Hindawi
International Journal of Aerospace Engineering
Volume 2023, Article ID 5883080, 19 pages
https://doi.org/10.1155/2023/5883080

*Research Article*

# Intelligent Online Multiconstrained Reentry Guidance Based on Hindsight Experience Replay

**Qingji Jiang**[1], **Xiaogang Wang**[1], **Yuliang Bai**[1] and **Yu Li**[2]

[1]*School of Astronautics, Harbin Institute of Technology, Harbin 150001, China*
[2]*Beijing Aerospace Technology Institute, Beijing 100074, China*

Correspondence should be addressed to Xiaogang Wang; wangxiaogang@hit.edu.cn

Received 25 December 2022; Revised 12 February 2023; Accepted 17 February 2023; Published 6 March 2023

Academic Editor: Binbin Yan



Traditional guidance algorithms for hypersonic glide vehicles face the challenge of real-time requirements and robustness to multiple deviations or tasks. In this paper, an intelligent online multiconstrained reentry guidance is proposed to strikingly reduce computational burden and enhance the effectiveness with multiple constraints. First, the simulation environment of reentry including dynamics, multiconstraints, and control variables is built. Different from traditional decoupling methods, the bank angle command including its magnitude and sign is designed as the sole guidance variable. Secondly, a policy neural network is designed to output end-to-end guidance commands. By transforming the reentry process into a Markov Decision Process (MDP), the policy network can be trained by deep reinforcement learning (DRL). To address the sparse reward issue caused by multiconstraints, the improved Hindsight Experience Replay (HER) method is adaptively combined with Deep Deterministic Policy Gradient (DDPG) algorithm by transforming multiconstraints into multigoals. As a result, the novel training algorithm can realize higher utilization of failed data and improve the rate of convergence. Finally, simulations for typical scenes show that the policy network in the proposed guidance can output effective commands in much less time than the traditional method. The guidance is robust to initial bias, different targets, and online aerodynamic deviation.

## 1. Introduction

From the last few decades until now, hypersonic glide vehicles (HGV) have attracted the attention of researchers [1–5] due to their high velocity and wide flight airspace. Usually, cooperative guidance has been widely studied as an effective threat to valuable targets represented by hypersonic vehicles [6–10]. HGV shows its superiority against cooperation for its large flight envelope, consisting of the boost phase, initial descend phase, glide phase, and dive phase. To achieve long-range penetration, the glide phase plays an important role. For different mission requirements, the guidance law of the glide phase presents different results. This paper will focus on the flight mission in that HGV flies to the neighborhood of the target at a certain height, and the total time of the glide phase is constrained.

Undergoing years of development, the most popular methods of reentry guidance can be summarized in reference tracking guidance [11–22] and predictor-corrector guidance [23–29]. As far back as 1978, Harpold and Graves [11] described the theoretical basis and guidance architecture of reference tracking guidance, which has been successfully applied in the guidance of space shuttles and other reentry vehicles. The reference tracking guidance is composed of offline trajectory optimization and online trajectory tracking. Commonly, the optimal or suboptimal trajectory is calculated by planning or optimization and stored in an onboard computer before the flight. Once entering the glide phase, the trajectory tracker generates guidance commands according to errors between the actual flight and the stored standard trajectory. Not taking deviation and disturbance into account, the trajectory planning problem can be seen as an optimal control problem, which can be solved by the Legendre pseudospectral method [13, 14], improved Gauss pseudospectral method [15, 16], etc. Other commonly used optimization methods in reentry flight mainly include particle swarm optimization [17] and multiphase convex programming [18]. The trajectory tracking methods mainly include linear quadratic regulator [19],

sliding mode control [20], nonlinear geometric method [21], and fuzzy approximation method [22]. Reference tracking guidance is easy to realize, and the method does not consume much onboard computing resource. Nevertheless, the method is easily affected by initial state changes and kinds of disturbance such as aerodynamic errors, which means its robustness is not good in some situations.

Oppositely, predictor-corrector guidance requires no reference trajectory, which makes it flexible in more flight missions with initial errors and onboard disturbance. The task information stored in the onboard computer is merely terminal constraints. The predictor computes the terminal states by integrating dynamics equations. The corrector calculates the command of AOA (angle of attack) and bank angle to correct the errors between the calculated terminal states and terminal constraints. By repeating the guidance flow, the final errors of terminal constraints are limited to a very small range. Based on different prediction methods, predictor-corrector guidance is classified as analytical predictor-corrector guidance (APCG) and numerical predictor-corrector guidance (NPCG). Analytical prediction reduces the computational work by making some assumptions and simplifying equations of flight states. Kluever [23] proposed a guidance method for the skip-entry phase based on closed-form expressions that are derived from a matched asymptotic expansion analysis. Yu and Chen [24] innovatively presented a method for the derivation of analytical solutions to hypersonic gliding problems by transforming the equations into a linear system with variable coefficients and solving it based on spectral decomposition. Zhang et al. [25] established an auxiliary geocentric rotation frame, simplified the complex dynamics, and got a series of analytical expressions taking energy as the argument. Though the calculation of this kind of prediction is rapid, it faces the problem of insufficient accuracy. Numerical prediction is widely used due to its high accuracy. Joshi et al. [27] adopted NPCG at each guidance cycle to generate a feasible trajectory and presented a guidance algorithm that can satisfy path and terminal constraints. Lu [28] summarized the methods of predecessors and developed a unified predictor-corrector guidance method, based on a simple and robust numerical predictor with an altitude-rate-dependent, which is suitable for kinds of reentry vehicles. Cheng et al. [29] designed a deep neural network in place of the numerical integration and provided an intelligent multiconstrained predictor-corrector guidance, which can be seen as an improvement of NPCG. Methods based on NPCG improve the accuracy, meanwhile causing the problem of intensive computation. Limited by the computing power of the onboard computer, NPCG is not operative in practical use, especially in tasks with a short guidance cycle. Both the reference tracking guidance and predictor-corrector guidance have advantages and disadvantages. If there is no deviation or disturbance, the reentry problem can be solved under the framework of trajectory planning, which is widely studied [30–33]. However, the complex online deviation and rapidity requirement call for an intelligent online method.

In recent years, deep reinforcement learning (DRL) has made a great breakthrough [34–36]. Deep Q-learning (DQN) [35] was first proposed in 2013, which shows great performance in dealing with continuous states in the game field. Then, the Deep Deterministic Policy Gradient (DDPG) [36] further expands the action space from discrete to continuous, which aroused the rapid development of DRL [37–39]. With the strong representation ability of deep learning (DL) and the natural decision-making ability of reinforcement learning (RL), DRL is widely applied in various fields [40–42]. The sparse reward problem is common in DRL tasks, which describes that the agent may only receive a reward signal at the terminal time. To improve the efficiency of training and avoid the sparse reward, based on universal value function approximators [43], Andrychowicz et al. [44] innovatively proposed the Hindsight Experience Replay (HER) method. The HER method replaces the initial goal of the agent with an achieved goal, which greatly leads to abundant reward signals. The original intention of this design decides that the HER method is suitable for those tasks with multigoals. Prianto et al. [45] applied HER to soft actor-critic used for path planning for multiarm manipulators, and the trained agent can generate the shortest path for arbitrary scenarios. Manela and Biess [46] combined HER with curriculum learning on three challenging throwing tasks, which shows wonderful performance in multiple goals and sparse reward functions. Considering that the terminal constraints in the reentry task are similar to multigoals, we can explore the HER method in the DRL guidance frame.

This paper is aimed at finding a real-time plan and guidance scheme for reentry with the help of deep reinforcement learning. Compared with previous studies, the main contributions of this paper are as follows:

(1) The problem of reentry with multiple (path and terminal) constraints is transformed into a Markov Decision Process, which is the precondition of solving it in DRL. Different from the decoupling method in most previous studies, the guidance command of bank angle including magnitude and sign is combined as an action. This makes the guidance logic concise

(2) The HER method is introduced and improved in the training of guidance commands. The HER condition is presented creatively according to the peculiarities of the reentry problem. The flow of this method is easy to generalize to similar guidance problems with other additional constraints

(3) The guidance network is trained offline and called online. The guidance scheme provides an end-to-end guidance command which is calculated by the well-trained network rather than complex guidance schemes. The engineering implementation of the proposed algorithm is simple, and the real-time performance is excellent

The remainder of the paper is organized as follows. Section 2 describes the reentry problem with multiconstraints and the bounds of control variables. Section 3 introduces the basic theory of DRL and the benefit of the HER method.

Section 4 proposes the intelligent multiconstrained guidance algorithm and gives the architecture of two kinds of neural networks. Section 5 gives the simulations and analysis to verify the efficiency of the method. Finally, the conclusions of this paper are in Section 6.

## 2. Problem Formulation

*2.1. Reentry Dynamics Equations.* Considering the earth's rotation, the dynamics equations of the hypersonic glide vehicle are given as follows:

$$\begin{cases} \dot{V} = -\dfrac{D}{m} - g\sin\gamma + \omega^2 r\cos\phi(\sin\gamma\cos\phi - \cos\gamma\sin\phi\cos\psi), \\ \dot{\gamma} = \dfrac{L\cos\sigma}{mV} - \left(\dfrac{g}{V} - \dfrac{V}{r}\right)\cos\gamma + 2\omega\cos\phi\sin\psi + \dfrac{\omega^2 r\cos\phi}{V}(\cos\phi\cos\gamma + \sin\gamma\sin\phi\cos\psi), \\ \dot{\psi} = \dfrac{L\sin\sigma}{mV\cos\gamma} + \dfrac{V}{r}\cos\gamma\sin\psi\tan\phi - 2\omega(\tan\gamma\cos\phi\cos\psi - \sin\phi) + \dfrac{\omega^2 r\sin\phi\cos\phi\sin\psi}{V\cos\gamma}, \\ \dot{r} = V\sin\gamma, \\ \dot{\theta} = \dfrac{V\cos\gamma\sin\psi}{r\cos\phi}, \\ \dot{\phi} = \dfrac{V\cos\gamma\cos\psi}{r}, \end{cases} \tag{1}$$

where $V$ represents the velocity, $\gamma$ represents the flight path angle, $\psi$ represents the velocity heading angle, $r$ represents the distance between the Earth center and the vehicle, $\theta$ represents the longitude, $\phi$ represents the latitude, $m$ represents the vehicle mass, $g$ represents the gravitational acceleration, $\omega$ represents the self-rotation rate of the Earth, and $\sigma$ represents the bank angle. The expressions of lift $L$ and drag force $D$ are

$$\begin{cases} D = \dfrac{1}{2}\rho V^2 C_D S_m, \\ L = \dfrac{1}{2}\rho V^2 C_L S_m, \end{cases} \tag{2}$$

where $\rho$ represents the atmospheric density, $S_m$ represents the reference area, and $C_D$ and $C_L$ represent the drag coefficient and lift coefficient, respectively, which can be obtained by two interpolation functions for AOA and the Mach number.

$\rho$ is a function of height $H$ ($H = r - R_e$, $R_e$ is the radius of the Earth):

$$\rho = \rho_0 e^{-\beta H}. \tag{3}$$

Given an initial state, the trajectory can be obtained by integration based on Equation (1), command of AOA and bank angle.

*2.2. Multiple Constraints.* To complete the flight mission, the HGV must satisfy all the path constraints and terminal constraints.

