Chapter 20
Intelligent Reentry Guidance with Dynamic No-Fly Zones Based on Deep Reinforcement Learning

Qingji Jiang, Xiaogang Wang, and Yu Li

Abstract Aimed at avoiding multiple dynamic no-fly zones and satisfying path constraints and terminal constraints in the reentry process of hypersonic glide vehicles, intelligent reentry guidance based on deep reinforcement learning is developed. Firstly, the guidance is decoupled as longitudinal guidance and lateral guidance. The lateral guidance provides the sign of the bank angle to adjust the heading direction while the longitudinal guidance outputs the magnitude of the bank angle through the artificial intelligence interface. Then, the reentry guidance simulation is mapped to a Markov Decision Process, in which the essential elements including state, action, and reward are defined or designed adaptively. Finally, the policy neural network is trained by the twin delayed deep deterministic policy gradient (TD3) algorithm. By selecting proper hyperparameters and network architecture, the policy neural network is able to converge. Simulations imply that under the influence of dynamic no-fly zones, initial state errors, and kinds of online dispersion, the proposed guidance can avoid all the no-fly zones and reach the target accurately with all the satisfied path constraints.

Keywords Hypersonic glide vehicle · Reentry guidance · No-fly zones · Artificial intelligence · Deep reinforcement learning

20.1 Introduction

There has been increasing attention to hypersonic glide vehicles (HGV) due to their high speed, wide flight space, and strong maneuver capability [1–6]. After decades of development, reentry guidance has formed a relatively complete methodological

Q. Jiang · X. Wang (✉)
School of Astronautics, Harbin Institute of Technology, Harbin 150001, China
e-mail: wangxiaogang@hit.edu.cn

Y. Li
Beijing Aerospace Technology Institute, Beijing 100074, China

S. Li (ed.), *Computational and Experimental Simulations in Engineering*,
Mechanisms and Machine Science 143,
https://doi.org/10.1007/978-3-031-42515-8_20

system. For HGV, the task complexity and real-time performance are challenging in the future. For example, there are several dynamic no-fly zones whose information is not clear before the flight. HGV is requested to avoid all the no-fly zones and arrive at the specified target area under the premise of satisfying multiple path constraints and terminal constraints. With the help of artificial intelligence, HGV can fly out novel trajectories different from traditional algorithms and complete the mission.

The conventional guidance methods mainly include reference trajectory guidance [7–9] and predictor–corrector guidance [10–13]. For guidance issues with no-fly zones, current methods mainly include trajectory optimization, lateral guidance design, and route planning method. In trajectory optimization methods, the no-fly zones constraint is modeled in the optimization model and the problem is solved by off-line optimization algorithms. Zhao et al. [14] applied the Gauss pseudo-spectral method (GPM) to the multi-phase of the reentry problem and used waypoints to the complete optimization of the trajectory with no-fly zones. Zhao and Song [15] proposed a multiphase convex programming method for the path, waypoint, and no-fly zone constraints, and they solve the second-order conic problem (SOCP) problems with the help of the open-source solver ECOS. Zhang et al. [16] developed a time-optimal memetic whale optimization method based on GPM, which is excellent in both global searching and local convergent, and the simulation shows that the method is competitive in entry trajectory optimization with no-fly zones. The advantage of optimization methods is that by relying on strong search capability, the feasible solution is guaranteed if the scene parameters are in the range of dynamic ability. On the other hand, the computational complexity is commonly huge and not able to implement online with real-time performance.

There are many works based on lateral guidance design and route planning methods. Liang and Ren [17] presented a tentacle-based guidance method to satisfy the no-fly zone constraint, in which the sign of bank angle is determined by the feedback of tentacles. Gao et al. [18] proposed an improved tentacle-based bank angle transient lateral guidance method for avoiding static, dynamic, or unknown no-fly zones. Considering the concise mathematical expression and practicability, the artificial potential field (APF) method and its improved version are applied in reentry guidance with no-fly zone constraints. Zhang et al. [19] combined APF and velocity azimuth angle error threshold in lateral guidance to reduce heading error and avoid no-fly zones. Li et al. [20] proposed an improved APF method, in which the passing waypoints and avoiding no-fly zones problem is transformed into generating the reference heading angle. Li et al. [21] designed an adaptive cross corridor based on the concept of repulsion force in the APF method and the corridor is practicable for conventional guidance logic and avoiding no-fly zones logic. Hu et al. [22] presented an improved APF method for complex distributed no-fly zones, in which the reference heading angle is calculated according to geographic coordinate velocity and the designed potential field function. It can be seen that the methods based on APF are easy to achieve rapid real-time performance and robust to multiple complex no-fly zones. However, the design of the attractive and repulsive potential field is relevant to no-fly zones and other distances, which lacks robustness to unknown scenes and errors.

During recent decades, artificial intelligence technology has experienced rapid development and has been applied in lots of fields. More recently, deep reinforcement learning (DRL) shows excellent decision-making ability in complex high dimension tasks. DRL takes features of tasks as input and output decision results directly, naturally, the end-to-end characteristic makes it easy to handle different tasks. There are some applications of DRL in HGV or avoiding no-fly zones. For example, Yuksek et al. [23] used reinforcement learning and proposed a planning method for the unmanned aerial vehicle, which can avoid no-fly zones and ensure the time-of-arrival constraint.

In DRL algorithms, the AC (actor-critic) framework plays an important role, in which the policy from the actor is used to generate decision actions and the critic is in charge of evaluating the actions in the current state of the environment. Based on policy gradient theory, there is a family of progressive algorithms: deterministic policy gradient (DPG) [24], deep deterministic policy gradient (DDPG) [25], twin delayed deep deterministic policy gradient (TD3) [26], distributed distributional deterministic Policy Gradients (D4PG) [27]. Considering that the guidance command of HGV is generated according to its flight state, the guidance process can be seen as a decision-making mission. And the irreversibility of flight trajectory makes it easy to build a Markov decision process (MDP), that the problem can be solved by the TD3 algorithm.

The purpose of this paper is to develop an intelligent guidance method for reentry problems with several dynamic no-fly zones and constraints. The contributions of this paper can be summarized as follows: The reentry problem with several dynamic no-fly zones is described as an MDP. In MDP, the state is defined by parameters of HGV, the current no-fly zone, and the target. The guidance command of HGV is defined as the action of the agent, which can be seen as the output of a policy neural network and learned by training. Secondly, the action is trained by the TD3 algorithm. Finally, the converged policy network is invoked online with a tremendous real-time performance, which is an advantage in online guidance.

This paper is arranged as follows. Section 20.2 describes the reentry model with no-fly zones. Section 20.3 introduces the general principles of DRL and the TD3 algorithm. Section 20.4 proposes intelligent guidance based on TD3. Section 20.5 shows the training results and verifies the proposed method in simulations. Finally, the conclusions of this work are in Sect. 20.6.

