IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 7 May 2024, accepted 3 June 2024, date of publication 7 June 2024, date of current version 5 July 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3411015

RESEARCH ARTICLE

Large Language Model Guided Reinforcement Learning Based Six-Degree-of-Freedom Flight Control

YANQIAO HAN, MENGLONG YANG, (Member, IEEE), YANG REN, AND WEIZHENG LI
School of Aeronautics and Astronautics, Sichuan University, Chengdu 610065, China

Corresponding author: Menglong Yang (steinbeck@163.com)

This work was supported in part by the National Nature Science Foundation of China under Grant 62271334, and in part by the 173 Key Project under Grant 2019-JCJQ-ZD-342-00.

ABSTRACT As artificial intelligence (AI) technology advances rapidly, its increasing involvement in military defense fosters intelligent air combat domain development. The Intelligent Flight Controller (IFC) is a crucial technology and foundation for intelligent air combat decision-making systems. Controlling 6 Degree-of-freedom (DOF) aircraft in close-to-real-world environments requires an adaptable and dynamic decision-making controller. Most IFC researches focus on simplistic flight trajectory design and validation, while air combat requires aircraft that can perform complex tactical maneuvers. Deep reinforcement learning (DRL) provides a suitable technical paradigm. However, DRL suffers from sparse rewards, insufficient supervisory signals, low sampling efficiency, and slow convergence. In contrast, Large Language Model (LLM) possesses abundant knowledge about the real world, contextual understanding, and reasoning capabilities. By leveraging this, LLM can serve as prior knowledge for DRL, thereby reducing DRL training time. This paper proposes an LLM-guided deep reinforcement learning framework for IFC, which utilizes LLM-guided deep reinforcement learning to achieve intelligent flight control under limited computational resources. LLM provides direct guidance during training based on local knowledge, which improves the quality of data generated in agent-environment interaction within DRL, expedites training, and offers timely feedback to agents, thereby partially mitigating sparse reward issues. Additionally, we present an effective reward function to comprehensively balance the aircraft coupling control to ensure stable, flexible control. Finally, simulations and experiments show that the proposed techniques have good performance, robustness, and adaptability across various flight tasks, laying a foundation for future research in the intelligent air combat decision-making domain.

INDEX TERMS Intelligent flight control, large language model, deep reinforcement learning, 6 DOF aircraft.

I. INTRODUCTION

Marr proposed segmenting complex information processing systems into three levels [1]: computational theory, representation and algorithm, and physical implementation, each addressing "why," "what," and "how" questions, respectively. Machine perception and understanding correspond to the initial two levels, while physical implementation is the bridge linking them to reality; in the absence of this liking, a human can only think but not act. With artificial intelligence (AI) technology development, increasing AI technologies are integrated into military equipment, driving rapid progress in intelligent air combat [2]. IFC is critical in intelligent air combat decision-making systems like the aforementioned physical implementation level. As a complex information processing system, an intelligent air combat decision-making system relies on IFC to connect strategic decision-making and practical flight control actions, serving as the foundation of an intelligent air combat decision-making system. Extensive research has been conducted on IFC [3],

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

[4], mainly categorized into model-based and model-free methods.

Model-based flight control methods include nonlinear model predictive control [5], linear quadratic Gaussian model control [6], [7], [8], backstepping control [9], [10], sliding mode control [11], gain scheduling control [12], nonlinear neural network adaptive control [13], [14], etc. While model-based methods can address flight control issues and achieve objectives to some extent, they heavily rely on accurate mathematical models assumed by the system. In practice, achieving perfect mathematical modeling for flight control systems is challenging, and insufficient understanding may exist [15], which inevitably affects control effectiveness.

Model-free flight control methods include fuzzy system control [16], data-driven control [17], neural network control [18], deep reinforcement learning [19], [20], etc. Unlike model-based control, model-free control does not rely on accurate mathematical models and exhibits adaptability and robustness against uncertain interference factors.

IFC is a critical technology in intelligent air combat decision-making systems. Some studies achieve 3 DOF flight control using lateral overload, normal overload, and roll angle, simplifying flight control into sets of five basic maneuvers for air combat [21]. In some cases, further simplification to seven or six basic maneuvers, respectively [22], [23]. While this approach can achieve a certain level of control effectiveness, it lacks scalability to other flight control tasks. Additionally, using lateral overload, normal overload, and roll angle for three-dimensional flight control does not align with the practical control method of using elevator, aileron, and rudder.

Some studies utilized deep reinforcement learning to achieve end-to-end control of a 6 DOF aircraft, enabling it to perform cruise flights at specified speeds and altitudes [24]. However, this IFC lacked scalability, as it could only execute nominal commands, and the learned flight behavior included a sideslip. Some studies improved this by introducing yaw angle error as an input, effectively reducing sideslip during flight [25]. However, it still faces challenges in achieving agile control of a 6 DOF aircraft.

IFCs achieve basic maneuvering by inputting specified coordinates, but air combat strategies based on these controllers cannot directly control the aircraft's attitude, thus restricting maneuverability [26], [27]. To address this issue, employing deep reinforcement learning to introduce pitch and roll angle errors enhances maneuver flexibility. However, it still struggles to perform complex maneuvers effectively [28].

As mentioned earlier, deep reinforcement learning is well-suited for solving dynamic decision control problems. It provides an adaptable, model-free controller design framework for various objects. It also enables end-to-end integrated control, making it an effective approach for intelligent flight control. However, deep reinforcement learning relies on trial-and-error and continuous interaction between the agent and environment to find optimal strategies without direct guidance. With the emergence of Large Language Models (LLMs) that exhibit emergent and logical reasoning capabilities, we aim to utilize LLM-guided deep reinforcement learning to achieve intelligent flight control under limited computational resources. The main contribution of this paper is as follows:

- We propose a deep reinforcement learning training framework guided by the LLM, providing direct guidance during training. First, we construct a local textual knowledge base using resources like aircraft flight manuals from the Federal Aviation Administration (FAA). During agent-environment interaction, the LLM uses this knowledge base as background context to evaluate agent actions. Incorrect actions are rejected, prompting the agent to try new actions until meeting the criteria. This effectively enhances exploration data quality, accelerates training, and addresses sparse rewards by providing timely feedback.
- We present a practical reward function incorporating roll angle error, yaw angle error, altitude error, and velocity error. This function balances coupled control of the aircraft's ailerons, rudder, elevators, and throttle, and experiments in simulated environments resembling real-world scenarios validate our approach. Compared to other methods, our intelligent flight controller directly manipulates the aircraft's attitude, enabling complex tactical maneuvers. It also demonstrates robustness and extendability to various tactical maneuvers.
- We trained an IFC and conducted experimental flight control simulations using the proposed method. During the initial training guided by LLM, DRL significantly increased learning speed. Simulations of horizontal flights under no-wind and windy conditions, plus tactical maneuver analyses including Looping, Immelmann Turn, and Split S, demonstrated proficient execution by the IFC. The simulation results indicate precise, flexible control and the ability to perform complex maneuvers robustly. LLM enhanced sample data quality, increased supervisory signals, and alleviated sparse rewards during training, improving training convergence speed.

II. BACKGROUNDS

This section describes the background information required for this study.

A. FLIGHT CONTROL PROBLEM STATEMENT

In this research, the F-16 is chosen as the control object for flight controller training [29]. We utilize the open-source, high-fidelity flight dynamics model JSBSim to conduct physical simulations, ensuring they closely approximate real-world conditions. Table 1 presents the primary state variables of the 6 DOF aircraft.

As shown in FIGURE 1, we define the body coordinate system $oxyz$ and the aircraft-carried normal earth-fixed coordinate system $ox_gy_gz_g$ [30]. The body coordinate system $oxyz$ is fixed to the aircraft, with the origin o at the center of mass. Ox points along the longitudinal axis within the symmetry plane, towards the nose. Oy is perpendicular to the

TABLE 1. The state and action variables.

Parameter	Value	Meaning
alt	[0, 18000m]	Sea-level altitude
θ	[-π/2, π/2]	Pitch
ϕ	[-π, π]	Roll
φ	[0, 2π]	Yaw
v	[0, 700m/s]	Calibrated Airspeed (CAS)
vx	[0, 700m/s]	Longitudinal axis airspeed
vy	[0, 700m/s]	Lateral axis airspeed
vz	[0, 700m/s]	Vertical axis airspeed
δ_{th}^{cd}	[0,1]	Throttle
δ_{a}^{cd}	[-1,1]	Aileron
δ_{e}^{cd}	[-1,1]	Elevator
δ_{r}^{cd}	[-1,1]	Yaw

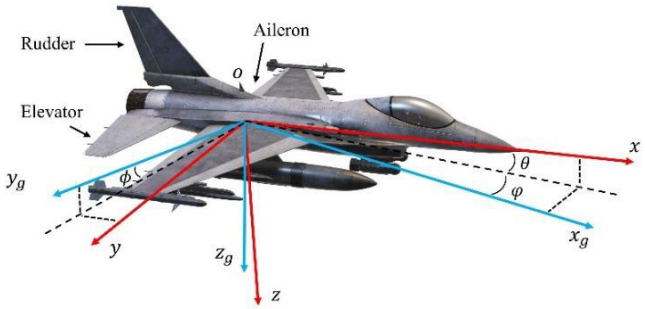


FIGURE 1. The body coordinate system and ground coordinate system.

symmetry plane, pointing to the right wing. Oz is within the symmetry plane, perpendicular to ox, and pointing down. The aircraft-carried normal earth-fixed coordinate system $ox_gy_gz_g$ has origin o at the center of mass, with ox_g in any arbitrarily selected direction within the horizontal plane. oz_g is perpendicular to the horizontal plane and points down due to the lead weight, while the oy_g is determined by the right-hand rule.

