2) PERFORMANCE INDEX OF OBSTACLE-AVOIDANCE

To be suitable for more complex confrontation environment, the obstacles is also considered in the model. It is assumed that the vehicle is equipped with a front lidar, which detects the distance of the nearest obstacle in the movement direction. It means that the real-time information $d_{\mathbf{v}_A}(\mathbf{p}_A, \mathbf{p}_d)$ is obtained. Based on this assumption, the performance index of obstacle-avoidance of vehicle A at time t is defined by:

$$I_{2,A} = \frac{-|\mathbf{v}_A|}{d_{\mathbf{v}_A}(\mathbf{p}_A, \mathbf{p}_d) + d_0} \tag{7}$$

where d_0 is a positive constant to ensure the denominator is higher than zero, and the other symbols are the same as attack advantage index (6).

Considering the attack advantage index and the performance index of obstacle-avoidance comprehensively, the overall advantage index at time t of vehicle A is defined by:

$$I_{T,A,t} = k_1 I_{1,A,t} + k_2 I_{2,A,t} \tag{8}$$

where k_1 and k_2 are the weighting factors.

III. PROGRESSIVE RL ALGORITHM

A. RL FRAMEWORK AND SOFT ACTOR-CRITIC (SAC) ALGORITHM

RL is mainly used to solve the optimal decision problem of the Markov decision process (MDP). For complex and unmodeled decision processes, reinforcement learning generates an optimal control policy by interacting with the environment. As shown in Figure 3, the basic composition of the framework includes two components: the environment and the agent [32]. Mathematically, the MDP is described by a 4 -tuple $(\mathcal{S}, \mathcal{A}, R, \gamma)$, where $\mathcal{S}$ indicates the state space of the environment; $\mathcal{A}$ is the action space; R is the reward; γ is the reward discount factor.

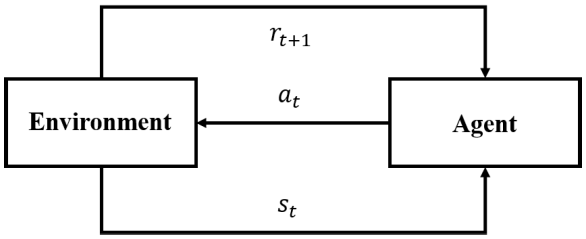


FIGURE 3. RL framework.

Under this framework, the agent is responsible for the real-time interaction with the environment. It gives a control action a_t based on the real-time state feedback information s_t from the environment. At the same time, the agent also receives the reward information r_t to optimize the policy.

In summary, the RL system includes the following five elements [32]:

(1) State s_t : the system information feedback from the environment to the agent.
(2) Action $a_t = \pi(s_t)$: the control policy $\pi(s_t)$ indicates the mapping function from s_t to a_t, which is determined by the agent conditions on the state information.
(3) State transition probability $p(s_{t+1} \mid s_t, a_t)$: the response of the environment to the action, describing the dynamic characteristics of the environment.
(4) Reward $r(s_t, a_t)$: the instant reward provided by the environment according to the state and action.
(5) State value function $V(s_t)$ and state-action value function $Q(s_t, a_t)$: the cumulative discounted reward defined by

$$Q(s_t, a_t) = \mathbf{E}_\pi\left(\sum_{i=0}^{\infty} \gamma^i r_{t+i} \mid s_t = s, a_t = a\right) \tag{9}$$

which is also the optimization objective of the RL.

If the optimal control policy at time t is $\pi_*(s_t)$, and the optimal state-action value function is $Q_*(s_t, a_t)$, then, according to the optimal principle, we have the optimal Bellman equation:

$$Q_*(s_t, a_t) = \mathbf{E}_{\pi_*}(r_t + \gamma Q_*(s_{t+1}, a_{t+1}) \mid s_t = s, a_t = a) \tag{10}$$

The objective of RL is to find an optimal policy $\pi_*(s, a)$ maximizing the state-action value function $Q(s_t, a_t)$. If $Q_*(s, a)$ is the solution of the optimal Bellman equation (10) for any s and a, then the optimal control policy is

$$\pi_*(s, a) = \arg\max_{a \in A(s)} Q_*(s, a) \tag{11}$$

The entropy of policy reflect the diversity of the policy or action. SAC just adds policy entropy to the optimization objective. Therefore, in order to explore the environment more efficiently, we use the SAC algorithm as the design framework. The SAC is another RL algorithm based on the Actor-Critic framework proposed by the team of Pieter Abbeel and Sergey Levine [39]. Different from the DDPG algorithm, the optimal policy of the SAC is defined as follows:

$$\pi_* = \arg\max_{\pi} \mathbf{E}_{(s_t, a_t) \sim \rho_\pi} \left(\sum_t r(s_t, a_t) + \alpha \mathcal{H}\left(\pi(\cdot \mid s_t)\right) \right) \tag{12}$$

where $\mathcal{H}(\pi(\cdot \mid s_t))$ indicates the information entropy of policy $\pi(\cdot \mid s_t)$; α is the temperature factor weighting entropy $\mathcal{H}(\pi(\cdot \mid s_t))$.

Similarly, in the SAC algorithm, the optimization is based on the following Bellman residual value:

$$J_Q(\theta) = \mathbf{E}_{(s_t, a_t) \sim \mathcal{D}} \left(\frac{1}{2} \left(Q_\theta(s_t, a_t) - \left(r(s_t, a_t) + \gamma \left(Q_{\bar{\theta}}(s_{t+1}, a_{t+1}) \right) \right) \right)^2 \right) \tag{13}$$

where $\theta, \bar{\theta}$ presents the parameters of the online value network and the target value network, respectively. The state value function $V(s_t)$ is defined as follows:

$$V(s_t) = \mathbf{E}_{a_t \sim \pi} \left(Q(s_t, a_t) - \alpha \log\left(\pi(a_t \mid s_t)\right) \right) \tag{14}$$

According to expression (13), the gradient of the residual value is calculated by:

$$\hat{\nabla} J_Q(\theta) = \nabla_\theta Q_\theta(s_t, a_t) \left(Q_\theta(s_t, a_t) - \left(r(s_t, a_t) + \gamma \left(Q_{\bar{\theta}}(s_{t+1}, a_{t+1}) - \alpha \log\left(\pi_\phi(a_{t+1} \mid s_{t+1})\right) \right) \right) \right) \tag{15}$$

where the $\hat{\nabla}$ denotes the approximate gradient operator.

The loss function of the policy network in the SAC is defined as follows:

$$J_\pi(\phi) = \mathbf{E}_{s_t \sim \mathcal{D}} \left(\mathbf{E}_{a_t \sim \pi_\phi} \left(\alpha \log\left(\pi_\phi(a_t \mid s_t)\right) - Q(s_t, a_t) \right) \right) \tag{16}$$

where ϕ denotes the parameters of the policy network. In order to use the backpropagation algorithm to train the neural network, the action a_t needs to be reparametrized, which is sampled from some fixed distribution function, such as spherical Gaussian distribution, described by:

$$a_t = f_\phi(\varepsilon_t; s_t) \tag{17}$$

where ε_t is an input noise vector. Substituting the above formula into the loss function (16), we have:

$$J_\pi(\phi) = \mathbf{E}_{s_t \sim \mathcal{D}, \varepsilon_t \sim \mathcal{N}} \left(\alpha \log\left(\pi_\phi\left(f_\phi(\varepsilon_t; s_t) \mid s_t\right)\right) - Q\left(s_t, f_\phi(\varepsilon_t; s_t)\right) \right) \tag{18}$$

Then the gradient of the loss function is determined as follows:

$$\hat{\nabla}_\phi J_\pi(\phi) = \nabla_\phi \alpha \log\left(\pi_\phi(a_t \mid s_t)\right) + \left(\nabla_{a_t} \alpha \log\left(\pi_\phi(a_t \mid s_t)\right) - \nabla_{a_t} Q(s_t, a_t) \right) \nabla_\phi f_\phi(\varepsilon_t; s_t) \tag{19}$$

In the SAC algorithm developed in 2019 [39], the temperature coefficient α also needs to be updated automatically by defining the loss function as follows:

$$J(\alpha) = \mathbf{E}_{a_t \sim \pi_t} \left(-\alpha \log\left(\pi_\phi(a_t \mid s_t)\right) - \alpha \overline{\mathcal{H}} \right) \tag{20}$$

The loss function's gradient of the temperature coefficient is given as follows:

$$\hat{\nabla}_\alpha J(\alpha) = \nabla_\alpha \mathbf{E}_{a_t \sim \pi_t} \left(-\alpha \log\left(\pi_\phi(a_t \mid s_t)\right) - \alpha \overline{\mathcal{H}} \right) \tag{21}$$

The above shows that the SAC algorithm maximizes the cumulative reward and the policy entropy simultaneously. It is precisely by maximizing the entropy of the policy to ensure the exploration ability of the algorithm, so that it is not easy to fall into the local optimal. In additions, the temperature coefficient α is automatically updated during the training process. By setting the large α in the early stage of training to ensure that the agent has good exploration.

