IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received December 21, 2019, accepted December 31, 2019, date of publication January 6, 2020, date of current version January 15, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2964031

Learning System for Air Combat Decision Inspired by Cognitive Mechanisms of the Brain

KAI ZHOU[1], RUIXUAN WEI[1], QIRUI ZHANG[1], AND ZHUOFAN XU[2]
[1]Aeronautics Engineering College, Air Force Engineering University, Xi'an 710038, China
[2]Joint Operations College, National Defence University of People's Liberation Army, Shijiazhuang 050084, China

Corresponding author: Kai Zhou (kaizhouu@gmail.com)

This work was supported in part by the National Natural Science Foundation of China under Grant 61573373.

ABSTRACT Unmanned aerial vehicles (UAVs) have played an important role in recent high-tech local wars. Seizing air control rights with UAVs will undoubtedly be a popular topic in future military development. Autonomous air combat is complex, antagonistic and mutable, and consequently, the decision-making that depends on unmanned systems is extremely challenging with very little research having been conducted on it. An intelligent air combat learning system inspired by the learning mechanisms of the brain is proposed in this paper. In accordance with research on learning, knowledge and memory, we constructed a cognitive mechanism model of the brain. Based on this model and the inferential abilities of humans, a long short-term hierarchical multi-line learning system is established. Then, the bio-inspired architecture and the basic learning principle of the system are clarified. Taking advantage of the conclusions of studies on information theory, the relationship between the knowledge updating cycle and the system learning performance is analysed. The updating cycle length adjustment problem is transformed into an optimization problem optimization problem, and system performance improvement is guaranteed. Experiments show that the system designed in this paper can acquire confrontation abilities through self-learning without prior rules; the parallel universe mechanism can significantly improve the system's learning speed when the number of parallels is within 40, and the performance of the system improves gradually and continuously. The system can master actions similar to classical tactical manoeuvres such as the high yo-yo and the barrel-roll-attack without prior knowledge. Compared with the Bayesian inference and moving horizon optimization (BI&MHO) method, the learning system proposed in this paper is more flexible in situation assessment and in the prediction of opponents' actions. Although it cannot be deployed quickly, it has a continuous learning ability.

INDEX TERMS Autonomous air combat, bio-inspired, cognitive mechanism, long short-term memory, learning system, unmanned aerial vehicles.

I. INTRODUCTION

Unmanned aerial vehicles (UAVs) have emerged as a new force in recent high-tech local wars. The interweaving of ground and air firepower will greatly threaten the survival of pilots and fighter aircraft in modern warfare. In future wars, if UAVs are used to achieve air supremacy, it will undoubtedly result in another profound military revolution.

Despite the increasing importance of stand-off strikes in modern air operations, state-of-the-art fighters such as the F22 and F35 are still equipped with guns for close combat. Moreover, future UAVs may have stronger stealth capability and smaller size, and we may need to deal with enemy aircraft suddenly appearing at close range in future battlefields. The US Defence Advanced Research Projects Agency (DARPA) is seeking proposals to automate air-to-air combat as part of its Air Combat Evolution (ACE) programme. The ACE programme is intended to exploit developments in artificial intelligence (AI) to enable the automation of within-visual-range air-to-air combat and bring UAVs to the dogfight [1]. Therefore, study on autonomous confrontation within the visual range is crucial for future UAVs.

Virtanen et al. used an influence diagram to model the air combat process and combined an autonomous inference

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

ability and human pilot experience in [2]; extended methods to acquire better goals were introduced in [3], [4]. Zhong *et al.* [5] took into account the decision maker's preferences under uncertain conditions and considered an active opponent; they solved the multistage influence diagram by converting it into a two-level optimization problem.

Game theory [6]–[8], the moving horizon optimization method [9] and the trial input method [10], [11] were introduced to explain and solve this problem. The general ideas of these methods are similar. They begin by using factors such as sight angle, distance, height advantage, velocity advantage, the weapon engagement zone and the kill probability [12] in a prediction time domain to build an objective function. Then, they find the optimal decision in the feasible region by online methods. McGrew [13], [14] and MA [15] proposed an approximate dynamic programming method [16], [17], the dimension explosion of traditional dynamic programming has been improved. In particular, they constructed the objective function through iterative learning. This idea provided us with inspiration.

There are also artificial intelligence methods such as expert systems [18], the artificial immune system method [19] and others. These methods usually work by establishing a manoeuvre database using basic fighter manoeuvres (BFM) or elementary manoeuvres [9] so that decisions can be made quickly. The database can be extended manually or selected by an immunity rule. Ernest et al. presented a genetic fuzzy tree (GFT) method [20]–[22] utilizing a collection of fuzzy inference systems (FIS). By breaking up the problem into many sub-decisions, the solution space is significantly reduced. The in-development simulation environment ALPHA was highly praised by a colonel of the air force who has been an opponent of ALPHA.

To address complex multi-step decision-making problems, researchers have attempted to find methods with higher levels of intelligence. Taking into account the complex dynamics of UAVs, Emel'Yanov *et al.* [23] proposed a cognitive architecture control system. It can solve a broad range of tasks and can raise the degree of autonomy of the control object significantly. Rollo *et al.* [24] built a modular architecture framework for complex unmanned aircraft systems. They tested the system in a cooperative collision avoidance task and achieved good results. Furthermore, the framework enables the study of further concepts such as additional payload and interaction among UAVs. Sanchez-Lopez *et al.* [25] presented an open-source software framework for the development of aerial robotic systems, which can provide higher degrees of autonomy and is more versatile in application to different types of hardware and different types of missions. Inspired by a biological model of the human cognitive system, a high-level processing approach for understanding human activities is proposed in [26] that allows the adaptation of the flight plan and fully autonomous surveillance in limited areas. Chithapuram *et al.* [27], [28] developed a new guidance scheme using Q-learning. The new guidance scheme performs better than standard existing guidance schemes in the presence of sensor noise and computational delays. Moreover, studies have been performed that aim to find solutions based on the structure and working principles of the human brain [29]–[31]. These explorations inspired us to solve the problem of air combat decision-making by imitating the cognitive mechanism of the brain.

We proposed a learning architecture that imitates the cognitive mechanism of the brain in our previous work [32]. The system can learn independently through simulated training. The training achievement is a mapping between situations and decisions. This paper is an extension of the previous one. The contributions of this paper are as follows:

1) A cognitive mechanism model, including multilevel memory and different knowledge content, describing how the brain learns and stores knowledge quickly from practise and interaction, is proposed in this paper. Applying this working principle to decision-making in autonomous air combat manoeuvres, we build an architecture that can learn by itself using interactive data. To our best knowledge, this is the first study to propose such a data-learning cognitive architecture.
2) Imitating the multi-line reasoning ability of humans, a simulation and data acquisition mechanism, which we call parallel universe, is designed. Increasing the number of parallel universes within a certain range can significantly improve learning efficiency.
3) Using relative entropy [33] (Section 2.6) to express the differences between the two policies, we analyse the relationship between the new and old policies under practical operation sampling conditions and prove that an appropriate length of the consolidation learning cycle (CLC) can ensure the stable increase in the learning performance. Transforming the CLC length adjustment problem into an optimization problem, a feasible method to guarantee learning performance improvement is given.

This paper is organized as follows. In Section II, we introduce some basic definitions. In Section III we introduce the biological study of learning, memory and knowledge, establish a model of the brain, and present the architecture of the learning system inspired by the cognitive mechanism of the brain. Next, the basic bio-inspired learning principle of the system is illustrated. Then, we add restrictions to the update of short-term procedural knowledge so that the performance of the system can be non-decreasing. Then, a detailed implementation of the method is presented. In Section IV, experiments are presented to illustrate the effectiveness of the proposed method. Conclusions are drawn in Section V.

II. PRELIMINARIES

First, some relevant definitions are given. We use an infinite-horizon Markov process to describe air combat confrontations. The elements are represented by the tuple $(\mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \pi, \gamma)$, where $\mathcal{S}$ is a finite situation space; $\mathcal{A}$ is a finite action space; $\mathcal{P} : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to \mathbb{R}$ is the

situation transition probability matrix; $\mathcal{R} : \mathcal{S} \rightarrow \mathbb{R}$ is the reward matrix; and $\pi : \mathcal{S} \times \mathcal{A} \sim \mathcal{N}\left(\mu, \sigma^2\right)$ denotes a stochastic action policy, which is a Gaussian distribution with mean μ and variance σ^2. $\gamma \in (0, 1)$ is the discount factor [34], the closer it is to 1, the higher proportion of the afterwards situation is. First, we define the situation-value and the situation-action-value in the form that is commonly used in reinforcement learning [35], [36].

Definition 1: The situation-value is the expected discounted reward in the rest of the Markov process:

$$V_\pi\left(s_t\right) = \mathbb{E}_{a_t, s_{t+1}, a_{t+1}, \ldots}\left[\sum_{l=0}^{\infty} \gamma^l r\left(s_{t+l}\right)\right] \tag{1}$$

where $a_t, a_{t+1} \ldots \in \mathcal{A} \sim \pi$, a_t denotes the action at time t, $s_{t+1}, \ldots \in \mathcal{S} \sim \mathcal{P}$, s_{t+1} is the situation at time $t+1$, $r\left(s_{t+l}\right) \in \mathcal{R}$, and $r\left(s_{t+l}\right)$ denotes the reward in the situation s_{t+l}.

Definition 2: The situation-action-value is the expected return of doing action a_t in situation s_t:

$$\begin{aligned} &Q_\pi\left(s_t, a_t\right) \\ &\quad = \mathbb{E}_{s_{t+1}, a_{t+1}, \ldots}\left[r\left(s_t, a_t\right) + \gamma\left[\sum_{l=0}^{\infty} \gamma^l r\left(s_{t+1+l}\right)\right]\right] \\ &\quad = \mathbb{E}_{s_{t+1}, a_{t+1}, \ldots}\left[r\left(s_t, a_t\right) + \gamma V_\pi\left(s_{t+1}\right)\right] \end{aligned} \tag{2}$$

In some situations, the actions do not have obvious influence on the situation-value or the situation-action-value. For example, when two aircraft are very far apart, most manoeuvres do not cause a significant change in the value of $V_\pi\left(s_t\right)$ or $Q_\pi\left(s_t, a_t\right)$. To make learning more sensitive in these situations, we separate the action-value and give the following definition.

Definition 3: The action-value is the expected profit of doing action a_t at time t:

$$\begin{aligned} &A_\pi\left(s_t, a_t\right) \\ &\quad = Q_\pi\left(s_t, a_t\right) - V_\pi\left(s_t\right) \\ &\quad = \mathbb{E}_{s_{t+1}, a_{t+1}, \ldots}\left[r\left(s_t, a_t\right) + \gamma V_\pi\left(s_{t+1}\right) - V_\pi\left(s_t\right)\right] \end{aligned} \tag{3}$$

Then, we can see that in a continuous Markov process, if we have $V_\pi\left(s_{t+1}\right)$ and $V_\pi\left(s_t\right)$, the action-value $A_\pi\left(s_t, a_t\right)$ can be obtained. A similar idea can be found in [37]; we expand it to a continuous problem, and we do not need neural networks with special structures for the calculation of A_π.

III. PROPOSED BRAIN-INSPIRED AIR COMBAT LEARNING SYSTEM

A. COGNITIVE MECHANISM OF THE BRAIN

Conventional methods seem to be less reliable and effective in dealing with complex decision-making problems. Some researchers have been trying to find answers from the brain. The Schultz team found that the error between the expected situation and the actual situation could activate midbrain dopamine neurons [38], which might be a motive force behind learning activity in the brain [39]. In recent years, additional studies have confirmed this view [40]–[43], which indicates that the distinction between cognition and real situations is one of the most important motivations for the brain to learn. These findings also provide a biological and neurological basis for reinforcement learning [44]–[46]. In our opinion, the learning process in the brain is not completely similar to reinforcement learning, so we look for other evidence in the study of biological neurology.

Learning achievement, that is, knowledge, is stored in short-term and long-term memory [47]–[49]. Short-term memory is a preliminary product of rapid learning; if the short-term knowledge can be proved to correctly represent the world in continuous learning, it will consolidate into long-term knowledge. Long-term memory is stable knowledge that cannot easily be changed by isolated experience. It is the type of memory we rely on to guide practice and to assess situations. Some studies suggest that long-term memory and short-term memory are stored in the same neural structure, differing in the method and extent of activation [47]. Moreover, there is another special type of memory in our brain called working memory [39], [47]. The working memory retains sequences of events temporarily. It stores external information obtained from the environment and internal information generated by the brain itself for some time during the learning process. To learn how to deal with a complex decision-making task, the brain divides knowledge into two types, declarative and procedural knowledge, and assigns them to different brain regions, so that complex learning tasks are simplified, and sub-tasks are allocated to different regions [47], [50]. By synthesizing the evidence from the study of the brain, a cognitive mechanism model of the brain in the decision-making learning process is established, as shown in Figure 1.