The flight control problem for a 6 DOF aircraft can decompose into spatial position movement and attitude control. Spatial position movement control includes longitudinal displacement controlled by the throttle, indirectly achieving lateral and vertical displacements. Spatial attitude control includes rotation about the longitudinal axis, known as roll motion control; rotation about the lateral axis, known as pitch motion control; and rotation about the vertical axis, known as yaw motion control. As shown in FIGURE 1, the pitch angle θ represents the angle between the body coordinate system ox and the horizontal plane. The yaw angle ϕ represents the angle between the body coordinate system ox projection onto the horizontal plane and the aircraft-carried normal earth-fixed coordinate system ox_g. The roll angle φ represents the angle between the plumb plane passing through the body coordinate system ox and the symmetry plane, which is also equivalent to the angle between the oxy plane and the horizontal plane, as shown in Figure 1.

In flight control, the aircraft's roll motion is controlled by δ_a^{cd}, pitch motion by δ_e^{cd}, and yaw motion by δ_r^{cd}. Here, $\{\delta_a^{cd}, \delta_e^{cd}, \delta_r^{cd}\}$ denote the control commands issued to the respective control surfaces of the aircraft rather than the actual angles of these surfaces at that moment.

Furthermore, compared to traditional flight control, which improves aircraft stability and maneuverability, IFC aims to achieve environmental perception, attitude stabilization, and high-performance flight control.

B. PROXIMAL POLICY OPTIMIZATION ALGORITHM

Deep reinforcement learning aims to obtain an optimal policy for an agent through interaction with the environment. At each time step t, the agent starts from the current state s_t, takes action a_t, receives the next state s_{t+1}, and reward information R from the environment. The agent aims to maximize the expected return G_t, represented by the action-state value function Q according to the Bellman equation:

$$Q = \hat{A} + V \tag{1}$$

The advantage function $\hat{A}$ indicates the advantage of a specific action relative to the average in a given state s, while the value function V denotes the value of a given state s.

Since its inception, the Proximal Policy Optimization (PPO) algorithm, has demonstrated outstanding performance, spawning numerous variants and becoming the preferred algorithm for OpenAI in reinforcement learning research. The main idea behind PPO is to limit the magnitude of policy updates to achieve stable and efficient training results. In this study, we utilize the Proximal Policy Optimization with clipping (PPO-clip) algorithm, which comprises two essential components, the Actor and the Critic, implemented using neural networks. The Actor represents the policy function π_θ, mapping states to action distributions, while the Critic represents the value function V_Θ, estimating the expected return under the current policy. The agent interacts with the environment, storing the outcomes in a buffer. At the end of each episode, the Generalized Advantage Estimation (GAE) is employed to calculate the advantage values $\hat{A}_t$ and the expected returns G_t, which are then stored in the buffer.

$$\hat{A}_t = \delta_t + (\gamma\lambda)\,\delta_t + \cdots + (\gamma\lambda)^{T-t+1}\,\delta_{T-1} \tag{2}$$

where

$$\delta_t = r_t + \gamma V(s_{t+1}) - V(s_t) \tag{3}$$

Here λ represents the hyperparameter of GAE, γ is the discount factor, T is the time step at the end of the episode, and t signifies the current time step.

The Actor and Critic is optimized based on the data stored in the buffer. The loss function in the PPO algorithm can be defined as:

$$L = L^{Actor}(\theta) + c_1 L^{Critic}(\Theta) + c_2 S[\pi_\theta] \tag{4}$$

where c_1 and c_2 are hyperparameters used to balance the weights of different loss functions, $L^{Critic}(\Theta)$ is the value loss

function. $S[\pi_\theta]$ is the entropy of the policy used to encourage policy exploration. $L^{Actor}(\theta)$ is the proximal ratio clipping loss employed to restrict the magnitude of policy updates.

$$L^{Actor}(\theta) = \mathbb{E}_t[\min\left(r_t(\theta)\hat{A}_t, CLIP\hat{A}_t\right)] \tag{5}$$

where

$$CLIP = clip\left(r_t(\theta), 1-\varepsilon, 1+\varepsilon\right) \tag{6}$$

$$r_t(\theta) = \frac{\pi_\theta(a_t \mid s_t)}{\pi_{\theta_{old}}(a_t \mid s_t)} \tag{7}$$

measures the action probability ratio between the new and old policies, where data from the buffer is generated by interacting with the environment under the old policy $\pi\_(\theta\_old)$. ε is a hyperparameter controlling clipping magnitude. Employing this loss function for policy optimization prevents excessive updates that could cause instability. The specific form of the value loss $L^{Critic}(\Theta)$ is as follows:

$$L^{Critic}(\Theta) = \frac{1}{2}\mathbb{E}\left[V_\Theta(s_t) - G_t\right]^2 \tag{8}$$

The value of G_t is obtained through Equation(1) and optimized using Equation(4). The PPO algorithm can enhance the Actor's performance, bringing it closer to the optimal.

III. PROPOSED METHOD

In this section, we will elaborate on the deep reinforcement learning training framework guided by LLM and the process and details of applying deep reinforcement learning methods to intelligent flight control.

A. LLM-GUIDED REINFORCEMENT LEARNING FRAMEWORK

Deep reinforcement learning is trial-and-error learning without direct guidance, and the training needs large amounts of data generated by agent-environment interactions [31]. Additionally, it exhibits delayed rewards. The principles and methods cannot be expressed with formulas in flight control tasks, as various flight control commands are interrelated. Large language models (LLMs) possess vast real-world knowledge, contextual understanding, and logical reasoning capabilities. However, directly using LLM to solve flight control problems is challenging. Therefore, we propose using LLM to transform relevant background knowledge from the local knowledge base into action guidance signals during deep reinforcement learning training. This method can better guide agent learning, improve sample quality generated by agent-environment interactions, reduce learning sample requirements, accelerate training, and enhance training stability. However, with limited computational resources, we provide guidance only for initial intelligent flight controller training in this paper. As the training episode increase, LLM is no longer used.

This approach provides a natural language expression of the aircraft's states, goals, and control actions. We expect the LLM to evaluate actions based on the states and goals, using the local knowledge base. Different flight states, goals, and control actions are represented with textual information, denoted as *L_state*, *L_goal*, and *L_action*, respectively. To construct the local knowledge base, publications like the Flight Control Manual from the Federal Aviation Administration (FAA) are primary knowledge sources, which are then embedded for storage in a vector database as the knowledge base. During evaluation, the LLM queries relevant information from the knowledge base as background knowledge.

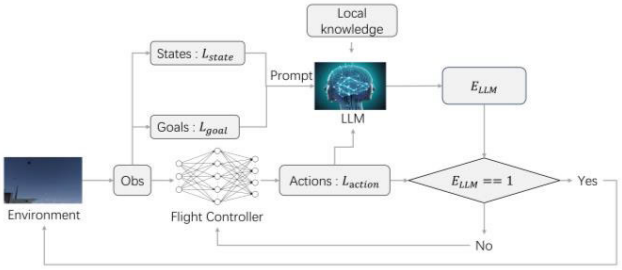


FIGURE 2. LLM-Guided reinforcement learning framework.

The LLM uses a backstep questioning strategy [32], requiring stepping back one step when answering questions, approaching it from a broader or more fundamental perspective. Logically, the core of the problem is contemplated step by step. The LLM will address the following: "The current aircraft state is *L_state*, with flight goal being *L_goal* and flight control command being *L_action*. Is this correct?" Response is ultimately "yes" or "no". This format is chosen because it can be easily converted to "0" or "1", and closed-ended responses tend to perform better than open-ended ones [33].After evaluating each flight control command, the LLM yields:

$$E_{LLM} = F_{LLM_{a1}} * F_{LLM_{a2}} * F_{LLM_{a3}} * F_{LLM_{a4}} \tag{9}$$

If all responses are "yes," the process proceeds; otherwise, the flight control actions are rejected, and agent makes a new one until each flight control action evaluation is "yes." For example, after evaluating each flight control action, if we obtain $E_{LLM} = 1 * 0 * 1 * 1 = 0$, indicating one control action does not comply with the current states and goals, the current action is rejected. During training, we predefine the max episode guided by LLM and stop the LLM guidance once the episode exceeds the max episode. The process and algorithm of this method are summarized in FIGURE 2 and Algorithm 1, respectively.

Based on limited computational resources, we adopted ChatGLM-6B from Zhipu as our baseline large model. In addition to the questions mentioned earlier, the prompt designed for the flight controller in this paper is presented in the appendix.

