drones

MDPI

Article

Adaptive Impact-Time-Control Cooperative Guidance Law for UAVs Under Time-Varying Velocity Based on Reinforcement Learning

Zhenyu Liu, Gang Lei, Yong Xian, Leliang Ren *, Shaopeng Li and Daqiao Zhang

Xi'an Research Institute of High Technology, Xi'an 710025, China; lzy568811@163.com (Z.L.)
* Correspondence: renleliang@126.com

Abstract: In this study, an adaptive impact-time-control cooperative guidance law based on deep reinforcement learning considering field-of-view (FOV) constraints is proposed for high-speed UAVs with time-varying velocity. Firstly, a reinforcement learning framework for the high-speed UAVs' guidance problem is established. The optimization objective is to maximize the impact velocity; and the constraints for impact time, dive attacking, and FOV are considered simultaneously. The time-to-go estimation method is improved so that it can be applied to high-speed UAVs with time-varying velocity. Then, in order to improve the applicability and robustness of the agent, environmental uncertainties, including aerodynamic parameter errors, observation noise, and target random maneuvers, are incorporated into the training process. Furthermore, inspired by the RL^2 algorithm, the recurrent layer is introduced into both the policy and value network. In this way, the agent can automatically adapt to different mission scenarios by updating the hidden states of the recurrent layer. In addition, a compound reward function is designed to train the agent to satisfy the requirements of impact-time control and dive attack simultaneously. Finally, the effectiveness and robustness of the proposed guidance law are validated through numerical simulations conducted across a wide range of scenarios.

Keywords: multiple high-speed UAVs; cooperative guidance; impact-time-control guidance; reinforcement learning; field-of-view constraints

check for updates

Academic Editors: Haibin Duan, Jialong Zhang and Shuangxi Liu

Received: 7 March 2025
Revised: 26 March 2025
Accepted: 27 March 2025
Published: 29 March 2025

Citation: Liu, Z.; Lei, G.; Xian, Y.; Ren, L.; Li S.; Zhang, D. Adaptive Impact-Time-Control Cooperative Guidance Law for UAVs Under Time-Varying Velocity Based on Reinforcement Learning. *Drones* **2025**, *9*, 262. https://doi.org/10.3390/drones9040262

Copyright: © 2025 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 (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

As mission scenarios for multiple cross-domain UAVs have become increasingly complex, the requirements of UAV guidance systems have also become more sophisticated [1–3]. The implementation of saturation attacks through impact-time-control guidance (ITCG) can effectively improve the penetration probability of cross-domain UAVs when facing a defense system [1]. It is therefore evident that impact-time-control guidance has become a crucial aspect of guidance technology.

The existing studies on ITCG could be divided into two categories. The first category is based on the biased proportional navigation guide (BPNG) law [4,5], which consists of the PNG law and a bias term for correcting the arrival time error. The second category is based on modern advanced control theories, including stability theory [6,7], sliding-mode variable structures [8–10], and backstepping control [11–13]. However, the BPNG laws are unable to guarantee robustness in complex environments, while guidance laws based on modern advanced control theories usually require accurate estimation of time-to-go.

To prevent high-speed UAVs from losing track of the target, the seeker's FOV limits must be considered. There are several approaches to dealing with FOV constraints, one

of which is to use compound guidance laws consisting of two or more phases [14–16]. However, the discontinuous guidance command at the moment of switching guidance laws may result in undesired transient responses. Another approach to addressing the FOV constraints is to shape the trajectory of the look angle [17–19], then the guidance commands are generated by designing reference-trajectory tracking control laws. Other approaches such as optimal control theory [2,20] or nonlinear control theory [21,22] have also been used to address FOV constraints. Although there have been many studies on ITCG and FOV constraints, existing methods are difficult to apply directly to high-speed UAVs with time-varying velocity. On the other hand, the coupling effects between the longitudinal and lateral channel in three-dimensional (3D) space cannot be ignored.

In recent years, computational guidance algorithms have received increasing attention due to the rapid development of embedded computing capabilities. The authors in ref. [23] proposed a guidance law with terminal time and impact angle constraints by employing model predictive static programming (MPSP). Similarly, the authors in ref. [24] proposed a guidance law for vehicle guidance with state and input constraints by using model predictive convex programming (MPCP) to solve guidance commands. Although these approaches can obtain optimal or suboptimal guidance commands online, a poor initial solution may result in difficulties in achieving convergence during the computational process [25].

Deep reinforcement learning (DRL) techniques have undergone substantial advancements and progress in recent years and have demonstrated exceptional performance in decision-making tasks involving high-dimensional data [26]. However, previous studies have focused on DRL applications in intelligent control, with few studies focusing on guidance problems in aerospace. To solve the missile guidance problem in the terminal flight phase, the DDPG algorithm was used to train the agent [27]. The authors in ref. [28] proposed an ITCG law with FOV constraints, where the guidance gain was obtained by DDPG. Similarly, the authors in ref. [29] proposed an impact-angle-control guidance (IACG) law based on BPNG, where the bias term was obtained by a series of training processes. Ref. [30] proposed an terminal flight guidance law for hypersonic gliding vehicles employing the meta-reinforcement learning technique. In Ref. [31], the authors proposed an ITCG algorithm based on a general prediction–correction framework; the method employed a DNN to predict the time-to-go, then the correction commands were learned by using proximal policy optimization (PPO). The author in [32] developed a cooperative interception guidance law for multiple missiles by using the action-enhanced multi-agent deep deterministic policy gradient (EA-MADDPG) algorithm. However, the FOV constraints and velocity with time-varying conditions were not considered. Although the agent training and learning process may be time-consuming when utilizing DRL methods, acquiring guidance commands after deployment requires only computing the output of the neural network.

Although many guidance laws have been developed so far, there are still challenges that need to be addressed through further research to achieve practical implementation. First, the majority of existing studies on ITCG are developed using ideal kinematic assumptions. However, the applicability of these assumptions needs to be carefully examined, especially in the context of UAVs with time-varying velocity and three-dimensional pursuit scenarios. Second, most existing ITCG laws require an accurate estimated value of the time-to-go; therefore, the accuracy of the time-to-go estimation could directly affect the performance of ITCG laws. Third, the application of the DRL method to the development of guidance laws still presents certain limitations that require further attention. These include the development of guidance laws for high-speed UAVs with time-varying velocity and the consideration of FOV constraints on the guidance process.

Based on the above discussion, this study proposes an adaptive ITCG law with FOV constraints for high-speed UAVs with time-varying velocity. The objective is to achieve a cooperative dive attack and maximize impact velocity. To this end, the high-speed UAVs' guidance problem is first reconstructed in the DRL framework. Furthermore, uncertainties of the environment are considered, including random perturbation biases in aerodynamic coefficients and atmospheric density, random maneuvers of the target, and observation noise. We also impose path constraints on the FOV angle and overload. Finally, a compound reward function is proposed to compensate for guide accuracy, energy consumption, and impact-time error. The state-of-the-art DRL algorithm PPO is used to train the agent that maps the observations to guidance commands. Inspired by the RL^2 algorithm [33], a gated recurrent unit (GRU) layer is introduced into the policy network. This enables the well-trained agent to rapidly adapt to the different environment and tasks by updating the hidden state of the GRU layer during deployment.

The contributions of this paper are summarized below. Firstly, the ITCG law for high-speed UAVs with time-varying velocity is developed on the basis of DRL techniques. The objective of this study was to achieve terminal time control for high-speed UAVs with time-varying velocity and to enable dive attacks; a compound reward function is proposed that consists of a positive reward and negative penalty to ensure the high-speed UAV arrives at the target while continuously correcting the arrival time. Secondly, a straightforward but effective time-to-go estimation approach for time-varying velocity is proposed, which has higher accuracy than the analytical formula prediction method. In addition, to improve the applicability of the guidance law in practical application scenarios, FOV constraints and overload constraints are considered. Third, inspired by the RL^2 algorithm [33], the gated recurrent unit (GRU) layers are introduced into the policy network and the value network to achieve rapid adaptation to different tasks by continuously updating the hidden states of the GRU layers.

This paper is organized as follows: Section 2 briefly reviews the PPO algorithm. Section 3 formulates the guidance problem for the UAV. Section 4 details the proposed guidance law. Section 5 shows the results of the numerical simulations. Finally, Section 6 provides the conclusion.

2. Preliminary

The proximal policy optimization algorithm (PPO) [34] is recognized for its benchmark performance in complex control tasks. It implements a two-stream neural network structure: the actor network is used to generate the action for the observation at the current time step, and the critic network is used to approximate the state value function, which evaluates the value of the current state.

Let θ, w denote the parameters of the actor network and the critic network, respectively. The probability ratio of the new policy π_θ to the old policy π_{θ^-} is calculated as follows:

$$p_t(\theta) = \frac{\pi_\theta(a_t|s_t)}{\pi_{\theta^-}(a_t|s_t)} \tag{1}$$

The optimization objective of the PPO algorithm is to maximize the objective function as follows:

$$J(\theta) = \mathbb{E}\big(\min(p_t(\theta), \mathrm{clip}(p_t(\theta), 1-\epsilon, 1+\epsilon))\hat{A}_t\big) \tag{2}$$

where ϵ represents the clip coefficient, and $\hat{A}_k$ represents the estimated value of the advantage function at time step k, which can be obtained as follows:

$$\hat{A}_t = \left[\sum_{k=t}^{T} \gamma^k r_{t+k}\right] - V_w^\pi(s_t) \tag{3}$$

where the state value function V_w^π is updated by minimizing the loss function as follows:

$$L(w) = \sum_{i=1}^{|D|} \left(V_w^\pi(s_t) - \left[\sum_{k=t}^{T} \gamma^k r_{t+k} \right] \right)^2 \tag{4}$$

where $|D|$ represents the total number of collected data. Then, the parameters θ can be updated by maximizing the objective function $J(\theta)$, and the parameters w can be minimized using the loss function $L(w)$, that is,

$$\theta^+ = \theta^- + \alpha_\theta \nabla_\theta J(\theta)|_{\theta=\theta^-} \tag{5}$$

$$w^+ = w^- - \alpha_w \nabla_w L(w)|_{w=w^-} \tag{6}$$

where α_θ and α_w denote the learning rates for the actor network $\pi_\theta(a_t|s_t)$ and the critic network $V_w^\pi(s_t)$, respectively.