According to the unmanned vehicle system modeled in Section II, Figure 4 illustrates the proposed RL framework of the autonomous confrontation and obstacle-avoidance maneuvering decision based on the SAC algorithm. In this framework, the state of two vehicles and the state information of the nearest obstacle to vehicle A are defined as state information s_t of the environment. Action a_t of the agent and the attack advantage index of vehicle A is determined according to state s_t. After the action is executed, and the state of the environment is updated, an instant reward r_t is calculated and feedback to the agent at time $t+1$. The interaction data $D = (s_t, a_t, r_t, s_{t+1})$ is obtained and stored into the experience replay buffer. According to the SAC algorithm, the interaction data in the replay buffer will be sampled to update the critic network and actor network, so that the policy of the agent is gradually optimized until the satisfied autonomous maneuvering decision policy for confrontation and obstacle-avoidance is realized.

B. DESIGN OF THE STATE AND ACTION SPACES

According to the framework of RL shown in Figure 3, at each time t, a set of observable information is defined as the state information, and at the same time, it will be used to evaluate the advantage of the current situation. For the one-to-one confrontation system, the state information is defined as follows:

$$\left[x_A, y_A, \varphi_A, v_A, x_B, y_B, \varphi_B, v_B, d_{v_A}(\mathbf{p}_A, \mathbf{p}_d) \right] \tag{22}$$

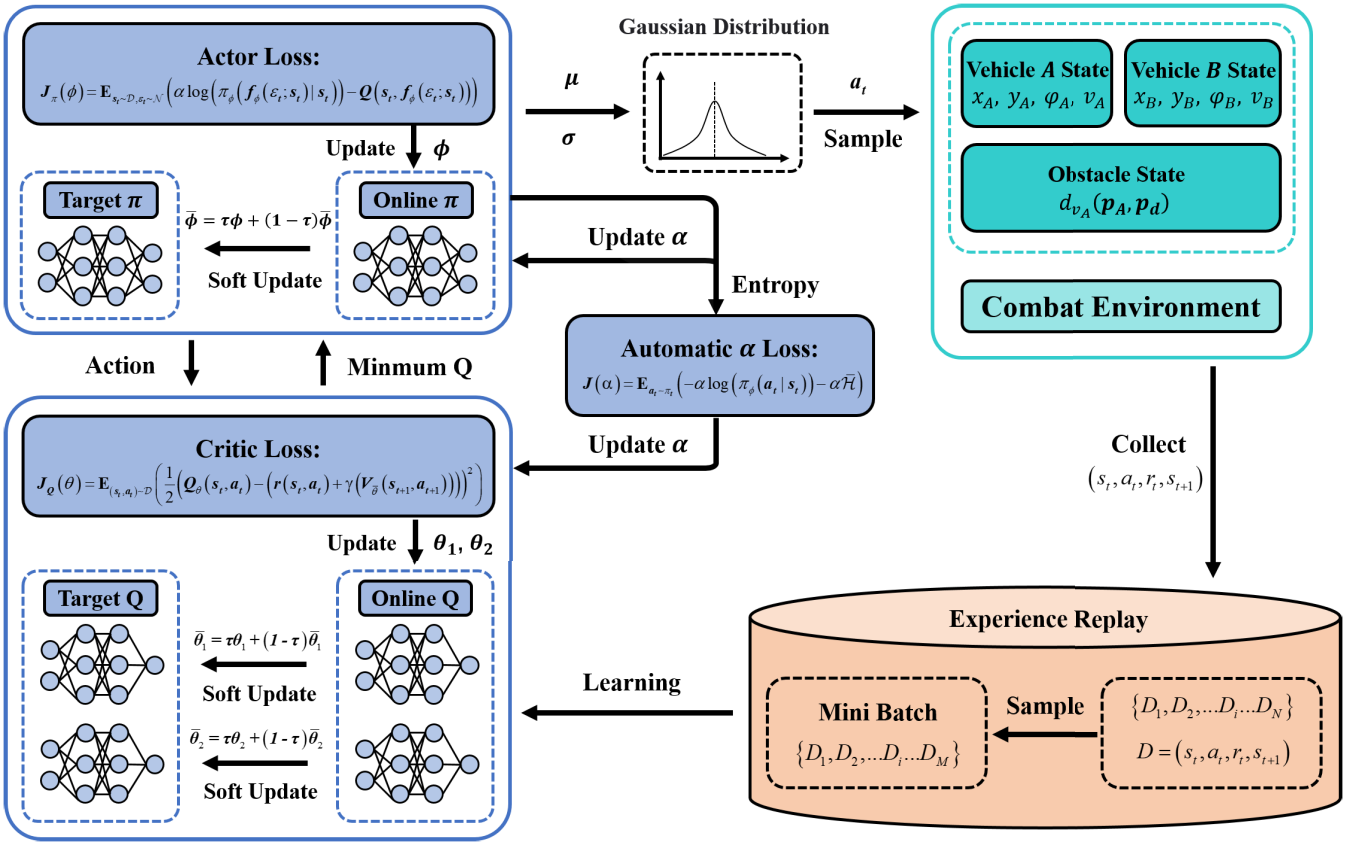


FIGURE 4. Design scheme for the autonomous maneuvering decision of the unmanned vehicle based on the SAC algorithm.

and the action is defined by:

$$[a_A, \delta_A] \tag{23}$$

where a_A represents the acceleration of vehicle A, corresponding to the drive force of the vehicle, and δ_A represents the steering angle of vehicle A, corresponding to the direction of the vehicle.

C. DESIGN OF THE REWARD

In this paper, the instant reward is defined based on the attack advantage index (8). During the confrontation, when the distance between the two vehicles exceeds a certain range, a sparse reward will lead to a large number of invalid samples, which will lead to the low efficiency of learning. To solve this problem, the following penalty is used when the distance d between the two vehicles is great than a certain value:

$$\mathrm{P}_{A,t} = \begin{cases} -w_4, & d > 0.5\sqrt{(x_{\max} - x_{\min})^2 + (y_{\max} - y_{\min})^2} \\ 0, & d \le 0.5\sqrt{(x_{\max} - x_{\min})^2 + (y_{\max} - y_{\min})^2} \end{cases} \tag{24}$$

where $w_4 > 0$, $x_{\min}$ and $y_{\min}$ respectively represent the minimum boundaries along the x axis and y axis, and $x_{\max}$ and $y_{\max}$ respectively represent the maximum boundaries along the x axis and y axis.

According to the overall advantage index (8) and the penalty (24), the instant reward at time t is defined by:

$$r_{A,t} = I_{T,A,t} + k_3 \mathrm{P}_{A,t} \tag{25}$$

where k_3 is the weighting factor of the penalty.