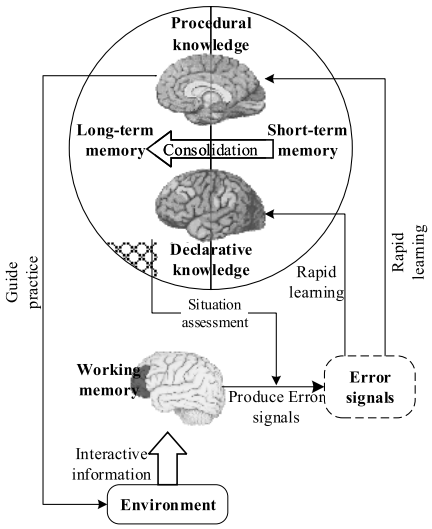


FIGURE 1. Cognitive mechanism of the brain during the decision-making learning process.

B. ARCHITECTURE OF THE LEARNING SYSTEM

The brain guides practice according to long-term knowledge, acts in the environment, stores the interactive information in

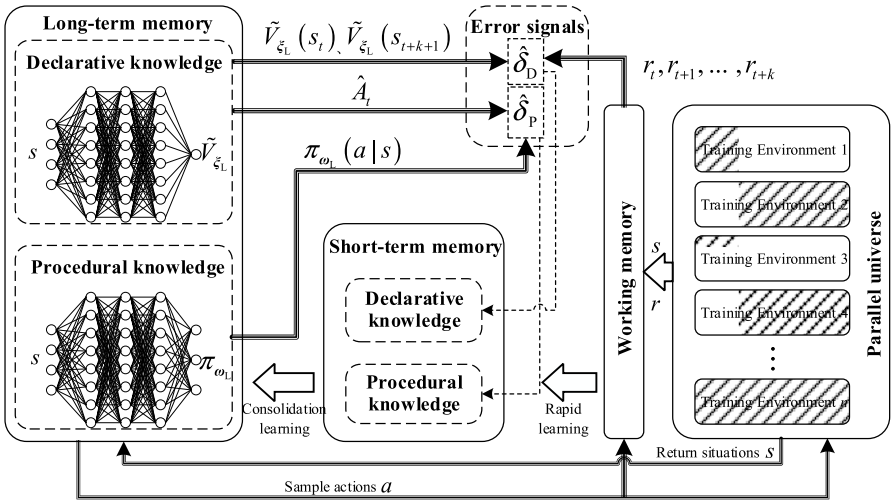


FIGURE 2. Architecture of the brain-inspired air combat learning system.

working memory, and then transforms the interactive information into error signals according to long-term knowledge. The error signal is converted into embryonic knowledge quickly and stored in short-term memory. As learning goes on, the short-term knowledge is consolidated gradually into stable knowledge, and the updated long-term knowledge will play a role in subsequent interactions and learning. This cognitive mechanism of the brain makes it possible for human beings to learn quickly from practice. The long- and short-term learning mechanism allows us to obtain knowledge quickly and to avoid the mutation of knowledge structure caused by low-probability events. Expecting the machine to have the ability to learn how to make decisions, we designed an air combat learning system by drawing lessons from the cognitive mechanism of the brain. The architecture of the system is shown in Figure 2.

Knowledge is divided into procedural and declarative parts. Pilots can clearly describe why a manoeuvre should be executed in a situation and what will happen after this action. Declarative knowledge plays the main role in this process. Outstanding pilots can respond in a very short time, as if the ability to fight is inborn, and they seem to defeat their opponents using intuition rather than thinking. This process is mainly based on procedural knowledge. Thus, declarative knowledge can be represented as a mapping: $\mathcal{S} \rightarrow \tilde{V} : \tilde{V}_{\pi\boldsymbol{\xi}}(s_t)$, where the superscript symbol $\tilde{\square}$ represents an estimation by the knowledge, $\boldsymbol{\xi}$ is a matrix that determines the value of $\tilde{V}_{\pi\boldsymbol{\xi}}(s_t)$ and $\boldsymbol{\xi}$ is fitted by learning. The content of procedural knowledge can be represented as $\mathcal{S} \rightarrow \pi : \pi_{\boldsymbol{\omega}}(s)$, where $\boldsymbol{\omega}$ is a parameter matrix that determines the value of $\pi_{\boldsymbol{\omega}}(s)$. The goal of learning is to obtain a policy which can maximize the profit [36]:

$$\begin{aligned}\max_{\boldsymbol{\omega}} J(\pi_{\boldsymbol{\omega}}) &= \int_{\mathcal{S}} \gamma^t \mathcal{P}_{\pi}(s) \int_{\mathcal{A}} \pi_{\boldsymbol{\omega}}(a|s) A(s,a) \mathrm{d}a\mathrm{d}s \\ &= \mathbb{E}_{s\sim\mathcal{P}^{\boldsymbol{\pi}},a\sim\pi_{\boldsymbol{\omega}}}\left[\gamma^t A(s,a)\right] \end{aligned} \tag{4}$$

where $\mathcal{P}_{\pi}(s)$ is the situation transition probability under the policy π. Thus, an accurate valuation $A(s,a)$ and an excellent action policy $\pi_{\boldsymbol{\omega}}(a|s)$ mean that both declarative and procedural knowledge will be promoted in learning. It has been proved that artificial neural networks can approximate functions with arbitrary complexity [51]. It is appropriate to use neural networks here to describe two kinds of knowledge; $\boldsymbol{\xi}$ and $\boldsymbol{\omega}$ are the weights of the neural networks. The knowledge structures in long and short-term memory are identical, and the differences are learning principle and frequency.

In addition, we noted that human beings could make multiple predictions about a future time based on the current situation. For example, excellent boxers can roughly predict several possible actions of their opponents and formulate a variety of coping strategies, as if he has performed several mock fights in multiple parallel universes. This is not an actual process occurring in the real world but a speculative process simulated in the mind. This means that the brain can store multi-branch simulation information in the working memory in parallel. The prediction of multiple steps and multiple possibilities in the future enables us to understand the value of the current situation more accurately and make a better decision. This is why only one who can predict many moves can be a top chess player. This mechanism could be used to improve learning efficiency. Therefore, we add a parallel universe that contains n identical training environments to the system. Different training environments are responsible for simulation on different timelines. Because the action policy $\pi \sim \mathcal{N}(\mu, \sigma^2)$ is a random distribution, the situation in each training environment will develop in different directions. The working memory stores the data generated by the parallel universe for a short period of time. Taking training environment i $(1 \leq i \leq n)$ as an example, the working memory takes a first-in-first-out principle to record the interactive data; the information flow diagram is shown in Figure 3.

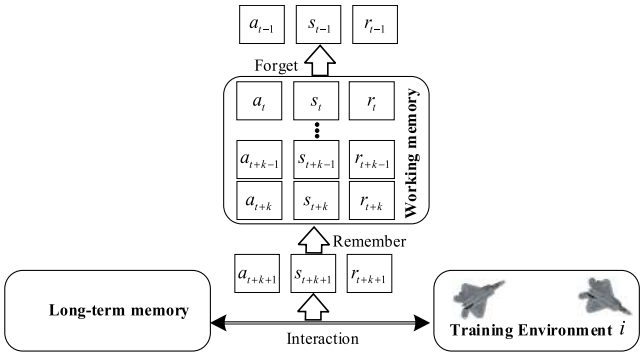


FIGURE 3. Information flow diagram of the working memory when recording the interactive data of the *i*th training environment.

We focus on building a system that can use data for self-learning. Compared with the existing cognitive frameworks, the functions of our system do not cover all features of cognition, such as social, reflective, deliberative, executive, reactive and physical layers. We perform self-learning using interactive data, including the architecture design, the simulation environment design, the data structure design and the method to ensure performance improvement.

In contrast to the existing research on autonomous air combat, there is no need to summarize the tactical manoeuvres created by human pilots, and because it is not dependent on exercise data, the system can learn from simulated training by itself. There is no need to construct a cost function or a score function artificially, and the system can learn a more objective data-driven optimization goal. The format of the decision outputs can meet the input requirement of general flight control systems for aircraft. Compared with the BFM or elementary manoeuvres, it has higher flexibility. From the experiment, we found that without prior knowledge, the system could master strategies such as classical tactical manoeuvres created by human pilots.

C. BASIC LEARNING PRINCIPLE OF THE SYSTEM

Long- and short-term memory have the same structure; both contain declarative and procedural knowledge. Long-term memory does not update as frequently as short-term memory. Short-term memory uses the data in the working memory for learning directly while long-term memory comes from the enhancement of short-term memory. Thus, the learning principle for long-term memory is designed as follows. Once the short-term knowledge has been updated n_{clc} times, the long-term knowledge clones the short-term knowledge in one round that we called a consolidation learning cycle (CLC); n_{clc} is called the length of the CLC. The parameters of the network in long- and short-term memory are recorded as $\boldsymbol{\xi}_{\mathrm{L}}$, $\boldsymbol{\omega}_{\mathrm{L}}$, $\boldsymbol{\xi}_{\mathrm{S}}$ and $\boldsymbol{\omega}_{\mathrm{S}}$, and the consolidation learning can be expressed as:

$$\begin{cases} \boldsymbol{\xi}_{\mathrm{L}} = \boldsymbol{\xi}_{\mathrm{S}} \\ \boldsymbol{\omega}_{\mathrm{L}} = \boldsymbol{\omega}_{\mathrm{S}} \end{cases} \tag{5}$$

In Section III-D, we will illustrate that the length of the CLC n_{clc} has an impact on learning performance (see Theorem 1). Thus, finding a method of deciding of the CLC becomes an important problem. The detailed principle will be clarified in the next section. Humans use stable knowledge to guide practice, imitating this mechanism, and the learning system selects actions and assesses situations according to the knowledge in long-term memory, that is:

$$\begin{cases} a_{t+1} \sim \pi_{\boldsymbol{\omega}_{\mathrm{L}}}(s_{t+1}) \\ \tilde{V}(s_t) = \tilde{V}_{\boldsymbol{\xi}_{\mathrm{L}}}(s_t) \end{cases} \tag{6}$$

The motivation of short-term learning, similar to what occurs in the biological brain, comes from cognitive bias, in other words, it comes from the error signal in this digital learning system. The short-term knowledge updates more often than that in long-term memory. Using the data stored in the working memory to build the error signals is an important problem in this stage of learning. In actual operation, calculating the situation-value according to Equation (1) is difficult because it must wait until the mission has ended. In fact, human beings also cannot obtain information about the whole process when dealing with a complex dynamic task. We usually only depend on the temporary information stored in working memory. On the one hand, this is because the capacity of working memory is limited; on the other hand, events closer to the current moment have greater impact on decision-making, and the farther the events are in time, the weaker the impact will be. Drawing lessons from this mechanism, the situation-value on a single timeline is estimated using the data in the working memory as:

$$\hat{V}^{(i)}(s_t) = r_t^{(i)} + \gamma r_{t+1}^{(i)} + \ldots + \gamma^k r_{t+k}^{(i)} + \gamma^{k+1} \tilde{V}\left(s_{t+k+1}^{(i)}\right) \tag{7}$$

where $1 \leq i \leq n$. This equation indicates how to use the data from the ith training environment to estimate a new approximate situation-value. Then, the error signal for the declarative knowledge produced by this timeline is as follows:

$$\hat{\delta}_{\mathrm{D}}^{(i)} = \hat{V}^{(i)}(s_t) - \tilde{V}\left(s_t^{(i)}\right) \tag{8}$$

There are n training environments in the parallel universe, and the multiple steps and multiple possibilities simulated allow us to give a more accurate error signal for the declarative knowledge; that is:

$$\hat{\delta}_{\mathrm{D}} = \frac{1}{n} \sum_{i=1}^{n} \left[\hat{V}^{(i)}(s_t) - \tilde{V}\left(s_t^{(i)}\right) \right] \tag{9}$$

Let α be the learning rate of short-term declarative knowledge and use ∇ to represent a gradient. The iterative learning principle of the declarative knowledge in short-term memory can be written as:

$$\boldsymbol{\xi}_{\mathrm{S}} = \boldsymbol{\xi}_{\mathrm{S}} + \alpha \nabla_{\boldsymbol{\xi}_{\mathrm{S}}} \hat{\delta}_{\mathrm{D}} \tag{10}$$