B. REINFORCEMENT LEARNING BASED FLIGHT CONTROL

This study designs an IFC based on the PPO algorithm and proposes a practical reward function. The reward function incorporates roll angle error, yaw angle error, altitude error,

Algorithm 1 Flight Controller Algorithm with LLM

```
Initialize actor and critic
for episode = 1 to episodes do
    Initialize the replay buffer D
    while D is not full do
        Reset the Environment
        for step = 1 to step_max do
            Get the state s_t
            Sample the action a_t from the policy π_θ(a_t|s_t)
            if episode < episode_target then
                Get the evaluation E_LLM from the LLM
                if E_LLM == 0 then
                    Sample the action a_t from the policy π_θ(a_t|s_t)
                end if
            end if
            Get the next state s_{t+1} and reward r_t and Done from the
            environment
            Store (s_{t+1}, a_t, r_t, Done) into D
            if Reach the target signal then
                Reset the target signal randomly
            end if
            if Done is True then
                Break
            end if
        end for
    end while
    Compute Â and Ĝ and store into D
    for Update times do
        Update actor and critic
    end for
end for
```

and velocity error into the training process, balancing the control of aileron, rudder, elevator, and throttle. By tracking the target roll angle, target yaw angle, target altitude, and target velocity, the corresponding errors are input to the flight controller, which outputs instructions for the aileron, elevator, rudder, and throttle to control the aircraft directly. The Markov Decision Process (MDP) for the aircraft can be defined as a tuple U = (S, A, R, D, P), where S is the observation space, A is the action space, R is the reward function, D is the termination signal, and P is the state transition function. The details are as follows:

1) OBSERVATION AND ACTION SPACES

The design of the observation space and action space are important in terms of learning efficiency and effectiveness. We design the observation space s_t, which is obtained from the flight control environment and consists of two parts. The first part includes the aircraft flight state s_k and comprises state variables describing the aircraft's current flight status and attitude, as listed in Table 1. The second part describes the aircraft's tracking of target signals, including roll angle error, yaw angle error, altitude error, and velocity error: $[\tilde{\phi}-\phi, \tilde{\varphi}-\varphi, \widetilde{alt}-alt, \tilde{v}_x-v_x]$. Here, $\tilde{\phi}, \tilde{\varphi}, \widetilde{alt}$, and $\tilde{v}_x$ represent the target roll angle, target yaw angle, target altitude, and target velocity, respectively.

The action space a_t is $[\delta_a^{cd}, \delta_e^{cd}, \delta_r^{cd}, \delta_{th}^{cd}]$, where δ_a^{cd}, δ_e^{cd}, δ_r^{cd}, and δ_{th}^{cd} represent the control commands for the aileron, elevator, rudder, and throttle, respectively. The aileron, elevator, and rudder can individually control the roll angle, pitch angle, and yaw angle of the aircraft.

2) REWARD FUNCTION

The reward function plays a crucial role in reinforcement learning. In this paper, roll angle error, yaw angle error, altitude error, and velocity error are incorporated into the training process to achieve flight control by tracking target signals while balancing the interrelated control of the aircraft's ailerons, rudder, elevator, and throttle. Based on the Potential-based Reward Shaping (PBRS) technique [34], a potential function consistent with the Gaussian function shape is designed in this paper:

$$\Phi(\Delta) = e^{-\frac{(\Delta)^2}{2\sigma^2}} \tag{10}$$

where Δ represents the error value, and σ^2 is the variance of this potential function. In order to achieve optimal performance of the flight controller, it should not only enable direct control of the aircraft's attitude but also balance various aspects of flight control. The reward function R is designed as follows:

$$R = \sqrt[4]{R_\phi \cdot R_\varphi \cdot R_{alt} \cdot R_{v_x}} + R_{altitude} \tag{11}$$

Here, R_ϕ represents the reward for the roll angle error, where the smaller the absolute value of the roll angle error, the greater the reward obtained.

$$R_\phi = e^{-\frac{(\Delta_\phi)^2}{2\sigma_\phi^2}} \tag{12}$$

The symbol Δ_ϕ represents the roll angle error, defined as $\tilde{\phi}-\phi$. σ_ϕ is the variance of the roll angle error reward, which can be adjusted to control the fluctuation range of the roll angle error. In this study, σ_ϕ is set to 0.25 during training, resulting in an error fluctuation range of $\pm 45°$. R_φ, R_alt, and $R\_(v\_x)$ represent the rewards for roll angle error, altitude error, and velocity error, respectively. The design of their reward functions follows the same pattern as R_ϕ. Specifically, they are formulated as follows:

$$R_\varphi = e^{-\frac{(\Delta_\varphi)^2}{2\sigma_\varphi^2}} \tag{13}$$

$$R_{alt} = e^{-\frac{(\Delta_{alt})^2}{2\sigma_{alt}^2}} \tag{14}$$

$$R_{v_x} = e^{-\frac{(\Delta_{v_x})^2}{2\sigma_{v_x}^2}} \tag{15}$$

The symbols Δ_φ, Δ_alt, and $\Delta\_(v\_x)$ represent errors in yaw angle, altitude, and velocity, respectively. σ_φ, σ_alt, and $\sigma\_(v\_x)$ denote the corresponding reward variances. During training, σ_φ is set to 1, resulting in a $\pm 3°$ error fluctuation range for yaw angle; σ_alt is set to 3.3, resulting in a $\pm$10m error fluctuation range for altitude; and $\sigma\_(v\_x)$ is set to 8.23, resulting in a $\pm$25m/s fluctuation range for velocity.

$R_{altitude}$ represents the altitude reward, which is based on the aircraft's flying altitude. A safe flying altitude $Safe_{altitude}$ is set to ensure routine flight without collisions.

$R_{altitude}$ is calculated from the current altitude Alt_{now} and the safe flying altitude $Safe_{altitude}$. This reward encourages the aircraft to maintain a safe altitude during flight, and its specific form is as follows:

$$R_{altitude} = \begin{cases} 0, & Alt_{now} > Safe_{altitude} \\ -clip\left(\dfrac{Alt_{now}}{Safe_{altitude}}, 0, 1\right), & otherwise \end{cases} \tag{16}$$

3) DONE SIGNAL DESIGN

Done represents the termination signal indicating the end of a task. The termination signal in this paper is defined as follows:

$$Done = Done_{signal} \vee Done_G \vee Done_{Altitude} \vee Done_{step} \tag{17}$$

In this study, the task involves tracking a target signal. The system checks whether the target signal has been reached within a specified time frame. If the target signal is reached, the system reinitializes the target randomly, indicating the task is not yet completed. However, the task is completed if the target signal is not reached within the specified time frame. The decision process is as follows:

$$Done_{signal} = \begin{cases} 0, & Reach\ the\ target\ signal \\ 1, & Unreach\ the\ target\ signal \end{cases} \tag{18}$$

Additionally, the task will be terminated if extreme flight conditions occur based on the aircraft's performance, including exceeding the maximum load factor G_{max} and surpassing the altitude limit $Altitude_{max}$. The specific conditions are as follows:

$$Done_G = \begin{cases} 0, & G < G_{max} \\ 1, & G \geq G_{max} \end{cases} \tag{19}$$

$$Done_{Altitude} = \begin{cases} 0, & alt < Altitude_{max} \\ 1, & alt \geq Altitude_{max} \end{cases} \tag{20}$$

To prevent the aircraft from endlessly tracking the target signal, a maximum number of steps per task episode has been set as $Step_{max}$. When this limit is exceeded, terminate the task. The specific condition is as follows:

$$Done_{step} = \begin{cases} 0, & step < Step_{max} \\ 1, & step \geq Step_{max} \end{cases} \tag{21}$$

The flight controller is trained using the PPO algorithm, as shown in Algorithm 1. Before training begins, initialize the environment to a valid initial state, and the actor and critic are initialized. At the start of each episode, clear the replay buffer $\mathcal{D}$. Reset the environment before each task and the actor provides control instructions a_t based on the current state s_t, where the main influence on the control instruction output is the error values of the target signals in state s_t. After a fixed period, check the current aircraft's proximity to the target signal. If it reaches the target, a new target signal is randomly generated; otherwise, the episode of the task ends. Considering the large action space of the 6 DOF aircraft, the randomness of the target signal is set to gradient ascent to enhance the training speed and stability. Subsequently, the generated $\{s_{t+1}, a_t, r_t, Done\}$ is stored in the replay buffer. At the end of each episode, compute the advantage value $\hat{A}$ and G_t and store in the replay buffer. When the actor interacts with the environment to generate sufficient data, optimization of the actor and critic is performed using Equation 1.

The feasibility of the algorithm proposed in this paper depends on the LLM guidance and DRL for solving flight control problems. Indeed, if an LLM can infer and solve tasks based on known information, it exhibits guided feasibility. The scalability theory suggests that LLM exhibits emergent capabilities as model parameters increase [35], including in-context learning (ICL), instruction following, and chain of thought. These emergent capabilities support LLM in solving respective tasks based on data formatted in natural language descriptions after obtaining natural language task descriptions. Meanwhile, the local knowledge base in this paper is derived from specialized textual data, ensuring the feasibility of LLM guidance. DRL is suitable for addressing dynamic decision control problems, with its inherent adaptability, real-time responsiveness, and robustness ensuring the feasibility of solving flight control problems. Subsequent tactical maneuver flight simulations conducted in high-fidelity environments validate the algorithm's feasibility.