20.2 Problem Model

20.2.1 Dynamics Equations in Reentry Process

Suppose the earth is a spherical non-rotating sphere, the dynamics equations in the reentry process are given by:

$$\begin{cases} \dot{V} = -\frac{D}{m} - g\sin\gamma \\ \dot{\gamma} = \frac{L\cos\sigma}{mV} - \left(\frac{g}{V} - \frac{V}{r}\right)\cos\gamma \\ \dot{\psi} = \frac{L\sin\sigma}{mV\cos\gamma} + \frac{V\cos\gamma\sin\psi\tan\phi}{r} \\ \dot{r} = V\sin\gamma \\ \dot{\theta} = \frac{V\cos\gamma\sin\psi}{r\cos\phi} \\ \dot{\phi} = \frac{V\cos\gamma\cos\psi}{r} \end{cases} \tag{20.1}$$

where V is the Earth-relative velocity, γ is the flight-path angle, ψ is the heading angle of velocity, r is the distance between the Earth center and HGV, θ is the longitude, ϕ is the latitude, m is the mass of HGV, g is the gravitational acceleration, σ is the bank angle. L and D represent the aerodynamic lift and drag respectively, which are expressed by

$$\begin{cases} D = \frac{1}{2}\rho V^2 C_D S_m \\ L = \frac{1}{2}\rho V^2 C_L S_m \end{cases} \tag{20.2}$$

where ρ is the atmospheric density, S_m is the reference area of HGV, C_D and C_L are the drag coefficient and lift coefficient respectively, which depend on the Mach number and angle of attack (AOA).

20.2.2 Constraints in Reentry Process

During the reentry flight, there are several hard path constraints: the maximum heating rate $\dot{Q}_{\max}$, the maximum dynamic pressure $q_{\max}$, and the maximum aerodynamic overload $n_{\max}$. HGV is required to satisfy these constraints:

$$\begin{cases} \dot{Q} = k_Q \rho^{0.5} V^{3.15} < \dot{Q}_{\max} \\ q = \frac{1}{2}\rho V^2 < q_{\max} \\ n = \frac{\sqrt{D^2+L^2}}{m} < n_{\max} \end{cases} \tag{20.3}$$

where $\dot{Q}$ is the heating rate, q is the dynamic pressure, n is the aerodynamic overload, k_Q is a constant.

Assume that no-fly zones are described as infinite-height cylinders with a central point (θ_i, φ_i) and radius R_i. Then the constraint of no-fly zones is expressed as:

$$S_i = R_e \arccos(\cos\phi\cos\phi_i\cos(\theta - \theta_i) + \sin\phi\sin\phi_i) > R_i + \Delta S \tag{20.4}$$

where S_i is the distance between HGV and the central point of the ith no-fly zone, R_e is the radius of the earth and ΔS is a safe threshold.

Terminal constraints include altitude, velocity, and distance to the target.

$$\begin{cases} \Delta H(t_f) = |H(t_f) - H^*| \le \Delta\tilde{H} \\ \Delta S(t_f) = |V(t_f) - V^*| \le \Delta\tilde{V} \\ s(t_f) \le s^* \end{cases} \tag{20.5}$$

where t_f represents the final flight time, H^*, V^*, and s^* are the required altitude, velocity, and distance respectively. In this paper, $\Delta\tilde{H} = 1000$ m, $\Delta\tilde{V} = 20$ m/s, $s^* = 300$ km.

20.2.3 Guidance Scheme

Longitudinal Guidance

In reentry guidance, the terminal constraints of altitude and velocity are combined as an energy-form variable e

$$e = \frac{1}{r} - \frac{V^2}{2\mu} \tag{20.6}$$

where μ is the Earth's gravitational constant. If the Earth rotation is ignored, e is monotonically increasing, which can be set as the termination condition of dynamics integration.

If the final height and velocity are determined, the terminal energy is determined:

$$e^* = \frac{1}{r^*} - \frac{V^{*2}}{2\mu} \tag{20.7}$$

The integration of dynamics will last until the termination condition is met: $e \ge e^*$.

During the reentry process, the trajectory is decided by the AOA α and the bank angle σ. Usually, the AOA profile is a piecewise linear function of velocity or energy. In this paper the AOA is expressed as:

$$\alpha = \begin{cases} \alpha_{\max} & V \ge \tilde{V} \\ \alpha_0 + \frac{\alpha_{\max} - \alpha_0}{V_1 - V_2}(V - V_2) & \tilde{V}_2 < V < \tilde{V} \\ \alpha_0 & V \le \tilde{V}_2 \end{cases} \tag{20.8}$$

where $\alpha_{\max}$ is the max AOA of HGV, α_0 is the AOA when the lift-drag radio gets the maximum value, $\tilde{V}_1$ and $\tilde{V}_2$ are designed values.

The purpose of longitudinal guidance is to generate the magnitude of the bank angle to satisfy the request for height, velocity, and other path constraints. In conventional guidance algorithms, the magnitude of the bank angle is updated iteratively to

make the final distance error $s(t_f)$ decrease to 0. In this paper, the magnitude of the bank angle is generated by the intelligent algorithm.

Lateral Guidance

The purpose of lateral guidance is to generate the sign of bank angle to make HGV avoid the no-fly zones and fly to the target. Hence, the lateral guidance in this paper is divided into two phases. In the first phase, there is a closest no-fly zone near the route of HGV, so lateral guidance is designed to avoid the no-fly zone. In the second phase, after all the no-fly zones are avoided, the lateral guidance is designed to satisfy the terminal constraints. Because the velocity and height constraints are satisfied by the termination condition of dynamics integration, the purpose of the second phase can be seen as to decrease the terminal range error. In the first phase, the relative position of the current no-fly zone is described in Fig. 20.1. The LOS (line of sight) angle of the ith no-fly zone ψ_i is calculated by:

$$\begin{aligned} \lambda_i &= \arccos(\sin\phi \sin\phi_i + \cos\phi \cos\phi_i \cos(\theta - \theta_i)) \\ \psi_i &= \arccos \frac{\sin\phi_i - \sin\phi \cos\lambda_i}{\cos\phi \sin\lambda_i} \end{aligned} \tag{20.9}$$

where λ_i is the geocentric angle between HGV and the ith no-fly zone.

