Chinese Journal of Aeronautics, (2024), xxx(xx): xxx–xxx

Chinese Society of Aeronautics and Astronautics
& Beihang University

Chinese Journal of Aeronautics

cja@buaa.edu.cn
www.sciencedirect.com

FULL LENGTH ARTICLE

Loyal wingman task execution for future aerial combat: A hierarchical prior-based reinforcement learning approach

Jiandong ZHANG [1], Dinghan WANG [1], Qiming YANG [*], Zhuoyong SHI, Longmeng JI, Guoqing SHI, Yong WU

School of Electronics and Information, Northwestern Polytechnical University, Xi'an 710129, China

Received 20 November 2023; revised 2 January 2024; accepted 26 February 2024

KEYWORDS

Beyond-visual-range;
Loyal wingmen;
Hierarchical prior-augmented proximal policy optimization;
Unmanned aerial vehicles;
Warfare

Abstract In modern Beyond-Visual-Range (BVR) aerial combat, unmanned loyal wingmen are pivotal, yet their autonomous capabilities are limited. Our study introduces an advanced control algorithm based on hierarchical reinforcement learning to enhance these capabilities for critical missions like target search, positioning, and relay guidance. Structured on a dual-layer model, the algorithm's lower layer manages basic aircraft maneuvers for optimal flight, while the upper layer processes battlefield dynamics, issuing precise navigational commands. This approach enables accurate navigation and effective reconnaissance for lead aircraft. Notably, our Hierarchical Prior-augmented Proximal Policy Optimization (HPE-PPO) algorithm employs a prior-based training, prior-free execution method, accelerating target positioning training and ensuring robust target reacquisition. This paper also improves missile relay guidance and promotes the effective guidance. By integrating this system with a human-piloted lead aircraft, this paper proposes a potent solution for cooperative aerial warfare. Rigorous experiments demonstrate enhanced survivability and efficiency of loyal wingmen, marking a significant contribution to Unmanned Aerial Vehicles (UAV) formation control research. This advancement is poised to drive substantial interest and progress in the related technological fields.

1. Introduction

In modern aerial combat, the coordinated operation between the lead and wingman aircraft is especially crucial. The coordinated BVR combat of the lead and wingman belongs to intelligent air combat and is widely adopted in actual air battles due to its numerous advantages. This paper focuses on the research of autonomous air combat decision-making for loyal wingman aircraft in intelligent BVR combat and discusses fea-

* Corresponding author.
E-mail address: yangqm@nwpu.edu.cn (Q. YANG).
[1] These authors contributed equally to this work and should be considered co-first authors.
Peer review under responsibility of Editorial Committee of CJA.

Production and hosting by Elsevier

https://doi.org/10.1016/j.cja.2024.03.009
1000-9361 © 2024 Chinese Society of Aeronautics and Astronautics. Production and hosting by Elsevier Ltd.

Please cite this article in press as: ZHANG J et al. Loyal wingman task execution for future aerial combat: A hierarchical prior-based reinforcement learning approach, *Chin J Aeronaut* (2024), https://doi.org/10.1016/j.cja.2024.03.009

sible intelligent coordination methods for the lead-wingman formation centered on this. We analyze the advantages and disadvantages of existing intelligent air combat algorithms, categorizing the promising intelligent air combat algorithms into those based on traditional intelligent algorithms and those based on reinforcement learning algorithms.[1]

Traditional intelligent algorithms have been extensively applied in aerial combat simulation. Xue et al.[2] utilized a genetic fuzzy tree approach to address the challenges of modeling complex aerial combat processes and the high real-time requirements of air battles. Huang et al.[3] proposed a performance evaluation method based on Relevance Vector Machines to enhance the capability of air combat performance assessment for fighter jets. Meng et al.[4] introduced an air combat target UAV maneuver intention recognition algorithm based on Bayesian theory to resolve one-on-one close-range maneuver decision-making. Furthermore, Guo et al.[5] developed an algorithm for one-on-one aerial combat based on Bayesian probability models and shadow graphs, which analyzed combat elements, predicted air combat situations, and autonomously made decisions. Liu et al.[6] presented an approach to BVR air combat stealth maneuver control based on sliding mode control, incorporating electronic support measures. However, the maneuver tactics considered by this method are relatively simplistic. Chen et al.[7] described an optimal decision-making method for UAVs based on Dynamic Bayesian Networks, which relies heavily on expert experience. Tan et al.[8] proposed an air refueling method based on an improved Artificial Fish Swarm (AFS) algorithm, a research direction closely associated with combat logistics, which is not strongly related to the research presented in this paper. Wu et al.[9] introduced a maneuver decision-making method based on fuzzy inference, yet this method suffers from the need for precise construction of situation assessment functions to quantify assessment indicators, which are closely related to maneuver decision-making. Zhong et al.[10] offered an adaptive nested particle swarm optimization algorithm and based on it, designed an air defense deployment model, providing a reference for the study of air defense deployment and operational decision-making. However, traditional algorithms exhibit difficulties in achieving smooth transitions between various task modules in the entire BVR combat process, leading to issues with stability.

With the growing computational power of computers, reinforcement learning algorithms based on large sample data have become mainstream. Methods based on reinforcement learning can be divided into single-agent reinforcement learning algorithms, multi-agent reinforcement learning algorithms, and hierarchical reinforcement learning algorithms.

Single-agent reinforcement learning algorithms are extensively studied in the field of air combat. Yang et al.[11] built an autonomous maneuver decision model for close-range UAV combat based on the Deep Q Network-Prioritized Experience Replay (DQN-PER) algorithm and proposed a "basic adversarial training" method, but the study's aircraft model was limited to three Degree of Freedom (3-DOF), which is insufficient for realistic simulation. Zhang et al.[12] researched air combat maneuver decision-making based on an improved DQN, incorporating Long Short-Term Memory (LSTM) units. However, extensive use of LSTM places high demands on the computer, unlike the more lightweight Gated Recurrent Unit (GRU). Yang et al.[13] studied autonomous air combat maneuver decisions for UAVs based on the Deep Deterministic Policy Gradient (DDPG) algorithm, addressing the problem that DQN cannot handle continuous action spaces. Zhang et al.[14] proposed a maneuver decision algorithm based on final reward estimation and Proximal Policy Optimization, solving the issue of poor maneuver decision performance. Wang et al.[15] studied the Pursuer-Evader (PE) problem using alternating frozen play and deep reinforcement learning algorithms, but the PE problem is somewhat outdated in modern air combat. Li et al.[16] proposed a Parallel Self-Play Soft Actor-Critic (PSP-SAC) algorithm to improve the generalization performance of UAV control decisions. Li et al.[17] introduced an autonomous maneuver decision model based on an expert-intelligence soft actor-critic algorithm, utilizing expert experience to reconstruct the experience pool. However, this method has strict requirements for expert experience data, which is generally difficult to obtain. Zhang et al.[18] studied the discrete missile launch action and continuous maneuver action modeling problem based on the PPO algorithm. Zhang et al.[19] developed a suitable deep neural network online maneuver decision-making model for the engineering implementation of intelligent decision-making for dual-aircraft close-range air combat maneuvers based on deep reinforcement learning. Shan and Zhang[20] proposed an air combat agent training method based on self-game to solve the problem that it is difficult to build a high-level air combat opponent with a single intelligence training method. Hui et al.[21] proposed a step-by-step no-fly method based on "predictive correction guidance - pre-training the roll angle guidance model based on supervised learning - further upgrading the roll angle guidance model based on reinforcement learning" Research framework of intelligent guidance for area-circuit flying in order to solve the problem of high-speed aircraft flying around uncertain no-fly zones. Jiang et al.[22] used deep reinforcement learning to complete BVR air combat 1v1 decisions, yet this method has limited scalability. Hu et al.[23] based on expert knowledge assistance and the introduction of imitation learning, used the DQN algorithm to complete close-range air combat, however, this method involved human-in-the-loop during training and also faced difficulties in acquiring expert experience. Zhou et al.[24] studied fighter maneuver decisions based on the Twin Delayed Deep Deterministic Policy Gradient (TD3) algorithm, but traditional TD3 algorithms are difficult to apply to multi-task complex decisions and have convergence issues in high-fidelity complex task environments. Kong et al.[25] modeled close-range air combat as a State-Adversarial Markov Decision Process (SA-MDP) and proposed a State-Adversarial Deep Deterministic Policy Gradient (SA-DDPG) algorithm for generating air combat maneuvers. However, this method used inverse reinforcement learning during training, highlighting the problem of constructing an efficient reward function. Wu et al.[26] combined the DQN algorithm with supervised learning methods, proposing a combat model more effective than traditional decision models based on knowledge engineering. Lee and Kim[27] employed a deep reinforcement learning approach to execute Combat Unmanned Aerial Vehicle (CUAV) tasks and accomplished autonomous missile evasion.

Hierarchical reinforcement learning algorithms address complex tasks by decomposing them into several sub-tasks, employing a divide-and-conquer approach. Pope et al.[28] employed multiple soft actor-critic network models at the

Please cite this article in press as: ZHANG J et al. Loyal wingman task execution for future aerial combat: A hierarchical prior-based reinforcement learning approach, *Chin J Aeronaut* (2024), https://doi.org/10.1016/j.cja.2024.03.009

low level to complete the entire close-range air combat dogfight. Zhang et al.[29] achieved autonomous decision-making for the full air combat process through the improved Option-Critic framework, though the simulation environment remained two-dimensional. Kong et al.[30] proposed a Hierarchical Multi-Agent Reinforcement Learning algorithm (HMARL) and applied it to a multi-agent close-range air combat scenario. This algorithm enables the formation to exhibit effective cooperative behavior in both symmetric and asymmetric scenarios. Zhou et al.[31] introduced a hierarchical command and control architecture composed of a single high-level agent operating in a dynamic environment and multiple low-level reinforcement learning agents. The proposed model demonstrates the advantages of flexibility and coordinated control. Yuan et al.[32] designed a novel Hierarchical Goal-Guided Learning (HGGL) approach that combines with traditional non-policy deep reinforcement learning algorithms to empower unmanned aerial vehicles with the ability to evade a series of air-to-air missiles. Wang et al.[33] introduced a hierarchical multi-agent reinforcement learning algorithm that incorporates macro-actions and human expertise for autonomous decision-making of aircraft. Chai et al.[34] addressed the within-visual-range air-to-air combat problem in six Degree of Freedom (6-DOF) dynamic scenarios using a combination of hierarchical reinforcement learning and fictitious self-play methods. Qian et al.[35] proposed a novel and efficient three-level hierarchical decision-making framework called H3E, which incorporates expert knowledge. It provides various strategies for high-fidelity one-on-one beyond-visual-range air combat games.

Overall, existing technologies and methods face many challenges. This paper aims to resolve several key issues through innovative algorithms and methods.

(1) Interpretability and control authority transfer in multi-agent systems

Current multi-agent reinforcement learning algorithms lack interpretability and interaction with human pilots, making it difficult for pilots to determine when to transfer control to artificial intelligence in tightly coordinated combat environments. Moreover, existing particle swarm optimization algorithms fail to provide necessary fine-grained control in large-scale scenarios.

(2) Task execution in high-fidelity complex environments

In 6-DOF complex simulation environments, current algorithms struggle to efficiently control loyal wingman aircraft for complex tasks such as target search, lock-on, and relay guidance, with slow training processes and sometimes even failing to converge.

(3) Stability and robustness of task-specific algorithms

The current algorithms lack stability and robustness when performing specific tasks; for instance, if the target is lost during the lock-on process, the algorithm stops and must switch to search mode. When approaching the edge of the detection zone, frequent switching by the algorithm leads to performance degradation.

(4) Lack of dependence on and adaptability to sensors

Traditional reinforcement learning methods need to be retrained due to changes in aircraft sensory equipment (such as radar), increasing the complexity and cost of algorithm deployment.

(5) Challenges in designing reward functions

Constructing a detailed and meaningful reward function is highly challenging, as it needs to accurately reflect the multi-faceted requirements of complex tasks.

To address these issues, this paper proposes a series of innovative strategies.

(1) Collaboration and interpretability in multi-agent systems

By efficiently decomposing the BVR coordination problem of the lead-wingman, we not only simplify the tasks for the lead aircraft, enabling it to perform simpler operations at different stages, but also enhance the system's interpretability, ensuring that control is pilot-driven at critical moments. Additionally, the proposed framework strengthens the wingman's autonomous decision-making ability, overall enhancing the intelligent level and operational granularity of the formation.

(2) Application of hierarchical reinforcement learning

We introduce a dual-layer hierarchical reinforcement learning architecture that processes high-level tactical decisions and specific control commands through high-level and low-level mapping, effectively accelerating the learning process in complex environments.