The algorithm proposed in this paper is based on deep reinforcement learning, where the design of the reward function determines the agent's objectives and outcomes. In this paper, each reward function is formulated as a Gaussian function, with the error variance adjusted to ensure rewards are within the desired error range. As the goal of DRL is to maximize reward returns, the design of this reward function ensures that after convergence, the agent's control remains within the desired error range, making flight states close to or equal to the flight target, thereby ensuring the stability of the intelligent flight controller. Subsequent simulation experiments conducted in varying wind conditions further validate the stability of this intelligent flight controller.

IV. EXPERIMENTAL ANALYSIS

A. EXPERIMENTAL PLATFORM

JSBSim is a multi-platform, open-source, object-oriented flight dynamics model (FDM) written in C++ [36]. Essentially, FDM defines the motion of aircraft, rockets, and so on under various control mechanisms, forces, and natural phenomena. The Johns Hopkins University Applied Physics Lab (JHU-APL) gym environment based on JSBSim was used in the AlphaDogfight Trials (ATD) competition conducted by DARPA [37]. Our experimental platform is a deep reinforcement learning environment developed based on JSBSim.

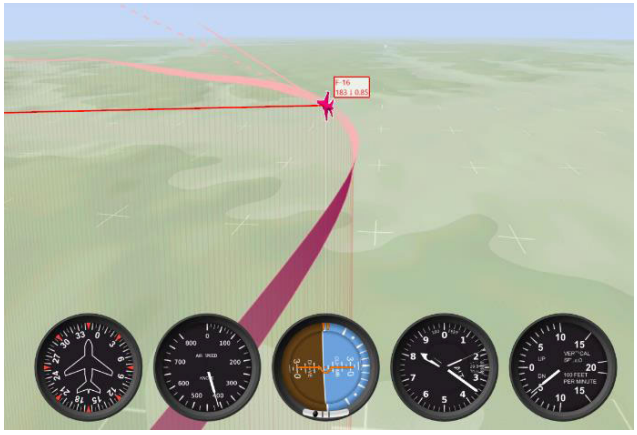


FIGURE 3. Deep reinforcement learning simulation environment based on JSBSim 1.

FIGURE 4. Deep reinforcement learning simulation environment based on JSBSim 2.

The rendering of this environment is shown in the following figure.

B. EXPERIMENTAL TRAINING AND RESULTS

The experimental simulations are based on the method proposed earlier. In the initial training stages, LLM is introduced for action guidance following Algorithm 1. FIGURE 5 demonstrates that the LLM-guided flight controller achieves local convergence with fewer samples in fewer steps. This indicates that LLM-guided action enhances sample data quality during training, reducing required samples for learning.

The aircraft is penalized for flying below the safe altitude due to the reward function design. As shown in FIGURE 5, the average reward per episode plateaus at 0, indicating the agent learns to fly above the safe altitude. However, it has yet to learn to track target signals for flight control. The simulation result shown in FIGURE 6 supports this conclusion.

Through comparison of simulation results in FIGURE 6 and FIGURE 7, we see that the IFC based on LLM guidance performs significantly better after the same number of training steps than the traditional DRL IFC, and the IFC without LLM guidance does not learn to fly above a safe altitude, and its attitude control is weaker.

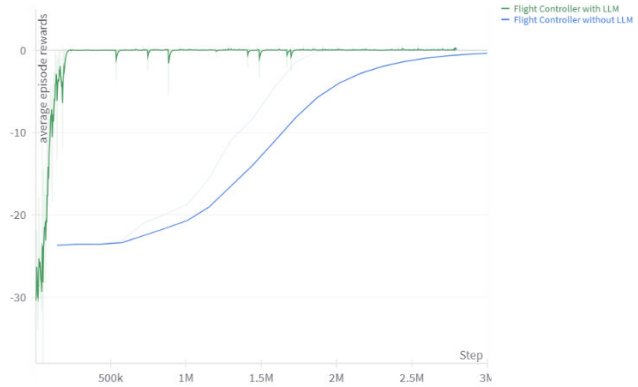


FIGURE 5. The average episode rewards with and without large model guidance.

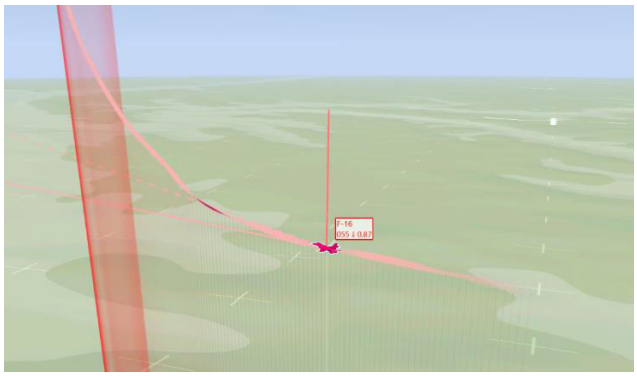


FIGURE 6. The simulation result with LLM guidance.

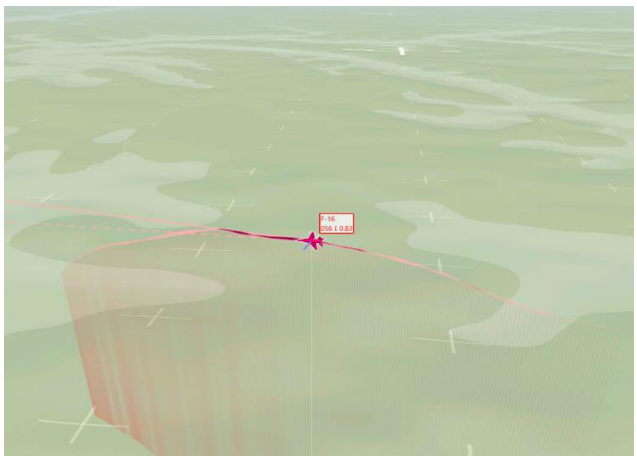


FIGURE 7. The simulation result without LLM guidance.

Subsequently, training was conducted according to Algorithm 2. Considering the action space of the 6 DOF aircraft, the target signal randomness was set to gradient ascent during training, allowing the agent to learn from easy to challenging tasks, improving training stability and

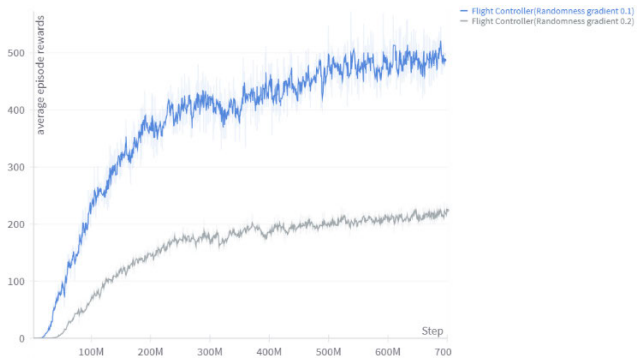


FIGURE 8. Average episode rewards with different randomness gradient.

speed. As shown in FIGURE 8, the agent with a target signal randomness gradient of 0.1 is easier to learn due to the smaller initial randomness of the target signal, effectively accelerating training. Additionally, from FIGURE 8, training with lower randomness ultimately achieved higher average rewards per episode. This is because lower randomness leads to more frequent target signal achievement within the same steps, ultimately resulting in higher average rewards per episode.

Algorithm 2 Flight Controller Algorithm

```
Initialize actor and critic
for episode = 1 to episodes do
    Initialize the replay buffer D
    while D is not full do
        Reset the Environment
        for step = 1 to step_max do
            Get the state s_t
            Sample the action a_t from the policy π_θ(a_t|s_t)
            Get the next state s_{t+1} and reward r_t and Done from the
            environment
            Store (s_{t+1}, a_t, r_t) into D
            if Reach the target signal then
                Reset the target signal randomly
            end if
            if Done is True then
                Break
            end if
        end for
    end while
    Compute Â and Ĝ and store into D
    for Update times do
        Update actor and critic
    end for
end for
```

Both training approaches ultimately achieved tracking of the target signal, as depicted in FIGURE 9. With increasing average reward per episode, the average error in tracking the target signal gradually converged to a range close to zero. Additionally, the results indicate that training with a smaller randomness gradient achieves faster and smoother convergence. Moreover, FIGURE 9(c)(d) suggests that this approach performs better in tracking the target roll and yaw angles, enhancing training stability and effectiveness.

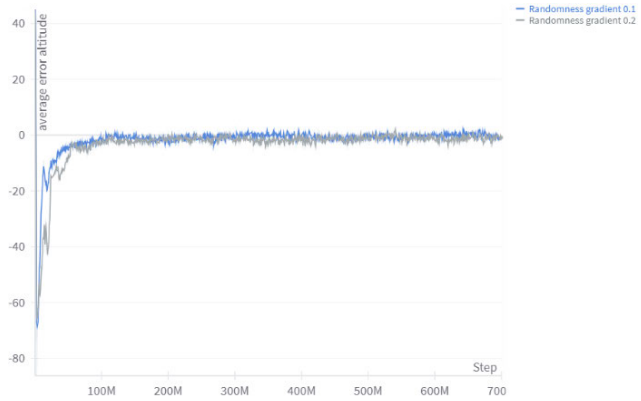
Based on subsequent analysis of various tactical maneuvers and flight simulation experiments under different wind disturbances, this flight controller demonstrates stable control over flight. It maintains stable flight states close to and consistent with the flight target, whether during different tactical maneuvers or disturbances. These experiments confirm the robustness and stability of the flight controller.