*2.2.1. Path Constraints.* During the reentry flight, the path constraints can be expressed as

$$\begin{cases} \dot{Q} = k_Q \rho^{0.5} V^{3.15} < \dot{Q}_{\max}, \\ q = \dfrac{1}{2}\rho V^2 < q_{\max}, \\ n = \dfrac{\sqrt{D^2 + L^2}}{m} < n_{\max}, \end{cases} \tag{4}$$

where $\dot{Q}$ is the heating rate of the stagnation point, $q$ is the dynamic pressure, $n$ is the aerodynamic load, and $k_Q$ is a constant. $\dot{Q}_{\max}$, $q_{\max}$, and $n_{\max}$ denote the maximum limits, respectively.

Apart from the three hard constraints, there is a soft QEGC (quasiequilibrium glide condition) constraint, which can make the reentry trajectory smooth:

$$L\cos\sigma - g + \frac{V^2}{r} \geq 0. \tag{5}$$

According to Equations (2)–(5), we can obtain the three lower bounds and one upper bound of $H$ vs. $V$, in which the

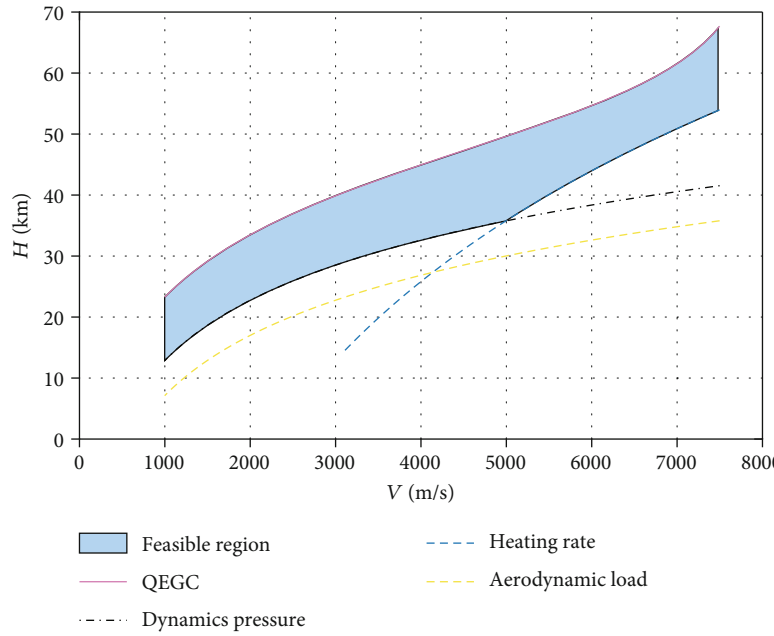


Figure 1: Height-velocity corridor of HGV during the reentry flight.

constraint of the upper bound is soft.

$$\begin{cases} H \geq \dfrac{2}{\beta} \ln \dfrac{k_Q {\rho_0}^{0.5} V^{3.15}}{\dot{Q}_{\max}}, \\ H \geq \dfrac{1}{\beta} \ln \dfrac{\rho_0 V^2}{2q_{\max}}, \\ H \geq \dfrac{1}{\beta} \ln \dfrac{\sqrt{{C_D}^2 + {C_L}^2}\rho_0 V^2 S_m}{2n_{\max} mg}, \\ H \leq \dfrac{1}{\beta} \ln \dfrac{C_L \rho_0 V^2 S_m r}{2m(gr - V^2)}. \end{cases} \tag{6}$$

According to Equation (6), the $H$-$V$ corridor in the glide phase is obtained. Only if the flight trajectory of HGV is limited in the corridor, the path constraints can be satisfied. Intuitively, the $H$-$V$ corridor is shown in Figure 1.

*2.2.2. Terminal Constraints.* Assume that HGV is required to reach the vicinity of the target at a specified height, the terminal constraints (height error $\Delta H$, distance error $\Delta d$) can be summarized as

$$\begin{cases} \Delta H = \left|H_f - H_{re}\right| < \Delta H_{td}, \\ \Delta d = R_e \arccos\left(\sin\phi_f \sin\phi_{\text{tar}} + \cos\phi_f \cos\phi_{\text{tar}} \cos\left(\theta_f - \theta_{\text{tar}}\right)\right) < \Delta d_{td}, \end{cases} \tag{7}$$

where $H_f$, $\phi_f$, and $\theta_f$ represent the final height, latitude, and longitude at the terminal time, $H_{re}$ is the required height, $\phi_{\text{tar}}$ and $\theta_{\text{tar}}$ represent the latitude and longitude of the target, and $\Delta H_{td}$ and $\Delta d_{td}$ are the thresholds of height and distance.

In many practical application scenes, the terminal constraint on velocity is not so tight, which means that the velocity constraint takes the form of

$$V_f > V_{re}, \tag{8}$$

where $V_f$ is the terminal velocity and $V_{re}$ is the required minimum velocity.

Compared to traditional reentry tasks, we consider more terminal constraints in different tasks. For example, to arrive at the required time $T_{re}$, there is another constraint:

$$\Delta T = \left|T_f - T_{re}\right| < \Delta T_{td}, \tag{9}$$

where $T_f$ is the flight time and $\Delta T_{td}$ is the threshold of time.

After this, we give the criterion for the success of a reentry flight. In cases where all the path constraints are satisfied, if the terminal states $\Delta d < 3 \times 10^5$ m, $\Delta H < 2000$ m, $\Delta T < 20$ s, and $V_f > 2500$ m/s, then the flight mission is successful.

Similarly, if the reentry task requires HGV to arrive at a desired terminal approach angle, the relevant constraint condition is adaptively formulated:

$$\Delta\psi = \left|\psi_{\text{LOS}} - \psi_{\text{approach}}\right| < \Delta\psi_{td}, \tag{10}$$

where $\psi_{\text{LOS}}$ is the LOS (line of sight) angle, $\psi_{\text{approach}}$ is the desired approach angle, and $\Delta\psi_{td}$ is the angle threshold.

*2.2.3. Switch Condition.* As is shown in Figure 2, HGV cannot begin sliding from an arbitrary initial status so there is a switch condition. The switch condition refers to the constraint condition at the junction of the initial descent phase and glide phase.

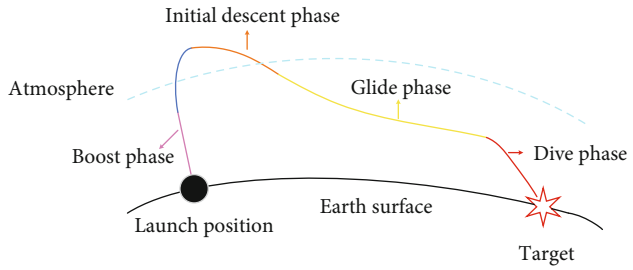


Figure 2: The flight phases of HGV.

Referring to [1], the switch condition is defined as that the absolute value of the difference between the slope of the QEGC at the current point and the actual slope of $r$-$V$ is less than a small preselected value.

$$\left|\frac{dr}{dV} - \left(\frac{dr}{dV}\right)_{\text{QEGC}}\right| < \delta_0, \tag{11}$$

where $\delta_0$ is a small value.

The first slope can be derived from Equation (1):

$$\frac{dr}{dV} = \frac{V \sin\gamma}{-D/m - g\sin\gamma}. \tag{12}$$

The QEGC equation is rewritten as

$$\frac{\rho V^2 C_L S_m}{2m}\cos\sigma + \frac{V^2\cos\gamma}{r} - g\cos\gamma = 0. \tag{13}$$

According to Equation (13), the derivate of $\rho$ to $V$ is obtained:

$$\frac{d\rho}{dV} = \frac{d\rho}{dr}\frac{dr}{dV} = -\beta\rho\frac{dr}{dV}. \tag{14}$$

Both sides of Equation (13) take a derivative to velocity, and then, we can obtain

$$\frac{\rho V^2 C_L S_m \cos\sigma}{2m}\left(\frac{2}{V} - \beta\frac{dh}{dV}\right) + \frac{2V}{r} - \frac{V^2}{r^2}\frac{dh}{dV} + \frac{2g}{r}\frac{dh}{dV} = 0. \tag{15}$$

Substituting Equation (13) into (15), then we can obtain

$$\left(g\cos\gamma - \frac{V^2\cos\gamma}{r}\right)\left(\frac{2}{V} - \beta\frac{dh}{dV}\right) + \frac{2V}{r} = \frac{dr}{dV}\left(\frac{V^2}{r^2} - \frac{2g}{r}\right). \tag{16}$$

Suppose that $\cos\gamma \approx 1$, Equation (16) is simplified as

$$\frac{2g}{V} = \frac{dr}{dV}\left[\frac{V^2}{r^2} - \frac{2g}{r} + \beta\left(g - \frac{V^2}{r}\right)\right]. \tag{17}$$

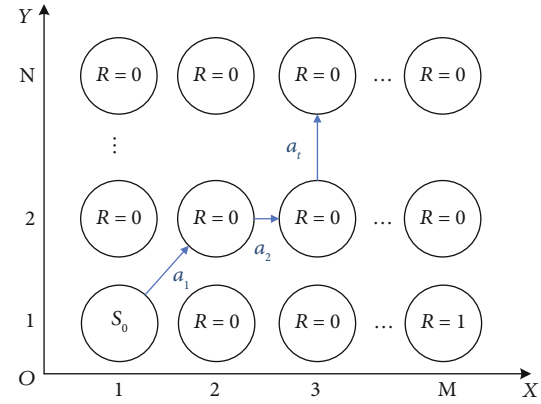


Figure 3: Simple example of HER.

Then, the second slope is obtained:

$$\left(\frac{dr}{dV}\right)_{\text{QEGC}} = \frac{2g/V}{\left[\left(V^2/r^2\right) - \left(2g/r\right) + \beta\left(g - \left(V^2/r\right)\right)\right]}. \tag{18}$$

Substituting Equations (12) and (18) into (11), the switch condition is obtained.

*2.3. Control Variables.* During the reentry process, the trajectory is decided by the AOA $\alpha$ and the bank angle $\sigma$. Commonly, the AOA profile is set as the following piecewise linear function:

$$\alpha = \begin{cases} \alpha_{\max}, & V \geq V_1, \\ \alpha_{\text{best}} + \dfrac{\alpha_{\max} - \alpha_{\text{best}}}{V_1 - V_2}(V - V_2), & V_2 \leq V < V_1, \\ \alpha_{\text{best}}, & V < V_2, \end{cases} \tag{19}$$

where $\alpha_{\max}$ is the max AOA limited by the aerodynamic and $\alpha_{\text{best}}$ is the best AOA when the ratio of lift-to-drag takes the maximum value, which is helpful to increase the flight range of HGV.