Based on the overall advantage index function, the autonomous maneuvering decision problem of vehicle A is described as the following optimization problem:

$$\max_{a,\delta} \sum_{t=1}^{M} \gamma^{t-1} r_{A,t} \quad \text{s.t.} \begin{cases} \text{Equation (1)} - (8) \\ \text{Equation (24)} - (25) \\ a_{\min} \le a \le a_{\max} \\ \delta_{\min} \le \delta \le \delta_{\max} \end{cases} \tag{26}$$

where M is the number of optimization steps; $a_{\min}$ and $a_{\max}$ represent the permitted minimum and maximum values of the vehicle acceleration A; $\delta_{\min}$ and $\delta_{\max}$ represent the permitted minimum and maximum wheel angles of the vehicle.

Problem (26) is essentially a complex nonlinear optimization problem that is difficult to solve. Based on the RL framework, the advantage index of the vehicle is adopted as the instant reward, and then the SAC algorithm is employed to solve the optimization problem (26).

D. PROGRESSIVE SAC ALGORITHM

While the RL algorithm presented in section III-A is directly used to solve the autonomous confrontation and obstacle-avoidance problems, there are two issues at the initial training stage:

(1) Since vehicle A has not any tracking policy at the initial stage of the training, it is easy to have a bad tracking performance, resulting in:

$$\left|\mathbf{p}_A - \mathbf{p}_{0,B}\right| \gg (w_1 \cos\langle \mathbf{v}_A, \mathbf{v}_B\rangle + w_2 \cos\langle \mathbf{v}_A, \mathbf{p}_B - \mathbf{p}_A\rangle) \quad (27)$$

(2) Since vehicle A has not any tracking policy at the initial stage of the training, it is easy to have a bad tracking performance, resulting in:

$$d_{\mathbf{v}_A}(\mathbf{p}_A, \mathbf{p}_d) \approx 0 \quad (28)$$

At this time, (7) shows that

$$\lim_{d_{\mathbf{v}_A}(\mathbf{p}_A, \mathbf{p}_d)\to 0} I_{2,A,t} = -\infty \quad (29)$$

Therefore, (6) shows that $I_{1,A} \approx 0$ and (7) shows that a large number of negative rewards interrupted the training process, which cause the sparse reward problem. To solve the above problems, we propose a progressive RL algorithm by imitating the learning of humans from simple to complex. The detailed schematic of the RL scheme is shown in Figure 5. In this learning scheme, a set of curricula with different tasks of confrontation and obstacle-avoidance are designed, and the learning procedure is divided into the following four stages:

- **Curriculum I:** Simple confrontation learning stage. In this curriculum, vehicle B is designed to move along some simple trajectories at a uniform or varying velocity, such as a straight or a circle way shown in Figure 5. At the same time, a few of obstacles are configured in the environment. In addition, both vehicle A and B are assumed to start randomly from the initial area. The learning task of the agent in this curriculum is to obtain basic tracking performance in this simple confrontation scenario.
- **Curriculum II:** Random confrontation learning stage. In this curriculum, it is assumed that vehicle B is controlled by a random maneuvering policy π_{random}. Compared with Curriculum I, there are more fixed obstacles and larger random initial areas of the two vehicles configured in the environment. The main learning target for the agent of vehicle A is to improve the autonomous confrontation policy to ensure the tracking performance when vehicle B moves randomly.
- **Curriculum III:** obstacle-avoidance learning stage. In this curriculum, it is assumed that vehicle B is driven by the policy of vehicle A obtained in the last curriculum. Moreover, more obstacles are continuously added to the environment with the geometry and distribution randomly changed during the training. The main training target in this curriculum is to improve the obstacle-avoidance performance of the vehicle A based on the confrontation policy obtained in Curriculum II.
- **Curriculum IV-VI:** Iterative self-play stage. After vehicle A gets a good confrontation and obstacle-avoidance performance, transfer the policy of vehicle A to vehicle B, and iteratively improve the policy of vehicle A through the confrontation between the two vehicles. At the same time, in this curriculum, the quantity and the distributions of the obstacles are continuously changed. The training purpose of these curricula is to optimize the maneuvering policy of vehicle A through the self-play until a satisfactory autonomous confrontation performance is obtained.

The difficulties of the curricula are gradually increased in terms of training complexity. It is seen that from the simulation results given in the next section that through the proper design for the difficulties of the learning stages, the proposed algorithm not only improve the learning efficiency but also ensure the iterative optimization of the confrontation and obstacle-avoidance performance.

E. POLICY OPTIMIZATION BASED ON THE PROGRESSIVE SAC ALGORITHM

The specific pseudo code of the specific RL algorithm is described in **algorithm** 1:

For the algorithm 1, we give some remarks as follows:

(1) The design algorithm is essentially a data-driven optimization algorithm without any requirement for the prior knowledge of the environment. It can be extended and applied to more complicated confrontation and obstacle-avoidance problems.
(2) Through appropriate environment configuration of the curricula, the problem of the sparse reward is avoided, and then the learning efficiency of the algorithm is improved.
(3) In the final curriculum (Curriculum IV), the self-play is adopted to realize the iterative optimization of the confrontation and obstacle-avoidance policy.
(4) The multi-task of confrontation and obstacle-avoidance is considered in the curricula, resulting in the solution for the complex maneuvering decision problems.

IV. SIMULATIONS AND DISCUSSIONS

In this section, to demonstrate the feasibility and effectiveness of the proposed design algorithm, the one-to-one autonomous confrontation and obstacle-avoidance of two unmanned cars are considered and simulated.

A. DESIGN OF THE TRAINING CURRICULA

As given in Table 1, not only the policy complexity of the car B gradually increseas but also the specified ranges for the quantity and the size of obstacles extended. After curriculum V, the policy obtained by Car A in the previous