1) HORIZONTAL FLIGHT

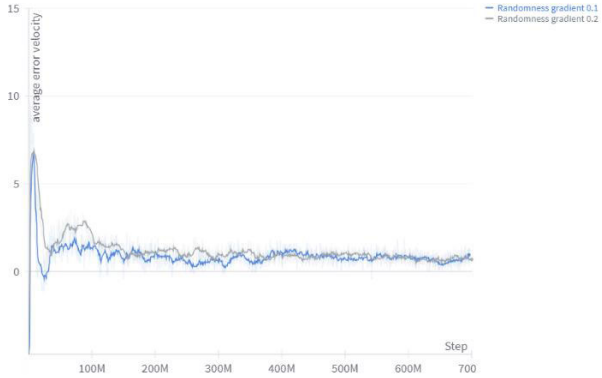
Subsequently, based on the abovementioned training results, we conducted simulation experiments of flight controller for flight control. Firstly, horizontal flight simulations were run. The target error commands input to the flight controller were $[\Delta\_alt = 0\,\text{m}, \Delta\_\varphi = 0\,\text{rad}, \Delta\_(v\_x) = 0\ m/s, \Delta\_\phi = 0\ rad]$, which means the flight controller tracked targets are the initial states at the start of the horizontal flight. Two simulations were performed, one with and one without wind. The results for horizontal flight are shown in FIGURE 10(a). There are altitude, velocity, yaw angle, and roll angle fluctuations, but all stay within reasonable ranges. The aircraft achieves horizontal flight. FIGURE 10(b) depicts the altitude curve during simulation, showing it remains stable near the initial altitude. Altitude fluctuates by ±6m, within the height error range set during training.

FIGURE 10(c) depicts the yaw angle variation curve during simulation, exhibiting stability around the initial yaw angle with fluctuations after 10 seconds, within ±1.5°. This range falls within the reasonable yaw angle error range set during training. FIGURE 10(d) illustrates the variation curve of the aircraft's v_x; there are some fluctuations before 50 seconds, then stabilizing near the initial velocity with fluctuations within ±5m, within the velocity error range set during training. FIGURE 10(e) presents the roll angle variation curve, stabilizing around the initial roll angle with fluctuations within ±15°, consistent with the reasonable roll angle error range set during training. By comparing simulation data curves with and without wind, it is observed that even with strong wind, the flight controller maintains control for horizontal flight. It demonstrates comparable attitude and altitude control to that without wind, with a slight speed increase, as shown in FIGURE 10(d), indicating a higher stable flight speed than the initial speed due to strong wind influence. This speed adjustment maintains flight stability, with deviations falling within a reasonable error range. This simulation experiment illustrates that the trained flight controller has stable control and is robust against environmental changes and uncertainties.

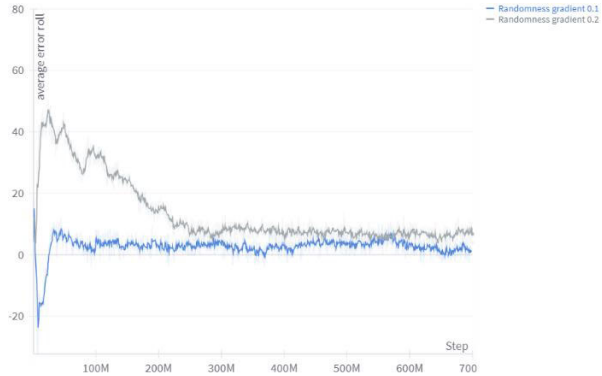
Simulation experiments will further validate the flight controller's ability to perform various maneuvers and assess its performance in executing multiple tactical maneuvers. Air combat demands a range of maneuvers, which are specialized flight actions by aircraft to achieve different goals. The following simulations will focus on conducting Looping,



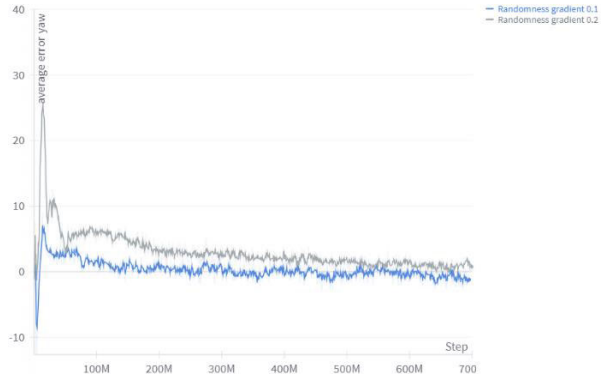
(a) Average error altitude



(b) Average error velocity



(c) Average error roll



(d) Average error yaw

FIGURE 9. Training error result with different randomness gradient.

Immelmann Turn, and Split S maneuvers. During simulation experiments, the aircraft's target signal will be directly set and processed with the current flight status to derive the target signal error. Finally, this error will be input into the flight controller for aircraft control.

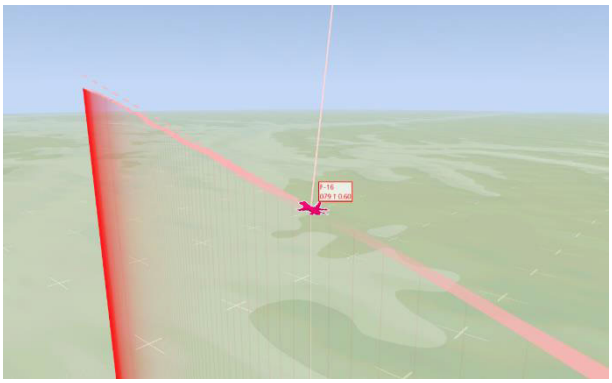
2) LOOPING

The looping maneuver can be regarded as a "spiral" in the vertical direction, which the flight controller cannot accomplish by tracking a fixed target signal. Therefore, during the looping simulation, we will adjust the tracked target signal according to the maneuver's attitude. The aircraft's initial altitude is set to 5575m, initial v_x is 280m/s, initial yaw angle is 22°, and initial roll angle is 0°. The specific target signal commands are as follows:

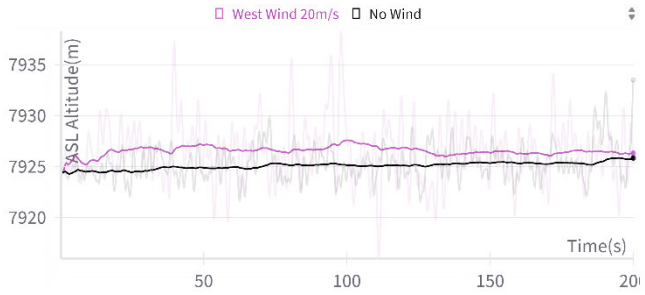
$$\begin{aligned} Target_{signal} = [&\left[8623m, 22°, \frac{80m}{s}, 0°\right], \\ &\left[5575m, 22°, \frac{80m}{s}, 180°\right], \\ &\left[5575m, 22°, \frac{280m}{s}, 0°\right]] \\ Interval_{step} = [50, 140] \end{aligned}$$

The command specifies that for time steps 0-50, the $Target_{signal} = [8623m, 22°, 80m/s, 0°]$; for time steps 50-140, $Target_{signal} = [5575m, 22°, 80m/s, 180°]$; and for time steps greater than 140, $Target_{signal} = [5575m, 22°, 280m/s, 0°]$. The contents of $Target_{signal}$ represent the target altitude, yaw angle, velocity, and roll angle, respectively.

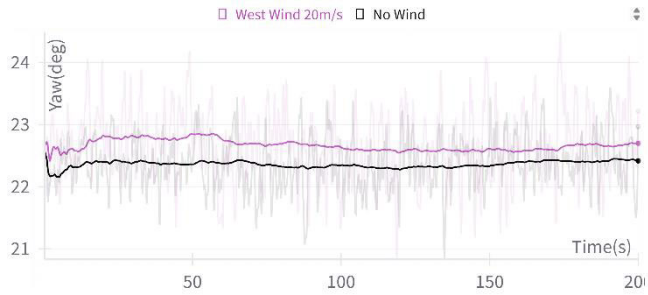
FIGURE 11 presents simulation results for the looping maneuver. As shown in FIGURE 11(a), the aircraft successfully executes the maneuver. FIGURE 11(b)(d) depict altitude and v_x curves during the maneuver. Observe that as the aircraft ascends from the bottom to the top of the loop, velocity gradually decreases with increasing altitude, indicating the conversion of kinetic to potential energy. Upon reaching peak altitude, potential energy converts back to kinetic, causing speed to increase until stabilizing at initial altitude and velocity upon maneuver completion. FIGURE 11(c) illustrates the yaw angle, showing a sudden change during the 1/4 loop due to attitude variation. Similarly, another abrupt change occurs during the 3/4 loop to return to the initial orientation. FIGURE 11(e) displays the roll angle, exhibiting multiple abrupt changes. A comparison with yaw angle and altitude curves reveals that during the 1/4 to 3/4 loop transition, the roll angle rapidly transitions from 0° to 180° and then back to 0°, indicating inverted flight. Furthermore, the roll angle undergoes multiple transitions between 180° and −180° during intervals between main transitions. This behavior arises from setting the roll angle range as [−180°, 180°], where both −180° and 180° represent inverted attitudes—consequently, control error fluctuations during inverted flight cause rapid transitions between −180° and 180°.



