Abstract—Cooperative guidance of multiple missiles is a challenging task with rigorous constraints of time and space consensus, especially when attacking dynamic targets. In this article, the cooperative guidance task is described as a distributed multiobjective cooperative optimization problem. To address the issues of nonstationarity and continuous control faced by cooperative guidance, the natural evolutionary strategy (NES) is improved along with an elitist adaptive learning technique to develop a novel natural coevolutionary strategy (NCES). The gradients of the original evolutionary strategy (ES) are rescaled to reduce the estimation bias caused by the interaction between the multiple missiles. A hybrid coevolutionary cooperative guidance law (HCCGL) is then developed by integrating the highly scalable co-ES and the proportional guidance law, with detailed convergence proof provided. Finally, simulations demonstrated the effectiveness and superiority of this guidance law in solving cooperative guidance tasks with high accuracy, with potential applications in other multiobjective optimization, dynamic optimization, and distributed control scenarios.

Index Terms—Cooperative guidance, evolutionary strategy (ES), multiobjective optimization, optimal control.

I. INTRODUCTION

M Odern penetration of air defense systems of the target requires coordinated attacks with multiple missiles. However, the rapid development of detection technologies and close-in weapon systems (CIWSs) has decreased the chances of successful impact with a single conventional missile [1]. In addition to increasing the difficulty of interception, the cooperative guidance strategy of multiple missiles is also crucial to the lethal effect of the final impact. Usually, the cooperative guidance of multiple missiles belongs to the phase of terminal guidance, where accurate target information can be obtained with active radar systems or other detection devices. The existing cooperative guidance laws can be roughly divided into two categories. One is the analytical method to find closed-form solutions, which is mainly based on sliding mode control, optimal control, and multiagent consensus theory. The other is the intelligent method, which generally adopts heuristic intelligent optimization algorithm and reinforcement learning (RL) theory.

The analytical cooperative guidance method has been proven to be robust and efficient for practical application [2], [3], [4], [5], [6], [7]. Based on fundamental proportional navigation (PN), Jeon et al. [1] developed cooperative proportional navigation (CPN) where the on-board time-to-go of the missile is used as the navigation gain. It is a simple but effective approach for achieving time consensus. Ma et al. [2] developed a composite guidance law, which can be decomposed into the direction along the line of sight (LOS) and the direction perpendicular to LOS, corresponding to time and space cooperative, respectively. Furthermore, time cooperative control is achieved with the combination of PNG and impact time error feedback [8], where the undirected topology is adopted to establish communication relationships. Based on the optimal control approach, a variant of the hyperbolic tangent function is proposed in [3] to force early control of velocity and impact angle.

However, with the increasing demand for developing high-precision weapon systems, intelligent cooperative guidance method is increasingly regarded as a necessary auxiliary option. In recent years, the RL theory has attracted much attention because of its ability to learn online based on environmental feedback [9], [10], [11], [12], [13], [14], [15]. According to the training structures, existing RL algorithms can be roughly divided into four types, which are fully decentralized training, decentralized execution; fully centralized training, decentralized execution; centralized training, centralized execution, and value decomposition methods. Some of these algorithms have achieved satisfactory results in coping with problems with low complexity and accuracy requirements. In [5], [16], and [17], the state-of-the-art RL frameworks have demonstrated their effectiveness in the guidance task. Zhang et al. [18] proposed a gradient-descent-based RL method in the actor-critic framework and achieved consensus control for multiagent systems by following a tracking leader. The two challenges of nonstationarity and partial observability [19] will lead to saturated output or coordination loss of multiagent systems, which greatly reduces the accuracy of the value function. In addition, the use of value function in RL is not suitable for continuous control tasks with large search spaces. Thus, these limita-
tions of RL impede the development of RL in cooperative guidance.

It is an excellent way to solve the above problems by removing the value function of RL and optimizing in solution space with evolutionary strategy (ES), which is more robust and invariant to real-time rewards, because it optimizes toward the objective function directly [20]. Moreover, as described in [21], ES is tolerant of long horizontal and implicit solutions, which is exactly consistent with the need for cooperative guidance. The natural ES (NES) is the latest branch of ES and shows good performance in solving high-dimensional continuous multimodal optimization problems, by using the natural gradient information estimated according to the fitness expectation of the population [20], [21], [22]. Similar algorithms named coevolutionary algorithm have been discussed in [23] and [24], which focus on solving multi-objective optimization problems by dividing the overall objective into subobjectives, to optimize and evaluate together. Another idea is to evolve multiple populations for the same goal and manually regulate the constraints of each population for faster convergence or fuller exploration [24]. As represented in [24], the concept of coevolution refers to multithreads of training processes. Note that these methods do not use the natural gradient information as in NES, and the nonstationary issue discussed above is not considered.

When optimizing in continuous parameter (solution) space, it is very important to apply adaptive technology. While a learning rate adaption method based on the quality of gradients is often not easy to estimate, a simple workaround would be leveraging the shifting distance of parameters to adapt the learning rate. As shown in [25], the size of population was adjusted depending on the novelty metric and quantity metric, which reflected the complexity of the dynamic environment. The estimation of distribution algorithm (EDA) was applied to continuous control by searching the optimal parameter distribution [26], [27]. A variety of evolutionary methods were investigated to design the multiobjective missile guidance law [28]. Maheswaranathan et al. [29] proposed a surrogate gradient to reduce the evaluation costs. These works reveal the enormous potential of searching in parameter space, rather than directly searching in parameter space.

Therefore, an NES-based coevolutionary algorithm naming as the natural co-ES (NCES) is developed in this article to distress the dilemma faced by RL in the cooperative guidance task. Considering the advantages of searching in parameter space, the coevolutionary algorithm is improved in this work by rescaling the gradient information to reduce the estimation bias introduced by neighboring populations. As discussed in [30] and [31], most of today’s bio-inspired algorithm innovations are based on experimental observation rather than meticulous theoretical support. In this work, we try to dig into the depths of complex optimization and provide proof as sensible as possible through the presentation of graphs and deduction. Via integrating the NCES algorithm, a hybrid coevolutionary cooperative guidance law (HCCGL) is further developed to solve the challenging missile guidance problem. Extensive empirical results on various engagement scenarios verified the effectiveness of the proposed guidance law. The main contributions of this work are summarized as follows.

1) To address the issues of nonstationarity and continuous control faced by cooperative guidance, an NCES algorithm is formulated and incorporated into a novel guidance law as an alternative to RL in the cooperative guidance task.

2) The rigorous constraints of time and space consensus in cooperative guidance are integrated and designed as the fitness function for each missile. An MLP-based policy network is constructed and learned to optimize the fitness function.

3) The proposed HCCGL has advantages in achieving high precision for cooperative guidance tasks, even with dynamic targets and random initial conditions.

The rest of this article is organized as follows. The problem formulation is elaborated in Section II, and the proposed cooperative guidance law is discussed in Section III. In Section V, experiments under various configurations are implemented. Finally, conclusions are made in Section VI.

II. PROBLEM FORMULATION

A. Engagement Geometry

The 2-D engagement geometry between multiple missiles and one target is shown in Fig. 1, where the inertial coordinate frame $OXY$ represents the horizontal plane. There are $n$ missiles in total. The index $M_i$ denotes the $i$th missile, and $T$ represents the target. $v_{mi}$, $\alpha_{mi}$, and $\delta_{mi}$ represent the velocity, LOS angle, flight-path angle, and heading angle of the $i$th missile, respectively. $a_{ii}$ and $a_{iv}$ represent the lateral acceleration and the thrust acceleration to be designed for the $i$th missile, which are perpendicular to and align with the direction of $V_{mi}$, respectively. $V_T$, $\alpha_T$, and $\delta_T$ are the velocity, LOS angle, flight-path angle, and heading angle of the target, respectively. The lateral acceleration of the target is denoted by $a_T$. 