The eventual aim of learning is to obtain a policy to maximize the profit; therefore, the gradient of the optimization

target in Equation (4) can intuitively be the learning error of the procedural knowledge:

$$\delta_{\mathrm{P}} = \nabla_{\boldsymbol{\omega}} J(\pi_{\boldsymbol{\omega}}) = \int_{\mathcal{S}} \gamma^t \mathcal{P}_{\pi}(s) \int_{\mathcal{A}} \nabla_{\boldsymbol{\omega}} \pi_{\boldsymbol{\omega}}(a|s) A(s,a) \mathrm{d}a\mathrm{d}s$$
$$= \mathbb{E}_{s\sim\mathcal{P}^{\pi}, a\sim\pi_{\boldsymbol{\omega}}} \left[\nabla_{\boldsymbol{\omega}} \log(\pi_{\boldsymbol{\omega}}(a|s)) \gamma^t A(s,a) \right] \quad (11)$$

The $A(s,a)$ here can be obtained by the long-term knowledge, according to Definition 3:

$$\hat{A}_t(s_t, a_t) = -\tilde{V}_{\xi_{\mathrm{L}}}(s_t) + r_t + \gamma \tilde{V}_{\xi_{\mathrm{L}}}(s_{t+1}) \quad (12)$$

Then, the practical error signal for procedural knowledge learning is as follows:

$$\hat{\delta}_{\mathrm{P}} = \mathbb{E}_{s\sim\mathcal{P}^{\pi_{\mathrm{L}}}, a\sim\pi_{\boldsymbol{\omega}_{\mathrm{L}}}} \left[\frac{\nabla_{\boldsymbol{\omega}_s} \pi_{\boldsymbol{\omega}_{\mathrm{S}}}}{\pi_{\boldsymbol{\omega}_{\mathrm{L}}}} \gamma^t \hat{A}_t \right] \quad (13)$$

Let β denote the learning rate of short-term procedural knowledge; the basic iterative learning principle can be expressed as follows:

$$\boldsymbol{\omega}_s = \boldsymbol{\omega}_s + \beta \hat{\delta}_{\mathrm{P}} \quad (14)$$

Under this learning principle, one single CLC can be considered as a process during which the short-term area learns new knowledge based on old knowledge. In the next section, we will illustrate how to guarantee the growth of the learning effect under this long- and short-term asynchronous learning principle.

D. MECHANISMS TO GUARANTEE IMPROVEMENT

In this section, we explore a way to make learning more stable and effective. The most intuitive criterion for evaluating a policy is the reward it can obtain; then, we have the following definition:

Definition 4: The criterion for evaluating the policy π could be defined as starting from the situation s_0 to the termination of the mission, and the expected discounted reward obtained by the policy is:

$$\kappa(\pi) = \mathbb{E}_{a_0,s_0,a_1,s_1,\ldots\sim\pi} \left[\sum_{t=t_0}^{\infty} \gamma^{t-t_0} r(s_t) \right] \quad (15)$$

Theorem 2: For different policies π_1 and π_2, their criteria have the following relationship:

$$\kappa(\pi_2) = \kappa(\pi_1) + \mathbb{E}_{a_0,s_0,\ldots\sim\pi_2} \left[\sum_{t=t_0}^{\infty} \gamma^{t-t_0} A_{\pi_1}(s_t, a_t) \right] \quad (16)$$

Proof: According to Definition 3, we can obtain:

$$\mathbb{E}_{a_0,s_0,\ldots\sim\pi_2} \left[\sum_{t=t_0}^{\infty} \gamma^{t-t_0} A_{\pi_1}(s_t, a_t) \right]$$
$$= \mathbb{E}_{a_0,s_0\ldots\sim\pi_2} \left[\sum_{t=t_0}^{\infty} \gamma^{t-t_0} \left(r(s_t, a_t) + \gamma V_{\pi_1}(s_{t+1}) - V_{\pi_1}(s_t) \right) \right]$$
$$= \mathbb{E}_{a_0,s_0,\ldots\sim\pi_2} \left[-V_{\pi_1}(s_0) + \sum_{t=t_0}^{\infty} \gamma^{t-t_0} r(s_t) \right]$$
$$= -\mathbb{E}_{s_0} \left[V_{\pi_1}(s_0) \right] + \mathbb{E}_{a_0,s_0,\ldots\sim\pi_2} \left[\sum_{t=t_0}^{\infty} \gamma^{t-t_0} r(s_t) \right]$$
$$= -\kappa(\pi_1) + \kappa(\pi_2) \quad (17)$$

Rearranging, Theorem 1 has been proved.

Theorem 1 can be written as:

$$\kappa(\pi_2) = \kappa(\pi_1) + \sum_{t=t_0}^{\infty} \sum_{\mathcal{S}} \mathcal{P}_{\pi_2}(s_t) \times \sum_{\mathcal{A}} \pi_2(a_t|s_t) \gamma^{t-t_0} A_{\pi_1}(s_t, a_t) \quad (18)$$

Let

$$\vartheta_{\pi} = \mathcal{P}_{\pi}(s_0) + \gamma \mathcal{P}_{\pi}(s_1) + \gamma^2 \mathcal{P}(s_2) + \ldots \quad (19)$$

where $s_0, s_1, s_2 \ldots$ is a situation trajectory sampled from π. Now, Theorem 1 can be rewritten as follows:

$$\kappa(\pi_2) = \kappa(\pi_1) + \sum_{\mathcal{S}} \vartheta_{\pi_2} \sum_{\mathcal{A}} \pi_2(a|s) A_{\pi_1}(s,a) \quad (20)$$

According to Section 4, the short-term area takes further steps learning and produces a new policy $\pi_{\omega_{\mathrm{S}}}$ based on the old policy $\pi_{\omega_{\mathrm{L}}}$ during each CLC. Using Theorem 1, we have:

$$\kappa(\pi_{\boldsymbol{\omega}_{\mathrm{S}}}) = \kappa(\pi_{\boldsymbol{\omega}_{\mathrm{L}}}) + \sum_{\mathcal{S}} \vartheta_{\pi_{\omega_{\mathrm{S}}}} \sum_{\mathcal{A}} \pi_{\boldsymbol{\omega}_{\mathrm{S}}}(a|s) A_{\pi_{\boldsymbol{\omega}_{\mathrm{L}}}}(s,a) \quad (21)$$

If the learning improves in every CLC, that is, $\kappa(\pi_{\boldsymbol{\omega}_{\mathrm{S}}}) \geq \kappa(\pi_{\boldsymbol{\omega}_{\mathrm{L}}})$, at the end of each CLC, we can conclude that the learning is effective. However, $\vartheta_{\pi_{\omega_{\mathrm{S}}}}$ is determined by the new policy $\pi_{\boldsymbol{\omega}_{\mathrm{S}}}$, and the new policy keeps updating during the learning process, so using Equation (20) to judge the learning effect is inoperable. In fact, during the practical learning process, we use long-term knowledge to guide action and to assess the situation in a single CLC. According to Equations (11), (12) and (13), what we truly have in practical learning is:

$$\kappa'_{\pi_{\omega_{\mathrm{L}}}}(\pi_{\boldsymbol{\omega}_{\mathrm{S}}}) = \kappa(\pi_{\boldsymbol{\omega}_{\mathrm{L}}}) + \sum_{\mathcal{S}} \vartheta_{\pi_{\omega_{\mathrm{L}}}} \sum_{\mathcal{A}} \pi_{\boldsymbol{\omega}_{\mathrm{S}}}(a|s) A_{\pi_{\boldsymbol{\omega}_{\mathrm{L}}}}(s,a) \quad (22)$$

Next, the relationship between $\kappa(\pi_{\boldsymbol{\omega}_{\mathrm{S}}})$ and $\kappa'_{\pi_{\omega_{\mathrm{L}}}}(\pi_{\boldsymbol{\omega}_{\mathrm{S}}})$ is analysed. Using $T \sim \pi$ to denote a trajectory sampled from π, and it is evident that $\mathbb{A}(s) = \mathbb{E}_{a\sim\pi_{\omega_{\mathrm{S}}}} \left[A_{\pi_{\boldsymbol{\omega}_{\mathrm{L}}}}(s,a) \right]$, $\kappa(\pi_{\boldsymbol{\omega}_{\mathrm{S}}})$ and $\kappa'_{\pi_{\omega_{\mathrm{L}}}}(\pi_{\boldsymbol{\omega}_{\mathrm{S}}})$ can be rewritten as:

$$\kappa(\pi_{\boldsymbol{\omega}_{\mathrm{S}}}) = \kappa(\pi_{\boldsymbol{\omega}_{\mathrm{L}}}) + \mathbb{E}_{T\sim\pi_{\boldsymbol{\omega}_{\mathrm{S}}}} \left[\sum_{t=0}^{\infty} \gamma^t \mathbb{A}(s) \right]$$
$$\kappa'_{\pi_{\omega_{\mathrm{L}}}}(\pi_{\boldsymbol{\omega}_{\mathrm{S}}}) = \kappa(\pi_{\boldsymbol{\omega}_{\mathrm{L}}}) + \mathbb{E}_{T\sim\pi_{\boldsymbol{\omega}_{\mathrm{L}}}} \left[\sum_{t=0}^{\infty} \gamma^t \mathbb{A}(s) \right] \quad (23)$$

The following theorem can describe the relationship between $\kappa(\pi_{\boldsymbol{\omega}_{\mathrm{S}}})$ and $\kappa'_{\pi_{\omega_{\mathrm{L}}}}(\pi_{\boldsymbol{\omega}_{\mathrm{S}}})$:

Theorem 3: Under the learning principle in Section 4, $\kappa'_{\pi_{\omega_L}}(\pi_{\omega_S})$ can be an approximate substitute for $\kappa(\pi_{\omega_S})$ to judge the learning effect.

Proof: Policies in long- and short-term memory are identical at the starting point of every CLC, so $\kappa'_{\pi_{\omega_L}}(\pi_{\omega_S})$ and $\kappa(\pi_{\omega_S})$ are first order approximations at $\boldsymbol{\omega}_S = \boldsymbol{\omega}_L$, that is:

$$\begin{aligned} \kappa'_{\pi_{\omega_L}}(\pi_{\omega_S})\Big|_{\boldsymbol{\omega}_S=\boldsymbol{\omega}_L} &= \kappa(\pi_{\omega_S})\big|_{\boldsymbol{\omega}_S=\boldsymbol{\omega}_L} \\ \nabla_{\boldsymbol{\omega}_S} \kappa'_{\pi_{\omega_L}}(\pi_{\omega_S})\Big|_{\boldsymbol{\omega}_S=\boldsymbol{\omega}_L} &= \nabla_{\omega_S} \kappa(\pi_{\omega_S})\Big|_{\boldsymbol{\omega}_S=\boldsymbol{\omega}_L} \end{aligned} \tag{24}$$

$\kappa'_{\pi_{\omega_L}}(\pi_{\omega_S})$ and $\kappa(\pi_{\omega_S})$ have the same initial value and change direction at the beginning of each CLC, so the gradient direction of $\kappa'_{\pi_{\omega_L}}(\pi_{\omega_S})$ is approximately the same as that of $\kappa(\pi_{\omega_S})$ near π_{ω_L} in each CLC. Combined with Equation (13), we have:

$$\nabla_{\boldsymbol{\omega}_S} \kappa(\pi_{\omega_S}) = \nabla_{\omega_S}\left[\kappa(\pi_{\omega_S}) - \kappa(\pi_{\omega_L})\right] \approx \nabla_{\boldsymbol{\omega}_S} \kappa'_{\pi_{\omega_L}}(\pi_{\omega_S}) = \hat{\delta}_P \tag{25}$$

We use the error signal $\hat{\delta}_P$ to drive the short-term procedural knowledge learning. $\hat{\delta}_P$ directly affects the change of the value of $\kappa'_{\pi_{\omega_L}}(\pi_{\omega_S})$. In each CLC, near π_{ω_L}, the development of $\kappa'_{\pi_{\omega_L}}(\pi_{\omega_S})$ can represent the real worth of the new policy $\kappa(\pi_{\omega_S})$. If the length of CLC n_{clc} is too large, the signs of $\nabla_{\boldsymbol{\omega}_S} \kappa'_{\pi_{\omega_L}}(\pi_{\omega_S})$ and $\nabla_{\omega_S} \kappa(\pi_{\omega_S})$ may be opposite, and the error signal $\hat{\delta}_P$ sampled by the old policy π_{ω_L} cannot guarantee the improvement of learning performance. On the other hand, if n_{clc} is too small, long-term memory will be updated frequently and brief data variations will have an impact on the long-term memory, making the stability of the system weak.

