Scilight

Advanced Mechatronics
https://www.sciltp.com/journals/am

Article

Hierarchical Safe Reinforcement Learning Control for UAVs via Adaptive Control Barrier Function and Interacting Multiple Model

Jian Gao and Jiankun Sun *

School of Artificial Intelligence and Automation, Huazhong University of Science and Technology, Wuhan 430074, China
* Correspondence: sunjiankun@hust.edu.cn

How To Cite: Gao, J.; Sun, J. Hierarchical Safe Reinforcement Learning Control for UAVs via Adaptive Control Barrier Function and Interacting Multiple Model. *Adv. Mechatron.* **2026**, *1*(1), 3.

Publication History
Received: 30 June 2026
Revised: 25 July 2026
Accepted: 30 July 2026
Published: 3 September 2026

Abstract: This paper considers the safety-critical control problem for UAVs in dynamic environments. To handle this, we propose a novel hierarchical safe reinforcement learning control via adaptive control barrier function (ACBF) and interacting multiple model (IMM). In the proposed hierarchical control method, the upper layer employs a proximal policy optimization (PPO) algorithm to generate reference control inputs that are capable of adapting to uncertain environments. While in the lower safety filter layer, we design a three-dimensional IMM to accurately predict the future finite-horizon trajectories of dynamic obstacles. Integrating these predictions, a novel ACBF with an adaptive coefficient is proposed to modify the reference inputs generated by the upper layer, thereby rigorously ensure the strict safety of UAVs in dynamic obstacle environments. The primary advantage of the proposed hierarchical control method lies in its dual capability to fully guarantee safety and well accommodate the variability of dynamic environments. The superiorities of the proposed control method are verified via simulation results.

Keywords: safe reinforcement learning; control barrier function (CBF); interacting multiple model (IMM); obstacle avoidance; proximal policy optimization (PPO)

1. Introduction

Unmanned aerial vehicles (UAVs) have been widely applied in complex scenarios including disaster rescue and aerial inspection, which imposes stringent requirements for robust and high-precision control strategies [1,2]. Conventional UAV control methodologies are predominantly based on model-driven approache. For example, PID control, a well-established technique celebrated for its structural simplicity, remains the core component of industrial autopilot systems [3,4]. Model predictive control (MPC) has been intensively studied to overcome the limitations of linear control in handling system constraints and optimizing flight trajectories [5]. Meanwhile, nonlinear control strategies such as sliding mode control and backstepping control have been developed to tackle the inherent nonlinear dynamics of quadrotors [6,7]. Nevertheless, the performance of these model-based approaches is highly dependent on the accuracy of physical system parameters. In practical plants including flight scenarios with aerodynamic disturbances and unmodeled dynamics, it is often difficult to establish an accurate mathematical model, which inevitably leads to the degradation of control performance [8–10].

To alleviate the heavy dependence on accurate dynamic models, reinforcement learning (RL) has emerged as a promising data-driven control paradigm [11–13]. By learning optimal policies through online interaction with the environment, RL exhibits superior adaptability in high-dimensional and complex mission scenarios. Notably, Kaufmann et al. recently demonstrated a landmark result, where a deep RL-based drone achieved superior performance over professional human pilots in high-speed autonomous racing [11]. Despite these impressive advances, conventional RL algorithms focus primarily on maximizing cumulative rewards while neglecting critical safety constraints. The absence of formal safety guarantees, together with the high risk of catastrophic failures during exploratory training, significantly restricts the practical deployment of RL in safety-critical UAV applications.

In this context, safe RL has garnered considerable research attention, as it integrates the environmental adaptability of learning-based methods with rigorous formal safety guarantees. State-of-the-art safe RL approaches are generally classified into two categories: soft constraint methods that impose safety penalties on reward functions, and hard constraint methods derived from classical control theory [14,15]. The former typically reformulate constrained Markov decision processes into unconstrained dual forms to strike a trade-off between reward maximization and constraint adherence [16, 17]. However, these methods often suffer from training instability and cannot strictly guarantee safety during the learning process due to the lagging nature of penalty updates. Conversely, hard constraint

methods enforce safety via explicit manifold restriction [18]. Beyond their rigorous certifications, their primary drawbacks lie in strict real-time rigidity and limited scalability, where the demand for high-frequency optimization scales poorly with increasing environmental complexity.

To implement such hard constraints in practical UAV navigation, researchers have traditionally drawn inspiration from geometric and force-field methods such as artificial potential field (APF) algorithms [19] and Voronoi diagram-based techniques [20, 21]. Although these methods can offer basic collision avoidance, they are often plagued by inherent drawbacks such as vulnerability to local minima and high computational complexity, which impairs real-time responsiveness in high-speed flight. To overcome these limitations, control barrier functions (CBFs) have emerged as a prominent formal safety technique [22–24]. Unlike APF, CBFs ensure the forward invariance of a safe set by mapping safety requirements into linear constraints within a real-time quadratic program (QP) [25]. This framework acts as a safety filter that provides rigorous certificates with minimal computational overhead. While CBFs can effectively maintain system safety, they typically require a performance-oriented nominal controller to guide the system toward its desired goal. Consequently, the integration of CBFs with RL has demonstrated a potent synergistic paradigm. In this framework, CBFs serve as rigorous safety filters that project the actions generated by an RL policy onto a predefined safe set in real time. This approach ensures strict safety guarantees throughout the control process, while simultaneously leveraging the adaptive and goal-oriented capabilities inherent in learning-based methods [26–28].

To handle environmental uncertainties that may compromise the safety performance of the system, safety-critical control methods through ACBF have been proposed in [29,30]. However, their practical implementation for UAVs operating in complex dynamic scenarios remains fraught with challenges due to the inability to perform accurate prediction. The construction of valid CBFs for dynamic obstacles demands accurate prediction of their future motion trajectories [31]. The IMM algorithm [32–34] is a well-established method renowned for its robust performance in maneuvering target tracking. Nevertheless, the development of a unified framework that can concurrently address dynamic environmental constraints and system parametric uncertainties remains an open research problem.

This paper addresses the safety-critical control problem for UAVs operating in dynamic environments. To tackle this problem, we propose a novel hierarchical safe reinforcement learning control framework based on the ACBF and IMM. In the proposed hierarchical control method, the upper layer employs a PPO algorithm to generate reference control inputs that can adapt to uncertain environments. In the lower safety filter layer, by leveraging the IMM algorithm to achieve high-precision state estimation and trajectory prediction of dynamic obstacles, the obtained predictive information can be explicitly embedded into the safety constraint formulation of ACBFs, thereby enabling proactive and real-time avoidance of moving objects. The main contributions of this work are summarized as follows:

(1) A hierarchical safe reinforcement learning framework is proposed, which integrates a PPO-based upper layer with a novel lower safety filter layer built upon the ACBF. This design synergizes the strong environmental adaptability of reinforcement learning with the rigorous safety guarantees of ACBF, where an adaptive coefficient dynamically adjusts control inputs to ensure strict safety in complex scenarios.

(2) Different from separated prediction or instantaneous CBF design in prior studies [22–24], we deduce a unified mapping formula converting 3D IMM predicted multi-step obstacle trajectories into HOCBF inequality constraints, and supplement prediction delay compensation terms for adaptive barrier coefficients. The integrated predictive safety module realizes proactive pre-collision evasion instead of passive emergency correction, and the theoretical derivation of prediction-safety coupling is original in UAV safe control research. No single IMM or CBF algorithm can achieve this forward-looking hard safety guarantee.

(3) Compared with conventional end-to-end RL obstacle-avoidance methods [35, 36], the proposed hierarchical safe reinforcement learning control framework substantially reduces training time and sample complexity by offloading safety-critical constraints to the ACBF module, while maintaining superior safety and task performance.

2. Preliminaries

The continuous-time dynamics of the quadrotor are modeled as follows:

$$\begin{cases} \dot{p}(t) = v(t) \\ \dot{v}(t) = -gE + \dfrac{1}{m} R E f_u(t) \end{cases} \tag{1}$$

where $p(t) = [p_x(t), p_y(t), p_z(t)] \in \mathbb{R}^3$ and $v(t) = [v_x(t), v_y(t), v_z(t)] \in \mathbb{R}^3$ denote the position and velocity vectors in the inertial frame, respectively. $f_u(t)$ represents the total propulsive force generated by the four rotors expressed in the body frame. $E = [0, 0, 1]^T$ is the unit vector along the vertical axis, g is the gravitational acceleration, and m is the mass of the quadrotor UAV. R denotes the rotation matrix from the body frame to the inertial frame. As illustrated in Figure 1, the system has six degrees of freedom to be controlled.