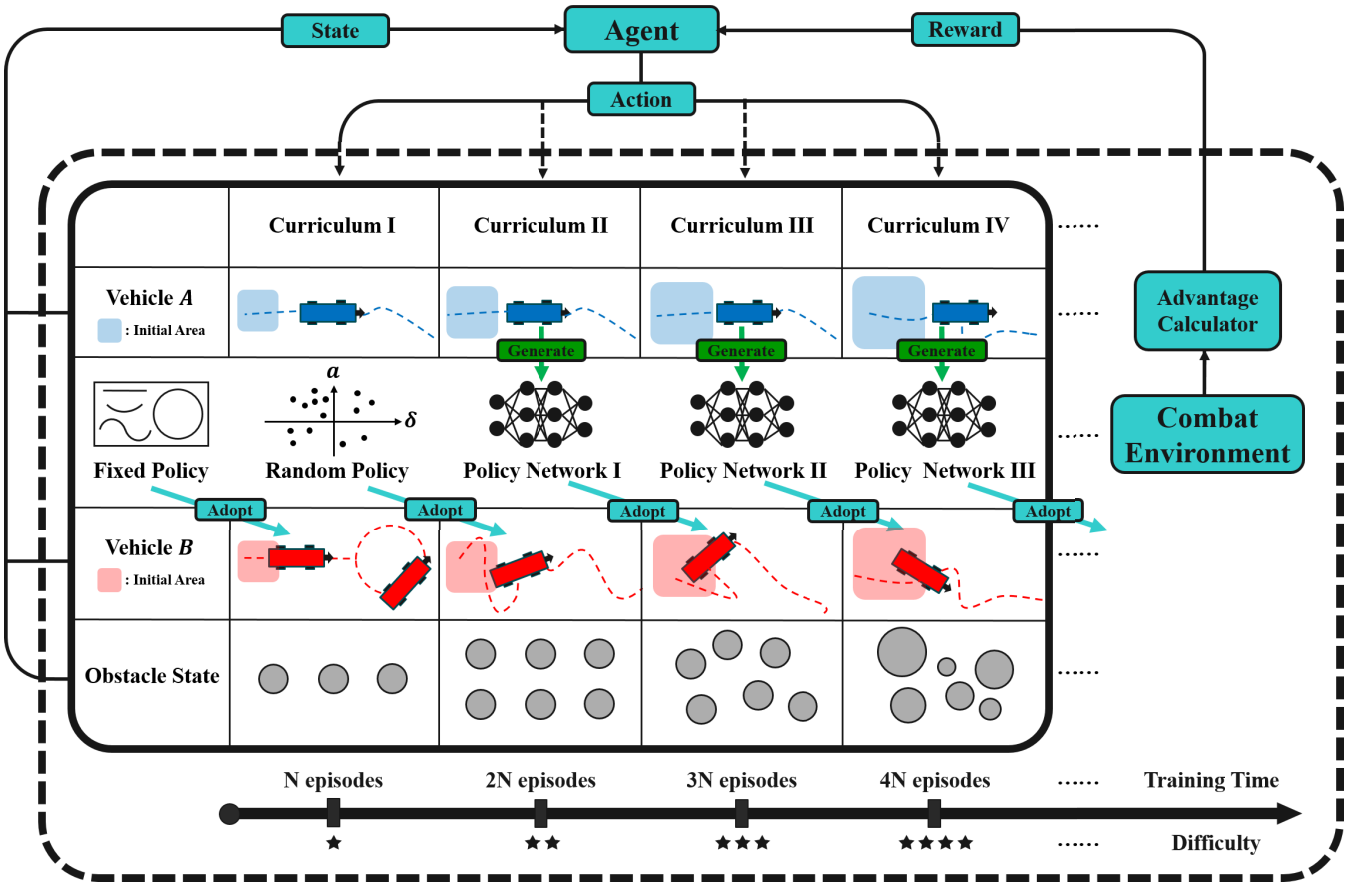


FIGURE 5. Schematic of autonomous maneuvering decision learning framework for unmanned vehicles based on progressive training curricula.

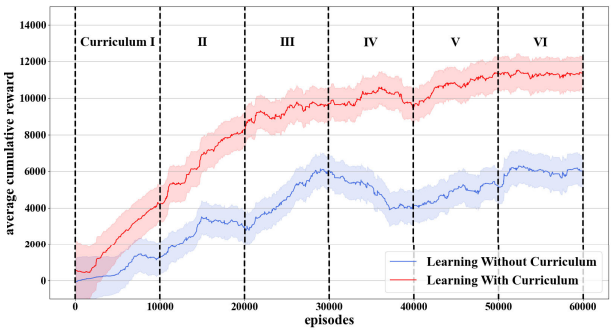


FIGURE 6. Learning curves of the progressive (red line) and non-progressive (blue line) RL algorithms.

curriculum is transferred to Car B, forming the iterative optimization (self-play) of the policy.

B. PERFORMANCE INDEXES

In the training algorithm, it is assumed that there are M maneuvering decision steps in each episode. To evaluate the confrontation and obstacle-avoidance performances of the obtained policy during the training, it is necessary to calculate the advantage time ratios of Car A and Car B after each training episode, respectively. If we define the difference of

TABLE 1. The hyperparameters of the training curricula.

Curriculum	Car B's policy	Obstacle quantity	Obstacle size (m)
I	Simple policy (straight line and circle)	[1,3]	[1,4]
II	Random movement policy	[2,5]	[2,5]
III	Random movement policy	[3,10]	[3,6]
IV	Random movement policy	[8,15]	[4,8]
V	Policy from Car A obtained in Curriculum IV	[10,17]	[4,9]
VI	Policy from Car A obtained in Curriculum V	[12,15]	[6,20]
⋮	⋮	⋮	⋮

the attack advantage index between Car A and Car B in each maneuvering decision step t as:

$$\Delta I_{1,t} = I_{1,A,t} - I_{1,B,t} \tag{30}$$

where $\Delta I_{1,t} > 0$ indicates that Car A has confrontation advantage over Car B, and the larger value of $\Delta I_{1,t}$ means the higher advantage. By counting the step numbers, m_k, $(k = 1, 2, 3)$, for $\Delta I_{1,t} > 0$, $\Delta I_{1,t} = 0$, and $\Delta I_{1,t} < 0$ in each episode, respectively, the time ratios of the advantage for Car A in each episode is computed by:

$$p_k = \frac{m_k}{M}, (k = 1, 2, 3) \tag{31}$$

Algorithm 1 Progressive SAC Algorithm

Initialize the value network parameters θ_1, θ_2 and policy network parameters ϕ
Initialize target value network parameters $\bar{\theta}_1 = \theta_1, \bar{\theta}_2 = \theta_2$, and target policy network parameters $\bar{\phi} = \phi$
Initialize the experience replay buffer R
Loop for each curriculum j (**Curriculum I** $\rightarrow$ **Curriculum II** $\rightarrow \cdots$):
  According to the configurations of curriculum j, initialize the motion policy of vehicle B
  According to the configurations of curriculum j, initialize the obstacle setting in the environment
  Initialize target entropy according to current policy $\overline{\mathcal{H}}_j$
  Loop for each episode:
    Get the initial state s_0 of the environment
    Loop for each step t:
      For state s_t, choose an action: $a_t \sim \pi_\phi\left(a_t \mid s_t\right)$
      vehicle A executes action a_t and vehicle B executes the given motion policy according to the configurations of the curriculum.
      The environment moves to the next state s_{t+1}, and vehicle A gets instant reward $r\left(s_t, a_t\right)$
      Store the experience data $\left(s_t, a_t, r\left(s_t, a_t\right), s_{t+1}\right)$ in the replay buffer R.
      Randomly collect m samples from the experience replay buffer R to update the networks as follows:
        Update the value network, according to the gradient (15):

$$\theta_i = \theta_i - \lambda_Q \hat{\nabla}_{\theta_i} J_Q\left(\theta_i\right), \quad i \in \{1, 2\}$$

        Update the policy network, according to the gradient (19):

$$\phi = \phi - \lambda_\pi \hat{\nabla}_\phi J_\pi(\phi)$$

        Update the temperature coefficient, α, according to the gradient (21):

$$\alpha = \alpha - \lambda \hat{\nabla}_\alpha J(\alpha)$$

        Update the target value network: $\bar{\theta}_i = \tau\theta_i + (1-\tau)\bar{\theta}_i, \quad i \in \{1, 2\}$
        Update the target policy network: $\bar{\phi} = \tau\phi + (1-\tau)\bar{\phi}$.
    end loop
  end loop
end loop

Furthermore, to evaluate the confrontation performance more accurately, the cumulative advantage of Car A in each episode is calculated by:

$$\sum_{t=1}^{M}\left(I_{1,A,t} - I_{1,B,t}\right) \tag{32}$$

Note the episode numbers of $\sum_{t=1}^{M}\left(I_{1,A,t} - I_{1,B,t}\right) > 0$, $\sum_{i=1}^{M}\left(I_{1,A,t} - I_{1,B,t}\right) = 0$ and $\sum_{t=1}^{M}\left(I_{1,A,t} - I_{1,B,t}\right) < 0$ are n_1, n_2 and n_3, respectively, then the episode ratios of the confrontation advantage is defined by:

$$P_k = \frac{n_k}{N}, (k = 1, 2, 3) \tag{33}$$

where $N = n_1 + n_2 + n_3$ is the total episode for validation. The larger value P_1 means that, statistically, the Car A has more time in the confrontation advantage. Both time ratio p_1 and episode ratio P_1 of the advantage will be used as the performance indexes of the confrontation policy.

C. LEARNING CURVES

In order to evaluate the learning efficiency of the progressive RL algorithm, the classical SAC algorithm is also designed and conducted as the baseline method. Figure 6 shows the cumulative rewards of the non-progressive learning algorithm (blue line) and the progressive learning algorithm (red line) with 6 curricula given in Table 1. Due to the relatively simple curriculum in the early stage of the training (Curriculum I and Curriculum II), the learning efficiency of the progressive RL is significantly higher than that of the non-progressive learning algorithm. After 20,000 training episodes, with the difficulty of the curriculum increased, the learning efficiency gradually decreases, but the cumulative reward of the progressive RL algorithm (red line) is always better than that of the non-progressive RL algorithm (blue line). Therefore, the proposed progressive RL algorithm in this paper not only has higher learning efficiency but also guarantee a more optimal policy.