Theorem 2 does not give an explicit constraint. To ensure that $\kappa'_{\pi_{\omega_L}}(\pi_{\omega_S})$ is always representative in the learning process, we must give further constraints.

Definition 5: Let π_1 be the marginal distribution of a_1 and π_2 be the marginal distribution of a_2; that is, $P\{a_1|s\} = \pi_1(s)$ and $P\{a_2|s\} = \pi_2(s)$. If $P\{a_1 \neq a_2|s\} = \varepsilon$, we call the joint distribution (π_1, π_2) ε − *coupling* policies.

If we use the ε − *coupling* policies π_{ω_S} and π_{ω_L} to sample and obtain two trajectories $(a_{Si}, a_{Li})|s$, where $i = 0, 1, 2, \ldots t$, and let n_t denote the number of times $a_{Si} \neq a_{Li}$ when $i < t$, then we have:

$$\begin{aligned} \mathbb{E}_{s\sim\pi_{\omega_S}}[\mathbb{A}(s_t)] &= P(n_t = 0)\,\mathbb{E}_{s_t\sim\pi_{\omega_S}|n_t=0}[\mathbb{A}(s_t)] \\ &\quad + P(n_t > 0)\,\mathbb{E}_{s_t\sim\pi_{\omega_S}|n_t>0}[\mathbb{A}(s_t)] \\ \mathbb{E}_{s\sim\pi_{\omega_L}}[\mathbb{A}(s_t)] &= P(n_t = 0)\,\mathbb{E}_{s_t\sim\pi_{\omega_L}|n_t=0}[\mathbb{A}(s_t)] \\ &\quad + P(n_t > 0)\,\mathbb{E}_{s_t\sim\pi_{\omega_L}|n_t>0}[\mathbb{A}(s_t)] \end{aligned} \tag{26}$$

Obviously, $P(n_t = 0) = (1-\varepsilon)^t$ and $P(n_t > 0) = 1 - (1-\varepsilon)^t$. When $n_t = 0$,

$$\mathbb{E}_{s_t\sim\pi_{\omega_S}|n_t=0}[\mathbb{A}(s_t)] = \mathbb{E}_{s_t\sim\pi_{\omega_L}|n_t=0}[\mathbb{A}(s_t)] \tag{27}$$

Then, Equation (26) becomes:

$$\begin{aligned} \mathbb{E}_{s\sim\pi_{\omega_S}}[\mathbb{A}(s_t)] &= (1-\varepsilon)^t\,\mathbb{E}_{s_t\sim\pi_{\omega_S}|n_t=0}[\mathbb{A}(s_t)] \\ &\quad + \left[1-(1-\varepsilon)^t\right]\mathbb{E}_{s_t\sim\pi_{\omega_S}|n_t>0}[\mathbb{A}(s_t)] \\ \mathbb{E}_{s\sim\pi_{\omega_L}}[\mathbb{A}(s_t)] &= (1-\varepsilon)^t\,\mathbb{E}_{s_t\sim\pi_{\omega_L}|n_t=0}[\mathbb{A}(s_t)] \\ &\quad + \left[1-(1-\varepsilon)^t\right]\mathbb{E}_{s_t\sim\pi_{\omega_L}|n_t>0}[\mathbb{A}(s_t)] \end{aligned} \tag{28}$$

We can obtain

$$\begin{aligned} &\left|\mathbb{E}_{s\sim\pi_{\omega_S}}[\mathbb{A}(s_t)] - \mathbb{E}_{s\sim\pi_{\omega_L}}[\mathbb{A}(s_t)]\right| \\ &= \left[1-(1-\varepsilon)^t\right]\left|\mathbb{E}_{s_t\sim\pi_{\omega_S}|n_t>0}[\mathbb{A}(s_t)]\right. \\ &\quad \left. - \mathbb{E}_{s_t\sim\pi_{\omega_L}|n_t>0}[\mathbb{A}(s_t)]\right| \\ &\le \left[1-(1-\varepsilon)^t\right]\left|\mathbb{E}_{s_t\sim\pi_{\omega_S}|n_t>0}[\mathbb{A}(s_t)]\right| \\ &\quad + \left|\mathbb{E}_{s_t\sim\pi_{\omega_L}|n_t>0}[\mathbb{A}(s_t)]\right| \\ &\le 2\left[1-(1-\varepsilon)^t\right]\max_s|\mathbb{A}(s_t)| \end{aligned} \tag{29}$$

Then,

$$\begin{aligned} &\left|\kappa(\pi_{\omega_S}) - \kappa'_{\pi_{\omega_L}}(\pi_{\omega_S})\right| \\ &= \sum_{t=0}^{\infty}\gamma^t\left|\mathbb{E}_{s\sim\pi_{\omega_S}}[\mathbb{A}(s_t)] - \mathbb{E}_{s\sim\pi_{\omega_L}}[\mathbb{A}(s_t)]\right| \\ &\le \frac{2\varepsilon\gamma}{(1-\gamma)(1-\gamma(1-\varepsilon))}\max_s|\mathbb{A}(s_t)| \\ &\le \frac{2\varepsilon\gamma}{(1-\gamma)^2}\max_s|\mathbb{A}(s_t)| \end{aligned} \tag{30}$$

In addition, using the fact that $\mathbb{E}_{a\sim\pi_{\omega_L}}\left[A_{\pi_{\omega_L}}(s,a)\right] = 0$, we have

$$\begin{aligned} \mathbb{A}(s) &= \mathbb{E}_{a\sim\pi_{\omega_S}}\left[A_{\pi_{\omega_L}}(s,a)\right] \\ &= \mathbb{E}_{(a_S,a_L)\sim(\pi_{\omega_S},\pi_{\omega_L})}\left[A_{\pi_{\omega_L}}(s,a_S) - A_{\pi_{\omega_L}}(s,a_L)\right] \\ &= P(a_S \neq a_L)\,\mathbb{E}_{(a_S,a_L)\sim(\pi_{\omega_S},\pi_{\omega_L})} \\ &\quad \times\left[A_{\pi_{\omega_L}}(s,a_S) - A_{\pi_{\omega_L}}(s,a_L)\right] \end{aligned} \tag{31}$$

Then, we obtain

$$|\mathbb{A}(s)| \le 2\varepsilon\max\left|A_{\pi_{\omega_L}}(s,a)\right| \tag{32}$$

So,

$$\begin{aligned} &\left|\kappa(\pi_{\omega_S}) - \kappa'_{\pi_{\omega_L}}(\pi_{\omega_S})\right| \\ &\quad \le \frac{4\varepsilon^2\gamma}{(1-\gamma)^2}\max_s\left|A_{\pi_{\omega_L}}(s,a)\right| \end{aligned} \tag{33}$$

$$\kappa(\pi_{\omega_S}) \ge \kappa'_{\pi_{\omega_L}}(\pi_{\omega_S}) - \frac{4\varepsilon^2\gamma}{(1-\gamma)^2}\max_s\left|A_{\pi_{\omega_L}}(s,a)\right| \tag{34}$$

According to Section 4.2 in [52], if μ is the distribution of x and ν is distribution of y, then $P\{x \neq y\} = \|\mu - \nu\|_{TV}$, where $\|.\|_{TV}$ is the total variation distance. Writing $\max_s\left|A_{\pi_{\omega_L}}(s,a)\right|$ as δ and $\|.\|_{TV}$ as $D_{TV}(.)$, we have

$$\kappa(\pi_{\omega_S}) \ge \kappa'_{\pi_{\omega_L}}(\pi_{\omega_S}) - \frac{4\gamma}{(1-\gamma)^2}\delta D_{TV}(\pi_{\omega_L}, \pi_{\omega_S})^2 \tag{35}$$

In addition, Pinsker's inequality [53] (Lemma 2.5) states that

$$\sup\left\{\|\mu-\nu\|_{\mathrm{TV}}\right\}\leq\sqrt{\frac{1}{2}\mathrm{D}_{\mathrm{KL}}\left(\mu,\nu\right)} \tag{36}$$

where $\mathrm{D}_{\mathrm{KL}}\left(\mu,\nu\right)$ is the Kullback-Leibler divergence or relative entropy [33] (Section 2.6). Thus, we can obtain the following theorem.

Theorem 4: Under the learning principle in Section 4, the $\kappa\left(\pi_{\boldsymbol{\omega}_{\mathrm{S}}}\right)$ have the following lower bound:

$$\kappa\left(\pi_{\boldsymbol{\omega}_{\mathrm{S}}}\right)\geq\kappa'_{\pi_{\omega_{\mathrm{L}}}}\left(\pi_{\boldsymbol{\omega}_{\mathrm{S}}}\right)-\frac{2\gamma}{\left(1-\gamma\right)^{2}}\delta\mathrm{D}_{\mathrm{KL}}\left(\pi_{\boldsymbol{\omega}_{\mathrm{L}}},\pi_{\boldsymbol{\omega}_{\mathrm{S}}}\right) \tag{37}$$

From the definitions of $\kappa\left(\pi_{\boldsymbol{\omega}_{\mathrm{S}}}\right)$ and $\kappa'_{\pi_{\omega_{\mathrm{L}}}}\left(\pi_{\boldsymbol{\omega}_{\mathrm{S}}}\right)$, we obtain

$$\kappa\left(\pi_{\boldsymbol{\omega}_{\mathrm{L}}}\right)=\kappa'_{\pi_{\omega_{\mathrm{L}}}}\left(\pi_{\boldsymbol{\omega}_{\mathrm{L}}}\right) \tag{38}$$

Combining (38) with Theorem 3, we obtain

$$\kappa\left(\pi_{\boldsymbol{\omega}_{\mathrm{S}}}\right)-\kappa\left(\pi_{\boldsymbol{\omega}_{\mathrm{L}}}\right)\geq\kappa'_{\pi_{\omega_{\mathrm{L}}}}\left(\pi_{\boldsymbol{\omega}_{\mathrm{S}}}\right)-\kappa'_{\pi_{\omega_{\mathrm{L}}}}\left(\pi_{\boldsymbol{\omega}_{\mathrm{L}}}\right)-\frac{2\gamma}{\left(1-\gamma\right)^{2}}\delta\mathrm{D}_{\mathrm{KL}}\left(\pi_{\boldsymbol{\omega}_{\mathrm{L}}},\pi_{\boldsymbol{\omega}_{\mathrm{S}}}\right) \tag{39}$$

Thus, we can conclude that, by maximizing $\kappa'_{\pi_{\omega_{\mathrm{L}}}}\left(\pi_{\boldsymbol{\omega}_{\mathrm{S}}}\right)-2\gamma\delta\big/(1-\gamma)^{2}\mathrm{D}_{\mathrm{KL}}\left(\pi_{\boldsymbol{\omega}_{\mathrm{L}}},\pi_{\boldsymbol{\omega}_{\mathrm{S}}}\right)$ at each CLC, the true learning objective $\kappa\left(\pi_{\boldsymbol{\omega}_{\mathrm{S}}}\right)$ is guaranteed non-decreasing. That is, the length of the CLC n_{clc} is not definite but a variable constrained by $2\gamma\delta\big/(1-\gamma)^{2}\mathrm{D}_{\mathrm{KL}}\left(\pi_{\boldsymbol{\omega}_{\mathrm{L}}},\pi_{\boldsymbol{\omega}_{\mathrm{S}}}\right)$, and the learning of the procedural knowledge in a single CLC can be considered as an optimization problem:

$$\max_{\boldsymbol{\omega}_{\mathrm{S}}}\left[\kappa'_{\pi_{\omega_{\mathrm{L}}}}\left(\pi_{\boldsymbol{\omega}_{\mathrm{S}}}\right)-2\gamma\delta\Big/\left(1-\gamma\right)^{2}\mathrm{D}_{\mathrm{KL}}\left(\pi_{\boldsymbol{\omega}_{\mathrm{L}}},\pi_{\boldsymbol{\omega}_{\mathrm{S}}}\right)\right] \tag{40}$$

In practice, δ is not definite, and $2\gamma\delta\big/(1-\gamma)^{2}$ is a relatively large value, so using Equation (40) above directly makes the learning step size very small. Therefore, we use a heuristic method instead; using a constant λ as the penalty coefficient, as shown in Equation (41), we found that the algorithm was not very sensitive to λ (See Section IV-B-2).

$$\max_{\boldsymbol{\omega}_{\mathrm{S}}}\left[\kappa'_{\pi_{\omega_{\mathrm{L}}}}\left(\pi_{\boldsymbol{\omega}_{\mathrm{S}}}\right)-\lambda\mathrm{D}_{\mathrm{KL}}\left(\pi_{\boldsymbol{\omega}_{\mathrm{L}}},\pi_{\boldsymbol{\omega}_{\mathrm{S}}}\right)\right] \tag{41}$$

That is, to guarantee the growth of the learning effect, we need to impose a restriction on the error signal for procedural knowledge on the basis of Equation (13), as follows:

$$\hat{\delta}_{\mathrm{P}}=\mathbb{E}_{s\sim\mathcal{P}^{\pi_{\mathrm{L}}},a\sim\pi_{\boldsymbol{\omega}_{\mathrm{L}}}}\nabla_{\boldsymbol{\omega}_{s}}\left[\frac{\pi_{\boldsymbol{\omega}_{\mathrm{S}}}}{\pi_{\boldsymbol{\omega}_{\mathrm{L}}}}\gamma^{t}\hat{A}_{t}-\lambda\mathrm{D}_{\mathrm{KL}}\left(\pi_{\boldsymbol{\omega}_{\mathrm{L}}},\pi_{\boldsymbol{\omega}_{\mathrm{S}}}\right)\right] \tag{42}$$

E. NEURAL NETWORKS IN DECLARATIVE KNOWLEDGE AND PROCEDURAL KNOWLEDGE

In this study, we use neural networks as approximators. There are two kinds of structures, because the networks of long- and short-term memory are the same.