(3) Enhanced algorithm robustness and stability

To overcome the instability of traditional algorithms during specific task transitions, this paper proposes and constructs a "prioritized training, deprioritized execution" zone model for the first time, smoothing the task transition process and allowing for automatic reacquisition even when the target is lost.

(4) Reduction in sensor specificity

We optimize the construction of state observations, no longer relying on specific sensor parameters but instead using a more general ratio method, improving the adaptability of the model.

(5) Design of efficient reward functions

To tackle the challenge of constructing reward functions, we integrate multiple factors to design a precise, reasonable, and effective reward system to guide the algorithm's decision-making process and training direction.

The paper is organized as follows: Section 2 describes the problem and preparatory work, providing the 6-DOF dynamics model of the unmanned wingman, the missile dynamics and guidance rate model, the zone model, and the relay guidance model. Section 3 presents the state-action space for the low-

Please cite this article in press as: ZHANG J et al. Loyal wingman task execution for future aerial combat: A hierarchical prior-based reinforcement learning approach, *Chin J Aeronaut* (2024), https://doi.org/10.1016/j.cja.2024.03.009

level control model and the high-level control model, the reward function, and the termination conditions. Section 4 introduces the proposed low-level control algorithm, hierarchical algorithm, and hierarchical prior algorithm. Section 5 presents and analyzes the experimental results. Finally, we present the research conclusions of this paper and a future lead-wingman cooperative combat model that offers numerous advantages.

2. Problem description and preparatory work

2.1. Scene description and assumptions

In this paper, we utilize a high-fidelity combat environment and operate under the assumption that the lead aircraft launches a missile and then ceases radar operations to maintain radio silence. The loyal wingman is tasked with detecting the target and guiding the missile launched by the lead aircraft. The missile employed is of medium-to-long-range with semi-active radar homing guidance, using a proportional navigation method. During the terminal guidance phase of the missile, there is no need for radar illumination. It is assumed that the information about the missile launched by the lead aircraft can be transmitted to the wingman via a data link, thus the wingman has global knowledge of the missile information.

2.2. Loyal unmanned wingman and missile model

2.2.1. 6-DOF model for UAV dynamic response

Within the 6-DOF framework, it is imperative to encompass a comprehensive range of factors to accurately capture the dynamic response of UAV. These factors include the physical parameters of the aircraft, the propulsion system, aerodynamic characteristics, and the control mechanisms in place. The true airspeed vector $\boldsymbol{V}$ of the aircraft often deviates from the longitudinal axis of the fuselage due to external influences such as wind speed. To resolve the fuselage's longitudinal velocity $\boldsymbol{V}_x$, a two-step projection process is necessary, involving the sideslip angle β_0 and the angle of attack α_0, as illustrated in Fig. 1.

From a dynamics perspective, a thorough analysis of the forces acting on the aircraft has been conducted. Changes in the aircraft engine thrust are mapped onto the variation of the throttle control C_t, while aerodynamic alterations are correlated with changes in the continuous control variables of the ailerons, elevators, and rudder, denoted as C_a, C_e and C_r respectively. These four control variables mirror the real-world scenarios of aircraft control velocities, roll, pitch, and yaw, involving the aircraft's inertial characteristics, rotational torque, the thrust produced by the engines and associated torques, and the forces and moments resulting from aerodynamic effects.

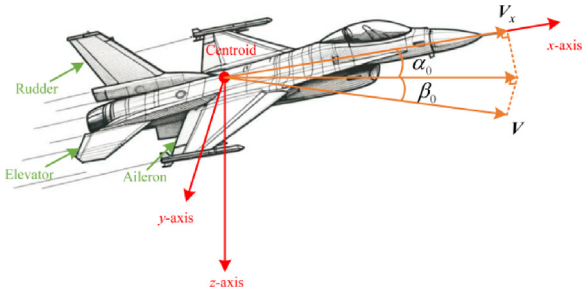


Fig. 1 6-DOF dynamics elements schematic.

Due to the constraints of this paper's length, we represent the aircraft's dynamic equations with a set of simplified variables. The accelerations fixed in the body-frame coordinate system are given by Eq. (1):

$$\begin{bmatrix}\dot{V}_x\\ \dot{V}_y\\ \dot{V}_z\end{bmatrix}=\begin{bmatrix}V_x\\ V_y\\ V_z\end{bmatrix}\times\begin{bmatrix}p\\ q\\ r\end{bmatrix}+\frac{1}{m}\begin{bmatrix}F_x\\ F_y\\ F_z\end{bmatrix}+\boldsymbol{C}_{\mathrm{g}}^{\mathrm{b}}\begin{bmatrix}0\\ 0\\ g\end{bmatrix} \tag{1}$$

In Eq. (1), V_x, V_y, V_z represent the velocities along the x, y, z axes of the body-fixed coordinate system respectively; p, q, r are the angular velocities about the x, y, z axes; F_x, F_y, F_z denote the external forces acting along the x, y, z axes; m is the mass of the aircraft, g is the acceleration due to gravity; and $\boldsymbol{C}_{\mathrm{g}}^{\mathrm{b}}$ is the transformation matrix from the ground coordinate system to the body-fixed coordinate system. The angular accelerations in the body-fixed coordinate system are given by Eq. (2):

$$\begin{bmatrix}\dot{p}\\ \dot{q}\\ \dot{r}\end{bmatrix}=\boldsymbol{I}^{-1}\left(-\begin{bmatrix}p\\ q\\ r\end{bmatrix}\times\boldsymbol{I}\begin{bmatrix}p\\ q\\ r\end{bmatrix}+\begin{bmatrix}M_x\\ M_y\\ M_z\end{bmatrix}\right) \tag{2}$$

In the aforementioned equation, $\boldsymbol{I}$ stands for the inertia matrix of the aircraft, encapsulating the mass distribution relative to the body-fixed coordinate system axes. The terms M_x, M_y, M_z correspond to the external torques along the x, y, z axes respectively. The integration of the aforementioned variables yields the aircraft's state quantities, which are crucial for capturing the dynamics of flight.

2.2.2. Missile dynamics and guidance law model

In the inertial coordinate system, the kinematic equations of missile motion are given by Eq. (3):

$$\begin{cases}\dot{x}_{\mathrm{m}}=V_{\mathrm{m}}\cos\theta_{\mathrm{m}}\cos\varphi_{\mathrm{m}}\\ \dot{y}_{\mathrm{m}}=V_{\mathrm{m}}\cos\theta_{\mathrm{m}}\sin\varphi_{\mathrm{m}}\\ \dot{z}_{\mathrm{m}}=V_{\mathrm{m}}\sin\theta_{\mathrm{m}}\end{cases} \tag{3}$$

where $(x_{\mathrm{m}}, y_{\mathrm{m}}, z_{\mathrm{m}})$ represent the missile's coordinates in the inertial frame; $(v_{\mathrm{m}}, \theta_{\mathrm{m}}, \varphi_{\mathrm{m}})$ denote the missile's velocity, flight path pitch angle, and yaw angle respectively, each being a function of the flight time. The particle dynamics equations of the missile in the trajectory coordinate system are provided by Eq. (4):

$$\begin{cases}\dot{V}_{\mathrm{m}}=g(n_x-\sin\theta_{\mathrm{m}})\\ \dot{\varphi}_{\mathrm{m}}=\frac{g}{V_{\mathrm{m}}}n_y\cos\theta_{\mathrm{m}}\\ \dot{\theta}_{\mathrm{m}}=\frac{g}{V_{\mathrm{m}}}(n_z-\cos\theta_{\mathrm{m}})\end{cases} \tag{4}$$

where n_x, n_y, n_z are the missile's lateral control overloads in the directions of velocity, yaw, and pitch respectively, derived through the proportional navigation method.

Please cite this article in press as: ZHANG J et al. Loyal wingman task execution for future aerial combat: A hierarchical prior-based reinforcement learning approach, *Chin J Aeronaut* (2024), https://doi.org/10.1016/j.cja.2024.03.009

The missile guidance employs a proportional guidance law, with the lateral control overloads in the yaw and pitch directions defined by Eq. (5):

$$\begin{cases} n_y = \frac{K_y V_m}{g} \cos\theta_m \dot{\beta}_m \\ n_z = \frac{K_z V_m}{g} \dot{\varepsilon}_m + \cos\theta_m \end{cases} \tag{5}$$

where β_m and ε_m represent the Line-of-Sight (LOS) angular deviations of the missile in yaw and pitch respectively, while $\dot{\beta}_m$ and $\dot{\varepsilon}_m$ are their time derivatives. K_y and K_z are guidance coefficients in y, z missile control plane.

$$\begin{cases} \beta_m = \arctan\left(\frac{r_y}{r_x}\right) \\ \varepsilon_m = \arctan\left(\frac{r_z}{\sqrt{r_x^2 + r_y^2}}\right) \end{cases} \tag{6}$$

the LOS vector, defined as the range vector $\boldsymbol{r}$, is given by $r_x = x_t - x_m$, $r_y = y_t - y_m$, $r_z = z_t - z_m$, where (x_t, y_t, z_t) are the target's position coordinates, and the magnitude is defined by $L = \sqrt{r_x^2 + r_y^2 + r_z^2}$. The derivatives of LOS angles with respect to time are presented in Eq. (7):

$$\begin{cases} \dot{\beta}_m = \frac{(\dot{r}_y r_x - r_y \dot{r}_x)}{(r_x^2 + r_y^2)} \\ \dot{\varepsilon}_m = \frac{(r_x^2 + r_y^2)\dot{r}_z - r_z(\dot{r}_x r_x + \dot{r}_y r_y)}{L^2 \sqrt{r_x^2 + r_y^2}} \end{cases} \tag{7}$$

When a loyal wingman radar illuminates both the missile and the enemy target, it can guide the missile toward the intended target.

2.3. Zone model

The zone model is the key to "prior-based training, prior-free execution" of our target lock-on model, and it enables loyal wingman to have automatic target reacquisition capability. It is also crucial for avoiding frequent switching between target search and target locking models under critical conditions, thereby enhancing system stability. Assuming that the loyal wingman aircraft lacks offensive capabilities (missile launch), the threat posed by the wingman to the target is solely in its role as a guide for the missiles launched by the lead aircraft. Consequently, the threat to the wingman from the target is significantly greater. In light of this characteristic, we consider the worst-case scenario, where the constructed zone model is based only on the wingman's own position, without considering the target's heading. (When the threat disparity is not substantial, it is common to consider the target's heading, as the area of interest may change significantly when our aircraft is tail-chasing or head-on with the target.)

The zone model is established based on the radar detection zone (the area encompassed by the deep red sector-shaped contour), covering the loss zone (prior zone), safe zone, danger zone, and termination zone (both inside and outside the sector), shown as Fig. 2.

Since the zone model is benchmarked to the radar detection zone, we introduce two definitions related to distance and angle ratios:

(1) Define the distance ratio F_D as the ratio of the target's distance within the area to the detection radius of our aircraft's radar; the range for the area distance ratio is defined as [0, 1.5].

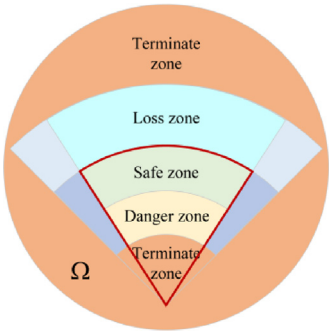


Fig. 2 Zone model.

(2) Define the angle ratio F_A as the ratio of the angle between the target and the wingman's velocity vector to the half of radar detection angle; the range for the zone angle ratio is defined as [−1.5, 1.5].

The angle ratio F_A is signed, with a positive value indicating that the target is to the left of the wingman's velocity vector, and a negative value indicating the right.

These definitions represent relative ratios and therefore, variations in radar parameters (such as different detection distances and angles) will not affect the target lock-on model. Accordingly, the distance and angle ratios for the different zones in this paper are defined as shown in Table 1.

The radar parameters utilized in this study are presented in Table 2.

2.4. Relay guidance model

As depicted in Fig. 3, the red aircraft represent our side's lead-wingman formation, where the aircraft marked with the number 1 denotes the manned lead aircraft, and the aircraft marked with the number 2 denotes the unmanned loyal wingman. The red dotted line between the aircraft illustrates the formation link, which is utilized for the transmission of data and information. The symbolic variables labeled in the figure will be elucidated in Section 3.2.3.

Table 1 Distance and angle ratios for different zones.