In Fig. 20.1, the horizontal axis represents the east and the vertical axis represents the north. According to the direction of V, there are four areas: I, II, III, and IV. When HGV is in the II area, it should output a negative bank angle to decrease ψ and avoid the current no-fly zone quickly. Conversely, When HGV is in the III area, it should output a positive bank angle to increase ψ. When HGV is in the I and IV area, it should output a negative and positive bank angle respectively to increase the angle

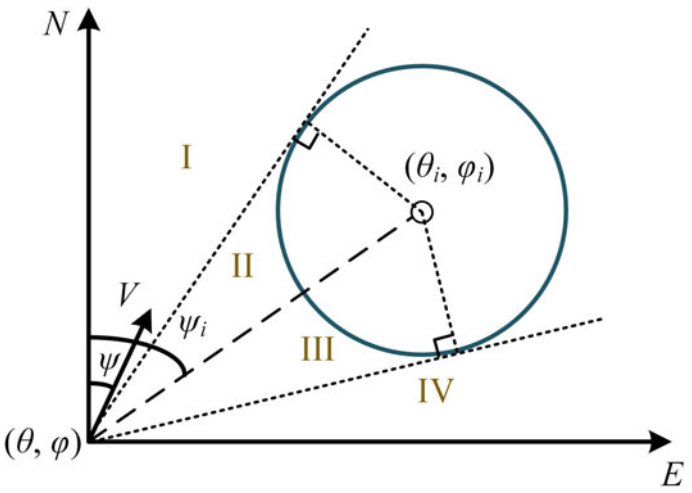

Fig. 20.1 HGV and current no-fly zone

between V and LOS direction $|\Delta\psi_i|$. There is a criterion for whether the current ith no-fly zone has been avoided:

$$|\Delta\psi_i| = |\psi - \psi_i| > 90^\circ \tag{20.10}$$

which means that when the HGV passes through the II area and enters the I area, the sign of the bank angle should keep minus until the criterion (20.10) is satisfied. Similarly, When HGV is in the III and IV area, it should output a positive bank angle.

Hence, in the first phase, the sign of the bank angle is decided by:

$$sign(\sigma) = \begin{cases} 1 & \psi > \psi_i \\ -1 & \psi \leq \psi_i \end{cases} \tag{20.11}$$

In the second phase, the LOS angle of target ψ_{tar} is expressed as:

$$\begin{aligned} \psi_{tar} &= \arccos \frac{\sin\phi_{tar} - \sin\phi\cos\lambda_{tar}}{\cos\phi\sin\lambda_{tar}} \\ \lambda_{tar} &= \arccos(\sin\phi\sin\phi_{tar} + \cos\phi\cos\phi_{tar}\cos(\theta - \theta_{tar})) \end{aligned} \tag{20.12}$$

where λ_{tar} is the geocentric angle between HGV and the target.

The sign of the bank angle is decided by the lateral corridor:

$$sign(\sigma) = \begin{cases} 1 & \Delta\psi > \Delta\psi_{up} \\ sign(\sigma) & \Delta\psi_{low} \leq \Delta\psi \leq \Delta\psi_{up} \\ -1 & \Delta\psi < \Delta\psi_{low} \end{cases} \tag{20.13}$$

where $\Delta\psi = \psi - \psi_{tar}$, is the heading error, $\Delta\psi_{up}$ and $\Delta\psi_{low}$ are the upper and lower bound:

$$\Delta\psi_{up} = \begin{cases} \psi_1 + \frac{\psi_2 - \psi_2}{V_2 - V_1}(V - V_1) & V_1 < V \leq V_2 \\ \psi_2 & V_2 < V \leq V_3 \\ \psi_2 + \frac{\psi_3 - \psi_2}{V_4 - V_3}(V - V_3) & V_3 < V \leq V_4 \end{cases} \tag{20.14}$$

where $V_1 = 2000$ m/s, $V_2 = 3500$ m/s, $V_3 = 6500$ m/s, $V_4 = 7000$ m/s, $\psi_1 = 2^\circ$, $\psi_1 = 2^\circ$, $\psi_1 = 10^\circ$. The corridor is shown in Fig. 20.2.

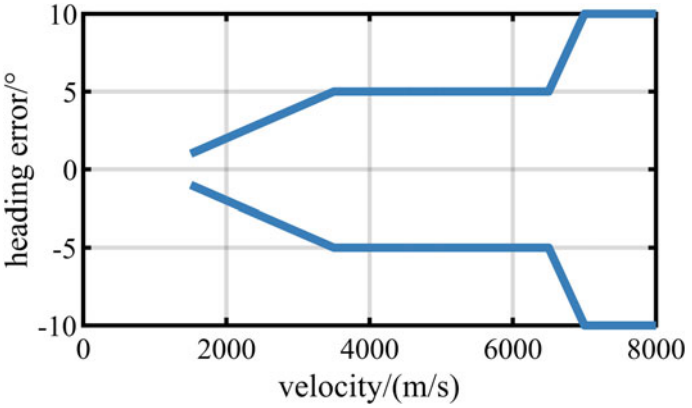

Fig. 20.2 Corridor of heading angle error

20.3 TD3 Algorithm

20.3.1 Deep Reinforcement Learning

Commonly the sequential decision-making problem can be modeled as a Markov Decision Process (MDP), in which there are elements including state, action, and reward. The object who makes decisions is called an agent and the agent is in a dynamic environment. The agent can interact with the environment State is a variable that can describe the features of the environment. The agent takes an action or decides according to the state. Then the environment is transformed from state S_1 to the next state S_2. At the same time, the agent gets a reward from the environment. The interaction progress will last until some termination condition is met. So there is a tuple $<S_i, \mathrm{A}_i, S_{i+1}, R_i>$ at every interaction time. For the agent, the goal is to maximize the total discounted reward in the whole process:

$$G_t = \sum_{k=0}^{\infty} \lambda^k R_{t+k+1} \tag{20.15}$$

where λ is the discount rate which determines the present value of future rewards.

In RL, the agent's policy π is a mapping from states to probabilities of selecting some possible action, and $\pi(a|s)$ means the probability that $A_t = a$ if $S_t = s$.

The action-value function for policy π is $q(s, a)$:

$$q_\pi(s, a) = E_\pi[G_t|S_t = s, A_t = a] = E_\pi\left[\sum_{k=0}^{\infty} \lambda^k R_{t+k+1}|S_t = s, A_t = a\right] \tag{20.16}$$

Similarly, the state-value function $v_\pi(s)$ is defined as:

$$v_\pi(s) = E_\pi[G_t|S_t = s] = E_\pi\left[\sum_{k=0}^{\infty} \lambda^k R_{t+k+1}|S_t = s\right] \tag{20.17}$$

There is an optimal action-value function $q^*(s, a)$ which satisfies that:

$$q^*(s, a) = \max_\pi q_\pi(s, a), \forall s \in \mathcal{S} \tag{20.18}$$

At this point, the policy π^* is called the optimal policy. In DRL, the policy is implemented by a neural network parameterized by θ^π. This is implemented by a neural network parameterized by θ^Q. The purpose of DRL training is to find the optimal parameters θ^π and θ^Q, which means that the best policy for the agent is found.

20.3.2 TD3 Algorithm

The baseline algorithm used in this paper for the neural network training is the twin delayed deep deterministic policy gradient (TD3), which is an improved version of the deep deterministic policy gradient (DDPG). In DDPG, there is a policy network actor parameterized by $\pi\ (s|\ \theta^\pi)$ and an evaluation network critic parameterized by $Q\ (s, a|\ \theta^Q)$. The input of the actor is the state of the environment s and it outputs the actions a. The input of the critic is the combination of s, a and it outputs the approximate action-value function $Q\ (s, a)$. And two target networks are designed to make the training process stable, the target actor network parameterized by $\pi'(s|\ \theta^{\pi'})$ and the target critic network parameterized by $Q'(s, a|\ \theta^{Q'})$.