3. Problem Statement

As part of the system design, we established a local geodetic reference frame (Figure 1) following the North–East–Down (NED) convention: the x-axis aligned with true north, the z-axis pointed eastward, and the y-axis completed the orthogonal right-handed coordinate system. The following foundational assumptions were introduced to formulate the problem.

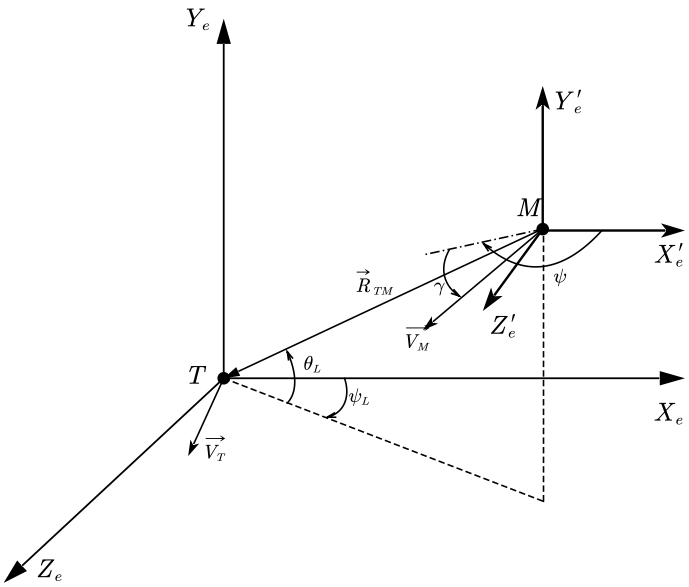


Figure 1. Engagement geometry.

Assumption 1. *The vehicle adopts a bank-to-turn (BTT) control scheme, that is, the side slip angle β of the high-speed UAV is always steered to be zero.*

Assumption 2. *The high-speed UAV and target are treated as mass points.*

Assumption 3. *The target's velocity is much smaller than that of the high-speed UAV.*

This paper focused primarily on the guidance problem of high-speed UAVs during the terminal flight phase; therefore, the effects of Earth's curvature were neglected. Thus, according to Assumptions 1 and 2, the 3D motion equations of a high-speed UAV could be expressed as follows [35]:

$$\dot{x}_M = V_M \cos\gamma \cos\psi \tag{7a}$$
$$\dot{y}_M = V_M \sin\gamma \tag{7b}$$
$$\dot{z}_M = V_M \cos\gamma \sin\psi \tag{7c}$$
$$\dot{V}_M = -\frac{D}{m} - g\sin\gamma \tag{7d}$$
$$\dot{\gamma} = \frac{L\cos\sigma}{mV_M} - \frac{g}{V}\cos\gamma \tag{7e}$$
$$\dot{\psi} = \frac{L\sin\sigma}{V_M\cos\gamma} \tag{7f}$$
$$\dot{\alpha} = \Delta\alpha \tag{7g}$$
$$\dot{\sigma} = \Delta\sigma \tag{7h}$$

where x_M, y_M, z_M represent the position coordinates of the vehicle, and V_M denotes the velocity of the UAV. The definitions of other symbols are shown in Figure 1. The gravity acceleration $g = 9.81\ \mathrm{m/s^2}$. α and σ represent the angle of attack and bank angle, respectively. The expressions of the aerodynamic lift and drag forces are as follows:

$$L = \frac{1}{2}\rho V_M^2 S_{ref} C_L \tag{8a}$$
$$D = \frac{1}{2}\rho V_M^2 S_{ref} C_D \tag{8b}$$

where ρ denotes the atmospheric density, S_{ref} represents the reference area, and C_L and C_D are the lift and drag coefficients.

In addition, the actuator delay was also considered, that is,

$$\dot{\Delta\alpha} = \frac{\Delta\alpha_c - \Delta\alpha}{\tau_a} \tag{9a}$$
$$\dot{\Delta\sigma} = \frac{\Delta\sigma_c - \Delta\sigma}{\tau_b} \tag{9b}$$

where τ_a and τ_b are the actuator time constants, $\Delta\alpha$ and $\Delta\sigma$ are the actual angular rates, and $\Delta\alpha_c$ and $\Delta\sigma_c$ are the command angular rates.

Consider a maneuvering target on the ground with the following motion equations:

$$\dot{\boldsymbol{r}}_T = \boldsymbol{v}_T \tag{10a}$$
$$\dot{\boldsymbol{v}}_T = \boldsymbol{a}_T \tag{10b}$$

where $\boldsymbol{r}_T, \boldsymbol{v}_T$, and $\boldsymbol{a}_T$ denote the position, velocity, and acceleration vector of the target, respectively. Because the target only maneuvers on the ground, $y_T = v_{yT} = a_{yT} = 0$. Moreover, the amplitude of the velocity and acceleration of the target need to satisfy the following limits:

$$\boldsymbol{r}_T(t_0) = [0,0,0]^T \tag{11a}$$
$$\|\boldsymbol{v}_T\| \le V_{T_{max}} \tag{11b}$$
$$\|\boldsymbol{a}_T\| \le a_{T_{max}} \tag{11c}$$

Let $\boldsymbol{r}_M = [x_M, y_M, z_M]^T$ denote the position vector of the vehicle and $\boldsymbol{r}_{TM}$ denote the vector from the vehicle to the target; then, the elevation angle θ_L and azimuth angle ψ_L of the line-of-sight (LOS) vector are defined as follows:

$$\theta_L = \arcsin \frac{y_T - y_M}{\|\boldsymbol{r}_{TM}\|} \tag{12a}$$

$$\psi_L = \arctan 2(z_T - z_M, x_T - x_M) \tag{12b}$$

where $\boldsymbol{r}_{TM} = \boldsymbol{r}_T - \boldsymbol{r}_M$, and $\arctan 2(\cdot, \cdot)$ denotes the four-quadrant inverse tangent function. Taking the time derivative of θ_L, ψ_L and combining it with Equation (7a–h), we can obtain

$$\dot{\theta}_L = \frac{V_M \cos\gamma \sin(\psi_L - \psi)}{S_{TM}} \tag{13a}$$

$$\dot{\psi}_L = -\frac{V_M(\cos\theta_L \sin\gamma - \sin\theta_L \cos\gamma \cos(\psi_L - \psi))}{\|\boldsymbol{r}_{TM}\|} \tag{13b}$$

where $S_{TM} = \sqrt{(x_T - x_M)^2 + (z_T - z_M)^2}$ represents the length of the projection of the LOS vector in the horizontal plane.

In addition, the leading angle η of the vehicle, which represents the angle between the LOS vector and the velocity vector of the vehicle, can be calculated as follows:

$$\eta = \arccos\left(\frac{\boldsymbol{r}_{TM}^T \cdot \boldsymbol{V}_M}{\|\boldsymbol{r}_{TM}\| \cdot \|\boldsymbol{V}_M\|}\right) \tag{14}$$

The primary goal of the designed guidance law is to achieve the following final conditions:

$$|t_f - T_d| \leq \epsilon_{\delta_t} \tag{15}$$

$$\|\boldsymbol{r}_M(t_f) - \boldsymbol{r}_T(t_f)\| \leq \epsilon_r \tag{16}$$

$$\gamma_f \leq \gamma_{dive} \tag{17}$$

where t_f and T_d represent the final time and the desired final time, respectively. γ_{dive} represents the minimum path angle required for dive attacking.

Moreover, the path constraints should also be considered, such as the overload constraints and FOV constraints, that is,

$$\frac{\sqrt{L^2 + D^2}}{mg} \leq n_{max} \tag{18}$$

$$|\eta(t) + \alpha(t)| \leq FOV_{max}, \forall t \leq t_f \tag{19}$$

where n_{max} and FOV_{max} denote the allowable maximum limit of the overload and FOV angle of vehicle, respectively.

It should be noted that the majority of existing studies [19,20] express the FOV constraint in the form of $|\eta(t)| \leq FOV_{max}$. This is due to the fact that these studies employ the small-angle-of-attack assumption, by which the angle of attack is regarded as being approximately zero. However, in reality, the vehicle's FOV angle is equal to the sum of the angle of attack and the leading angle (as illustrated in Figure 2), so we expressed the FOV constraints in the form of Equation (19).

Our objective was for the velocity of the vehicle to be as large as possible when it reached the target. Consequently, the designation of the guidance law was reduced to solving the following constrained optimization formulation:

$$\min \quad -V_M(t_f) \tag{20}$$

s.t. (7), (15)–(19).

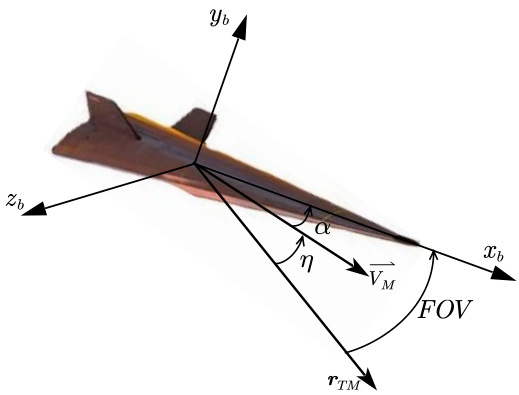


Figure 2. The FOV angle schematic.

4. Method

This section presents a description of the guidance problem within the context of the DRL framework, together with an optimization of the guidance policy through the application of the PPO algorithm. The representative images in Figure 3 illustrate the reinforcement learning guidance framework employed in this study, where the IMU is the inertial measurement unit. The agent generates acceleration commands according to the observation information, and then the autopilot generates the angle of attack and bank angle change rate commands based on the acceleration command.