By applying the forward Euler method with a sampling period Δt, the discretized version of system (1) at time step k is expressed as

$$\begin{cases} p(k+1) = p(k) + \Delta t v(k) \\ v(k+1) = v(k) - \Delta t g E + \dfrac{\Delta t}{m} R E f_u(k). \end{cases} \tag{2}$$

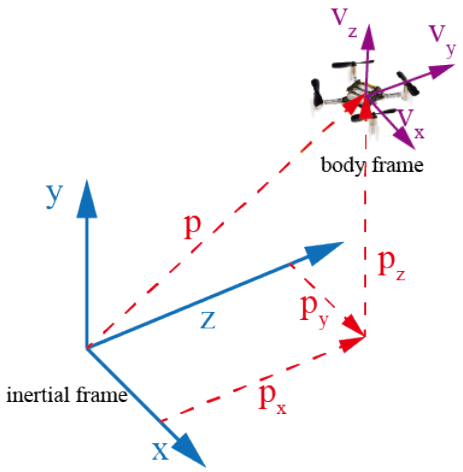

Figure 1. The schematic diagram of position and velocity vectors in the inertial frame.

Let $x(k) = [p(k)^T, v(k)^T]^T \in \mathbb{R}^6$. The discrete-time dynamics derived from (2) can be compactly expressed as

$$x(k+1) = \Phi x(k) + \Gamma u(k) + \mathbf{g}_{vec} \tag{3}$$

where $u(k) = \frac{1}{m} R E f_u(k) \in \mathbb{R}^3$ is the control input, $\mathbf{g}_{vec} = [0, 0, 0, -\Delta t g E^T]^T$ represents the gravitational term, and the system matrices are defined as:

$$\Phi = \begin{bmatrix} I_3 & \Delta t I_3 \\ 0_3 & I_3 \end{bmatrix}, \quad \Gamma = \begin{bmatrix} 0_3 \\ \Delta t I_3 \end{bmatrix} \tag{4}$$

where I_3 and 0_3 denote the identity matrix and zero matrix with 3×3 dimension, respectively.

To design the safety-critical position controller for the UAV, we introduce high-order discrete-time control barrier functions. This framework systematically constructs safety constraints for systems where the relative degree $r > 1$.

Definition 1 (Relative Degree [37]). *A scalar function $h(x(k)) : \mathbb{R}^n \to \mathbb{R}$ is said to have relative degree r for discrete-time system (3) if the control input $u(k)$ explicitly appears for the first time in the r-step ahead prediction $h(x(k+r))$.*

Next, we specifically design the high-order discrete-time CBF. For the discrete-time system (3), a sequence of functions $\psi_i(x(k))$ is defined based on the given function $h(x(k))$ of relative degree r:

$$\begin{aligned} \psi_0(x(k)) &:= h(x(k)) \\ \psi_i(x(k)) &:= \Delta\psi_{i-1}(x(k)) + \gamma_i \psi_{i-1}(x(k)), \\ &\quad i \in \{1, \dots, r-1\} \\ \psi_r(x(k), u(k)) &:= \Delta\psi_{r-1}(x(k), u(k)) + \gamma_r \psi_{r-1}(x(k)) \end{aligned} \tag{5}$$

where $\Delta\psi_{i-1}(x(k)) = \psi_{i-1}(x(k+1)) - \psi_{i-1}(x(k))$ denotes the one-step difference, and $\gamma_i \in (0, 1]$ are parameters associated with the linear class $\mathcal{K}$ functions.

Definition 2 (High-Order Discrete-Time CBF [37]). *For the discrete-time system, a continuous function $h : \mathbb{R}^n \to \mathbb{R}$ is a high-order discrete-time CBF of relative degree r if there exist auxiliary functions $\psi_i(x(k))$, $i \in \{0, 1, \dots, r\}$ defined by (5) and the sequence of sets $\mathcal{C}_1, \dots, \mathcal{C}_r$ is defined as:*

$$\mathcal{C}_i = \{x(k) \in \chi | \psi_{i-1}(x(k)) \geq 0\}, \quad i \in \{1, \dots, r\} \tag{6}$$

such that for all $x(k) \in \bigcap_{i=1}^r \mathcal{C}_i$, there exists a control input $u(k) \in \mathcal{U}$ satisfying:

$$\psi_r(x(k), u(k)) \geq 0. \tag{7}$$

Lemma 1 ([37]). *Given a sequence of sets $\mathcal{C}_i$, $i \in \{1, \dots, r\}$, defined by (6), and a continuous function $h : \mathbb{R}^n \to \mathbb{R}$. If h is a high-order discrete-time CBF of relative degree r defined on $\bigcap_{i=1}^r \mathcal{C}_i$, then any discrete-time controller $u(k)$ that ensures condition (7) renders the set $\bigcap_{i=1}^r \mathcal{C}_i$ forward invariant.*

3. Hierarchical Learning-Based Safe Control Design via ACBF

In this section, a novel hierarchical learning-based safe control method is proposed to achieve position control while guaranteeing obstacle avoidance of UAVs in a dynamic environment.

In the proposed control method as shown in Figure 2, the upper layer employs reinforcement learning to generate reference control inputs that adapt to uncertain environments, while the lower safety filter layer designs a high-order discrete-time ACBF with IMM to modify these reference inputs, thereby rigorously ensuring system safety in the presence of dynamic obstacles.

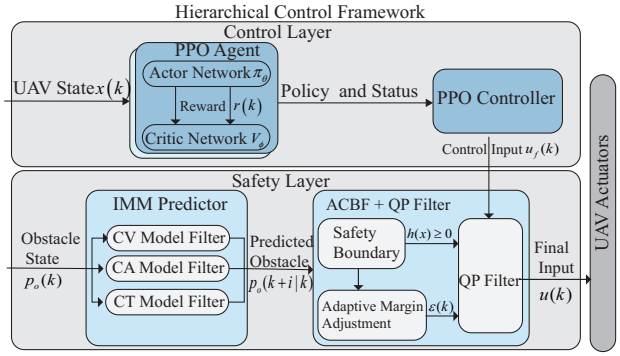

Figure 2. The block diagram of the proposed hierarchical safe reinforcement learning control method.

3.1. Reinforcement Learning-Based Control

To enhance the adaptability of UAVs in uncertain environments, we employ PPO, a state-of-the-art reinforcement learning algorithm, to learn optimal control policies for UAVs. The training environment is formulated as a Markov decision process (MDP) with a designed tuple $(\mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \gamma)$ to capture the full dynamics of the UAV. The agent learns the optimal policy π_θ to maximize the cumulative expected return through interaction with the environment.

Specifically, the state vector $s(k) \in \mathcal{S}$ at time step k is defined as $s(k) = [v(k)^T, e(k)^T]^T$, where $e(k) = p^*(k) - p(k)$ represents the relative position error vector from the UAV to the target position $p^*(k)$. This state vector corresponds to the observation returned by the environment. The action vector

$a(k) \in \mathcal{A} \subseteq \mathbb{R}^3$ is normalized and mapped to the reference control input $u_f(k)$ used in dynamics (3) as follows:

$$u_f(k) = k_{ma} \cdot a(k) + gE \tag{8}$$

where k_{ma} is the maximum maneuvering acceleration coefficient, and gE compensates for the influence of gravity.

To guide the agent effectively, we design a multi-objective composite reward function at time step k as

$$r(k) = r_{tr}(k) + r_{ds}(k) + r_{bp}(k) + r_{tp}(k). \tag{9}$$

The components in (9) are rigorously defined as follows:

- $r_{tr}(k)$ is the target-reaching reward, defined as

$$r_{tr}(k) = \begin{cases} M_{tr}, & \|e(k)\| < \epsilon_{tr} \\ 0, & \text{otherwise} \end{cases} \tag{10}$$

  where $M_{tr} > 0$ is the target coefficient and $\epsilon_{tr} > 0$ is the acceptance radius. This sparse reward is granted when the UAV enters the acceptance radius.
- $r_{ds}(k)$ denotes a distance-shaping reward that promotes goal-directed velocity. A potential-based dense reward is designed as

$$r_{ds}(k) = M_{ds} \cdot (\|e(k-1)\| - \|e(k)\|) \tag{11}$$

  where $M_{ds} > 0$ scales the improvement in distance from the previous step to the current step. For the initial step ($k = 0$), $\|e(-1)\|$ is set to $\|e(0)\|$, resulting in $r_{ds}(0) = 0$.
- $r_{bp}(k)$ represents the boundary penalty term, defined as

$$r_{bp}(k) = \begin{cases} -M_{bp}, & p(k) \notin \mathcal{B} \\ 0, & \text{otherwise} \end{cases} \tag{12}$$

  where M_{bp} is the boundary penalty coefficient and $\mathcal{B}$ is the flight arena.
- $r_{tp}(k)$ is the time penalty term, designed as

$$r_{tp}(k) = -M_{tp} \cdot \Delta t \tag{13}$$

  where $M_{tp} > 0$ penalizes the agent for each elapsed time step, which encourages time-optimal trajectories.

