aerospace

MDPI

Article

Hierarchical Decision-Making for UAV Close-Range Dynamic Tracking Using a Pursuit-Strategy Action Space

Yu Lai, Yong Chen, Yang Yang, Jialong Jian * and Yuanfei Liu *

Aviation Engineering School, Air Force Engineering University, Xi'an 710038, China
* Correspondence: j330809@163.com (J.J.); 13572287887@163.com (Y.L.)

Abstract

In close-range dynamic UAV tracking, the sharp decrease in relative distance and rapidly changing relative-motion conditions require UAVs to execute highly dynamic maneuvers. Traditional autonomous decision-making systems struggle with the curse of dimensionality in continuous action spaces or suffer from strategy-level rigidity when using predefined discrete maneuver primitives. This paper aims to resolve these limitations by developing a dimension-reduced yet highly continuous decision-making framework. We propose a hierarchical deep reinforcement learning architecture based on a geometric pursuit-strategy action space. The top-level Proximal Policy Optimization agent evaluates the relative-motion state to output discrete guidance-mode commands: lag pursuit, lead pursuit, or pure pursuit. A mid-level guidance translator converts these intents into continuous flight reference commands based on angular geometry and energy maneuverability. The bottom-level guidance translator utilizes a high-fidelity JSBSim fixed-wing aircraft flight-dynamics model for precise aerodynamic control. Monte Carlo simulations and comparative experiments across representative initial postures show that the proposed framework improves training convergence compared with a conventional continuous-control PPO baseline and achieves more stable high-level guidance-mode selection than a Double-DQN baseline. In simulation tests under predefined geometric tracking-success criteria, the model achieved a 91.5% success rate in initially favorable configurations and a 64.0% success rate when starting from a challenging configuration. By abstracting complex maneuvers into geometric pursuit strategies, this hierarchical framework lowers exploration dimensionality while maintaining the continuous kinematic logic of flight trajectories, providing an interpretable and simulation-validated decision-making framework for UAV close-range dynamic tracking and autonomous flight control.

Keywords: close-range dynamic tracking; deep reinforcement learning; hierarchical decision-making; pursuit strategy; geometric guidance law; action space dimensionality reduction

Check for updates

Academic Editor: Gokhan Inalhan

Received: 8 April 2026
Revised: 17 May 2026
Accepted: 28 May 2026
Published: 29 May 2026

Copyright: © 2026 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license.

1. Introduction

With the rapid development of unmanned aerial vehicle (UAV) technology, close-range dynamic tracking has become an active research topic in UAV autonomous decision-making [1]. During close-range dynamic-tracking scenarios, the dramatically reduced relative distance often compels UAVs to perform drastic maneuvers; consequently, the relative-motion process exhibits highly dynamic and strongly nonlinear characteristics [2], thereby placing stringent demands on the real-time capability and rationality of autonomous

decision-making systems [3]. Recently, deep reinforcement learning (DRL) has been increasingly applied to UAV maneuvering and dynamic-tracking decision-making, primarily owing to its capacity to optimize action strategies through interaction with the environment while reducing reliance on manually designed decision rules [4]. To mitigate the curse of dimensionality associated with high-dimensional continuous decision spaces, many recent studies have adopted hierarchical reinforcement learning (HRL) frameworks, which decompose complex decision-making problems into temporally abstract subtasks or options, thereby reducing the effective exploration horizon and improving policy-learning efficiency [5]. Nevertheless, existing hierarchical UAV maneuvering decision-making frameworks still possess certain limitations in how they define the "strategy-level execution layer" and design the "top-level action space." These limitations motivate a closer examination of the gap between high-level guidance-mode representation and low-level executable flight commands.

Beyond UAV maneuvering studies, interaction-aware intent prediction has also been extensively investigated in autonomous driving, where an autonomous agent must infer the latent preferences, aggressiveness, or semantic intentions of surrounding agents before selecting its own maneuver. For example, Hu et al. [6] formulated lane-changing for autonomous heavy vehicles as socially interactive game-theoretic agents under asymmetric driving aggressiveness, indicating that game-theoretic reasoning can support maneuver decisions in highly interactive traffic scenarios. Deng et al. [7] further modeled lane-change decision-making as an incomplete-information game and quantified social driving preferences through driver aggressiveness, thereby reducing uncertainty in the interaction process. In addition to game-theoretic mechanisms, recent multimodal decision-making studies have explored the fusion of visual, LiDAR, and task-oriented textual cues for interpretable high-level driving decisions through semantic reasoning [8]. Although close-range UAV dynamic tracking differs from highway lane-changing in vehicle dynamics, task objectives, and safety constraints, both tasks require interaction-aware high-level decision-making under rapidly changing multi-agent situations. Inspired by these studies, the high-level PPO policy in this paper is interpreted as a guidance-mode selection module that maps the current relative-motion state to one of three pursuit intents—lag pursuit, lead pursuit, or pure pursuit—while the mid-level guidance laws and low-level flight controller ensure physically feasible execution.

Some studies have attempted to define the top-level action space as continuous flight reference commands, such as reference heading, pitch angle, or velocity commands. Yuan et al. [9] developed a UAV maneuvering decision model with a continuous action space utilizing the Deep Deterministic Policy Gradient algorithm, where the action space was defined as continuous changes in throttle, angle of attack, and roll angle. Chai et al. [10] introduced a hierarchical deep reinforcement learning decision framework whose top-level outer loop utilizes the Proximal Policy Optimization (PPO) algorithm to directly yield reference roll and pitch angular rates as macroscopic continuous action commands; by integrating a self-play mechanism for policy iteration, it demonstrated complex maneuvering capabilities in simulations. To meet the real-time requirements of continuous trajectory control and decision-making, Tan et al. [11] proposed a Surrogate-Assisted Differential Evolution algorithm for online trajectory planning, utilizing continuous variables such as throttle, angle of attack, and roll angle as decision variables, and achieved high-fidelity trajectory tracking in combination with Nonlinear Dynamic Inversion control, validating its effectiveness under various initial situations through Monte Carlo simulations. By utilizing variable-scale virtual tracking points and velocities as action outputs and transforming them into continuous flight path angle commands, Wang et al. [12], in conjunction with a BP-LSTM trajectory prediction network, attained a superior task success rate in close-range

maneuvering simulations relative to conventional strategies employing fixed action amplitudes. Although continuous action spaces can theoretically provide a complete flight envelope, this underlying control space is excessively large, causing the agent to face the dilemmas of low exploration efficiency and sparse rewards, ultimately making it highly susceptible to falling into local optima and difficult to converge [13].

In an effort to avoid the combinatorial explosion inherent in continuous spaces, numerous researchers have shifted towards discretization strategies, defining the middle layer as a set of predefined discrete maneuver primitives [14] or pre-established expert rules. Cao et al. [15] discretized the action space into a set of 7 NASA basic maneuvers and combined the Double Deep Q-Network with the minimax algorithm for maneuver decision-making. By extracting expert knowledge from expert maneuvering manuals, Yang et al. [16] developed an automated maneuvering decision system using a Behavior Tree model, allowing aircraft to select appropriate predefined maneuver modes under different relative-motion configurations. For long-range UAV maneuvering scenarios, Wang et al. [17] introduced a hierarchical hybrid decision architecture, defining the strategy-level execution layer's action space as a set of predefined discrete maneuvering rules and realizing action transitions via a rule-driven state switching mechanism. To reduce the dimensional complexity of decision actions when employing hierarchical reinforcement learning for multi-UAV maneuvering decision-making, Wang et al. [18] discretized flight control actions into fundamental discrete action sets comprising heading, altitude, and speed.

While defining the action space as predefined discrete maneuver primitives or fundamental discrete commands effectively lowers the decision-making dimensions, such an approach fragments the continuous nature of close-range flight dynamics [19]. When tracking rapidly maneuvering objects, relying on the switching of predefined discrete maneuvers not only constrains the UAV's kinematic capabilities but often results in rigid policy execution, thereby making the UAV susceptible to missing transient geometric alignment opportunities. To overcome the constraints associated with fixed-duration strategy-level execution and inflexible actions, several studies have started exploring more adaptable hierarchical frameworks. Qian et al. [20] introduced a three-layer hierarchical decision framework (H3E) embedding expert knowledge, which sought to improve decision flexibility by employing a top-level strategy selector to coordinate mid-level guidance modes for different relative-motion conditions. Li et al. [21] proposed a hierarchical reinforcement learning framework based on a maximum entropy objective, designed three guidance strategies with clear geometric and energy logic—angle, snapshot, and energy—at the bottom layer, and realized the dynamic termination and flexible selection of guidance modes at the high layer.

As indicated by the preceding research, a key aspect of strategy-level decision-making in close-range interactive tracking is to maintain relative angular advantage [22]. However, the above studies still leave a methodological gap between high-level guidance-mode representation and low-level continuous flight execution. Continuous-action methods, such as those in [9–12], directly output flight-control or reference-command variables and therefore preserve maneuver flexibility, but they do not explicitly reduce the exploration burden caused by the large continuous control space. In contrast, maneuver-template- or rule-library-based methods [15–18] reduce the action dimensionality by introducing predefined discrete maneuvers, but their fixed maneuver templates may fragment the continuous geometric evolution of close-range relative motion and limit the timely capture of transient geometric alignment opportunities. Existing hierarchical frameworks [20,21] improve strategy-level abstraction to some extent, yet they still do not explicitly construct an action space that is both compact at the decision level and continuously executable at

the guidance level. To bridge this gap and address the control requirements of maintaining favorable relative geometry and anticipatory tracking in close-range dynamic flight, this paper proposes a hierarchical UAV maneuvering reinforcement learning decision-making method based on a pursuit-strategy action space. In this framework, the high-level policy selects among three interpretable guidance-mode intents—lag pursuit, lead pursuit, and pure pursuit—while the mid-level guidance laws continuously translate these intents into heading, altitude, and velocity commands for the low-level flight controller. The primary contributions of this study, which distinguish the proposed framework from existing continuous-action methods, maneuver-template-based discrete-action methods, and general hierarchical UAV maneuvering decision methods, are summarized below:

1. Formulation of a dimensionally reduced but trajectory-continuous action space anchored in pursuit strategies: Different from existing methods that assign top-level outputs either to low-level continuous control commands or to fixed-duration maneuver templates, this study defines the top-level decision-making action space as three geometric pursuit strategies: lag pursuit, lead pursuit, and pure pursuit. Unlike conventional discrete maneuvers, each action corresponds to a continuously updated guidance law rather than a rigid maneuver template.
2. Development of strategy-level guidance laws grounded in angular geometry: Specifically addressing the characteristics of close-range anticipatory tracking, mathematical models for lag pursuit, lead pursuit, and pure pursuit are formulated. These models serve as an intermediary layer to translate top-level guidance-mode intentions into continuous low-level flight reference parameters.
3. Realization of high-fidelity hierarchical closed-loop validation: by constructing a low-level flight control model within the JSBSim flight-dynamics environment, a hierarchical closed-loop architecture is established alongside mid-level geometric guidance and top-level pursuit-strategy selection, which enables the proposed framework to be evaluated under nonlinear aerodynamic constraints.

2. Materials and Methods

2.1. Comprehensive Architectural Overview

To address the challenges posed by continuous decision spaces and highly dynamic relative-motion conditions in close-range UAV tracking, this paper presents a hierarchical decision framework designed to decouple the intricate "maneuvering-control" process into logically distinct task, guidance, and execution layers. By means of functional decoupling, this framework mitigates the training complexity associated with reinforcement learning algorithms, with its general architecture illustrated in Figure 1.