![Fig. 1. 2-D engagement geometry.](image-url)
The dynamic equations of the $i$th missile and the target are as follows:

$$\begin{align*}
    \dot{r}_i &= -V_{mi} \cos \delta_{mi} + V_T \cos \delta_{T_i}, \\
    \dot{\xi}_{mi} &= -V_{mi} \sin \delta_{mi} + V_T \sin \delta_{T_i}, \\
    \dot{a}_{mi} &= a_{ti} / V_{mi}, \\
    \dot{a}_T &= a_T / V_T, \\
    \dot{V}_{mi} &= a_{vi}, \\
    \delta_{mi} &= a_{mi} - \xi_{mi}, \\
    \delta_{T_i} &= a_T - \xi_{mi},
\end{align*}$$

(1)

where $r_i$ represents the relative range between the $i$th missile and the target. The time-to-go of the $i$th missile $t_{go}^i$ refers to the time left from the current time until the interception

$$t_{go}^i = \frac{-r_i}{\dot{r}_i}.$$  (2)

B. Communication Topology

The communication relationship of the multiple missiles is depicted by a topology, where a set of nodes $\mathcal{V} = \{v_1, v_2, \ldots, v_n\}$ represents the $n$ missiles. The communications are represented by a set of edges $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ with an adjacency matrix $A = [a_{ij}] \in \mathbb{R}^{n \times n}$, where $a_{ij} = 1$ if missile $j$ is able to communicate directly with missile $i$, otherwise $a_{ij} = 0$. $\mathcal{N}_i = \{j \in \mathcal{V} : (i, j) \in \mathcal{E}\}$ is the set of neighboring missiles of the $i$th missile. In practical engineering, the communication topology is determined through comprehensive considerations of the communication cost and actual demand. In this work, the undirected topology shown in Fig. 2 is adopted, enabling neighboring missiles to share information.

C. Observation

For the multimissile system, the complete observation information of the entire system is not available to each agent. Thus, the cooperative guidance problem is a partially observable Markov decision process (POMDP) described by

$$O_i \times A_i \rightarrow O'_i$$

(3)

where $O_i$ and $A_i$ represent the observation and action of the $i$th missile. $O'_i$ is the observation of the $i$th missile at next time step.

The full state information of each missile consists of three components: personal features, target features, and error features shown in Table I. $P_{mi}$ and $P_T$ represent the positions of the missile $i$ 2-D coordinates. The target features are estimated or detected through onboard equipment, and the estimation error is assumed to be negligible compared with the required guidance precision. The acquisition of accurate location information requires the support of powerful global positioning systems; here, we only need relative error information. $e_i'$ is the consensus error of time of the missile $i$

$$e_i' = \sum_{j \in \mathcal{N}_i} (t_{go}^j - t_{go}^i).$$

(4)

The consensus error of LOS angle of the missile $i$ is defined as follows:

$$e_a' = \sum_{j \in \mathcal{N}_i} (e_a^j - e_a^i)$$

(5)

where $e_a^j = \xi_{mi} - \xi_{di}$ is the LOS angle error of the missile $i$, and $\xi_{di}$ is the desired impact angle of missile $i$

$$\xi_{di} = \xi_{d1} + \sum_{j=1}^{i-1} \delta_{d}^{ij}$$

(6)

where $\delta_{d}^{ij}$ is the desired relative impact angle between two missiles, and $\xi_{d1}$ is the nominal desired impact angle of the first missile which is determined online. To increase the flexibility and autonomy of the intelligent missile system, the desired $\xi_{di}$ can be adjusted adaptably instead of being a fixed value.

D. Fitness Evaluation

The reward of each missile at one evaluation step consists of a terminal reward and a flight reward. The objective of the cooperative guidance task is to minimize the error $e_i'$, $e_a'$, and $e_a^i$.

Then, the terminal reward is defined as follows:

$$r_T^i = \left(\gamma_a \cdot e_a'^i + \gamma_{\xi} \cdot e_\xi |\xi_i| \right) \cdot \epsilon(k)$$

(7)

where $\gamma_a$, $\gamma_{\xi}$, and $\gamma_t$ are constant coefficients. $\epsilon(k)$ is the step function defined as follows:

$$\epsilon(k) = \begin{cases} 
1, & \text{if } k \text{ is terminal step} \\
0, & \text{otherwise}.
\end{cases}$$

(8)

Thus, the terminal reward only reflects the results at the terminal step, and $r_T^i = \gamma_a + \gamma_{\xi}$ if and only if $e_a^i = 0$ and $e_a' = 0$. The flight reward is defined as follows:

$$r_F^i = \beta_a \left(1 + e^{-k_i |\xi_i|}\right) + \beta_t \left(1 + e^{-k_i |\xi_i|}\right)$$

(9)
where $k_a, k_r, \beta_a$, and $\beta_r$ are positive constant coefficients. It can be inferred that $r^2 \leq 0$ is always true. $r^2 = 0$ if and only if $e^a = 0$ and $e^r = 0$. Then, the fitness function of missile $i$ for the cooperative guidance task is defined by

$$F_i = \int (r^2 + r^2)dt.$$  

(10)

Thus, the objective of the cooperative guidance task can be achieved by maximizing the fitness function of each missile.

E. Design of the Cooperative Guidance Law

Based on the requirements of the cooperative guidance task, the guidance law proposed in this article includes two parts: the tracking control part and the consensus control part. The tracking control part is obtained by proportional navigation guidance (PNG)

$$u_{pi} = [\beta \vec{Z}_{mi} V_{mi}, 0]^T$$  

where $\beta$ is the navigation constant. Note that the tracking control part only designs the lateral acceleration.

The consensus control part is modeled by a neural network expressed as follows:

$$u_{ei} = W_{xi}^T \cdot \psi(Z_{2i})$$  
$$Z_{2i} = W_{2i}^T \cdot \phi(Z_{1i})$$  
$$Z_{1i} = W_{1i}^T \cdot \phi(X_i)$$  

(12)

where $W_{3i} \in \mathbb{R}^{q_3 \times 2}, W_{2i} \in \mathbb{R}^{q_2 \times q_3},$ and $W_{1i} \in \mathbb{R}^{q_1 \times q_2}$ denote the weight matrices of the output layer. $Z_{1i}$ and $Z_{2i}$ are the outputs of the first and second hidden layers. $q_1$ and $q_2$ are the numbers of neurons in each layer. $\psi(\cdot)$ is the bounded activation function $Tanh(\cdot)$ with $||\psi(\cdot)|| \leq a_{max}$, and $\phi(\cdot)$ is the common activation function $Sigmoid(\cdot)$. The input state vector $X_i$ is selected as follows:

$$X_i = [e^a_i, e^r_i, e^p_i]^T.$$  

(13)

Thus, the guidance law of the missile $i$ is presented as follows:

$$u_i = (1 - \eta)u_{pi} + \eta u_{ei}$$  

(14)

where $\eta$ is the guidance gain trading off the tracking control part and the consensus control part.

III. NATURAL COEVOLUTIONARY STRATEGY

A. Bottleneck of RL

RL is a generic term for a class of value-oriented algorithms. It focuses on solving problems of Markov decision process (MDP), which also applies to the guidance problem.