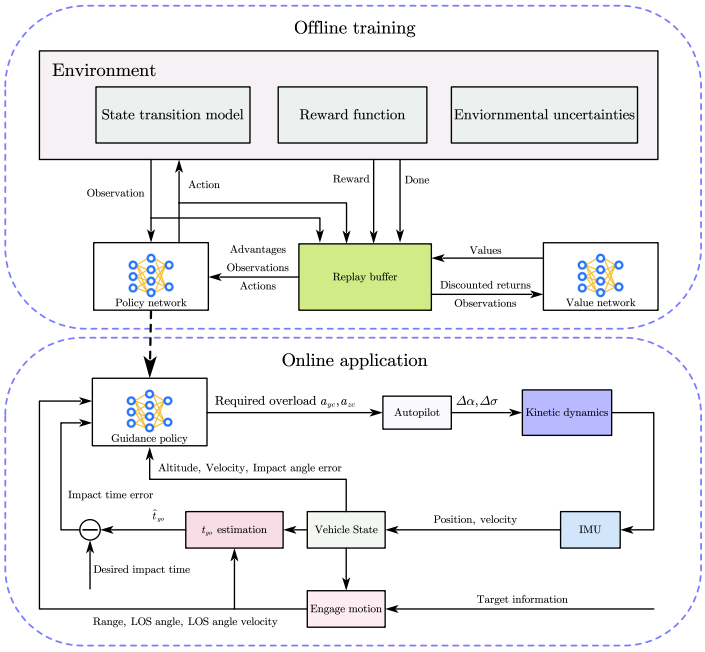


Figure 3. Block diagram of the guidance law.

4.1. Observation Space and Action Space

For the guidance and control problems, the observation vector must fully describe the information about the relative state of the vehicle and target. Therefore, the observation vector was designed as follows:

$$\widetilde{o}_t = \left[\|\boldsymbol{r}_{TM}\|, \boldsymbol{\lambda}, V_M, \dot{\theta}_L, \dot{\psi}_L, \hat{t}_{go}, \delta_t, \delta_\gamma\right]^T \tag{21}$$

where $\boldsymbol{\lambda} = \frac{\boldsymbol{r}_{TM}}{\|\boldsymbol{r}_{TM}\|}$ denotes the LOS uni vector, $\hat{t}_{go}$ denotes the estimated value of the time-to-go t_{go}, $\delta_t = t + \hat{t}_{go} - T_d$ is the estimated arrival time error, and $\delta_\gamma = \frac{\pi}{2} - \theta_L$. Moreover, the observation noise was considered, that is,

$$\boldsymbol{o}_t = \widetilde{\boldsymbol{o}}_t \cdot \mathcal{N}(1.0, \varepsilon_{obs}, n) \tag{22}$$

where $\mathcal{N}(1.0, \varepsilon_{obs}, n)$ denotes the n-dimensional independent Gaussian distributed random variables with mean 1.0 and standard deviation ε_{obs}, and n is the dimension of the observation.

The majority of existing time-to-go estimation methods employ constant velocity and PNG guidance assumptions. However, the velocity is not constant during the terminal flight phase, and the guidance methods used do not necessarily adhere to the principles of PNG. Therefore, existing time-to-go estimation methods cannot be applied directly in vehicles with time-varying velocity. For this reason, we propose a straightforward but effective iterative approach to estimate the time-to-go. First, the initial guess for the remaining flight time estimate can be expressed as:

$$\hat{t}_{go} = \frac{\|\boldsymbol{r}_{TM}\|}{\|\dot{\boldsymbol{r}}_{TM}\|} \approx \frac{\|\boldsymbol{r}_{TM}\|}{V_M \cos\eta} \tag{23}$$

Here, we utilized Assumption 3, as $V_T \ll V_M$, and thus $V_{MT} = V_M - V_T \approx V_M$. Based on the assumption that the derivative of the vehicle's velocity with respect to time is constant, the predicted value of the terminal velocity can be calculated as follows:

$$V_f = V_M - \dot{V}_M \hat{t}_{go} \tag{24}$$

It is important to note that, since the deviations in aerodynamic parameters and atmospheric density cannot be determined in advance, it is not feasible to calculate $\dot{V}_M$ using Equation (7d) in practice. However, in real-world applications, $\dot{V}_M$ can be directly obtained from the output of an accelerometer.

Then, the average velocity over the entire flight process can be obtained using the following equation:

$$V_{ave} = \frac{V_M + V_f}{2} \tag{25}$$

Therefore, V_M in Equation (23) can be replaced with the average velocity V_{ave}, thereby obtaining a new estimate for the time-to-go, expressed as

$$\hat{t}_{go}^{new} = \frac{\|\boldsymbol{r}_{TM}\|}{V_{ave} \cos\eta} \tag{26}$$

In general, $\hat{t}_{go}^{new}$ is not directly equal to $\hat{t}_{go}$. However, after a few iterations of the above process, the estimate of time-to-go $\hat{t}_{go}$ converges quickly.

It is worth noting that in the above analysis, we implicitly used two assumptions: $\dot{V}_M$ is constant, and the leading angle η is also constant. These assumptions may introduce errors in the final estimate. However, the estimation error of the time-to-go will also gradually converge to zero as the distance between the UAV and the target continues to decrease. The time-to-go estimation approach is shown in Algorithm 1.

The proposed time-to-go estimation algorithm has the following advantages: First, it does not rely on any specific guidance law, making it applicable to a broader range of scenarios. In contrast, traditional analytical prediction formulas [8–10] can only predict the remaining flight time for trajectories derived from PNG. Second, compared to numerical integration methods, the proposed method benefits from its simple iterative process, offering superior real-time performance.

Algorithm 1 Improved time-to-go estimation algorithm

Input: Error tolerance ϵ_e, maximum number of iterations N_{iter}, distance $\|\boldsymbol{r}_{TM}\|$, leading angle η, velocity V_M and the rate of change in velocity $\dot{V}_M$
Output: Estimated value of time-to-go $\hat{t}_{go}$

Let $k = 0$, $\hat{t}_{go}^{old} = \frac{\|\boldsymbol{r}_{TM}\|}{V_M \cos\eta}$
while $|\hat{t}_{go}^{old} - \hat{t}_{go}^{new}| > \epsilon_e$ **do**
  Let $\hat{t}_{go}^{old} = \hat{t}_{go}^{new}$
  Estimated the final velocity of vehicle $V_f = V_M + \dot{V}_M \hat{t}_{go}^{old}$
  Obtain the mean velocity $V_{ave} = \frac{V_M + V_f}{2}$
  Obtain the new estimated value of time-to-go $\hat{t}_{go}^{new} = \frac{\|\boldsymbol{r}_{TM}\|}{V_{ave} \cos\eta}$
  $k = k + 1$
  if $k \geq N_{iter}$ **then**
    Break
  end if
end while
return The estimated value of time-to-go $\hat{t}_{go}^{new}$.

To avoid the excessive overload of the vehicle during flight, the action space was formed as an acceleration command, that is,

$$\boldsymbol{a}_t = \left[a_{yc}, a_{zc}\right]^T \tag{27}$$

where a_{yc} and a_{zc} are the required acceleration of the vehicle in the horizontal and vertical planes, respectively.

There are two principal reasons for adopting the acceleration command as the action. Firstly, the overload of the vehicle can be easily made to satisfy the constraints by applying a clip operation to the output of the policy $\pi(\boldsymbol{o}_t)$. Secondly, although taking the command angular rates $[\Delta\alpha, \Delta\sigma]$ as action is an end-to-end solution, which makes the problem formulation more concise, it may prove challenging in terms of achieving convergence on the policy of the agent.

Although a method for calculating the angle of attack and bank angle commands was proposed in ref. [35], applying that method implies that when the sign of the required lift changes, the sign of the bank angle must also change accordingly. When the sign of the bank angle change occurs for larger orders of magnitude of the bank angle, it leads to the occurrence of undesired transient processes. To circumvent this challenging problem, we adopted an unlimited bank angle scheme which was employed in [12] (as illustrated in Figure 4). Consequently, a simple and effective computational approach is given.

Based on the action of the agent, the commands α_c can be obtained by solving the following equation:

$$\alpha_c = \underset{\alpha_l \leq \alpha \leq \alpha_u}{\arg\min} |L - L_d| \tag{28}$$

where α_l and α_u represent the minimum and maximum values of the vehicle's angle of attack, respectively. L_d denotes the required lift force, whose expressions are given by the following equations:

$$L_d = m\sqrt{(a_{yc} + g\cos\gamma)^2 + a_{zc}^2} \tag{29}$$

Equation (28) is essentially a single-variable root-finding problem. In this paper, the Newton–Raphson method was employed to solve Equation (28). However, alternative methods such as the secant method or Brent's method can also achieve rapid solutions for Equation (28).

Then, the commanded bank angle σ_c can be obtained by the following equation with the unlimited bank angle control scheme:

$$\sigma_c = \arctan 2(a_{zc}, a_{yc} + g\cos\gamma) \tag{30}$$

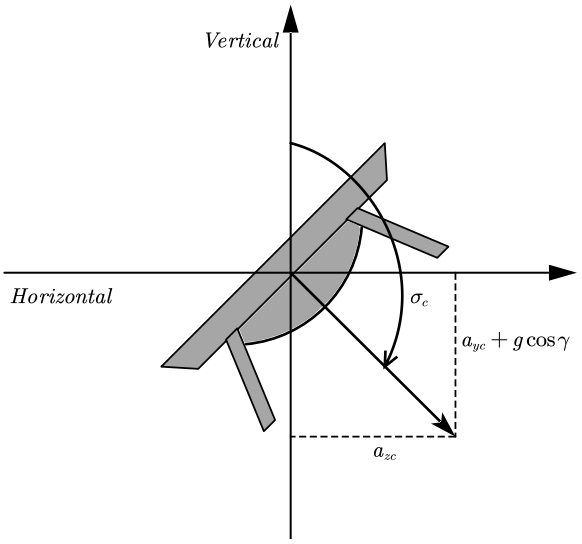


Figure 4. Unlimited bank angle control scheme [12].