1) NEURAL NETWORKS IN PROCEDURAL KNOWLEDGE

The inputs of procedural knowledge are situation vectors, such as s_t in Equation (44). The outputs of procedural knowledge are the action policies, corresponding to $a_t=\left[\begin{array}{cccc}n_x & n_z & \phi\end{array}\right]_t$; the concrete form of the outputs are three pairs of mean and variance, which are μ_{n_x} and $\sigma^2_{n_x}$, μ_{n_z} and $\sigma^2_{n_z}$, and μ_{ϕ} and σ^2_{ϕ}. We use multi-layer non-convolutional deep belief nets (DBN) [54] to express procedural knowledge, as demonstrated in Figure 4. The activation functions of the hidden layers are selected as a rectified linear unit (ReLU) [55]. In the output layer, the activation functions of the mean units are selected as tanh, and softplus functions [56] are selected as the activation functions of the variance units. We use an empirical and heuristic method to determine the net nodes. Ultimately, the structure of the neural networks in procedural knowledge is confirmed as 16-500-500-300-6; that is, there are 16 input nodes, 3 hidden layers with 500, 500, and 300 nodes, and 6 output nodes.

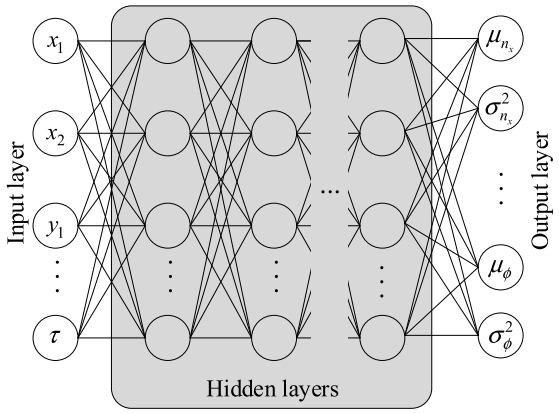


FIGURE 4. Structure of the neural networks in procedural knowledge.

2) NEURAL NETWORKS IN DECLARATIVE KNOWLEDGE

The inputs of declarative knowledge are same as those of procedural knowledge. The output of declarative knowledge is the situation-value $V_{\boldsymbol{\pi}}\left(s_t\right)$. We use another multi-layer DBN to express declarative knowledge, as demonstrated in Figure 5. The activation functions of the hidden layers are selected as ReLUs, and a linear function was selected as the activation function of the output layer. The structure of the nets in declarative knowledge are ultimately selected as 16-300-300-300-1.

F. DETAILED LEARNING PROCESS OF THE SYSTEM

First, we use the Xavier method [57] to initialize the parameters of the four neural networks. Then, the action policy $\pi_{\boldsymbol{\omega}_{\mathrm{L}}}$ consisting of μ_{n_x}, $\sigma^2_{n_x}$, μ_{n_z}, $\sigma^2_{n_z}$, μ_{ϕ} and σ^2_{ϕ} produced by long-term procedural knowledge is used to simple the action and act on the parallel training environment. The working memory collects the interactive data until its storage limit is reached. Next, according to Equations (9) and (42), the error signal for the short-term declarative and procedural

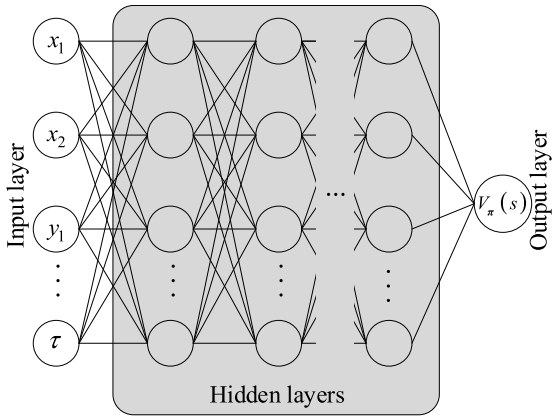


FIGURE 5. Structure of the neural networks in declarative knowledge.

knowledge, respectively, are produced. Next, a stochastic gradient ascent method called Adam [58] is used to update the parameters $\boldsymbol{\omega}_{\mathrm{S}}$ and $\boldsymbol{\xi}_{\mathrm{S}}$ of networks in short-term memory. Finally, the short-term knowledge is consolidated into the long-term one, and the parameter of the short-term knowledge is cloned to long-term knowledge. Then, the learning process starts the next cycle until the reward obtained in each round is stable. After training, mature procedural knowledge is used to guide the confrontation process.

The training algorithm can be expressed as:

Algorithm 1 Training Algorithm of the Learning System

While (not stop) **do**
  For $i = 1, \ldots, n$ **do**
    Run policy $\pi_{\boldsymbol{\omega}_{\mathrm{L}}}$ for k timesteps in each parallel universe
    Store $\left[s_t^{(i)} \ a_t^{(i)} \ r_t^{(i)} \right]$ in the working memory
End for
Estimate the action-value:
$\hat{A}_t\left(s_t, a_t\right) = -\tilde{V}_{\boldsymbol{\xi}_{\mathrm{L}}}\left(s_t\right) + r_t + \gamma \tilde{V}_{\boldsymbol{\xi}_{\mathrm{L}}}\left(s_{t+1}\right)$
Use a stochastic gradient ascent method to update the short-term procedural knowledge ω_{S} and to maximize $\frac{\pi_{\boldsymbol{\omega}_{\mathrm{S}}}}{\pi_{\boldsymbol{\omega}_{\mathrm{L}}}} \gamma^t \hat{A}_t - \lambda \mathrm{D}_{\mathrm{KL}}\left(\pi_{\boldsymbol{\omega}_{\mathrm{L}}}, \pi_{\boldsymbol{\omega}_{\mathrm{S}}}\right)$
Use a stochastic gradient descend method to update the short-term declarative knowledge $\boldsymbol{\xi}_{\mathrm{S}}$ to minimize $\frac{1}{n}\sum_{i=1}^{n}\left[r_t^{(i)} + \gamma r_{t+1}^{(i)} + \ldots + \gamma^k r_{t+k}^{(i)} + \gamma^{k+1}\tilde{V}\left(s_{t+k+1}^{(i)}\right) - \tilde{V}\left(s_t^{(i)}\right)\right]$
Consolidate the short-term knowledge into long-term knowledge: $\boldsymbol{\xi}_{\mathrm{L}} = \boldsymbol{\xi}_{\mathrm{S}}, \boldsymbol{\omega}_{\mathrm{L}} = \boldsymbol{\omega}_{\mathrm{S}}$

IV. EXPERIMENT

A. DESIGN OF THE TRAINING ENVIRONMENT AND REWARD

The training environment plays the role of calculating the position and attitude of the two planes to provide the reward data for the training. It can also test the training results as a simulator. Assuming the aircraft is a rigid body, angles of attack and sideslip are usually ignored, and the kinematic model of the aircraft can be expressed as [9], [59]–[61]:

$$\begin{aligned}
\dot{x} &= v\cos\theta\cos\psi \\
\dot{y} &= v\cos\theta\sin\psi \\
\dot{z} &= v\sin\theta \\
\dot{v} &= g\left(n_x - \sin\theta\right) \\
\dot{\theta} &= \frac{g}{v}\left(n_z\cos\phi - \cos\theta\right) \\
\dot{\psi} &= \frac{g n_z \sin\phi}{v\cos\theta}
\end{aligned} \tag{43}$$

where θ is the climbing flight path angle, ψ is the heading angle measured from north, ϕ is the roll angle, v is the ground reference speed, x, y, z are the position of the aircraft in north-east-height (NEH) coordinates, and n_x and n_z are the coefficients of forward and normal overload. The movement of the aircraft is controlled by $\left[\, n_x \ n_z \ \phi \,\right]$, and the manoeuvrability of an aircraft is determined by the ranges of n_x, n_z and v. The output action of the learning system is $a_t = \left[\, n_x \ n_z \ \phi \,\right]_t$.

We abandoned the index functions summarized by the existing research [9], [19], [62], [63], in which the reward is encouraged or punished determined only by the success or failure of the mission. The purpose is to exclude subjective human factors, to learn more objectives $A\left(s, a\right)$ and policies $\pi_{\boldsymbol{\omega}}\left(a|s\right)$ using the reward data generated by simulated combat, and to verify whether the method can achieve autonomous learning in the absence of human prior knowledge.

The goal of air combat is to attain and maintain a position of advantage in the rear of the enemy. That is, the learning system needs to guide the blue aircraft in Figure 6 to keep the angles η and τ as small as possible, while the goal of the red aircraft is the opposite. Therefore, we set a score principle for the reward feedback. Once $\eta \leq 20°$, $\tau \leq 30°$ and the distance from the enemy is between 100 and 500 metres, we call the situation *Almost lock* and return a reward $r = 1$ at this situation. If the *Almost lock* situation is maintained for more than 5 seconds, then the situation becomes *Lock*; in this case, a reward $r = 10$ is returned. Otherwise, if the enemy occupies the advantaged position, the situations are called *Almost be locked* and *Be locked* and the reward is $r = -1$ and $r = -10$, respectively. Moreover, if the altitude is lower than 10 m or the distance from the enemy aircraft is less than 10 m, the situation is judged to be *Crashed*, and a reward $r = -10$ is returned. In situations other than the above, $r = -0.1$ is returned each time. The learning system sends the guidance command and asks for the situation states and reward every 0.5 seconds. The vector shown in Equation (44) is the situation states returned by the training environment.

$$s_t = [x_1\ x_2\ y_1\ y_2\ z_1\ z_2\ v_1\ v_2\ \phi_1\ \phi_2\ \psi_1\ \psi_2\ \theta_1\ \theta_2\ \eta\ \tau]_t \tag{44}$$

B. EXPERIMENTAL RESULTS

We evaluated our approach on four common aerial encounter scenarios. The opposing aircraft have the same manoeuvrability as ours, so that the experimental data can better reflect

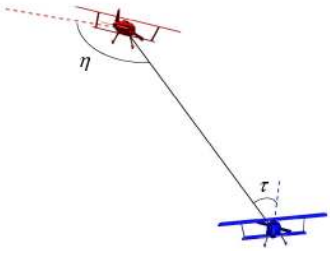


FIGURE 6. Diagram of η and τ.

the performance of the method. From the initial states until the situation is judged as *Lock*, *Be locked* or *Crashed*, or could not achieve any one above after 10,000 steps, we recorded one process like this as one confrontation round. The two aircraft are reset to the initial states to prepare for another round of training when the previous confrontation round ends. The discount factor is set as $\gamma = 0.999$ and the working memory storage length as $k = 50$. The experiments described in the rest of this section are performed on a computer with Intel i7-8700k CPU, NVIDIA GTX1070 GPU, 32 GB RAM and the Ubuntu 16.04 operating system.

In the first scenario, the initial speed of both sides was the same and the opposing aircraft marked as red appeared ahead of the blue one's nose; however, the blue aircraft was not located in the rear attack zone but instead on the side of the red one. The red aircraft was escaping with max roll angle to leave the area in front of the blue and to expand the angles η and τ. The initial states of both sides are shown in Table 1.