Zone	Distance ratio	Angle ratio
Safe	(0.5, 1]	[−1, 1]
Danger	(0.2, 0.5]	[−1, 1]
Termination(inside)	[0, 0.2]	
Termination(outside)	> 1.5	
Loss: Distance	(1, 1.5]	[−1, 1]
Loss: Angle	(0.2, 1]	(1, 1.5]∪[−1.5, −1)
Loss: Distance + angle	(1, 1.5]	(1, 1.5]∪[−1.5, −1)

Table 2 Radar parameters.

Type	Detection angle (°)	Distance (km)
Loyal wingman radar	120	30
Lead aircraft radar		

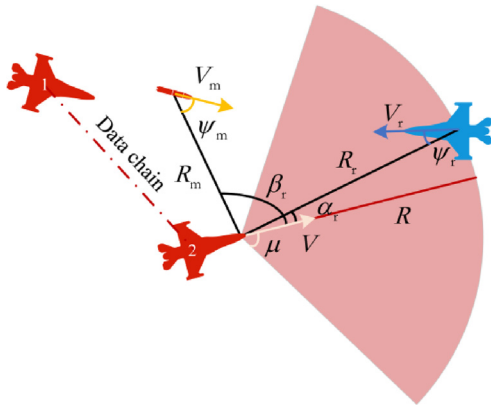


Fig. 3 Relay guidance model.

In Fig. 3, R_m is the relative distance between the guided missile and the wingman; ψ_m denotes the angle between the distance vector between the wingman and the missile and the missile speed vector on the horizontal plane; α_r denotes the angle between the distance vector between the wingman and the target and the wingman speed vector in the horizontal plane; β_r denotes the angle between the distance vector between the wingman and the missile and the wingman speed vector in the horizontal plane; μ represents a half of the radar detection angle. R represents the radar detection distance; R_r is the relative distance between the target enemy aircraft and the wingman; ψ_r is the angle between the velocity vector of the target enemy aircraft and the LOS of the wingman; V_r denotes the velocity of the target enemy aircraft.

3. Model construction method

In this study, we have designed a hierarchical reinforcement learning approach, integrating a low-level control model with a high-level control model to accomplish target search, target lock-on, and relay guidance maneuver decisions for a loyal wingman. The low-level control model is capable of manipulating basic control elements such as the throttle, ailerons, elevators, and rudders to achieve desired headings, velocities, and altitudes. When constructing high-level tactics, there is no need to consider the intricacies of low-level control; the high-level model simply needs to produce desired headings, velocities, and altitudes based on the input observation features. The low-level model takes the outputs of the high-level model as partial input observations to complete the low-level control tasks. Fig. 4. demonstrates the layered model architecture that we have developed for this paper. Different colored dashed lines represent the execution of different tasks, while solid lines indicate the common interaction process. Together, they form the execution cycle of different tasks. The controller operates at the same frequency, which is 5 Hz. The update frequency of the environment is 60 Hz.

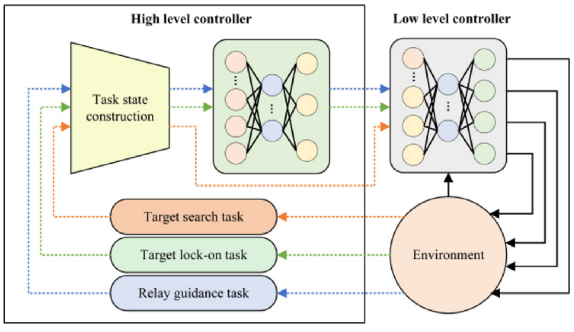


Fig. 4 Hierarchical control architecture.

3.1. Low-level control model

3.1.1. State space and action space

The construction of state features is crucial for training. Well-designed features can accelerate convergence and reduce the dimensionality of parameters. Considering the complexity of the environment and the combat mission, we have identified and designed a tuple consisting of 13 state variables to characterize the state space of the loyal wingman's low-level control model:

$$[\Delta H, \Delta\psi, \Delta V, H_{abs}, \cos\phi, \sin\phi, \sin\theta, \cos\theta, V_x, V_y, V_z, V] \quad (8)$$

where ΔH denotes the difference between the desired and current altitude of the wingman; $\Delta\psi$ is the difference between the desired and current heading; ΔV is the difference between the desired and current velocity; H_{abs} is the current altitude of the wingman; ϕ represents the roll angle; θ is the pitch angle; and V is the resultant velocity of the wingman. The units of ΔH and H_{abs} are km; $\Delta\psi$, ϕ, and θ are rad. ΔV, V_x, V_y, V_z, and V are described by Ma.

The action space of the low-level control model is constituted by a tuple of four continuous control variables:

$$[C_a, C_e, C_r, C_t] \quad (9)$$

the ranges for each control variable are as shown in Table 3.

To balance computational complexity with fidelity of simulation, this study also discretizes the continuous action space into a multi-discrete action space with a granularity of 50 for each continuous action.

3.1.2. Reward function and termination conditions

We expect the UAV to fly according to the desired heading, altitude, and velocity while limiting the roll angle to prevent stalling and loss of altitude due to excessive roll. Therefore, we have defined the following four reward functions:

$$\begin{cases} R_\psi = \exp\left(-\frac{\Delta\psi^2}{25}\right) \\ R_H = \exp\left(-100\Delta H^2\right) \\ R_V = \exp\left(-\frac{\Delta V^2}{400}\right) \\ R_\phi = \exp\left(-\frac{\phi^2}{0.09}\right) \end{cases} \quad (10)$$

Table 3 Ranges for control variables.

Parameter	Range
C_a	[0, 1]
C_e	[−1, 1]
C_r	[−1, 1]
C_t	[−1, 1]

Please cite this article in press as: ZHANG J et al. Loyal wingman task execution for future aerial combat: A hierarchical prior-based reinforcement learning approach, *Chin J Aeronaut* (2024), https://doi.org/10.1016/j.cja.2024.03.009

All four reward functions are of a Gaussian form, limiting the reward value to the interval (0,1]. Different standard deviations of the Gaussian reward functions represent different tolerances for different indices. The smaller the standard deviation, the lower the tolerance and the higher the precision required. To quickly meet the requirements for heading, altitude, velocity, and roll angle during training, we use the geometric mean of the four rewards as an overall measure of these four indicators:

$$R_{\mathrm{c}} = \sqrt[4]{R_{\psi} \cdot R_{H} \cdot R_{V} \cdot R_{\phi}} \tag{11}$$

Additionally, if the UAV descends to a dangerous altitude of 1 km, a penalty is given:

$$R_{\mathrm{ph}} = -2 \tag{12}$$

Finally, we compute the sum of the above rewards as the total reward for the wingman's low-level control model:

$$R_{\mathrm{s}} = R_{\mathrm{c}} + R_{\mathrm{ph}} \tag{13}$$

where R_{ψ} denotes the reward function for heading; R_H for altitude; R_V for velocity; R_{ϕ} for roll; R_{c} is the overall control reward function; R_{ph} is the altitude penalty constant; R_{s} is the overall reward for the low-level control model. The ranges for the above rewards is shown in Table 4.

In accordance with the characteristics of the mission, we define the following seven termination conditions for the loyal wingman's low-level control model:

(1) Failure to reach the expected heading, velocity, and altitude within a specified number of simulation steps, 500 steps.
(2) The loyal wingman descends to a dangerous altitude of 1 km.
(3) Operation exceeds the time range: beyond 2×10^3 simulation steps.
(4) The loyal wingman's altitude exceeds 1×10^5 km.
(5) The loyal wingman's rotational angular velocities p, q, r exceed 1×10^3 rad/s.
(6) The loyal wingman's velocity exceeds 100 Ma.
(7) The loyal wingman's acceleration exceeds 20 g.

3.2. High-level control model

3.2.1. Target search model

The target search scheme presented in this paper utilizes a waypoint navigation approach, where several waypoints are predefined. The loyal wingman aircraft navigate through these waypoints to scan the battlefield, while also providing cover and clearing the path for the lead aircraft. A schematic representation of the waypoints assigned to the lead and wingman aircraft is shown in Fig. 5.

Table 4 Ranges for low-level control model rewards.

Reward	Range
R_{ψ}	(0, 1]
R_V	(0, 1]
R_{c}	(0, 1]
R_{s}	(−2, 1]
R_H	(0, 1]
R_{ϕ}	(0, 1]
R_{ph}	−2

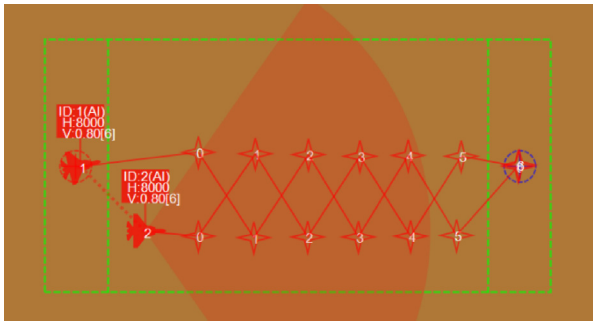


Fig. 5 Waypoints search.

This model only requires processing of the high-level waypoint coordinates, which are transformed into desired headings, velocities, and altitudes for the loyal wingman aircraft. The differences between these desired parameters and the current parameters of the aircraft are calculated and used as partial observations for the low-level control model, facilitating tactical control.

Let us define that the loyal wingman aircraft is currently heading towards the waypoint O_i, with geographical coordinates (λ_i, ϕ_i, h_i). The aircraft's current geographical position is (λ, ϕ, h), with a horizontal velocity vector $\boldsymbol{V}_{xy} = [V_x, V_y]$. The origin of the operational area's geographical coordinates is $(\lambda_{\mathrm{o}}, \phi_{\mathrm{o}}, h_{\mathrm{o}})$. By converting the geographical coordinates into a navigational coordinate system, we obtain the waypoint coordinates as (x_i, y_i, z_i), and the aircraft's coordinates as (x, y, z). Therefore, the vertical difference in altitude can be calculated as

$$\Delta H = z_i - z \tag{14}$$

On the horizontal plane, the position vector of the target waypoint relative to the loyal wingman aircraft is given by:

$$\boldsymbol{L} = [x_i - x, y_i - y] \tag{15}$$

This allows us to determine the magnitude of the heading angle difference as

$$\Delta\psi = \arccos\left(\frac{\boldsymbol{L} \cdot \boldsymbol{V}_{xy}}{\|\boldsymbol{L}\| \cdot \|\boldsymbol{V}_{xy}\|}\right) \tag{16}$$

However, the sign of the angle difference, indicating whether the aircraft should turn left or right, remains undetermined. We utilize Eq. (17) to determine the sign of the angle difference:

$$\mathrm{Sign}(\boldsymbol{L} \times \boldsymbol{V}_{xy}) \tag{17}$$

where $\mathrm{Sign}(x)$ is the sign function, defined as:

$$\mathrm{Sign}(x) = \begin{cases} -1 & , x < 0 \\ 0 & , x = 0 \\ 1 & , x > 0 \end{cases} \tag{18}$$

The velocity difference ΔV can be chosen as per the mission's requirements, either to accelerate (positive difference), decelerate (negative difference), or maintain current speed (zero difference).

Please cite this article in press as: ZHANG J et al. Loyal wingman task execution for future aerial combat: A hierarchical prior-based reinforcement learning approach, *Chin J Aeronaut* (2024), https://doi.org/10.1016/j.cja.2024.03.009

With the high-level processing complete and the differences in heading, speed, and altitude obtained, these values can now serve as inputs to the low-level control model to execute the waypoint-based target search task.

3.2.2. Target lock-on model

During the target search process by the loyal wingman, if a target is detected, the system switches to the target lock-on model. The target lock-on model is based on the zone model defined in Section 2.3. What distinguishes this model is the creation of a 'loss zone' outside the radar detection area. This loss zone represents that the locked target has left the detection area. When a target is within the loss zone, it signifies the target is lost. We also refer to the loss zone as the prior area because, during training, we assume that the state information of the targets within the loss zone is known, whereas, during the deployment phase, since this area is outside the detection zone, the target information within it is unknown. Since we have introduced priors, namely the distance ratio and angle ratio of the target are known, they can guide our aircraft to quickly reacquire the target during the training phase. In the assessment phase, since the target is outside the detection area, we cannot know its distance ratio and angle ratio, but we can infer the type of loss based on the target's last known distance ratio $F_{\mathrm{D}_{t-1}}$ and angle ratio $F_{\mathrm{A}_{t-1}}$.

Type 1. Distance loss

If $|F_{\mathrm{D}_{t-1}}| \geqslant 0.95$ and $|F_{\mathrm{A}_{t-1}}| < 0.8$, the loss type at time t is distance loss. The observed value of the distance ratio is set to the worst-case value of 1.5, and the observed value of the angle ratio is set to $F_{\mathrm{A}_{t-1}}$.