Finally, the commanded angular rate can be obtained by the following equations

$$\Delta\alpha_c = \text{clip}\left(\frac{\alpha_c - \alpha_t}{\Delta t}, -\Delta\alpha_{max}, \Delta\alpha_{max}\right) \tag{31a}$$

$$\Delta\sigma_c = \text{clip}\left(\frac{\sigma_c - \sigma_t}{\Delta t}, -\Delta\sigma_{max}, \Delta\sigma_{max}\right) \tag{31b}$$

where Δt is the guidance period.

The adoption of an unlimited bank angle control scheme is motivated by two primary reasons: First, to satisfy the requirement for dive attacks, the high-speed UAV must generate a negative lifting force. If the bank angle is restricted to the range $[-\pi/2, \pi/2]$, the negative lift force can only be achieved through the negative angles of attack. However, for non-axisymmetric high-speed UAVs, the aerodynamic effects of negative and positive angles of attack often differ significantly, which increases control complexity. Second, when the high-speed UAV approaches the target, oscillations in guidance commands inevitably occur. When the sign of the required lateral acceleration a_{zc} changes, the unlimited bank angle control scheme allows the bank angle to adjust by a small magnitude. In contrast, if the bank angle is restricted to $[-\pi/2, \pi/2]$, a sign change in a_{zc} would require large variations in the bank angle, severely compromising the flight stability of high-speed UAVs.

4.2. Reward Function and Termination Conditions

The most significant challenge in using DRL to address the UAV with time-varying velocity guidance problem is developing an effective reward function in a sparse reward environment. If the reward signal is given only if the vehicle arrives at the target and satisfies the process and terminal constraints, it is probable that, with a limited number of episodes, the policy of the agent may encounter difficulty in identifying and exploiting positive samples.

The potential-based reward shaping (PBRS) method was proposed in [36] to avoid the challenge of a sparse reward environment. Inspired by PBRS, a compound reward function was designed that provided cues to the agent within each time step, thereby motivating the

vehicle to reach the target. The shaping reward was designed for keeping the LOS angular rate of the vehicle as small as possible, that is,

$$r_{miss} = \beta_{miss} \exp\left(-\frac{\|\mathbf{\Omega}\|^2}{\kappa_\Omega^2}\right) \tag{32}$$

where $\mathbf{\Omega} = \left[\dot{\theta}_L, \dot{\psi}_L\right]^T$, β_{miss} is a positive constant, and $\kappa_\Omega > 0$ is the LOS angular rate's scaling factor.

Then, we also need to keep the arrival time error within tolerance. The reward signal on arrival time error is

$$r_{time} = \beta_{time} \exp\left(-\frac{|\delta_t|^2}{\kappa_t^2}\right) \tag{33}$$

where β_{time} is a positive constant, and $\kappa_t > 0$ is the arrival time error's scaling factor.

Meanwhile, in order to achieve a dive attack, an overly smooth path angle is undesirable behavior. Therefore, a penalty signal must be applied to the dive angle error, that is,

$$r_{dive} = -\beta_{dive} \left|\frac{\delta_\gamma}{\kappa_\gamma}\right|^2 \tag{34}$$

where β_{dive} is a positive constant, and κ_γ is the dive angle error's scaling factor.

From Equations (28) and (29), it can be derived that the required lift L_d and the commanded acceleration $\boldsymbol{a}_c = [a_{yc}, a_{zc}]^T$ are positively correlated. This implies that as $\boldsymbol{a}_c$ increases, L_d also increases, resulting in a higher commanded angle of attack α_c. Consequently, the drag coefficient C_D increases, leading to greater energy loss. Therefore, to minimize the energy consumption of the vehicle guidance policy, we introduced a negative reward signal to penalize excessive vehicle commanded acceleration, that is,

$$r_e = -\beta_e \|\boldsymbol{a}_t\|^2 \tag{35}$$

where $\boldsymbol{a}_t$ is the output of the policy.

Finally, for terminal conditions that satisfy the constraints, it is necessary to provide an appropriate bonus signal:

$$r_{bonus} = \begin{cases} \xi + \frac{V_M(t_f)}{\kappa_{V_M}}, & \text{if } \|\boldsymbol{r}_{TM}\| \le \epsilon_r \text{ and } \delta_t \le \epsilon_{\delta_t} \\ & \text{and } \gamma < \gamma_{dive} \text{ and done} \\ 0 & \text{else} \end{cases} \tag{36}$$

where ξ is a positive constant, and κ_{V_M} is the impact velocity scaling factor.

Combining Equations (32), (33), (35), and (36), the reward signal for each time step can be obtained as follows:

$$r_t = r_{miss} + r_{time} + r_{dive} + r_e + r_{bonus} \tag{37}$$

Equation (7a–h) was solved by employing the fourth-order Adams prediction-correction numerical integration method with a 0.1 s time step, and the initial four steps were integrated using the Runge–Kutta method. In addition, to ensure sufficient accuracy, the time step was reduced by a factor of 100 when the distance was less than 800 m.

Given the mission profile and operational constraints, the terminal criteria were formally established as follows:

1. The height of the vehicle is less than zero;
2. The FOV constraints or the overload constraints are violated;

3. The vehicle is flying in a direction that is away from the target;
4. The vehicle has arrived at the target successfully.

It is of significant importance to note that the current episode will be terminated if the constraints are violated. In such an instance, the agent will not receive a terminal reward. This incentive is sufficient to encourage the agent to learn to satisfy the constraints; therefore, no negative reward is required when the constraint is violated.

4.3. Policy Optimization

In light of the satisfactory performance of GRU in long-term historical memory, a GRU layer was introduced into the actor network and the critic network with the objective of enhancing the generalization ability of the policy under different task conditions. This integration enabled the policy to adapt to varying tasks through the updating of the hidden states of the GRUs.

It should be noted that when GRU layers are introduced into policy and value networks, the way to update the hidden state of the GRU layer during the rollout collection process has a significant impact on the training effect. Inspired by [33,37], the hidden state was reset before the start of each rollout. However, the hidden state of each episode in the same rollout was inherited from the previous episode. The interaction protocol between the learning agent and its operational environment is illustrated in Figure 5. It should be noted that the length of each episode may vary, although this is not explicitly shown in Figure 5. Through this design, the aim was to encourage the agent to update the parameters of the GRU layer so that updates to its hidden state could be used to describe an embedded representation of the current environment or task. This provided the agent with multi-level input features for rapid adaptation to different environments and tasks, thereby enhancing the adaptability and generalization performance of the final network.

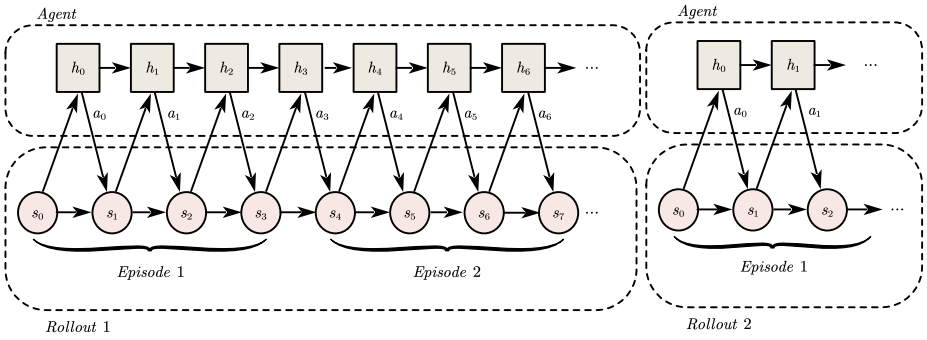


Figure 5. Procedure of agent–environment interaction.

The PPO algorithm has two versions of the objective function. The first is the clipped surrogate objective, whose expression is shown in Equation (2). The second involves incorporating the KL divergence into the objective function as a penalty, and the penalty coefficient is adaptively adjusted so that the KL divergence can reach a specific target value d_{targ} during each policy update.

In general, it is more straightforward to achieve convergence when utilizing the clipped surrogate objective. However, this approach inevitably requires a longer training process. In contrast, it is challenging to achieve convergence when employing the adaptive KL penalty coefficient. Therefore, in order to strike a balance between the speed and stability of the training process, the two aforementioned methods were combined in a novel

manner, whereby the clipped factor ϵ and the learning rate α_θ of the policy network were adaptively adjusted in order to achieve the target value of the KL divergence d_{targ}, that is,

$$\epsilon = \begin{cases} \min(\epsilon_{max}, 1.5\epsilon) & \text{if } d < \frac{1}{2} d_{targ} \\ \max(\epsilon_{min}, 1.5\epsilon) & \text{if } d > 2 d_{targ} \\ \epsilon & \text{others} \end{cases} \tag{38}$$

$$\alpha_\theta = \begin{cases} 1.5\alpha_\theta & \text{if } d < \frac{1}{2} d_{targ} \text{ and } \epsilon > \frac{1}{2}\epsilon_{max} \text{ and } \alpha_\theta < \alpha_\theta^{max} \\ \frac{1}{1.5}\alpha_\theta & \text{if } d > 2 d_{targ} \text{ and } \epsilon < 2\epsilon_{min} \text{ and } \alpha_\theta > \alpha_\theta^{min} \\ \alpha_\theta & \text{others} \end{cases} \tag{39}$$

In the training phase, the experience data were fully utilized through the conduct of E update epochs, The training process involved a range of generic operations, including ratio calculation, the computation of various losses, and gradient clipping. The adaptive PPO (APPO) algorithm is shown in Algorithm 2.