2.2. Flight Controller

The flight controller is constructed in the JSBSim flight-dynamics simulation environment using a fixed-wing aircraft model that provides nonlinear six-degree-of-freedom state feedback [23]. Therefore, the proposed guidance-mode decisions are evaluated not only at the geometric-kinematic level but also through a nonlinear flight-dynamics loop involving attitude response, velocity variation, actuator deflection, and aerodynamic constraints. Given the strongly coupled nature of the aircraft's equations of motion, we follow the methodology of Li et al., utilizing the PPO algorithm to train the flight controller [24]. The network architecture and training hyperparameters of the low-level flight-controller PPO are provided in Appendix A, Table A1. The designated state space incorporates attitude angles, three-axis velocity components, and expected command tracking errors, whereas the output actions map directly to normalized commands for the elevator, ailerons, rudder,

and throttle. To ensure smooth control commands, a first-order low-pass filter is applied prior to action output:

$$\delta(k) = \alpha\delta_{\text{ppo}}(k) + (1-\alpha)\delta(k-1) \tag{1}$$

Addressing the control conflict between turn rate and flight speed, a combined reward function based on adaptive weights was designed [25]:

$$r = W_1(\phi_{err})e_{\text{heading}} + W_2 e_{\text{speed}} + W_3 e_{\text{altitude}} \tag{2}$$

Here, since the aircraft cannot sustain the maximum turn rate at high velocities, dynamic speed weights are established such that the speed weight progressively increases as the heading error decreases, thereby achieving coordinated execution of turning and acceleration.

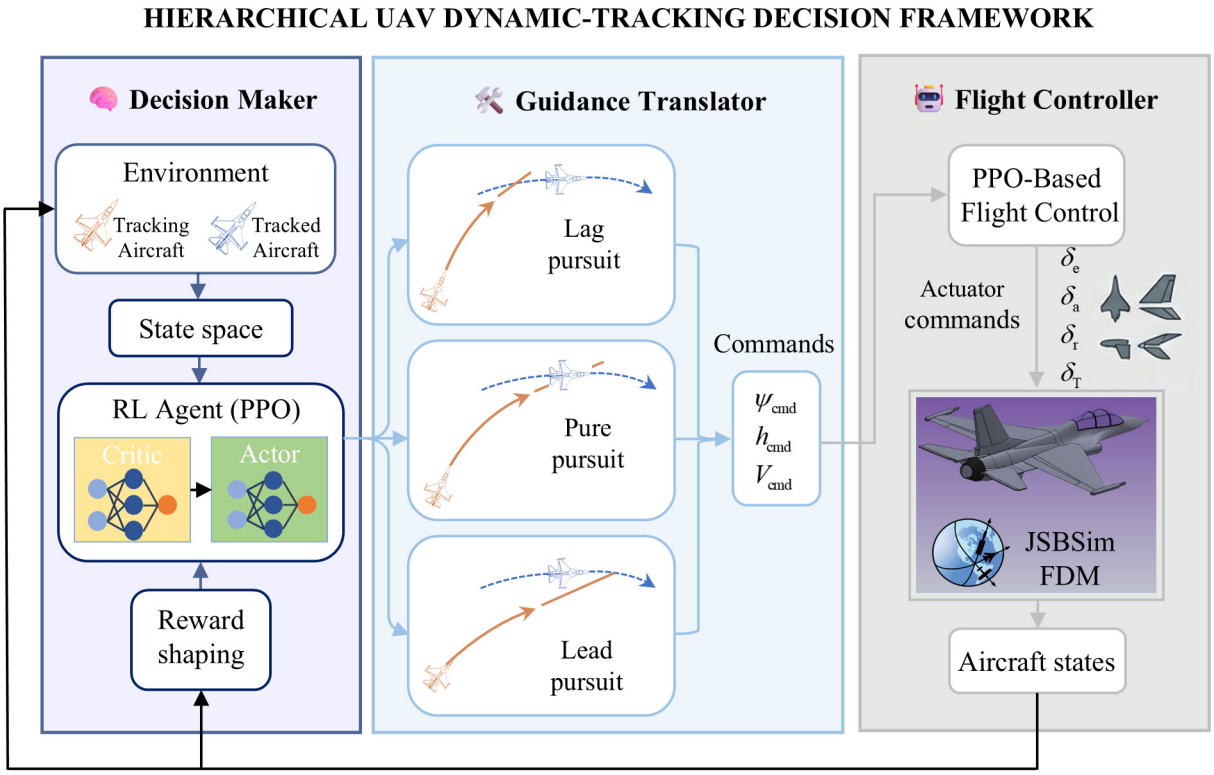


Figure 1. Overall architecture of the hierarchical UAV dynamic-tracking decision framework. The framework consists of three core modules: (1) the decision maker utilizes a PPO agent to output discrete high-level guidance-mode intents based on the current relative-motion state; (2) the guidance translator converts these intents into three typical pursuit strategies and calculates the desired heading, altitude, and velocity commands; and (3) the flight controller tracks these commands and outputs actuator and throttle deflections to the JSBSim flight-dynamics model. This closed-loop design allows the guidance-mode selection policy to be evaluated under the tracking errors and dynamic response delays generated by the flight controller and the JSBSim flight-dynamics model.

After training in a parallelized environment, this low-level model can track the reference heading, altitude, and velocity commands generated by the mid-level geometric guidance laws. Consequently, the high-level guidance-mode selection policy is tested under physically constrained command tracking rather than idealized instantaneous maneuver execution, which improves the credibility of the validation for highly dynamic level-flight, turning, and climbing maneuvers.

2.3. Definition of State Space and Relative-Motion Characteristics

To quantitatively depict the close-range relative-motion state and effectively steer top-level decision-making, the state space essential for state assessment must satisfy the Markov property while reflecting the spatial geometry and kinematic states of both aircraft [26]. In the present implementation, the MDP is formulated using ideal state observations directly provided by the JSBSim-based simulation environment; wind disturbances, sensor measurement noise, and partial observability are not explicitly modeled in the state-transition process, which limits the direct extension of the current results to real-world disturbed flight conditions. Therefore, the state space vector S defined in Table 1 should be interpreted as a fully observable simulation-state representation.

Table 1. Definition of state space feature vectors.

Symbol	Definition & Physical Meaning
R	Line-of-sight range between both aircraft, used to determine whether the tracked aircraft satisfies the predefined close-range geometric condition.
$\phi = \cos^{-1}\left[\frac{\mathbf{V_A}\cdot\boldsymbol{\rho}}{\lvert\mathbf{V_A}\rvert\lvert\boldsymbol{\rho}\rvert}\right]$	Antenna train angle (ATA), the angle between the tracking aircraft's velocity vector and the line-of-sight vector.
$q = \pi - \cos^{-1}\left[\frac{\mathbf{V_T}\cdot\boldsymbol{\rho}}{\lvert\mathbf{V_T}\rvert\lvert\boldsymbol{\rho}\rvert}\right]$	Aspect angle (AA), the angle between the tracked aircraft's velocity vector and the line-of-sight vector.
Δz	Relative altitude, used to evaluate the margin for potential energy conversion and assess the potential energy difference between the two aircraft.
$V_{\mathrm{A}},\ V_{\mathrm{T}}$	Flight airspeeds of the tracking aircraft and the tracked aircraft, which determine their respective kinetic energy levels.
$n_z,\ n_{z_t}$	Current normal load factors of both aircraft, reflecting the intensity of their maneuvers.
α, β, γ	α, β, γ represent the angle of attack, sideslip angle, and flight-path angle, respectively. In the local North-East-Up inertial frame, the flight-path angle is defined by $\gamma = \arcsin(V_U/\lVert V\rVert)$, where V_U is the vertical component of the velocity vector and $\lVert V\rVert$ is the flight speed. The pitch angle θ describes the attitude of the aircraft body, whereas γ describes the direction of the velocity vector; for coordinated small-sideslip flight, $\theta \approx \gamma + \alpha$.
$S = \{x_{\mathrm{A}}, y_{\mathrm{A}}, h_{\mathrm{A}}, \psi_{\mathrm{A}}, V_{\mathrm{A}}, \phi_{\mathrm{A}}, q_{\mathrm{T}}, \Delta V, \Delta h, R\}$	The complete state-space vector used for relative-motion state assessment.

Leveraging these feature vectors, the system is capable of performing real-time evaluations to determine whether the current relative geometry is favorable, neutral, or challenging, thereby providing a foundational basis for guidance-mode transitions. The geometric relative-motion configuration between the two aircraft is illustrated in Figure 2. Unless otherwise stated, all positions are expressed in a local North-East-Up inertial frame $F_{\mathrm{I}} = \{O_{\mathrm{I}}, x_{\mathrm{N}}, y_{\mathrm{E}}, z_{\mathrm{U}}\}$, where x_{N} points north, y_{E} points east, and z_{U} is positive upward and is equivalent to the altitude h used in the figures. The tracking-aircraft body frame F_{A} and tracked-aircraft body frame F_{T} are fixed to the corresponding aircraft, with the x_b axis pointing forward along the fuselage, y_b pointing to the right wing, and z_b completing the right-handed frame. The line-of-sight vector is defined as $\rho = \mathrm{Pos}_{\mathrm{T}} - \mathrm{Pos}_{\mathrm{A}}$, and the line-of-sight range is $R = \lVert\rho\rVert$.

From the perspective of the tracking aircraft, four representative relative-motion configurations are depicted in Figure 3.

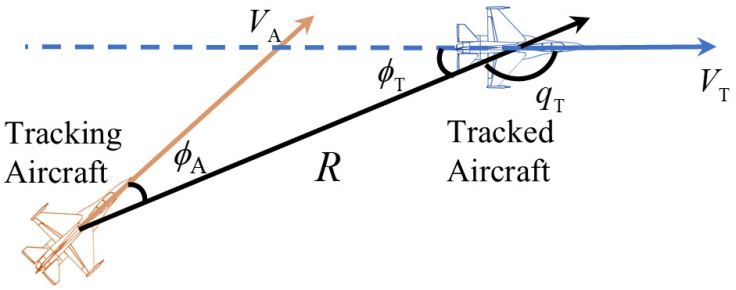
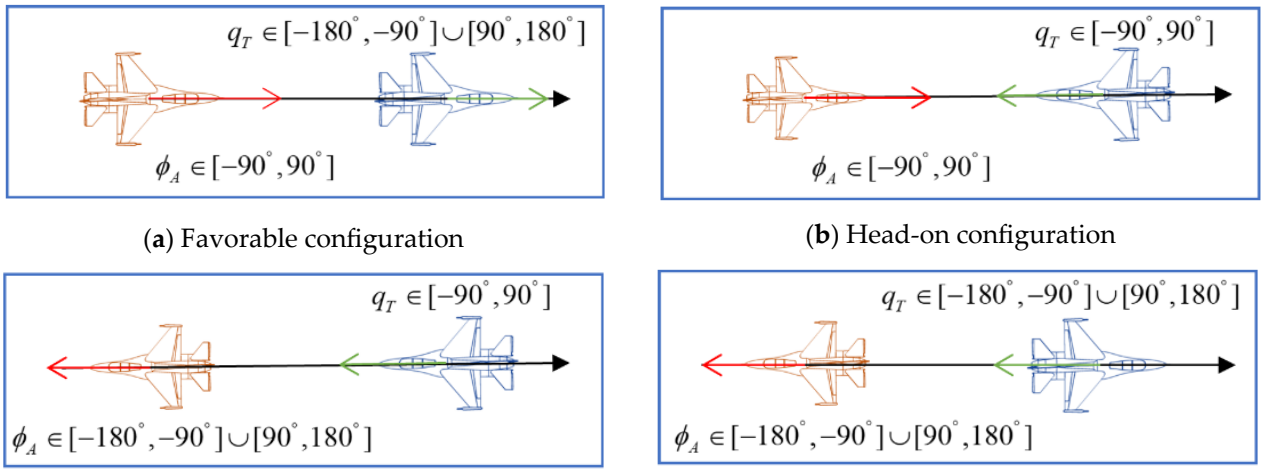


Figure 2. The geometric relative-motion configuration of the two aircraft.