D. SIMULATION RESULTS

To demonstrate the autonomous maneuvering decision performances of the proposed progressive RL scheme, the policies obtained in different curricula are implemented in the one-to-one confrontation, and the simulation results indicate the improvement of the policy.

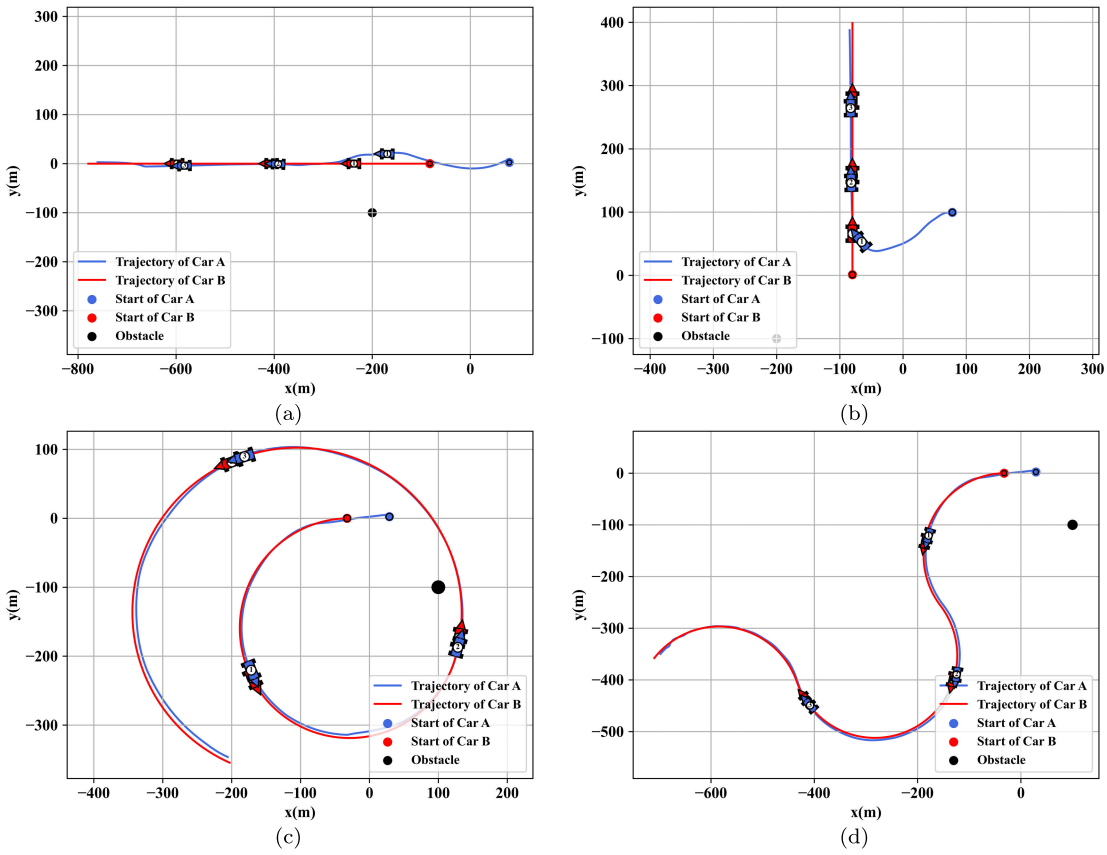


FIGURE 7. The tracking performances of Car A when Car B moves with a fixed velocity (a)(d) and varying velocity (b)(c) along a straight and circle way.

1) PERFORMANCES OF THE MANEUVERING DECISION POLICY AFTER CURRICULUM I AND II

In the curriculum configurations given in Table 1, Car B adopts a simple policy to train the basic tracking policy of Car A. Figure 7 shows the tracking performance of Car A based on the maneuvering decision policy obtained in the first curriculum when Car B moves with a uniform (Figure 7(a)) and variable (Figure 7(b)) velocity alone a straight line. In the following figures, the two cars are plotted and numbered to indicate the instant positions sampled during the confrontation. The same number denotes the same instant time. Figure 7 shows that after the curriculum I, Car A already obtains a good autonomous maneuvering decision policy for the simple tracking tasks. Figure 7(c) and Figure 7(d) show the tracking performance of Car A when Car B performs circle-like movements. It is also shown that Car A has a good autonomous maneuvering decision ability to track a circle-like movement. These results demonstrate that the training objectives of the curriculum I have been achieved.

For the curriculum II given in Table 1, the goal of training is to acquire the autonomous decision policy to track the random motion of Car B. The policy obtained after 20,000 episode training is applied to the confrontation. Figure 8(a) and (b) show the autonomous tracking performances of Car A starting from a random point when Car B performs the random movements. These results show that for any starting point, Car A always performs a good tracking to Car B.

TABLE 2. The time ratios for confrontation scenario shown in Figure 8(b) and episode ratios of confrontation advantage in total 100 episode confrontation after curriculum I and II.

Car	Time ratio (%)	Episode ratio (%) (total 100 episode)
A	96.50	100.00
B	3.50	0.00

For the one-to-one confrontation scenario shown in Figure 8(b), Figure 8(c) show the distance curve between Car A and Car B, where the pink area indicates the superior attack distance of Car A. Figure 8(d) gives the attack angles of Car A and Car B, where the pink-shaded area indicates the superior attack angle of Car A. It is easy to see from these figures that Car A has much more attack advantage than Car B. Moreover, we calculated the time ratios of the attack advantage of the two cars in this episode and the episode ratios of the confrontation advantage of the two cars in 100 episodes. The results are given in Table 2. These results show that Car A has an absolute advantage in the confrontation. All of the simulation results illustrate that the training objectives of curriculum II are fully achieved.

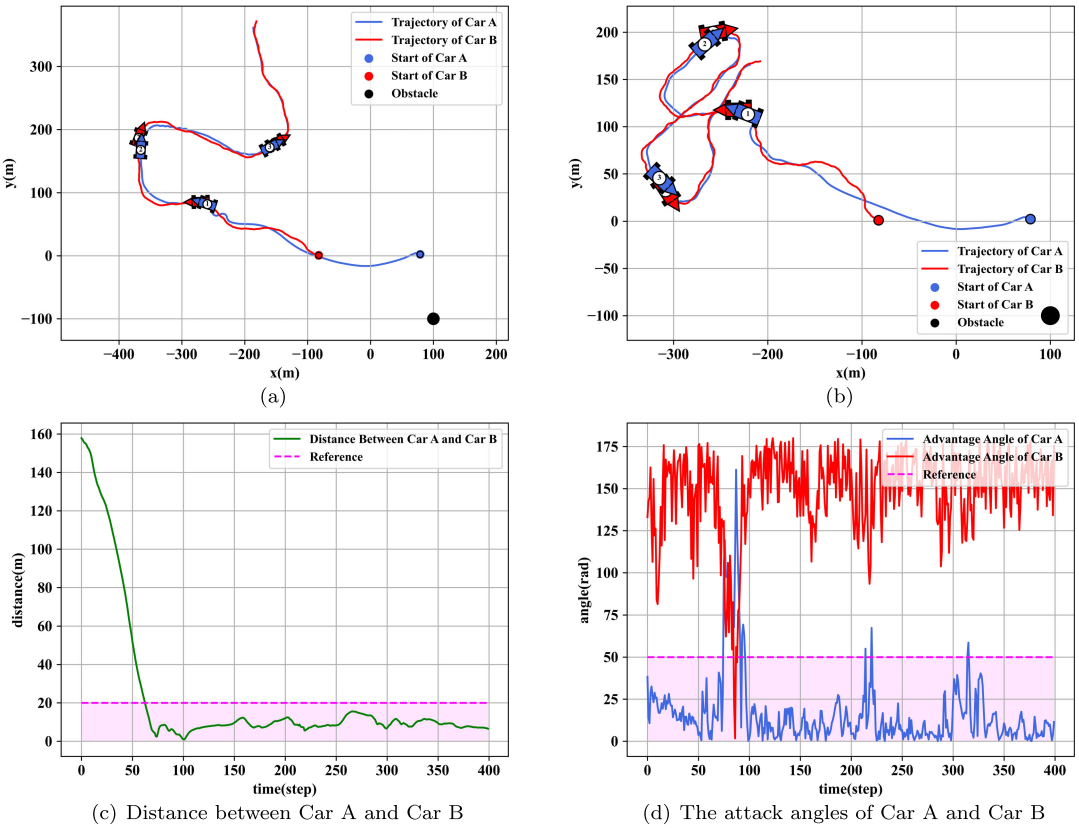


FIGURE 8. The track performance of Car A for the random movement of Car B starting from different initial points.