To solve the formulated MDP and maximize the cumulative reward defined in (9), we employ the PPO algorithm based on the actor-critic architecture. Specifically, the actor network $\pi_\theta(a(k)|s(k))$ serves as the controller, mapping the state $s(k)$ to the optimal action distribution, while the critic network $V_\phi(s(k))$ evaluates the value of the state by estimating the value function. PPO stabilizes the iterative training process by utilizing a clipped surrogate objective that effectively limits the magnitude of policy updates to prevent performance collapse. The networks are updated iteratively to maximize a composite objective that balances policy improvement, value estimation accuracy, and an entropy bonus for exploration, ensuring the UAV learns an efficient navigation strategy that converges to the target.

While the optimized control inputs trained by the PPO algorithm demonstrate strong adaptability in uncertain environments, they cannot strictly guarantee obstacle avoidance. To address this limitation, we propose a safety-critical control method that combines ACBF with an improved IMM algorithm. This approach constrains the reference control inputs trained by the PPO algorithm within a designed safe set. The proposed safety-critical control framework employs a novel IMM method to predict the future trajectories of dynamic obstacles and formulates an adaptive safety constraint using ACBF. By integrating these predicted trajectory estimates with the adaptive safety constraints, an optimal control problem is established to achieve strict obstacle avoidance in cluttered environments.

3.2. IMM-Based Obstacle State Estimation

To ensure obstacle avoidance in dynamic environments, it is desirable to predict the future states of obstacles. We design an IMM method to track and predict the states of dynamic obstacles. Let $p_o(k)$ denote the obstacle's position vector. To capture different maneuvering behaviors of obstacles, we define a model set $\mathcal{M} = \{1, 2, 3\}$, where $1, 2, 3$ stand for the constant velocity (CV), constant acceleration (CA), and constant turn rate (CT) models, respectively.

3.2.1. Kinematic Models

The discrete-time state transition process for the j-th model ($j \in \mathcal{M}$) is governed by

$$\begin{aligned} \xi_j(k+1) &= F_j \xi_j(k) + w_j(k) \\ y_j(k) &= C_j \xi_j(k) + q_j(k) \end{aligned} \tag{14}$$

where F_j is the state transition matrix, $w_j(k) \sim \mathcal{N}(0, Q_j)$ is the process noise, and $y_j(k)$ is the measurement output.

In the IMM framework, the models in (14) exhibit distinct mathematical forms while producing the same position measurement output $y_j(k) = p_o(k) = [p_{o,x}(k), p_{o,y}(k), p_{o,z}(k)]^T \in \mathbb{R}^3$ for $j = 1, 2, 3$. Their respective expressions are specified as follows.

- CV Model: The state vector ξ_1 is defined as $\xi_1 = [p_o^T, v_o^T]^T \in \mathbb{R}^6$, where $v_o = [v_{o,x}, v_{o,y}, v_{o,z}]^T \in \mathbb{R}^3$ denotes the velocity vector of the obstacle in the inertial frame. The transition matrix is given as

$$F_1 = \begin{bmatrix} I_3 & \Delta t I_3 \\ 0_3 & I_3 \end{bmatrix}. \tag{15}$$

- CA Model: The state vector ξ_2 is designed as $\xi_2 = [p_o^T, v_o^T, a_o^T]^T \in \mathbb{R}^9$, where $a_o = [a_{o,x}, a_{o,y}, a_{o,z}]^T \in \mathbb{R}^3$ is the acceleration of the obstacle. The transition matrix is designed as follows:

$$F_2 = \begin{bmatrix} I_3 & \Delta t I_3 & \frac{\Delta t^2}{2} I_3 \\ 0_3 & I_3 & \Delta t I_3 \\ 0_3 & 0_3 & I_3 \end{bmatrix}. \quad (16)$$

- CT Model: This model captures coordinated turns in the horizontal x-y plane with a turn rate $\Omega \in \mathbb{R}$, coupled with linear motion along the z-axis. The state vector is defined as $\xi_3 = [p_{o,x}, v_{o,x}, p_{o,y}, v_{o,y}, p_{o,z}, v_{o,z}, \Omega]^T \in \mathbb{R}^7$. The transition matrix F_3 depends on the turn rate and is defined as

$$F_3(\Omega) = \begin{bmatrix} 1 & 0 & 0 & \frac{\sin(\Omega\Delta t)}{\Omega} & -\frac{1-\cos(\Omega\Delta t)}{\Omega} & 0 & 0 \\ 0 & 1 & 0 & \frac{1-\cos(\Omega\Delta t)}{\Omega} & \frac{\sin(\Omega\Delta t)}{\Omega} & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & \Delta t & 0 \\ 0 & 0 & 0 & \cos(\Omega\Delta t) & -\sin(\Omega\Delta t) & 0 & 0 \\ 0 & 0 & 0 & \sin(\Omega\Delta t) & \cos(\Omega\Delta t) & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}. \quad (17)$$

Note that the acceleration states are not explicitly tracked in this kinematic primitive. During the interaction step of the IMM, the augmentation operator $\mathcal{T}_{CT}$ maps this 6-dimensional estimate to the common 9-dimensional space by appending zero-acceleration components or estimating centripetal acceleration.

3.2.2. IMM Filter Cycle

Based on the models in (14), we employ the IMM algorithm to obtain predictions of the obstacle position over a finite horizon $N > 0$. The IMM algorithm recursively updates the mode probabilities $\mu^{(j)}(k)$ and the state estimates through three steps: Interaction, Filtering, and Fusion.

The IMM algorithm operates recursively through three sequential steps: interaction, model-matched filtering, and probability update & fusion [32]. The overall architecture of the proposed estimation and prediction pipeline is visually depicted in Figure 3.

(1) Interaction (Mixing): First, the algorithm computes the mixed initial conditions for each filter. To address the dimension mismatch between heterogeneous models (e.g., CV's $\mathbb{R}^6$ vs. CA's $\mathbb{R}^9$), we employ a projection operator $\mathcal{T}_i$ to map all local estimates to a common augmented state space (here, $\mathbb{R}^9$). The mixed initial state $\bar{\xi}_{k-1}^{(0j)}$ for filter j is computed as:

$$\bar{\xi}_{k-1}^{(0j)} = \sum_{i \in \mathcal{M}} \mu_{i|j}(k-1|k-1) \mathcal{T}_i(\hat{\xi}_{k-1}^{(i)}) \quad (18)$$

where $\mu_{i|j}$ is the mixing probability derived from the Markov transition matrix $[\pi_{ij}]$ and the mode probabilities $\mu^{(i)}(k-1)$ from the previous step.

(2) Model-Matched Filtering: Next, each filter j runs independently using the mixed initial state $\bar{\xi}_{k-1}^{(0j)}$ and the current measurement to generate the updated state estimate $\hat{\xi}_k^{(j)}$ and the model likelihood $\Lambda^{(j)}(k)$.

(3) Probability Update and Fusion: Finally, the mode probabilities $\mu^{(j)}(k)$ are updated via Bayes' rule based on the likelihoods. The global estimated obstacle position $\hat{p}_o(k)$ is obtained by fusing the augmented states from all filters:

$$\hat{p}_o(k) = \sum_{j \in \mathcal{M}} \mu^{(j)}(k) \cdot \text{Proj}_p(\hat{\xi}_k^{(j)}) \quad (19)$$

where $\text{Proj}_p(\cdot)$ extracts the position components from the augmented state. This fused estimate and its propagated future trajectory are directly fed into the ACBF module to construct the safety barrier.

To construct the safety barrier for the finite horizon N, the ACBF framework requires the predicted obstacle trajectory sequence $\bar{p}_o(k) = [p_o(k|k), p_o(k+1|k), \ldots, p_o(k+N|k)]$, where $p_o(k+i|k)$ denote the i-step ahead prediction of $p_o(k+i)$ based on $p_o(k)$. When $i = 0$, we have $p_o(k|k) = p_o(k)$.

Based on the updated state estimate $\hat{\xi}_k^{(j)}$ and mode probability $\mu^{(j)}(k)$ obtained from the IMM filter cycle, the future position at prediction step i is computed by propagating each model's dynamics independently and then fusing them:

$$\begin{cases} \hat{\xi}_{k+i|k}^{(j)} = (F_j)^i \hat{\xi}_k^{(j)}, \quad \forall i \in \{0, \ldots, N\}, \forall j \in \mathcal{M} \\ p_o(k+i|k) = \sum_{j \in \mathcal{M}} \mu^{(j)}(k) \cdot \text{Proj}_p\left(\hat{\xi}_{k+i|k}^{(j)}\right). \end{cases} \quad (20)$$

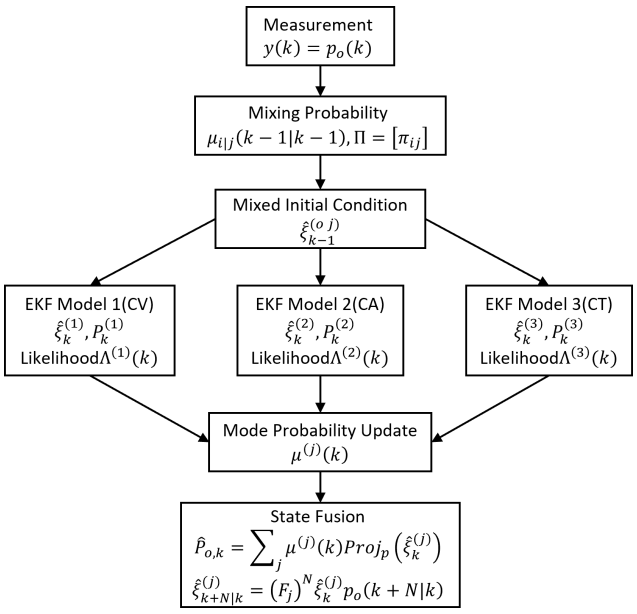

Figure 3. The block diagram of the proposed IMM prediction framework.