Assume there are $n$ agents in the environment. The joint action set is denoted by $A = A_1 \times A_2 \times \cdots \times A_n$, and the system state is denoted by $S$. At each timestep, each agent takes one step, and with a certain probability, the transition occurs. This can be represented as $S \times A \rightarrow S'$, where $S'$ is the next system state after the transition. The reward is given as $S \times A \times S' \rightarrow \mathbb{R}$. In a deterministic environment, the transition probability is 1. For the multiagent system, $S$ can be decomposed into individual observations: $S = O_1 + O_2 + \cdots + O_n$. A sufficient set of observations must be capable of representing the complete system state. In most cases, the agent does not have access to the complete information of the system. This means that they only get partial observations instead of the complete state, making the problem a POMDP.

Two challenges, nonstationarity and partial observability [19], impede the research for multiagent systems. Tons of algorithms have popped up focusing on solving this kind of problem. According to the training process, we can roughly divide them into four types.

1) Fully decentralized training, decentralized execution.
2) Fully centralized training, decentralized execution.
3) Centralized training, centralized execution.
4) Value decomposition methods.

The existing works have achieved satisfactory results in coping with problems of less complexity and less requirement for precise control. It has been well investigated that an ill-distributed value function would seriously stagnate performance. Exploration technologies, such as Ornstein–Uhlenbeck noise and stochastic exploration, are used to alleviate this problem.

Applying RL theory, it is possible to handle control tasks with either a single missile [32], [33] or discrete action space [12], [34]. However, for control tasks with multiple agents (missiles), inefficient exploration and nonstationarity can lead to a deterioration of the accuracy of the value function, resulting in either saturated control or coordination loss. Value functions can be advantageous for discrete control but can be flawed for continuous control tasks with large search spaces. Approaches that constrain the policy space have been discussed in [35], but they heavily rely on prior knowledge and do not scale well to different scenarios. As an alternative, evolutionary strategies have abandoned the use of value functions and have shown the outstanding capability for the aforementioned issues.

B. NES in Multiagent POMDP

In the ES, individual agent (or its policy) is expressed as a population, the group of populations, and the environment constitute the ecosystem. The objective is to develop the optimal strategy for the group of populations to maximize the fitnesses of the ecosystem. For cooperative tasks, the optimal strategy of the ecosystem will be exactly the optimal policy for each population

$$\arg \max_{u_{tot}^*} F_{tot}(u_{tot}^*) = \left( \begin{array}{c} \arg \max_{u_1^*} F_1(u_1^*) \\ \vdots \\ \arg \max_{u_n^*} F_n(u_n^*) \end{array} \right).$$  

(15)

where $u_i$ which is defined in (14) represents the policy of the $i$th population and $u_{tot}^* = \left[ u_1^* \cdots u_n^* \right]$ is the joint matrix of individual optimal policy. $F_i(\cdot)$ is its corresponding fitness function and $F_{tot}(\cdot)$ is the joint policy fitness function; more details can be viewed in [36]. However, the inverse is not true

$$\left( \begin{array}{c} \arg \max_{u_1} F_1(u_1) \\ \vdots \\ \arg \max_{u_n} F_n(u_n) \end{array} \right) \neq \arg \max_{u_{tot}} F_{tot}(u_{tot}).$$  

(16)

This is because the optimal fitness obtained by one population may be based on the suboptimal fitness obtained by other
populations. When the other populations evolve, the previous optima are easy to be broken. To overcome this nonstationary issue, it is best for all populations to evolve simultaneously, that is, coevolution. Each generation updates its parameter at the same time, instead of updating sequentially, mapping in slight variance in fitness values.

C. Optimization in Coevolutionary Parameter Space

The gradient information is obtained by measuring the contribution of each sample. The parameters of the population are defined as $\theta$, and $\theta'$ represents that of the next generation. $p_\psi(\theta'|\theta)$ is the distribution function of $\theta'$ under $\theta$, where $\psi$ is the intrinsic parameter. Then, the expectation fitness of the next generation is expressed as follows:

$$
\mathbb{E}_{\theta \sim p_\psi(\theta'|\theta)} F(\theta') = \int_{\theta'} p_\psi(\theta'|\theta) F(\theta') d\theta'.
$$

(17)

The derivative of Eq. (17) with respect to $\theta$ is

$$
\nabla_\theta \mathbb{E}_{\theta \sim p_\psi(\theta'|\theta)} F(\theta') = \mathbb{E}_{\theta} \{ \nabla_\theta \log p_\psi(\theta'|\theta) F(\theta') \}. 
$$

(18)

If we represent $\theta'$ as $\theta + \epsilon$, then we have the similar equation

$$
\nabla_\theta \mathbb{E}_{\theta \sim p_\psi(\epsilon)} F(\theta + \epsilon) = \mathbb{E}_{\theta} \{ \nabla_\epsilon \log p_\psi(\epsilon) F(\theta + \epsilon) \}. 
$$

(19)

In an ecosystem with multiple populations, populations will interact and affect the evolutionary process. Thus, the fitness function of the $i$th population is represented by $F_i(\varsigma_i)$, where $\varsigma_i = \{\theta_i, \theta_j : j \in N_i\}$ represents the parameter set of the $i$th population and its neighboring populations. The expected joint fitness of the next generation is expressed as follows:

$$
\mathbb{E}[F_i(\varsigma'_i)] = \int_{\varsigma'_i} p_\psi(\varsigma'_i|\varsigma_i) F_i(\varsigma'_i) d\varsigma'_i.
$$

(20)

where $p(\varsigma'_i|\varsigma_i)$ is the joint probability distribution of the next generation over $\varsigma_i$. Assume that $\theta'_i$ and $\theta'_j$ are sampled independently, we have $p(\varsigma'_i|\varsigma_i) = p(\theta'_i) \prod_{j \in N_i} p(\theta'_j)$.

The gradient of the joint fitness with respect to $\theta_i$ is expressed as follows:

$$
\nabla_{\theta_i} \mathbb{E}[F_i(\varsigma'_i)] = \nabla_{\theta_i} \int_{\varsigma'_i} p(\varsigma'_i|\varsigma_i) F_i(\varsigma'_i) d\varsigma'_i = \int_{\varsigma'_i} \nabla_{\theta'_i} \prod_{j \in N_i} p(\theta'_j) d\theta'_i \times p(\theta'_i) \prod_{j \in N_i} d\theta'_j
$$

$$
= \int_{\varsigma'_i} \nabla_{\theta_i} \log p(\theta'_i) F_i(\varsigma'_i) p(\theta'_i) \prod_{j \in N_i} d\theta'_j
$$

$$
= \mathbb{E}_{\varsigma'_i} \{ \nabla_{\theta_i} \log p(\theta'_i) F_i(\varsigma'_i) \}. 
$$

(21)

Note that it has the same format as the version of a single population; it seems to be fine if we just keep the original equation. The influence of $\theta'_i$ is counteracted through the calculation of its expectation. However, it is known that the expectation of the joint distribution is approximated through sampling with a limited size. Although individuals are sampled without bias(unbiased estimation), there exists an intrinsic bias for inadequate sampling, and the bias will grow linearly with an increment of distribution dimensionality. So it can be a serious issue when taking the expectation of all neighboring parameters, and the sample size stays relatively small.