Type 2. Angle loss

If $|F_{\mathrm{D}_{t-1}}| < 0.95$ and $|F_{\mathrm{A}_{t-1}}| \geqslant 0.8$, the loss type at time t is angle loss. If $F_{\mathrm{A}_{t-1}} < 0$, the observed value of the angle ratio is set to the worst-case value of -1.5; if $F_{\mathrm{A}_{t-1}} > 0$, it is set to 1.5, and the observed value of the distance ratio is set to $F_{\mathrm{D}_{t-1}}$.

Type 3. Distance + angle loss

If $|F_{\mathrm{D}_{t-1}}| \geqslant 0.95$ and $|F_{\mathrm{A}_{t-1}}| \geqslant 0.8$, the loss type at time t is distance & angle loss. If $F_{\mathrm{A}_{t-1}} < 0$, the observed value of the angle ratio is set to -1.5, and if $F_{\mathrm{A}_{t-1}} > 0$, it is set to 1.5; the observed value of the distance ratio is set to the worst-case value of 1.5.

By setting the observed values of the distance ratio and angle ratio in this way, even without prior information, the model can guide the wingman to quickly and automatically reacquire the target during the deployment phase.

Additionally, by introducing the concept of the 'loss zone' outside the detection area as a buffer for target search and localization, the frequency of switching between different algorithm models is reduced, enhancing the robustness and stability of the entire combat system.

(1) State space and action space

We modify and supplement the state space of the low-level control model and eventually design a tuple of 16 state variables to represent the state space of the high-level target lock-on control model of the loyal wingman:

$$\begin{aligned}&[\Delta H\prime, \Delta\psi\prime, \Delta V_{\mathrm{r}}, H_{\mathrm{abs}}, \cos\phi, \sin\phi,\\ &\sin\theta, \cos\theta, V_x, V_y, V_z, V, R_{\mathrm{r}}, \psi_{\mathrm{r}}, F_{\mathrm{A}}, F_{\mathrm{D}}]\end{aligned} \tag{19}$$

where $\Delta H\prime$ denotes the difference in altitude between the target enemy aircraft and the current altitude of the wingman; $\Delta\psi\prime$ represents the difference in the direction of the horizontal plane speed of the target enemy aircraft from the current heading of the wingman; ΔV_{r} signifies the difference in the speed of the target enemy aircraft from the current speed of the wingman; $\Delta\psi\prime$, ψ_{r} are in radians; ΔV_x is in Ma; F_{A}, and F_{D} are defined in Eq. (27) and are dimensionless.

The action space of the high-level target lock-on control model consists of a triplet of three discrete error amounts, which control the wingman's heading, speed, and altitude across five discrete values:

$$[\Delta H_{\mathrm{d}}, \Delta\psi_{\mathrm{d}}, \Delta V_{\mathrm{d}}] \tag{20}$$

where ΔH_{d} is the altitude error amount; $\Delta\psi_{\mathrm{d}}$ is the heading error amount; and ΔV_{d} is the speed error amount. The discrete values considered for each action are shown in Table 5.

These actions will be part of the observations for the low-level control model.

(2) Reward function and termination conditions

Firstly, we consider the reward in the loss zone. The reward factors in the loss zone are based on distance loss, angle loss, and the combination of them. The presence of an enemy target within this zone implies that the target can be automatically retrieved through the model. We define the penalties for the three cases as:

$$\begin{cases} P_{\mathrm{DL}} = -2F_{\mathrm{D}} \\ P_{\mathrm{AL}} = -2F_{\mathrm{A}} \\ P_{\mathrm{DAL}} = \sqrt{P_{\mathrm{AL}} \cdot P_{\mathrm{DL}}} \end{cases} \tag{21}$$

Secondly, we consider the penalty in the safe zone. Since our goal is to keep the target within the safe zone for as long as possible through maneuvering, the penalty when the target is in the safe zone is:

$$P_{\mathrm{saf}} = 0 \tag{22}$$

Thirdly, for the penalty in the danger zone, we desire to keep the target as far from the danger zone as possible through our maneuvers. Therefore, we designed a Gaussian reward to encourage the wingman to maneuver away from danger zone:

$$P_{\mathrm{dan}} = 2\exp\left(-(F_{\mathrm{D}} - 0.5)^2\right) - 2 \tag{23}$$

Fourthly, we consider the penalty in the termination zone. When the target is within the termination zone (inside the fan), it signifies that the distance is too close and the wingman lacks countermeasures, leading to the termination of target lock.

Table 5 Ranges for discrete control variables.

Parameter	Range
ΔH_{d}(km)	{−0.2, −0.1, 0, 0.1, 0.2}
$\Delta\psi_{\mathrm{d}}$(rad)	{−π/6, −π/12, 0, π/12, π/6}
$\Delta V_{\mathrm{d}}/Ma$	{−0.2, −0.1, 0, 0.1, 0.2}

Please cite this article in press as: ZHANG J et al. Loyal wingman task execution for future aerial combat: A hierarchical prior-based reinforcement learning approach, *Chin J Aeronaut* (2024), https://doi.org/10.1016/j.cja.2024.03.009

When the target is outside the termination zone (outside the fan), it means the distance is too far, and the target cannot be retrieved, thus ending the round:

$$P_{\text{ter}} = -5 \tag{24}$$

Finally, applying only the above penalties could lead the model to end rounds prematurely to avoid accumulated penalties. Therefore, we also consider a reward for detecting targets outside the termination zone. We provide a reward when the target is located in the safe and danger zones because it remains within the detection area:

$$R_{\text{det}} = 1 \tag{25}$$

where P_{DL} represents the guidance penalty for distance loss, aimed at reducing the penalty by shortening the distance between our aircraft and the target; P_{AL} represents the guidance penalty for angle loss, aimed at reducing the angle difference between our aircraft's radar detection zone and the target; P_{DAL} is the geometric mean of P_{DL} and P_{AL}, aimed at simultaneously reducing the distance and angle discrepancies; P_{saf} represents the penalty within the safe zone; P_{dan} represents the penalty within the danger zone; P_{ter} represents the penalty within the termination zone; and $R_{\text{det}} = 1$ indicates the reward within the detection zone. The ranges for these penalties and rewards is shown in Table 6.

In addition to the termination conditions of the low-level control model, based on the specific nature of this task, the round should also end when the enemy target is within the termination zone.

3.2.3. Relay guidance model

(1) State space and action space

Building on the state space of the high-level target lock-on control model, we have supplemented it with the missile's relative observational information. Ultimately, we designed a tuple composed of 21 state variables to characterize the state space of the high-level relay guidance control model, as detailed in Section 2.4:

$$\begin{aligned}&[\Delta H\prime, \Delta\psi\prime, \Delta V_{\text{r}}, \Delta V_{\text{m}}, H_{\text{abs}}, \cos\phi, \sin\phi, \sin\theta, \cos\theta,\\ &V_x, V_y, V_z, V, R_{\text{r}}, R_{\text{m}}, \psi_{\text{r}}, F_{\text{A}}, F_{\text{D}}, \psi_{\text{m}}, F_{\text{Am}}, F_{\text{Dm}}]\end{aligned} \tag{26}$$

$$\begin{cases} F_{\text{A}} = \frac{\alpha_{\text{r}}}{\mu} \\ F_{\text{Am}} = \frac{\beta_{\text{r}}}{\mu} \\ F_{\text{D}} = \frac{R_{\text{r}}}{R} \\ F_{\text{Dm}} = \frac{R_{\text{m}}}{R} \end{cases} \tag{27}$$

Table 6 Ranges for penalties and rewards.

Penalty/reward	Range
P_{dl}	[−3, 2)
P_{al}	[−3, 2)
P_{dal}	[−3, 2)
P_{saf}	0
P_{dan}	(−2, 0]
P_{ter}	−5
R_{det}	1

where ΔV_{m} refers to the discrepancy between the guided missile and the wingman's current velocity; F_{Am} represents the angular ratio of the guided missile relative to the wingman; F_{Dm} indicates the distance ratio of the guided missile relative to the wingman. The unit of ψ_{m} is rad; ΔV_{m} is defined in Ma; F_{Am} and F_{Dm} are dimensionless quantities.

Similarly, the action space of the high-level target lock-on control model is composed of a triplet of discrete error metrics, with the ranges being the same as shown in Table 5. These actions will also serve as part of the observations for the low-level controller.

(2) Reward function and termination conditions

Within the relay guidance model, the wingman is solely tasked with the rapid simultaneous illumination of both the target hostile aircraft and the missile slated for guidance, without the need to consider the lead aircraft's positioning maneuvers or the timing of missile launch. The high-level relay guidance maneuver is initiated when the lead aircraft completes its positioning maneuver and launches the missile, and when the wingman locks onto the target, provided that the relative situational parameters between the missile and the wingman satisfy the conditions:

$$\begin{cases} |F_{\text{A}} - F_{\text{Am}}| \leqslant 2 \\ F_{\text{Dm}} \leqslant 1 \end{cases} \tag{28}$$

Therefore, this study establishes a reward when both the target and the missile are simultaneously illuminated, denoted by:

$$R_{\text{g}} = 0.25 \tag{29}$$

Conversely, a penalty is imposed when a missile lock-on failure or target loss occurs, expressed as:

$$P_{\text{g}} = -1 \tag{30}$$

In addition to the termination conditions of the low-level control model, this task-specific framework dictates that the episode should also conclude when the guided missile is destroyed, either through energy depletion or upon hitting the target.

4. Loyal wingman algorithms for beyond visual range air combat maneuvers

4.1. PPO-based control algorithm for loyal wingman

Proximal Policy Optimization (PPO), as a reinforcement learning algorithm with an Actor-Critic structure, has been widely applied across various domains. To comprehend the PPO algorithm profoundly, this study initiates from the concept of "policy gradient." Policy gradient methods directly utilize neural networks to approximate the policy function. The core formula for updating the parameters of the policy network is as Eq. (31):

$$\nabla\bar{R}_{\Theta} = \mathbb{E}_{\tau\sim p_{\Theta}(\tau)}[R(\tau)\nabla \lg p_{\Theta}(\tau)] \tag{31}$$

where τ represents the trajectory samples of the agent within an episode, following the probability distribution $p_{\Theta}(\tau)$ as defined by policy parameters Θ. $R(\tau)$ denotes the cumulative discounted reward for the trajectory. Our objective is to com-

pute the gradient of the expected value of the cumulative discounted reward with respect to the policy parameters. The expected value of the cumulative discounted rewards is augmented by performing gradient ascent on the policy parameters.

In conventional Policy Gradient (PG) algorithms, each trajectory sample is utilized only once for updating the policy network parameters. In contrast, the PPO algorithm integrates the strengths of both Advantage Actor-Critic (A2C) and Trust Region Policy Optimization (TRPO), permitting multiple iterative updates of policy network parameters from trajectory samples under the old policy. This feature is facilitated by the introduction of importance sampling to ensure unbiasedness, and a clipping mechanism is incorporated to restrain the divergence between two policy functions, particularly in terms of the probability ratio.

Importance sampling is employed as demonstrated in the Eq. (32):

$$\nabla \bar{R}_{\Theta} = \mathbb{E}_{\tau \sim p_{\Theta'}(\tau)} \left[\frac{p_{\Theta}(\tau)}{p_{\Theta'}(\tau)} R(\tau) \nabla \lg p_{\Theta}(\tau) \right] \tag{32}$$

where $\Theta\prime$ signifies policy network parameters that are in proximity to Θ. The plan is to sample under the policy with parameters $\Theta\prime$ and employ these samples to update Θ.

Thus far, we have discussed entire trajectories. Now, let us decompose the trajectory into state-action pairs (s, a). Furthermore, to avert blindly increasing the selection probability of a corresponding action merely because the return is positive—which could potentially decrease the selection probability of more optimal actions due to lower sampling frequency—we introduce a baseline, specifically the negative value of the state value function $V_{\pi}(s)$, to adjust the selection probabilities of different actions during updates. This return function with a baseline is known as the advantage function, defined as Eq. (33):

$$A_{\pi}(s, a) = Q_{\pi}(s_t, a_t) - V_{\pi}(s_t) \tag{33}$$

Intuitively, this formula indicates the difference between the expected future returns after taking the current action in a specific state and the expected value prior to taking the action, serving as a metric to assess the quality of actions. Typically, the Generalized Advantage Estimation (GAE) is utilized to calculate advantage values since it balances between Temporal Difference (TD) methods and Monte Carlo (MC) approaches, while concurrently moderating bias and variance:

$$A_{\pi}(s, a) = \hat{A}_{\pi}(s_t, a_t) = \delta_t + (\gamma\lambda)\delta_{t+1} + \ldots + (\gamma\lambda)^{T-t+1}\delta_{T-1} \tag{34}$$

where $\delta_t = r_t + \gamma V(s_{t+1}) - V(s_t)$, T indicates the termination moment of the episode, and γ and λ are hyperparameters used to diminish bias and variance.