Once the AOA profile is determined, the only control variable is the bank angle $\sigma$. According to the QEGC constraint and the three hard path constraints, the upper magnitude bound of $\sigma$ can be derived as follows:

$$\begin{cases} |\sigma| < \arccos\left[\left(g - \dfrac{V^2}{r}\right)\dfrac{2mV^{4.3}k_Q^2}{C_L S_m \dot{Q}_{\max}^2}\right] = \sigma_{\dot{Q}\max}, \\ |\sigma| < \arccos\left[\left(g - \dfrac{V^2}{r}\right)\dfrac{m}{C_L S_m q_{\max}}\right] = \sigma_{q\max}, \\ |\sigma| < \arccos\left[\left(g - \dfrac{V^2}{r}\right)\dfrac{\sqrt{C_D^2 + C_L^2}}{C_L g n_{\max}}\right] = \sigma_{n\max}. \end{cases} \tag{20}$$

The traditional approach to generating guidance command is to calculate the magnitude of $\sigma$ iteratively to satisfy longitudinal constraint and change the sign of $\sigma$ in the horizontal plane in the heading angle corridor to fly HGV

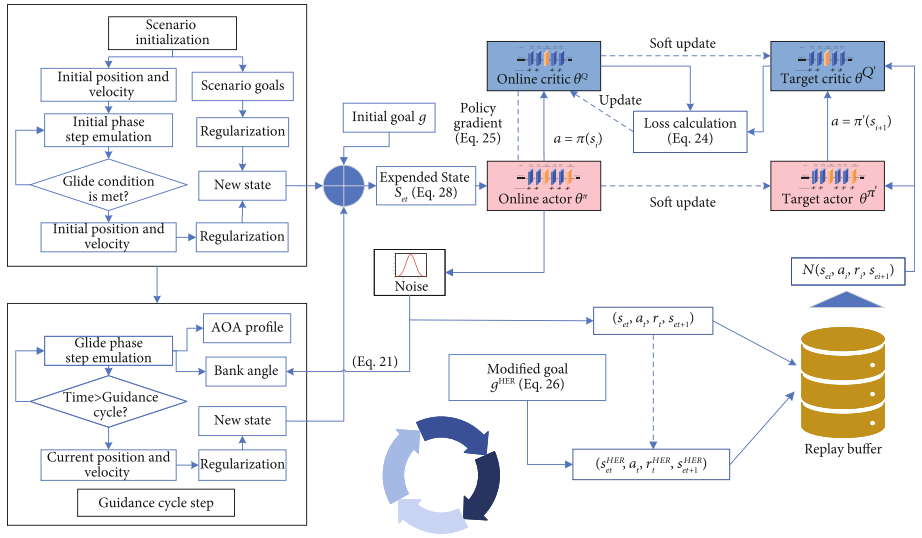


Figure 4: The overall design of the algorithm.

toward the target. However, the application of neural networks allows us to generate both the magnitude and sign of $\sigma$ at the same time. If the output of a policy network is $a \in [-1, 1]$, the command of $\sigma$ is as follows:

$$\sigma_{\text{cmd}} = \text{sign}\ (a)[\sigma_{\min} + |a|(\sigma_{\max} - \sigma_{\min})], \tag{21}$$

where $\sigma_{\max} = \max\ \{\sigma_{Q\max}, \sigma_{q\ \max}, \sigma_{n\ \max}\}$, $\sigma_{\min} = 0$.

## 3. Deep Deterministic Policy Gradient with Hindsight Experience Replay

*3.1. Deep Reinforcement Learning.* Based on Markov Decision Process (MDP), standard reinforcement learning consists of an environment, an agent, and their interactions. The environment is initiated as a state $s_1 \in \mathcal{S}$, which can describe its initial features. The agent observes the state of the environment and chooses an action $a_1 \in \mathcal{A}$. The action affects the environment to transfer to the next state $s_2$ with a certain probability $p_1$ and generates a reward $r_1 \in \mathcal{R}$ to the agent. Then, the agent interacts with the environment repeatedly, so there is a sequence of tuples $<s_i, a_i, p_i, s_{i+1}, r_i>$. The agent's goal is to increase its accumulated rewards.

$$G_t = \sum_{i=t}^{\infty} \lambda^i r_i, \tag{22}$$

where $\lambda$ is the discount factor, which is used to balance the current and future reward.

Action-value function $Q^\pi(s_t, a_t)$ is set to evaluate how good the action $a_t$ is: $Q^\pi(s_t, a_t) = \mathbb{E}(G_t | s_t, a_t)$. And if the policy $\pi$ is optimal, $Q^\pi(s_t, a_t)$ is the optimal action-value function, which satisfies the Bellman equation:

$$Q^*(s, a) = \mathbf{E}\left[r(s, a) + \lambda \max_{a' \in A} Q^*\left(s', a'\right)\right]. \tag{23}$$

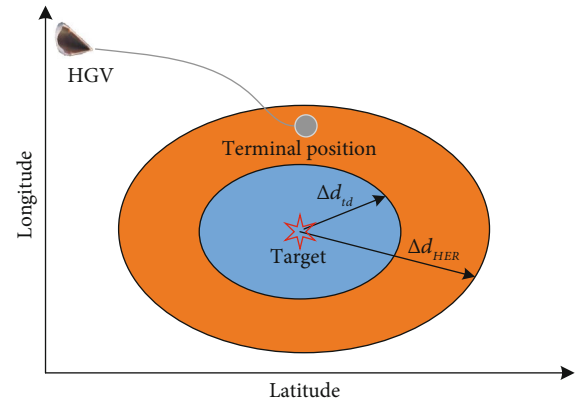


Figure 5: HER condition of HGV.

In DRL, if the policy $\pi$ is deterministic, it can be considered as a mapping from states to actions, commonly in a form of neural network with the parameter $\theta^\pi$.

*3.2. Deep Deterministic Policy Gradient (DDPG).* The baseline algorithm used for the training of neural networks is Deep Deterministic Policy Gradient (DDPG), which belongs to the family of model-free policy gradient DRL methods. There are four neural networks in DDPG, which are called the online actor network $\pi(s|\theta^\pi)$, the target actor network $\pi'(s|\theta^{\pi'})$, the online critic network $Q(s, a \mid \theta^Q)$, and the target critic network $Q'(s, a \mid \theta^{Q'})$. The online actor network parameterized by $\theta^\pi$ exports actions of the agent, and the online critic network parameterized by $\theta^Q$ is designed to approximate action-value function $Q(s, a)$. The actual behavior policy $a$ of the agent is presented from $\pi$ with an added noise $\mathcal{N}$.

The parameter $\theta^Q$ is optimized by decreasing the loss:

$$L_{\theta^Q} = \mathbf{E}\left[\left(r(s_t, a_t) + \lambda Q'\left(s_{t+1}, \pi'\left(s_{t+1}|\theta^{\pi'}\right)\middle|\theta^{Q'}\right) - Q\left(s_t, a_t|\theta^Q\right)\right)^2\right]. \tag{24}$$

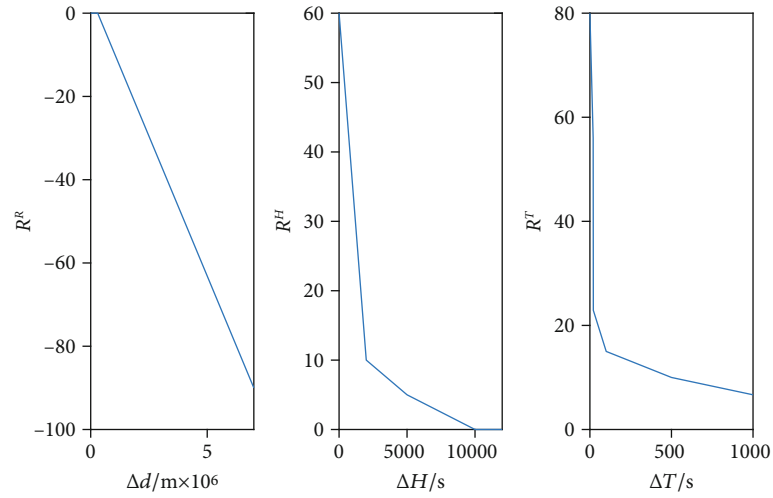


Figure 6: The curves of subrewards.

The actor $\pi(s|\theta^\pi)$ is updated by the theory of policy gradient:

$$\begin{aligned}\nabla_{\theta^\pi} J &= \mathbf{E}_{s_t\sim\xi}\left[\nabla_{\theta^\pi} Q\left(s, a|\theta^Q\right)\Big|_{s=s_t, a=\pi(s_t,\theta^\pi)}\right] \\ &= \mathbf{E}_{s_t\sim\xi}\left[\nabla_a Q\left(s, a|\theta^Q\right)\nabla_{\theta^\pi}\pi(s|\theta^\pi)_{s=s_t}\right],\end{aligned} \tag{25}$$

where $\xi$ is the distribution of the state.

The existence of the target actor network and the target critic network can make the process of training more stable. Besides, the data is stored in a replay buffer to reduce the relevance between experiences and deal with the experiences effectively.

*3.3. Hindsight Experience Replay (HER).* The key idea behind Hindsight Experience Replay (HER) is derived from universal value function approximators (UVFA) [43]. In UVFA, there is a goal space $\mathcal{G}$. For any goal $g \in \mathcal{G}$, there is a pseudoreward $R_g(s, a, s')$: $\mathcal{S}\times\mathcal{A}\longrightarrow\mathcal{R}$. The first state-goal pair in every episode is sampled from a distribution $p(s_0, g)$, and in the same episode, the goal $g$ is fixed. At every step in an episode, the agent gets an action from the current state and goal $\pi : \mathcal{S}\times\mathcal{G}\longrightarrow\mathcal{A}$. On this premise, the traditional $Q$-function is expanded to $Q(s_t, a_t, g) = \mathbb{E}[Rt \mid s_t, a_t, g]$.

In a task where the agent needs to achieve multiple goals, the designed reward is quite likely to be sparse. To deal with sparse rewards, an effective trick is to exploit the failed data samples too. In the method of HER, the states of the agent are expanded to combinations of states and goals. The essence of HER rests with designing a virtual goal $g'$, which is achieved in a failed trajectory. For example, there is an RL task in Figure 3, where the agent has a goal position. Only if the agent arrives at the goal $g(M, 1)$ it gets a reward 1, otherwise 0. In most episodes, the agent gets 0 rewards. It is quite difficult for conventional RL algorithms to achieve good results for the sparse reward problem especially when $M$ and $N$ are large numbers. In the HER method, if the final position is $(3, N)$, we can define the position as a virtual goal $g'$, and then, the final reward $R_t \mid s_t, a_t, g = 0$, but $R_t \mid s_t, a_t, g' = 1$. By doing this, the algorithm can learn from failed experiences, and the reward is changed from sparse to dense.

With the help of virtual goals, the efficiency of experience is improved, which makes the convergence of neural networks significantly ameliorated.

## 4. Design of Intelligent Guidance and Training of Neural Networks

*4.1. Overall Design.* In this section, the HER-DDPG-based guidance is proposed. First of all, we normalize the reentry simulation into two segments (scenario initialization and guidance step) and open two corresponding interfaces to the intelligent algorithm. In the segment of scenario initialization, the position and velocity of HGV are randomly initialized within a certain range. The simulation starts from the specified point at the initial phase and continues to the switch point decided by Equation (11). After entering the glide phase, at each decision step later, HGV receives the guidance command of bank angle via the interface to the intelligent algorithm and calculates AOA according to the profile given by Equation (19). The simulation proceeds until distance error $\Delta d < \Delta d_{td}$ or the velocity $V_f < V_{re}$ or any path constraint is not satisfied.