It is not necessary to take account of all parameters, since only the expectation of $\theta_i$ is actually needed. To alleviate the incremental bias, we propose to approximate only the expectation of the parameter of the current population $\theta_i$ and ignore its neighbor parameters, which is

$$
\mathbb{E}_{\theta_i} F_i(\theta'_i) = \int_{\theta'_i} F_i(\theta'_i) p(\theta'_i) d\theta'_i. 
$$

(22)

Though $p(\theta'_i)$ is available for independent distribution, it is infeasible to obtain $F_i(\theta'_i)$, since all agents are sampled and evaluated together. However, the expectation of individual fitness can be approximated by the multiplication between the original fitness $F_i(\varsigma'_i)$ and its confidence. The rectified expectation is expressed as

$$
\mathbb{E}_{\theta_i} F_i(\theta'_i) = \int_{\theta'_i} F_i(\varsigma'_i) p(\theta'_i) \prod_{c \in N_i} p(\theta'_c) d\theta'_i 
$$

(23)

where $\prod_{c \in N_i} p(\theta'_c)$ is the confidence, and $\theta'_i$ represents the samples that appear along with $\theta'_i$. In this way, the bias of estimating the expectation of the neighboring distributions is addressed. The gradient after modification is

$$
\nabla_{\theta_i} \mathbb{E}_{\theta_i} F_i(\theta'_i) = \int_{\theta'_i} \{ \nabla_{\theta'_i} \log p(\theta'_i) F_i(\varsigma'_i) \} p(\theta'_i) \prod_{c \in N_i} p(\theta'_c) d\theta'_i
$$

$$
= \mathbb{E}_{\theta_i} \{ \nabla_{\theta_i} \log p(\theta'_i) F_i(\varsigma'_i) \} \prod_{c \in N_i} p(\theta'_c) 
$$

(24)

Remark 1: We refer to strategies that use (24) as the updating policy as the NCES. The NCES is a strategy for evolutionary algorithms that updates the weights of the population in multiagent systems. Compared to other existing ES algorithms, NCES alleviates the incremental bias caused by neighboring parameters and achieves better performance in cooperative continuous control tasks.

The core idea is that although the individual fitness $F_i(\theta'_i)$ does not exist, the expectation of the individual fitness does, and is invariant to the parameter distributions of its neighboring agents, so the expectation of the individual agent’s fitness should be calculated instead of including the expectation of neighboring agents. Let us denote the expectation of the objective function over $\varsigma'_i$, which is $\nabla_{\theta'} \log p(\theta'_i) F_i(\varsigma'_i)$, by $\phi(\varsigma'_i)$ and the expectation of the objective function over $\theta'_i$ by $\phi(\theta'_i)$, such that

$$
\phi(\theta'_i) = \int_{\theta'_i} \phi(\varsigma'_i) p(\theta'_i) d\theta'_c. 
$$

(25)

To visualize the sampling estimation process, we use a variant of eggholder as the objective function for demonstration, which is defined in (26) shown in Table II, since the real objective is too expensive to obtain. Assuming there exists one neighboring population $\theta_c$ for $\theta$, with the size of 400, the sampled individuals are shown in Fig. 3, following a bivariate normal distribution, and the
Bias to the level of a single population, resulting in a more accurate estimation of gradient information, empirical results also supported this conclusion. However, when the population size is large enough (e.g., thousands), this approach may not result in additional accuracy improvements.

The modified expression is also desirable for parallel computing, as only the perturbation of the neighboring populations is needed, which can be easily obtained through communication among processes, and the probabilities can be calculated in a distributed approach.

### D. Elitist Adaptation Techniques

The performance of NES is sensitive to hyperparameters, and the learning rate is usually the most critical hyperparameter of NES. Thus, an elitist adaptation method for the learning rate is applied in this article. First, a list of learning rates is linearly selected in the neighborhood of the original learning rate \( \eta_0 \) as follows:

\[
\eta_{\text{cad}} = \{ \text{clip}((1 + 0.1k)\eta_0, \eta_{\text{min}}, \eta_{\text{max}}) : k \in \mathbb{Z}, -l/2 \leq k \leq l/2 \}
\]  

(27)

where \( \eta_{\text{cad}} \in \mathbb{R}^{m+1} \). The \( \eta_{\text{min}} \) and \( \eta_{\text{max}} \) are the minimum and maximum values of \( \eta_0 \). \( l \) is the size of perturbations which is clipped by \( \text{clip}() \). To evaluate the quality of the candidate learning rates, the evaluation function \( G_i(\cdot) \) is defined

\[
G_i(\eta_{\text{cad}}) = \left( \begin{array}{c}
F_i(\theta_i + \eta_{\text{cad}}^{1/2}g_{0i}) - F_i(\theta_i + \eta_0g_{0i}) \\
\vdots \\
F_i(\theta_i + \eta_{\text{cad}}^{1/2}g_{ni}) - F_i(\theta_i + \eta_0g_{ni})
\end{array} \right)
\]

(28)

where \( \eta_{\text{cad}}^{1/2} \) is the \( k \)-th-sampled learning rate of the candidate list. The gradient \( g_0 \) is kept after evaluation. Therefore, by comparing the candidate learning rates with the original one, the next update can be better than the previous one. Considering peer pressure, each missile is assigned the same learning rate. The learning rate of the next generation is obtained by

\[
\eta'_i = \operatorname{arg} \max_{\eta'_i} \left( \sum_{i=1}^{n} G_i(\eta_{\text{cad}}) \right).
\]

(29)

A similar approach is employed to obtain the optimal \( \mathcal{Z}_{\theta_1}^n \) during the training process

\[
\mathcal{Z}_{\theta_1}^n = \operatorname{arg} \max_{\mathcal{Z}_{\theta_1}^n} \left( \begin{array}{c}
H(\mathcal{Z}_{\theta_1}^1) \\
\vdots \\
H(\mathcal{Z}_{\theta_1}^n)
\end{array} \right)
\]

(30)

where \( \mathcal{Z}_{\theta_1}^n \) is uniformly sampled from the region \([-\pi, \pi]\). \( H(\cdot) \) is the fitness function of sampled LOS angle that is defined as follows:

\[
H(\mathcal{Z}_{\theta_1}^i) = F_{\text{los}}(\theta_{\text{init}}) |_{\theta_{\text{init}}=\mathcal{Z}_{\theta_1}^i}
\]

(31)
where $\theta_{\text{init}} = [\theta_{i,\text{init}}]$ is the joint initial individual parameters. In this way, the desired impact angles are established automatically.

A rank-based fitness shaping method that is in the same spirit as the one proposed in [22] is employed in shaping the raw fitness. Conventionally, we still let $F_i(\cdot)$ denote the fitness function after shaping. Another technique called mirrored sampling [20] is also applied for sampling parameter perturbations.

**IV. HYBRID COEVOLUTIONARY COOPERATIVE GUIDANCE ALGORITHM**

To achieve coordinated attack, the NCES is applied to optimize the parameter matrices $\theta_i = [W_{i1}, W_{i2}, W_{i1}]$ of the neural network controller.

The univariate Gaussian distribution with zero means and standard deviation $\sigma$ is used to sample perturbations. According to (24), it can be obtained that

$$g_{\theta_i} = \mathbb{E}_{\epsilon \sim \mathcal{N}(0, \sigma^2)} \left\{ \nabla_u \log p(\theta_i') F_i(\epsilon'_i) \prod_{c \in \mathcal{N}_i} p(\epsilon'_c) \right\} = \frac{1}{m \sigma^2} \sum_{i=1}^{m} F_i(\epsilon'_i) \epsilon_i \prod_{c \in \mathcal{N}_i} p(\epsilon_c).$$

The complete implementation algorithm of the proposed guidance law is shown in Algorithm 1. The conceptual diagram in Fig. 5 figuratively revealed the parallel simulation process. A parent/child (or fully distributed) model [37], [38] is used for large-scale parallel computation. In this case, each population is evaluated in a separate process, and the results of the ecosystem are aggregated to calculate the rescaled gradient (32) and sent to produce guided generations. The sampled generations are then distributed to each parallel process, and the gradient is recalculated and updated.