Algorithm 2 Adaptive PPO (APPO) algorithm

Input: Target value of the KL divergence d_{targ}, number of epochs M
Output: The optimized parameters of actor θ, and critic w

Initialize network parameters, including the actor θ, and critic w
for epoch = 1 : M **do**
  Initialize replay buffer R and reset the hidden states of the RNNs
  for episodes = 1 : N **do**
    Reset the environment;
    while not done **do**
      Obtain the current observation s_t
      The observation s_t is fed to the actor to obtain the action a_t
      Execute one step in the environment
      Store $(o_t, a_t, r_t, done_t)$ into R
    end while
  end for
  Calculate discounted returns G and advantages $\hat{A}$
  for opt = 1 : E **do**
    Calculate policy network loss, value network loss, and KL divergence
    Update θ, w
    Adjust the clipped factor ϵ and the learning rate α_θ
  end for
end for
return The optimized parameters θ, w

5. Simulation Results

5.1. Training Scenarios

The parameters of the vehicle referenced in [12] were adopted; the full parameters of the vehicles are given in Table 1, and the detail expressions of the aerodynamic coefficients C_L and C_D can be found in ref. [12].

In order to ensure that the agent exhibits good generalization ability, it was trained on a range of initial conditions. The details of the initial conditions are listed in Table 2.

Table 1. Vehicle parameters.

Parameter	Value	Parameter	Value
Mass m (kg)	1600	Reference area S_{ref} (m^2)	0.5026
Guidance period (s)	0.5	FOV angle limit (degrees)	60
Angle of attack α (degrees)	$[0, 28]$	Bank angle σ (degrees)	$[-180, 180]$
Angle of attack rate $\Delta\alpha$ (degrees/s)	$[-5, 5]$	Bank angle rate $\Delta\sigma$ (degrees/s)	$[-40, 40]$
Observation standard deviation ϵ_{obs}	0.003	Aerodynamic lift coefficient of variation (%)	$[-5, 5]$
Aerodynamic drag coefficient of variation (%)	$[-5, 5]$	Atmospheric density variation (%)	$[-5, 5]$

Table 2. Initial conditions.

Parameter	Value	Parameter	Value
Range R_{init} (km)	[120, 120]	Altitude h_{init} (km)	[28.8, 29.2]
Velocity V_{init} (m/s)	[2060, 2120]	Flight path angle γ_{init} (degrees)	$[-5, 0]$
Azimuth ψ_{init} (degrees)	$[-10, 10]$	Heading error $\delta\psi_{init}$ (degrees)	$[-8, 8]$
Angle of attack α_{init} (degrees)	[1, 3]	Bank angle σ_{init} (degrees)	$[-2, 2]$
Desired impact time T_d (s)	[75, 80]	Target maximum speed $V_{T_{max}}$ (m/s)	15.0
Target maximum acceleration $a_{T_{max}}$ (m/s^2)	0.5		

It should be noted that four-layer network structures are frequently employed in DRL applications [38]. Accordingly, a four-layer neural network was employed to implement the policy and value functions, where the first hidden layer of the policy and value functions was a GRU [39] layer. The network architectures are shown in Table 3.

Table 3. Network layer size.

	Policy Network		Value Network	
Layer	**Size**	**Activation**	**Size**	**Activation**
Input layer	(10, 100)	tanh	(10, 100)	tanh
GRU layer	(100, 44)	tanh	(100, 22)	tanh
Hidden layer 2	(44, 20)	tanh	(22, 5)	tanh
Output layer	(20, 2)	linear	(5, 1)	linear

5.2. Training Results

Tables 4 and 5 provide a detailed list of the hyperparameter values used during the training process. In practice, the selection of appropriate hyperparameters is often a crucial determinant of the performance of APPO. Different problems may require the use of different sets of hyperparameters. Accordingly, these hyperparameters were tuned for the guiding problem based on a series of trial-and-error tests.

Table 4. Hyperparameters in the reward function.

Parameter	Value	Parameter	Value	Parameter	Value	Parameter	Value
β_{miss}	0.2	β_{time}	0.05	β_{dive}	0.01	β_e	0.01
ξ	10	κ_Ω (rad/s)	0.1	κ_t (s)	0.5	κ_γ (rad)	$\delta_\gamma(t_0)/2$
κ_{V_M} (m/s)	200	ϵ_r (m)	10	ϵ_t (s)	0.5	γ_{dive} (degrees)	75

Table 5. Hyperparameter settings.

Parameter	Value	Parameter	Value
Maximum episode steps	800	Maximum epochs M	2500
Collected episodes per epoch N	60	Update epochs E	20
Discounting factor,γ	0.95	Value network learning rate α_w	2.0×10^{-3}
Maximum learning rate α_θ^{max}	1.5×10^{-3}	Minimum learning rate α_θ^{min}	1.5×10^{-5}
Maximum clipped factor ϵ_{max}	0.5	Minimum clipped factor ϵ_{min}	0.01
Target KL divergence d_{targ}	0.001		

For comparison, simulations were also performed that replaced the GRU layer with an MLP layer. For the sake of simplicity, the APPO algorithm with GRU is henceforth referred to as APPO-GRU, and the APPO algorithm employing an MLP layer is referred to as APPO-MLP.

The results of comparing the learning curves are shown in Figure 6. The figure clearly demonstrates that the introduction of the GRU layer resulted in a significant increase in the average steady-state reward value. This also validated the previous analysis in this paper, which indicated that using the hidden state of the GRU layer as a deep embedded representation of environmental features could effectively enhance the adaptability and generalization performance of the agent.

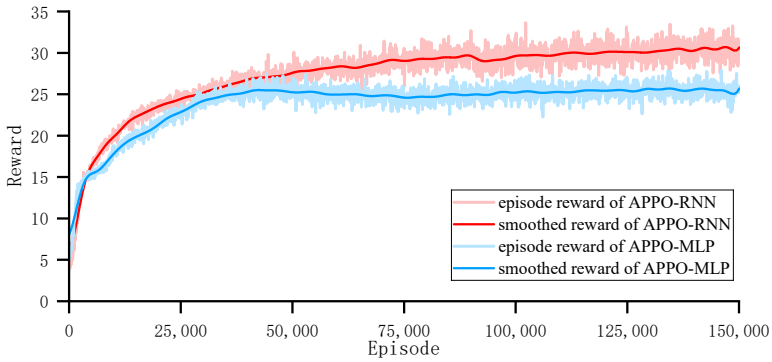


Figure 6. Comparison of learning curves.

The final impact-time error change process during the training process is illustrated in Figure 7. As can be observed from Figure 7, the arrival-time error of APPO-GRU eventually converged to around 0.2 s, while the arrival-time error of APPO-MLP converged to approximately 0.7 s. This also demonstrated that the incorporation of the GRU layer could effectively enhance the guidance performance of the agent.

The incorporation of a GRU layer contributed to the agent's performance through two primary mechanisms. Primarily, the hidden state of the GRU layer enabled the agent to delineate the characteristics of various task scenarios, thereby augmenting its capacity for generalization and robustness. Secondly, the hidden state could retain the historical information of the trajectory, endowing the guidance policy with the capability to draw insights from the historical state.

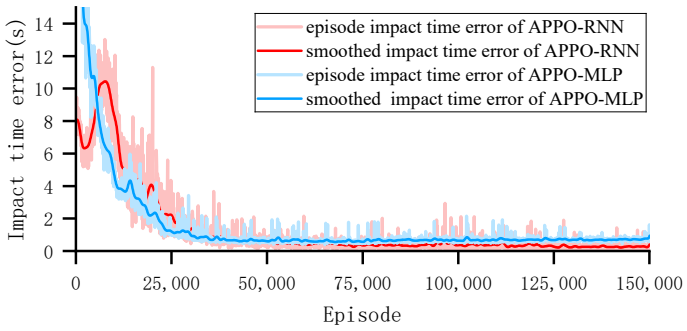


Figure 7. Comparison of $\delta_t(t_f)$ curves.

Furthermore, Figure 8 illustrates the curves of the KL divergence of the policy throughout the training process. This figure demonstrates that the proposed adaptive PPO algorithm could maintain the KL divergence of the policy at a level closely aligned with the target KL divergence, thereby ensuring that the training process remained stable and ultimately converged.

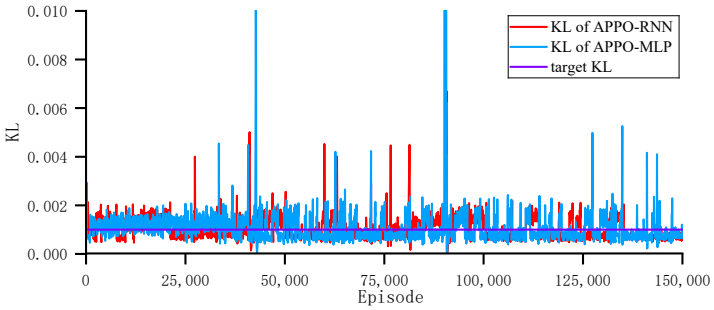


Figure 8. Comparison of KL divergence values.

5.3. Test Results

To verify the impact-time-control capability of the proposed guidance policy, simulations were performed on scenarios with the same initial state but different preset impact times. The initial state settings are shown in Table 6 and the preset impact times were 76, 78, and 80 s, respectively. For comparison, the PNG law was also simulated. The simulation results for different desired arrival times are presented in Figures 9 and 10. In addition, the final miss distance, flight path angle, and arrival-time error results are listed in Table 7.

Table 6. Initial state of vehicles.

Parameter	Value	Parameter	Value
Range (km)	120	Altitude (km)	29
Velocity (m/s)	2100	Flight path angle (degrees)	−2
Azimuth (degrees)	5	Heading error (degrees)	4
Angle of attack (degrees)	1.5	Bank angle (degrees)	0
Desired impact time (s)	76, 78, 80		

As illustrated in Figure 9, the proposed ITGC law primarily controlled the arrival time by controlling the vehicle to turn in the horizontal plane. In other words, different specified arrival times were achieved by controlling the magnitude of lateral maneuvers; the larger the specified arrival time, the greater the amplitude of the turning maneuvers in the lateral plane. At the same time, it can be observed from Table 7 that the impact time errors of the proposed ITCG law were all less than 0.5 s. Furthermore, Figure 10f illustrates the FOV angle curves for different specified arrival times; it can be seen that the proposed ITGC

law was capable of maintaining the FOV constraint at all times, thereby providing further evidence of the efficiency of the proposed ITCG law.

Table 7. Final state for different impact times.