After constructing the two segments, the transformation from simulation to MDP is accomplished. The kinematical parameters of HGV are mapped to states, and the guidance command of bank angle is designed as the action. Based on the theory of HER, the terminal constraints are transformed into multigoals and the expanded states are defined. The reward function is designed as a combination of several linear functions of terminal height, terminal distance error, and flight time error, which is used for generating dense signals of policy gradient. Finally, based on HER and DDPG,

Randomly initialize parameters of actor network $\pi(s|\theta^{\pi})$ and critic network $Q(s, a \mid \theta^{Q})$.
Initialize target actor $\pi'(s|\theta^{\pi'})$ and target critic $Q'(s, a \mid \theta^{Q'})$ with $\theta^{\pi'} \longleftarrow \theta^{\pi}$, $\theta^{Q'} \longleftarrow \theta^{Q}$.
Initialize basic replay buffer $\boldsymbol{R_B}$
**for** episode = 1, $N_e$ **do**
  Initialize an HER replay buffer $\boldsymbol{R}_{\mathbf{HER}}$ and a random process noise $\mathcal{N}$ for exploration
  Initialize an initial goal $g = [\theta_{\text{tar}}, \phi_{\text{tar}}, H_{re}, T_{re}]$ randomly
  Run the Scenario Initialization Function, sample a basic state $s_0 = [V, \gamma, \psi, r, \theta, \phi]$
  **for** $t = 0$, $N_T$ **do**
    Combine basic state and goal to expanded state $s_{et} = [s_t \parallel g]$
    Sample an action from actor and noise: $a_t = \pi(s_{et}|\theta^{\pi}) + \mathcal{N}_t$
    Execute the action in the Policy Step Function and observe a new state $s_{t+1}$
    Combine basic new state and goal to expand new state $s_{et+1} = [s_{t+1} \parallel g]$
    Store the transition $(s_{et}, a_t, r_t, s_{et+1})$ in $\boldsymbol{R_B}$ and $\boldsymbol{R}_{\mathbf{HER}}$
    **if** the episode is done
      Judge whether the HER condition is met and record it
    **end if**
  **end for**
  **if** the HER condition is met
    **for** $t = 0$, $N_T$ in $\boldsymbol{R}_{\mathbf{HER}}$ **do**
      Calculate $g^{\text{HER}} = [\theta_{\text{tar}}^{\text{HER}}, \phi_{\text{tar}}^{\text{HER}}, H_{re}^{\text{HER}}, T_{re}^{\text{HER}}]$ and $\Delta H^{\text{HER}}$, $\Delta d^{\text{HER}}$, $\Delta T^{\text{HER}}$
      Recalculate reward $r_t^{\text{HER}}$ according to Equations (30)–(33)
      Combine basic states and goal to expand states:
      $s_{et}^{\text{HER}} = [s_{et} \parallel g^{\text{HER}}]$, $s_{et+1}^{\text{HER}} = [s_{et+1} \parallel g^{\text{HER}}]$
      Store the transition $(s_{et}^{\text{HER}}, a_t, r_t^{\text{HER}}, s_{et+1}^{\text{HER}})$ in $\boldsymbol{R_B}$
    **end for**
    Clear data in $\boldsymbol{R}_{\mathbf{HER}}$
  **end if**
  **for** $t = 0$, $N_{\text{train}}$ **do**
    Sample a minibatch $\boldsymbol{B}$ from the replay buffer $\boldsymbol{R_B}$
    Update critic by Equation (24), update actor by Equation (25)
    Update target network periodically: $\theta^{\pi'} \longleftarrow \tau\theta^{\pi} + (1-\tau)\theta^{\pi'}$, $\theta^{Q'} \longleftarrow \tau\theta^{Q} + (1-\tau)\theta^{Q'}$
  **end for**
**end for**

Algorithm 1: Training of multiconstrained reentry guidance based on HER.

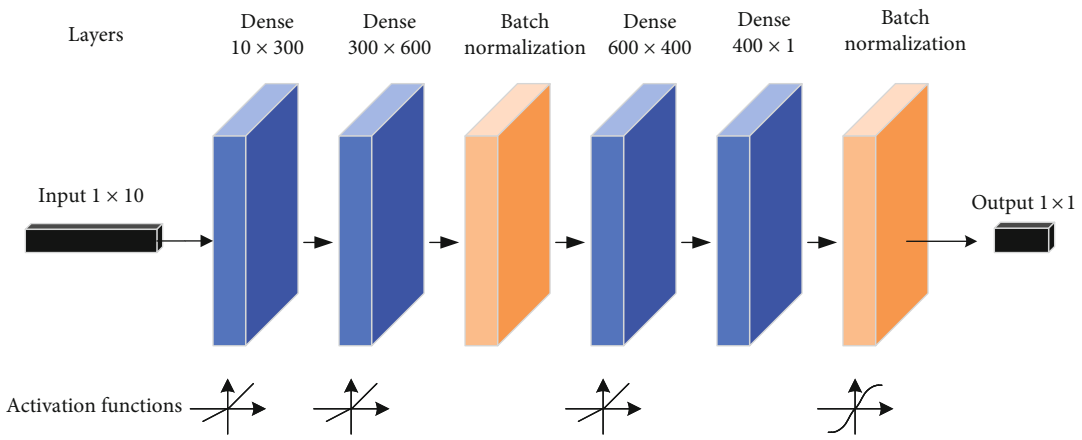


Figure 7: The architecture of the actor network.

the algorithm goes into operation as is shown in Figure 4. When the neural networks converge, the training process is terminated.

*4.2. Markov Decision Process Variables.* The fundamental variables in MDP are defined as follows.

*4.2.1. States.* The basic state of the HGV agent is defined as $s = [V, \gamma, \psi, r, \theta, \phi]$, which can uniquely express the features of the motion.

*4.2.2. Goals.* Goals describe the terminal constraints of HGV with a certain fixed tolerance of position and time. The

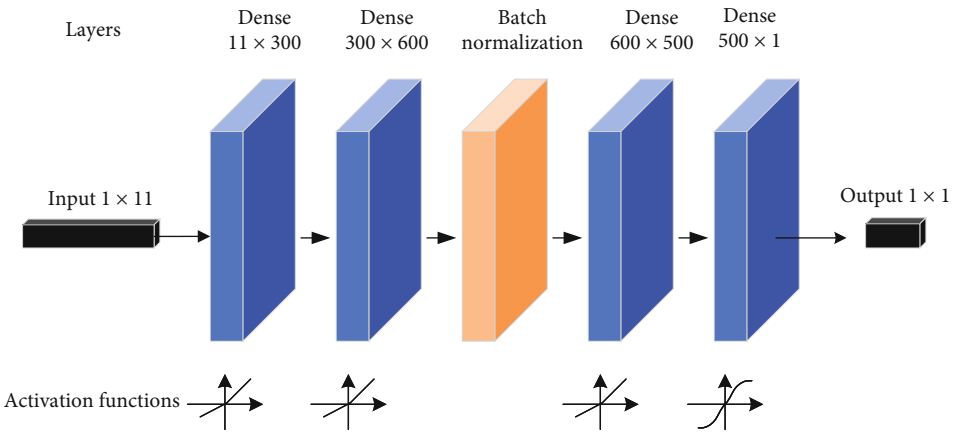


Figure 8: The architecture of the critic network.

Table 1: The architecture of the actor network.

| Number | Layer type | Layer nodes | Activation function |
|---|---|---|---|
| 1 | Dense | 10 × 300 | Leaky ReLU |
| 2 | Dense | 300 × 600 | Leaky RELU |
| 3 | Batch normalization | 600 | — |
| 4 | Dense | 600 × 400 | Leaky ReLU |
| 5 | Dense | 400 × 1 | — |
| 6 | Batch normalization | 1 | Tanh |

Table 2: The architecture of the critic network.

| Number | Layer type | Layer nodes | Activation function |
|---|---|---|---|
| 1 | Dense | 11 × 300 | Leaky ReLU |
| 2 | Dense | 300 × 600 | Leaky RELU |
| 3 | Batch normalization | 600 | — |
| 4 | Dense | 600 × 500 | Leaky ReLU |
| 5 | Dense | 500 × 1 | Tanh |

Table 3: Parameters of initial reentry point and target point.

| Initial parameters | Value |
|---|---|
| Initial height $H_0$ | 60~64 km |
| Initial velocity $V_0$ | 6200~6800 m/s |
| Initial path angle $\sigma_0$ | −0.02~−0.01 deg |
| Target longitude $\theta_{\text{tar}}$ | 60~65 deg |
| Target latitude $\phi_{\text{tar}}$ | −7~7 deg |
| Required height $H_{re}$ | 26~32 km |
| Required time $T_{re}$ | 1200~1500 s |
| The error of drag coefficient $\Delta C_D$ | 0~10% |
| The error of lift coefficient $\Delta C_L$ | 0~10% |

initial goal is defined as $g = [\theta_{\text{tar}}, \phi_{\text{tar}}, H_{re}, T_{re}]$. In this paper, $\Delta d_{td} = 300$ km, if the final distance error $\Delta d < 300$ km then the terminal distance constraint is satisfied. If the loose constraint $\Delta d < \Delta d_{\text{HER}} = 2000$ km, we decide that the HER condition is met.

Table 4: The hyperparameters in the training.

| Hyperparameter | DDPG | PPO | DDPG+HER |
|---|---|---|---|
| Discount factor $\lambda$ | 0.99 | 0.99 | 0.99 |
| Batch size | 64 | 64 | 64 |
| Replay buffer size | 20000 | — | 20000 |
| Actor learning rate | $10^{-4}$ | $10^{-3}$ | $10^{-4}$ |
| Critic learning rate | $10^{-3}$ | $10^{-3}$ | $10^{-3}$ |
| Target update rate $\tau$ | 0.001 | — | 0.001 |
| Maximum number of steps | 1000 | 1000 | 1000 |
| Exploration policy | OU | — | $\mathcal{N}(0,0.1)$ |
| GAE factor | — | 0.98 | — |
| Clip factor | — | 0.2 | — |

The HER condition is shown in Figure 5, which means that despite its failure of arriving at the given target, HGV has arrived at the terminal position successfully. Here, we define $\theta_{\text{tar}}^{\text{HER}}$, $\phi_{\text{tar}}^{\text{HER}}$, $H_{re}^{\text{HER}}$, and $T_{re}^{\text{HER}}$ as target longitude, target latitude, required height, and required time on the HER condition. The intention of HER is to efficiently increase the sampled data of failed episodes, which means that they may be successful in other target tasks. Therefore, the required height and time are modified as the actual values at the terminal time when the errors concerning initial goals are in a specifically designed range. The concrete values of them in this paper are given as

$$\begin{cases} \theta_{\text{tar}}^{\text{HER}} = \theta_f, \phi_{\text{tar}}^{\text{HER}} = \phi_f & \\ H_{re}^{\text{HER}} = H_f, & \Delta H \le 8\,\text{km}, \\ H_{re}^{\text{HER}} = H_{re}, & \Delta H > 8\,\text{km}, \\ T_{re}^{\text{HER}} = T_f, & \Delta T \le 200\,\text{s}, \\ T_{re}^{\text{HER}} = T_{re}, & \Delta T > 200\,\text{s}. \end{cases} \tag{26}$$

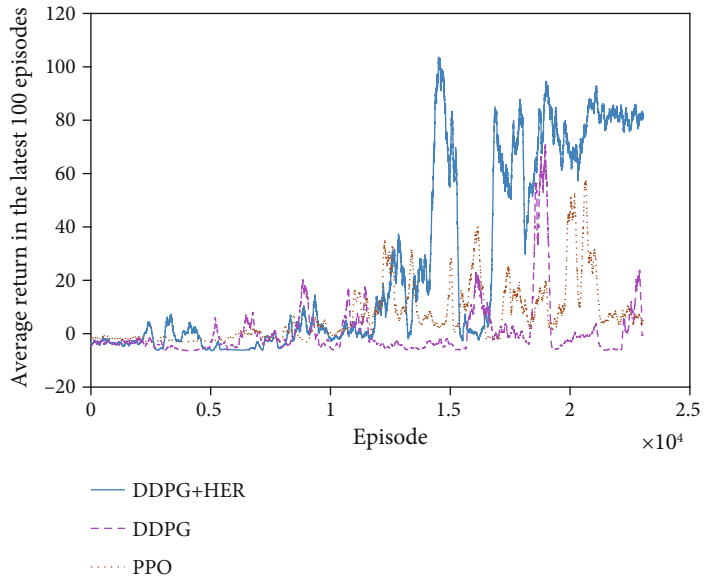


Figure 9: Contrast curves of the average return in the latest 100 episodes.