(**a**) Favorable configuration (**b**) Head-on configuration

(**c**) Challenging configuration (**d**) Neutral

Figure 3. Schematic diagram of four typical close-range relative-motion configurations.

Since long-range interaction mechanisms are outside the scope of this study, the simulation outcome is determined by a predefined close-range geometric tracking rule. Each aircraft is assigned an initial simulation score of 100. A tracked aircraft is considered to satisfy the predefined geometric tracking condition when the range is not greater than 900 m and the tracking-aircraft-side ATA is not greater than 25°. To avoid unrealistic instantaneous success judgments caused by transient entry into the predefined geometric condition, the simulation score is updated only after the tracked aircraft remains continuously inside the specified geometric region for 0.2 s. After this dwell-time threshold is satisfied, the tracked aircraft's simulation score decreases at a rate of 50 score units per second. The aircraft whose simulation score reaches zero first is judged to have failed to maintain the required tracking condition. If both aircraft reach zero simulation score within the same simulation step, the result is recorded as a simultaneous termination case. To reduce boundary jitter, a hysteresis mechanism is adopted: after entering the specified geometric region, the tracked aircraft is regarded as remaining inside the region until the range exceeds 950 m or the ATA exceeds 30°.

2.4. Geometric Guidance Laws and High-Level Decision-Making Model

To convert the guidance-mode intents from high-level decisions into low-level executable control commands, this paper designs three typical pursuit guidance laws based on angular-maneuver geometry. Lag pursuit is used to maintain and improve favorable relative geometry, lead pursuit is employed to predict the tracked aircraft's future position and reserve an anticipatory tracking angle, and pure pursuit acts as a rapid angular-recovery strategy under challenging conditions [27]. These strategies are selected because they correspond to three representative guidance requirements in close-range dynamic tracking: maintaining favorable relative geometry, capturing an anticipatory tracking solution, and re-

covering from angular misalignment. As summarized in Table 2, the three-strategy set is not intended to exhaust all possible maneuvers; rather, it provides a compact and interpretable guidance-mode action space for validating the proposed hierarchical decision framework.

Table 2. Guidance purpose, applicable relative-motion condition, and limitation of the three pursuit strategies.

Strategy	Guidance Purpose	Typical Applicable Relative-Motion Condition	Possible Limitation
Lag pursuit	Maintains or enlarges angular alignment advantage while preventing overshoot by guiding the tracking aircraft toward the rear region of the tracked aircraft.	The tracking aircraft has obtained a preliminary favorable relative position or is approaching the tracked aircraft with excessive closure speed; the main objective is to remain within a favorable tracking region while avoiding crossing the tracked aircraft's turn circle.	It may delay geometric-alignment acquisition when the tracking aircraft needs to rapidly align the nose with the tracked aircraft; in extreme challenging situations, lag pursuit alone may not provide sufficient angular recovery.
Lead pursuit	Captures an anticipatory tracking geometry by aiming ahead of the tracked aircraft and compensating for prediction delay and vertical motion effects.	The tracking aircraft is close to satisfying the predefined geometric tracking condition and needs to reserve an anticipatory angle for tracking a maneuvering object.	The current implementation uses a guidance-level first-order prediction of the tracked aircraft position; under rapidly varying three-dimensional maneuvers, higher-order trajectory-prediction and acceleration modeling may be required.
Pure pursuit	Rapidly reduces the line-of-sight angle by pointing directly toward the tracked aircraft's current position, serving as an angular-recovery mechanism under challenging relative geometry.	The tracking aircraft is in a challenging or neutral angular position and needs to quickly regain nose-pointing direction toward the tracked aircraft.	Pure pursuit can cause excessive closure speed and may lead to overshoot or altitude loss if not coordinated with energy and safety constraints. It is not intended to replace more complex emergency recovery maneuvers.

2.4.1. Relative-Motion State Assessment and High-Level Decision-Making Mechanism

The PPO algorithm is adopted at the top level, with the action space defined as $A = \left\{a_{lag}, a_{lead}, a_{pure}\right\}$. The three actions correspond to three guidance-mode intents: a_{lag} aims to maintain angular advantage and avoid overshoot, a_{lead} aims to capture an anticipatory tracking geometry by aiming ahead of the tracked aircraft, and a_{pure} aims to rapidly reduce the line-of-sight angle under challenging conditions. The high-level PPO does not rely on artificially established angular or distance thresholds for policy switching; instead, through environmental interaction and long-term cumulative reward maximization, it implicitly learns the mapping $\pi(a\|S)$ from varying relative-motion states S to optimal guidance-mode actions a. This process can also be interpreted as online guidance-mode selection, since interaction-aware UAV decision-making studies have shown that inferring another aircraft's motion tendency from trajectory and relative-state features is a key prerequisite for state awareness and autonomous maneuver decision-making [28]. In other words, the above guidance-mode roles are used to define the meaning of the action space, while the final switching boundary among lag, lead, and pure pursuit is learned from data rather than manually prescribed.

Based on the variations in the azimuth angle ϕ and the tracked-aircraft aspect angle q, and following the common practice of constructing relative-motion state and reward indicators from angular, range, and altitude-related factors [29,30], the situation value S_{A} of the tracking UAV and the situation value S_{T} of the tracked UAV are evaluated as follows. Specifically, the azimuth angle reflects the nose-pointing and alignment advantage of the

tracking aircraft, whereas the tracked-aircraft aspect angle reflects whether the tracking aircraft is approaching the tracked aircraft's rear tracking region.

$$S_{\mathrm{A}} = \begin{cases} \dfrac{k(\pi - |\phi_{\mathrm{A}}|)^2 + (1-k)\left|q_{\mathrm{T}}\right|^2}{\pi^2}, & R \le d_{\mathrm{ref}} \\ \dfrac{k(\pi - |\phi_{\mathrm{A}}|)^2 + (1-k)\left|q_{\mathrm{T}}\right|^2}{\pi^2} \cdot e^{\frac{d_{\mathrm{ref}} - R}{d_{\mathrm{ref}}}}, & R > d_{\mathrm{ref}} \end{cases} \tag{3}$$

$$S_{\mathrm{T}} = \begin{cases} \dfrac{k(\pi - |\phi_{\mathrm{T}}|)^2 + (1-k)\left|q_{\mathrm{A}}\right|^2}{\pi^2}, & R \le d_{\mathrm{ref}} \\ \dfrac{k(\pi - |\phi_{\mathrm{T}}|)^2 + (1-k)\left|q_{\mathrm{A}}\right|^2}{\pi^2} \cdot e^{\frac{d_{\mathrm{ref}} - R}{d_{\mathrm{ref}}}}, & R > d_{\mathrm{ref}} \end{cases} \tag{4}$$

where d_{ref} is the maximum reference range for the predefined tracking condition, used as a geometric range reference in the situation-value and reward-shaping functions, and $k \in [0, 1]$ is the weight coefficient used to balance two guidance preferences: the azimuth-angle term associated with nose-pointing accuracy and the aspect-angle term associated with tail-chase positioning. This function accounts for the correlation among azimuth-angle alignment, tracked-aircraft-aspect geometry, and the geometric range reference d_{ref}. When $R \le d_{ref}$, angular alignment directly contributes to the state value; when $R > d_{ref}$, an exponential range-decay factor is introduced to penalize geometrically favorable but tracking-ineffective long-range states. The larger the value of k, the greater the variation of the situational value relative to the azimuth angle $|\phi|$, meaning that the reward becomes more sensitive to nose-pointing accuracy. Conversely, a smaller k assigns relatively greater importance to target-aspect superiority and tail-chase positioning. A related analysis of the situation-evaluation function is shown in Figure 4.

The guidance reward function is formulated based on the state-evaluation function as follows. Compared with using only sparse terminal rewards, this shaped process reward provides dense feedback for angular alignment and effective-range acquisition, thereby guiding the PPO agent to learn meaningful guidance-mode transitions before the terminal success/failure state is reached:

$$r_{process} = \begin{cases} \dfrac{k(\pi - |\phi_{\mathrm{A}}|)^2 + (1-k)\left|q_{\mathrm{T}}\right|^2}{\pi^2}, & R \le d_{ref} \\ \dfrac{k(\pi - |\phi_{\mathrm{A}}|)^2 + (1-k)\left|q_{\mathrm{T}}\right|^2}{\pi^2} \cdot e^{\frac{d_{ref} - R}{d_{ref}}}, & R > d_{ref} \end{cases} \tag{5}$$

Furthermore, to reinforce the agent's awareness of the terminal simulation states, a terminal reward function is configured as follows. The terminal reward distinguishes success, simultaneous termination, and failure outcomes, while the process reward in Equation (5) shapes the intermediate guidance behavior; thus, the final reward combines geometric guidance during close-range tracking with explicit termination-state evaluation.

$$r_{end} = \begin{cases} 10, \text{ success} \\ 0, \text{ simultaneous termination} \\ -10, \text{ failure} \end{cases} \tag{6}$$

Thus, the ultimate reward function is given by $r = r_{\mathrm{process}} + r_{\mathrm{end}}$.

It should be emphasized that the coefficient k is not updated online during PPO training. Instead, k is treated as a fixed design parameter that determines the shape of the reward landscape before training starts. The surfaces in Figure 4 are used to analyze the sensitivity of the state-assessment function to different k values: a smaller k assigns relatively more importance to target-aspect superiority and tail-chase positioning, whereas a larger k makes the reward more sensitive to azimuth-angle reduction and nose-pointing accuracy. In the present training process, a fixed value of k is used, and the PPO agent adapts

its guidance-mode policy to the corresponding reward landscape rather than adaptively changing k during interaction.