Scenarios	Miss Distance (m)	Impact Time (s)	Velocity (m/s)	Impact Angle (deg)
$T_d = 76$ s	7.146	76.367	991.347	−79.766
$T_d = 78$ s	8.453	77.795	1003.193	−82.438
$T_d = 80$ s	9.022	79.674	975.535	−83.945
PNG	2.334	79.853	559.952	−18.650

It should be noted that while the PNG law could also meet FOV constraints, it was not capable of controlling impact time or diving attack. Moreover, as can be found in Figure 10b, the terminal velocity of the vehicle guided by the PNG law was around 600 m/s, while the terminal velocities of the proposed guidance law were around 1000 m/s. Therefore, the proposed ITCG law also demonstrated significant superiority over PNG in terms of optimizing performance metrics.

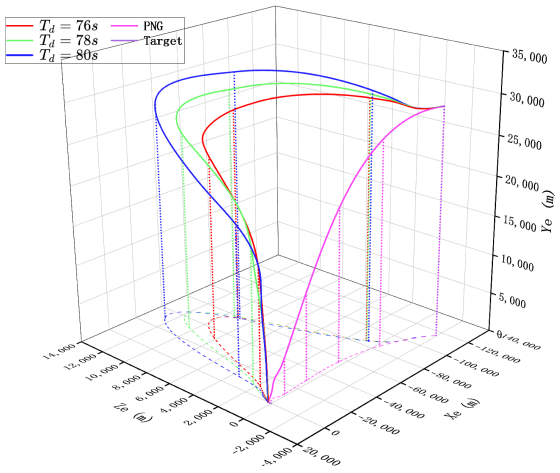
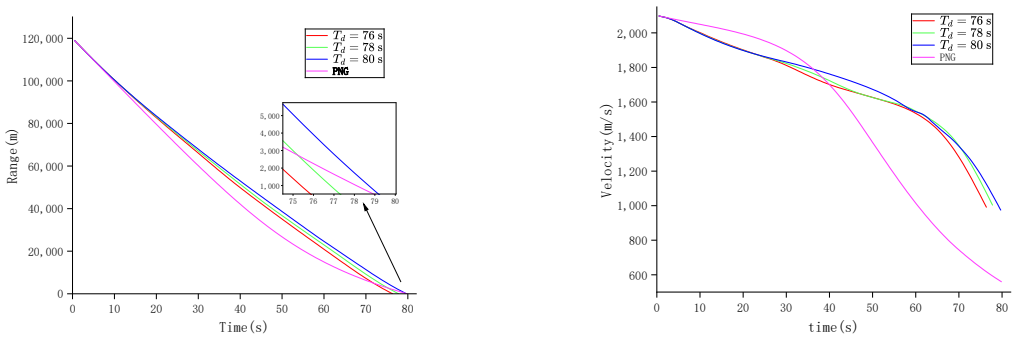
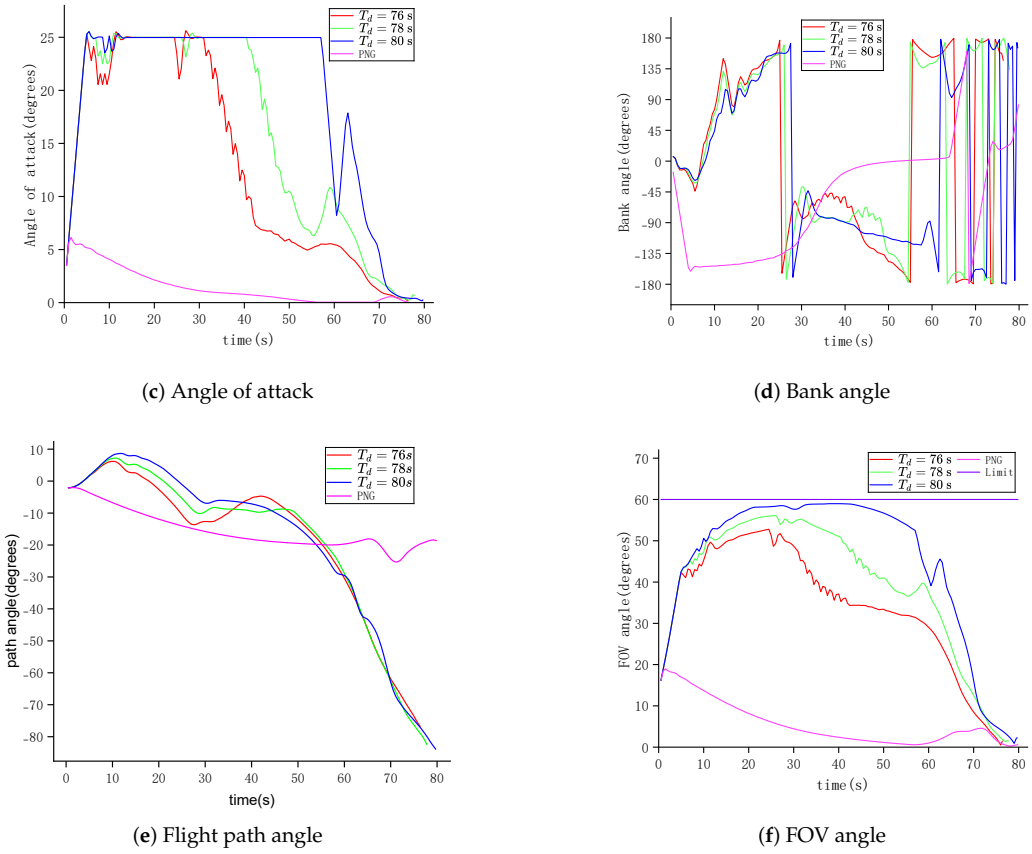


Figure 9. Guidance trajectories for different impact times.



(**a**) Range (**b**) Velocity

Figure 10. *Cont.*



(**c**) Angle of attack (**d**) Bank angle

(**e**) Flight path angle (**f**) FOV angle

Figure 10. Results for different specified arrival times.

It should be noted that, as shown in Figure 10d, when the ITCG law proposed in this paper was employed, the bank angle frequently alternated between $-\pi$ and π. However, this did not violate the bank angle rate constraint specified in Equation (31b), as the bank angle adjustments were controlled through the minimum rotation angle (as illustrated in Figure 11).

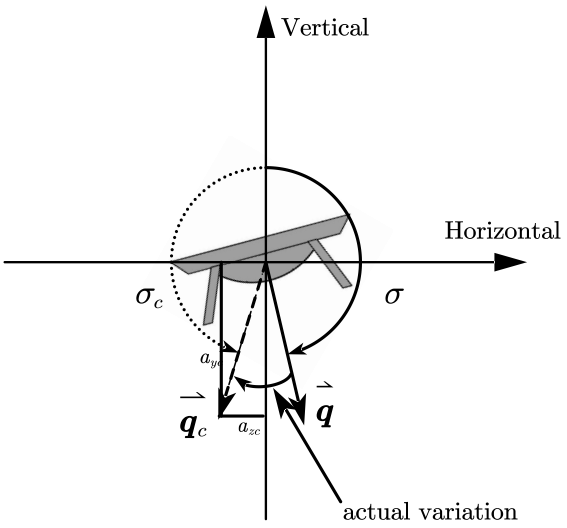


Figure 11. Schematic diagram for the calculation of the rate of change in the bank angle.

According to Figure 11, we can define the corresponding vectors for σ and σ_c as follows:

$$\boldsymbol{q} = [\cos\sigma, \sin\sigma, 0]^T \tag{40a}$$

$$\boldsymbol{q}_c = [\cos\sigma_c, \sin\sigma_c, 0]^T \tag{40b}$$

Subsequently, the cross-product of $\boldsymbol{q}$ and $\boldsymbol{q}_c$ can be calculated using the following equation:

$$\boldsymbol{q} \times \boldsymbol{q}_c = [0, 0, \sin\sigma_c \cos\sigma - \cos\sigma_c \sin\sigma]^T = [0, 0, \sin(\sigma_c - \sigma)]^T \tag{41}$$

Therefore, according to Figure 11, when $\sin(\sigma_c - \sigma) < 0$, the bank angle σ rotates clockwise towards the commanded bank angle σ_c, whereas when $\sin(\sigma_c - \sigma) > 0$, σ rotates counterclockwise towards σ_c.

Finally, Equation (31b) can be rewritten as follows:

$$\Delta\sigma_c = \begin{cases} \text{clip}\left(\frac{\sigma_c - \sigma + 2\pi}{\Delta t}, -\Delta\sigma_{max}, \Delta\sigma_{max}\right), & \text{if } \sin(\sigma_c - \sigma) < 0 \\ \text{clip}\left(\frac{\sigma - \sigma_c - 2\pi}{\Delta t}, -\Delta\sigma_{max}, \Delta\sigma_{max}\right), & \text{if } \sin(\sigma_c - \sigma) > 0 \end{cases} \tag{42}$$

Taking the flight trajectory with $T_d = 76$ s as an example, in Figure 10d, the bank angle of the UAV changed from -177.28 deg to 177.76 deg at $t = 5$ s. However, the actual change in the bank angle was only -4.96 deg During the guidance cycle, the corresponding bank angle rate of change was -4.96 deg/s, which was significantly lower than the bank angle rate constraint specified in Table 1.