Hence, Eq. (32) is transformed into Eq. (35):

$$\mathbb{E}_{(s_t, a_t) \sim \pi_{\Theta'}} \left[\frac{p_{\Theta}(s_t|a_t)}{p_{\Theta'}(s_t|a_t)} A^{\Theta'}(s_t, a_t) \nabla \log p_{\Theta}(a_t|s_t) \right] \tag{35}$$

In practice, what is being optimized is represented by Eq. (36):

$$J^{\Theta'}(\Theta) = \mathbb{E}_{(s_t, a_t) \sim \pi_{\Theta'}} \left[\frac{p_{\Theta}(s_t|a_t)}{p_{\Theta'}(s_t|a_t)} A^{\Theta'}(s_t, a_t) \right] \tag{36}$$

Finally, we utilize the clipping operation to ensure that the difference between the policies corresponding to parameters Θ and $\Theta\prime$ does not become too large, as shown in the Eq. (37):

$$J^{\Theta'}_{\text{clip}}(\Theta) \approx \sum_{(s_t, a_t)} \min \left(\frac{p_{\Theta}(s_t|a_t)}{p_{\Theta'}(s_t|a_t)} A^{\Theta'}(s_t, a_t), \right. \\ \left. \text{clip}\left(\frac{p_{\Theta}(s_t|a_t)}{p_{\Theta'}(s_t|a_t)}, 1-\varepsilon, 1+\varepsilon \right) A^{\Theta'}(s_t, a_t) \right) \tag{37}$$

where ε is a hyperparameter that limits the divergence between the target and behavior policies.

To enhance training efficiency, we implement PPO-clip algorithm from scratch. We designed a parallelized training algorithm that encapsulates $\mathcal{N}$ individual environments into a vectorized environment to expedite the training process. We initialized the network parameters using orthogonal initialization and assigned a smaller gain to the actor network to ensure that all actions have the opportunity to be selected. Additionally, we set the size of the experience buffer $\mathcal{D}$ equal to the product of the number of environments $\mathcal{N}$ and the batch size $\mathcal{B}$, and defined the total number of execution steps $\mathcal{T}$ divided by the batch size $\mathcal{B}$ as the total number of rollouts, during which experiences are collected and stored in the experience buffer. We also implemented a learning rate α annealing technique to improve algorithm performance. During experience collection, each environment is required to randomly change its desired altitude h, course ψ, and speed v at specific intervals to fully explore the state space. Following experience collection, the Generalized Advantage Estimation is utilized to calculate the advantage estimate $\hat{A}$ and the return G, which are then stored in the experience buffer. In the training phase, to make full use of the experience data, we conduct $\mathcal{E}$ update epochs, where experience buffer samples are shuffled and partitioned into minibatches of size b for batch training and network parameter updates. The training process also involves generic operations such as ratio calculation, various loss computations, and gradient clipping, which are not elaborated here. The Parallelized Training Algorithm for Low-Level Control is shown as Algorithm 1.

Algorithm 1. Parallelized training algorithm for low-level control.

```
Initialize parameters of actor Θ_a and critic Θ_c orthogonally;
Initialize the experience pool D;
Reset the vector environment;
for n = 1 to T/B do
    α ← 1.0 − (n − 1.0) × α × B/T;
    for t = 1 to B/N do
        Change desired h, ψ, v randomly at fixed time intervals;
        Get s_t, d_t;
        Get a_t, log π_{Θ_a}(a_t|s_t), v_{Θ_c}(s_t) from actor and critic network;
        Execute one step in the vector environment;
        Store {s_t, a_t, log π_{Θ_a}(a_t|s_t), v_{Θ_c}(s_t), r_{t+1}, d_t} into D;
    Calculate advantages Â and returns G and store them into D;
    for e = 1 to E do
        Shuffle D;
        for i = 1 to B/b do
            Calculate policy loss, value loss and entropy loss;
            Update Θ_a, Θ_c;
```

4.2. Hierarchical PPO-based loyal wingman high-level relay guidance control algorithm

The algorithm is founded on a hierarchical concept, where the low-level model network parameters are frozen, and only the high-level model network parameters are updated. During the interaction with the environment, signals are relayed from the high level to the low level, thereby facilitating the state transition of the environment. Throughout the interaction process, the high-level state observation focuses on more abstract, task-level aspects. For the actions controlled by the high-level, we pay closer attention to more semantic and comprehensible quantities, such as desired heading, speed, and altitude. Furthermore, the control actions of the high-level can be utilized as observation signals for the low level, thus guiding the more complex low-level flight control logic. Hierarchical PPO-based Relay Guidance for Loyal Wingman Control algorithm is shown as Algorithm 2.

Algorithm 2. Hierarchical PPO-based relay guidance for loyal wingman control.

 # Initialize parameters and reset envs
Load low-level model $\mathcal{M}_{low}$ and set it to eval mode;
for $n = 1$ **to** $\mathcal{T}/\mathcal{B}$ **do**
  # Anneal learning rate
  for $t = 1$ **to** $\mathcal{B}/\mathcal{N}$ **do**
    # Transfer actions into low-level states
    $\mathcal{M}_{low}$ takes one step to update the vector env;
    # Store high-level experience
  Store $\hat{A}$ and G into $\mathcal{D}$;
  for $e = 1$ **to** $\mathcal{E}$ **do**
    Shuffle $\mathcal{D}$;
    for $i = 1$ **to** $\mathcal{B}/b$ **do**
      # Update parameters of high-level networks

4.3. Loyal wingman high-level target locking control algorithm based on HPE-PPO

The algorithm is divided into two phases: training and deployment. During training, the enemy aircraft's initial position is randomly located within the loss zone, yet its information is perceivable, and its heading and speed are randomly initialized. Throughout the training process, the enemy aircraft's heading and speed are altered at fixed time intervals to allow the algorithmic model to fully explore the state space. In the deployment phase, the information about the enemy aircraft within the loss zone is no longer known, and the algorithmic model will apply control rules matched to the type of loss. If the target enemy aircraft is successfully reacquired within the time interval N, it will continue to be tracked. Otherwise, the model will switch to the target search mode. The HPE-PPO based Target Locking for Loyal Wingman Control algorithm is shown as Algorithm 3.

Algorithm 3. HPE-PPO based target lock-on for loyal wingman control.

Load low-level model $\mathcal{M}_{low}$ and set it to eval mode;
Training phase:
 # Initialize parameters and reset envs
 # Initialize enemy states randomly in loss zone
for $n = 1$ **to** $\mathcal{T}/\mathcal{B}$ **do**
  # Anneal learning rate
  for $t = 1$ **to** $\mathcal{B}/\mathcal{N}$ **do**
    Change ψ and v of enemy randomly at fixed time intervals;
    # Transfer actions into low-level states
    $\mathcal{M}_{low}$ takes one step to update the vector env;
    # Store high-level experience
  Store $\hat{A}$ and G into $\mathcal{D}$;
  for $e = 1$ **to** $\mathcal{E}$ **do**
    Shuffle $\mathcal{D}$;
    for $i = 1$ **to** $\mathcal{B}/b$ **do**
      # Update parameters of high-level networks
Deployment phase:
if *Enemy is loss* **then**
  Return $F_{D_{t-1}}$ and $F_{A_{t-1}}$ before loss;
  if $|F_{D_{t-1}}| \geq 0.95$ ***and*** $|F_{A_{t-1}}| < 0.8$ **then**
    Dist. Loss, **Set** $F_D = 1.5$, $F_A = F_{A_{t-1}}$;
  else if $|F_{D_{t-1}}| < 0.95$ ***and*** $|F_{A_{t-1}}| \geq 0.8$ **then**
    Angle Loss, **Set** $F_A = -1.5$ **if** $F_{A_{t-1}} < 0$ **else** 1.5; **Set** $F_D = F_{D_{t-1}}$
  else
    Dist. + Angle Loss, **Set** $F_A = -1.5$ **if** $F_{A_{t-1}} < 0$ **else** 1.5; **Set** $F_D = 1.5$
  for $n = 1$ **to** N **do**
    if *Relock enemy* **then**
      break;
    else
      Switch to target search model;

5. Experiments and results

5.1. Experiment platform and parameters

In our experiments, we employ the following hardware and software platforms shown in Tables 7 and 8.

The parameter settings for the algorithmic model are shown in the Table 9.

Table 7 Hardware platform.

Hardware	Specification
CPU	Intel i7-10700KF
GPU	Nvidia RTX 3080
Memory	16 GB
GPU Memory	8 GB

Table 8 Software platform.

Work	Platform
Algorithm Implementation	Python run in PyCharm
Environment Development	C++ compiled by VS2019
Experiment Visualization	Tacview

Table 9 Hyperparameters configuration.

Parameter	Value	Parameter	Value
$\mathscr{E}$	4	seed	1024
$\mathscr{B}$	3000	clip_param	0.2
hidden_size	128×128	entropy_coef	1×10^{-3}
λ	0.95	γ	0.99
α	3×10^{-4}	max_grad_norm	2
$\mathscr{N}$	2	$\mathscr{T}$	1×10^{8}
b	1200	N	200
gain_actor	0.01	gain_critic	1

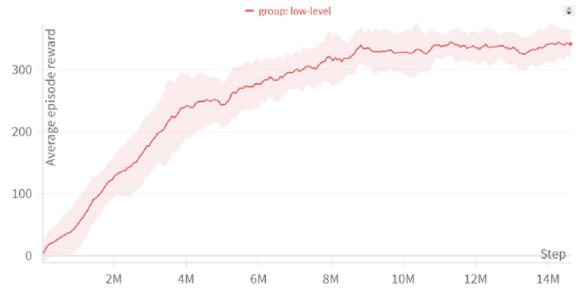


Fig. 6 Low-level control average episodic reward.

5.2. Training and simulation results

5.2.1. Experiment 1: Target search

The target search model relies on the precise control of the low-level model. To accomplish this task, we initially train the low-level control model using Algorithm 1. The average episodic rewards during the training process are depicted in Fig. 6. As can be observed from the figure, the training process was relatively slow, with the reward curve tending to stabilize and the model achieving satisfactory performance after nearly 8 million steps. The reward curve uses exponential smoothing with a smoothing factor of 0.6.

Furthermore, to analyze the control precision and response speed of the low-level control model, we set the initial heading of the loyal wingman aircraft to 60, initial speed to $0.7Ma$, and initial altitude to 6000 meters. The target heading was 0 degrees, target speed 1 Ma, and target altitude 8000 meters. We record the control curves for heading, altitude, and speed as the aircraft transitioned from its initial state to the target state. The entire experimental procedure is conducted over 700 simulation steps, with the results recorded as shown in Fig. 7.

In the graph, the vertical axis represents the absolute value of the difference between the aircraft state values and their initial values, relative to the initial values. This value serves as a relative measure to eliminate the effects of dimensional units and orders of magnitude of different state quantities. The horizontal axis represents the simulation time steps. The yellow, blue, and orange curves respectively represent the relative ratio of aircraft altitude, heading, and speed as they change over time. A light blue bar spanning the entire simulation process marks the error band ranging from −7% to 5%. It is evident that the relative ratios of the three state quantities decrease significantly, indicating the synchronicity of control. Among them, the heading control enters the error band around Step 160 and stabilizes near 0; altitude control meets the requirements near Step 210; and speed control lags slightly, satisfying the requirements around Step 360. The difference in control speeds can be reflected in the variance setting of the reward function, as the variance indicates the algorithm's tolerance

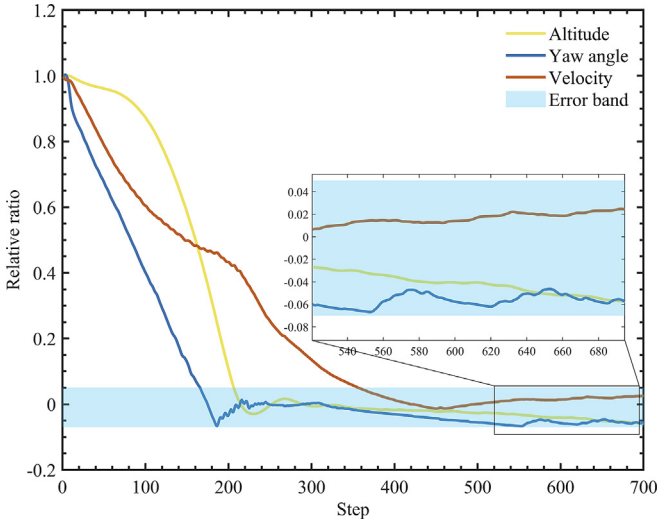


Fig. 7 Key state ratio curve over steps.

for control errors. Finally, all three state quantities stabilize within the error band, demonstrating the stability of the flight control.