Remark 1. *In the practical implementation of the IMM filter, several key parameters are configured to ensure robust state estimation. The process and measurement noise covariance matrices are initialized as* $Q_j = 0.05I$ *and* $R = 0.05I$*, respectively. The initial mode probabilities are set uniformly to* $\mu^{(j)}(0) = 1/3$*. To reflect the persistence of the obstacle's motion modes, the Markov transition probability matrix is defined with diagonal elements* $\pi_{jj} = 0.6$ *and off-diagonal elements* $\pi_{ij} = 0.2$ $(i \neq j)$*. Furthermore, a sliding window of* 20 *time steps is utilized to heuristically estimate the initial turn rate for the CT model, and a moving average over the last* 2 *steps is applied to smooth the model likelihoods, thereby preventing*

excessive mode switching. Finally, the prediction horizon is set to $N = 5$ steps to balance computational efficiency and predictive foresight.

3.3. ACBF-Based Safe Filter

To acquire the position and velocity predictions of the UAV, we define $\bar{p}(k) = [p(k|k), p(k+1|k), \ldots, p(k+N|k)]$ and $\bar{v}(k) = [v(k|k), v(k+1|k), \ldots, v(k+N|k)]$, where $p(k) = p(k|k)$, $v(k) = v(k|k)$, and $p(k+i|k)$, $v(k+i|k)$ denote the i-step predictions at the current time k. Based on the system dynamics (3) and assuming the control input $u(k)$ is constant over the prediction horizon, we compute the predictions $\bar{p}(k)$ and $\bar{v}(k)$ as follows:

$$\begin{cases} p(k+1|k) = p(k) + v(k)\Delta t \\ v(k+1|k) = v(k) + (u(k) - gE)\Delta t \\ p(k+2|k) = p(k+1|k) + v(k+1|k)\Delta t \\ \vdots \\ v(k+N-1|k) = v(k+N-2|k) + (u(k) - gE)\Delta t \\ p(k+N|k) = p(k+N-1|k) + v(k+N-1|k)\Delta t. \end{cases} \tag{21}$$

Combining the position predictions of the UAV and obstacle, we define the safety function $h(\bar{p}(k), \bar{p}_o(k))$ as

$$\begin{aligned} h(\bar{p}(k), \bar{p}_o(k)) &= \begin{bmatrix} h_0(p(k|k), p_o(k|k)) \\ \vdots \\ h_N(p(k+N|k), p_o(k+N|k)) \end{bmatrix} \\ &= \begin{bmatrix} \frac{1}{2}\|p(k|k) - p_o(k|k)\|^2 - d_s^2 \\ \vdots \\ \frac{1}{2}\|p(k+N|k) - p_o(k+N|k)\|^2 - d_s^2 \end{bmatrix} \end{aligned} \tag{22}$$

where $d_s = r_d + r_o + s_d$ represents the minimum safety distance, comprising the radius of the UAV r_d, the radius of obstacles r_o and a cushion of protection s_d.

Remark 2. *Unlike traditional CBFs that only consider the safety constraints at the current time instant [25,29], the designed CBFs (22) incorporate safety considerations over a finite future time horizon N via IMM and prediction technique, resulting in more flexible safety constraints for highly dynamic obstacles.*

Based on the system dynamics (3), it can be observed that the defined CBF h in (22) is of second order. We define

$$\psi_1(\bar{p}(k), \bar{p}_o(k)) = \Delta h(\bar{p}(k), \bar{p}_o(k)) + \alpha_1 h(\bar{p}(k), \bar{p}_o(k)), \tag{23}$$

where $\alpha_1 \in (0, 1]$ and $\Delta h(\bar{p}(k), \bar{p}_o(k)) = h(\bar{p}(k+1), \bar{p}_o(k+1)) - h(\bar{p}(k), \bar{p}_o(k))$.

To ensure forward invariance of the safe set, the control input $u(k)$ must satisfy the second-order discrete-time CBF constraint as:

$$\Delta\psi_1(\bar{p}(k), \bar{p}_o(k)) \geq -\alpha_2\varepsilon(k)\psi_1(\bar{p}(k), \bar{p}_o(k)) \tag{24}$$

where $\Delta\psi_1(\bar{p}(k), \bar{p}_o(k)) = \psi_1(\bar{p}(k+1), \bar{p}_o(k+1)) - \psi_1(\bar{p}(k), \bar{p}_o(k))$, and $\alpha_2\varepsilon(k) \in (0, 1]$ denotes the decay rate for the second-order CBF constraint.

In constraint (24), the time-varying term $\varepsilon(k)$ represents an adaptive coefficient that modulates the strictness of the constraint. To regulate the convergence of $\varepsilon(k)$ towards a desired reference $\varepsilon^*(k)$, we construct a control Lyapunov function (CLF) candidate V_ε as

$$V_\varepsilon(\varepsilon(k)) = (\varepsilon(k) - \varepsilon^*(k))^2. \tag{25}$$

The evolution of $\varepsilon(k)$ is subject to a specific decay condition to ensure its exponential convergence to $\varepsilon^*(k)$. To achieve this, we impose the following constraint on $\varepsilon(k)$:

$$\Delta V_\varepsilon(\varepsilon(k)) + \eta V_\varepsilon(\varepsilon(k)) \leq \delta \tag{26}$$

where $0 < \eta < 1$ and $\delta \geq 0$ is a relaxation factor to ensure feasibility. Here, $\Delta V_\varepsilon(\varepsilon(k)) = V_\varepsilon(\varepsilon(k+1)) - V_\varepsilon(\varepsilon(k))$.

The desired reference penalty $\varepsilon^*(k)$ adapts based on the relative kinematics to tighten safety margins during dangerous encounters. We define the relative collision angle θ_r via the cosine similarity:

$$\cos\theta_r = \frac{p_r \cdot v_r}{\|p_r\|\|v_r\|} \tag{27}$$

where $p_r = p_o(k) - p(k)$ is the relative position vector and $v_r = v_o(k) - v(k)$ is the relative velocity vector.

As shown in Figure 4, a positive $\cos\theta_r$ indicates the drone and obstacle are converging. The desired adaptive reference $\varepsilon^*(k)$ is designed as

$$\varepsilon^*(k) = \varepsilon_{\max}\frac{\|v_r\|\max(0, \cos\theta_r)}{\|p_r\|} \tag{28}$$

where $\varepsilon_{\max}$ is a parameter to be regulated. This formulation ensures that the penalty upper bound decreases (safety tightens) as the relative velocity increases or the distance decreases.

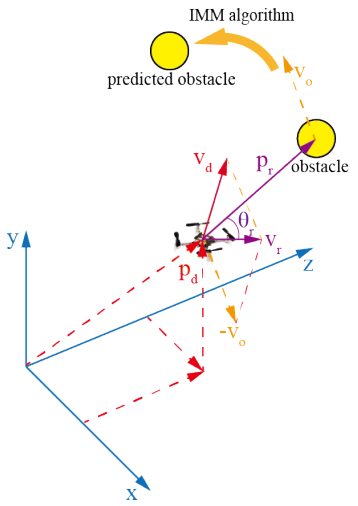

Figure 4. The schematic diagram of the proposed obstacle avoidance algorithm with IMM.

With (24) and (26) in mind, we formulate the optimization problem as follows:

$$\begin{aligned}\min_{u,\varepsilon,\delta}\quad & \lambda_u\|u-u_f\|^2+\lambda_\varepsilon(\varepsilon(k)-\varepsilon^*(k))^2+\lambda_\delta\delta^2\\ \text{s.t.}\quad & \Delta\psi_1(\bar{p}(k),\bar{p}_o(k))\geq-\alpha_2\varepsilon(k)\psi_1(\bar{p}(k),\bar{p}_o(k))\\ & \Delta V_\varepsilon(\varepsilon(k))+\eta V_\varepsilon(\varepsilon(k))\leq\delta,\quad \delta\geq 0\\ & u_{\min}\leq u\leq u_{\max}\\ & 0\leq\varepsilon(k)\leq\frac{1}{\alpha_2}\end{aligned}\tag{29}$$

where λ_u, λ_ϵ, and λ_δ are three positive parameters.