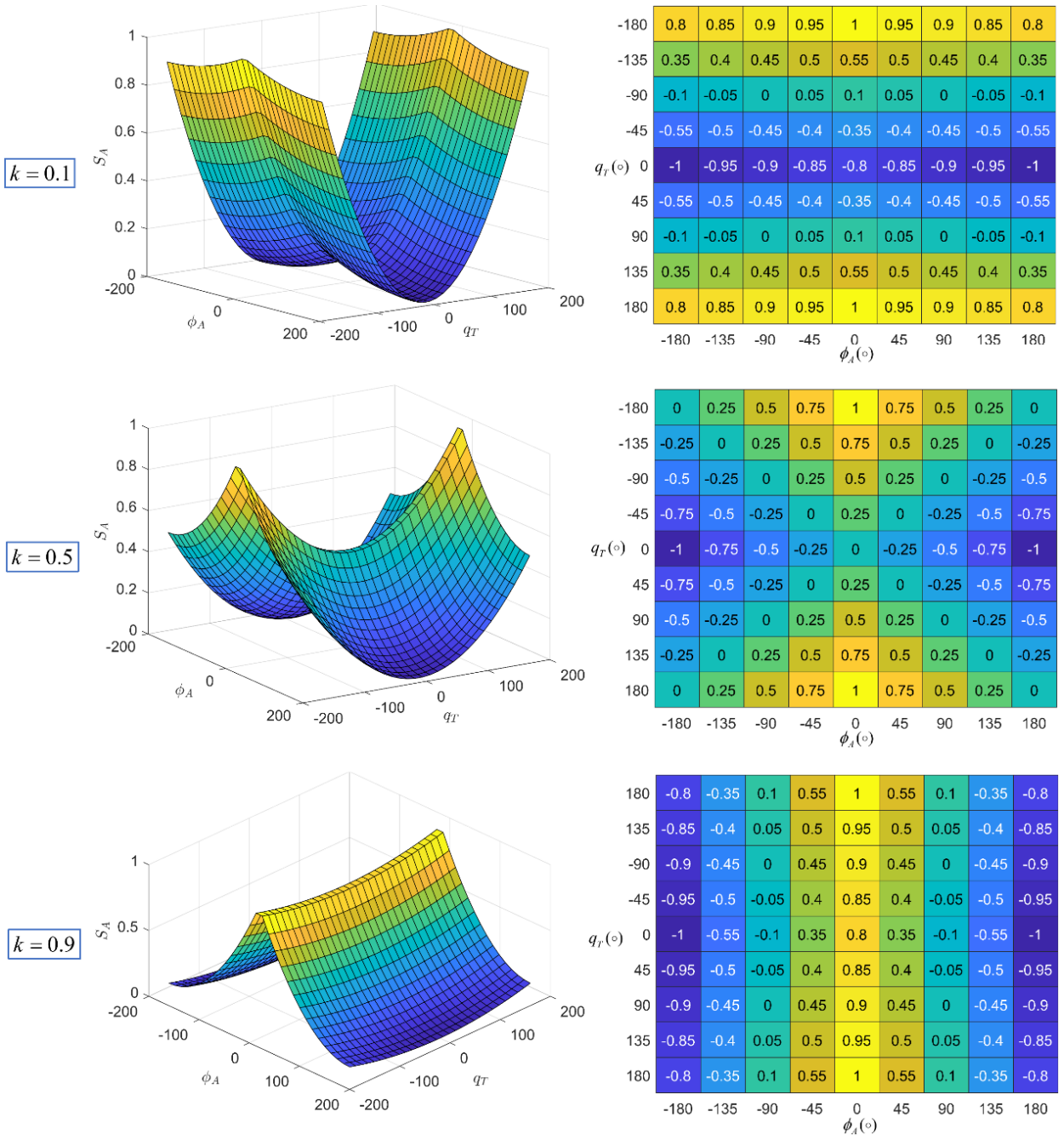


Figure 4. An analysis of the state-evaluation function. The three-dimensional surfaces illustrate the mapping from relative geometric states to the normalized relative-geometry evaluation value under different k values, while the heat maps show the corresponding angular reward distribution over ϕ_A and q_T. The evolution of the surface from $k = 0.1$ to $k = 0.9$ demonstrates the sensitivity of the assessment logic to different guidance priorities: a lower k emphasizes the acquisition of favorable rear-region tracking geometry, while a higher k prioritizes the precision of the nose-pointing direction required for close-range alignment. Warmer colors indicate higher normalized relative-geometry evaluation values, whereas cooler colors indicate lower values.

2.4.2. Lag Pursuit Guidance Law

When the tracking aircraft has a favorable relative position, lag pursuit involves pointing the nose towards the rear of the tracked aircraft, effectively preventing an overshoot due to excessive speed while simultaneously closing the distance. Its geometric guidance core lies in reserving the reference tracking distance d_{ref} and designing the position command based on the coordinate transformation matrix:

$$\begin{cases} \mathbf{Pos}_{\text{cmd}}^{\text{lag}} = \mathbf{Pos}_{\text{T}} + \mathbf{C}_b^I(\psi_{\text{T}}, \theta_{\text{T}}, \varphi_{\text{T}})[d_{\text{ref}}, 0, 0] \\ \mathbf{e}_{\text{P}}^{\text{lag}} = \mathbf{Pos}_{\text{cmd}}^{\text{lag}} - \mathbf{Pos}_{\text{A}} \end{cases} \tag{7}$$

Here, $\mathbf{Pos}_{\text{T}} = [x_{\text{T}}, y_{\text{T}}, h_{\text{T}}]^T$ represents the current position vector of the tracked aircraft in the local North-East-Up inertial frame. $\mathbf{C}_b^I(\psi_{\text{T}}, \theta_{\text{T}}, \varphi_{\text{T}})$ denotes the transformation matrix from the tracked-aircraft body frame to the local inertial frame, where ψ_{T}, θ_{T}, and φ_{T} are the yaw, pitch, and roll angles of the tracked aircraft, respectively. The flight-path angle γ_{T} is computed from the tracked-aircraft velocity vector and is not used as an Euler attitude angle; under coordinated small-sideslip flight, $\theta_{\text{T}} \approx \gamma_{\text{T}} + \alpha_{\text{T}}$.

Upon closing the distance, a velocity tracking error is introduced for braking to ensure that the tracking aircraft does not overshoot the tracked aircraft's turning circle. Consequently, the velocity command under the lag pursuit strategy is designed as:

$$\begin{cases} \mathbf{e}_{\text{V}}^{\text{lag}} = \mathbf{V}_{\text{T}} - \mathbf{V}_{\text{A}} \\ \mathbf{V}_{\text{cmd}}^{\text{lag}} = \mathbf{V}_{\text{T}} + \mathbf{C}_b^I(\psi_{\text{T}}, \theta_{\text{T}}, \varphi_{\text{T}})(K_{\text{VP}}\mathbf{e}_{\text{P}} + K_{\text{VD}}\mathbf{e}_V) \end{cases} \tag{8}$$

The tracking UAV must not only track the velocity of the tracked aircraft V_{T}, but also apply dynamic corrections via the feedback from the position error e_p and velocity error e_v, with K_{VP} and K_{VD} acting as control gain coefficients.

After determining the position command and velocity command, these commands need to be converted into heading angle commands and altitude commands. Among them, the heading command is computed using the four-quadrant arctangent function to avoid quadrant ambiguity:

$$\psi_{\text{cmd}}^{\text{lag}} = atan2\left(\frac{y_{\text{cmd}} - y_{\text{A}}}{x_{\text{cmd}} - x_{\text{A}}}\right) \tag{9}$$

To avoid excessive depletion of kinetic energy during maneuvers, an altitude compensation command is formulated based on the energy maneuverability theory [31]: $\Delta h = h - h' = \frac{V^2 - V_{\text{cmd}}^2}{2g}$.

Consequently, the modified altitude command is: $h'^{\text{lag}}_{\text{cmd}} = h_{\text{cmd}} + \Delta h$.

2.4.3. Lead Pursuit Guidance Law

Lead pursuit is employed to establish an anticipatory tracking geometry when the predefined close-range geometric condition is satisfied. By forecasting the future position of the tracked aircraft, the tracking aircraft is guided to proactively capture the required anticipatory angle for close-range tracking. Let the reference propagation velocity be v_{ref} and the tracked aircraft's velocity be V_{T}. For computational efficiency in the guidance layer, the prediction time offset is approximately computed as:

$$t_{\text{PRE}} \approx \frac{R}{v_{\text{ref}} + V_{\text{T}}\cos(\phi)} \tag{10}$$

The relationship between the anticipated tracking point in lead pursuit $\mathbf{Pos}_{\text{Lead}}$ and the tracked aircraft's position at the subsequent time step $\mathbf{Pos}_{\text{T}+t_{\text{PRE}}}$ is defined as:

$$\begin{aligned} \mathbf{Pos}_{\text{Lead}} &= \mathbf{Pos}_{\text{T}+t_{\text{PRE}}} - [0,0,h_g] + [0,0,h_\alpha] \\ &= \mathbf{Pos}_{\text{T}+t_{\text{PRE}}} - \left[0,0,\tfrac{1}{2}gt_{\text{PRE}}^2\right] + [0,0,d\tan\alpha] \end{aligned} \tag{11}$$

In the formula, h_g represents the compensation for vertical deviation during the prediction interval, and $h_\alpha = d\tan\alpha$ denotes the axis-elevation compensation caused by the aircraft's angle of attack. This simplified compensation improves the consistency between the lead-pursuit command and the simulated flight state, but it does not fully account for unmodeled propagation effects, tracked-aircraft jerk, or rapidly varying three-dimensional rolling maneuvers.

By integrating the tracked aircraft's current load factor and velocity to predict its future location, the guidance command for the anticipated tracking point is generated as follows:

$$\begin{cases} \mathbf{Pos}_{\text{D}}^{\text{lead}} = \mathbf{Pos}_{\text{T}} + \mathbf{C}_b^I(\psi_{\text{T}},\theta_{\text{T}},\varphi_{\text{T}})[d_{\text{ref}},0,0] \\ \mathbf{e}_p = \mathbf{Pos}_{\text{D}}^{\text{lead}} - \mathbf{Pos}_{\text{A}} \\ \mathbf{Pos}_{\text{cmd}}^{\text{lead}} = \mathbf{Pos}_{\text{Lead}} + \mathbf{C}_b^I(\psi_{\text{T}},\theta_{\text{T}},\varphi_{\text{T}})(K_p\mathbf{e}_p + [d\text{ref},0,0]) \end{cases} \tag{12}$$

This prediction is used as a guidance-level approximation for lead-pursuit guidance rather than as a complete high-fidelity trajectory-prediction solution.

In lead pursuit, an appropriate reference tracking range from the tracked aircraft needs to be maintained. When the two aircraft approach each other, the tracking aircraft's speed is regulated to track that of the tracked aircraft; therefore, the velocity command under the lead pursuit strategy is formulated as:

$$\begin{cases} \mathbf{e}_{\text{V}}^{\text{lead}} = \mathbf{V}_{\text{T}} - \mathbf{V}_{\text{A}} \\ \mathbf{V}_{\text{cmd}}^{\text{lead}} = \mathbf{V}_{\text{T}} + \mathbf{C}_b^I(\psi_{\text{T}},\theta_{\text{T}},\varphi_{\text{T}})(K_{\text{VP}}\mathbf{e}_{\text{P}} + K_{\text{VD}}\mathbf{e}_V) \end{cases} \tag{13}$$

Likewise, upon deriving the position and velocity commands, they are transformed into heading angle and altitude commands that can be processed by the flight controller:

$$\begin{cases} \psi_{\text{cmd}}^{\text{lead}} = atan2\left(\frac{y_{\text{cmd}}-y_{\text{A}}}{x_{\text{cmd}}-x_{\text{A}}}\right) \\ h'^{\text{lead}}_{\text{cmd}} = h_{\text{cmd}} + \Delta h \end{cases} \tag{14}$$

2.4.4. Pure Pursuit Guidance Law

Pure pursuit dictates that the aircraft's nose constantly points straight at the current position of the tracked aircraft, acting as a rapid angular recovery mechanism under challenging relative-motion conditions. The guidance heading command ψ_{cmd} is continuously modified to align with the current line-of-sight vector direction.

$$\psi_{\text{cmd}}^{\text{pure}} = atan2\left(\frac{y_{\text{T}}-y_{\text{A}}}{x_{\text{T}}-x_{\text{A}}}\right) \tag{15}$$

Flight energy is maintained by converting the velocity difference into an altitude difference. The spatial position command for pure pursuit is directly assigned to the tracked aircraft's present location, i.e., $\mathbf{Pos}_{\text{D}}^{\text{pure}} = \mathbf{Pos}_{\text{T}}$.

When separated by a large distance, the aircraft can accelerate to close in; as the distance narrows, the velocity difference predominantly takes effect to prevent the tracking aircraft's speed from being excessive, which would cause an overshoot of the tracked aircraft. Accordingly, the velocity command under the pure pursuit strategy is designed as:

$$\begin{cases} \mathbf{Pos}_{\text{cmd}}^{\text{pure}} = \mathbf{Pos}_T, \mathbf{e}_p^{\text{pure}} = \mathbf{Pos}_{\text{cmd}}^{\text{pure}} - \mathbf{Pos}_A \\ \mathbf{e}_V^{\text{pure}} = \mathbf{V}_T - \mathbf{V}_A \\ \mathbf{V}_{\text{cmd}}^{\text{pure}} = \mathbf{V}_T + \mathbf{C}_b^I(\psi_{\text{T}},\theta_{\text{T}},\varphi_{\text{T}})(K_{\text{VP}}\mathbf{e}_p^{\text{pure}} + K_{\text{VD}}\mathbf{e}_V^{\text{pure}}) \end{cases} \tag{16}$$

The design of the altitude command for the pure pursuit strategy also needs to consider the law of conservation of energy, and the altitude command is corrected to $h'^{\text{pure}}_{\text{cmd}} = h_{\text{cmd}} + \Delta h$.