**Theorem 1:** Under the control policy (14) and the update strategy shown in Algorithm 1, by selecting an appropriate learning rate, sampling variance, and population size, the obtained control policy will converge to a small neighborhood of the optimal control policy when $\ell \to \infty$.

**Proof:** The approximation error of the control policy at the $\ell$th iteration is defined by

$$E_{ui}^{[\ell]} = u_i^{[\ell]} - u_i^*$$

where $u_i^*$ is the optimal control policy of the $i$th agent, and we have

$$u_i^{[0]} = (1 - \eta)u_{pi} + \eta u_i^{[\ell]}. \quad (34)$$

The control policy of the neural network is represented by its parameter set. Since a neural network with a single hidden layer can approximate a multivariate continuous function with arbitrary precision [39], which implies that a single hidden layer perceptron with sufficient units is equivalent to the neural network with three hidden layers used in this work. In this way, the neural network controller can be represented by

$$u_i^{[\ell]} = W_i^{[\ell]T}$$

which is a column matrix and the activation function parameters are regarded as constants. By combining (35) and (34) and substituting it into (33), we have

$$E_{ui}^{[\ell]} = (1 - \eta)u_{pi} + \eta W_i^{[\ell]T} - u_i^* \quad (36)$$

and

$$E_{ui}^{[\ell-1]} = (1 - \eta)u_{pi} + \eta W_i^{[\ell-1]T} - u_i^*. \quad (37)$$

Further combining (36) and (37), the term of fixed controllers are eliminated, and we have

$$E_{ui}^{[\ell]} - E_{ui}^{[\ell-1]} = \eta \left(W_i^{[\ell]T} - W_i^{[\ell-1]T}\right) \quad (38)$$

Algorithm 1: Hybrid Cooperative Coevolutionary Guidance Law (HCCGL)

**Require:** $\eta_{\alpha}, \eta_{\beta}, \sigma, \theta_{\text{init}} = [\theta_{i,\text{init}}]$, agent number $n$.

**Sample** $\Xi_k \in \mathbb{R}^k \sim U(-\pi, \pi)$, obtain $\Xi_k$ using (30)

**repeat**

**for** $k = 1... m$ do

Sample group of individuals:

$$\epsilon_i^k = \mathcal{N}(0, \sigma^2) : i \in \{1, ..., n\}$$

$$\epsilon_i^k = \{\epsilon_i^k, \epsilon_j^k : j \in \mathcal{N}_i\} : i \in \{1, ..., n\}$$

**evaluate fitness** $F_i(\epsilon_i^k)$, for $i \in \{1, ..., n\}$

**end for**

**for each agent** $i = 1... n$ do

**calculate natural gradient:**

$$g_\theta_i \leftarrow \frac{1}{m \sigma^2} \sum_{k=1}^{m} F_i(\epsilon_i^k) \prod_{c \in \mathcal{N}_i} p(\epsilon_c^k)$$

$$\theta_i \leftarrow \theta_i + \eta_{\alpha} \cdot g_\theta_i$$

**end for**

**if** time for adaptation **then**

sample $\gamma_{cad}$ using (27)

$$\gamma_{\alpha} \leftarrow \arg \max_{\gamma_{cad}} \left(\sum_{i=1}^{n} G_i(\gamma_{cad})\right)$$

**end if**

until stopping criterion is met

**Fig. 5.** Conceptual framework of our proposed HCCGL, the upper box connected by dotted lines is a detailed expansion of the evaluation and evolutionary processes in the lower box.
where \( \eta_0 \) is the learning rate and \( \eta \) is the guidance gain, with \( \eta_0, \eta > 0 \). The focus of this equation, \( \delta_{\theta_0}^{l-1} \), is the policy update gradient at the \( l \)-th iteration, which is followed by (32). Expanding this equation, we have

\[
E_{ui}^{[l]} - E_{ui}^{[l-1]} = \eta_0 \delta_{\theta_0}^{l-1} = \eta_0 \frac{1}{m \sigma^2} \sum_{k=1}^{m} F_i \left( E_{ui}^{[k]} \right) \epsilon_i \prod_{c \in N_i} p(c) \\
= \eta_0 \frac{1}{m \sigma^2} \sum_{k=1}^{m} F_i \left( E_{ui}^{[k]} \right) \left( E_{ui}^{[k]} - E_{ui}^{[l-1]} \right) \\
\times \prod_{c \in N_i} p(c) .
\]  

(39)

Note that in (39), \( F_i \left( E_{ui}^{[k]} \right) : \mathbb{R}^p \to \mathbb{R}^p \) is the transformed fitness function for the evaluation of the policy error, with \( p \) as the number of parameters. Thus, it is different from the fitness function discussed in the previous sections, which evaluates the policy directly. It is assumed that \( F_i \left( E_{ui}^{[k]} \right) \) is fully differentiable to the policy controller, and considering that \( |E_{ui}^{[k]}| \) represents the quality of the policy globally.

In an effort to linearize the fitness evaluation function, Taylor’s formula is utilized to expand the equation at \( |E_{ui}^{[l-1]}| \) and the higher order terms are ignored, and we obtain

\[
E_{ui}^{[k]} = G_i^{[l-1]} \left( |E_{ui}^{[l-1]}| - |E_{ui}^{[l-1]}| \right) + F_i \left( |E_{ui}^{[l-1]}| \right) 
\]

(41)

where \( G_i^{[l-1]} \in \mathbb{R}^{p \times p} \) is a diagonal Jacobian matrix defined by

\[
G_i^{[l-1]} = \frac{\partial F_i \left( |E_{ui}^{[k]}| \right)}{\partial |E_{ui}^{[k]}|} \biggr|_{|E_{ui}^{[l-1]}|=0}
\]

(42)

with negative entries and \( p \) as the number of total parameters. Since \( |E_{ui}^{[k]}| \) is located within a tiny vicinity of \( |E_{ui}^{[l-1]}| \), (41) is of considerable accuracy.

Then, by taking the absolute value of the approximation error and substituting (41) into (39), and considering \( \Delta E_i^k = |E_{ui}^{[k]}| - |E_{ui}^{[l-1]}| \), we obtain

\[
\Delta E_{ui}^{[l]} = E_{ui}^{[k]} - E_{ui}^{[l-1]} = \eta_0 \frac{1}{m \sigma^2} \sum_{k=1}^{m} G_i^{[l-1]} \Delta E_i^k + F_i \left( |E_{ui}^{[l-1]}| \right) \\
\times \Delta E_i^k \prod_{c \in N_i} p(c) \\
= \eta_0 \frac{1}{m \sigma^2} G_i^{[l-1]} \\
\times \sum_{k=1}^{m} \left[ \Delta E_i^k \prod_{c \in N_i} p(c) \right] \\
\times \left( |E_{ui}^{[k]}| - |E_{ui}^{[l-1]}| \right) \\
\times \prod_{c \in N_i} p(c).
\]

(43)

For brevity, we define \( A_i^k, P_i^k \), and \( B_i^k \) by

\[
A_i^k = \Delta E_i^k \prod_{c \in N_i} p(c) \\
P_i^k = \sum_{k=1}^{m} \left[ \prod_{c \in N_i} p(c) \right] \\
B_i^k = \sum_{k=1}^{m} \left( |E_{ui}^{[l]}| - \prod_{c \in N_i} p(c) \right)
\]