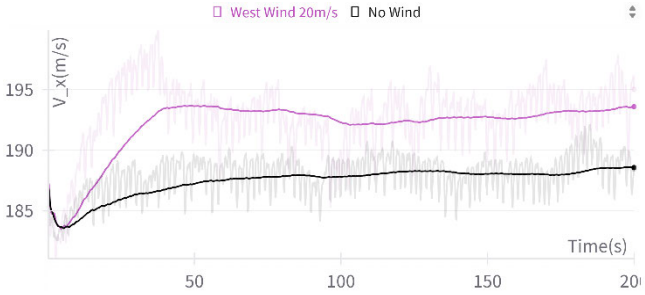
(a) Horizontal flight simulation result



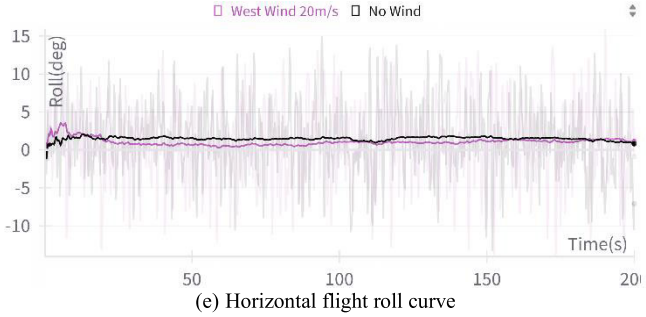
(b) Horizontal flight altitude curve



(c) Horizontal flight yaw curve



(d) Horizontal flight v_x curve



(e) Horizontal flight roll curve

FIGURE 10. **Horizontal flight simulation.**

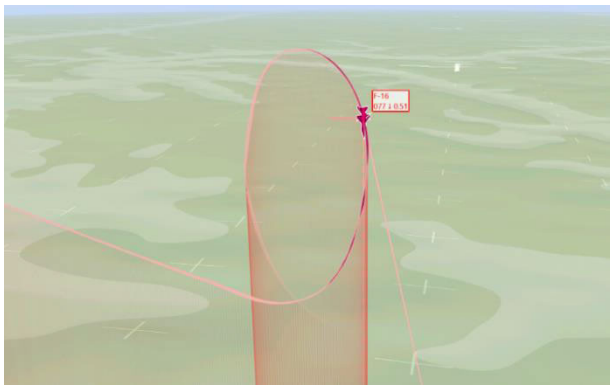
3) IMMELMANN TURN

The Immelmann Turn maneuver, named after the renowned German flying ace Max Immelmann of World War I, combines a loop and a half-roll. It starts with horizontal flight, followed by pulling into a loop, completing a half-roll, and returning to horizontal flight. It converts kinetic to potential energy in combat while altering the heading by 180°. In our simulation experiments, the aircraft's initial state was identical to that in the loop maneuver simulation. The specific target signal commands are as follows:

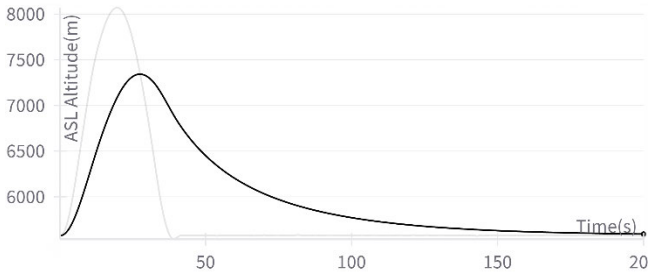
$$Target_{signal} = [[5575m, 22°, 280m/s, 0°],$$
$$\left[8623m, 22°, \frac{80m}{s}, 0°\right],$$
$$\left[5575m, 22°, \frac{80m}{s}, 180°\right],$$
$$\left[6794m, 22°, \frac{80m}{s}, 180°\right],$$
$$\left[7403m, 202°, \frac{180m}{s}, 0°\right],$$
$$[7220m, 202°, 180m/s, 0°]]$$

$$Interval_{step} = [10, 60, 90, 120, 160]$$

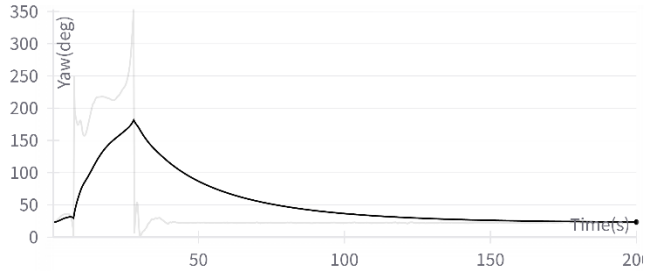
FIGURE 12(a) illustrates the execution of the Immelmann Turn by the aircraft. FIGURE 12(b)(d) depict the altitude and speed curves during the maneuver. A precise observation from comparison shows the conversion of kinetic energy into potential energy after the maneuver, with the aircraft stabilizing at the target altitude and speed for horizontal flight. FIGURE 12(c) displays the yaw angle curve during the Immelmann Turn. A sudden change in yaw angle occurs as the aircraft passes through a 1/2 semicircle due to attitude



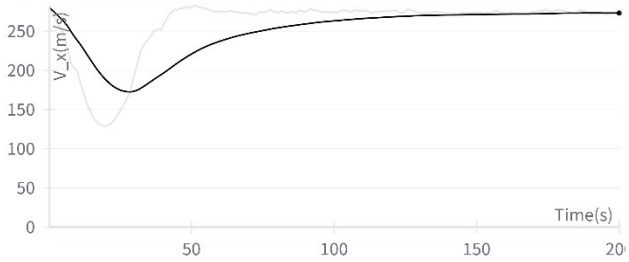
(a) Looping simulation result



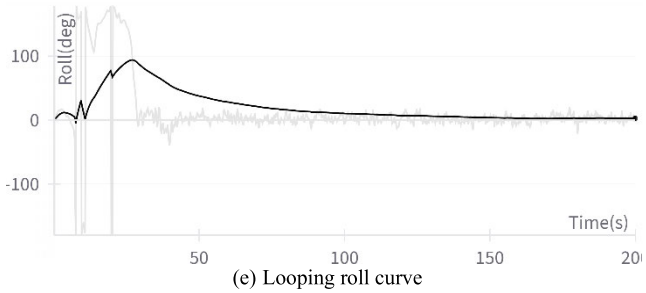
(b) Looping altitude curve



(c) Looping yaw curve



(d) Looping v_x curve



(e) Looping roll curve

FIGURE 11. **Looping simulation.**

changes, followed by a return to the target yaw angle. Upon completion, the heading changes by 180°. FIGURE 12(e) presents the roll angle curve during the Immelmann Turn. The first rapid change in roll angle to −180° occurs when the aircraft passes through a 1/2 semicircle, resulting from attitude changes. At this point, the aircraft enters an inverted flight state. Subsequently, upon reaching the apex and returning to horizontal flight, the roll angle quickly returns to 0°. Similar to the Looping maneuver, there are multiple instances of rapid roll angle changes between the two significant transitions stemming from the constraints of the simulation setup.

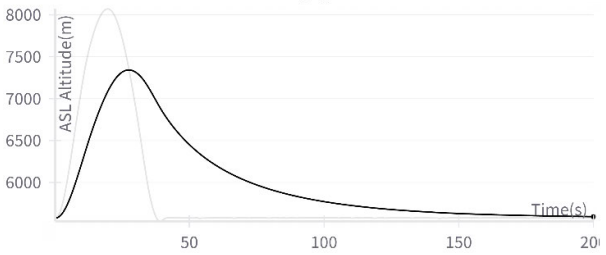
4) SPLIT S

The Split S maneuver is often used for escape when having altitude superiority. Its maneuvering process is essentially the reverse of the Immelmann Turn, converting altitude potential energy into speed kinetic energy while making a 180° turn. When simulating this maneuver, the aircraft's initial altitude is 7924m, initial v_x velocity is 188m/s, initial yaw angle is 22°, and initial roll angle is 0°. The target signal commands for completing the simulation experiment are as follows:

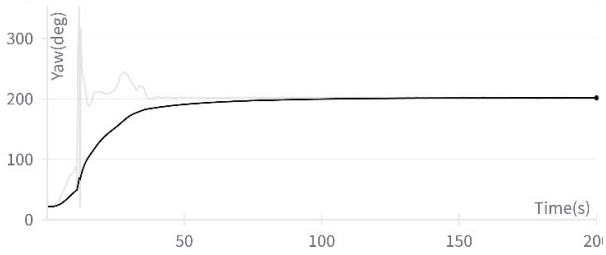
$$\begin{aligned} Target_{signal} = &[[7924m, 22°, 188m/s, 180°], \\ &[1828m, 202°, -412m/s, 180°], \\ &[1828m, 202°, 188m/s, 180°], \\ &[6095m, 202°, 188m/s, 0°]] \\ Interval_{step} = &[20, 45, 75] \end{aligned}$$