The update of the critic is based on the gradient descent method. According to the Bellman Equations, the loss of critic $L(\theta^Q)$ is expressed as:

$$L(\theta^Q) = \mathbb{E}[(r(s_t, a_t) + \lambda Q'(s_{t+1}, \pi'(s_{t+1}|\theta^{\pi'})|\theta^{Q'}) - Q(s_t, a_t|\theta^Q))^2] \tag{20.19}$$

By updating the parameter θ^Q, the critic is closer and closer to the optimal $Q(s, a)$, which means that the evaluation of action is getting accurate gradually.

The actor $\pi(s, a|\theta^\pi)$ is updated according to the theory of policy gradient:

$$\begin{aligned} \nabla_{\theta^\pi} J &\approx \mathbb{E}_{s_t\sim\xi}[\nabla_{\theta^\pi} Q(s, a|\theta^Q)|_{s=s_t, a=\pi(s_t,\theta^\pi)}] \\ &= \mathbb{E}_{s_t\sim\xi}[\nabla_a Q(s, a|\theta^Q)_{s=s_t, a=\pi(s_t)} \nabla_{\theta^\pi} \pi(s|\theta^\pi)_{s=s_t}] \end{aligned} \tag{20.20}$$

where J is the objective to be optimized and ξ is the distribution of state.

Compared with DDPG, twin delayed deep deterministic policy gradient (TD3) has three improvements. First of all, TD3 provides two different critic networks including critic 1 parameterized by $Q_1(s, a|\theta^{Q_1})$ and critic 2 parameterized by $Q_2(s, a|\theta^{Q_2})$. In the training process, the smaller output value of critic 1 and critic 2 is set as the target Q value, which can overcome the overestimation of the Q value.

$$y = r(s_t, a_t) + \lambda \min\{Q_1'(s_{t+1}, \tilde{a}_{t+1}|\theta^{Q_1'}), Q_2'(s_{t+1}, \tilde{a}_{t+1}|\theta^{Q_2'})\} \tag{20.21}$$

where the next action is calculated by

$$\tilde{a}_{t+1} = \pi'(s_{t+1}|\theta^{\pi'}) + \varepsilon \tag{20.22}$$

where $\varepsilon \sim \text{clip}(\mathcal{N}(0, \tilde{\sigma}), -c, c)$ is the clipped noise in the range of $[-c, c]$, in which $\tilde{\sigma}$ is the variance of the noise.

The loss of critic 1 $L(\theta^{Q_1})$ and the loss of critic 2 $L(\theta^{Q_2})$ are

$$\begin{cases} L(\theta^{Q_1}) = \mathbb{E}[(y - Q(s_t, a_t|\theta^{Q_1}))^2] \\ L(\theta^{Q_2}) = \mathbb{E}[(y - Q(s_t, a_t|\theta^{Q_2}))^2] \end{cases} \tag{20.23}$$

Secondly, the update of policy is delayed, which means that TD3 updates critic networks more frequently than the actor and gets a higher quality policy update. The delayed update is meaningful because only if the critic is accurate, the improvement of policy is valuable.

Thirdly, TD3 adds a small amount of random noise to the target policy in Eq. (20.22), and the noise is clipped to keep the target close to the original action, in which way target policy smoothing is realized.

The three target networks are updated periodically:

$$\begin{cases} \theta^{\pi'} \leftarrow (1-\tau)\theta^{\pi} + \tau\theta^{\pi'} \\ \theta^{Q_1'} \leftarrow (1-\tau)\theta^{Q_1} + \tau\theta^{Q_1'} \\ \theta^{Q_2'} \leftarrow (1-\tau)\theta^{Q_2} + \tau\theta^{Q_2'} \end{cases} \tag{20.24}$$

where τ is the soft update factor.

20.4 Intelligent Guidance Law Based on TD3

20.4.1 Framework for Intelligent Guidance

In this section, the TD3-based guidance is proposed.

Firstly, the reentry including the no-fly zones process is normalized into two functions (scenario initialization function and policy cycle function) and the interfaces are open to the TD3 algorithm. In the scenario initialization function, the motion

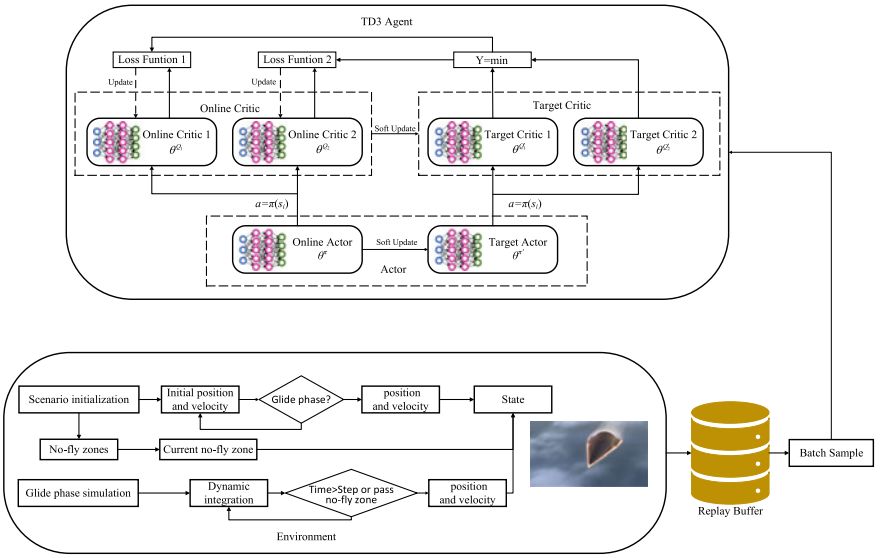

Fig. 20.3 Algorithm flow of intelligent guidance law

parameters of HGV are initialized randomly within a certain range. The information of N no-fly zones is also set randomly. At each policy step, HGV gets a magnitude command of the bank angle and generates the sign of the bank angle according to lateral guidance. The simulation proceeds until the energy $e > e^*$ or any path constraint is not satisfied. Then, the transformation from the reentry process 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. The reward function is designed according to whether avoiding the no-fly zones and arriving at the neighborhood of the target. Finally, based on TD3, the algorithm goes into operation as is shown in Fig. 20.3.

20.4.2 Markov Decision Process

The fundamental variables in MDP are defined as follows. Firstly, we divide the glide phase into two phases. In Phase I, there is one no-fly zone in HGV's flight path, which need to be avoided. In Phase II, HGV has passed through all the no-fly zones, so it needs to fix its path and approach the target.

(1) States

The basic state of the HGV agent in Phase I is defined as $s = s_{\mathrm{I}} = [V, \gamma, \psi, r, \theta, \phi, 1, \theta_{now}, \phi_{now}, R_{now}, V^*]$, where θ_{now}, ϕ_{now}, and R_{now} are the longitude, latitude, and geocentric distance of current no-fly zone, which can uniquely express the features

of the current situation. Considering dynamic no-fly zones, the information about no-fly zones is unknown before the flight.