The solution to the optimization problem (29) produces a control input $u(k)$ that strictly ensures obstacle avoidance and asymptotically drives the UAV to the target position.

Remark 3. *The optimization problem (29) serves as a hierarchical safety filter that addresses the conflict between mission performance and strict safety. By minimizing the control deviation $\|u-u_f\|^2$, the framework ensures that the RL-based guidance is minimally modified to maintain the forward invariance of the safe set. The choice of $\lambda_\delta \gg \lambda_u$ ensures the hierarchy of safety over performance, while the specific configuration of $\alpha_1 > \alpha_2$ accommodates the inherent actuation lag in the drone's second-order kinematics. The adaptive coefficient $\varepsilon(k)$ allows the safety margin to adjust dynamically based on relative kinematics, tightening constraints during high-risk encounters to prioritize obstacle avoidance. Additionally, the integration of the CLF and relaxation factor δ ensures smooth parameter transitions and numerical feasibility, achieving a robust balance between tracking efficiency and formal safety guarantees. The notations used are listed in Table 1.*

Table 1. The parameters of the three control methods.

Symbol	Definition	Value
RL Parameters		
k_{ma}	Max acceleration limit per axis	1.0 m/s^2
M_{tr}	Terminal reward for reaching target threshold (ϵ_{tr})	100.0
M_{ds}	Weight for dense distance-to-target reduction	1.0
M_{bp}	Penalty for exceeding flight boundaries (Out-of-Bounds)	10.0
M_{tp}	Step-wise time penalty for efficiency	0.1
PID Controller Gains		
$K_{P,pos}$	Proportional gain (Position: *X*, *Y*, *Z*)	[0.4, 0.4, 1.25]
$K_{I,pos}$	Integral gain (Position: *X*, *Y*, *Z*)	[0.05, 0.05, 0.05]
$K_{D,pos}$	Derivative gain (Position: *X*, *Y*, *Z*)	[0.2, 0.2, 0.5]
$K_{P,att}$	Proportional gain (Attitude: R, P, Y)	[312k, 312k, 70k]
ACBF Parameters		
u_{max}	Physical motor thrust limit (normalized)	$\pm$1.0
a_{max}	Physical acceleration limits (m/s^2)	$\pm$3.0
α_1, α_2	Convergence rates for safe set	0.8, 0.5
$\varepsilon_{\max}$	Maximum adaptive safety margin	2.0
$\lambda_u, \lambda_\varepsilon$	Tracking and adaptive safety weights	1.0, 10.0
λ_δ	Penalty for safety constraint violation	10^4

4. Simulation Validation

To rigorously evaluate the advantages of the proposed control method (named RL+ACBF), we compare it against two baseline configurations. The first is a classical tracking controller (named PID+ACBF), which is augmented with our proposed safety filter with the traditional PID method [3]. The second is a standalone PPO controller (named Pure RL), which is trained for goal-reaching without an explicit safety layer and represents a purely stochastic approach to obstacle avoidance.

In this section, we implement simulations using gym-pybullet-drones [38], a Python-based physics simulation engine that includes a swarm of Crazyflie drones and obstacles represented by spheres. The parameters of the CrazyFlie drones are tabulated in Table 2.

Table 2. Modelling parameters of Crazyflie.

Variables	Definition	Value
Δt	Simulation time step	0.05 s
g	Gravitational acceleration	9.8 kg · m/s^2
m	Mass of quadrotor	0.027 kg
r_d	Drone collision radius	0.2 m
k_{ma}	Maximum acceleration limit	12.0 m/s^2
$\mathcal{B}$	Flight space boundaries (Min/Max)	$\pm$10.0 m
ϵ_{tr}	Target reached threshold	0.1 m
l_x, l_y	Inertia about *x*, *y*-axis	2.39×10^{-5} kg·m^2
l_z	Inertia about *z*-axis	3.23×10^{-5} kg·m^2
k_f	Motor's thrust constant	3.16×10^{-10}
k_m	Motor's torque constant	7.94×10^{-12}

To verify the superiority of the proposed RL+ACBF controller, three distinct scenarios are designed for evaluation. These cases cover quadrotor interaction with: (1) linearly moving obstacles, (2) obstacles with accelerated motion, and (3) obstacles traveling along an eight-shaped trajectory. For fair comparison, identical control parameters are adopted across all three scenarios, as summarized in Table 1.

The RL parameters are identical for both RL+ACBF and Pure RL, and the ACBF parameters are consistent between RL+ACBF and PID+ACBF. In the proposed RL+ACBF control method, the prediction horizon is set as $m = 5$. The baseline PID gains $K_{P,pos}$, $K_{I,pos}$, and $K_{D,pos}$ are adopted to regulate the spatial position and velocity of the drone, whereas $K_{P,att}$ guarantees stable attitude tracking by converting orientation errors into corrective motor torques.

Case 1: linearly moving obstacles. In this baseline scenario, a spherical obstacle moves at a constant linear velocity and is placed directly on the optimal path connecting the start and target positions.

As illustrated in Figure 5, the trajectories generated by the Pure RL method are significantly smoother and more direct than those generated by the PID+ACBF method. Furthermore, the trajectory produced by our proposed RL+ACBF framework initiates a gradual turning maneuver well before reaching the obstacle. This demonstrates that the IMM prediction mechanism accurately anticipates the obstacle's future trajectory, enabling the controller to achieve obstacle avoidance with minimal control effort. Although the PID+ACBF method successfully avoids collisions, it exhibits severe overcorrection when approaching the safety boundary, leading to an unstable flight path.

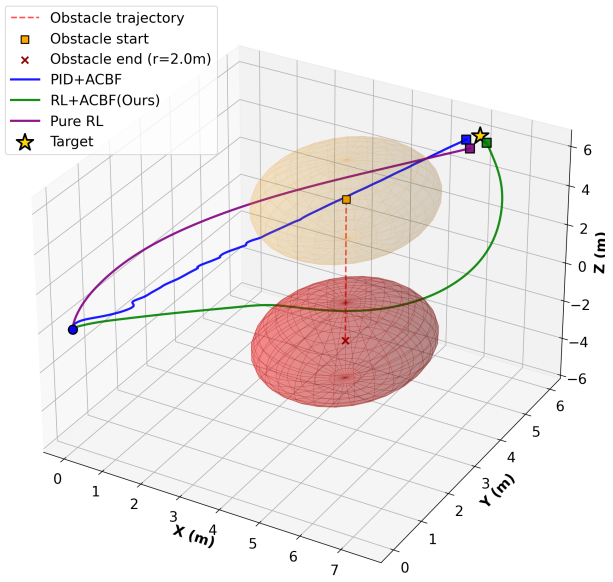

Figure 5. Case 1: 3D trajectories for three control methods.

Figure 6 presents the minimum distance to the obstacle for each control approach, while Figure 7 shows the 3D position, velocity, and acceleration profiles of the UAV under the three controllers. Despite the fact that the Pure RL agent exhibits slight oscillations near the obstacle boundary due to the lack of hard safety constraints, both ACBF-integrated methods strictly maintain $h(\mathbf{x}, t) \geq 0$ throughout the entire trajectory. Compared with the PID+ACBF scheme, the proposed RL+ACBF framework exhibits a more efficient approach to the target. Although the 3D trajectory profiles in Figure 5 show similar path lengths, the state curves in Figure 7 reveal that the PID+ACBF method undergoes significant speed fluctuations and overcorrections when the safety constraints are active. These control oscillations result in a prolonged stabilization phase, making the total navigation time longer than the proposed method.

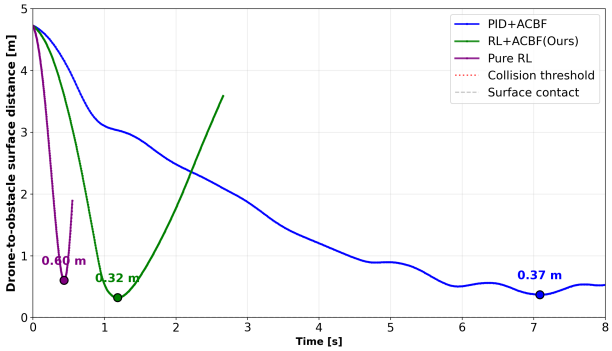

Figure 6. Case 1: Minimum distance between UAV with obstacle for three control methods.