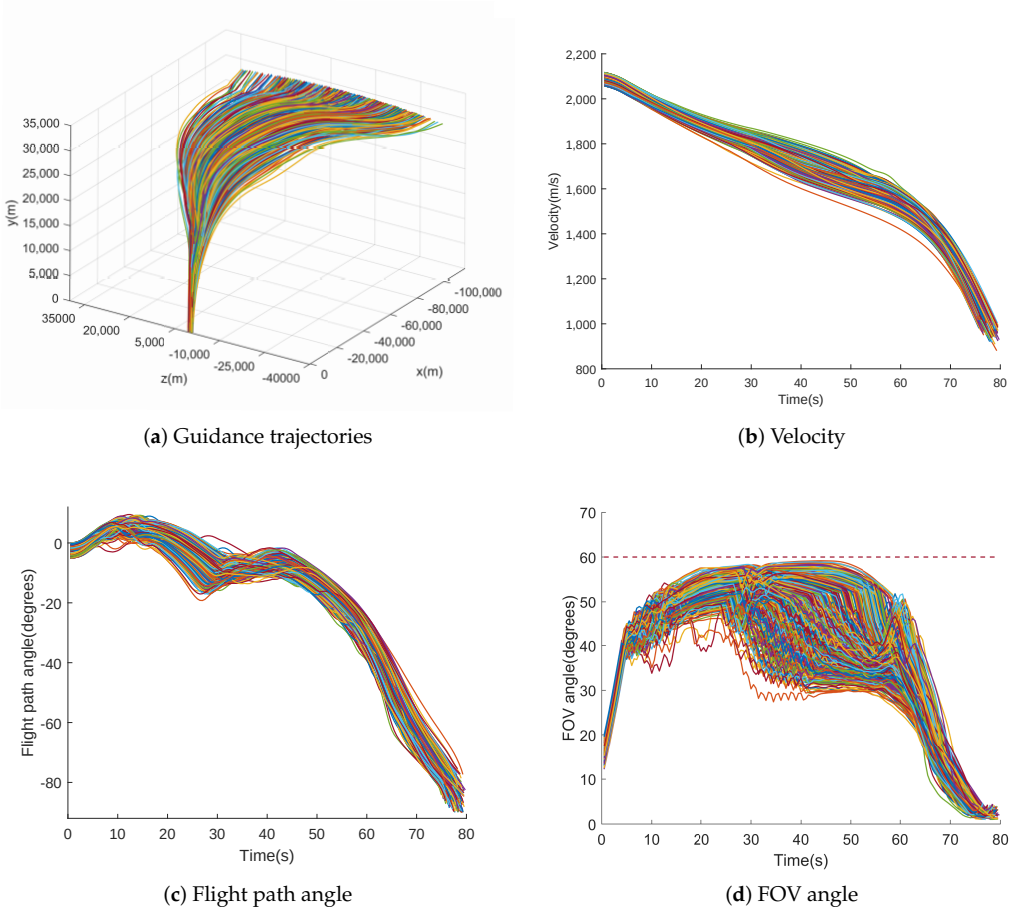
5.4. Monte Carlo Simulations

Monte Carlo simulations are performed to assess the adaptability and robustness of the proposed ITCG law, considering factors such as aerodynamic disturbances and measurement noise. The initial state variables of the vehicle were stochastically generated based on the parameter ranges outlined in Table 2, while the environmental uncertainties are defined in Table 1. The results of 500 independent Monte Carlo trials are summarized in Figures 12–14.

As illustrated in Figure 12b, the terminal velocity of the vehicle was approximately 1000 m/s, its velocity loss was significantly less than that of the PNG law. The results presented in Figure 12c shows that the final flight path angles were all around $-80°$, ensuring compliance with the dive-attack specifications. Furthermore, Figure 12d illustrates that the FOV constraints were also satisfied. The results indicate that the proposed guidance law exhibited exceptional robustness, even when subjected to aerodynamic deviations and sensor noise.

To gain further insight into guidance performance, a series of key variables were subjected to statistical analysis, the results of which are presented in Table 8. Figure 13 illustrates the distribution of the miss distance in the 2D plane, and Figure 14 presents the statistical histograms that include the miss distance, impact time error, terminal flight path angle, and terminal velocity.

Figure 13 depicts the impact point distribution results from Monte Carlo simulations. As illustrated in Figure 13, despite the presence of aerodynamic parameter deviations and observation noise, the miss distances of the proposed guidance policy remained within 10 m, with impact time errors below 0.5 s. The results demonstrate that the proposed ITCG law displayed robust performance. Furthermore, Table 8 indicates that the minimum impact velocity was 880.43 m/s, which was considerably higher than the impact velocity of the PNG law.



(**a**) Guidance trajectories (**b**) Velocity

(**c**) Flight path angle (**d**) FOV angle

Figure 12. Monte Carlo simulation results.

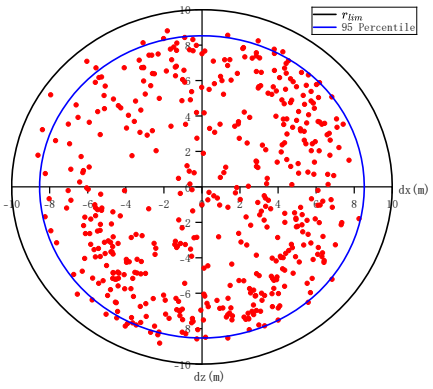
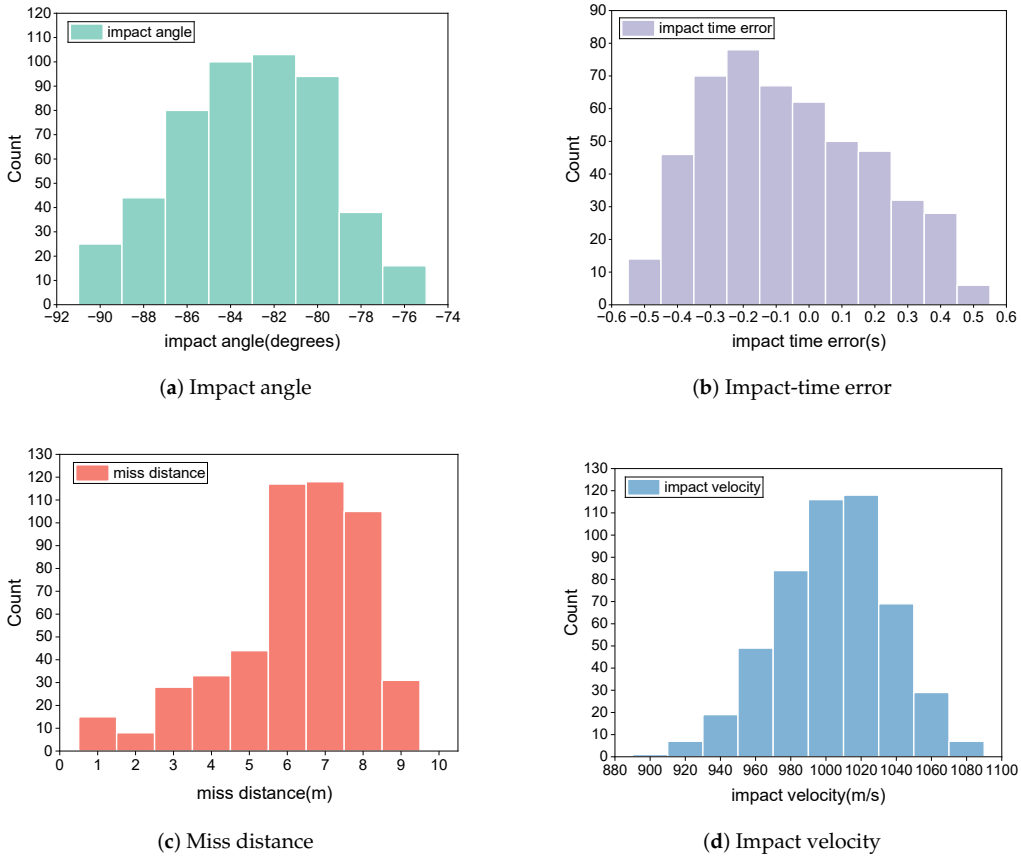


Figure 13. Miss distance statistics of Monte Carlo simulation.

Table 8. The statistical results of the Monte Carlo experiment.

Parameters	Mean	Min	Max
Impact angle (degrees)	−83.137	−89.999	−75.357
Impact-time error (s)	−0.0601	−0.4986	0.4990
Miss distance (m)	6.2235	0.4892	9.4947
Impact velocity	1003.53	880.43	1082.00



(**a**) Impact angle (**b**) Impact-time error

(**c**) Miss distance (**d**) Impact velocity

Figure 14. Monte Carlo simulation results.

6. Conclusions

This study proposed a DRL-based ITCG law for a high-speed UAV with time-varying velocity, enabling impact-time control and dive attack while maintaining FOV constraints. Firstly, a reinforcement learning framework was designed for high-speed UAV guidance with time-varying velocity, in which the guidance system mapped observations, including time-to-go estimation, to the commanded acceleration of the vehicle. Secondly, a multi-objective reward architecture was developed to jointly optimize both time-constrained impact and dive attack. Furthermore, in order to improve the adaptability of the guidance law, environmental uncertainties were considered during the training process. In particular, GRU layers were introduced into the policy network and value network to make the policy adaptive to different task scenarios. Finally, simulations were conducted in various challenging scenarios, and the results illustrated that the proposed ITCG law exhibited good performance even in the presence of aerodynamic parameter deviation and environmental uncertainties.

Future research will prioritize three key directions. First, the development of an integrated guidance framework incorporating simultaneous impact-time and terminal-angle control is expected to significantly enhance high-speed UAVs' operational capabilities in executing complex mission profiles; second, it would be beneficial to consider extending this method to integrated guidance-and-control frame; third, based on the establishment of communication links among UAV swarms, multi-agent reinforcement learning (MARL) techniques could be utilized to construct a distributed cooperative guidance decision-making method, enabling the realization of collective intelligence in a true sense.

Author Contributions: Conceptualization, G.L.; formal analysis, Z.L.; methodology, Z.L. and Y.X.; resources, G.L., Y.X. and S.L.; validation, L.R.; writing—original draft, Z.L.; writing—review and editing, L.R., S.L. and D.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The original contributions presented in the study are included in the article, further inquiries can be directed to the corresponding author.