In the second scenario, the blue and red aircraft engaged nose-to-nose. The red one turned left, across the tail of the blue, trying to approach in a nose-to-tail fashion. The initial states are shown in Table 2.

TABLE 1. Initial states of both sides in the lateral encounter scenario.

States	Blue value	Red value
Speed	50 m/s	50 m/s
Speed range	[20 100] m/s	[20 100] m/s
Roll angle	-60°	-80°
Location (N, E, H)	(1700, 1000, 2000)	(2000, 1000, 2000)
Yaw angle	0°	-60°
Pitch angle	-10°	0°
n_z range	[-3 9]	[-3 9]
n_x range	[-3 5]	[-3 5]

TABLE 2. Initial states of both sides in the head-on encounter scenario.

States	Blue value	Red value
Speed	50 m/s	50 m/s
Speed range	[20 100] m/s	[20 100] m/s
Roll angle	0°	-80°
Location (N, E, H)	(2000,1100,2000)	(1950,900,2000)
Yaw angle	-90°	90°
Pitch angle	0°	0°
n_z range	[-3 9]	[-3 9]
n_x range	[-3 5]	[-3 5]

TABLE 3. Initial states of both sides in the lag-pursuit roll scenario.

States	Blue value	Red value
Speed	90 m/s	50 m/s
Speed range	[20 100] m/s	[20 100] m/s
Roll angle	-30°	-75°
Location (N, E, H)	(1900,1150,1980)	(2000,1000,2000)
Yaw angle	-90°	-90°
Pitch angle	22.5°	0°
n_z range	[-3 9]	[-3 9]
n_x range	[-3 5]	[-3 5]

TABLE 4. Initial states of both sides in the rolling scissors scenario.

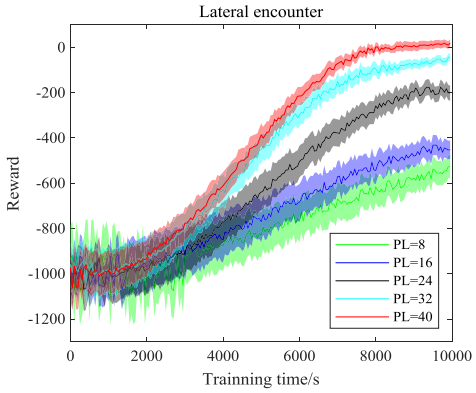
States	Blue value	Red value
Speed	70 m/s	50 m/s
Speed range	[20 100]m/s	[20 100]m/s
Roll angle	75°	0°
Location (N, E, H)	(2600,3000,2300)	(2000,3000,2000)
Yaw angle	180°	-90°
Pitch angle	-30°	0°
n_z range	[-3 9]	[-3 9]
n_x range	[-3 5]	[-3 5]

In the third scenario, the blue aircraft is located behind its opponent, its velocity is too great, and its initial pitch angle is too large. The red one tries to detour behind the blue through a horizontal turn. If the blue aircraft tries to directly lock the red one in the current state, it may overshoot because of the high speed and go from a position of advantage to one of disadvantage. The initial states are shown in Table 3.

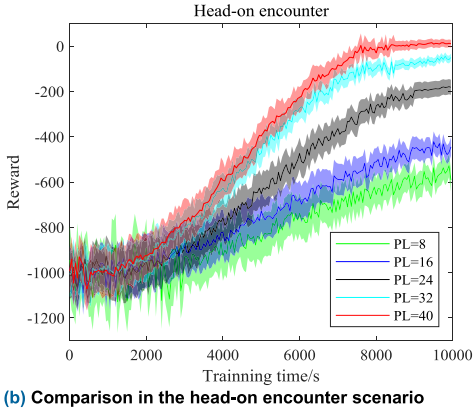
In the fourth scenario, the headings of the two aircraft are perpendicular, and the blue side is faster and its pitch angle is smaller. Whether the situation is an advantage or disadvantage to each side is unclear. The red side is looking for breakthrough opportunities through a continuous rolling scissors manoeuvre. See Table 4 for the initial states of both sides.

1) COMPARISON OF THE NUMBER OF PARALLEL UNIVERSES

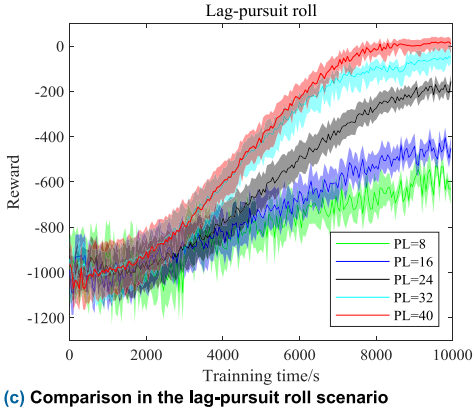
First, we illustrate the effect of the parallel spaces through comparative experiments. We evaluated the algorithm with a number of parallel universes $n = 8, 16, 24, 32, 40$ and with KL penalty coefficient $\lambda = 10$ on the four encounter scenarios. We executed the learning algorithm for 10000 seconds under each parameter setting. Figure 7 shows the reward obtained by the system during the learning; the curve is the mean value of the 20 experiments, and the shadow represents the boundary of the experimental data. It can be seen that the number of parallel spaces has a significant effect on the learning speed and the reward distribution. As the number of parallel spaces increases, the amount of data that needs to be processed also increases, it takes up more CPU and memory resources, and the speed-up dividends do not always increase.



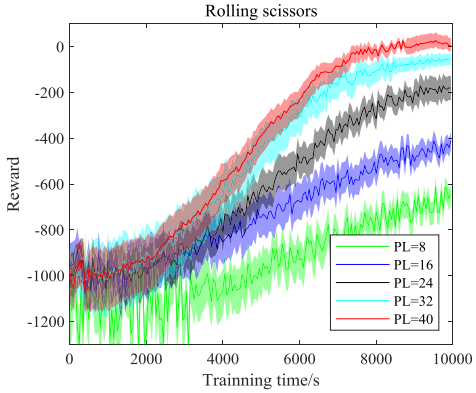
(a) Comparison in the lateral encounter scenario



(b) Comparison in the head-on encounter scenario



(c) Comparison in the lag-pursuit roll scenario



(d) Comparison in the rolling scissors scenario

FIGURE 7. Reward obtained during the parallel universe comparison experiment.

2) COMPARISON OF THE PENALTY COEFFICIENTS

Next, we give the experimental results under different KL penalty coefficients λ. We use the same hyperparameters as in the previous experiment and set the number of parallel universes PL=40, taking $\lambda = 0.5, 1, 2, 5, 10$ heuristically. As in the previous section, we tested 20 times with each parameter on the two scenarios and executed the learning algorithm for 10000 seconds under each parameter setting; the score during the learning is shown in Figure 8. As we can see, the learning is not very sensitive to the KL penalty coefficient λ in the range of $[0.5, 10]$; a smaller λ leads to relatively faster learning but also a greater variance, and a greater λ makes the learning relatively slower but performs better on stability.

In our opinion, this phenomenon is caused by many factors. On the one hand, we adopt a smaller learning rate in updating the parameters of the neural network, so we can use a smaller penalty coefficient; on the other hand, $2\gamma/(1-\gamma)^2\delta$ in Theorem 3 is a relatively large value, especially in the later stages of learning, so larger values can also be effective.

3) DIRECT EXHIBITION OF LEARNING ACHIEVEMENTS

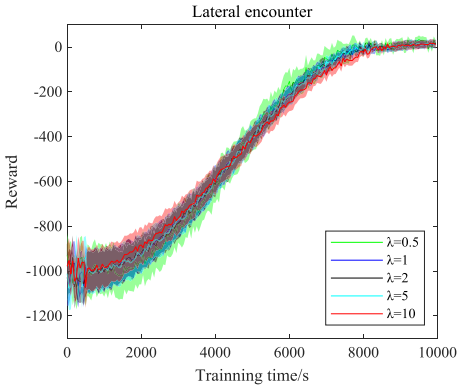
After the training, we connected the trained long-term module directly to the training environment to verify the learning effect, and the role of the training environment was converted to a simulator. We obtained confrontation trajectories for each encounter scenario as shown in Figure 9.

As can be seen, without any flight rules summarized by humans, the system can learn from the confrontation data by itself. We only set a score principle to help the computer distinguish success from failure. The confrontation trajectory shows that the system can produce different manoeuvres to deal with different situations. It has mastered the conversion between speed and height so that it can reverse nose quickly with less energy loss. Manoeuvres obtained from learning are similar to the classical fight tactic high yo-yo and the barrel-roll attack [64].

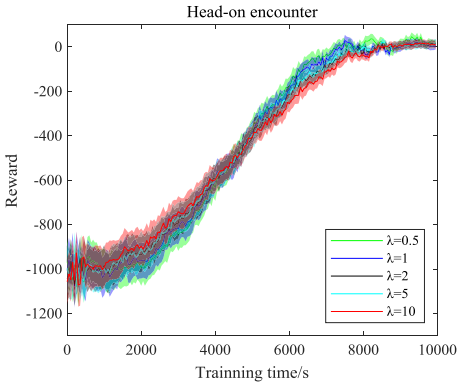
4) COMPARISON TO OTHER ALGORITHMS

In the engagement scenarios of the previous experiments, the opponent aircraft only performed some simple manoeuvres to escape. However, real air combat is consumed with fierce attack and defence, so we created an antagonistic opponent for the system. The red aircraft had the ability to evaluate the situation, to predict the opponent's intentions and to decide on manoeuvres; it was commanded by the Bayesian inference and moving horizon optimization method (BI&MHO) [9]. We set up a fair arena, as shown in Table 3, and the two aircraft engaged nose-to-nose; their manoeuvrability, initial speed, height, roll angle and pitch angle were the same.

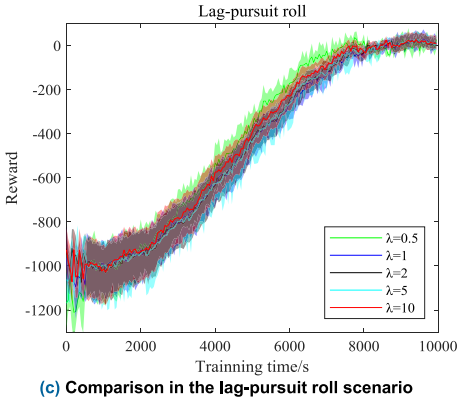
Figure 10 (a) shows the reward obtained by the system. We executed the learning algorithm 20 times; the curve is the mean value of the reward in the 20 experiments, and



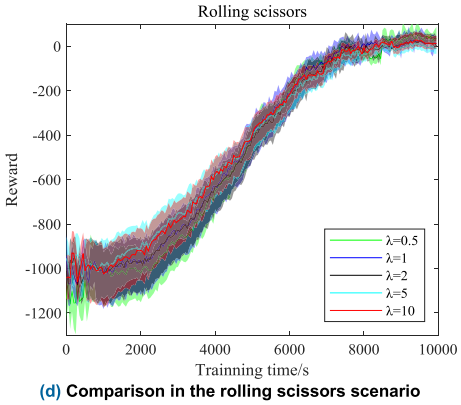
(a) Comparison in the lateral encounter scenario



(b) Comparison in the head-on encounter scenario

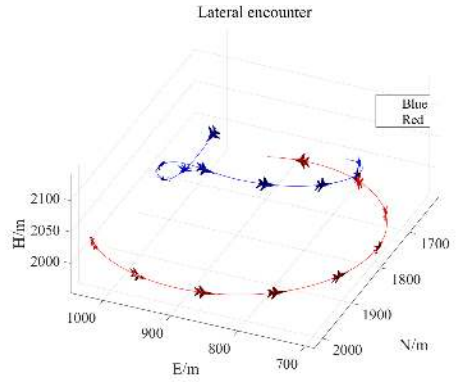


(c) Comparison in the lag-pursuit roll scenario

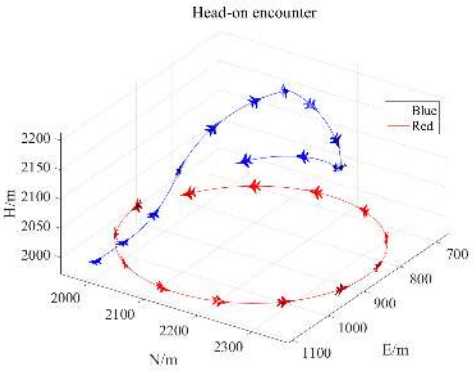


(d) Comparison in the rolling scissors scenario

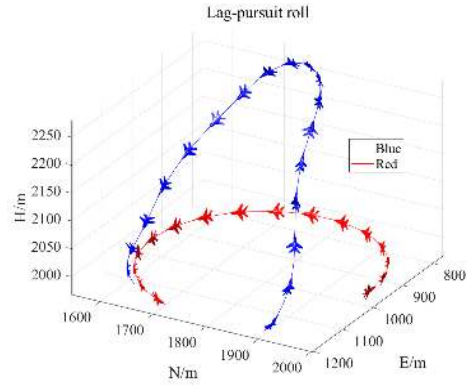
FIGURE 8. Reward obtained during the penalty coefficient comparison experiment.