Then, after the HGV has passed through all the no-fly zones, the state in Phase II is redefined as $s = s_{\mathrm{II}} = [V, \gamma, \psi, r, \theta, \phi, 0, \theta_{tar}, \phi_{tar}, H^*, V^*]$, where θ_{tar}, ϕ_{tar} are the longitude, latitude of the target. The design of s_{II} aims to guide the HGV to the target.

(2) Actions

Since the AOA is decided by profile and the sign of the bank angle is decided by lateral guidance, the action of the agent is mapped to the magnitude of the bank angle. So no matter what phase the HGV is in, the action is defined as $a \in [0, \sigma_{\max}]$, where $\sigma_{\max}$ is the maximum bank angle. Then the command of bank angle σ_{cmd} is obtained:

$$\sigma_{cmd} = sign(\sigma) \cdot a \tag{20.25}$$

where the sign of the bank angle $sign(\sigma)$ is given by lateral guidance.

(3) Rewards

The reward function plays a decisive role in guiding HGV to avoid the no-fly zones and achieve the target accurately. The reward function is shown as follows:

$$R_I = \begin{cases} 10, & |\psi - \psi_i| > 90^\circ \\ 0, & |\psi - \psi_i| \leq 90^\circ \end{cases} \tag{20.26}$$

where R_I is the no-fly-zone-related reward. The design of R_I means that when the HGV passes through the current no-fly zone, the agent will get a positive reward.

$$R_{\mathrm{II}} = \begin{cases} 50 + \frac{40-50}{300-0}(s(t_f) - 0), & s(t_f) \leq 300 \\ 40 + \frac{10-40}{500-300}(s(t_f) - 300), & 300 < s(t_f) \leq 500 \\ 10, & 500 < s(t_f) \leq 2000 \\ 1, & s(t_f) > 2000 \end{cases} \tag{20.27}$$

where $s(t_f)$ is the terminal distance error (unit km) and R_{II} is the target-related reward in the second phase. The piecewise linear functions are designed to guide the agent to reduce the final distance to the target. When HGV is in Phase I and Phase II, the reward is R_I and R_{II} respectively.

20.4.3 Steps of the Algorithm

Based on TD3, the intelligent reentry guidance is proposed as follows:

Reentry guidance with dynamic no-fly zones based on TD3

Randomly initialize actor network $\pi(s|\theta^{\pi})$ and critic networks $Q_1(s,a\,|\,\theta^{Q_1}), Q_2(s,a\,|\,\theta^{Q_2})$

Initialize target networks $\theta^{\pi'} \leftarrow \theta^{\pi}, \theta^{Q_1'} \leftarrow \theta^{Q_1}, \theta^{Q_2'} \leftarrow \theta^{Q_2}$, initialize replay buffer $\boldsymbol{R_B}$

for episode = 1~M do

  Initialize a random noise

  Run the Scene initialization Function, get the initial state $s = [V,\gamma,\psi,r,\theta,\phi,\theta_{now},\phi_{now},R_{now}]$

  for t=0~T do

    Sample an action from actor and noise: $a_t = \pi(s_t|\theta^{\pi}) + \mathcal{N}_t$

    Execute the action in the Policy Step Function and observe a new state s_{t+1}

    Store the transition tuple in $\boldsymbol{R_B}$

    Sample a random minibatch of N transitions from $\boldsymbol{R_B}$

    Add noise to target action $a' = \pi'(s'\,|\,\theta^{\pi'}) + \varepsilon, \varepsilon \sim clip((0,\tilde{\sigma}), -c, c)$

    Get minimum Q value $y = r(s_t,a_t) + \lambda \min\{Q_1'(s_{t+1},a'\,|\,\theta^{Q_1'}), Q_2'(s_{t+1},a'\,|\,\theta^{Q_2'})\}$

    Update parameters of the critic network $\theta^{Q_1} \leftarrow \arg\min_{\theta^{Q_1}} \sum[y - Q_1(s,a)]^2$

    Update parameters of the critic network $\theta^{Q_2} \leftarrow \arg\min_{\theta^{Q_2}} \sum[y - Q_2(s,a)]^2$

    If t mod d then

      Update parameters of the actor network by the policy gradient in Eq. (20)

      Update target networks periodically according to Eq. (24)

    End if

  End for

End for

20.4.4 Structures of Neural Networks

The structure of the actor is shown in Table 20.1 and the structure of the two critics is shown in Table 20.2.

Table 20.1 Structure of the actor

Layer number	Layer type	Nodes number	Activation function
1	Dense	11 × 300	Leaky ReLU
2	Dense	300 × 200	Leaky ReLU
3	Batch normalization	200	–
4	Dense	200 × 100	Leaky ReLU
5	Dense	100 × 1	–
6	Batch normalization	1	Tanh

Table 20.2 Structure of the critics

Layer number	Layer type	Nodes number	Activation function
1	Dense	12×300	Leaky ReLU
2	Dense	300×200	Leaky ReLU
3	Batch normalization	200	–
4	Dense	200×100	Leaky ReLU
5	Dense	100×1	Leaky ReLU

Table 20.3 Hyperparameters in the training

Number	Hyperparameter	Value
1	Discount factor	0.99
2	Batch size	128
3	Replay buffer size	5000
4	Learning rate	10^{-5}
5	Target update rate	0.001

20.5 Verification and Simulation

20.5.1 Parameters Settings

The hardware device used in the simulation is Intel-i5 CPU, RTX 3060Ti GPU, and 16GB RAM. The software used for training is PyTorch and Python. The hyperparameters in the training are given in Table 20.3.

HGV parameters are set according to the common aero vehicle (CAV-H). The mentioned parameters in simulation are set as follows: $m = 907\ kg$, $g = 9.8066\ \text{m/s}^2$, $S_m = 0.4839\ \text{m}^2$, $\rho_0 = 1.225\ \text{kg/m}^3$, $\beta = 0.000141$, $R_e = 6378004\ \text{m}$, $k_Q = 5 \times 10^{-5}$, $q_{\max} = 100\ \text{kPa}$, $n_{\max} = 3$, $\dot{Q}_{\max} = 2000\ \text{kW/m}^2$, $V_{re} = 2500\ \text{m/s}$, $\Delta S = 1000\ \text{m}$, $\alpha_{\max} = 20°$, $\alpha_0 = 10°$, $\tilde{V}_1 = 5000\ \text{m/s}$, $\tilde{V}_2 = 3000\ \text{m/s}$, $\sigma_{\max} = 85°$. The number of no-fly zones is 3. The integration step size is 0.01 s and the guidance (policy) step size is 200 s.

The random parameters used in simulations are shown in Table 20.4.

20.5.2 Training Result of Policy Network

After the training of 6665 episodes, the policy network converges. The average returns and success rates in the latest 100 episodes are shown in Fig. 20.4.