3. Results

3.1. Simulation and Validation of Pursuit Guidance Laws

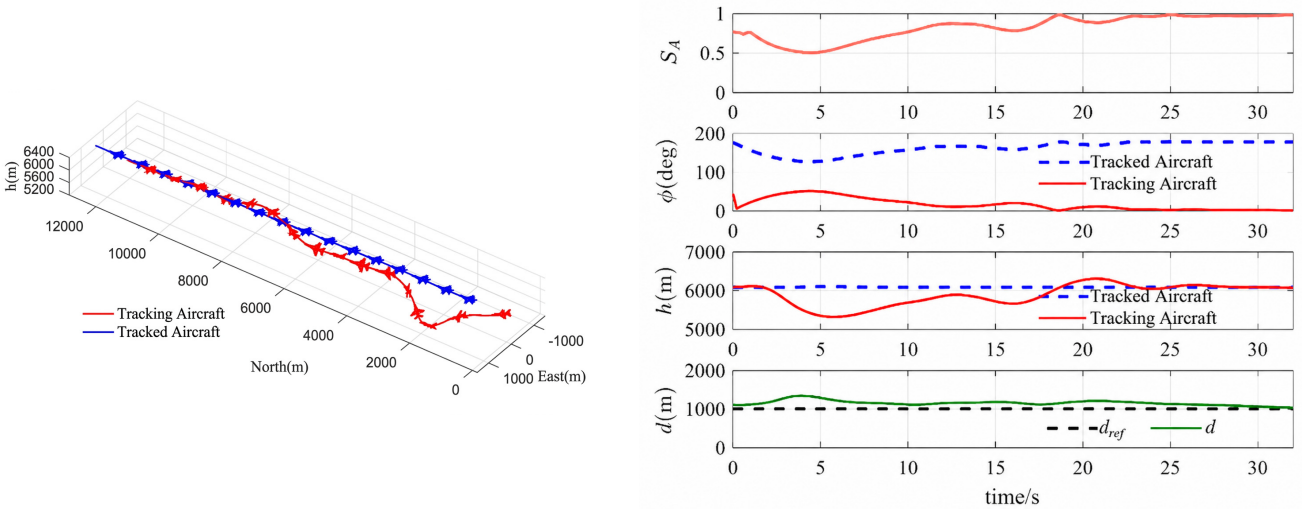
To validate the effectiveness of the proposed pursuit guidance laws, simulation experiments were conducted in which the tracking UAV applied different pursuit strategies while the tracked aircraft executed various maneuvers, with the initial states of both aircraft listed in Table 3.

Table 3. Initial state settings of both aircraft under different pursuit strategies.

Scenario	Tracked Aircraft Maneuver	Initial Velocity V_A, V_T (m/s)	Initial Altitude h_A, h_T(km)	Initial Heading ψ_A, ψ_T Angle (○)
Lag pursuit	Level flight	244, 183	6, 6	45, 0
Lag pursuit (Tracking aircraft inside the turning circle)	Turning	304, 183	6, 6	45, 0
Lag pursuit (Tracking aircraft outside the turning circle)	Turning	304, 183	6, 6	45, 0
Lead pursuit	Turning	244, 183	6, 6	45, 0
Pure pursuit	Lag pursuit	183, 183	6, 6	0, 0

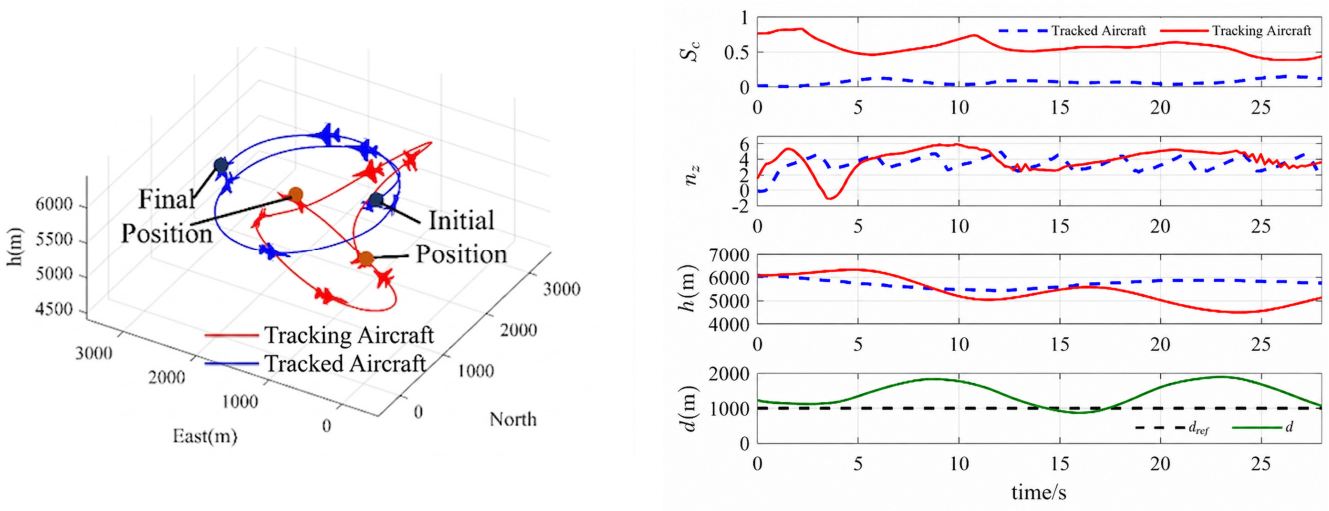
The corresponding simulation results are presented in Figures 5–9.

Figure 5 illustrates that during the tracked aircraft's level flight, the tracking aircraft maintains a stable tracking distance behind it. As indicated by the state curves, the distance converges to the predetermined reference tracking range, while the altitude is dynamically regulated to conserve energy.



(**a**) (**b**)

Figure 5. Maneuver trajectory and states under lag pursuit for a level-flying tracked aircraft. (**a**) A 3D spatial representation of the flight paths during the simulation. (**b**) Time-series plots of critical parameters.

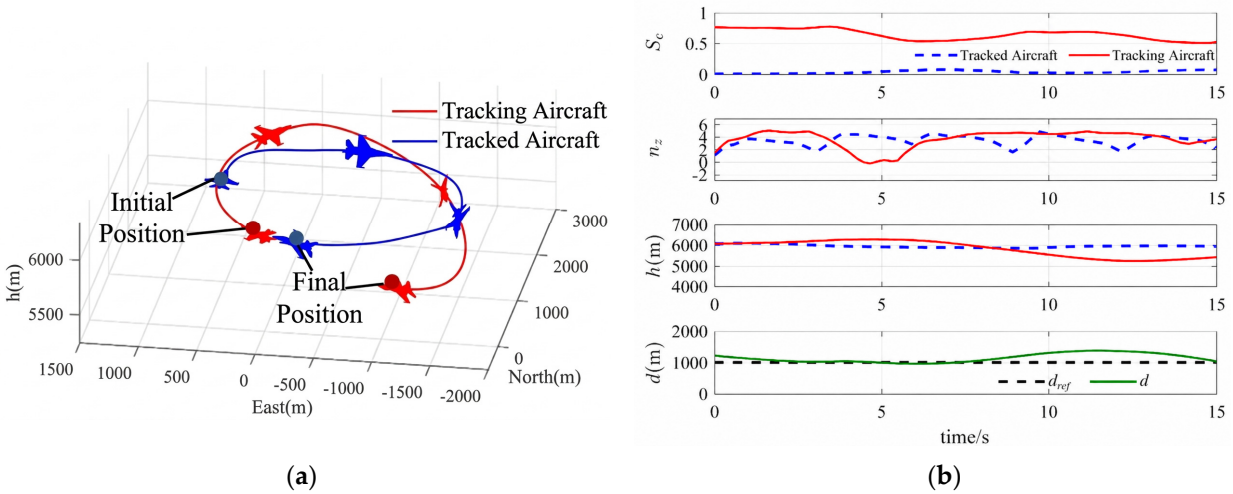


(**a**) (**b**)

Figure 6. Maneuver trajectory and states under lag pursuit with the tracking aircraft initially outside the turning circle. (**a**) A 3D spatial representation of the flight paths during the simulation. (**b**) Time-series plots of critical parameters.

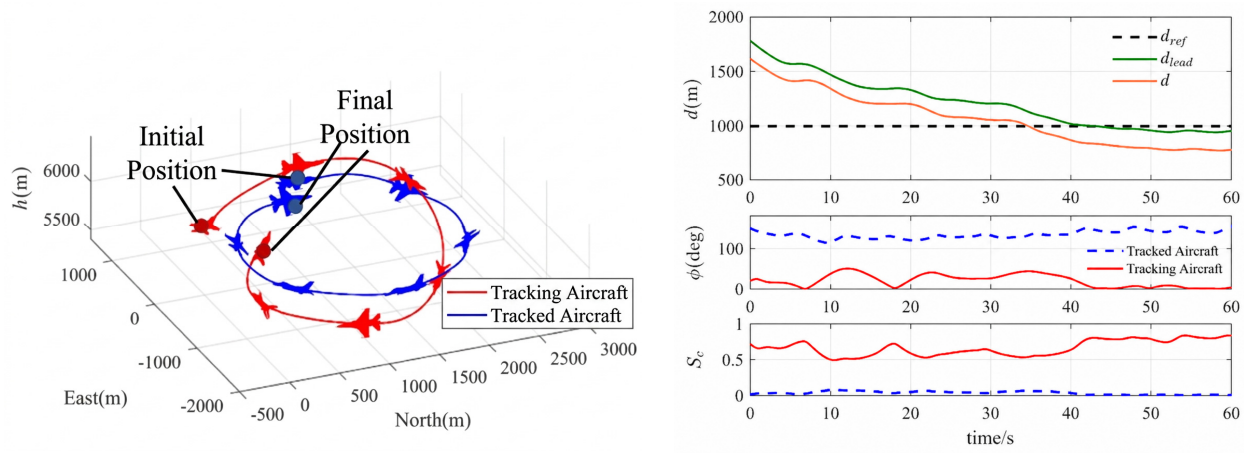
Under turning maneuver conditions, the lag pursuit control law reduces the risk of overshoot by directing the aircraft's heading toward the rear region of the tracked aircraft. Whether initiated from within or outside the tracked aircraft's turn circle, the tracking aircraft converges to a steady tracking state, suggesting the effectiveness of the position and velocity commands derived from the coordinate transformation matrix and energy maneuverability theory.

Lead pursuit is employed when the tracking aircraft approaches a favorable anticipatory tracking geometry and needs to aim ahead of the tracked aircraft. Figure 8 demonstrates that the tracking aircraft can move toward a favorable lead-pursuit geometry by forecasting the tracked aircraft's future location based on the prediction time offset and current maneuvering state.



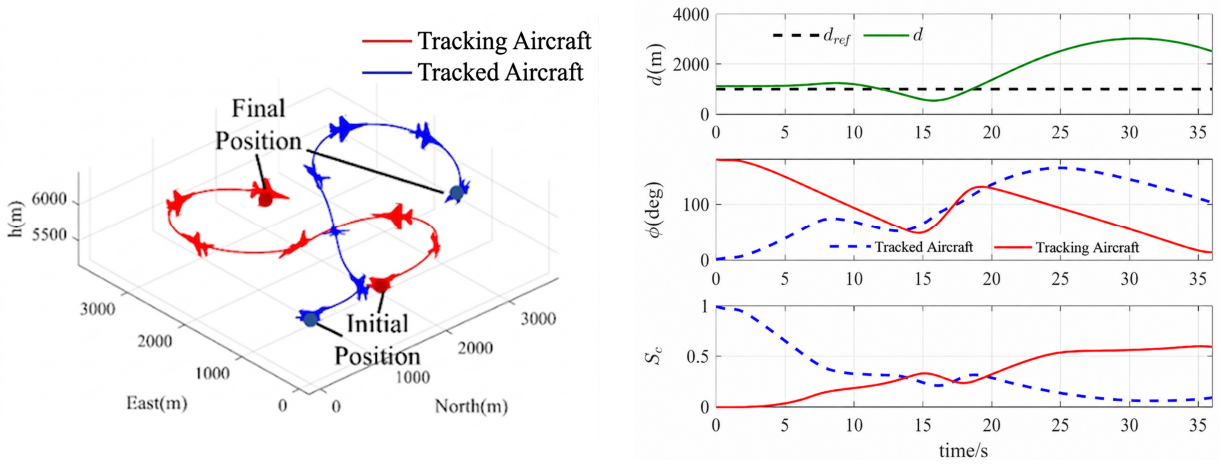
(**a**) (**b**)

Figure 7. Maneuver trajectory and states under lag pursuit with the tracking aircraft initially inside the turning circle. (**a**) A 3D spatial representation of the flight paths during the simulation. (**b**) Time-series plots of critical parameters.