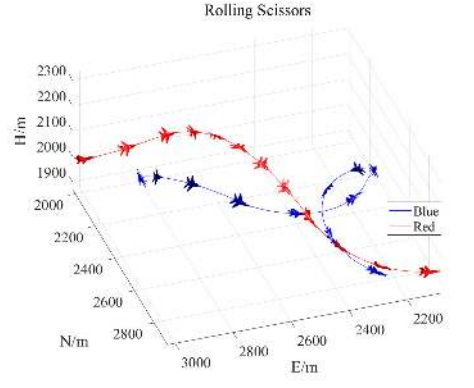
(a) Trajectories in the lateral encounter scenario



(b) Trajectories in the head-on encounter scenario



(c) Trajectories in the lag-pursuit roll scenario



(d) Trajectories in the rolling scissors scenario

FIGURE 9. The confrontation trajectories of both sides.

TABLE 5. Initial state of both sides in the dogfight scenario.

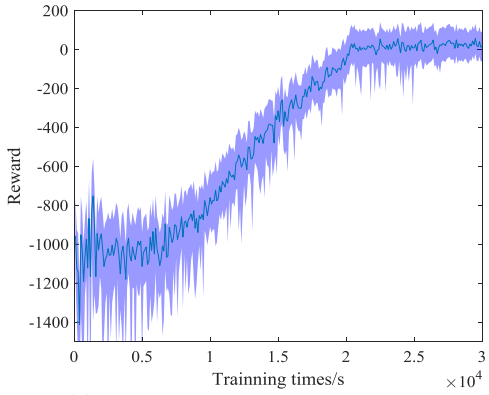
States	Blue value	Red value
Speed	50 m/s	50 m/s
Speed range	[20 100] m/s	[20 100] m/s
Roll angle	0°	0°
Location (N, E, H)	(2000, 1000, 2000)	(2000, 3000,2000)
Yaw angle	90°	-90°
Pitch angle	0°	0°
n_z range	[-3 9]	[-3 9]
n_x range	[-3 5]	[-3 5]

TABLE 6. Comparison of the two methods.

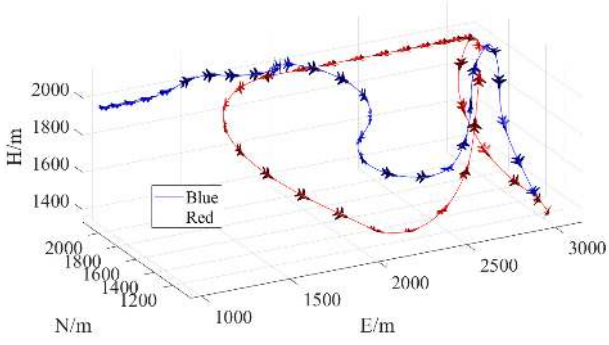
Item	BI&MHO method	Method in this paper
Situation assessment	Four levels with different weighting coefficients	No grading boundary but continuous values
Situation prediction	Predict opponent's position using basic manoeuvres	Establish knowledge about the situation development tendency
Execution mode	Solve online	Execute after training
Characteristics	Can deploy quickly, but cannot learn	Unusable before training but can keep learning

the shadow represents the boundary of the reward data. The data fluctuates greatly in the early period, and the blue side might even be defeated by its opponent. As the learning went on, the system understood its enemy more comprehensively. Then, the fluctuation of the data reduced gradually and the mean of the reward came to rise continuously. In the later stage, the reward tended to be stable, and the blue side found the action policy to lock the red within a short time. Consider a case in which a blue aircraft driven by a trained procedural network confronts a red aircraft driven by the BI&MHO method in a training environment. Two different trajectories in the dogfight experiment are shown in Figure 10 (b), (c) and (d). The trajectories of the two confrontations are not the same. This is because the action policy is a Gaussian distribution, so actions sampled according to the policy do not have a definite value. When the opponent is an aggressive aircraft, different action choices may cause great changes in the development of confrontation trajectories. This also explains the reward data fluctuation in the later stage of learning.

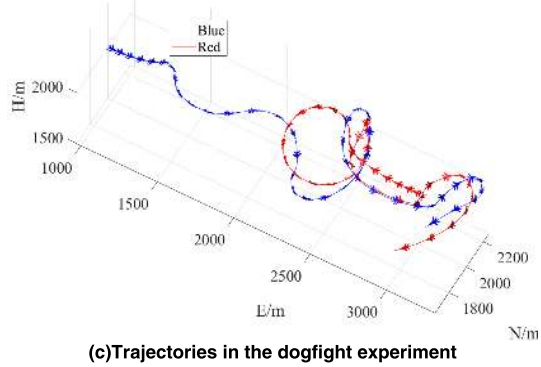
In contrast to the BI&MHO method, our approach is a new way to learn how to make decisions without existing rules. First, the confrontation situations are divided into the four categories *Advantage*, *Disadvantage*, *Mutual Safety* and *Mutual Disadvantage* in the BI&MHO method. Then, linear addition of the fuzzy angle, height, distance and speed membership functions is used as the optimization objective of manoeuvre decisions. The linear addition weights for the four kinds of situations are set as constant, and thus, the manoeuvring strategy may have a regular pattern that can be grasped through learning. By contrast, our method assesses situations



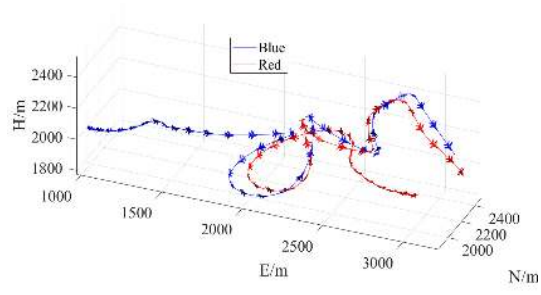
(a) Reward obtained in the dogfight scenario



(b) Trajectories in the dogfight experiment



(c)Trajectories in the dogfight experiment



(d)Trajectories in the dogfight experiment

FIGURE 10. Results of the dog-fight experiment.

as continuous values, and action strategies guided by this type of assessment will be more diversified. Second, BI&MHO uses a linear combination of five basic manoeuvres to predict the position of the opponent. However, the blue aircraft is directed by continuous guidance commands $\left[n_x \; n_z \; \phi \right]$ that

are sampled according the Gaussian distribution policy π in our method. The manoeuvres of the blue aircraft are equivocal and are more complex than the ones the BI&MHO method could predict. As the forecast period of the moving horizon optimization method increases, the prediction error will increase sharply. In our method, the situation-value considers not only the current states but also the trend of the situation in the future. The action choices of both sides will be reflected in the situation-value. We do not predict the opponent's specific actions but establish knowledge about the situation development tendency through statistical learning. These reasons lead the learning system to a final victory. In addition, the BI&MHO method makes decisions by solving optimal values online. Once this method is designed, it is workable, because the designer puts his prior knowledge about situation assessment, position prediction and decision basis into the algorithm. In contrast, the method proposed in this paper is unusable before learning. Because the algorithm is a learning framework, the knowledge for decision-making is obtained through interactions with the training environment. The BI&MHO method has the ability to deploy quickly, but no ability to learn. Our method can only be used after training but can keep learning through interactive data.

V. CONCLUSION AND FUTURE WORK

In this paper, the learning model of the human brain was analysed and a novel brain-like air combat learning system was designed. The main conclusions are: By applying the cognitive mechanism of the brain to autonomous decision of air combat manoeuvres, the learning system designed in this paper is an effective self-learning structure. The parallel universe, parallel simulation and the data acquisition method proposed can significantly improve learning efficiency within a certain range. An appropriate length of the consolidation learning cycle (CLC) can ensure learning performance growth. Transforming the CLC length adjustment problem into an optimization problem can make the algorithm easier to execute. Good results have been achieved in digital experiments.

To make the method proposed in this paper more practical, several issues need further research: How to add humans' prior knowledge about aircraft kinematic models, situation assessment, intention prediction and decision-making into the learning system so that the system will have certain availability without training. Ideally, prior knowledge will not only have no conflict with learning but will also be able to improve learning speed. In this paper, we design and validate the learning system in a 1-vs-1 scenario. To extend the method to a multiplayer confrontation, cooperative learning framework, reasonable reward and knowledge sharing mechanism need to be explored. To fully apply the method in a real UAV, many problems must be solved, including computing power, platform load, power supply, sensor accuracy, communication timeliness, safety and so on.

ACKNOWLEDGMENT

The authors would like to thank Dr. J. Wang and Dr. X. Wang for suggesting improvements after reading early versions of this manuscript and Dr. M. Zhou for fruitful discussions regarding the algorithmic programming.

REFERENCES

[1] R. Scott, "Bringing UAVs to the dogfight: ACE looks to automate close quarters air combat," Jane's Int. Defence Rev., News Rep., 2019. [Online]. Available: https://air.dfns.net/2019/06/26/bringing-uavs-to-the-dogfight-ace-looks-to-automate-close-quarters-air-combat/
[2] K. Virtanen, T. Raivio, and R. P. Hämäläinen, "Decision theoretical approach to pilot simulation," *J. Aircr.*, vol. 36, no. 4, pp. 632–641, Jul. 1999.
[3] K. Virtanen, T. Raivio, and R. P. Hamalainen, "Modeling pilot's sequential maneuvering decisions by a multistage influence diagram," *J. Guid., Control, Dyn.*, vol. 27, no. 4, pp. 665–677, Jul. 2004.
[4] K. Virtanen, J. Karelahti, and T. Raivio, "Modeling air combat by a moving horizon influence diagram game," *J. Guid., Control, Dyn.*, vol. 29, no. 5, pp. 1080–1091, Sep. 2006, doi: 10.2514/1.17168.
[5] 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.
[6] H. Mukai, A. Tanikawa, I. Tunay, I. A. Ozcan, I. N. Katz, and H. Schättler, "Sequential linear-quadratic method for differential games with air combat applications," *Comput. Optim. Appl.*, vol. 25, nos. 1–3, pp. 193–222, 2003.
[7] P. Mauro and A. C. Bruce, "Numerical solution of the three dimensional orbital pursuit-evasion games," *J. Guid., Control, Dyn.*, vol. 32, no. 2, pp. 474–487, 2009.
[8] J. Poropudas and K. Virtanen, "Game-theoretic validation and analysis of air combat simulation models," *IEEE Trans. Syst., Man, Cybern. A, Syst. Humans*, vol. 40, no. 5, pp. 1057–1070, Sep. 2010.
[9] 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.
[10] Y. Dong and J. Ai, "Trial input method and own-aircraft state prediction in autonomous air combat," *J. Aircr.*, vol. 49, no. 3, pp. 947–954, May 2012.
[11] Y. Dong, J. Huang, and J. Ai, "Visual perception-based target aircraft movement prediction for autonomous air combat," *J. Aircr.*, vol. 52, no. 2, pp. 538–552, Mar. 2015.
[12] H. R. Sonawane and S. P. Mahulikar, "Effect of missile turn rate on aircraft susceptibility to infrared-guided missile," *J. Aircr.*, vol. 50, no. 2, pp. 663–667, Mar. 2013.
[13] J. S. Mcgrew, J. P. How, L. Bush, B. Williams, and N. Roy, "Air combat strategy using approximate dynamic programming," in *Proc. AIAA Guid., Navigat. Control Conf. Exhibit, Amer. Inst. Aeronaut. Astronaut.*, 2008, pp. 1–20.
[14] J. S. Mcgrew, J. P. How, B. Williams, and N. Roy, "Air-combat strategy using approximate dynamic programming," *J. Guid., Control, Dyn.*, vol. 33, no. 5, pp. 1641–1654, Sep. 2010, doi: 10.2514/1.46815.
[15] Y. Ma, X. Ma, and X. Song, "A case study on air combat decision using approximated dynamic programming," *Math. Problems Eng.*, vol. 2014, Sep. 2014, Art. no. 183401.
[16] F.-Y. Wang, H. Zhang, and D. Liu, "Adaptive dynamic programming: An introduction," *IEEE Comput. Intell. Mag.*, vol. 4, no. 2, pp. 39–47, May 2009, doi: 10.1109/MCI.2009.932261.
[17] L. Busoniu, D. Ernst, B. De Schutter, and R. Babuška, "Approximate reinforcement learning: An overview," in *Proc. IEEE Symp. Adapt. Dyn. Program. Reinforcement Learn. (ADPRL)*, Apr. 2011, pp. 1–8.
[18] L. Xiao, D. Sun, Y. Liu, and Y. Huang, "A combined method based on expert system and BP neural network for UAV systems fault diagnosis," in *Proc. Artif. Intell. Comput. Intell.*, 2010, pp. 3–6.
[19] J. Kaneshige and K. Krishnakumar, "Artificial immune system approach for air combat maneuvering," *Proc. SPIE, Intell. Comput., Theory Appl. V, Int. Soc. Opt. Photon.*, vol. 6560, pp. 656009–656012, Apr. 2007.
[20] N. Ernest and K. Cohen, "Fuzzy logic based intelligent agents for unmanned combat aerial vehicle control," *J. Defense Manage.*, vol. 6, no. 1, p. 139, 2015, doi: 10.4172/2167-0374.1000139.
[21] N. D. Ernest, "Genetic fuzzy trees for intelligent control of unmanned combat aerial vehicles," Ph.D. dissertation, College Eng. Appl. Sci., Univ. Cincinnati, Cincinnati, OH, USA, 2015.