Then, the new goal is defined as $g^{\mathrm{HER}} = [\theta_{\mathrm{tar}}^{\mathrm{HER}}, \phi_{\mathrm{tar}}^{\mathrm{HER}}, H_{re}^{\mathrm{HER}}, T_{re}^{\mathrm{HER}}]$.

In this case, the new terminal errors can be expressed as

$$\begin{cases} \Delta H^{\mathrm{HER}} = \left| H_f - H_{re}^{\mathrm{HER}} \right|, \\ \Delta d^{\mathrm{HER}} = R_e \arccos\left( \sin\phi_f \sin\phi_{\mathrm{tar}}^{\mathrm{HER}} + \cos\phi_f \cos\phi_{\mathrm{tar}}^{\mathrm{HER}} \cos\left( \theta_f - \theta_{\mathrm{tar}}^{\mathrm{HER}} \right) \right), \\ \Delta T^{\mathrm{HER}} = \left| T_f - T_{\mathrm{tar}}^{\mathrm{HER}} \right|. \end{cases} \tag{27}$$

By doing this, lots of failed data is transformed into successful data with new terminal constraints. The reward signal is more abundant in the training process, and the average reward can increase. In DRL training, the more abundant the reward signal, the faster the rate of convergence.

*4.2.3. Expanded States.* The expanded state can be defined as

$$s_e = [s \| g] = [V, \gamma, \psi, r, \theta, \phi, \theta_{\mathrm{tar}}, \phi_{\mathrm{tar}}, H_{re}, T_{re}]. \tag{28}$$

On the HER condition, the expanded state is modified accordingly:

$$s_e^{\mathrm{HER}} = \left[ s \| g^{\mathrm{HER}} \right] = \left[ V, \gamma, \psi, r, \theta, \phi, \theta_{\mathrm{tar}}^{\mathrm{HER}}, \phi_{\mathrm{tar}}^{\mathrm{HER}}, H_{re}^{\mathrm{HER}}, T_{re}^{\mathrm{HER}} \right]. \tag{29}$$

*4.2.4. Rewards.* In the paper where HER is proposed [44], the success of the task is merely relative to the position of a robotic arm, so it is feasible to learn from rewards that are sparse and binary. However, the number of terminal constraints in the reentry task is so up to 4 that the binary reward is no longer suitable. For the reentry task, the priorities of satisfying different terminal constraints are not at the same level. Intuitively, the range constraint is the most important, for it requires the agent to adjust both the magnitude and the sign of the bank angle to arrive in the vicinity of the target. Secondly, the height constraint is also important. After the two constraints are under consideration, the time-related reward is added. Considering the following dive phase after the glide phase, the constraint thresholds are defined: $\Delta d_{td} = 3 \times 10^5$ m, $\Delta H_{td} = 2000$ m, and $\Delta T_{td} = 20$ s.

Piecewise linear functions are applied to design the range-relative reward $R^R$, height-relative reward $R^H$, and time-relative reward $R^T$:

$$R^R = \begin{cases} 0 + \dfrac{0 - (-90)}{(3 - 70) \times 10^5} \left( \Delta d - 3 \times 10^5 \right), & \Delta d > 3 \times 10^5, \\ 0, & \Delta d \le 3 \times 10^5, \end{cases} \tag{30}$$

$$R^H = \begin{cases} 60 + \dfrac{60 - 10}{0 - 2000} (\Delta H - 0), & \Delta H \le 2000, \\ 10 + \dfrac{10 - 5}{2000 - 5000} (\Delta H - 2000), & 2000 < \Delta H \le 5000, \\ 5 + \dfrac{5 - 0}{5000 - 10000} (\Delta H - 5000), & 5000 < \Delta H \le 10000, \\ 0, & \Delta H > 10000, \end{cases} \tag{31}$$

$$R^T = R^H = \begin{cases} 80 + \dfrac{80 - 0}{0 - 20} (\Delta T - 0), & \Delta T \le 20, \\ 0 + \dfrac{0 - (-10)}{20 - 100} (\Delta T - 20), & 20 < \Delta T \le 100, \\ -10 + \dfrac{-10 - (-12)}{100 - 500} (\Delta T - 100), & 100 < \Delta T \le 500, \\ -12 + \dfrac{-12 - (-20)}{500 - 10000} (\Delta T - 10000), & \Delta T > 500, \end{cases} \tag{32}$$

The 3 subrewards are shown in Figure 6.

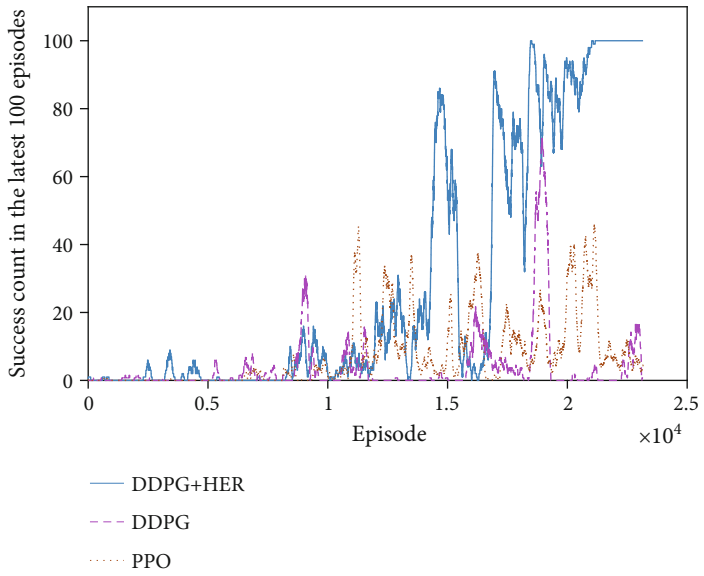


Figure 10: Contrast curves of the success count in the latest 100 episodes.

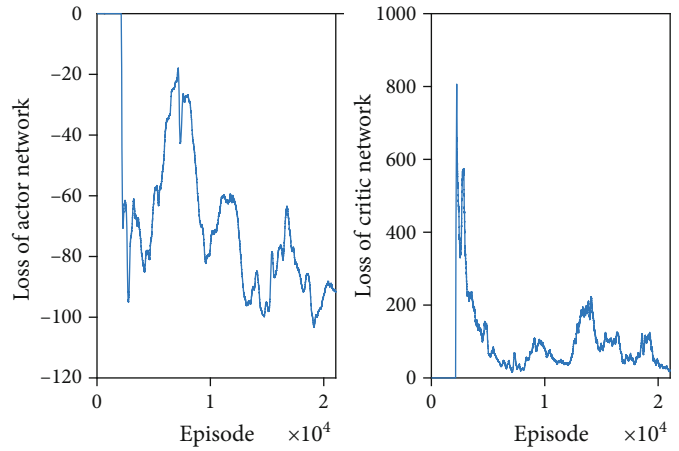


Figure 11: The loss of actor and critic in DDPG+HER.

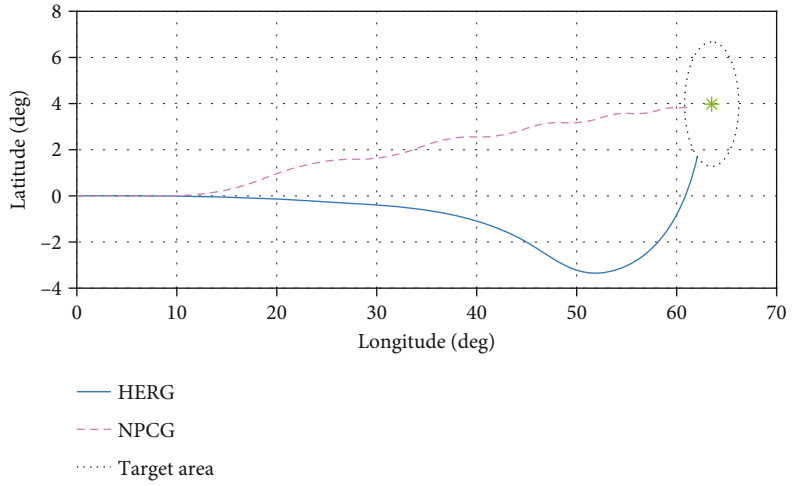


Figure 12: Reentry trajectory of HGV.

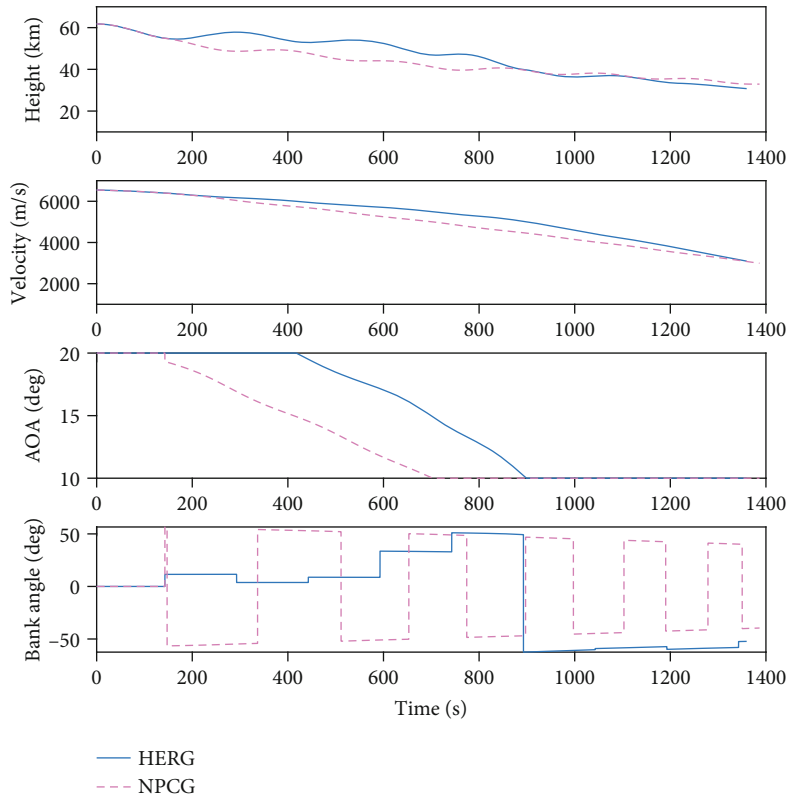


Figure 13: State-time curves.