Table 20.4 Range of random parameters in simulations

Parameters	Unit	Max value	Min value
Initial HGV longitude θ_0	°	5	0
Initial HGV latitude ϕ_0	°	10	−10
Initial HGV velocity V_0	m/s	7100	6800
Required terminal velocity V^*	m/s	2600	2400
Initial HGV height H_0	Km	67	65
Initial HGV path angle γ_0	°	−0.001	−0.1
Target longitude θ_{tar}	°	75	65
Target latitude ϕ_{tar}	°	5	−5
Target height H_{tar}	Km	31	26
Drag coefficient error ΔC_D	–	5%	0%
Lift coefficient error ΔC_L	–	5%	0%

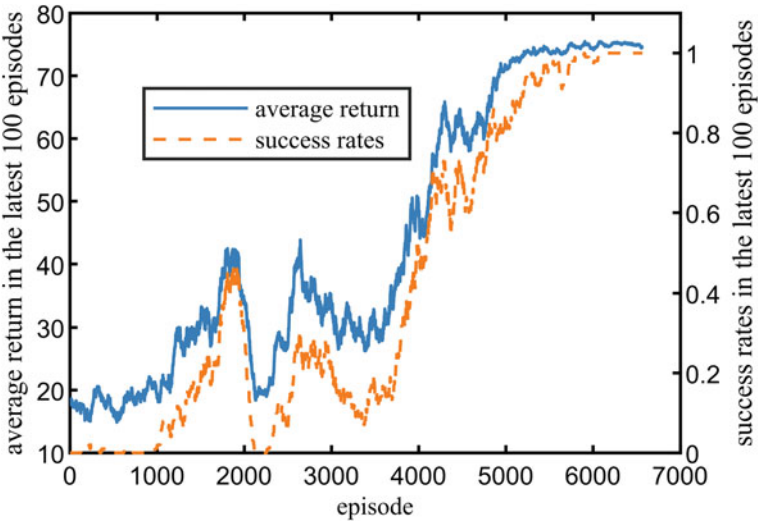

Fig. 20.4 Average return and success rates in the latest 100 episodes

At the end of the training process, the success rates reach and stabilize at 100%, which means that the policy network can output stable and valid commands. The average return fluctuates slightly in the range of 70–80, which coincides with the design of reward in Eqs. (20.26) and (20.27).

20.5.3 Verifications on Random Trajectories

The well-trained policy network is verified in random simulation.

(1) Scene 1.

The parameters of HGV and target are: $\theta_0 = 1.270°$, $\phi_0 = -3.734°$, $V_0 = 7000$ m/s, $H_0 = 65$ km, $\gamma = -0.1°$, $\psi_0 = 86.286°$, $\theta_{tar} = 71.163°$, $\phi_{tar} = 2.212°$, $H_{tar} = 30.020$ km, $V^* = 2482.800$ m/s. The parameters of three no-fly zones are listed in Table 20.5.

The simulation results are shown in Figs. 20.5, 20.6, 20.7 and 20.8.

(2) Scene 2.

The parameters of HGV and target are: $\theta_0 = 3.059°$, $\phi_0 = 3.008°$, $V_0 = 6823.816$ m/s, $H_0 = 65.891$ km, $\gamma_0 = -0.1°$, $\psi_0 = 95.121°$, $\theta_{tar} = 69.605°$, $\phi_{tar} = -3.514°$, $H_{tar} = 28.203$ km, $V^* = 2545.289$ m/s. The parameters of three no-fly zones are listed in Table 20.6.

Table 20.5 Parameters of no-fly zones in scene 1

Serial number	Center longitude (°)	Center latitude (°)	Radius (km)
1	34.928	0.599	274.823
2	48.809	−6.812	268.502
3	59.079	−3.261	316.207