Returning to the target search model, as depicted in Fig. 5, the operational area for this study is confined to a rectangle delineated by green frame lines, with the penetration point located within the opponent's circular region. Prior to mission execution, a series of waypoints are predefined, and the loyal wingman aircraft is tasked to fly ahead of the lead aircraft following these waypoints to complete the target search. Utilizing the method for the target search model presented in Section 3.2.1, the waypoint information and the aircraft's own state information are compiled into input signals for the low-level model, which then generate the control parameters for the loyal wingman. Experimental results show that the actual flight trajectory of the low-level model, following the waypoint information as shown in Fig. 5, is as illustrated in Fig. 8. It is apparent that the model is well-capable of accomplishing this task.

5.2.2. Experiment 2: Target lock-on

Building upon the foundational control model, we employ Algorithm 3 to train the target lock-on model. The average episodic reward curve of the model is presented in Fig. 9. It is evident that due to the presence of the prior knowledge domain and the low-level model, the algorithm converges rapidly, achieving a stable state after approximately 500 k training steps, with the model exhibiting good performance. The reward curve uses exponential smoothing with a smoothing factor of 0.84.

The significance of this model lies in its ability to lock onto the target enemy aircraft for as long as possible under different scenarios, thereby affording the lead aircraft ample time to complete an attack positioning. Moreover, to enhance the robustness of the model and the stability of the entire loyal wingman combat system, the model is designed with a target reacquisition feature that enables automatic target retrieval upon loss of lock. To simulate a variety of scenarios that may be encountered during combat, we design three typical adversarial experiments: enemy aircraft performing triangular maneuvers, enemy aircraft flying towards our aircraft (simulating a head-on attack), and enemy aircraft flying away from our aircraft (simulating an enemy escape).

The enemy aircraft triangular maneuver scenario simulates the behavior of the enemy aircraft frequently changing course to test the model's comprehensive response capability. The simulation results of the experiment are shown in Fig. 10(a) and (b). From the complete flight trajectories depicted in the figures, it is evident that the loyal wingman has learned to employ strategies such as using climbs to decelerate rapidly, diving to accelerate quickly to maintain a safe distance, and slowing the approach speed when laterally aligned with the enemy aircraft. During the initial phase of the simulation, when the target enemy aircraft is near the loss-of-lock region, the loyal wingman is capable of rapidly controlling the turn and accelerating to close the distance between F_A and F_D. When the target is within the safe zone, the loyal wingman slowly climbs to reduce speed, ensuring that the target remains in the safe zone for an extended period. Upon entering the danger zone, the loyal wingman can climb quickly to reduce speed sharply, ultimately maintaining a safe distance without losing sight of the target.

To analyze the entire simulation process in a more direct and quantitative manner, we examine the curves of the angle

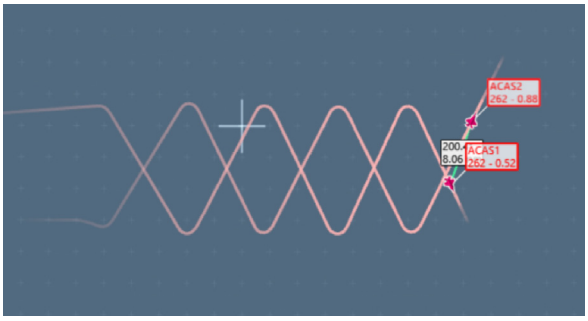


Fig. 8 Target search (in tacview).

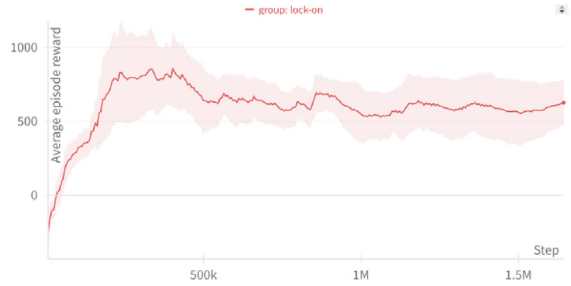
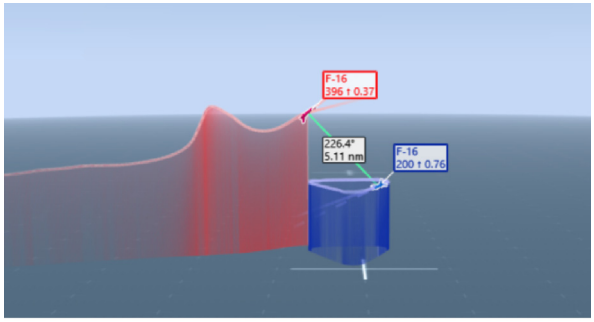
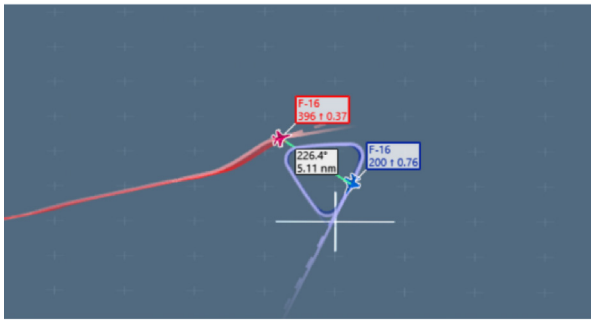


Fig. 9 Target lock-on average episodic reward.



(a) Triangular maneuvering target lock-on (side view).



(b) Triangular maneuvering target lock-on (top view).

Fig. 10 Head-on maneuvering target lock-on.

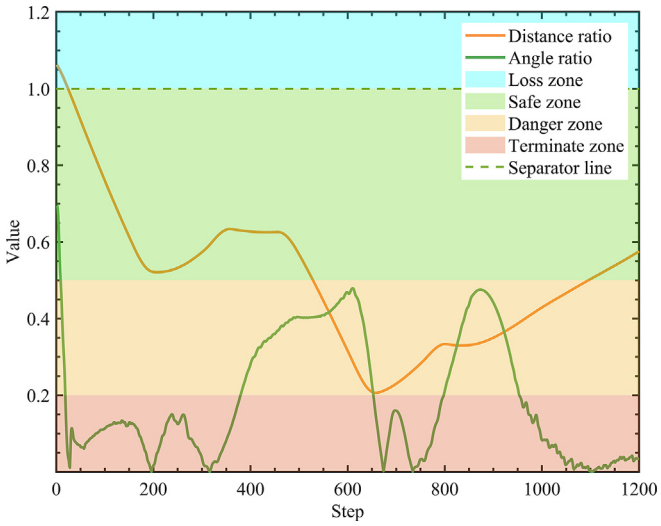


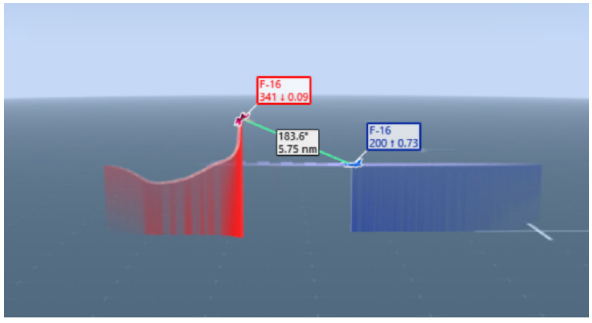
Fig. 11 Triangular maneuvering relative ratio over steps (just consider the distance ratio).

ratio F_A and the distance ratio F_D as they change with the simulation deployment. The loss zone, safe zone, and danger zone are clearly marked in the graph, as shown in Fig. 11.

From the analysis of the graph, the orange curve represents the distance ratio F_D, the green curve represents the absolute value of the angle ratio F_A, and the green dashed line indicates the dividing line for angle loss. If the value of F_A is below that of the dividing line, it indicates that there is no angle loss for the target enemy aircraft. The entire simulation experiment lasts for 1200 simulation steps, with the target enemy aircraft initially within the loss zone (the blue area in the graph), and then rapid maneuvers by the aircraft brought the target into the safe zone (the green area). The rate of change for both F_A and F_D are quite significant at this point, implying that rapid adjustments in course and speed are made to quickly reacquire the target enemy aircraft upon entering the loss zone. Between Steps 200–500, F_D can be seen to change more gradually, indicating that the model controls the loyal wingman to maintain an appropriate distance from the target, allowing the target to remain in the safe zone for longer periods. Subsequently, the target enters the danger zone (the orange area), where the rates of change for both F_A and F_D are again significant, indicating that the loyal wingman is maneuvering rapidly to prevent entry into the termination zone (the red area). Around Step 650, it is evident that the target is very close to the termination zone; finally, through maneuvering, the target is locked on while being moved away from the termination zone and ultimately returns to the safe zone.

The scenario where the enemy aircraft flies towards the loyal wingman simulates the behavior of a head-on attack by the enemy, testing the model's ability to respond to danger. The simulation results of the experiment are shown in Figs. 12 (a) and (b). From the complete flight trajectories depicted in the figures, it is clear that the loyal wingman is capable of executing a sharp climb to decelerate when the enemy aircraft approaches at high speed and has learned the ability to rapidly change direction in a pendulum-like maneuver.

We also analyze the changes in the distance ratio F_D and the absolute value of the angle ratio F_A curves with respect to the number of simulation steps under this scenario, which are shown in Fig. 13. From the F_A curve, it is apparent that at the initial moment, both parties discover each other and complete the turn to begin the approach. The reason for the relatively gentle changes in F_D at this time is due to the random initial state of the target, which is oriented away from the loyal wingman, necessitating a turning maneuver. Around Step 50, the target enters the safe zone, and the loyal wingman begins to decelerate, resulting in a decrease in the rate of change of the F_D curve. Since the experimental setup has the target



(a) Head-on maneuvering target lock-on (side view).

(b) Head-on maneuvering target lock-on (top view).

Fig. 12 Head-on maneuvering target lock-on.

Please cite this article in press as: ZHANG J et al. Loyal wingman task execution for future aerial combat: A hierarchical prior-based reinforcement learning approach, *Chin J Aeronaut* (2024), https://doi.org/10.1016/j.cja.2024.03.009

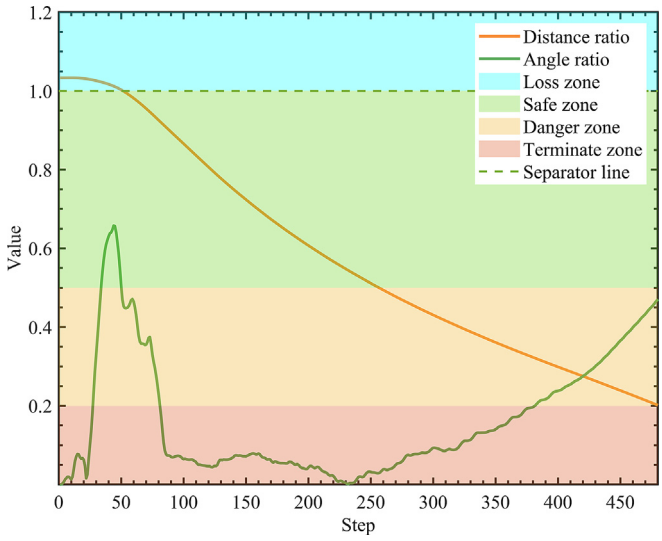
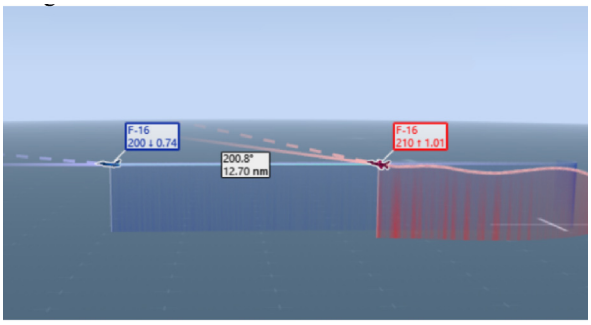


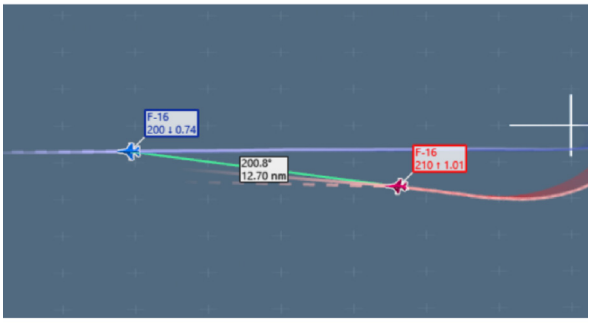
Fig. 13 Head-on maneuvering relative ratio over steps. (just consider the distance ratio).

enemy aircraft always maneuvering towards the loyal wingman, it is impossible for the loyal wingman to maintain a lock on the target while also constantly controlling the distance. Therefore, the entire simulation process does not reach the maximum execution step count of 1200, and the process lasts for 480 Steps.