(44)

such that

\[
\Delta E_{ui}^{[l]} = \eta_0 \frac{1}{m \sigma^2} \left[ \sum_{k=1}^{m} A_i^k \prod_{c \in N_i} p(c) + F_i \left( |E_{ui}^{[l-1]}| \right) \prod_{c \in N_i} p(c) \right].
\]

(45)

Since \( |E_{ui}^{[k]}| \) is sampled from an unbiased normal distribution which is centered at \( |E_{ui}^{[l-1]}| \), as shown in the analysis of Section III-C, we have

\[
B_i^k \to 0, \quad m \to \infty.
\]

(46)

Also, from the matrix Hadamard product, we have

\[
A_i^k > 0
\]

(47)

and

\[
P_i^k > 0.
\]

(48)

\( G_i^{[l-1]} \) is negative definite. Given sufficient large \( m \), it is evident that

\[
\Delta E_{ui}^{[l]} < 0, \quad l = 1, 2, \ldots
\]

(49)

Therefore, by adjusting the learning rate \( \eta_0 \) attentively, the approximation error can be decreased to a considerably small range \( \delta_e \), such that

\[
\lim_{l \to \infty} E_{ui}^{[l]} = \delta_e
\]

\[
\delta_e \to 0, \quad m \to \infty.
\]

(50)

Thus, the control policy \( u_i \) converges to a small neighborhood of the optimal control policy \( u_i^* \), resulting in a stabilizing control system.

\[\square\]

V. Simulations and Analysis

To verify the validity of the proposed method, a variety of simulations based on the cooperative guidance framework are designed. Both cases with stationary target and maneuvering target are simulated. Furthermore, comparison experiments are performed to fully demonstrate the superiority of the proposed guidance method.
TABLE III
CONSTRAINTS OF THE MISSILES

| Parameter                      | Value |
|-------------------------------|-------|
| maximum lateral overload (g)  | 50    |
| maximum trust overload (g)    | 5     |
| upper bound of velocity (m/s)| 900   |
| lower bound of velocity (m/s)| 350   |

TABLE IV
HYPERPARAMETERS OF THE COOPERATIVE GUIDANCE ALGORITHM

| Parameter                      | Value |
|-------------------------------|-------|
| simulation step (ms), τ       | 5     |
| guidance gain, η              | 0.3   |
| Initial learning rate, η₀     | 0.015 |
| standard deviation for sampling population, σ | 0.2 |
| size of learning rate adaptation, l | 20 |
| size of population, m         | 140   |
| adaptation cycle, ρ           | 50    |
| navigation constant, β        | 4     |
| kₐ                           | 1     |
| kₜ                           | 0.2   |
| ξ₀                           | 10    |
| ξₜ                           | 1     |
| λₐ                           | 4000  |
| λₜ                           | 2000  |
| βₐ                           | 10    |
| βₜ                           | 2     |

A. Parameter Setup

The acceleration constraint and velocity constraint of the missiles are listed in Table III. The hyperparameters of the algorithm are listed in Table IV.

Now that frameskip has been extensively employed in continuous control problems [21]. In this work, this parameter of frameskip is set to 12 for cases 1 and 2, and 40 for case 3. Appropriate adjustment of this parameter will facilitate the training process without affecting the final results.

B. Case 1: Comparison Experiments

In this section, the proposed guidance law is compared with the time and space cooperative guidance law (TASCGL) proposed in [40], which considers the space and time cooperative guidance under the distributed communication topology. However, different from the method proposed in this work, the compared method is susceptible and brittle to the initial conditions. Therefore, in order to verify the generalization ability of the control methods, a uniform initial condition was adopted in the comparison simulation, which differs slightly from the initial condition in the comparison method. The initial conditions are shown in Table V. Four missiles are engaged in the cooperative scenario with different desired relative impact angles δᵢ as 20°, 60°, and 30°, for each i = 1, 2, 3, respectively. The target is located at (9500, 9000) m.

Although the reference method is primarily designed for directed topology, it can be well extended to an undirected topology condition, thus in order to conduct effective comparison experiments, we conducted comparison experiments under both directed and undirected communication topologies, and the directed topology was adopted to be the same as the one used in the compared method. We use TASCGLᵃ and TASCGLᵇ to denote the comparative experiments performed under undirected and directed communication topologies, respectively, similar to HCCGLᵃ and HCCGLᵇ.

Fig. 6 shows the trajectories of the two guidance laws. As depicted in the figure, the trajectory of TASCGL is twisted at the initial stage, as the missiles try to consensus their LOS angles and velocities. In comparison, the trajectory of the proposed HCCGL exhibited better damping performance with no oscillations.

It can be seen from Table VI that the zero-effort miss (ZEM) and the consensus angle error for both guidance laws have achieved competitive final accuracy. The consensus time error of TASCGL was up to 5 s under both directed and undirected topologies, whereas the proposed method achieved an error of less than 0.1 s, under both directed and undirected topologies. Further analysis of the velocity curve shows that in the case of TASCGL, the velocities are prohibited from reaching their ideal values due to the velocity boundary, which is not considered in its design, thus leading to desynchronization in impact time. The profiles of the two methods are shown in Figs. 7 and 8; it can be observed that the flight time of all missiles under HCCGL tends to be identical. For HCCGL under directed topology, the decomposition of acceleration commands is shown in Fig. 9. The left figure shows the decomposition of lateral accelerations, in which the solid line represents the command from the tracking controller while the dashed line represents the command from the consensus controller before weighing. Since the tracking part is derived from proportional navigation, the vertical acceleration shown on the right one is completely derived from the consensus controller. The two parts of accelerations have similar trends but do not coincide, demonstrating the effectiveness of the consensus controller, which is trained with the improved co-ES.

The result reveals that the proposed guidance law outperforms the compared method with higher precision in consensus performance and smoother trajectories. Moreover, as the traditional guidance law is usually constrained to boundary conditions and missile’s superb maneuverability, the proposed guidance law is more resilient to limited conditions and more intelligent to be aware of the time-varying states of missiles of collaboration.

C. Case 2: Nonstationary Target

In this part, an engagement scenario with a nonstationary target is designed and simulated to verify the effectiveness of the proposed method against unknown dynamic target.
The target is maneuvering with lateral acceleration $\alpha_t = 5g \sin(\pi/7t)$ with its velocity fixed at $V_t = 130$ m/s, and its initial flight-path angle $\alpha_T = 162^\circ$. Other initial conditions are the same in case 1. Simulation trajectory and the result can be seen in Fig. 10 and Table VII.

From Table VII, we can see that the consensus angle error is within one degree, which is sufficient for the accuracy requirement, and salvo attack is achieved with negligible consensus time error. The result demonstrates the effectiveness of the proposed guidance method in intercepting the dynamic target. As far as the author knows, it is the first time achieving cooperative guidance against nonstationary target with intelligent control, which shows its extraordinary robustness against disturbance from nonstationary objectives.

D. Case 3: Monte Carlo Simulation

Monte Carlo simulation has been extensively employed to examine the robustness of an algorithm under varying initial conditions; thus, it is applied in this section. In the existing literature, the target is usually regarded as stationary as interception of a stationary target is more exclusive of unpredictable disturbance. In this case, five missiles are engaged, and each missile’s position is randomly sampled from a uniform distribution, which is denoted by $U(\cdot, \cdot)$. Specifically, for the $i$th missile, the $x$-coordinate of its position is $U(2000, 2600)$ and the $y$-coordinate is $U(11000, 13000) - 2000i$, which makes the missiles arranged in an orderly manner. The initial flight-path angles of all missiles are set to...
Fig. 7. Consensus angle error profiles of the two methods. (a) TASCGL\(^a\). (b) TASCGL\(^b\). (c) HCCGL\(^a\). (d) HCCGL\(^b\).