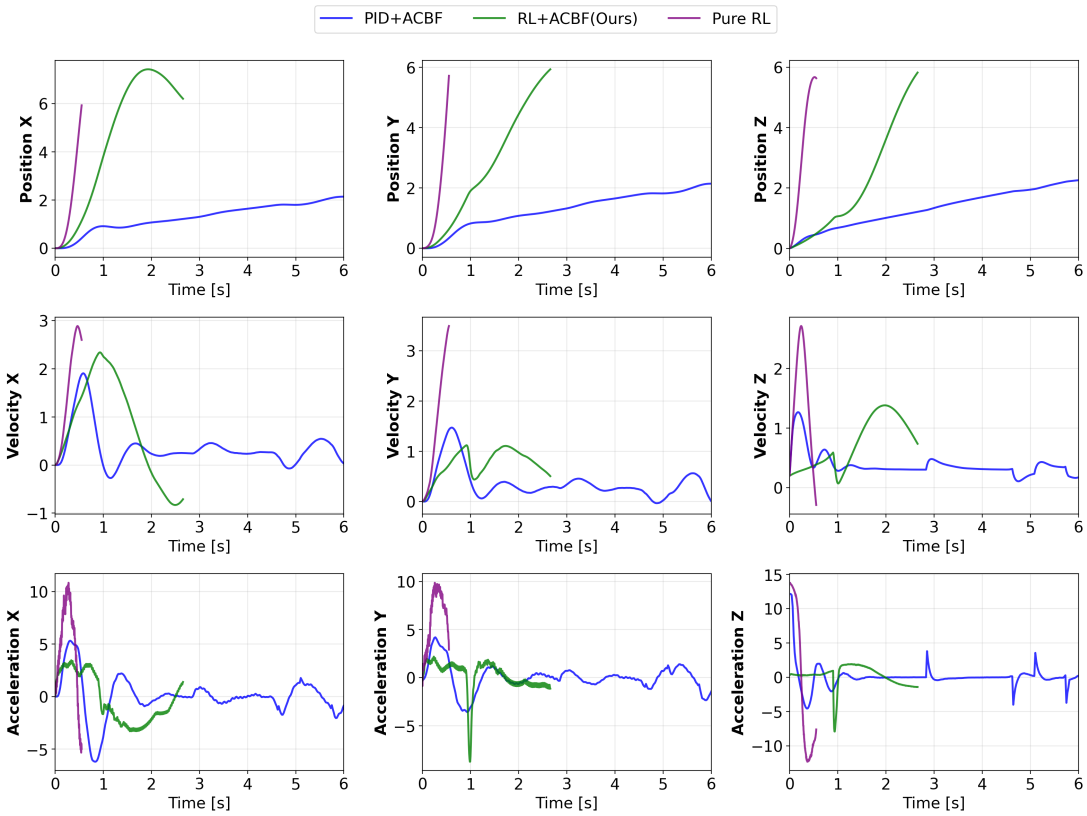

Figure 7. Case 1: State for three control methods.

Case 2: Obstacles with accelerated motion. In the second scenario, the complexity is increased by introducing an obstacle moving along a linear trajectory with constant acceleration. This case evaluates the responsiveness of the controller to time-varying velocity.

Figures 8–10 present the 3D flight trajectories, minimum distances between the UAV and the obstacle, and state curves, respectively. As shown in Figures 9 and 10, the Pure RL method yields a straight flight path toward the target but relies on complex constrained optimization during training. In contrast, our

framework removes the requirement for retraining the RL policy to handle obstacle avoidance. A core advantage of the proposed method lies in its high training efficiency and modular architecture. Moreover, the kinematic curves in Figure 8 verify that RL+ACBF maintains smoother and more stable velocity and acceleration responses. Without the aggressive last-minute corrections observed in the PID+ACBF strategy, our approach shortens the overall mission time while achieving higher energy efficiency.

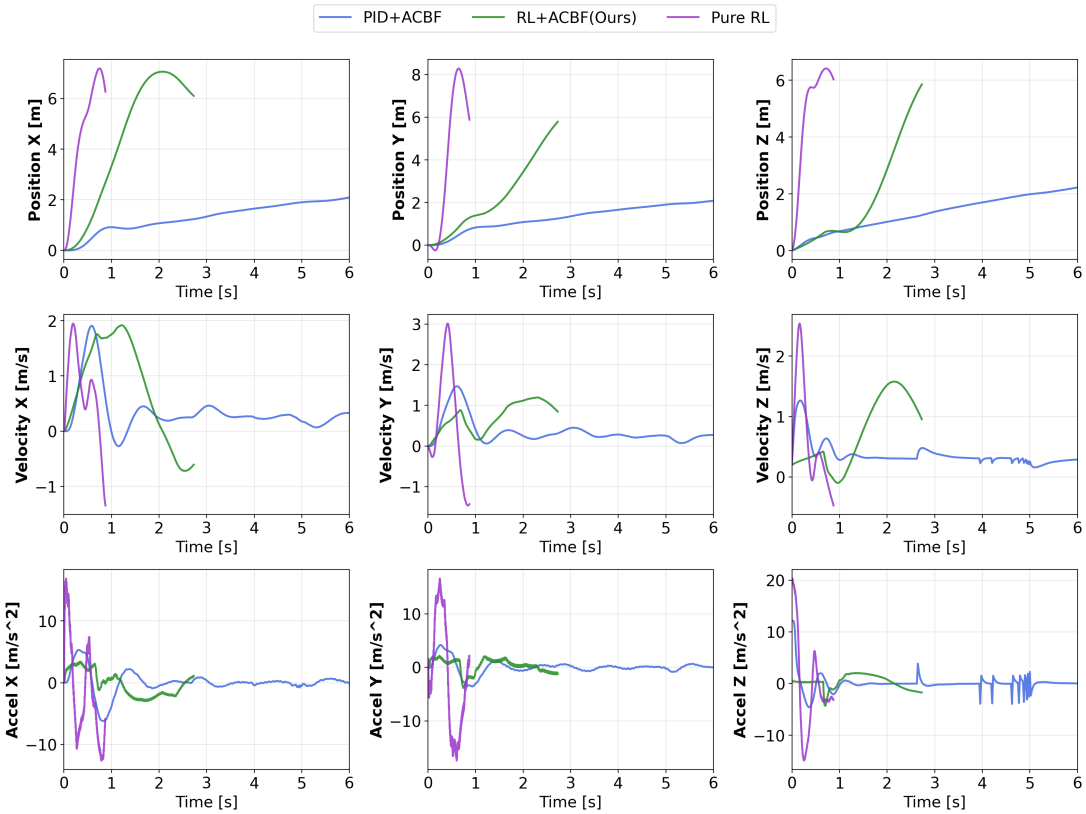

Figure 8. Case 2: State for three control methods.

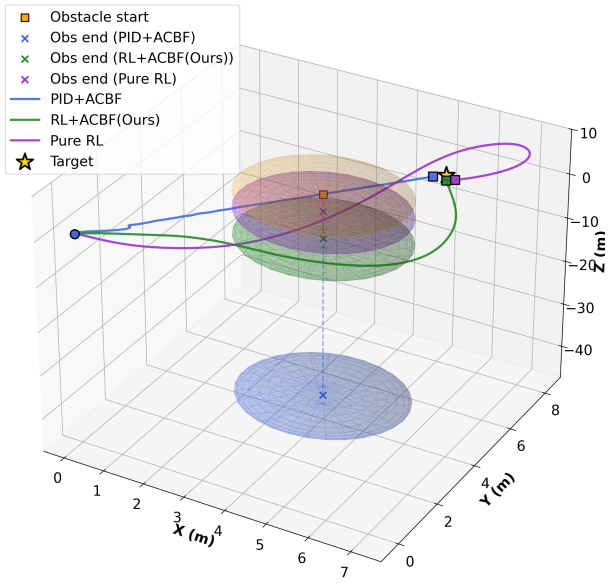

Figure 9. Case 2: 3D trajectories for three control methods.

Case 3: Obstacles with complex non-linear maneuvering. The most challenging scenario involves the obstacle following an "8"-shaped trajectory, characterized by continuous and simultaneous changes in both acceleration and heading. To evaluate adaptability in dynamic environments, Figure 11 gives the 3D trajectories under the three control methods. As shown in Figure 11, while the Pure RL agent generates a direct path, its end-to-end nature requires full knowledge of the obstacle's trajectory during training, leading to severe overfitting and excessive training time. Conversely, our hierarchical framework exhibits robust zero-shot generalization: the RL agent is trained only once with a static obstacle, while the ACBF filter handles unknown dynamic maneuvers in real-time. This allows for a safe detour where the traditional PID method fails and collides (Figure 12).

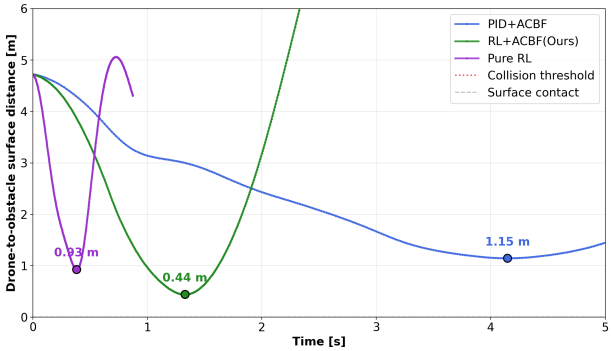

Figure 10. Case 2: Minimum distance between UAV with obstacle for three control methods.