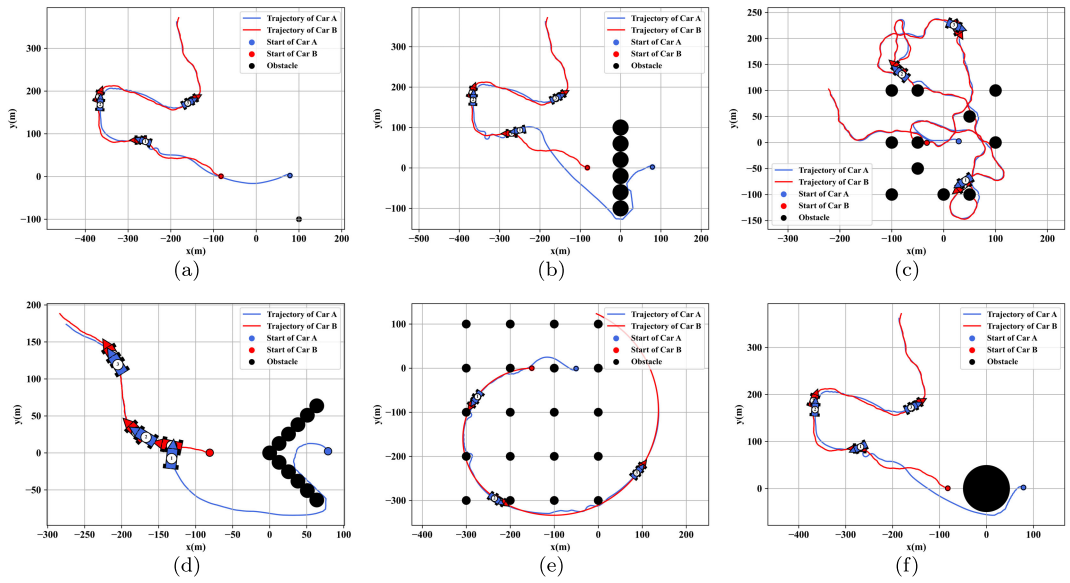
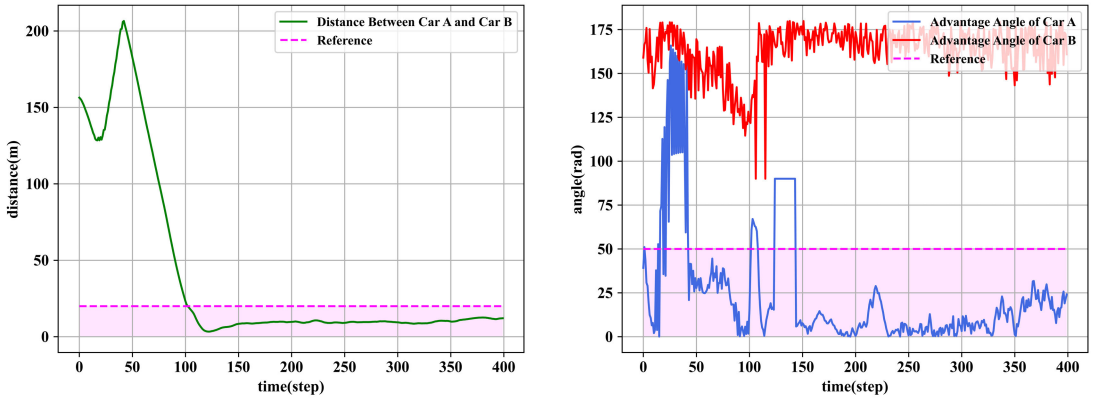


FIGURE 9. The autonomous maneuvering decision performances under the environments with obstacles.

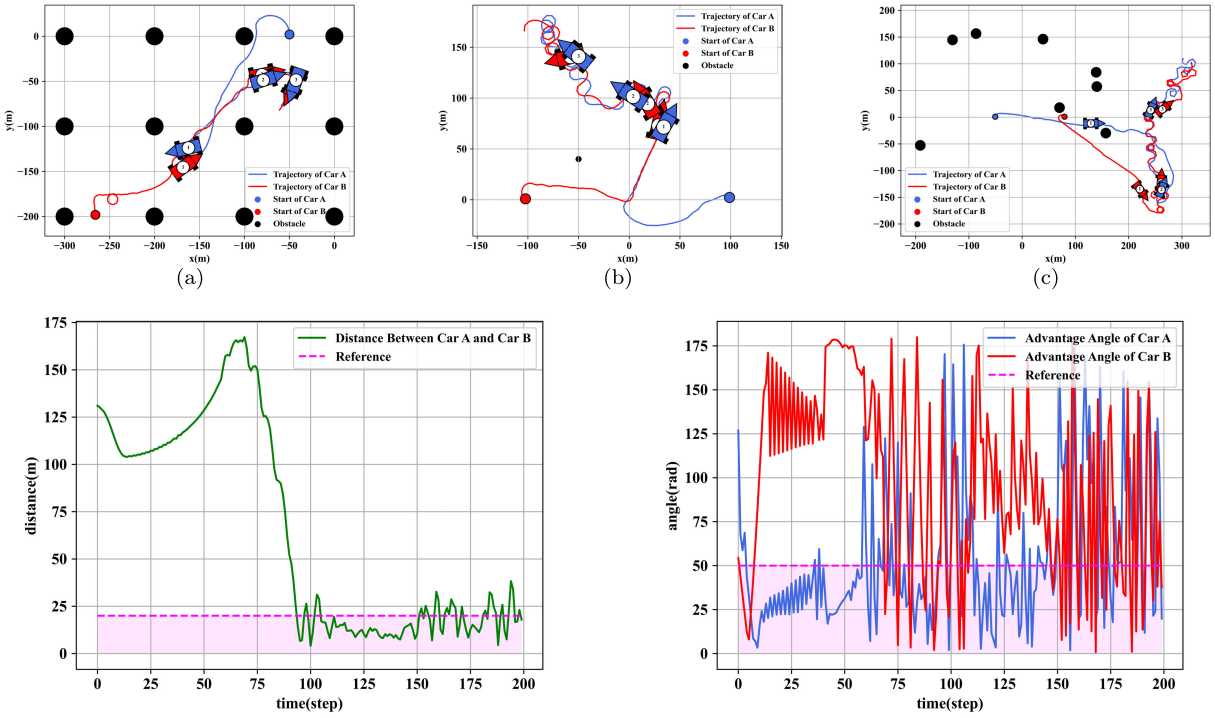
2) PERFORMANCES OF THE MANEUVERING DECISION POLICY AFTER CURRICULUM III AND IV

In the curriculum designs given in Table 1, more and more obstacles are randomly set in the environment from 20000 to 40000 episodes, and the goal of training for these two curricula is to improve the ability of autonomous obstacle-avoidance of Car A. After the 3000 episodes of training, the autonomous maneuvering



(a) Distance between Car A and Car B (b) The attack angles of Car A and Car B

FIGURE 10. Performance of Car A and Car B in two combat scenarios with random obstacle setting.



(a) (b) (c)

(d) Distance between Car A and Car B (e) The attack angles of Car A and Car B

FIGURE 11. The trajectories of Car A and Car B in two combat scenarios with random obstacle setting.