Simulations with randomly sampled conditions are conducted in 200 episodes. The diverse trajectories are depicted in Fig. 12, and the statistical result after taking the absolute value is shown in Table VIII. From the result, we can see 0\(^\circ\), with identical velocities of 600 m/s and the same desired relative impact angles of 25\(^\circ\). Additionally, the target’s position is (10000 m, 9000 m).
that the mean errors of impact angles are within 1°, and the consensus error of impact time holds within 1 s most of the time. The result shows that for any initial state with limited error, the proposed scheme can always find the relative optimal solution.

E. Optimization Process Analysis

Fig. 13 shows the learning curves in the three cases. The mean fitness in case 1 keeps moving upper and merges together at the final phase. From the curve of case 2, we can see that two of the missiles get ahead about 1000 scores, but finally back to meet with the other missiles. A similar phenomenon also appears in case 3. It can be inferred that the policies asymptotically evolved to the equilibrium state, and one reason is that the rescaled gradient prohibited the ever-increasing gap between individual groups, which is crucial for mutual improvement. If one group gets ahead too much, then the other groups may never chase up due to the interrelationship, which is to say that the improvement of the poorly performed group is prohibited when more significant drops in the better-performed ones will occur. Fig. 14 presents the adaptation profiles of learning rates applying the aforementioned technique. For cases 1 and 2, the learning rates start from high values and gradually converge to the minimal value, which corresponds with the quality of estimated gradients. However, due to the random initial conditions in case 3, the learning rates will not settle easily. The extensive empirical result shows that without the learning rate adaptation, the fitness profiles will jitter in the end instead of converging to satisfactory ranges (regardless of the types of optimizer). Note that it is pretty common
when training neural networks and may presumably have been caused by overfitting, according to related research in the field. Employing the simple adaptation technique contributes to distressing this deficiency.
TABLE VIII
RESULT FOR CASE 3

| Index  | $e_1^t$ | $e_2^t$ | $\epsilon^t$ |
|--------|---------|---------|-------------|
| $e_1^t$ | Mean  | 450E-1 | 820E-1 | 710E-1 | 2200E-1 | 930E-1 |
|        | Max    | 185E-0 | 373E-0 | 196E-0 | 610E-1 | 2370E-0 |
|        | Min    | 456E-3 | 640E-3 | 701E-4 | 458E-4 | 646E-3 |
| $e_2^t$ | Mean  | 616E-1 | 550E-1 | 530E-1 | 450E-1 | 530E-1 |
|        | Max    | 178E-0 | 157E-0 | 163E-0 | 154E-0 | 144E-0 |
|        | Min    | 150E-2 | 100E-2 | 178E-15 | 500E-3 | 178E-15 |
| $\epsilon^t$ | Mean  | 585E-3 | 589E-3 | 734E-4 | 905E-4 | 939E-4 |
|        | Max    | 103E-2 | 102E-2 | 777E-3 | 233E-3 | 389E-3 |
|        | Min    | 218E-3 | 204E-3 | 642E-5 | 168E-5 | 272E-6 |

VI. CONCLUSION

In this article, an improved co-ES NCES has been developed to solve the nonstationarity issue in multiantigen dynamic environments. The HCCGL has been proposed to integrate with the improved strategy, and the neural network has been used to construct the consensus controller. To fully demonstrate its effectiveness in synchronizing impact time and angles, three experiments under different conditions have been carried out. Experiment on maneuvering target has been proven effective with satisfactory precision. The proposed method is shown to be robust and can be well scaled to solve the cooperative guidance problem for the multiagent system, which is the first time an intelligent cooperative guidance law is applied to intercept a nonstationary target with time and angle constraints in the existing studies.

The proposed algorithm combines traditional control theories with intelligent algorithms, revealing the enormous potential in this field. It is always meaningful to explore the limits of modern control tasks. Despite the satisfactory results that have been acquired, this work still left space to be improved. Future works may include exploring the effectiveness of incremental guidance gain, or control strategies that tackle actuation failure and system uncertainty.