FIGURE 13(a) depicts the simulation results of the Split S maneuver, showing successful execution. FIGURE 13(b)(d) represent the altitude and v_x curve, respectively. As the altitude decreases, potential energy is converted to kinetic energy, eventually in horizontal flight post-maneuver. FIGURE 13(c) illustrates the yaw angle curve, showing a sudden change midway, eventually stabilizing at a 180°

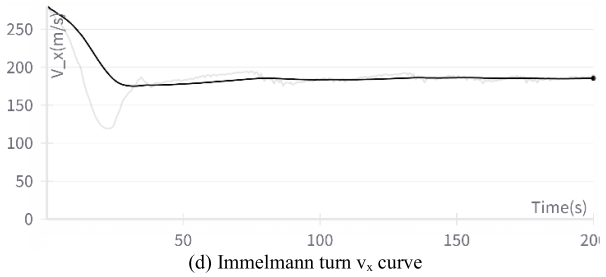
(a) Immelmann turn simulation result



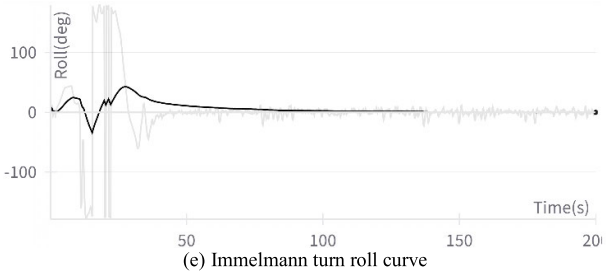
(b) Immelmann turn altitude curve



(c) Immelmann turn yaw curve

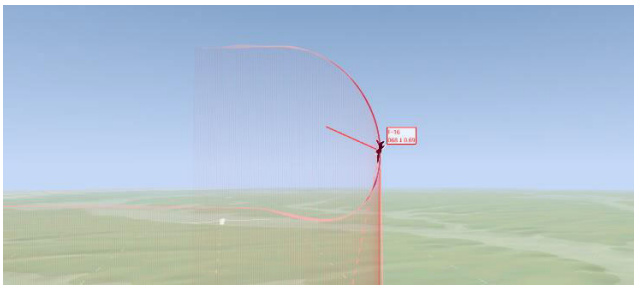


(d) Immelmann turn v_x curve

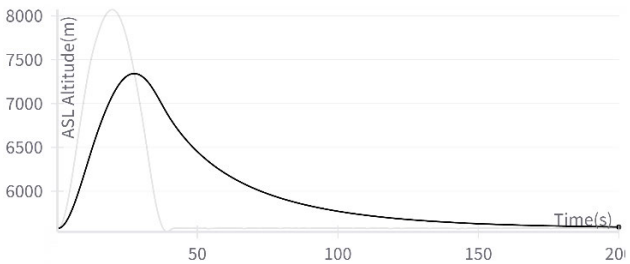


(e) Immelmann turn roll curve

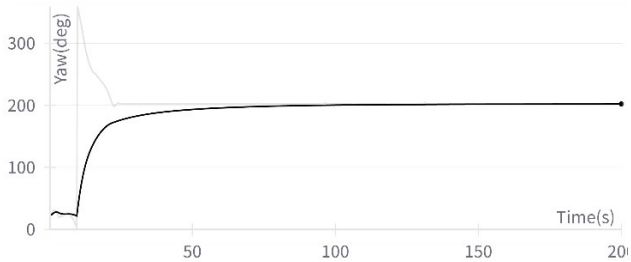
FIGURE 12. Immelmann turn simulation.



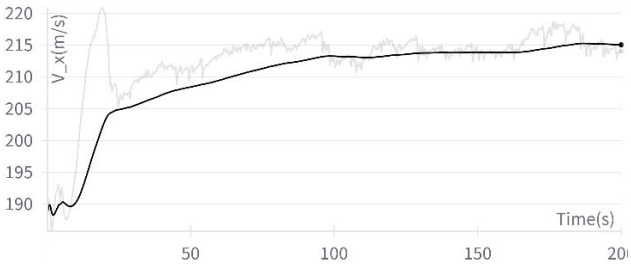
(a) Split S simulation result



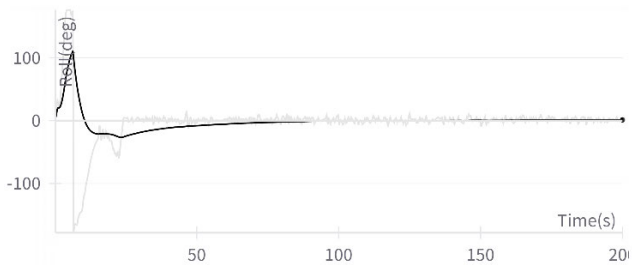
(b) Split S altitude curve



(c) Split S yaw curve



(d) Split S v_x curve



(e) Split S roll curve

FIGURE 13. Split S simulation.

change in heading. FIGURE 13(e) displays the roll angle curve, demonstrating variation during the maneuver. Like the Looping maneuver discussed earlier, the roll angle experiences abrupt transitions between 180° and −180° due to maneuver nature.

The simulation results of various tactical maneuvering actions demonstrate that the IFC trained using the proposed

method can achieve precise aircraft control. Moreover, it allows direct control of the aircraft's attitude, enhancing control flexibility and enabling the execution of complex tactical maneuvers. Balancing control commands improves flight control and is robust, enabling extension to a wide range of tactical maneuvering actions.

V. CONCLUSION

In this study, we propose a deep reinforcement learning training framework based on the guidance of the Large Language Model (LLM), which provides direct guidance during deep reinforcement learning training. The LLM guidance effectively enhances the quality of data generated during the agent's exploration, accelerates training, mitigates sparse rewards, and promptly provides feedback to the agent. However, 6 DOF fixed-wing aircraft control is complex. Due to limited computational resources, the logical reasoning ability of the benchmark LLM used in this study is lacking, and its guidance based on local knowledge for this complex task has deficiencies, limiting effectiveness [38]. Additionally, practical reward functions are proposed to achieve precise and flexible air control and equip it with robustness and complex task ability. These incorporate roll angle error, yaw angle error, altitude error, and velocity error to balance the coupling relationships between various flight controls. Moreover, to improve flight controller training efficiency and effectiveness, a gradient for target error randomness is introduced during training. In future research, we will focus on combining LLM and deep reinforcement learning for decision-making, aiming to develop a scalable, intelligent air combat decision-making system. We expect to develop an intelligent air combat decision-making system that is scalable.

APPENDIX

LLM BACKSTEP PROMPT

As an expert in flight operations, you excel in using the backward questioning strategy, carefully considering and evaluating whether the provided flight operation instructions are correct. A backward questioning strategy is a thinking approach aimed at logically dissecting the core of a problem step by step and providing answers. This strategy requires us to ''step back'' when facing a specific problem, asking and pondering from a broader or more fundamental perspective. The purpose of doing so is to help us gain a deeper understanding of the problem, thereby providing a better answer to the original question.

Strategy: Identification of the core issue: Firstly, identify the core of the problem. Based on the known information, provide a concise and professional response to the flight control instructions needed under the current aircraft flight goals.

Judgment of the problem: Compare the flight control instructions given in the original problem with those identified in the core issue recognition and determine whether they match.

Based on the judgment of the problem, provide a final answer to the original question: ''Yes'' or ''No'' (answer requirement).

<Known Information> {{context}} <Known Information> <Original Question> {{question}} <Original Question>.

REFERENCES

[1] D. Marr, *Vision: A Computational Investigation Into the Human Representation and Processing of Visual Information*. Cambridge, MA, USA: MIT Press, 2010, doi: 10.7551/mitpress/9780262514620.001.0001.s

[2] M. W. Byrnes, ''Nightfall: Machine autonomy in air-to-air combat,'' *Air Space Power J.*, vol. 28, pp. 48–75, Jan. 2014.

[3] F. Santoso, M. A. Garratt, and S. G. Anavatti, ''State-of-the-art intelligent flight control systems in unmanned aerial vehicles,'' *IEEE Trans. Autom. Sci. Eng.*, vol. 15, no. 2, pp. 613–627, Apr. 2018.

[4] S. A. Emami, P. Castaldi, and A. Banazadeh, ''Neural network-based flight control systems: Present and future,'' *Annu. Rev. Control*, vol. 53, pp. 97–137, Jan. 2022.

[5] L. Singh and J. Fuller, ''Trajectory generation for a UAV in urban terrain, using nonlinear MPC,'' in *Proc. Amer. Control Conf.*, Jun. 2001, pp. 2301–2308.

[6] S. A. Bezvitniy, ''Control systems for unmanned aerial vehicles,'' in *Proc. Wave Electron. Appl. Inf. Telecommun. Syst. (WECONF)*, Jun. 2019, pp. 1–4.

[7] F. Santoso, G. Egan, and M. Liu, ''H_2 and H_∞ robust autopilot synthesis for longitudinal flight of a special unmanned aerial vehicle: A comparative study,'' *IET Control Theory Appl.*, vol. 2, no. 7, pp. 583–594, Jul. 2008.

[8] F. Santoso, M. Liu, and G. K. Egan. (2007). *Linear Quadratic Optimal Control Synthesis for a UAV*. [Online]. Available: https://api.semanticscholar.org/CorpusID

[9] J. Farrell, M. Sharma, and M. Polycarpou, ''Backstepping-based flight control with adaptive function approximation,'' *J. Guid., Control, Dyn.*, vol. 28, no. 6, pp. 1089–1102, Nov. 2005.