The scenario where the enemy aircraft flies away from the loyal wingman simulates the behavior of the enemy escaping, testing the model's ability to continuously track and lock onto the target. The simulation results of the experiment are shown in Figs. 14(a) and (b). It can be seen from the graphs that the flight trajectory is relatively stable because the threat posed by the target enemy aircraft to the loyal wingman is minimal.



(a) Escape maneuvering target lock-on (side view).



(b) Escape maneuvering target lock-on (top view).

Fig. 14 Escape maneuvering target lock-on.

From Fig. 15, it can be observed that the loyal wingman maintains a very steady control of the distance to the target, effectively keeping the target within the safe zone throughout, with minimal fluctuations in the angle ratio F_A.

Finally, we present the number of steps during which the target enemy aircraft remains within the loss zone, the safe zone, and the danger zone for the three scenarios. It is evident that our algorithmic model is capable of controlling the loyal wingman with excellent performance, ensuring that the target remains in the safe zone for an extended duration. When faced with the threat of the approaching target enemy aircraft, the number of steps during which the target is in the danger zone increases, as shown in Fig. 16.

We evaluated the efficacy of model deployment, aiming to demonstrate the enhancement of system stability through the implementation of our model. To this end, we trained a target localization model without loss zones (prior zones). For the simulation of target evasion, hostile aircraft targets were randomly positioned at the boundaries of the detection area and programmed to fly away from this zone. If a target exited the detection area, the loyal wingman decision system would opt for the target search model. Conversely, if a target entered the detection area, the system would switch to the target lock-on model. Fig. 17 illustrates the comparative impact on system decision-making between the target lock-on model with a prior zone and one without. It is evident that the model incorporating a prior zone facilitates the system's transition from the target search to the target lock-on model upon target detection and maintains this strategy without switching for up to 1000 steps. In contrast, the absence of a prior zone in the target lock-on model led to frequent toggling between the target lock-on and search models, resulting in system instability.

Moreover, we test the effectiveness of the model upon deployment. In order to quantitatively analyze the stability performance improvement brought by the introduction of a

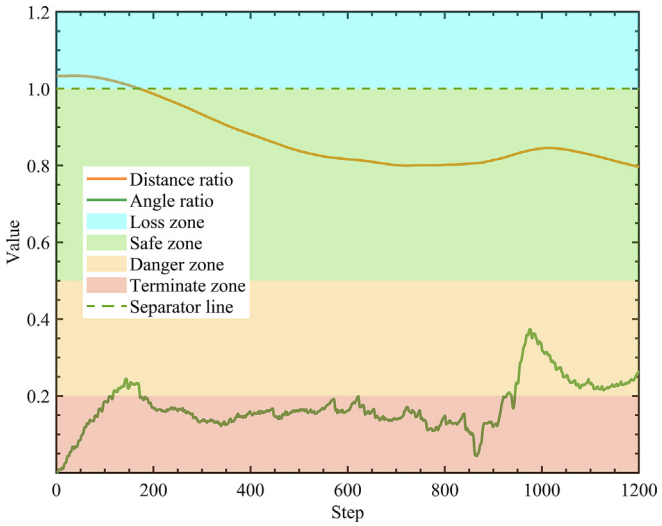


Fig. 15 Escape maneuvering relative ratio over steps. (just consider the distance ratio).

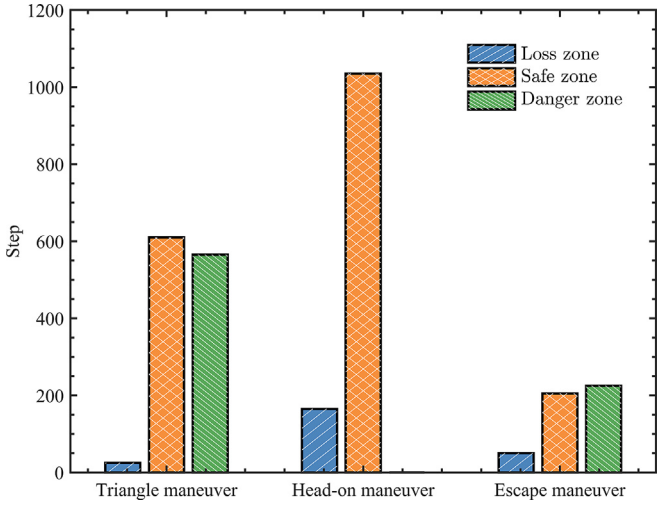


Fig. 16 Stride lengths of three target maneuvers in different zones.

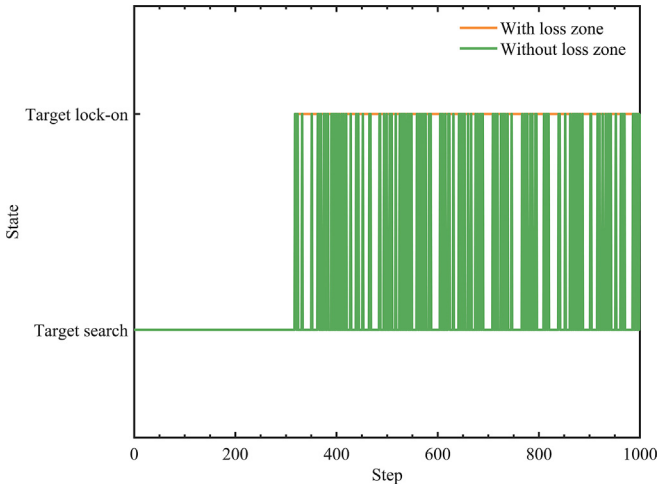


Fig. 17 Comparison of state transitions over steps between two models.

prior zone in the model, we place the enemy aircraft randomly at the boundary between the detection zone and the loss zone and had the enemy aircraft fly towards the loss zone. This setup was used to simulate three types of target losses. Each type of loss is experimented with 1000 times to statistically analyze how often the model could autonomously reacquire the target within 200 simulation steps without switching to other models, as well as the average number of steps required for successful re-acquisition. The statistical results are presented in Fig. 18.

It is evident that upon deployment, the model can reacquire the target with a very high probability in a short amount of time, which means that the entire control system does not need to frequently switch between different models to address situations of target loss and reacquisition. Notably, when the target is located at the boundary of a region, it is easy to meet the critical conditions for different models, which can lead to system instability due to jitter caused by frequent model switching. This model effectively establishes a buffer zone at the trigger boundaries of different models, resolving this instability.

5.2.3. Experiment 3: Relay guidance

The relay guidance model is also based on a hierarchical concept and was trained using Algorithm 2. The curve of the average episodic reward during the training process is shown in Fig. 19. It is apparent that this model converges stably at around 2 million steps, and the performance of the model is outstanding. The reward curve uses exponential smoothing with a smoothing factor of 0.84.

To further demonstrate the capability of the model for relay guidance, we conduct 100 experiments and calculate the proportion of the average effective and ineffective guidance time from the launch of the missile by the lead aircraft until the disappearance of the missile. Effective guidance is defined as the situation where the lead aircraft achieves the attack position, such that the relevant state variables of the loyal wingman satisfy the conditions expressed by Eq. (28). At this moment, the lead aircraft launches the missile, and the loyal wingman needs to maneuver within a limited 50 simulation steps to simultaneously illuminate the target enemy aircraft and the missile. The loyal wingman must continue to illuminate both the target and

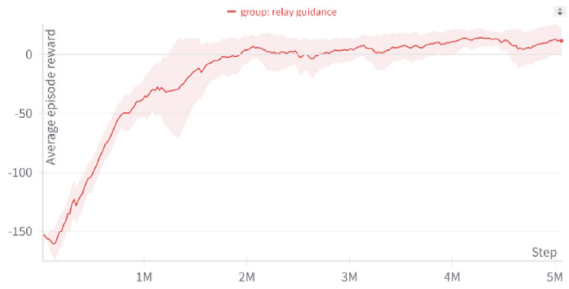


Fig. 19 Relay guidance average episodic reward.

the missile throughout the entire duration of the missile's lifecycle for it to be considered as one effective guidance. If the illumination is not achieved within the 50 simulation steps or if it fails to continuously illuminate the target or the missile during the remaining missile lifecycle, it is considered as an ineffective guidance. The experimental results indicate that the effective guidance time accounts for 82.3% of the total, while the ineffective guidance time makes up 17.7%.

During the simulation experiments, the lead aircraft is controlled by a human, satisfying the attack position and launching the missile as soon as the simulation begins. Throughout the simulation, if the missile's launch position is not within the radar detection range (guidance range) of the loyal wingman, the wingman will rapidly divert to bring both the target and the missile under radar illumination. Screenshots of the simulation process are shown in Fig. 20(a), (b), and (c). From the trajectory of the loyal wingman and the radar illumination situation depicted in Fig. 20(a), it is clear that the wingman (ID:2) quickly diverts its course to accomplish relay guidance, directing the missile (ID:103) launched by lead aircraft (ID:1) towards the target enemy aircraft. In the label information, H represents altitude, V represents speed, and L indicates whether the missile has lost the target or not. Fig. 20(b) shows the scenario where the lead aircraft (ACAS1) launches the missile (Msl13) and then disengages from the battlefield; it can be seen that the loyal wingman (ACAS2) flies towards the direction of the missile and the target enemy aircraft (ACAS3) to complete the relay guidance. Fig. 20(c) displays the situation where the lead aircraft launches multiple missiles, and the loyal

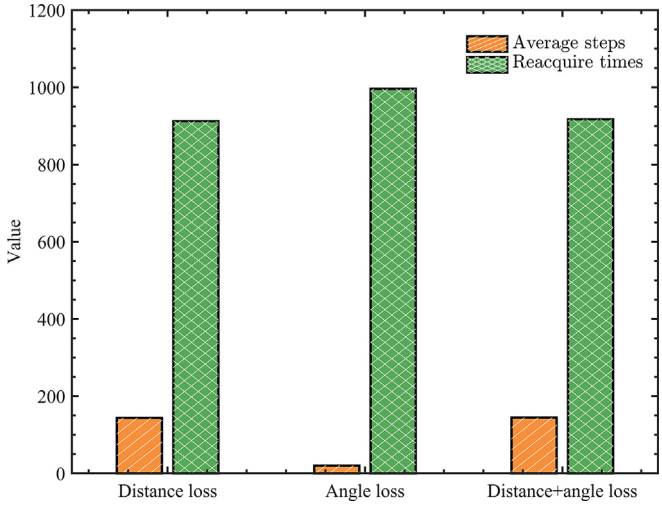
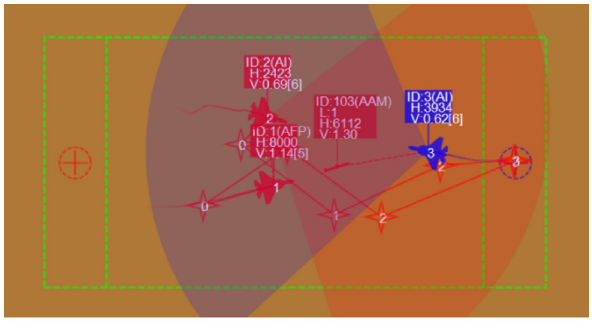
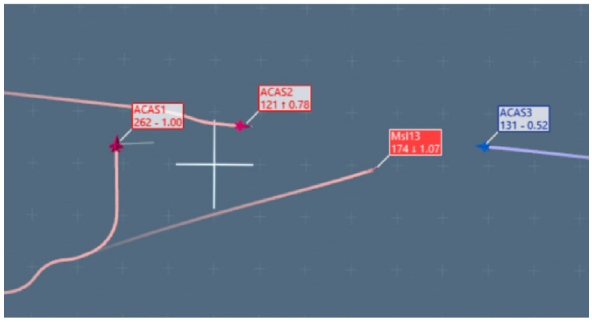


Fig. 18 Reacquire counts and average step length for three loss types.