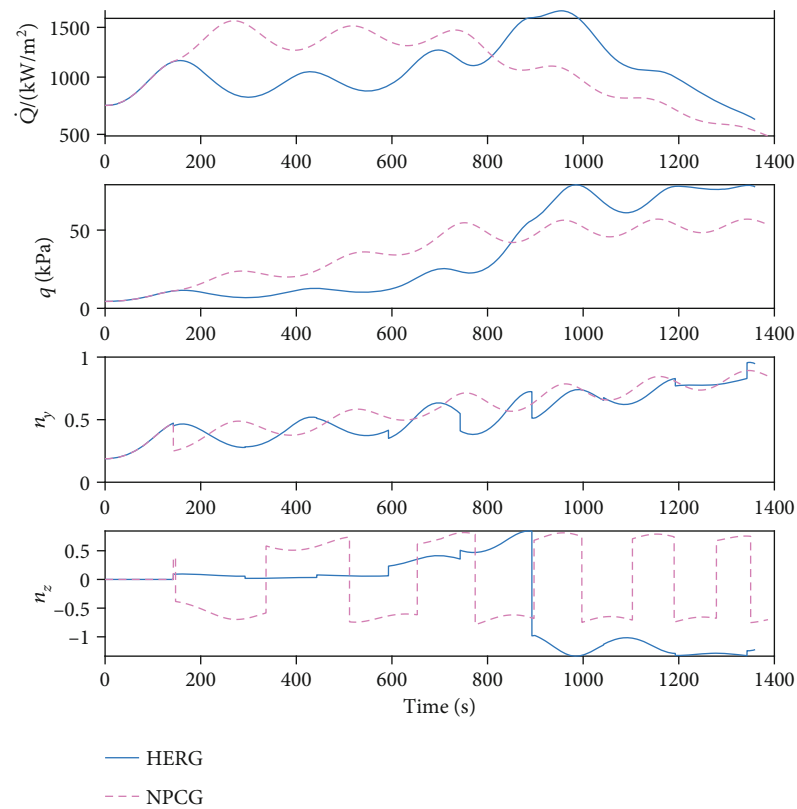


Figure 14: Path constraint-time curves.

Table 5: Computing time of NPCG.

| Order | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|---|---|---|---|---|---|---|---|---|---|
| Time (s) | 3.387 | 2.796 | 2.411 | 2.220 | 2.002 | 1.397 | 1.072 | 0.628 | 0.291 |

Table 6: Contrast simulation results.

| Method | Computing time (s) | Range error (km) | Height error (km) | Time error (s) |
|---|---|---|---|---|
| NPCG | 6.431 | 299.998 | 2.487 | 16.6 |
| HERG | 0.0937 | 299.999 | 0.326 | 10.3 |

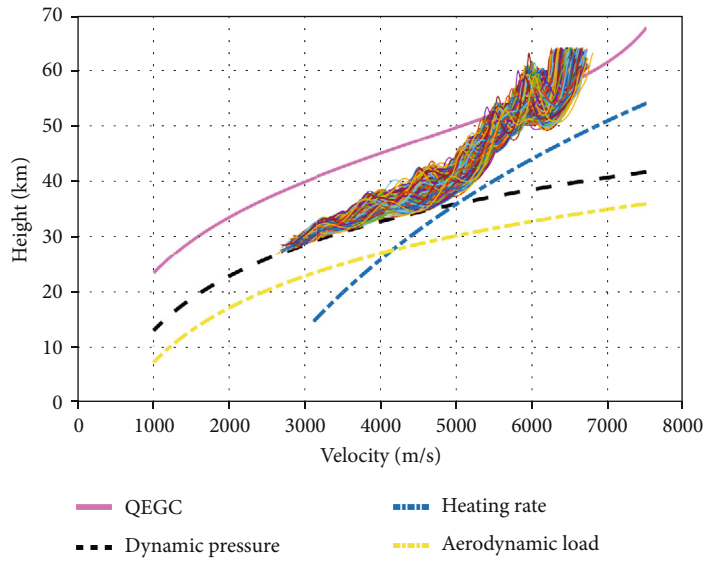


Figure 15: Height-velocity curve.

Then, the total reward $R$ is designed as follows:

$$R = \begin{cases} R^R, & \Delta d > 3 \times 10^5, \\ R^R + R^H, & \Delta d \le 3 \times 10^5, \Delta H \ge 2000, \\ R^R + R^H + R^T, & \Delta d < 3 \times 10^5, \Delta H < 2000. \end{cases} \quad (33)$$

$R^R$ is designed to reduce the terminal distance error in the range of $\Delta d_{td}$, which means that if the distance error $\Delta d > 300$ km then the subreward $R^R < 0$, and if $\Delta d \le 300$ km, then the total reward is determined by $R^H$ and $R^T$. $R^H$ is designed to reduce the terminal height error, so the smaller the $\Delta H$, the larger the $R^H$. And in the whole training process, the number of samples with larger $\Delta H$ is more than smaller $\Delta H$, so as $\Delta H$ decreases, the slopes of piecewise linear functions increase. Finally, only if the height error $\Delta H < \Delta H_{td}$ is the subreward $R^T$ added in $R$. The design of $R^T$ is similar, so the smaller the $\Delta T$, the larger the $R^T$. And as $\Delta T$ decreases, the slopes of piecewise linear functions increase too.

If the terminal states $\Delta d < 3 \times 10^5$ m, $\Delta H < 2000$ m, $\Delta T < 20$ s, and $V_f > 2500$ m/s, then the flight mission is successful. According to Equations (30)–(33), the reward of a successful episode is in the range of $10 \sim 140$. For some successful episodes, the average reward value will be in the range of 10 to 140, which depends on the final data distribution of $\Delta H$ and $\Delta T$.

*4.2.5. Actions.* Different from the widely used decoupling methods in current guidance algorithms, the magnitude and sign command of the bank angle are provided at the same time in this paper. Commonly action $A$ is calculated by a policy network. In DDPG, the output of the policy network is a continuous value $A \in [-1, 1]$. The action is mapped to the bank angle command in the form of Equation (21).

*4.3. Algorithm Flow for Training.* Based on HER, the training algorithm of multiconstrained reentry guidance is given in Algorithm 1.

*4.4. Neural Network Design.* The designed actor network model and critic network model are shown in Figures 7 and 8. And the details of every layer are listed in Tables 1 and 2.

## 5. Simulations and Analysis

*5.1. Simulation Settings.* The HGV model used in this paper is the common aero vehicle (CAV-H). In Equation (1), $m = 907$ kg and $g = 9.8066$ m/s$^2$. In Equation (2), $S_m = 0.4839$ m$^2$. In Equation (3), $\rho_0 = 1.2250$ kg/m$^3$, $\beta = 0.00014065$, and $R_e = 6378004$ m. In Equation (4), $k_Q = 5 \times 10^{-5}$, $q_{\max} = 100$ kPa,

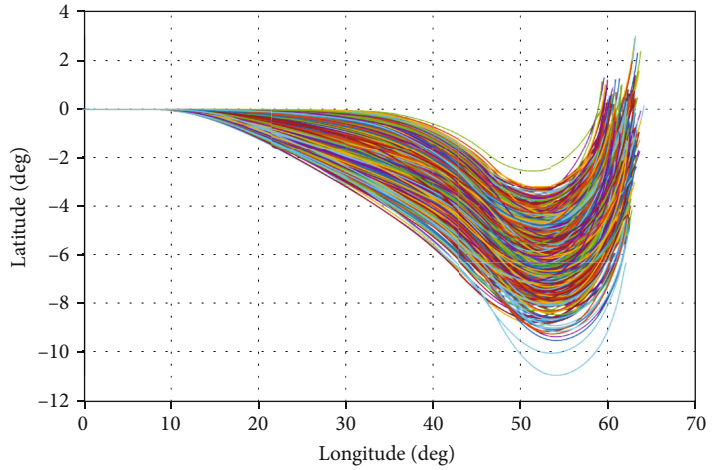


Figure 16: Latitude-longitude curve.

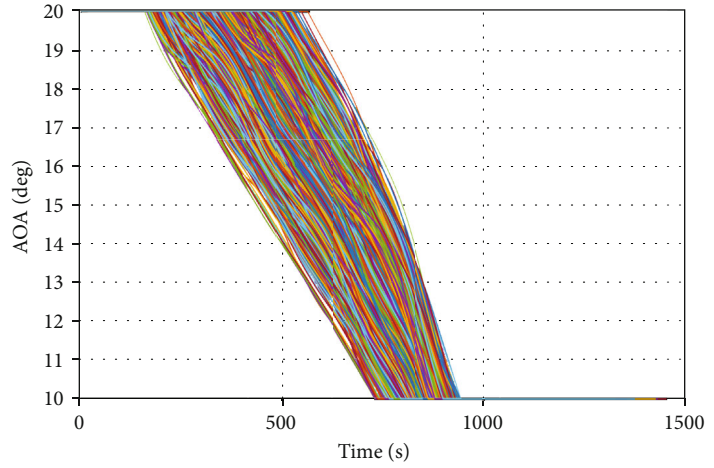


Figure 17: AOA-time curve.

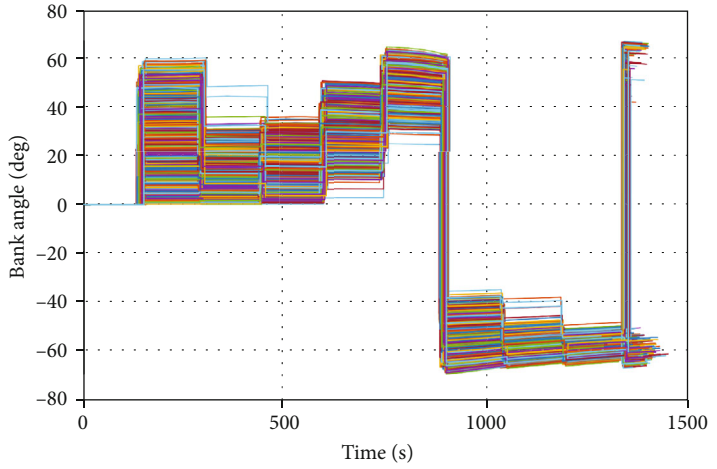


Figure 18: Bank angle-time curve.

$n_{\max} = 3$, and $\dot{Q}_{\max} = 2000\,\text{kW/m}^2$. In Equation (8), $V_{re} = 2500$ m/s. In Equation (19), $\alpha_{\max} = 20$ deg, $\alpha_{\text{best}} = 10$ deg, $V_1 = 6000$ m/s, and $V_2 = 5000$ m/s. The initial longitude and latitude of HGV are 0 deg. The size of the integral step is 0.01 s, and the size of the guidance (policy) step is 150 s. All used random variables in Table 3 are subject to the uniform distribution.

After setting the above parameters, the networks are trained by Algorithm 1.

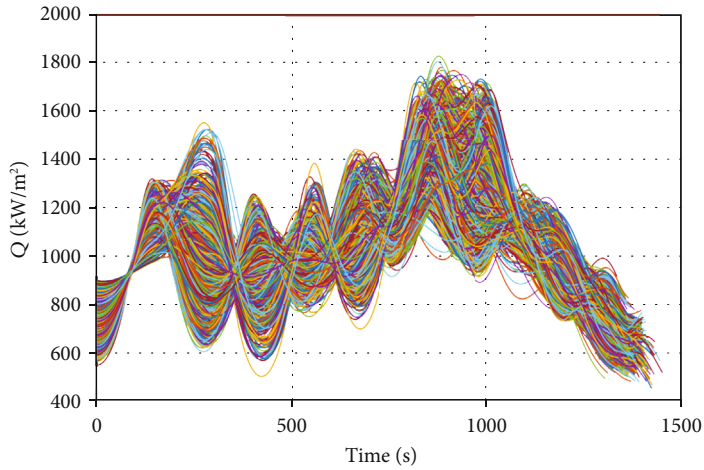


Figure 19: Heating rate-time curve.