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

Abbreviations

FOV	Field-of-view
ITCG	Impact-time-control guidance
PNG	Proportional navigation guidance
3D	Three-dimensional
MPSP	Model predictive static programming
MPCP	Model predictive convex programming
DRL	Deep reinforcement learning
RL	Reinforcement learning
DDPG	Deep deterministic policy gradient
PPO	Proximal policy optimization
RNN	Recurrent neural network
GRU	Gated recurrent unit
IMU	Inertial measurement unit
PBRS	Potential-based reward shaping
MLP	Multi-layer perceptron

References

1. Lee, C.H.; Seo, M.G. New Insights into Guidance Laws with Terminal Angle Constraints. *J. Guid. Control. Dyn.* **2018**, *41*, 1832–1837. [CrossRef]
2. Park, B.G.; Kim, T.H.; Tahk, M.J. Range-to-go weighted optimal guidance with impact angle constraint and seeker's look angle limits. *IEEE Trans. Aerosp. Electron. Syst.* **2016**, *52*, 1241–1256. [CrossRef]
3. Ren, L.; Guo, W.; Xian, Y.; Liu, Z.; Zhang, D.; Li, S. Deep reinforcement learning based integrated evasion and impact hierarchical intelligent policy of exo-atmospheric vehicles. *Chin. J. Aeronaut.* **2025**, *38*, 103193. [CrossRef]
4. Jeon, I.S.; Lee, J.I.; Tahk, M.J. Impact-Time-Control Guidance with Generalized Proportional Navigation Based on Nonlinear Formulation. *J. Guid. Control. Dyn.* **2016**, *39*, 1885–1890. [CrossRef]
5. Livermore, R.; Shima, T. Deviated Pure-Pursuit-Based Optimal Guidance Law for Imposing Intercept Time and Angle. *J. Guid. Control. Dyn.* **2018**, *41*, 1807–1814. [CrossRef]
6. Saleem, A.; Ratnoo, A. Lyapunov-Based Guidance Law for Impact Time Control and Simultaneous Arrival. *J. Guid. Control. Dyn.* **2016**, *39*, 164–173. [CrossRef]
7. Kim, M.; Jung, B.; Han, B.; Lee, S.; Kim, Y. Lyapunov-based impact time control guidance laws against stationary targets. *IEEE Trans. Aerosp. Electron. Syst.* **2015**, *51*, 1111–1122. [CrossRef]
8. Hou, Z.; Yang, Y.; Liu, L.; Wang, Y. Terminal sliding mode control based impact time and angle constrained guidance. *Aerosp. Sci. Technol.* **2019**, *93*, 105142. [CrossRef]
9. Kim, H.G.; Cho, D.; Kim, H.J. Sliding Mode Guidance Law for Impact Time Control Without Explicit Time-to-Go Estimation. *IEEE Trans. Aerosp. Electron. Syst.* **2019**, *55*, 236–250. [CrossRef]
10. Yang, Z.; Wang, H.; Lin, D.; Zang, L. A New Impact Time and Angle Control Guidance Law for Stationary and Nonmaneuvering Targets. *Int. J. Aerosp. Eng.* **2016**, *2016*, 6136178. [CrossRef]
11. Ye, Q.; Liu, C.; Sun, J. A Backstepping-Based Guidance Law for an Exoatmospheric Missile with Impact Angle Constraint. *IEEE Trans. Aerosp. Electron. Syst.* **2019**, *55*, 547–561. [CrossRef]
12. Wang, J.; Zhang, R. Terminal Guidance for a Hypersonic Vehicle with Impact Time Control. *J. Guid. Control. Dyn.* **2018**, *41*, 1790–1798. [CrossRef]
13. Xian, Y.; Ren, L.; Xu, Y.; Li, S.; Wu, W.; Zhang, D. Impact point prediction guidance of ballistic missile in high maneuver penetration condition. *Def. Technol.* **2023**, *26*, 213–230. [CrossRef]

14. Erer, K.S.; Merttopçuoglu, O. Indirect Impact-Angle-Control Against Stationary Targets Using Biased Pure Proportional Navigation. *J. Guid. Control. Dyn.* **2012**, *35*, 700–704. [CrossRef]
15. Park, B.G.; Kim, T.H.; Tahk, M.J. Biased PNG With Terminal-Angle Constraint for Intercepting Nonmaneuvering Targets Under Physical Constraints. *IEEE Trans. Aerosp. Electron. Syst.* **2017**, *53*, 1562–1572. [CrossRef]
16. Ratnoo, A. Analysis of Two-Stage Proportional Navigation with Heading Constraints. *J. Guid. Control. Dyn.* **2016**, *39*, 156–164. [CrossRef]
17. Sharma, Y.R.; Ratnoo, A. A Bearings-Only Trajectory Shaping Guidance Law With Look-Angle Constraint. *IEEE Trans. Aerosp. Electron. Syst.* **2019**, *55*, 3303–3315. [CrossRef]
18. Kim, H.G.; Lee, J.Y.; Kim, H.J.; Kwon, H.H.; Park, J.S. Look-Angle-Shaping Guidance Law for Impact Angle and Time Control with Field-of-View Constraint. *IEEE Trans. Aerosp. Electron. Syst.* **2020**, *56*, 1602–1612. [CrossRef]
19. Hu, Q.; Cao, R.; Han, T.; Xin, M. Field-of-view limited guidance with impact angle constraint and feasibility analysis. *Aerosp. Sci. Technol.* **2021**, *114*, 106753. [CrossRef]
20. He, S.; Lee, C.H.; Shin, H.S.; Tsourdos, A. Optimal three-dimensional impact time guidance with seeker's field-of-view constraint. *Chin. J. Aeronaut.* **2021**, *34*, 240–251. [CrossRef]
21. Wang, X.; Zhang, Y.; Wu, H. Sliding mode control based impact angle control guidance considering the seeker's field-of-view constraint. *ISA Trans.* **2016**, *61*, 49–59. [CrossRef]
22. Kim, H.G.; Lee, J.Y.; Kim, H.J. Look Angle Constrained Impact Angle Control Guidance Law for Homing Missiles With Bearings-Only Measurements. *IEEE Trans. Aerosp. Electron. Syst.* **2018**, *54*, 3096–3107. [CrossRef]
23. Li, A.; Hu, X.; Yang, S.; Dong, K. Adaptive Terminal Time and Impact Angle Constraint Cooperative Guidance Strategy for Multiple Vehicles. *Drones* **2024**, *8*, 134. [CrossRef]
24. Hong, H.; Maity, A.; Holzapfel, F.; Tang, S. Model Predictive Convex Programming for Constrained Vehicle Guidance. *IEEE Trans. Aerosp. Electron. Syst.* **2019**, *55*, 2487–2500. [CrossRef]
25. Pan, B.; Ma, Y.; Yan, R. Newton-Type Methods in Computational Guidance. *J. Guid. Control. Dyn.* **2019**, *42*, 377–383. [CrossRef]
26. Piccinin, M.; Lavagna, M.R. Deep Reinforcement Learning approach for Small Bodies Shape Reconstruction Enhancement. In Proceedings of the AIAA Scitech 2020 Forum, Orlando, FL, USA, 6–10 January 2020. [CrossRef]
27. He, S.; Shin, H.S.; Tsourdos, A. Computational Missile Guidance: A Deep Reinforcement Learning Approach. *J. Aerosp. Inf. Syst.* **2021**, *18*, 571–582. [CrossRef]
28. Wang, N.; Wang, X.; Cui, N.; Li, Y.; Liu, B. Deep reinforcement learning-based impact time control guidance law with constraints on the field-of-view. *Aerosp. Sci. Technol.* **2022**, *128*, 107765. [CrossRef]
29. Lee, S.; Lee, Y.; Kim, Y.; Han, Y.; Kwon, H.; Hong, D. Impact Angle Control Guidance Considering Seeker's Field-of-View Limit Based on Reinforcement Learning. *J. Guid. Control. Dyn.* **2023**, *46*, 2168–2182. [CrossRef]
30. Gaudet, B.; Furfaro, R. Terminal Adaptive Guidance for Autonomous Hypersonic Strike Weapons via Reinforcement Metalearning. *J. Spacecr. Rocket.* **2023**, *60*, 286–298. . [CrossRef]
31. Liu, Z.; Wang, J.; He, S.; Shin, H.S.; Tsourdos, A. Learning prediction-correction guidance for impact time control. *Aerosp. Sci. Technol.* **2021**, *119*, 107187. [CrossRef]
32. Cai, H.; Li, X.; Zhang, Y.; Gao, H. Interception of a Single Intruding Unmanned Aerial Vehicle by Multiple Missiles Using the Novel EA-MADDPG Training Algorithm. *Drones* **2024**, *8*, 524. [CrossRef]
33. Duan, Y.; Schulman, J.; Chen, X.; Bartlett, P.L.; Sutskever, I.; Abbeel, P. RL2: Fast Reinforcement Learning via Slow Reinforcement Learning. *arXiv* **2016**, arXiv:1611.02779.
34. Schulman, J.; Wolski, F.; Dhariwal, P.; Radford, A.; Klimov, O. Proximal Policy Optimization Algorithms. *arXiv* **2017**, arXiv:1707.06347.
35. Lu, P.; Doman, D.B.; Schierman, J.D. Adaptive Terminal Guidance for Hypervelocity Impact in Specified Direction. *J. Guid. Control. Dyn.* **2006**, *29*, 269–278. [CrossRef]
36. Ng, A.; Harada, D.; Russell, S.J. Policy Invariance Under Reward Transformations: Theory and Application to Reward Shaping. In Proceedings of the International Conference on Machine Learning, Bled, Slovenia, 27–30 June 1999.
37. Mishra, N.; Rohaninejad, M.; Chen, X.; Abbeel, P. A Simple Neural Attentive Meta-Learner. In Proceedings of the International Conference on Learning Representations, Toulon, France, 24–26 April 2017.
38. Henderson, P.; Islam, R.; Bachman, P.; Pineau, J.; Precup, D.; Meger, D. Deep reinforcement learning that matters. *Proc. AAAI Conf. Artif. Intell.* **2018**, *32*. [CrossRef]
39. Chung, J.; Gulcehre, C.; Cho, K.; Bengio, Y. Gated feedback recurrent neural networks. In Proceedings of the 32nd International Conference on International Conference on Machine Learning (ICML'15), Lille, France, 6–11 July 2015; Volume 37, pp. 2067–2075.

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.