(**a**) (**b**)

Figure 8. Maneuver trajectory and states under lead-pursuit guidance. (**a**) A 3D spatial representation of the flight paths during the simulation. (**b**) Time-series plots of critical parameters.



(**a**) (**b**)

Figure 9. Maneuver trajectory and states under pure-pursuit guidance. (**a**) A 3D spatial representation of the flight paths during the simulation. (**b**) Time-series plots of critical parameters.

When the tracking aircraft is in a challenging relative position, pure pursuit is used for angular recovery. Figure 9 indicates that the tracking aircraft is guided to aim directly at the present position of the tracked aircraft, swiftly diminishing the line-of-sight angle, while its velocity is effectively regulated upon closing in on the tracked aircraft to prevent an early overshoot.

3.2. Convergence Evaluation of the Hierarchical Decision Framework

The proposed hierarchical decision framework adopts a multi-frequency coupling mechanism, in which the interaction frequency between the underlying JSBSim environment and the control network is set to 60 Hz to ensure the continuity of nonlinear aerodynamic calculations and the stability of attitude control, whereas the top-level PPO guidance-mode decision frequency is set to 5 Hz. In this architecture, the 5 Hz decision frequency corresponds to a guidance-mode update period of 0.2 s, while the mid-level guidance laws and low-level flight controller continuously update the executable heading, altitude, velocity, and actuator commands at a higher frequency. This configuration prevents command oscillations triggered by high-frequency guidance-mode switches, allowing the UAV ample time to execute the selected pursuit intent [32]. Since lag pursuit,

lead pursuit, and pure pursuit are guidance-mode intents rather than actuator-level commands, excessively high-frequency switching may lead to unstable intent alternation without improving physical maneuver execution. The representative simulation analyzed in Section 3.4 further shows that the agent can switch from pure pursuit to lag pursuit and then to lead pursuit in response to changes in angular alignment, indicating that the 5 Hz guidance-mode update can support the current simulated close-range tracking settings.

A conventional continuous-control PPO baseline, which directly outputs low-level control commands, is used for the convergence comparison in Section 3.2. To ensure a fair comparison, the proposed high-level guidance-mode PPO and the conventional PPO baseline use the same network architecture and training hyperparameters, as summarized in Table 4. The training process was conducted over a total of 4 million steps across 32 parallel environments. All PPO algorithms were implemented on a Windows operating system using Python 3.8.1, PyTorch 2.1.2, Gym 0.20.0, and JSBSim package version 1.2.3. The hardware platform consisted of an Intel® Core™ i5-13600KF CPU, an NVIDIA GeForce RTX 4070 Super GPU, and 64 GB of RAM. The curve illustrating the variation in cumulative reward per episode is presented in Figure 10.

Table 4. Shared network architecture and training hyperparameters of the high-level guidance-mode PPO and the conventional continuous-control PPO baseline.

Hyperparameter	Value
Number of hidden layers in the critic network	3
Hidden-layer sizes of the critic network	{512, 256, 128}
Optimizer of the critic network	Adam
Learning rate of the critic network	3×10^{-4}
Hidden-layer sizes of the actor network	{256, 256, 128}
Optimizer of the actor network	Adam
Learning rate of the actor network	3×10^{-4}
Number of parallel environments	32
Discount factor γ	0.99
Buffer size	512
Entropy coefficient	1×10^{-3}
GAE parameter λ	0.95

Figure 10 shows that incorporating the guidance-mode generation layer improves both the convergence rate of the training process and the final cumulative reward under the current training setting. This improvement is attributed to the reduction of the high-level exploration space from continuous low-level control commands to three interpretable pursuit-strategy intents, while the mid-level guidance laws preserve continuous trajectory-generation capability. Therefore, the high-level agent performs policy learning in a compact three-action guidance-mode space rather than in a continuous or high-dimensional maneuver-control space.

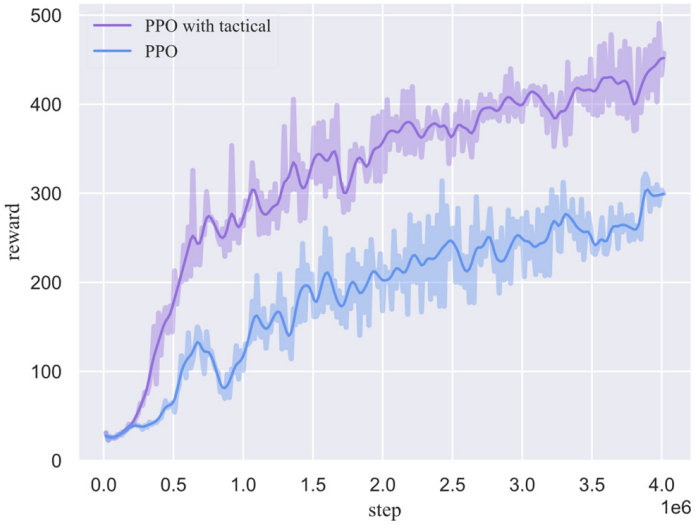


Figure 10. Variation curves of the cumulative reward per episode.

The comparison in this subsection focuses on the benefit of the pursuit-strategy action abstraction over direct low-level continuous control, whereas Section 3.3 further compares different high-level learning algorithms under the same pursuit-strategy action space.

3.3. Comparison Between PPO and Double-DQN for High-Level Guidance-Mode Decision-Making

To further evaluate whether the performance gain originates from the proposed high-level guidance-mode decision mechanism rather than from the low-level flight controller, an additional cross-algorithm comparison was conducted under the same hierarchical framework. In this comparison, the proposed PPO policy was compared with a Double-DQN baseline. Both methods used the same pursuit-strategy action space, namely lag pursuit, lead pursuit, and pure pursuit, and both shared the same pre-trained low-level flight controller. Therefore, the comparison focuses on the learning algorithm used for high-level guidance-mode selection while keeping the executable guidance laws and JSBSim flight-dynamics loop unchanged.

During evaluation, the counterpart aircraft follows a fixed rule-based pursuit-switching policy rather than a learning-based policy. This rule-based counterpart shares the same 16-dimensional high-level observation vector, the same three pursuit-strategy action space, the same mid-level guidance laws, and the same pre-trained low-level PPO flight controller as the learning agent. The only difference is that its high-level guidance mode is selected by hand-coded geometric rules instead of a neural network. Specifically, at each 5 Hz high-level decision step, the rule-based baseline selects lead pursuit when the range satisfies $R \leq d_{\mathrm{ref}}$, the absolute azimuth angle satisfies $|\phi| \leq 15^{\circ}$, and the absolute tracked-aircraft aspect angle satisfies $|q| \leq 60^{\circ}$. It switches to pure pursuit when $|\phi| \geq 75^{\circ}$ or $|q| \geq 120^{\circ}$ and defaults to lag pursuit otherwise. The selected pursuit intent is then converted into heading, altitude, and velocity commands by the same mid-level guidance laws and executed by the same low-level flight controller. Therefore, the rule-based counterpart provides a deterministic and reproducible geometric evaluation reference, while the observed performance difference between PPO and Double-DQN mainly reflects the learned high-level decision policy rather than differences in low-level control capability. To avoid mixing this cross-algorithm setting with the convergence experiment in Section 3.2, the algorithm-specific hyperparameters used for the PPO–Double-DQN comparison are listed separately in Appendix A.2, Table A2.

All evaluation episodes were conducted under the simplified close-range simulation setting and the simulation-score-based geometric termination rule defined in Section 2.3. The rule-based baseline followed the same pursuit-switching policy, and the initial simulation scenarios were sampled from the four representative posture categories used in the Monte Carlo evaluation. Each method was trained for 1 million transitions and evaluated over 200 episodes. In addition to the conventional success, failure, and simultaneous-termination rates, crash-aware metrics were reported. In particular, the non-crash success rate counts only the episodes in which the learning agent achieves success through the predefined geometric tracking condition, thereby excluding passive success cases caused by opponent crashes.

Table 5 summarizes the comparison results. Compared with Double-DQN, the proposed PPO achieves a higher success rate, increasing from 57.0% to 83.5% under the same pursuit-strategy action space and simulation-score-based geometric termination rule. The non-crash success rate increases from 34.5% to 67.0%, indicating that the proposed PPO is more capable of maintaining a favorable geometric tracking condition. Meanwhile, the learning-agent crash rate decreases from 20.5% to 1.0%, suggesting that the policy-gradient-based high-level decision model learns a more stable pursuit-strategy switching policy under the nonlinear JSBSim flight-dynamics constraints.

Table 5. Cross-algorithm comparison under the simulation-score-based geometric termination rule.

Method	Avg. Reward	Success (%)	Non-Crash Success (%)	Failure (%)	Simultaneous Termination (%)	Learning-Agent Crash (%)
Double-DQN	634.98	57.0	34.5	29.0	14.0	20.5
Proposed PPO	827.35	83.5	67.0	8.5	8.0	1.0

These results indicate that, under the same hierarchical pursuit-strategy action abstraction, PPO provides more stable high-level guidance-mode decision-making than the value-based Double-DQN baseline. The improvement is particularly evident in the non-crash success rate and learning-agent crash rate, which provide additional information beyond the raw success rate by separating active geometric-condition success cases from crash-induced outcomes.

3.4. Analysis of Representative Dynamic-Tracking Trajectories

To further illustrate the decision-making mechanism of the agent in simulated close-range tracking scenarios, this section analyzes a representative simulation episode. As observed in Figure 11, starting from an initially neutral state, both aircraft perform turning maneuvers to avoid entering a challenging angular position.

As observed in Figure 11, from an initially balanced configuration, both aircraft perform turning maneuvers to avoid potentially challenging postures. During the first 10–20 s of the simulation, the rule-based aircraft temporarily obtains a slight angular-alignment advantage. In response to this relative-motion state, the high-level network issues a “pure pursuit” command to increase the angular correction rate. After the 20-s mark, the tracked aircraft loses its relative angular-alignment advantage owing to a degraded turn rate, and the network autonomously transitions to “lag pursuit” to reduce the relative distance. At approximately 34 s, the azimuth angle decreases to below 15°, and the policy switches to lead pursuit to approach a favorable anticipatory tracking geometry. Around 44 s, the predefined geometric tracking condition in Section 2.3 is satisfied, leading to a successful simulation-score-based termination of the rule-based counterpart.