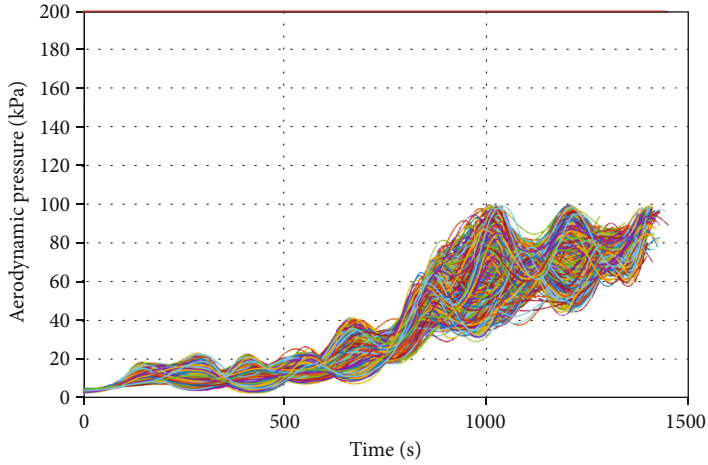


Figure 20: Dynamic pressure-time curve.

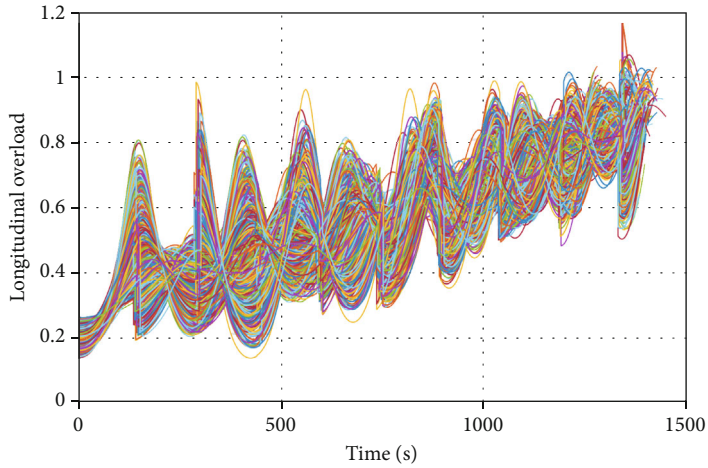


Figure 21: Longitudinal overload-time curve.

*5.2. Training Process of Neural Networks.* Comparative experiments are made at the same time to verify the superiority of DDPG+HER method. DDPG algorithm [36] and Proximal Policy Optimization (PPO) algorithm [37] are selected as control groups. The hyperparameters of the three algorithms used in the training process are set in Table 4.

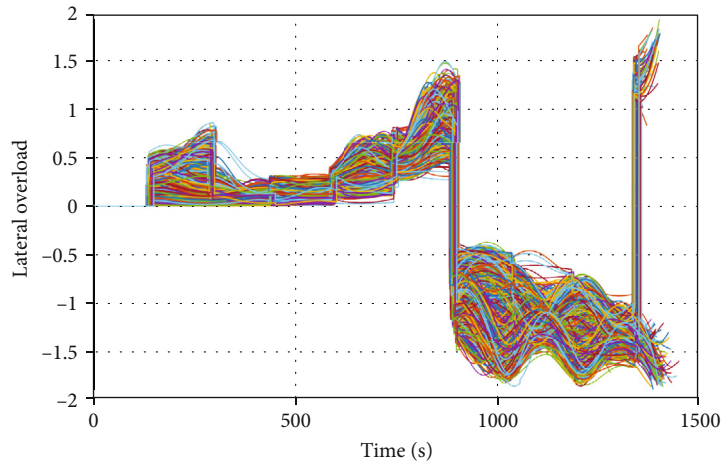


Figure 22: Lateral overload-time curve.

Table 7: Terminal constraints of the Monte Carlo simulations.

| Terminal constraints | Distance error (km) | Height error (km) | Time error (s) |
|---|---|---|---|
| Mean value | 299.850 | 1.084 | 9.907 |
| Minimum value | 299.652 | 0.0002 | 0.0301 |
| Maximum value | 299.999 | 1.993 | 19.961 |

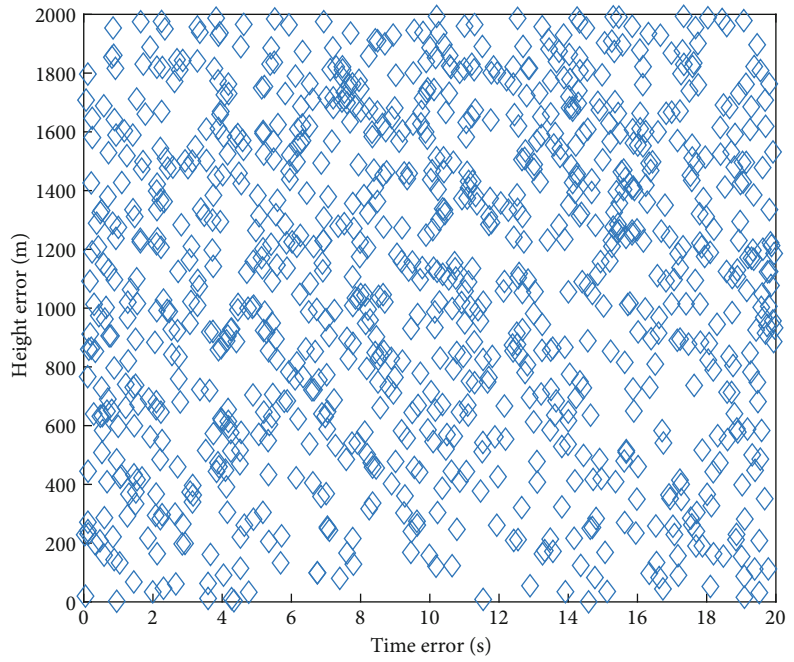


Figure 23: Terminal height-time error.

After the training, the average returns and success count in the latest 100 episodes are shown in Figures 9 and 10. The total episode number is 23161.

It can be seen from Figure 9 that the average return of the DDPG+HER method begins rising first and gets more rewards than other methods. The sparse reward problem is effectively alleviated. At the end time of training, the success rate of the DDPG+HER method in Figure 10 converges to 100%, which implies that the policy given by the actor is operative. The best success count of DDPG is less than 75 and the best success count of PPO is less than 60, which powerfully illustrates that the proposed DDPG+HER method is more effective in the training of the policy network.

In Figure 11, the loss of actor and critic in the DDPG+HER method is given.

It can be seen from Figure 11 that the loss of the critic is decreasing to zero along with the progress of training.

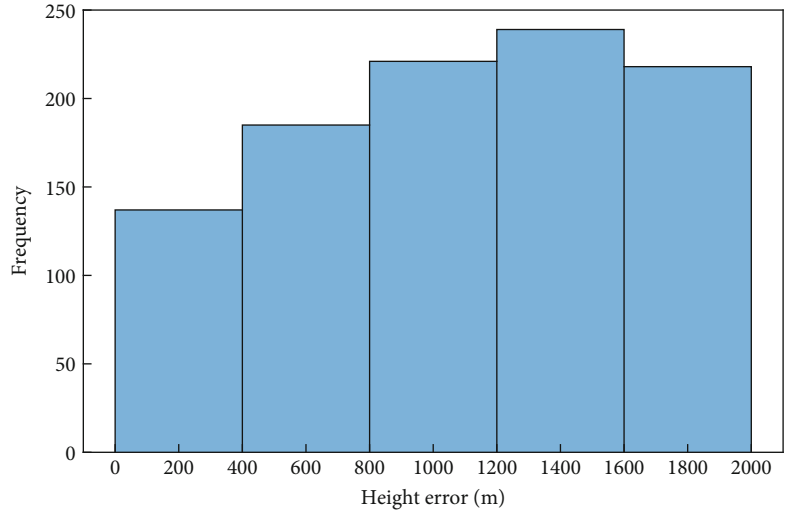


Figure 24: Histogram of height error.

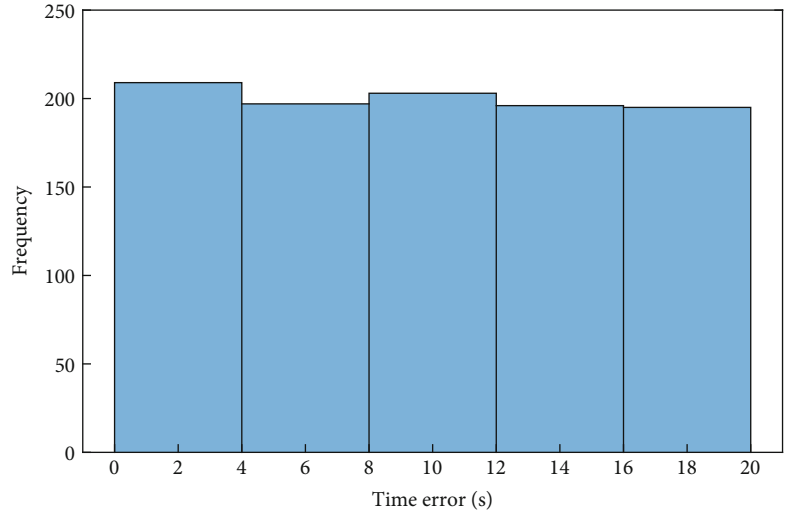


Figure 25: Histogram of time error.

Meanwhile, the magnitude of actor loss is increasing in the process, which means that the score of the agent is on a trend of improvement.

*5.3. Random Single Trajectory Analysis.* After the convergence of neural networks, the typical simulation is performed randomly. To evaluate the performance of the proposed guidance, a contrast simulation based on NPCG is also performed. The proposed method in this paper is called HERG for short.

The terminal constraints are set as target longitude $\lambda_{\text{tar}} = 63.505$ deg, latitude $\phi_{\text{tar}} = 3.976$ deg, required height $H_{re} = 30.429$ km, and required time $T_{re} = 1370.0$ s. The errors of aerodynamics are set as $\Delta C_D = 3.5\%$ and $\Delta C_L = 4.5\%$. For the trajectory of HERG, the terminal height is 30.755 km, and the terminal time is 1359.7 s. For the trajectory of NPCG, the terminal height is 32.916 km, and the terminal time is 1386.6 s.

The results are shown in Figures 12–14.

Figure 12 shows that the HGV can arrive at the required target area. Figure 13 shows that the policy decision is executed 9 times, and terminal height and time are in the required range. The reverse number of the bank angle is reduced greatly for HERG than NPCG. And as is seen in Figure 14, all of the path constraints are satisfied.

The simulations are performed in a system with a 12th Gen Intel Core i5-12600KF CPU and 16 GB RAM. In the NPCG method, the corrector works a total of 9 times. As the time of integration decreases, the computing time decreases. The computing time of NPCG is listed in Table 5, and the total computing time is 6.431 s. In the HERG method, the policy network is called 9 times and the total computing time is 0.0937 s.

The contrast simulation results are shown in Table 6. It is obvious that with similar terminal errors, the calculation time of the proposed method is less than 1/68 of NPCG. By contrast with NPCG, the proposed method makes a breakthrough in real-time performance.

*5.4. Verifications on Monte Carlo Simulations.* To evaluate the effectiveness of the proposed guidance, 1000 Monte Carlo simulations are performed, whose initial states and errors of aerodynamic coefficients are set according to Table 3. The results are shown in Figures 15–22.

As is seen in Figure 15, all of the 1000 trajectories of the glide phase are in the range of the $H$-$V$ corridor, and naturally, in Figures 16–19, all of the path constraints are satisfied. Moreover, the AOA and bank angle in different trajectories take on a continuous trend as is shown in Figures 17 and 18, which indicates that the output of the policy network is steady and robust.