TABLE 3. The time ratios for confrontation scenario shown in Figure 9(f) and episode ratios of confrontation advantage in total 100 episode confrontation after curriculum III and IV.

car	Time ratio (%)	Episode ratio (%) (total 100 episode)
A	94.88	100.00
B	5.12	0.00

decision policy is applied to a confrontation with random obstacles. Figures 9(a)-(b) give the autonomous maneuvering decision performances of Car A under the environments without or with some obstacles. These results show that Car A shows a good autonomous obstacle-avoidance performance. Figures 9(c)-(d) give the simulation results under the environments with more obstacles. These results show that the policy of Car A gives good maneuvering decisions not only for obstacle-avoidance but also for confrontation. Figures 9(c)-(d) show the simulation results of confrontation under two different obstacle settings, where Car A shows good autonomous maneuvering decision ability for tracking and obstacle-avoidance.

For the confrontation scenarios shown in Figure 9(f), Figure 10(a) gives the distance curve between the Car A

TABLE 4. The time ratios for confrontation scenario shown in Figure 11(c) and episode ratios of confrontation advantage in total 100 episode confrontation after curriculum V and VI.

Car	Time ratio (%)	Episode ratio (%) after curriculum V (total 100 episode)	Episode ratio (%) after curriculum VI (total 100 episode)
A	68.00	66.00	85.00
B	32.00	34.00	15.00

and Car B, where the pink-shaded area indicates the superior attack distance of Car A, and Figure 10(b) gives the attack angles of Car A and Car B. These results show that Car A has the significant confrontation advantage in this scenario. Moreover, we compare the time ratios of the attack advantage of the two cars in each episode and the episode ratios of the confrontation advantage of the two cars over 100 episodes, as given in Table 2. All the results demonstrate that in the more complex obstacles simulations, Car A still has the significant advantage in confrontation, and the training objectives of curricula III and IV are achieved.

3) PERFORMANCES OF THE MANEUVERING DECISION POLICY AFTER CURRICULUM V AND VI

In this simulation, the maneuvering decision policies of Car A in curriculum IV and V are transplanted to Car B as the corresponding confrontation policies of curriculum V and VI, which leads to the self-play stage of the maneuvering decision policy. The training goal is to improve the autonomous confrontation and obstacle-avoidance performance of Car A until the expected performance is obtained. Figures 11 show the confrontation results of the two cars under the environments with random obstacles. These results show that Car A has more confrontation advantage than that Car B. Figure 11(d) gives the distance between Car A and Car B during the confrontation, where the pink-shaded area indicates the superior distance of Car A. Figure 11(e) gives the attack angles of Car A and Car B during the confrontation, where the pink-shaded area indicates the superior attack angle of Car A. In Table 4, we give the time ratios of attack advantage of the two cars for the scenario shown in Figure 11(c) and the episode ratios of the confrontation advantage of the two cars under 100 episode combats after curriculum V and VI respectively. These results show that through the self-play of the policy, the episode ratio of Car A is significantly increased, which demonstrates the improvement of the maneuvering decision policy by the self-play.

V. CONCLUSION

For the autonomous confrontation and obstacle-avoidance policy design of unmanned vehicles, this paper proposes a progressive RL algorithm based on the SAC framework. The proposed learning algorithm divides the training procedure of the agent into different curricula. By properly planning the learning objectives and the difficulty of the training curricula, the algorithm not only significantly improves the learning efficiency but also improves the ability of multi-task learning. In addition, the proposed algorithm realizes the iterative optimization of the confrontation policy, resulting in the confrontation performance improved persistently. This paper takes the autonomous confrontation and obstacle-avoidance of unmanned vehicles as the practical scenario and conducts the modeling and performance index design for the one-to-one confrontation and obstacle-avoidance. The effectiveness of the design algorithm is demonstrated through the numerical simulations. It is noted that the proposed design scheme is not only used for autonomous confrontation and obstacle-avoidance policy design of unmanned vehicles but also provides a feasible AI-based solution for the design of the automatic pilot system and the autonomous air combat maneuvering decision system for the UAVs.

REFERENCES

[1] S. Feng, J. Xi, C. Gong, J. Gong, S. Hu, and Y. Ma, "A collaborative decision making approach for multi-unmanned combat vehicles based on the behaviour tree," in *Proc. 3rd Int. Conf. Unmanned Syst. (ICUS)*, Nov. 2020, pp. 395–400.

[2] H. Lee, N. Kim, and S. W. Cha, "Model-based reinforcement learning for eco-driving control of electric vehicles," *IEEE Access*, vol. 8, pp. 202886–202896, 2020.

[3] Y. Tian, X. Cao, K. Huang, C. Fei, Z. Zheng, and X. Ji, "Learning to drive like human beings: A method based on deep reinforcement learning," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 7, pp. 6357–6367, Jul. 2022.

[4] S. Tang, H. Shu, and Y. Tang, "Research on decision-making of lane-changing of automated vehicles in highway confluence area based on deep reinforcement learning," in *Proc. 5th CAA Int. Conf. Veh. Control Intell. (CVCI)*, Oct. 2021, pp. 1–8.

[5] X. Zhang, H. Gao, M. Guo, G. Li, Y. Liu, and D. Li, "A study on key technologies of unmanned driving," *CAAI Trans. Intell. Technol.*, vol. 1, no. 1, pp. 4–13, Jan. 2016.

[6] J.-G. Lee, K. J. Kim, S. Lee, and D.-H. Shin, "Can autonomous vehicles be safe and trustworthy? Effects of appearance and autonomy of unmanned driving systems," *Int. J. Hum.-Comput. Interact.*, vol. 31, no. 10, pp. 682–691, Oct. 2015.

[7] J.-G. Lee, J. Gu, and D.-H. Shin, "Trust in unmanned driving system," in *Proc. 10th Annu. ACM/IEEE Int. Conf. Hum.-Robot Interact. Extended Abstr.*, Mar. 2015, pp. 7–8.

[8] B. R. Kiran, I. Sobh, V. Talpaert, P. Mannion, A. A. A. Sallab, S. Yogamani, and P. Pérez, "Deep reinforcement learning for autonomous driving: A survey," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 6, pp. 4909–4926, Jun. 2022.

[9] J. Chen, S. E. Li, and M. Tomizuka, "Interpretable end-to-end urban autonomous driving with latent deep reinforcement learning," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 6, pp. 5068–5078, Jun. 2022.

[10] J. Liu, L. Zhao, K. Zheng, and Q. Zhou, "A distributed driving decision scheme based on reinforcement learning for autonomous driving vehicles," in *Proc. IEEE 91st Veh. Technol. Conf. (VTC-Spring)*, May 2020, pp. 1–5.

[11] Z. Xu, J. Chen, and M. Tomizuka, "Guided policy search model-based reinforcement learning for urban autonomous driving," 2020, *arXiv:2005.03076*.

[12] P. X. Hien and G. Kim, "Goal-oriented navigation with avoiding obstacle based on deep reinforcement learning in continuous action space," in *Proc. 21st Int. Conf. Control, Autom. Syst. (ICCAS)*, Oct. 2021, pp. 8–11.

[13] L. Yue, Q. Xiaohui, L. Xiaodong, and X. Qunli, "Deep reinforcement learning and its application in autonomous fitting optimization for attack areas of UCAVs," *J. Syst. Eng. Electron.*, vol. 31, no. 4, pp. 734–742, Aug. 2020.