[10] J. R. Azinheira and A. Moutinho, ''Hover control of an UAV with backstepping design including input saturations,'' *IEEE Trans. Control Syst. Technol.*, vol. 16, no. 3, pp. 517–526, May 2008.

[11] S. Zhang, Q. Wang, K. He, and Y. Shao, ''An improved dynamic surface control law and its application in post-stall maneuvers,'' *Acta Aerodynamica Sinica*, vol. 35, no. 5, pp. 718–726, 2017.

[12] R. G. Hernández-García and H. Rodríguez-Cortés, ''Transition flight control of a cyclic tiltrotor UAV based on the gain-scheduling strategy,'' in *Proc. Int. Conf. Unmanned Aircr. Syst. (ICUAS)*, Jun. 2015, pp. 951–956.

[13] A. J. Calise and R. T. Rysdyk, ''Nonlinear adaptive flight control using neural networks,'' *IEEE Control Syst. Mag.*, vol. 18, no. 6, pp. 14–25, Dec. 1998.

[14] W. Gu, K. P. Valavanis, M. J. Rutherford, and A. Rizzo, ''UAV model-based flight control with artificial neural networks: A survey,'' *J. Intell. Robotic Syst.*, vol. 100, nos. 3–4, pp. 1469–1491, Dec. 2020, doi: 10.1007/s10846-020-01227-8.

[15] Q. Wang, W. Qian, and D. Ding, ''A review of unsteady aerodynamic modeling of aircrafts at high angles of attack,'' *Acta Aeronauticate Astronautica Sinica*, vol. 37, no. 8, p. 2331, 2016.

[16] A. Sarabakha, C. Fu, E. Kayacan, and T. Kumbasar, ''Type-2 fuzzy logic controllers made even simpler: From design to deployment for UAVs,'' *IEEE Trans. Ind. Electron.*, vol. 65, no. 6, pp. 5069–5077, Jun. 2018.

[17] E. Ju, J. Won, J. Lee, B. Choi, J. Noh, and M. G. Choi, ''Data-driven control of flapping flight,'' *ACM Trans. Graph.*, vol. 32, no. 5, pp. 1–12, Sep. 2013.

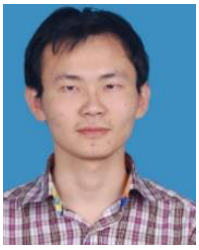
[18] V. Artale, M. Collotta, C. Milazzo, G. Pau, and A. Ricciardello, ''An integrated system for UAV control using a neural network implemented in a prototyping board,'' *J. Intell. Robotic Syst.*, vol. 84, nos. 1–4, pp. 5–19, Dec. 2016, doi: 10.1007/s10846-015-0324-x.

[19] J. Hwangbo, I. Sa, R. Siegwart, and M. Hutter, ''Control of a quadrotor with reinforcement learning,'' *IEEE Robot. Autom. Lett.*, vol. 2, no. 4, pp. 2096–2103, Oct. 2017.

[20] Y. Zhu, S. Lian, W. Zhong, and W. Meng, ''A reinforcement learning method for quadrotor attitude control based on expert information,'' in *Proc. 8th Int. Conf. Autom., Control Robot. Eng. (CACRE)*, Jul. 2023, pp. 281–286.

[21] W. Ma, "Research on air combat game decision based on deep reinforcement learning," Sichuan Univ. Softw. Eng., Sichuan, China, Tech. Rep., 2021.
[22] Y. Li, J. Shi, W. Zhang, and W. Jiang, "Maneuver decision of UCAV in air combat based on deep reinforcement learning," *J. Harbin Inst. Technol.*, vol. 53, no. 12, pp. 33–41, 2021.
[23] L. Li, Z. Yang, Z. Sun, G. Zhan, H. Piao, and D. Zhou, "Generation method of autonomous evasive maneuver strategy in air combat," in *Proc. 22nd Int. Conf. Control, Autom. Syst. (ICCAS)*, Nov. 2022, pp. 360–365.
[24] S. Zhang, X. Du, J. Xiao, J. Huang, and K. He, "Reinforcement learning control for 6 DOF flight of fixed-wing aircraft," in *Proc. 33rd Chin. Control Decis. Conf. (CCDC)*, May 2021, pp. 5454–5460.
[25] S. Zhang, X. Du, J. Xiao, and J. Huang, "Fixed-wing aircraft 6-DOF flight control based on deep reinforcement learning," *J. Command Control*, vol. 8, no. 2, pp. 179–188, Jun. 2022.
[26] P. Heidlauf, A. Collins, M. Bolender, and S. Bak, "Verification challenges in F-16 ground collision avoidance and other automated maneuvers," in *Proc. ARCH@ADHS*, vol. 54, 2018, pp. 208–217. [Online]. Available: https://api.semanticscholar.org/CorpusID:53082578
[27] H. Shin, J. Lee, H. Kim, and D. H. Shim, "An autonomous aerial combat framework for two-on-two engagements based on basic fighter maneuvers," *Aerosp. Sci. Technol.*, vol. 72, pp. 305–315, Jan. 2018.
[28] J. Chai, W. Chen, Y. Zhu, Z.-X. Yao, and D. Zhao, "A hierarchical deep reinforcement learning framework for 6-DOF UCAV air-to-air combat," *IEEE Trans. Syst., Man, Cybern., Syst.*, Jan. 2023, pp. 1–13.
[29] J. H. Bae, H. Jung, S. Kim, S. Kim, and Y.-D. Kim, "Deep reinforcement learning-based air-to-air combat maneuver generation in a realistic environment," *IEEE Access*, vol. 11, pp. 26427–26440, 2023.
[30] S. T. Wu, *Flight Control Systems*, 2nd ed. Beijing, China: Beihang University Press, 2013.
[31] Q. Liu, J. Qu, Z. Zhang, S. Zhong, Q. Zhou, P. Zhang, and J. Xu, "A survey on deep reinforcement learning," *Chin. J. Comput.*, vol. 41, no. 1, pp. 1–27, Jan. 2018.
[32] H. S. Zheng, S. Mishra, X. Chen, H.-T. Cheng, E. H. Chi, Q. V. Le, and D. Zhou, "Take a step back: Evoking reasoning via abstraction in large language models," 2023, *arXiv:2310.06117*.
[33] Y. Du, O. Watkins, Z. Wang, C. Colas, T. Darrell, P. Abbeel, A. Gupta, and J. Andreas, "Guiding pretraining in reinforcement learning with large language models," in *Proc. Int. Conf. Mach. Learn.*, no. 346, 2023, pp. 8657–8677. [Online]. Available: https://api.semanticscholar.org/CorpusID:256846700
[34] A. Ng, D. Harada, and S. J. Russell, "Policy invariance under reward transformations: Theory and application to reward shaping," in *Proc. Int. Conf. Mach. Learn.*, 1999, pp. 278–287. [Online]. Available: https://api.semanticscholar.org/CorpusID:5730166
[35] J. Kaplan, S. McCandlish, T. Henighan, T. B. Brown, B. Chess, R. Child, S. Gray, A. Radford, J. Wu, and D. Amodei, "Scaling laws for neural language models," 2020, *arXiv:2001.08361*.
[36] J. Berndt, "JSBSim: An open source flight dynamics model in C++," in *Proc. AIAA Model. Simul. Technol. Conf. Exhib.*, Aug. 2004, ch. 4923, doi: 10.2514/6.2004-4923.
[37] A. P. Pope, J. S. Ide, D. Micovic, H. Diaz, D. Rosenbluth, L. Ritholtz, J. C. Twedt, T. T. Walker, K. Alcedo, and D. Javorsek, "Hierarchical reinforcement learning for air-to-air combat," in *Proc. Int. Conf. Unmanned Aircr. Syst. (ICUAS)*, Jun. 2021, pp. 275–284.
[38] S. Arora and A. Goyal, "A theory for emergence of complex skills in language models," 2023, *arXiv:2307.15936*.

YANQIAO HAN received the bachelor's degree from Sichuan University, China, where he cultivated a strong foundation in engineering and theoretical knowledge. He is currently pursuing the master's degree with the School of Aeronautics and Astronautics, Sichuan University. His research interests include machine learning and reinforcement learning, with a particular focus on the control of multi-agent systems and intelligent perception.



MENGLONG YANG (Member, IEEE) received the B.S. degree from the School of Chemical Engineering, Sichuan University, in 2005, and the M.S. degree from the School of Computer Science and Engineering, Sichuan University, in 2008. He is currently an Associate Professor with the School of Aeronautics and Astronautics, Sichuan University. He is also an Engineer with Wisesoft Company Ltd. He has published more than 20 journal articles. His research interests include computer vision, pattern recognition, and machine learning.

YANG REN received the bachelor's degree from Sichuan University, China, where she is currently pursuing the master's degree. Her research interests include computer vision and machine learning, with a particular focus on object detection, few-shot learning, and generative learning.

WEIZHENG LI is currently pursuing the master's degree with the School of Aeronautics and Astronautics, Sichuan University, China. He is immersed in the dynamic and evolving fields of deep learning and reinforcement learning, with a particular emphasis on intelligent perception and multi-agent control systems.