Furthermore, the state curves in Figure 13 reveal that our method maintains smooth, bounded velocity and acceleration

throughout the evasive maneuver. Unlike the baseline methods that may exhibit erratic oscillations when constraints tighten, our framework ensures control stability and numerical feasibility, effectively bridging the gap between high-level intelligence and low-level physical safety.

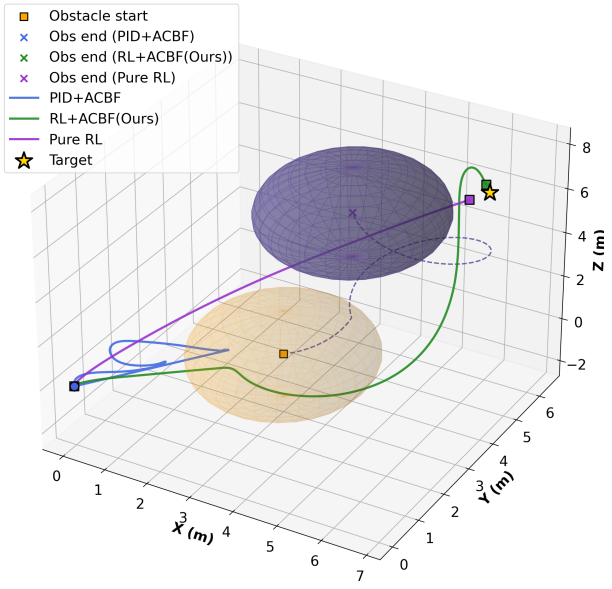

Figure 11. Case 3: 3D trajectories for three control methods.

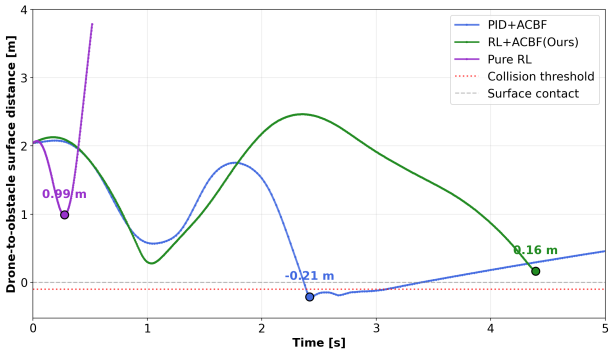

Figure 12. Case 3: Minimum distance between UAV with obstacle for three control methods.

As summarized in Table 3, the results reflect a critical trade-off between mission-specific optimization and environmental adaptability. While end-to-end Pure RL methods achieve local optimality in their respective training cases, they fail to generalize to unseen maneuvers, resulting in collisions (zero minimum distance) when the obstacle trajectory changes. In contrast, our proposed RL+ACBF framework, trained only on static obstacles, ensures consistent safety margins across all test cases with a significantly reduced training overhead. The results confirm that our proposed scheme not only guarantees strict safety but also optimizes the control effort, outperforming traditional and end-to-end learning-based methods.

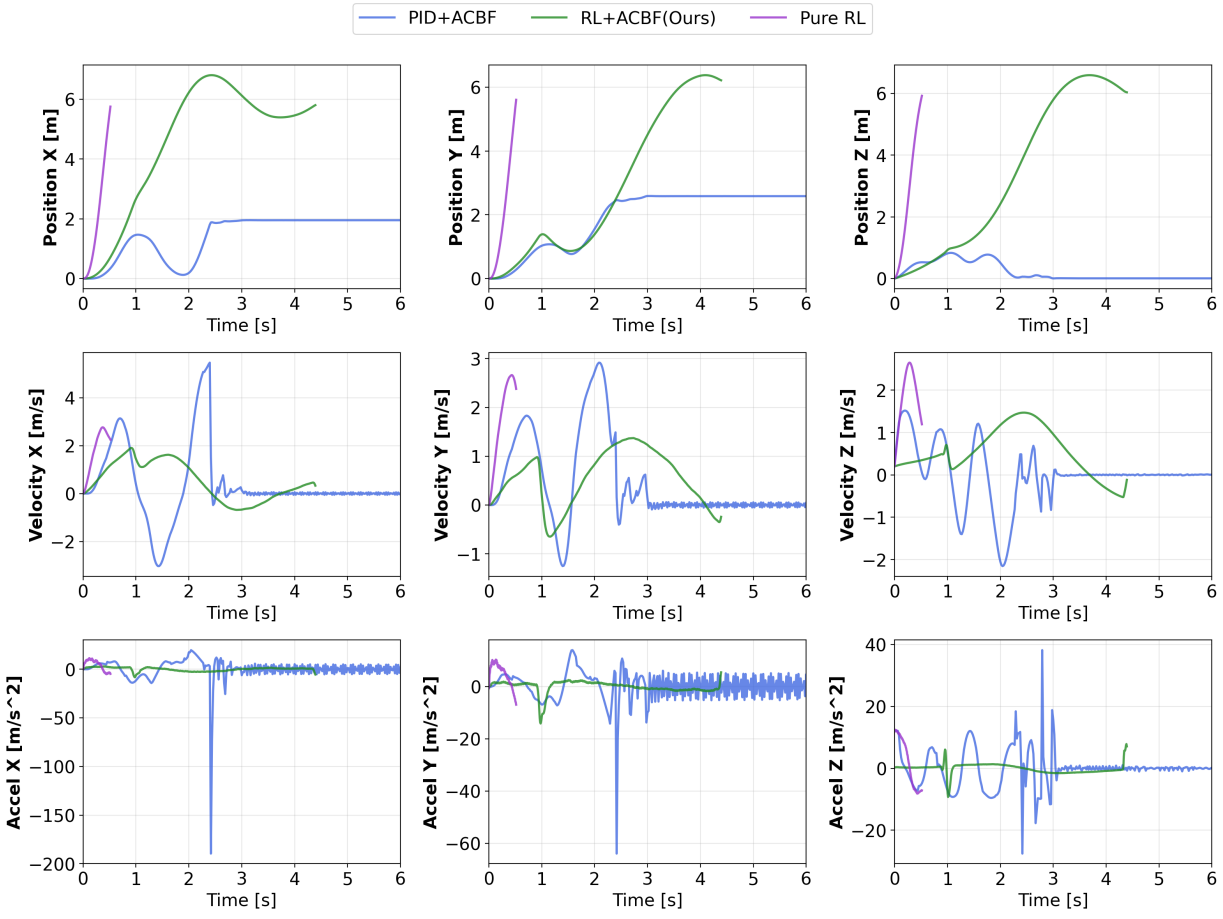

Figure 13. Case 3: State for three control methods.

Table 3. Comparison of training efficiency and safety performance

Method	Training Time	Min Distance (Case 1/2/3)
Pure RL (Case 1)	142 s	0.6 m / 0 m / 0 m
Pure RL (Case 2)	160 s	0.84 m / 0.93 m / 0 m
Pure RL (Case 3)	187 s	0 m / 0 m / 0.99 m
PID + ACBF	N/A	0.37 m / 1.15 m / 0 m
RL + ACBF	136 s	0.32 m / 0.44 m / 0.16 m

5. Conclusions

In this paper, we proposed a hierarchical safe control framework for UAV navigation in complex dynamic environments. By integrating RL with ACBF and IMM, we addressed the conflict between optimal mission performance and strict safety guarantees. The RL agent provides high-level tracking guidance, while the ACBF acts as a formal safety layer that minimally modifies control commands to ensure forward invariance of the safe set, even under high relative degree constraints. To handle the uncertainty of dynamic obstacles, the IMM filter provides robust state predictions, significantly enhancing the filter's situational awareness. Simulation results across various scenarios, including complex nonlinear maneuvers, demonstrate that our proposed RL+ACBF framework achieves a superior success rate and lower tracking error compared to baseline methods. Future work will focus on extending this framework to multi-UAV swarm coordination and validating the performance on physical hardware platforms.

Author Contributions

J.G.: Conceptualization, methodology, investigation, software, validation, writing—original draft preparation, writing—reviewing and editing; J.S.: Conceptualization, methodology, supervision, writing—reviewing and editing. All authors have read and agreed to the published version of the manuscript.

Funding

This research was funded by the National Natural Science Foundation of China under Grant 62373159.

Institutional Review Board Statement

Not applicable.

Informed Consent Statement

Not applicable.

Data Availability Statement

Not applicable.

Conflicts of Interest

The authors declare no conflict of interest.

Use of AI and AI-Assisted Technologies

During the preparation of this work, the authors used Doubao to polish and refine the manuscript text. After using this tool/service, the authors reviewed and edited the content as needed and take full responsibility for the content of the published article.