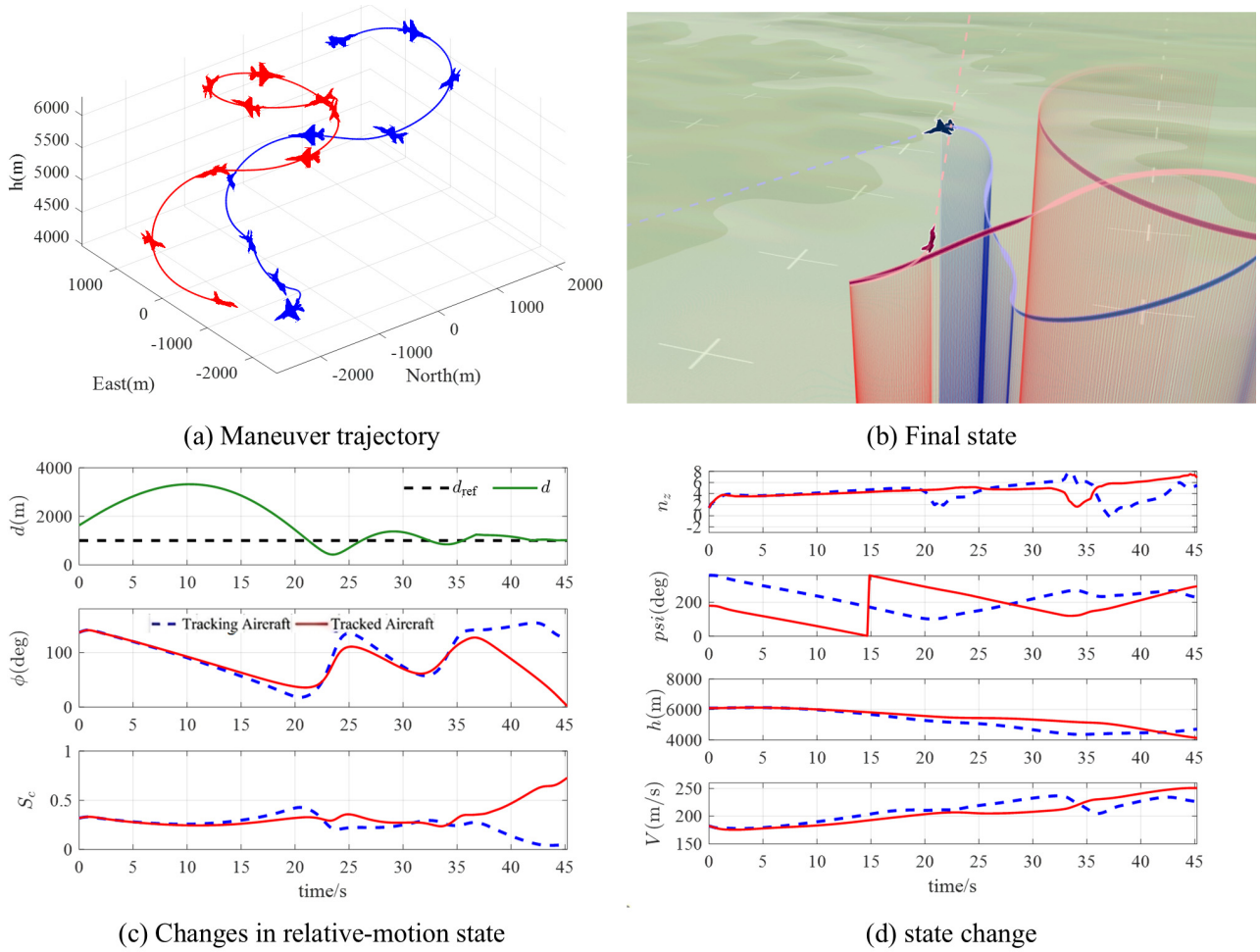


Figure 11. Typical dynamic-tracking maneuver trajectories and state variation curves.

3.5. Results of Monte Carlo Randomized Testing

To reduce the influence of random initial conditions and assess the algorithm's generalization capability under the current simulation assumptions, this study employs the Monte Carlo method to conduct 200 randomized simulation tests across four categories of initial configurations. The duration of each simulation episode is capped at 512 high-level decision steps. An episode is recorded as a success or failure when one aircraft's simulation score reaches zero under the simulation-score-based geometric termination rule, or when one aircraft violates the flight-safety termination conditions while the other aircraft remains within the valid flight envelope. If neither aircraft reaches a terminal condition before the maximum step count is reached, the episode is recorded as a maximum-step termination case. If both aircraft reach a zero simulation score or violate the termination condition within the same simulation step, the result is treated as a simultaneous termination case. The statistical results for the four representative initial configurations are summarized in Figure 12.

The statistical data show that, under the predefined simulation rules, the ideal state-observation assumption, and the JSBSim-based fixed-wing aircraft flight-dynamics model, the hierarchical decision-making method obtains stable simulated performance across the four initial configuration categories. Notably, in scenarios with an initially favorable relative configuration, the success rate reaches 91.5% under the current test settings. An analysis of several simulation trajectories indicates that some simultaneous termination outcomes are associated with both aircraft violating the low-altitude safety condition before the maximum step limit is reached. This phenomenon suggests that, in some

challenging recovery situations, the current reward design may overemphasize altitude-to-speed conversion and angular recovery while insufficiently penalizing unsafe low-altitude states. This limitation does not negate the effectiveness of the proposed hierarchical action-space design, because such unsafe terminations are counted as non-success outcomes in the Monte Carlo statistics; however, it indicates that future versions of the reward function should introduce more explicit safety constraints, such as altitude-margin rewards, ground-collision penalties, and safety-aware termination conditions.

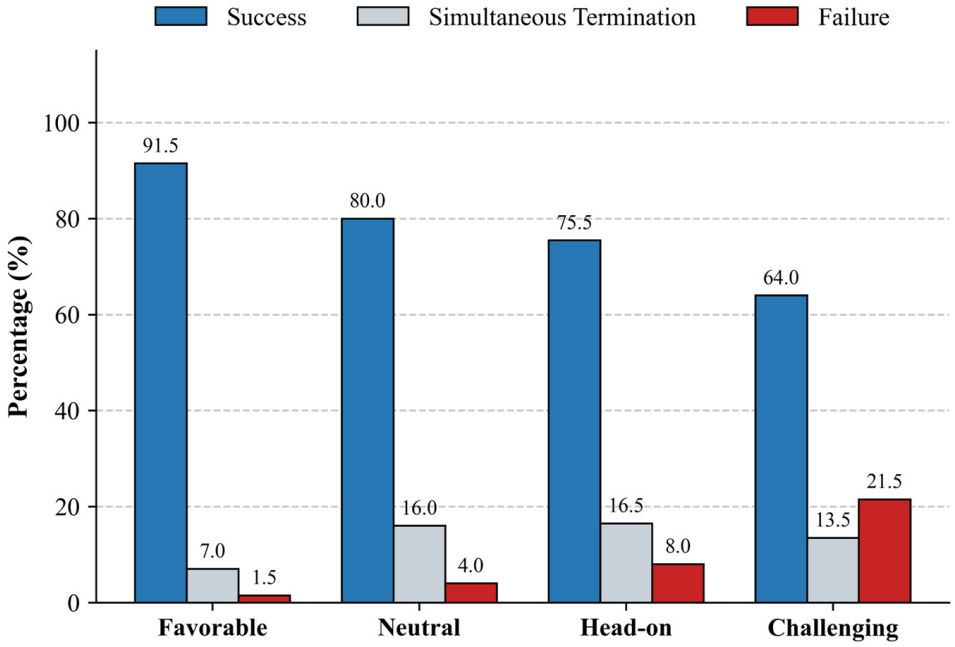


Figure 12. Statistical chart of simulation results for the hierarchical decision model under different initial configurations.

4. Discussion and Conclusions

To address the high exploration burden of continuous action spaces and the rigidity of traditional discrete maneuver primitives in close-range dynamic tracking, this paper proposes a hierarchical reinforcement-learning decision-making method based on a pursuit-strategy action space. By decoupling low-level flight-dynamics control from high-level guidance-mode selection, the proposed framework enables closed-loop intelligent maneuvering decision-making under nonlinear flight-dynamics constraints.

(1) A dimension-reduced action space for close-range dynamic-tracking decision-making is proposed. The complex six-degree-of-freedom close-range relative-motion problem is abstracted into three geometric guidance modalities: lag pursuit, lead pursuit, and pure pursuit. The simulation results indicate that, compared with a baseline that directly outputs low-level continuous control commands, the proposed method reduces the exploration burden and improves both the convergence rate and the achieved cumulative reward under the current training setting.

(2) A high-fidelity hierarchical closed-loop execution mechanism is established. Based on the geometric mechanism of close-range dynamic tracking and energy maneuverability theory, mid-level guidance laws, including velocity compensation and altitude compensation, are designed for the three pursuit strategies, and low-level error tracking is realized using the JSBSim high-fidelity flight-dynamics model. This framework mitigates the latency and limitations inherent in traditional discrete strategy-level maneuvers when tracking highly dynamic aircraft.

(3) Improved simulated tracking performance is observed across representative initial configurations. Monte Carlo randomized simulation tests demonstrate that, under the predefined simulation assumptions and simulation-score-based geometric termination criteria, the proposed decision-making model achieves a success rate of up to 91.5% in initially favorable configurations; even under initially challenging configurations, the agent obtains a 64.0% success rate through adaptive switching among pursuit strategies. These results support the effectiveness of the proposed hierarchical strategy within the current simulation scope, while broader validation under incomplete observations, environmental disturbances, multiple random seeds, high-dynamic maneuvers, and more diverse comparative policies remains necessary.

This study provides a hierarchical decision-making framework for UAV close-range dynamic tracking that balances the exploration efficiency of reinforcement learning with the physical constraints of flight control. It should be noted that the current Monte Carlo evaluation is conducted under predefined simulation assumptions, without explicitly modeling incomplete observations, wind disturbances, or sensor noise. Future work will further relax these assumptions by incorporating incomplete information, sensor noise, wind disturbances, multi-seed statistical evaluation, safety-constrained altitude penalties, ground-collision penalties, higher-order trajectory-prediction models, and additional recovery and energy-management maneuvers. In particular, maneuvers such as coordinated turn, vertical recovery, altitude-exchange maneuver, roll-based recovery, and energy-recovery guidance modes can be introduced into an expanded strategy action space to improve recovery robustness under low-altitude or strongly challenging tracking conditions. The extension of this hierarchical framework to multi-UAV cooperative and interactive dynamic-tracking scenarios will also be investigated.

Author Contributions: Conceptualization, Y.L. (Yu Lai), J.J. and Y.L. (Yuanfei Liu); Methodology, Y.L. (Yu Lai), Y.Y., J.J. and Y.L. (Yuanfei Liu); Software, J.J.; Validation, Y.L. (Yu Lai), Y.C., Y.Y., J.J. and Y.L. (Yuanfei Liu); Formal analysis, Y.L. (Yu Lai), Y.C., Y.Y., J.J. and Y.L. (Yuanfei Liu); Investigation, Y.L. (Yu Lai), Y.Y. and Y.L. (Yuanfei Liu); Resources, Y.C. and Y.Y.; Data curation, Y.L. (Yu Lai), Y.C. and Y.Y.; Writing—original draft, Y.L. (Yu Lai) and J.J.; Writing—review & editing, Y.L. (Yu Lai), Y.C., Y.Y., J.J. and Y.L. (Yuanfei Liu); Visualization, Y.L. (Yu Lai) and J.J.; Supervision, Y.L. (Yu Lai), Y.C., J.J. and Y.L. (Yuanfei Liu); Project administration, Y.C., Y.Y. and Y.L. (Yuanfei Liu); Funding acquisition, Y.C. and Y.L. (Yuanfei Liu). All authors have read and agreed to the published version of the manuscript

Funding: This research received no external funding.

Data Availability Statement: The data that support the findings of this study are available on request from the corresponding author. The data are not publicly available due to privacy restrictions.

DURC Statement: This research is conducted within the fields of aerospace engineering and autonomous control, aiming to enhance the agility, trajectory tracking, and intelligent decision-making capabilities of unmanned aerial vehicles (UAVs) in highly dynamic environments. The algorithms developed focus on interaction-aware decision-making and geometric pursuit strategies, which are intended to support autonomous navigation, trajectory tracking, and collision-avoidance research. The authors confirm that this work is limited to autonomous flight control, trajectory tracking, and simulation-based guidance-mode selection, without addressing real-world deployment or application-specific implementation. We adhere to relevant ethical guidelines and advocate for the responsible and transparent use of artificial intelligence in aviation research.