[22] N. Ernest, D. Carroll, C. Schumacher, M. Clark, K. Cohen, and G. Lee, "Genetic fuzzy based artificial intelligence for unmanned combat aerial vehicle control in simulated air combat missions," *J. Defense Manage.*, vol. 6, no. 1, p. 144, 2016, doi: 10.4172/2167-0374.1000144.

[23] S. Emel'Yanov, D. Makarov, A. I. Panov, and K. Yakovlev, "Multi-layer cognitive architecture for UAV control," *Cognit. Syst. Res.*, vol. 39, pp. 58–72, Sep. 2016, doi: 10.1016/j.cogsys.2015.12.008.

[24] M. Rollo, M. Selecký, P. Losiewicz, J. Reade, and N. Maida, "Framework for incremental development of complex unmanned aircraft systems," in *Proc. Integr. Commun., Navigat. Surveill. Conf.*, 2015.

[25] J. L. Sanchez-Lopez, M. Molina, H. Bavle, C. Sampedro, R. A. S. Fernández, and P. Campoy, "A multi-layered component-based approach for the development of aerial robotic systems: The aerostack framework," *J. Intell. Robot. Syst.*, vol. 88, nos. 2–4, pp. 683–709, Dec. 2017, doi: 10.1007/s10846-017-0551-4.

[26] M. F. Pinto, A. G. Melo, A. L. M. Marcato, and C. Urdiales, "Case-based reasoning approach applied to surveillance system using an autonomous unmanned aerial vehicle," in *Proc. IEEE 26th Int. Symp. Ind. Electron. (ISIE)*, Jun. 2017, pp. 1324–1329.

[27] C. U. Chithapuram, A. K. Cherukuri, and Y. V. Jeppu, "Aerial vehicle guidance based on passive machine learning technique," *Int. J. Intell. Comput. Cybern.*, vol. 9, no. 3, pp. 255–273, Aug. 2016, doi: 10.1108/ijicc-12-2015-0042.

[28] C. Chithapuram, Y. V. Jeppu, and C. A. Kumar, "Artificial intelligence guidance for unmanned aerial vehicles in three dimensional space," in *Proc. Int. Conf. Contemp. Comput. Inform. (ICI)*, 2014, pp. 1256–1261.

[29] T. Zeng and B. Si, "Cognitive mapping based on conjunctive representations of space and movement," *Frontiers Neurorobotics*, vol. 11, pp. 1–16, Nov. 2017, doi: 10.3389/fnbot.2017.00061.

[30] S. Ullman, "Using neuroscience to develop artificial intelligence," *Science*, vol. 363, no. 6428, pp. 692–693, Feb. 2019, doi: 10.1126/science.aau6595.

[31] A. Banino *et al.*, "Vector-based navigation using grid-like representations in artificial agents," *Nature*, vol. 557, no. 7705, pp. 429–433, May 2018, doi: 10.1038/s41586-018-0102-6.

[32] K. Zhou, R. Wei, Z. Xu, Q. Zhang, H. Lu, and G. Zhang, "An air combat decision learning system based on a brain-like cognitive mechanism," *Cogn. Comput.*, pp. 1–12, Sep. 2019. [Online]. Available: https://link.springer.com/article/10.1007/s12559-019-09683-7, doi: 10.1007/s12559-019-09683-7.

[33] D. J. C. Mackay, *Information Theory, Inference, and Learning Algorithms*. Cambridge, U.K.: Cambridge Univ. Press, 2003.

[34] P. A. Laurent, "A neural mechanism for reward discounting: Insights from modeling hippocampal-striatal interactions," *Cogn. Comput.*, vol. 5, no. 1, pp. 152–160, 2013, doi: 10.1007/s12559-012-9178-8.

[35] D. Silver, A. Huang, C. J. Maddison, A. Guez, L. Sifre, G. Van Den Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot, S. Dieleman, D. Grewe, J. Nham, N. Kalchbrenner, I. Sutskever, T. Lillicrap, M. Leach, K. Kavukcuoglu, T. Graepel, and D. Hassabis, "Mastering the game of go with deep neural networks and tree search," *Nature*, vol. 529, no. 7587, pp. 484–489, Jan. 2016, doi: 10.1038/nature16961.

[36] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*, 2nd ed. Cambridge, MA, USA: MIT Press, 2016.

[37] Z. Wang, T. Schaul, M. Hessel, H. V. Hasselt, M. Lancto, and N. D. Freitas, "Dueling network architectures for deep reinforcement learning," 2016, *arXiv:1511.06581*. [Online]. Available: https://arxiv.org/abs/1511.06581

[38] W. Schultz, P. Dayan, and P. R. Montague, "A neural substrate of prediction and reward," *Science*, vol. 275, no. 5306, pp. 1593–1599, Mar. 1997.

[39] M. Takac and A. Knott, "A neural network model of episode representations in working memory," *Cogn. Comput.*, vol. 7, no. 5, pp. 509–525, Oct. 2015.

[40] E. Koechlin and C. Summerfield, "An information theoretical approach to prefrontal executive function," *Trends Cognit. Sci.*, vol. 11, no. 6, pp. 229–235, Jun. 2007.

[41] K. Zhang, J. Z. Guo, Y. Peng, W. Xi, and A. Guo, "Dopamine-mushroom body circuit regulates saliency-based decision-making in drosophila," *Science*, vol. 316, no. 5833, pp. 1901–1904, Jul. 2007.

[42] Y. Niv and G. Schoenbaum, "Dialogues on prediction errors," *Trends Cognit. Sci.*, vol. 12, no. 7, pp. 265–272, Jul. 2008.

[43] J. Garrison, B. Erdeniz, and J. Done, "Prediction error in reinforcement learning: A meta-analysis of neuroimaging studies," *Neurosci. Biobehav. Rev.*, vol. 37, no. 7, pp. 1297–1310, Aug. 2013.

[44] Y. Niv, N. D. Daw, and P. Dayan, "How fast to work: Response vigor, motivation and tonic dopamine," in *Proc. Adv. Neural Inf. Process. Syst.*, 2005, vol. 60, no. 3, pp. 1019–1026.

[45] Y. Niv, "Reinforcement learning in the brain," *J. Math. Psychol.*, vol. 53, no. 3, pp. 139–154, 2009.

[46] P. Dayan and Y. Niv, "Reinforcement learning: The good, the bad and the ugly," *Current Opinion Neurobiol.*, vol. 18, no. 2, pp. 185–196, Apr. 2008.

[47] M. S. Gazzaniga, R. B. Ivry, G. R. Mangun, and M. S. Steven, *Cognitive Neuroscience: The Biology of the Mind*. New York, NY, USA: W. W. Norton, 2009.

[48] C. A. Kumar, M. Ishwarya, and C. K. Loo, "Formal concept analysis approach to cognitive functionalities of bidirectional associative memory," *Biologically Inspired Cognit. Archit.*, vol. 12, pp. 20–33, Apr. 2015, doi: 10.1016/j.bica.2015.04.003.

[49] R. Shivhare and A. K. Cherukuri, "Three-way conceptual approach for cognitive memory functionalities," *Int. J. Mach. Learn. Cybern.*, vol. 8, no. 1, pp. 21–34, Feb. 2017, doi: 10.1007/s13042-016-0593-0.

[50] S. P. Miller and P. J. Hudson, "Using evidence-based practices to build mathematics competence related to conceptual, procedural, and declarative knowledge," *Learn. Disabilities Res. Pract.*, vol. 22, no. 1, pp. 47–57, Feb. 2007.

[51] M. A. Nielsen, *Neural Networks and Deep Learning*. New York, NY, USA: Determination Press, 2015.

[52] D. A. Levin, Y. Peres, and E. L. Wilmer, *Markov Chains and Mixing Times*. Providence, RI, USA: American Mathematical Society, 2009.

[53] A. B. Tsybakov, *Introduction to Nonparametric Estimation*. New York, NY, USA: Springer, 2009.

[54] L. Zhao, Y. Zhou, H. Lu, and H. Fujita, "Parallel computing method of deep belief networks and its application to traffic flow prediction," *Knowl.-Based Syst.*, vol. 163, pp. 972–987, Jan. 2019.

[55] V. Nair and G. E. Hinton, "Rectified linear units improve restricted Boltzmann machines," in *Proc. 27th Int. Conf. Mach. Learn. (ICML)*, 2010.

[56] C. Dugas, Y. Bengio, F. Bélisle, C. Nadeau, and R. Garcia, "Incorporating second-order functional knowledge for better option pricing," in *Proc. 13th Int. Conf. Neural Inf. Process. Syst.*, 2001, pp. 451–457.

[57] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *Proc. 13th Int. Conf. Artif. Intell. Statist.*, 2010, pp. 249–256.

[58] D. P. Kingma and J. L. Ba, "Adam: A method for stochastic optimization," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2015.

[59] T.-Y. Sun, S.-J. Tsai, Y.-N. Lee, S.-M. Yang, and S.-H. Ting, "The study on intelligent advanced fighter air combat decision support system," in *Proc. IEEE Int. Conf. Inf. Reuse Integr.*, Sep. 2006, pp. 39–44.

[60] K. Dong, H. Huang, C. Huang, and Z. Zhang, "Trajectory online optimization for unmanned combat aerial vehicle using combined strategy," *J. Syst. Eng. Electron.*, vol. 28, no. 5, pp. 963–970, Oct. 2017, doi: 10.21629/jsee.2017.05.14.

[61] F. He and Y. Yao, "Maneuver decision-making on air-to-air combat via hybrid control," in *Proc. IEEE Aerosp. Conf.*, Mar. 2010, pp. 1–6.

[62] 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, doi: 10.5139/ijass.2016.17.2.204.

[63] X. Chen and M. Zhao, "The decision method research on air combat game based on uncertain interval information," in *Proc. 5th Int. Symp. Comput. Intell. Design*, 2012, pp. 456–459.

[64] R. L. Shaw, *Fighter Combat: Tactics and Maneuvering*. Annapolis, MD, USA: Naval Institute Press, 1985.

KAI ZHOU received the B.S. and M.S. degrees in control science and engineering from Air Force Engineering University, China, in 2013 and 2016, respectively. He is currently pursuing the Ph.D. degree. His research interests include cognitive computation and intelligent operational decision of UAVs.

RUIXUAN WEI received the Ph.D. degree from Xi'an Jiaotong University, Xi'an, China. He is currently a Professor with Air Force Engineering University. His research interests include cognitive computation, robotics, intelligent control, and operational decision of UAVs and bio-inspired algorithms.

QIRUI ZHANG received the B.S. and M.S. degrees in control science and engineering from Air Force Engineering University, China, in 2014 and 2017, respectively, where he is currently pursuing the Ph.D. degree in control science and engineering.

ZHUOFAN XU received the B.S. degree in control science and engineering from the Nanjing University of Aeronautics and Astronautics, in 2008, and the M.S. and Ph.D. degrees in control science and engineering from Air Force Engineering University, China, in 2014 and 2019, respectively. He is currently a Lecturer with the National Defence University of the People's Liberation Army.