References

1. Xiao, B.; Yin, S. A New Disturbance Attenuation Control Scheme for Quadrotor Unmanned Aerial Vehicles. *IEEE Trans. Ind. Inform.* **2017**, *13*, 2922–2932.
2. Xian, B.; Yang, S. Robust Tracking Control of a Quadrotor Unmanned Aerial Vehicle-Suspended Payload System. *IEEE/ASME Trans. Mechatron.* **2021**, *26*, 2653–2663.
3. Najm, A.A.; Ibraheem, I.K. Nonlinear PID controller design for a 6-DOF UAV quadrotor system. *Eng. Sci. Technol. Int. J.* **2019**, *22*, 1087–1097.
4. Dong, J.; He, B. Novel fuzzy PID-type iterative learning control for quadrotor UAV. *Sensors* **2019**, *19*, 24.
5. Lindqvist, B.; Mansouri, S.S.; Agha-mohammadi, A.; et al. Nonlinear MPC for Collision Avoidance and Control of UAVs With Dynamic Obstacles. *IEEE Robot. Autom. Lett.* **2020**, *5*, 6001–6008.
6. Zheng, E.-H.; Xiong, J.-J.; Luo, J.-L. Second order sliding mode control for a quadrotor UAV. *ISA Trans.* **2014**, *53*, 1350–1356.
7. Yu, D.; Ma, S.; Liu, Y.-J.; et al. Finite-time adaptive fuzzy backstepping control for quadrotor UAV with stochastic disturbance. *IEEE Trans. Autom. Sci. Eng.* **2024**, *21*, 1335–1345.
8. Zhu, Q.; Liu, Z.; Golestani, M.; et al. Strong robust and large time-delay-independent control for unknown nonlinear nonaffine neutral-type systems. *J. Frankl. Inst.* **2026**, *363*, 108859.
9. Zhu, Q.; Zhang, W.; Li, S.; et al. U-control—A universal platform for control system design with inversion/cancellation of nonlinearity, dynamic and coupling through model-based to model-free procedures. *Int. J. Syst. Sci.* **2025**, *56*, 484–501.
10. Zhu, Q.; Lei, C.; Shi, B.; et al. Model-free robust underactuated control of inverted pendulums. *J. Frankl. Inst.* **2025**, *362*, 107911.
11. Kaufmann, E.; Bauersfeld, L.; Loquercio, A.; et al. Champion-level drone racing using deep reinforcement learning. *Nature* **2023**, *620*, 982–987.
12. Yu, Z.; Li, J.; Xu, Y.; et al. Reinforcement Learning-Based Fractional-Order Adaptive Fault-Tolerant Formation Control of Networked Fixed-Wing UAVs With Prescribed Performance. *IEEE Trans. Neural Netw. Learn. Syst.* **2024**, *35*, 3365–3379.
13. Wang, X.; Wang, S.; Liang, X.; et al. Deep Reinforcement Learning: A Survey. *IEEE Trans. Neural Netw. Learn. Syst.* **2024**, *35*, 5064–5078.
14. Brunke, L.; Greeff, M.; Hall, A.W.; et al. Safe learning in robotics: From learning-based control to safe reinforcement learning. *Annu. Rev. Control Robot. Auton. Syst.* **2022**, *5*, 411–444.
15. Liu, J.; Yang, J.; Mao, J.; et al. Flexible Active Safety Motion Control for Robotic Obstacle Avoidance: A CBF-Guided MPC Approach. *IEEE Robot. Autom. Lett.* **2025**, *10*, 2686–2693.
16. Tessler, C.; Mankowitz, D.J.; Mannor, S. Reward constrained policy optimization. *arXiv* **2018**, arXiv:1805.11074.
17. Achiam, J.; Held, D.; Tamar, A.; et al. Constrained Policy Optimization. In Proceedings of the 34th International Conference on Machine Learning, Sydney, Australia, 6–11 August 2017;

Volume 70, pp. 22–31.

18. Tan, X.; Dimarogonas, D.V. Distributed Implementation of Control Barrier Functions for Multi-agent Systems. *IEEE Control Syst. Lett.* **2022**, *6*, 1879–1884.
19. Rao, J.; Xiang, C.; Xi, J.; et al. Path planning for dual UAVs cooperative suspension transport based on artificial potential field-A* algorithm. *Knowl. Based Syst.* **2023**, *277*, 110797.
20. Hu, J.; Wang, M.; Zhao, C.; et al. Formation control and collision avoidance for multi-UAV systems based on Voronoi partition. *Sci. China Technol. Sci.* **2020**, *63*, 65–72.
21. Zhou, M.; Wang, Z.; Wang, J.; et al. Multi-Robot Collaborative Hunting in Cluttered Environments with Obstacle-Avoiding Voronoi Cells. *IEEE/CAA J. Autom. Sin.* **2024**, *11*, 1643–1655.
22. Ames, A.D.; Grizzle, J.W.; Tabuada, P. Control barrier function based quadratic programs with application to adaptive cruise control. In Proceedings of the 53rd IEEE Conference on Decision and Control, Los Angeles, CA, USA, 15–17 December 2014; pp. 6271–6278.
23. Ames, A.D.; Coogan, S.; Egerstedt, M.; et al. Control Barrier Functions: Theory and Applications. In Proceedings of the 2019 18th European Control Conference (ECC), Naples, Italy, 25–28 June 2019; pp. 3420–3431.
24. Sun, J.; Yang, J.; Zeng, Z. Safety-Critical Control with Control Barrier Function Based on Disturbance Observer. *IEEE Trans. Autom. Control* **2024**, *69*, 4750–4756.
25. Ames, A.D.; Xu, X.; Grizzle, J.W.; et al. Control barrier function based quadratic programs for safety critical systems. *IEEE Trans. Autom. Control* **2017**, *62*, 3861–3876.
26. Wang, P.; Yu, C.; Deng, F.; et al. Reinforcement Learning-Based Optimal Formation Tracking for UAVs with Safety Constraints. *IEEE Trans. Neural Netw. Learn. Syst.* **2026**, *37*, 2969–2982.
27. Zhang, C.; Wang, S.; Meng, S.; et al. Safe Exploration of Reinforcement Learning with Data-Driven Control Barrier Function. In Proceedings of the 2022 China Automation Congress (CAC), Xiamen, China, 25–27 November 2022; pp. 1008–1013.
28. Emam, Y.; Notomista, G.; Glotfelter, P.; et al. Safe Reinforcement Learning Using Robust Control Barrier Functions. *IEEE Robot. Autom. Lett.* **2025**, *10*, 2886–2893.
29. Xiao, W.; Belta, C.; Cassandras, C.G. Adaptive Control Barrier Functions. *IEEE Trans. Autom. Control* **2022**, *67*, 2267–2281.
30. Lopez, B.T.; Slotine, J.-J.E.; How, J.P. Robust Adaptive Control Barrier Functions: An Adaptive and Data-Driven Approach to Safety. *IEEE Control Syst. Lett.* **2021**, *5*, 1031–1036.
31. Yuan, T.; Bar-Shalom, Y.; Willett, P.; et al. A multiple IMM estimation approach with unbiased mixing for thrusting projectiles. *IEEE Trans. Aerosp. Electron. Syst.* **2012**, *48*, 3250–3267.
32. Hanumegowda, A.; Dewangan, S.; Bhupala, S.; et al. Extended Object Tracking with IMM Filter for Automotive Pre-Crash Safety Applications. In Proceedings of the 2021 18th European Radar Conference (EuRAD), London, UK, 5–7 April 2022; pp. 177–180.
33. Mazor, E.; Averbuch, A.; Bar-Shalom, Y.; et al. Interacting multiple model methods in target tracking: A survey. *IEEE Trans. Aerosp. Electron. Syst.* **1998**, *34*, 103–123.
34. Kim, B.; Yi, K.; Yoo, H.-J.; et al. An IMM/EKF Approach for Enhanced Multitarget State Estimation for Application to Integrated Risk Management System. *IEEE Trans. Veh. Technol.* **2015**, *64*, 876–889.
35. Cui, R.; Yang, C.; Li, Y.; et al. Adaptive Neural Network Control of AUVs with Control Input Nonlinearities Using Reinforcement Learning. *IEEE Trans. Syst. Man Cybern. Syst.* **2017**, *47*, 1019–1029.
36. Ding, L.; Li, S.; Gao, H.; et al. Adaptive Partial Reinforcement Learning Neural Network-Based Tracking Control for Wheeled Mobile Robotic Systems. *IEEE Trans. Syst. Man Cybern. Syst.* **2020**, *50*, 2512–2523.
37. Xiong, Y.; Zhai, D.-H.; Tavakoli, M.; et al. Discrete-Time Control Barrier Function: High-Order Case and Adaptive Case. *IEEE Trans. Cybern.* **2023**, *53*, 3231–3239.
38. Panerati, J.; Zheng, H.; Zhou, S.; et al. Learning to fly—A gym environment with pybullet physics for reinforcement learning of multi-agent quadcopter control. In Proceedings of the 2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Prague, Czech Republic, 27 September–1 October 2021; pp. 7512–7519.