[14] G. Xu, S. Wei, and H. Zhang, "Application of situation function in air combat differential games," in *Proc. 36th Chin. Control Conf. (CCC)*, Jul. 2017, pp. 5865–5870.
[15] H. Park, B.-Y. Lee, M.-J. Tahk, and D.-W. Yoo, "Differential game based air combat maneuver generation using scoring function matrix," *Int. J. Aeronaut. Space Sci.*, vol. 17, no. 2, pp. 204–213, Jun. 2016.
[16] R.-Z. Xie, J.-Y. Li, and D.-L. Luo, "Research on maneuvering decisions for multi-UAVs air combat," in *Proc. 11th IEEE Int. Conf. Control Autom. (ICCA)*, Jun. 2014, pp. 767–772.
[17] Z. Lin, T. Minq'an, Z. Wei, and Z. Shcnquun, "Sequential maneuvering decisions based on multi-stage influence diagram in air combat," *J. Syst. Eng. Electron.*, vol. 18, no. 3, pp. 551–555, Sep. 2007.
[18] S. Zhang, Y. Zhou, Z. Li, and W. Pan, "Grey wolf optimizer for unmanned combat aerial vehicle path planning," *Adv. Eng. Softw.*, vol. 99, pp. 121–136, Sep. 2016.
[19] R. E. Smith, B. A. Dike, R. K. Mehra, B. Ravichandran, and A. El-Fallah, "Classifier systems in combat: Two-sided learning of maneuvers for advanced fighter aircraft," *Comput. Methods Appl. Mech. Eng.*, vol. 186, nos. 2–4, pp. 421–437, Jun. 2000.
[20] H. Changqiang, D. Kangsheng, H. Hanqiao, T. Shangqin, and Z. Zhuoran, "Autonomous air combat maneuver decision using Bayesian inference and moving horizon optimization," *J. Syst. Eng. Electron.*, vol. 29, no. 1, pp. 86–97, Feb. 2018.
[21] Q. Pan, D. Zhou, J. Huang, X. Lv, Z. Yang, K. Zhang, and X. Li, "Maneuver decision for cooperative close-range air combat based on state predicted influence diagram," in *Proc. IEEE Int. Conf. Inf. Autom. (ICIA)*, Jul. 2017, pp. 726–731.
[22] D. Wang, W. Zu, H. Chang, and J. Zhang, "Research on automatic decision making of UAV based on plan goal graph," in *Proc. IEEE Int. Conf. Robot. Biomimetics (ROBIO)*, Dec. 2016, pp. 1245–1249.
[23] Y. Wang, C. Huang, and C. Tang, "Research on unmanned combat aerial vehicle robust maneuvering decision under incomplete target information," *Adv. Mech. Eng.*, vol. 8, no. 10, pp. 1–12, 2016.
[24] H.-F. Guo, M.-Y. Hou, Q.-J. Zhang, and C.-L. Tang, "UCAV robust maneuver decision based on statistics principle," *Acta Armamentarii*, vol. 38, no. 1, pp. 160–167, 2018.
[25] W.-X. Geng, F. Kong, and D.-Q. Ma, "Study on tactical decision of UAV medium-range air combat," in *Proc. 26th Chin. Control Decis. Conf. (CCDC)*, May 2014, pp. 135–139.
[26] L. Fu, F. Xie, G. Meng, and D. Wang, "An UAV air-combat decision expert system based on receding horizon control," *J. Beijing Univ. Aeronaut. Astronaut.*, vol. 41, no. 11, p. 1994, 2015.
[27] R. W. Schvaneveldt, T. E. Goldsmith, A. E. Benson, and W. L. Waag, "Neural network models of air combat maneuvering," New Mexico State Univ., Las Cruces, NM, USA, Tech. Rep. AL-TR-1992-0037, 1992.
[28] L. J. Ding and Q. M. Yang, "Research on air combat maneuver decision of UAVs based on reinforcement learning," *Avionics Technol.*, vol. 49, no. 2, pp. 29–35, 2018.
[29] P. Liu and Y. Ma, "A deep reinforcement learning based intelligent decision method for UCAV air combat," in *Proc. 17th Asia Simulation Conf. (AsiaSim)*, Melaka, Malaysia. Singapore: Springer, Aug. 2017, pp. 274–286.
[30] J. Zuo, R. Yang, Y. Zhang, Z. Li, and M. Wu, "Intelligent maneuver decision in air combat maneuvering based on heuristic reinforcement learning," *Acta Aeronautica Et Astronautica Sinica*, vol. 38, no. 10, pp. 1–14, 2017.
[31] X. Zhang, G. Liu, C. Yang, and J. Wu, "Research on air combat maneuver decision-making method based on reinforcement learning," *Electronics*, vol. 7, no. 11, p. 279, Oct. 2018.
[32] R. S. Sutton and A. G. Barto, "Reinforcement learning: An introduction," *IEEE Trans. Neural Netw.*, vol. 9, no. 5, p. 1054, Sep. 1998.
[33] V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra, and M. Riedmiller, "Playing Atari with deep reinforcement learning," 2013, *arXiv:1312.5602*.
[34] Q. Yang, J. Zhang, G. Shi, J. Hu, and Y. Wu, "Maneuver decision of UAV in short-range air combat based on deep reinforcement learning," *IEEE Access*, vol. 8, pp. 363–378, 2020.
[35] X. Ma, L. Xia, and Q. Zhao, "Air-combat strategy using deep Q-learning," in *Proc. Chin. Autom. Congr. (CAC)*, Nov. 2018, pp. 3952–3957.
[36] Y. Zhang, W. Zu, Y. Gao, and H. Chang, "Research on autonomous maneuvering decision of UCAV based on deep reinforcement learning," in *Proc. Chin. Control Decis. Conf. (CCDC)*, Jun. 2018, pp. 230–235.
[37] W. Kong, D. Zhou, Z. Yang, Y. Zhao, and K. Zhang, "UAV autonomous aerial combat maneuver strategy generation with observation error based on state-adversarial deep deterministic policy gradient and inverse reinforcement learning," *Electronics*, vol. 9, no. 7, p. 1121, Jul. 2020.
[38] T. Haarnoja, A. Zhou, P. Abbeel, and S. Levine, "Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor," in *Proc. Int. Conf. Mach. Learn.*, 2018, pp. 1861–1870.
[39] T. Haarnoja, A. Zhou, K. Hartikainen, G. Tucker, S. Ha, J. Tan, V. Kumar, H. Zhu, A. Gupta, P. Abbeel, and S. Levine, "Soft actor-critic algorithms and applications," 2018, *arXiv:1812.05905*.
[40] Z. Wang, H. Li, H. Wu, and Z. Wu, "Improving maneuver strategy in air combat by alternate freeze games with a deep reinforcement learning algorithm," *Math. Problems Eng.*, vol. 2020, pp. 1–17, Jun. 2020.

CHENGDONG MA received the B.S. degree in physics from the Beijing University of Aeronautics and Astronautics, Beijing, China, in 2020. He is currently pursuing the master's degree with Xiamen University, Xiamen, China. His research interests include RL, game theory, intelligent control, and multi-agent systems.

JIANAN LIU received the B.S. degree in mechanical and electrical engineering from Xidian University, Xi'an, China, in 2015, and the M.S. degree in mechatronics and information technology from the Karlsruhe Institute of Technology, Karlsruhe, Germany, in 2020. He is currently pursuing the Ph.D. degree with the Institute of Artificial Intelligence, Xiamen University, Xiamen, China. His current research interests include iterative learning control, RL, and intelligent control.

SAICHAO HE received the B.S. degree in automation from Northeastern University, Shenyang, China, in 2021. He is currently pursuing the master's degree with the School of Information, Xiamen University, Xiamen, China. His research interests include machine learning, knowledge RL, intelligent control, and autonomous driving.

WENJING HONG has been a full professor on process control and chemical engineering, since 2015. His research interests include control engineering and artificial intelligence. More than 100 peer-reviewed articles are published in top journals of chemical engineering and artificial intelligence, including articles on *Nature Materials*, *Nature Chemistry*, *Science Advances*, *Chemistry*, *Nature Communications*, and *Matter*. These articles have been cited for more than 7000 times, while the most cited paper has been cited more than 700 times, since 2008.

JIA SHI received the M.Sc. degree in operational research and cybernetics from Xiamen University, China, in 1997, and the Ph.D. degree in control science and engineering from Zhejiang University, China, in 2006. Since 2008, he has been an Associate Professor with the Department of Chemical and Biochemical Engineering, Xiamen University. His current research interests include intelligent learning control and optimization for complex industrial processes, which mainly include iterative learning control, RL, and composite intelligent control combined with various advanced control techniques. So far, about more than 40 research papers are published in the process control and optimization.