Conflicts of Interest: The authors declare no conflicts of interest.

Appendix A

Appendix A.1. Low-Level Flight-Controller PPO Agent

Table A1. Network architecture and training hyperparameters of the low-level glight-controller PPO agent.

Hyperparameter	Value
Number of hidden layers in the critic network	3
Hidden-layer sizes of the critic network	{256, 128, 128}
Optimizer of the critic network	Adam
Learning rate of the critic network	$3 \times 10-4$
Hidden-layer sizes of the actor network	{512, 256, 128}
Optimizer of the actor network	Adam
Learning rate of the actor network	$3 \times 10-4$
Number of parallel environments	30
Discount factor γ	0.99
Buffer size	900
Entropy coefficient	$1 \times 10-3$
GAE parameter λ	0.95
Clipping parameter	0.2

Appendix A.2. Hyperparameters for the Cross-Algorithm Comparison

Table A2. Algorithm-specific hyperparameters for the PPO–Double-DQN comparison.

Group	Parameter	Proposed PPO	Double-DQN
Algorithm-specific setting	Implementation	Stable-Baselines3 PPO	Custom PyTorch Double-DQN
	Learning paradigm	On-policy policy gradient	Off-policy value-based learning
	Learning rate	3×10^{-4}	3×10^{-4}
	Discount factor γ	0.99	0.99
	Batch size	256	256
	Network architecture	SB3 MlpPolicy, 64×64	Q-network, 256×256 with ReLU
	Exploration mechanism	Entropy regularization	ε-greedy
	Replay buffer	Not used	200,000 transitions
	Target network	Not used	Updated every 2000 steps
PPO-specific setting	Rollout length n_steps	1024	--
	Effective rollout size	$16 \times 1024 = 16{,}384$	--
	Number of epochs	10	--
	GAE parameter λ	0.95	--
	Clip range	0.3	--
	Entropy coefficient	3×10^{-4}	--
Double-DQN-specific setting	Learning starts	--	10,000 transitions
	Training frequency	--	Every 4 environment steps
	Gradient clipping	--	10.0
	ε schedule	--	1.0 to 0.05
	ε decay length	--	500,000 transitions

Table A2 lists the algorithm-specific hyperparameters used for the cross-algorithm comparison in Section 3.3. These settings are separated from Table 4 because Table 4 corresponds to the convergence comparison between the proposed high-level PPO and the conventional continuous-control PPO baseline, whereas Table A2 corresponds to the PPO–Double-DQN comparison under the same pursuit-strategy action space and simulation-score-based geometric termination rule.

References

1. Yang, Q.; Zhang, J.; Shi, G.; Hu, J.; Wu, Y. Maneuver Decision of UAV in Short-Range Air Combat Based on Deep Reinforcement Learning. *IEEE Access* **2020**, *8*, 363–378. [CrossRef]
2. Zhang, H.; Wei, Y.; Zhou, H.; Huang, C. Maneuver Decision-Making for Autonomous Air Combat Based on FRE-PPO. *Appl. Sci.* **2022**, *12*, 10230. [CrossRef]
3. Li, B.; Huang, J.; Bai, S.; Gan, Z.; Liang, S.; Evgeny, N.; Yao, S. Autonomous Air Combat Decision-Making of UAV Based on Parallel Self-Play Reinforcement Learning. *CAAI Trans. Intell. Technol.* **2023**, *8*, 64–81. [CrossRef]
4. Pope, A.P.; Ide, J.S.; Mićović, D.; Diaz, H.; Twedt, J.C.; Alcedo, K.; Walker, T.T.; Rosenbluth, D.; Ritholtz, L.; Javorsek, D. Hierarchical Reinforcement Learning for Air Combat at DARPA's AlphaDogfight Trials. *IEEE Trans. Artif. Intell.* **2023**, *4*, 1371–1385. [CrossRef]
5. Barto, A.G.; Mahadevan, S. Recent Advances in Hierarchical Reinforcement Learning. *Discret. Event Dyn. Syst.* **2003**, *13*, 41–77. [CrossRef]
6. Hu, W.; Deng, Z.; Yang, Y.; Zhang, P.; Cao, K.; Chu, D.; Zhang, B.; Cao, D. Socially Game-Theoretic Lane-Change for Autonomous Heavy Vehicle Based on Asymmetric Driving Aggressiveness. *IEEE Trans. Veh. Technol.* **2025**, *74*, 17005–17018. [CrossRef]
7. Deng, Z.; Hu, W.; Sun, C.; Chu, D.; Huang, T.; Li, W.; Yu, C.; Pirani, M.; Cao, D.; Khajepour, A. Eliminating Uncertainty of Driver's Social Preferences for Lane Change Decision-Making in Realistic Simulation Environment. *IEEE Trans. Intell. Transp. Syst.* **2024**, *26*, 1583–1597. [CrossRef]
8. Peng, F.; She, S.; Deng, Z. Semantic-Aligned Multimodal Vision–Language Framework for Autonomous Driving Decision-Making. *Machines* **2026**, *14*, 125. [CrossRef]
9. Yuan, W.; Xiwen, Z.; Rong, Z.; Shangqin, T.; Huan, Z.; Wei, D. Research on UCAV Maneuvering Decision Method Based on Heuristic Reinforcement Learning. *Comput. Intell. Neurosci.* **2022**, *2022*, 1477078. [CrossRef]
10. Chai, J.; Chen, W.; Zhu, Y.; Yao, Z.-X.; Zhao, D. A Hierarchical Deep Reinforcement Learning Framework for 6-DOF UCAV Air-to-Air Combat. *IEEE Trans. Syst. Man Cybern. Syst.* **2023**, *53*, 5417–5429. [CrossRef]
11. Tan, M.; Sun, H.; Ding, D.; Zhou, H.; Han, T.; Luo, Y. Hierarchical Online Air Combat Maneuver Decision Making and Control Based on Surrogate-Assisted Differential Evolution Algorithm. *Drones* **2025**, *9*, 106. [CrossRef]
12. Wang, L.; Wang, J.; Liu, H.; Yue, T. Decision-Making Strategies for Close-Range Air Combat Based on Reinforcement Learning with Variable-Scale Actions. *Aerospace* **2023**, *10*, 401. [CrossRef]
13. Zhang, Y.; Yang, Z.; Zhang, B.; Wang, X.; Piao, H.; Zhou, D. Air Combat Joint Strategy Learning Based on a Dual-Loop Framework and Hindsight Experience Replay. *J. Comput. Des. Eng.* **2026**, *13*, 1–22. [CrossRef]
14. Wang, C.; Tu, J.; Yang, X.; Yao, J.; Xue, T.; Ma, J.; Zhang, Y.; Ai, J.; Dong, Y. Explainable Basic-Fighter-Maneuver Decision Support Scheme for Piloting within-Visual-Range Air Combat. *J. Aerosp. Inf. Syst.* **2024**, *21*, 501–514. [CrossRef]
15. Cao, Y.; Kou, Y.-X.; Li, Z.-W.; Xu, A. Autonomous Maneuver Decision of UCAV Air Combat Based on Double Deep Q Network Algorithm and Stochastic Game Theory. *Int. J. Aerosp. Eng.* **2023**, *2023*, 3657814. [CrossRef]
16. Yang, K.; Kim, S.; Lee, Y.; Jang, C.; Kim, Y.-D. Manual-Based Automated Maneuvering Decisions for Air-to-Air Combat. *J. Aerosp. Inf. Syst.* **2024**, *21*, 28–36. [CrossRef]
17. Wang, W.; Ru, L.; Lv, M.; Hou, Y.; Yin, H. Exploring Hierarchical Hybrid Autonomous Maneuvering Decision-Making Architecture in Beyond Visual Range Air Combat. *IEEE Trans. Veh. Technol.* **2025**, *74*, 15491–15506. [CrossRef]
18. Wang, H.; Wang, J. Enhancing Multi-UAV Air Combat Decision Making via Hierarchical Reinforcement Learning. *Sci. Rep.* **2024**, *14*, 4458. [CrossRef]
19. Wang, X.; Wang, Y.; Su, X.; Wang, L.; Lu, C.; Peng, H.; Liu, J. Deep Reinforcement Learning-Based Air Combat Maneuver Decision-Making: Literature Review, Implementation Tutorial and Future Direction. *Artif. Intell. Rev.* **2024**, *57*, 1. [CrossRef]
20. Qian, C.; Zhang, X.; Li, L.; Zhao, M.; Fang, Y. H3E: Learning Air Combat with a Three-Level Hierarchical Framework Embedding Expert Knowledge. *Expert Syst. Appl.* **2024**, *245*, 123084. [CrossRef]
21. Li, Y.; Dong, W.; Zhang, P.; Zhai, H.; Li, G. Hierarchical Reinforcement Learning with Automatic Curriculum Generation for Unmanned Combat Aerial Vehicle Tactical Decision-Making in Autonomous Air Combat. *Drones* **2025**, *9*, 384. [CrossRef]
22. Zheng, Z.; Duan, H. UAV Maneuver Decision-Making via Deep Reinforcement Learning for Short-Range Air Combat. *Intell. Robot.* **2023**, *3*, 76–94. [CrossRef]

23. De Marco, A.; D'Onza, P.M.; Manfredi, S. A Deep Reinforcement Learning Control Approach for High-Performance Aircraft. *Nonlinear Dyn.* **2023**, *111*, 17037–17077. [CrossRef]
24. Li, L.; Zhang, X.; Qian, C.; Wang, R. Basic Flight Maneuver Generation of Fixed-Wing Plane Based on Proximal Policy Optimization. *Neural Comput. Appl.* **2023**, *35*, 10239–10255. [CrossRef]
25. Li, L.; Zhang, X.; Qian, C.; Wang, R.; Zhao, M. Autopilot Controller of Fixed-Wing Planes Based on Curriculum Reinforcement Learning Scheduled by Adaptive Learning Curve. *IEEE Trans. Emerg. Top. Comput. Intell.* **2024**, *8*, 2182–2196. [CrossRef]
26. Yang, J.; Wang, L.; Han, J.; Chen, C.; Yuan, Y.; Yu, Z.L.; Yang, G. An Air Combat Maneuver Decision-Making Approach Using Coupled Reward in Deep Reinforcement Learning. *Complex Intell. Syst.* **2025**, *11*, 364–380. [CrossRef]
27. Shaw, R.L. *Fighter Combat: Tactics and Maneuvering*; Naval Institute Press: Annapolis, MD, USA, 1985; pp. 62–97.
28. Yang, Z.; Sun, Z.; Piao, H.; Huang, J.; Zhou, D.; Ren, Z. Online Hierarchical Recognition Method for Target Tactical Intention in Beyond-Visual-Range Air Combat. *Def. Technol.* **2022**, *18*, 1349–1361. [CrossRef]
29. Zhang, T.; Wang, Y.; Sun, M.; Chen, Z. Air Combat Maneuver Decision Based on Deep Reinforcement Learning with Auxiliary Reward. *Neural Comput. Appl.* **2024**, *36*, 13341–13356. [CrossRef]
30. Zhou, H.; Liu, A.; Li, H. UAV Air Combat Situation Assessment Method Based on Improved Clustering and Self-Learning Network. In Proceedings of the 2023 7th International Conference on Electronic Information Technology and Computer Engineering, Xiamen, China, 20–22 October 2023; ACM: New York, NY, USA, 2023; pp. 1753–1759.
31. Li, H.; Lin, Q.; Han, T.; He, Y. Close-Range Air Combat Model Based on Energy Maneuverability and Its Applications. *Acta Aeronaut. Astronaut. Sin.* **2025**, *46*, 1753–1759. [CrossRef]
32. Chen, R.; Li, H.; Yan, G.; Peng, H.; Zhang, Q. Hierarchical Reinforcement Learning Framework in Geographic Coordination for Air Combat Tactical Pursuit. *Entropy* **2023**, *25*, 1409. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.