REFERENCES

[1] I.-S. Jeon, J.-I. Lee, and M.-J. Tahk, “Homing guidance law for cooperative attack of multiple missiles,” J. Guid., Control, Dyn., vol. 33, no. 1, pp. 275–280, Jan. 2010.
[2] K. Ma, H. K. Khalil, and Y. Yao, “Guidance law implementation with performance recovery using an extended high-gain observer,” Aerosp. Sci. Technol., vol. 24, no. 1, pp. 177–186, Jan. 2013.
[3] S. Xiong, M. Wei, M. Zhao, H. Xiong, W. Wang, and B. Zhou, “Hyperbolic tangent function weighted optimal intercept angle guidance law,” Aerosp. Sci. Technol., vol. 78, pp. 604–619, Jul. 2018.
[4] Z. Li and Z. Ding, “Robust cooperative guidance law for simultaneous arrival,” IEEE Trans. Control Syst. Technol., vol. 27, no. 3, pp. 1360–1367, May 2019.
[5] S. He, H.-S. Shin, and A. Tsourdos, “Computational missile guidance: A deep reinforcement learning approach,” J. Aerosp. Inf. Syst., vol. 18, no. 8, pp. 571–582, Aug. 2021.
[6] A. Rattoo and D. Ghose, “Impact angle constrained interception of stationary targets,” J. Guid., Control, Dyn., vol. 31, no. 6, pp. 1817–1822, Nov. 2008.
[7] X. Lan, W. Xu, Z. Zhao, and G. Liu, “Autonomous control strategy of a swarm system under attack based on projected view and light transmittance,” IEEE/CAA J. Auton. Syst., vol. 8, no. 3, pp. 648–655, Mar. 2021.
[8] I.-S. Jeon, J.-I. Lee, and M.-J. Tahk, “Impact-time-control guidance law for anti-ship missiles,” IEEE Trans. Control Syst. Technol., vol. 14, no. 2, pp. 260–266, Mar. 2006.
[9] B. Gautret, R. Furfaro, and R. Linares, “Reinforcement learning for angle-only intercept guidance of maneuvering targets,” Aerosp. Sci. Technol., vol. 99, Apr. 2020, Art. no. 105746.
[10] H. M. La, R. Lim, and W. Sheng, “Multirobot cooperative learning for predator avoidance,” IEEE Trans. Control Syst. Technol., vol. 23, no. 1, pp. 52–63, Jan. 2015.
[11] H. Dong and X. Zhao, “Composite experience replay-based deep reinforcement learning with application in wind farm control,” IEEE Trans. Control Syst. Technol., vol. 30, no. 3, pp. 1281–1295, May 2022.
[12] W. Kong, D. Zhou, Z. Yang, K. Zhang, and L. Zeng, “ Maneuver strategy generation of UAV for with VSA range air combat based on multi-agent reinforcement learning and target position prediction,” Appl. Sci., vol. 10, no. 15, p. 5198, Jul. 2020.
[13] B. M. Alabba and Y. Yildiz, “Driver modeling through deep reinforcement learning and behavioral game theory,” IEEE Trans. Control Syst. Technol., vol. 30, no. 2, pp. 885–919, Mar. 2022.
[14] T. Chen, K. Zhang, G. B. Giannakis, and T. Basar, “Communication-efficient policy gradient methods for distributed reinforcement learning,” IEEE Trans. Control Netw. Syst., vol. 9, no. 2, pp. 917–929, Jun. 2022.
[15] X. Lan, Y. Liu, and Z. Zhao, “Cooperative control for swarming systems based on reinforcement learning in unknown dynamic environment,” Neurocomputing, vol. 410, pp. 410–418, Oct. 2020.
[16] H. Dong, X. Zhao, and H. Yang, “Reinforcement learning-based approximal control strategy for attitude reorientation under state constraints,” IEEE Trans. Control Syst. Technol., vol. 29, no. 4, pp. 1664–1673, Jul. 2021.
[17] L. Chen, W. W. Hong, and L. Chao, “Deep reinforcement meta-learning guidance with impact angle constraint,” J. Astronaut., vol. 42, no. 5, pp. 611–620, May 2021.
[18] H. Zhang, H. Jiang, Y. Luo, and G. Xiao, “Data-driven optimal consensus control for discrete-time multi-agent systems with unknown dynamics using reinforcement learning method,” IEEE Trans. Ind. Electron., vol. 64, no. 5, pp. 4091–4100, May 2017.
[19] T. T. Nguyen, N. D. Nguyen, and S. Nahavandi, “Deep reinforcement learning for multiagent systems: A review of challenges, solutions, and applications,” IEEE Trans. Cybern., vol. 50, no. 9, pp. 3826–3839, Sep. 2020.
[20] D. Brockhoff, A. Auger, N. Hansen, D. V. Arnold, and T. Hohn, “Mirrored sampling and sequential selection for evolution strategies,” in Proc. Int. Conf. Parallel Problem Solving From Nature, Berlin, Germany: Springer, 2010, pp. 21–29.
[21] T. Salimans, J. Ho, X. Chen, S. Sidor, and I. Sutskever, “Evolution strategies as a scalable alternative to reinforcement learning,” 2017, arXiv:1703.03864.
[22] D. Wierstra, T. Schaul, T. Glasmachers, Y. Sun, J. Peters, and J. Schmidhuber, “Natural evolution strategies,” J. Mach. Learn. Res., vol. 15, no. 1, pp. 949–980, Mar. 2014.
[23] B. Xu, Y. Zhang, D. Gong, Y. Guo, and M. Rong, “Environment sensitivity-based cooperative co-evolutionary algorithms for dynamic multi-objective optimization,” IEEE/ACM Trans. Comput. Biol. Bioinf., vol. 15, no. 6, pp. 1877–1890, Nov. 2018.
[24] H. Qu, K. Xing, and T. Alexander, “An improved genetic algorithm with co-evolutionary strategy for global path planning of multiple mobile robots,” Neurocomputing, vol. 120, pp. 509–517, Nov. 2013.
[25] Z. Wang, C. Chen, and D. Dong, “Instance weighted incremental evolution strategies for reinforcement learning in dynamic environments,” IEEE Trans. Neural Netw. Learn. Syst., early access, Mar. 29, 2022, doi: 10.1109/TNNLS.2022.3160173.
[26] P. Larrañaga and J. A. Lozano, Estimation of Distribution Algorithms: A New Tool for Evolutionary Computation, vol. 2, Springer, 2001. [Online]. Available: link.springer.com/book/10.1007/978-1-4615-1539-5
[27] H. Karshenas, R. Santana, C. Bielez, and P. Larrañaga, “Regularized continuous estimation of distribution algorithms,” Appl. Soft Comput., vol. 13, no. 3, pp. 2412–2432, May 2013.
[28] T. E. Marquardt and M. A. Chichio, “Multiobjective evolutionary algorithm for designing fuzzy-based missile guidance laws,” J. Aerosp. Eng., vol. 24, no. 1, pp. 89–94, Jan. 2011.
[29] N. Maheswaranathan, L. Metz, G. Tucker, D. Choi, and J. Sohl-Dickstein, “Guided evolutionary strategies: Augmenting random search with surrogate gradients,” in Proc. Int. Conf. Mach. Learn., 2019, pp. 4264–4273.
[30] J. Del Ser et al., “Bio-inspired computation: Where we stand and what’s next,” Swarm Evol. Comput., vol. 48, pp. 220–250, Aug. 2019.
[31] R. Gray, A. Franci, V. Srivastava, and N. E. Leonard, “Multiagent decision-making dynamics inspired by honeybees,” IEEE Trans. Control Netw. Syst., vol. 5, no. 2, pp. 793–806, Jun. 2018.

[32] B. Gaudet, R. Furfaro, and R. Linares, “Reinforcement learning for angle-only intercept guidance of maneuvering targets,” Aerosp. Sci. Technol., vol. 99, Apr. 2020, Art. no. 105746.

[33] D. Han and S. N. Balakrishnan, “State-constrained agile missile control with adaptive-critic-based neural networks,” IEEE Trans. Control Syst. Technol., vol. 10, no. 4, pp. 481–489, Jul. 2002.

[34] H. Li, Y. Wu, and M. Chen, “Adaptive fault-tolerant tracking control for discrete-time multiagent systems via reinforcement learning algorithm,” IEEE Trans. Cybern., vol. 51, no. 3, pp. 1163–1174, Mar. 2021.

[35] B. Thananjeyan et al., “Recovery RL: Safe reinforcement learning with learned recovery zones,” IEEE Robot. Autom. Lett., vol. 6, no. 3, pp. 4915–4922, Jul. 2021.

[36] K. Son, D. Kim, W. J. Kang, D. E. Hostallero, and Y. Yi, “QTRAN: Learning to factorize with transformation for cooperative multi-agent reinforcement learning,” in Proc. Int. Conf. Mach. Learn., 2019, pp. 5887–5896.

[37] Y.-J. Gong et al., “Distributed evolutionary algorithms and their models: A survey of the state-of-the-art,” Appl. Soft Comput., vol. 34, pp. 286–300, Sep. 2015.

[38] A. Mendiburu, J. A. Lozano, and J. Miguel-Alonso, “Parallel implementation of EDAs based on probabilistic graphical models,” IEEE Trans. Evol. Comput., vol. 9, no. 4, pp. 406–423, Aug. 2005.

[39] S. Trenn, “Multilayer perceptrons: Approximation order and necessary number of hidden units,” IEEE Trans. Neural Netw., vol. 19, no. 5, pp. 836–844, May 2008.

[40] T. Lyu, Y. Guo, C. Li, G. Ma, and H. Zhang, “Multiple missiles cooperative guidance with simultaneous attack requirement under directed topologies,” Aerosp. Sci. Technol., vol. 89, pp. 100–110, Jun. 2019.

Junda Chen received the B.Eng. degree in robotic engineering from Guangzhou University, Guangzhou, China, in 2023. He is currently a Software Engineer with Computer Vision Company. His research interests include multimodal control, dynamic optimization, and machine vision.

Zhijia Zhao (Member, IEEE) received the B.Eng. degree in automatic control from the North China University of Water Resources and Electric Power, Zhengzhou, China, in 2010, and the M.Eng. and Ph.D. degrees in automatic control from the South China University of Technology, Guangzhou, China, in 2013 and 2017, respectively. He is currently an Associate Professor with the School of Mechanical and Electrical Engineering, Guangzhou University, Guangzhou. His research interests include flexible systems, adaptive and learning control, and intelligent control.

Authorized licensed use limited to the terms of the applicable license agreement with IEEE. Restrictions apply.