The statistical data of terminal constraints is shown in Table 7.

The detailed data is shown in Figures 23–25.

Figure 23 shows the terminal time error and height error, which implies that all the time errors are in the required range of 20 s and all the height errors are in the required range of 2 km. Figures 24 and 25 show histograms of the terminal time error and height error, which implies that the proposed method can satisfy all the terminal constraints.

## 6. Conclusions

In this paper, an intelligent multiconstrained reentry guidance is developed to generate super real-time guidance commands with no need for a decoupling method. First, the magnitude and sign of the bank angle are combined as the sole guidance command rather than the form of longitudinal guidance and lateral guidance. Then, a policy network is constructed to output the guidance command with the help of DRL. Based on DDPG and HER, the terminal constraints are transformed into multigoals. The improved HER method shows more excellent performance in the training simulation. In simulations, the policy network in the proposed guidance can give commands online in less than 1/68 of NPCG time. The robustness to initial state bias and online aerodynamic errors is excellent. However, the deflection of bias and errors is limited in the range of the training phase. If online parameters are beyond the range, the policy network should be retrained. The relation between the scale of the network and the deflection range deserves further study, and the method proposed in this paper can be used as a reference.

## Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

## Conflicts of Interest

The authors declare that they have no conflicts of interest.

## Acknowledgments

This work was supported by the National Natural Science Foundation of China under Grant No. U20B2005.

## References

[1] Z. Shen and P. Lu, "Onboard generation of three-dimensional constrained entry trajectories," *Journal of Guidance, Control, and Dynamics*, vol. 26, no. 1, pp. 111–121, 2003.

[2] V. Morio, F. Cazaurang, and P. Vernis, "Flatness-based hypersonic reentry guidance of a lifting-body vehicle," *Control Engineering Practice*, vol. 17, no. 5, pp. 588–596, 2009.

[3] X. Yan, S. Lyu, and S. Tang, "Analysis of optimal initial glide conditions for hypersonic glide vehicles," *Chinese Journal of Aeronautics*, vol. 27, no. 2, pp. 217–225, 2014.

[4] Z. Guo, J. Guo, and J. Zhou, "Adaptive attitude tracking control for hypersonic reentry vehicles via sliding mode-based coupling effect-triggered approach," *Aerospace Science and Technology*, vol. 78, pp. 228–240, 2018.

[5] Y. Ding, X. Yue, G. Chen, and J. Si, "Review of control and guidance technology on hypersonic vehicle," *Chinese Journal of Aeronautics*, vol. 35, no. 7, pp. 1–18, 2022.

[6] S. Liu, B. Yan, T. Zhang, X. Zhang, and J. Yan, "Coverage-based cooperative guidance law for intercepting hypersonic vehicles with overload constraint," *Aerospace Science and Technology*, vol. 126, article 107651, 2022.

[7] W. Ma, W. Fu, Y. Fang, S. Liu, and X. Liang, "Prescribed-time cooperative guidance with time delay," *Aeronautical Journal*, pp. 1–24, 2022.

[8] S. Liu, B. Yan, T. Zhang, P. Dai, R. Liu, and J. Yan, "Three-dimensional cooperative guidance law for intercepting hypersonic targets," *Aerospace Science and Technology*, vol. 129, article 107815, 2022.

[9] S. Liu, B. Yan, T. Zhang, X. Zhang, and J. Yan, "Three-dimensional coverage-based cooperative guidance law with overload constraints to intercept a hypersonic vehicle," *Aerospace Science and Technology*, vol. 130, article 107908, 2022.

[10] W. Ma, X. Liang, Y. Fang, T. Deng, and W. Fu, "Three-dimensional prescribed-time pinning group cooperative guidance law," *International Journal of Aerospace Engineering*, vol. 2021, Article ID 4490211, 19 pages, 2021.

[11] J. C. Harpold and C. A. Graves, "Shuttle entry guidance," *American Astronautical Society*, vol. 27, no. 3, pp. 239–268, 1978.

[12] T. R. Jorris and R. G. Cobb, "Multiple method 2-D trajectory optimization satisfying waypoints and no-fly zone constraints," *Journal of Guidance, Control, and Dynamics*, vol. 31, no. 3, pp. 543–553, 2008.

[13] S. Josselyn and I. M. Ross, "Rapid verification method for the trajectory optimization of reentry vehicles," *Journal of Guidance, Control, and Dynamics*, vol. 26, no. 3, pp. 505–508, 2003.

[14] B. Tian and Q. Zong, "Optimal guidance for reentry vehicles based on indirect Legendre pseudospectral method," *Acta Astronautica*, vol. 68, no. 7–8, pp. 1176–1184, 2011.

[15] Y. Sun, M. Hou, G. Duan, and X. Liang, "On-line optimal autonomous reentry guidance based on improved Gauss pseudospectral method," *Science China Information Sciences*, vol. 57, no. 5, pp. 1–16, 2014.

[16] Y. Mao, D. Zhang, and L. Wang, "Reentry trajectory optimization for hypersonic vehicle based on improved Gauss pseudospectral method," *Soft Computing*, vol. 21, no. 16, pp. 4583–4592, 2017.

[17] H. Zhou, X. Wang, B. Bai, and N. Cui, "Reentry guidance with constrained impact for hypersonic weapon by novel particle

swarm optimization," *Aerospace Science and Technology*, vol. 78, pp. 205–213, 2018.

[18] D. J. Zhao and Z. Y. Song, "Reentry trajectory optimization with waypoint and no-fly zone constraints using multiphase convex programming," *Acta Astronautica*, vol. 137, pp. 60–69, 2017.

[19] G. A. Dukeman, "Profile-following entry guidance using linear quadratic regulator theory," in *AIAA Guidance, Navigation, and Control Conference and Exhibit*, Monterey, CA, USA, 2002.

[20] B. Tian, W. Fan, Q. Zong, J. Wang, and F. Wang, "Nonlinear robust control for reusable launch vehicles in reentry phase based on time-varying high order sliding mode," *Journal of the Franklin Institute*, vol. 350, no. 7, pp. 1787–1807, 2013.

[21] K. D. Mease and J. P. Kremer, "Shuttle entry guidance revisited using nonlinear geometric methods," *Journal of Guidance, Control, and Dynamics*, vol. 17, no. 6, pp. 1350–1356, 1994.

[22] X. Bu and Q. Qi, "Fuzzy optimal tracking control of hypersonic flight vehicles via single-network adaptive critic design," *IEEE Transactions on Fuzzy Systems*, vol. 30, no. 1, pp. 270–278, 2022.

[23] C. A. Kluever, "Entry guidance using analytical atmospheric skip trajectories," *Journal of Guidance, Control, and Dynamics*, vol. 31, no. 5, pp. 1531–1535, 2008.

[24] W. Yu and W. Chen, "Entry guidance with real-time planning of reference based on analytical solutions," *Advances in Space Research*, vol. 55, no. 9, pp. 2325–2345, 2015.

[25] W. Zhang, W. Chen, and W. Yu, "Analytical solutions to three-dimensional hypersonic gliding trajectory over rotating Earth," *Acta Astronautica*, vol. 179, pp. 702–716, 2021.

[26] R. D. Braun and R. W. Powell, "Predictor-corrector guidance algorithm for use in high-energy aerobraking system studies," *Journal of Guidance, Control, and Dynamics*, vol. 15, no. 3, pp. 672–678, 1992.

[27] A. Joshi, K. Sivan, and S. S. Amma, "Predictor-corrector reentry guidance algorithm with path constraints for atmospheric entry vehicles," *Journal of Guidance, Control, and Dynamics*, vol. 30, no. 5, pp. 1307–1318, 2007.

[28] P. Lu, "Entry guidance: a unified method," *Journal of Guidance, Control, and Dynamics*, vol. 37, no. 3, pp. 713–728, 2014.

[29] L. Cheng, F. Jiang, Z. Wang, and J. Li, "Multiconstrained real-time entry guidance using deep neural networks," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 57, no. 1, pp. 325–340, 2021.

[30] X. Wang, H. Peng, S. Zhang, B. Chen, and W. Zhong, "A symplectic pseudospectral method for nonlinear optimal control problems with inequality constraints," *ISA Transactions*, vol. 68, pp. 335–352, 2017.

[31] X. Wang, B. Li, X. Su et al., "Autonomous dispatch trajectory planning on flight deck: a search-resampling- optimization framework," *Engineering Applications of Artificial Intelligence*, vol. 119, article 105792, 2023.

[32] X. W. Wang, H. J. Peng, J. Liu, X. Z. Dong, X. D. Zhao, and C. Lu, "Optimal control based coordinated taxiing path planning and tracking for multiple carrier aircraft on flight deck," *Defence Technology*, vol. 18, no. 2, pp. 238–248, 2022.

[33] X. Wang, J. Liu, X. Su, H. Peng, X. Zhao, and C. Lu, "A review on carrier aircraft dispatch path planning and control on deck," *Chinese Journal of Aeronautics*, vol. 33, no. 12, pp. 3039–3057, 2020.

[34] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*, MIT Press, Cambridge, UK, 1998.

[35] V. Mnih, K. Kavukcuoglu, D. Silver et al., "Playing Atari with deep reinforcement learning," 2013, http://arxiv.org/abs/1312.5602.

[36] T. P. Lillicrap, J. J. Hunt, A. Pritzel et al., "Continuous control with deep reinforcement learning," 2019, http://arxiv.org/abs/1509.02971.

[37] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," 2017, http://arxiv.org/abs/1707.06347.

[38] S. Fujimoto, H. van Hoof, and D. Meger, "Addressing function approximation error in actor-critic methods," 2018, http://arxiv.org/abs/1802.09477.

[39] T. Haarnoja, A. Zhou, K. Hartikainen et al., "Soft actor-critic algorithms and applications," 2019, http://arxiv.org/abs/1812.05905.

[40] J. Yao, X. Li, Y. Zhang et al., "Three-dimensional path planning for unmanned helicopter using memory-enhanced dueling deep Q network," *Aerospace*, vol. 9, no. 8, p. 417, 2022.

[41] S. Liu, Z. Yang, Z. Zhang et al., "Application of deep reinforcement learning in reconfiguration control of aircraft anti-skid braking system," *Aerospace*, vol. 9, no. 10, p. 555, 2022.

[42] A. Scorsoglio, A. D'Ambrosio, L. Ghilardi, B. Gaudet, F. Curti, and R. Furfaro, "Image-based deep reinforcement meta-learning for autonomous lunar landing," *Journal of Spacecraft and Rockets*, vol. 59, no. 1, pp. 153–165, 2022.

[43] T. Schaul, D. Horgan, K. Gregor, and D. Silver, "Universal value function approximators," in *International conference on machine learning*, pp. 1312–1320, Lille, France, 2015.

[44] M. Andrychowicz, F. Wolski, A. Ray et al., "Hindsight experience replay," 2018, http://arxiv.org/abs/1707.01495.

[45] E. Prianto, M. Kim, J. H. Park, J. H. Bae, and J. S. Kim, "Path planning for multi-arm manipulators using deep reinforcement learning: soft actor-critic with hindsight experience replay," *Sensors*, vol. 20, no. 20, p. 5911, 2020.

[46] B. Manela and A. Biess, "Curriculum learning with hindsight experience replay for sequential object manipulation tasks," *Neural Networks*, vol. 145, pp. 260–270, 2022.