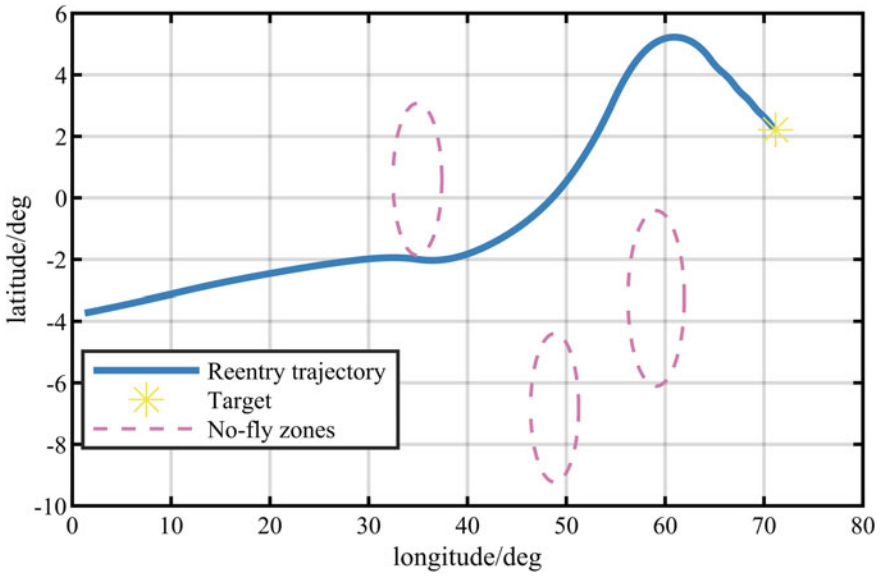

Fig. 20.5 Ground track of HGV in scene 1

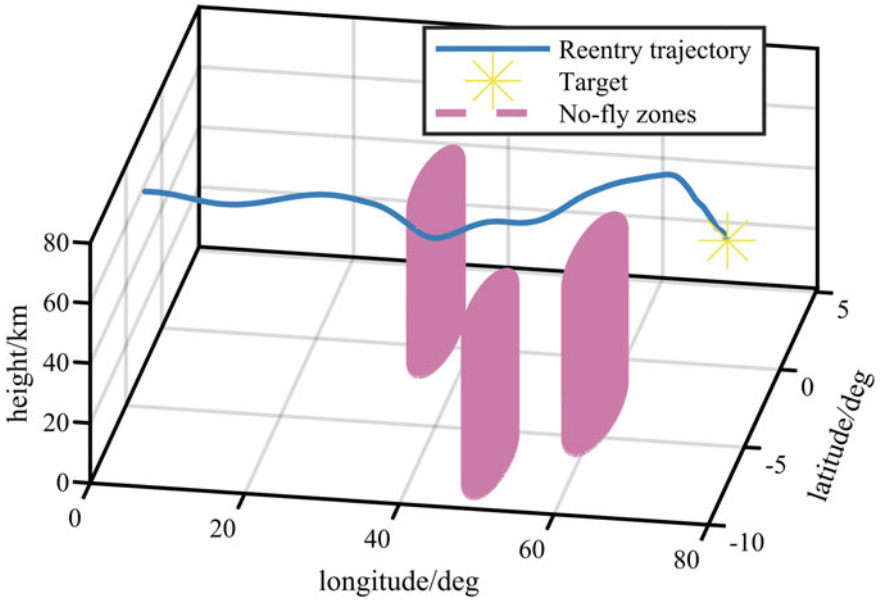

Fig. 20.6 Space trajectory of HGV in scene 1

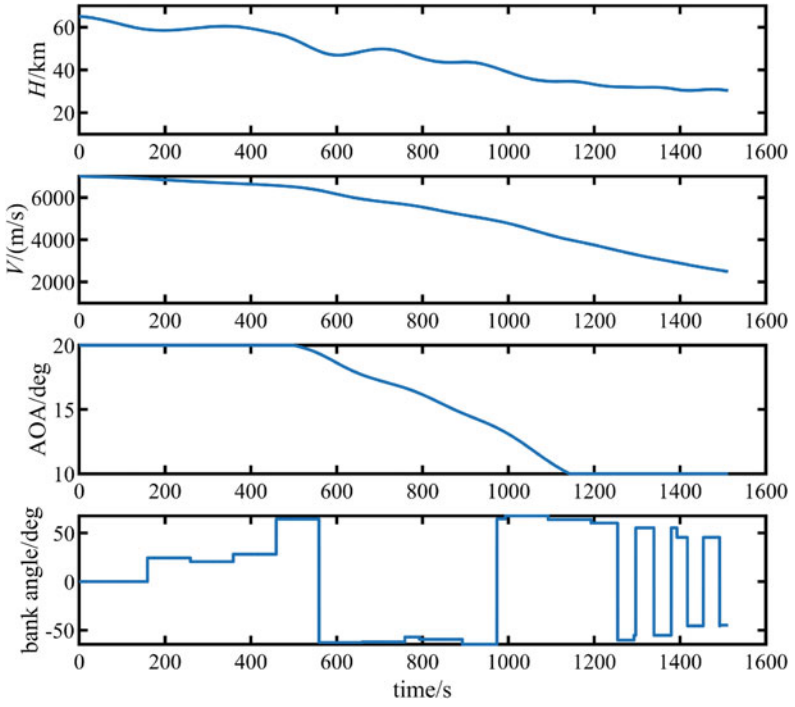

Fig. 20.7 States curves of HGV in scene 1

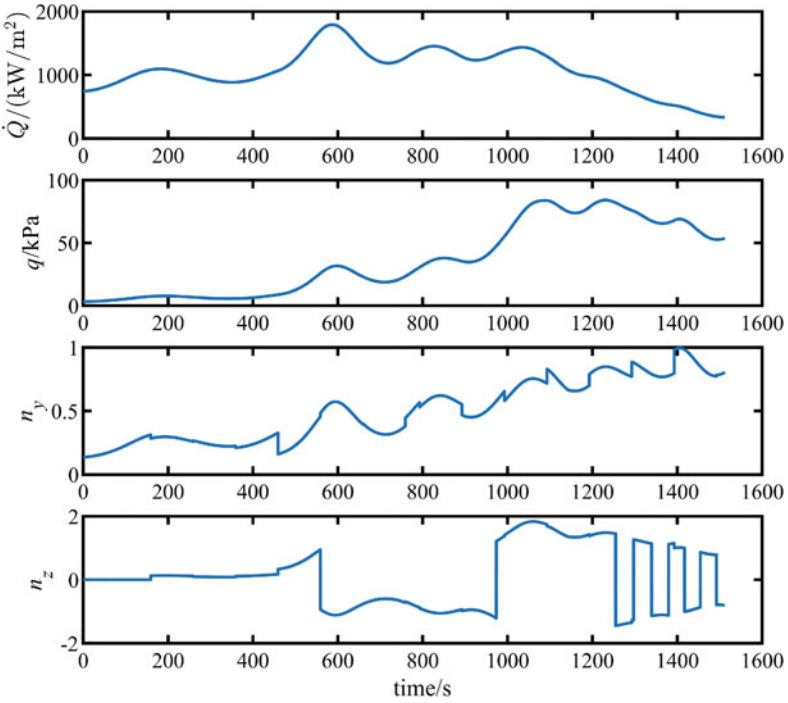

Fig. 20.8 Path constraints curves of HGV in scene 1

Table 20.6 Parameters of no-fly zones in scene 2

Serial number	Center longitude (°)	Center latitude (°)	Radius (km)
1	31.691	−3.667	285.678
2	44.948	3.199	224.935
3	57.354	−5.675	252.818

The simulation results are shown in Figs. 20.9, 20.10, 20.11 and 20.12.

We can see from Fig. 20.5, 20.6, 20.7, 20.8, 20.9, 20.10, 20.11 and 20.12 that in the two scenes, HGV can pass through all the dynamic no-fly zones and reach the target. In the flight process, all the path constraints are satisfied. The terminal errors of constraints are shown in Table 20.7.

Under the influence of initial parameter perturbation and aerodynamic deviations, HGV agent can satisfy all the constraints and avoid dynamic no-fly zones.