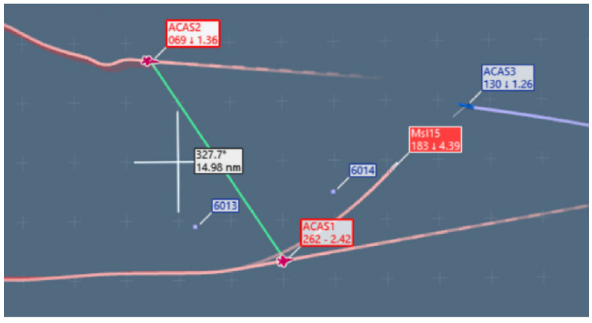
Please cite this article in press as: ZHANG J et al. Loyal wingman task execution for future aerial combat: A hierarchical prior-based reinforcement learning approach, *Chin J Aeronaut* (2024), https://doi.org/10.1016/j.cja.2024.03.009



(a) Relay guidance (in detail).



(b) Lead aircraft disengages from battlefield after missile launch.



(c) Lead aircraft launches multiple missiles.

Fig. 20 Relay guidance.

wingman conducts the relay guidance. The missiles labeled 6013 and 6014 become ineffective due to being launched from too great a distance, while the missile in flight (Msl15) is being diverted towards the target under the guidance of the loyal wingman.

6. Future outlook

Based on the research conducted in this paper, we explore an efficient and feasible model for BVR cooperative engagement between lead and wingman aircraft, specifically a highly intelligent unmanned loyal wingman operating in concert with a manned lead aircraft assisted by Artificial Intelligence (AI). The combat operation is based on the following process:

(1) The loyal wingman performs target searching along predefined waypoints.
(2) The lead aircraft follows the loyal wingman along predefined waypoints (as the risk factor is lower, this process can be controlled by AI).
(3) Upon target acquisition, the loyal wingman executes target lock-on.
(4) While the wingman locks onto the target, the lead aircraft completes an attack positioning maneuver.
(5) If the target is lost, the loyal wingman attempts to automatically reacquire the target within a specific number of time steps.
(6) The lead aircraft awaits target reacquisition.
(7) If the loyal wingman reacquires the target, it continues to lock on; if not, it switches to the target search model.
(8) The lead aircraft completes positioning and launches the missile.
(9) The loyal wingman executes relay guidance, and if the strike is not successful, it continues to lock on.
(10) The lead aircraft repositions and launches another missile.
(11) The loyal wingman guides the missile to hit all targets, concluding the strike mission.
(12) Both the lead aircraft and the loyal wingman continue to fly to the breakthrough point according to the waypoints, concluding the breakthrough mission.

This cooperative combat model has several advantages: independent control of each aircraft, controllable operational authority of the lead aircraft, high level of intelligence of the wingman, high degree of collaborative intelligence with strong interpretability, all of which are of practical combat significance.

7. Conclusions

This research provides a detailed modeling approach and an efficient training algorithm for intelligent unmanned loyal wingman systems. First, to accomplish efficient flight control, we design a 6-DOF control method for the loyal wingman based on the PPO algorithm and complete the waypoint target search task on this basis; next, we propose a hierarchical method that employs the flight control model as the bottom layer, training the top layer to complete relay guidance tasks; furthermore, based on the hierarchical approach, we introduce a "prioritized training, deprioritized execution" method to accomplish the top layer's target locking tasks. Experiments have proven that the proposed methods can efficiently execute various complex combat tasks of the loyal wingman, and the algorithms converge rapidly. Lastly, we explore an efficient and feasible model for cooperative combat formations between lead aircraft and loyal wingmen, which holds promise for application in future beyond-visual-range aerial combat scenarios. In future work, we will further improve the algorithm model to be applicable to both homogeneous and heterogeneous formations of multiple agents. Addressing issues of interpretability, interoperability logic, and permission handover will be key tasks.

CRediT authorship contribution statement

Jiandong Zhang: Conceptualization, Writing - Review & Editing, Validation, Software, Funding acquisition. **Dinghan**

Please cite this article in press as: ZHANG J et al. Loyal wingman task execution for future aerial combat: A hierarchical prior-based reinforcement learning approach, *Chin J Aeronaut* (2024), https://doi.org/10.1016/j.cja.2024.03.009

Wang: Conceptualization, Investigation, Writing - Original Draft, Methodology, Software, Data curation, Visualization, Formal analysis. **Qiming Yang:** Writing - Review & Editing, Software, Project administration, Resources. **Zhuoyong Shi:** Formal analysis. **Longmeng Ji:** Visualization. **Guoqing Shi:** Funding acquisition, Supervision. **Yong Wu:** Resources, Supervision.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

This study was co-supported by the Natural Science Basic Research Program of Shaanxi, China (No. 2022JQ-593), the Key R&D Program of Shaanxi Provincial Department of Science and Technology, China (No. 2022GY-089) and the Aeronautical Science Foundation of China (No. 20220013053005).

References

1. Sutton RS, Barto AG. *Reinforcement learning: an introduction*. Cambridge: MIT Press; 1998.
2. Xue LL, Zhou R, Ran H. Air combat decision based on genetic fuzzy tree. Yan L, Duan HB, Yu X. editors. Advances in guidance, navigation and control. *ICGNC 2020: Proceedings of 2020 international conference on guidance, navigation and control*. Singapore: Springer; 2022. p. 5515-25.
3. Huang HQ, Ding LY, Yang LW, et al. Air combat effectiveness evaluation for fighter based on relevance vector machine. *2020 IEEE international conference on artificial intelligence and computer applications (ICAICA)*. Piscataway: IEEE Press; 2020. p. 275–9.
4. Meng HD, Sun C, Feng YC, et al. One-to-one close air combat maneuver decision method based on target maneuver intention prediction. *2022 IEEE international conference on unmanned systems (ICUS)*. Piscataway: IEEE Press; 2022. p. 1454–65.
5. Guo CB, Zhang JN, Hu JW, et al. UAV air combat algorithm based on Bayesian probability model, In: *International conference on autonomous unmanned systems*. Singapore: Springer; 2023. p. 3176–85.
6. Liu Y, Yang Z, Huang JC, et al. A maneuver control method for stealthy engagement in beyond-visual-range air combat based on sliding mode control. *2022 22nd international conference on control, automation and systems (ICCAS)*. Piscataway: IEEE Press; 2022. p. 1333–8.
7. Chen YF, Sun XP, Liu DJ, et al. *Optimal guidance method for UCAV in close free air combat. 2019 IEEE international conferences on ubiquitous computing & communications (IUCC) and data science and computational intelligence (DSCI) and smart computing, networking and services (SmartCNS)*. Piscataway: IEEE Press; 2019. p. 356–60.
8. Tan RS, Gan XS, Wu N, et al. Location method of refueling airspace in air combat based on modified AFS algorithm. *2022 IEEE 5th international conference on automation, electronics and electrical engineering (AUTEEE)*. Piscataway: IEEE Press; 2022. p. 929–34.
9. Wu A, Yang RN, Liang XL, et al. Visual range maneuver decision of unmanned combat aerial vehicle based on fuzzy reasoning. *Int J Fuzzy Syst* 2022;**24**(1):519–36.
10. Zhong WJ, Li XB, Chang HT, et al. Design of air defense deployment optimization model based on adaptive nested PSO algorithm. *2021 2nd international conference on intelligent design (ICID)*. Piscataway: IEEE Press; 2021. p. 172–7.
11. Yang QM, Zhang JD, Shi GQ, et al. Maneuver decision of UAV in short-range air combat based on deep reinforcement learning. *IEEE Access* 2019;**8**:363–78.
12. Zhang JD, Yu YF, Zheng LH, et al. Situational continuity-based air combat autonomous maneuvering decision-making. *Def Technol* 2023;**29**:66–79.
13. Yang QM, Zhu Y, Zhang JD, et al. UAV air combat autonomous maneuver decision based on DDPG algorithm. *2019 IEEE 15th international conference on control and automation. (ICCA)*. Piscataway: IEEE Press; 2019. p. 37–42.
14. Zhang HP, Wei YJ, Zhou H, et al. Maneuver decision-making for autonomous air combat based on FRE-PPO. *Appl Sci* 2022;**12**(20):10230.
15. Wang Z, Li H, Wu HL, et al. Improving maneuver strategy in air combat by alternate freeze games with a deep reinforcement learning algorithm. *Math Probl Eng* 2020;**2020**:7180639.
16. Li B, Huang JY, Bai SX, et al. Autonomous air combat decision-making of UAV based on parallel self-play reinforcement learning. *CAAI Trans Intel Tech* 2023;**8**(1):64–81.
17. Li B, Bai SX, Liang SY, et al. Manoeuvre decision-making of unmanned aerial vehicles in air combat based on an expert actor-based soft actor critic algorithm. *CAAI Trans Intell Technol* 2023;**8**(4):1608–19.
18. Zhang YH, Yang Z, Chai SY, et al. Maneuver and attack strategy generation method for autonomous air combat in hybrid action space based on proximal policy optimization. *2023 42nd Chinese control conference (CCC)*. Piscataway: IEEE Press; 2023. p. 3946–53.
19. Zhang S, Zhou P, He Y, et al. Air combat maneuver decision-making test based on deep reinforcement learning. *Acta Aeronautica et Astronautica Sinica*. 2023;**44**(10) [Chinese].
20. Shan SZ, Zhang WW. Air combat intelligent decision-making method based on self-play and deep reinforcement learning. *Acta Aeronautica et Astronautica Sinica*. 2024;**45**(3) [Chinese]:028723.
21. Hui JP, Wang R, Guo JF. Intelligent guidance for no-fly zone avoidance based on reinforcement learning. *Acta Aeronautica et Astronautica Sinica*. 2023;**44**(11) [Chinese]:327416.
22. Jiang Y, Yu JL, Li QD. A novel decision-making algorithm for beyond visual range air combat based on deep reinforcement learning. *2022 37th youth academic annual conference of Chinese association of automation (YAC)*. Piscataway: IEEE Press; 2022. p. 516–21.
23. Hu TM, Hu JW, Zhao CH, et al. Autonomous decision making of UAV in short-range air combat based on DQN aided by expert knowledge, In: *International conference on autonomous unmanned systems*. Singapore: Springer; 2023. p. 1661–70.
24. Zhou XY, Huang JT, Zhu Z, et al. Intelligent air combat maneuvering decision based on TD3 algorithm, In: *International conference on autonomous unmanned systems*. Singapore: Springer; 2023. p. 1082–94.
25. Kong WR, Zhou DY, Yang Z, et al. UAV autonomous aerial combat maneuver strategy generation with observation error based on state-adversarial deep deterministic policy gradient and inverse reinforcement learning. *Electronics* 2020;**9**(7):1121.
26. Wu YF, Lei YL, Zhu Z, et al. Decision modeling and simulation of fighter air-to-ground combat based on reinforcement learning. In: *2022 4th international conference on image processing and machine vision (IPMV)*. New York: ACM; 2022. p. 102–9.
27. Lee GT, Kim CO. Autonomous control of combat unmanned aerial vehicles to evade surface-to-air missiles using deep reinforcement learning. *IEEE Access* 2020;**8**:226724–36.
28. Pope AP, Ide JS, Mićović D, et al. Hierarchical reinforcement learning for air-to-air combat. In: *2021 international conference on unmanned aircraft systems (ICUAS)*. Piscataway: IEEE Press; 2021. p. 275–84.

29. Zhang JD, Wang DH, Yang QM, et al. Multi-dimensional decision-making for UAV air combat based on hierarchical reinforcement learning. *Acta Armamentarii* 2023;**44**(6):1547 [Chinese].
30. Kong WR, Zhou DY, Du YJ, et al. Hierarchical multi-agent reinforcement learning for multi-aircraft close-range air combat. *IET Contr Theory Appl* 2023;**17**(13):1840–62.
31. Zhou WJ, Subagdja B, Tan AH, et al. Hierarchical control of multi-agent reinforcement learning team in real-time strategy (RTS) games. *Expert Syst Appl* 2021;**186**:115707.
32. Yuan YL, Yang J, Yu ZL, et al. Hierarchical goal-guided learning for the evasive maneuver of fixed-wing UAVs based on deep reinforcement learning. *J Intell Rob Syst* 2023;**109**(2):43.
33. Wang BL, Li SG, Gao XZ, et al. UAV swarm confrontation using hierarchical multiagent reinforcement learning. *Int J Aerosp Eng* 2021;**2021**:3360116.
34. Chai JJ, Chen WZ, Zhu YH, et al. A hierarchical deep reinforcement learning framework for 6-DOF UCAV air-to-air combat. *IEEE Trans Syst Man Cybern Syst* 2023;**53**(9):5417–29.
35. Qian CX, Zhang XB, Li L, et al. H3E: learning air combat with a three-level hierarchical framework embedding expert knowledge. *Expert Syst Appl* 2024;**245**:123084.

Please cite this article in press as: ZHANG J et al. Loyal wingman task execution for future aerial combat: A hierarchical prior-based reinforcement learning approach, *Chin J Aeronaut* (2024), https://doi.org/10.1016/j.cja.2024.03.009