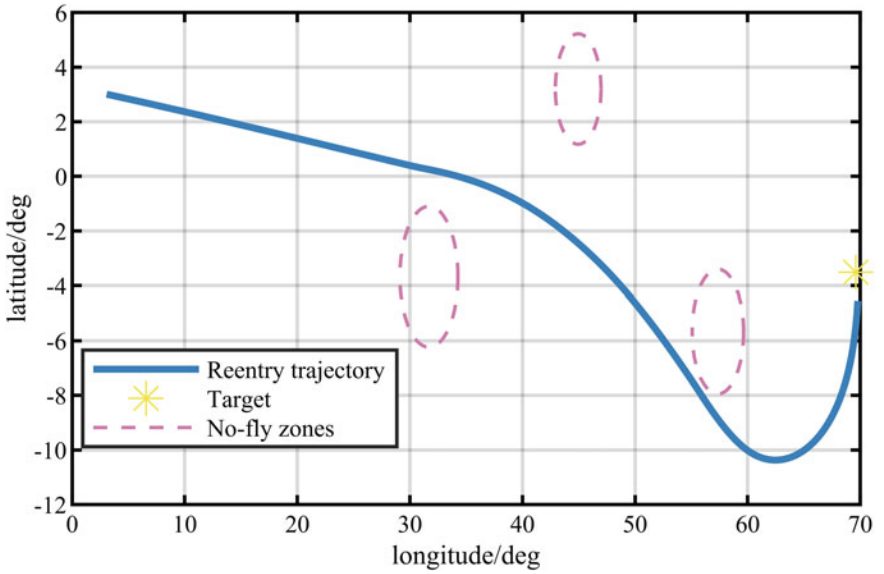

Fig. 20.9 Ground track of HGV in scene 2

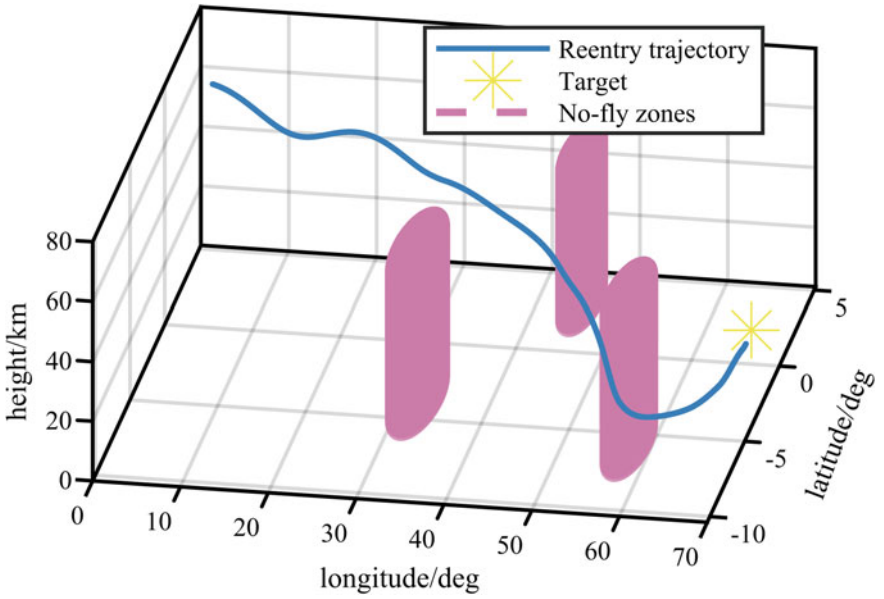

Fig. 20.10 Space trajectory of HGV in scene 2

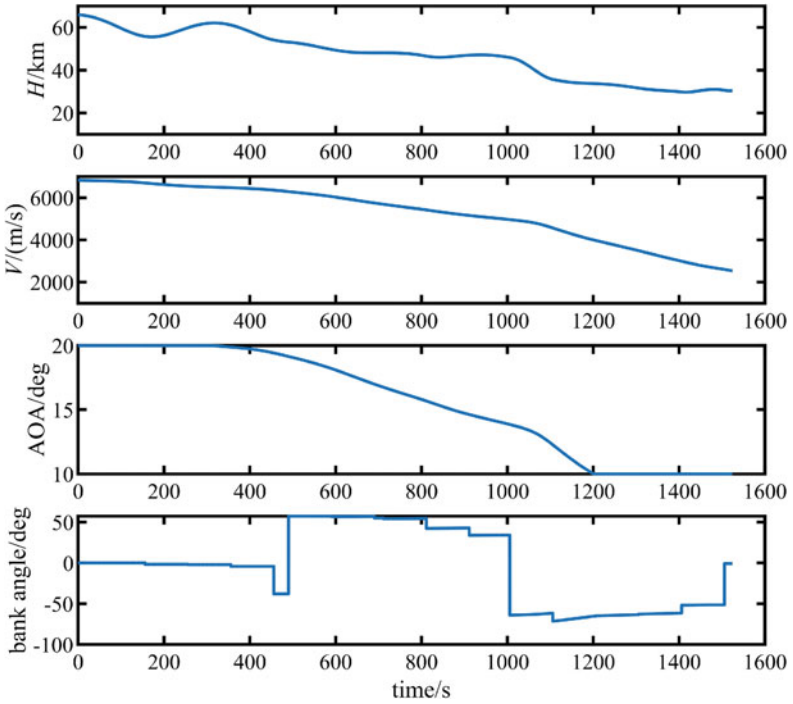

Fig. 20.11 States curves of HGV in scene 2

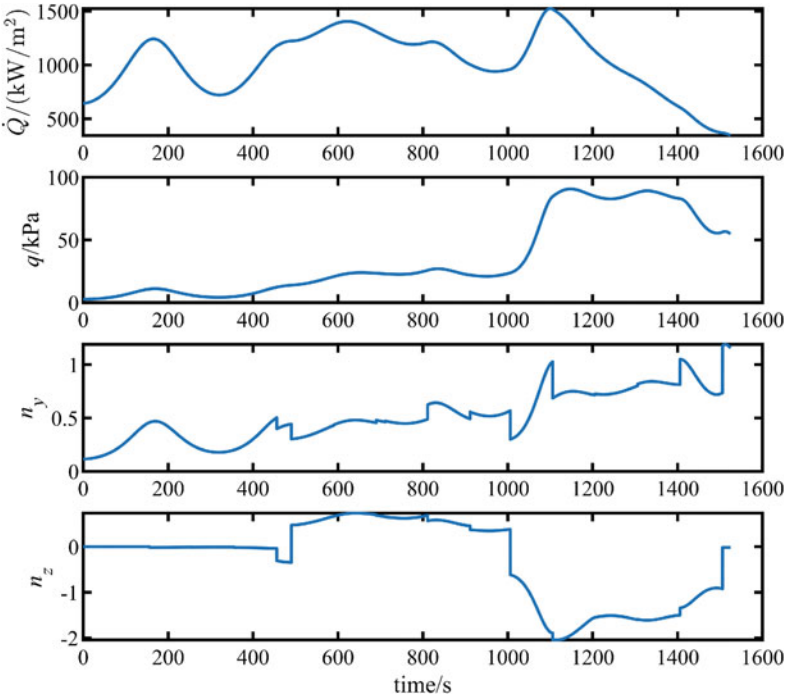

Fig. 20.12 Path constraints curves of HGV in scene 2

Table 20.7 Terminal errors in the scenes

Terminal errors	Scene 1	Scene 2
Height error $\Delta H(t_f)$	257.283 m	701.66 m
Velocity error $\Delta V(t_f)$	1.008 m/s	2.690 m/s
Distance error $s(t_f)$	166.830 km	116.250 km

20.6 Conclusions

Based on deep reinforcement learning, an intelligent method for reentry guidance with dynamic no-fly zones is studied in the paper. First of all, the mathematical model of HGV is established. Facing the dynamic no-fly zones, the reentry process of HGV is divided into two phases and the guidance scheme is given accordingly. Then, the problem is transformed into a Markov decision process, where the action is used to output guidance commands. State and reward are designed according to the flight phase. With the help of the TD3 algorithm, the policy network is trained to converge. Finally, the policy network is verified on random trajectories and proved to be robust to dynamic parameters of no-fly zones and other deviations.

References

1. Shen, Z., Lu, P.: Onboard generation of three-dimensional constrained entry trajectories. J. Guid. Control. Dyn. **26**(1), 111–121 (2003)
2. Zhao, J., Zhou, R.: Pigeon-inspired optimization applied to constrained gliding trajectories. Nonlinear Dyn. **82**(4), 1781–1795 (2015)
3. Zhu, J., He, R., Tang, G., Bao, W.: Pendulum maneuvering strategy for hypersonic glide vehicles. Aerosp. Sci. Technol. **78**, 62–70 (2018)
4. Ding, Y., Yue, X., Liu, C., Dai, H., Chen, G.: Finite-time controller design with adaptive fixed-time anti-saturation compensator for hypersonic vehicle. ISA Trans. **122**, 96–113 (2022)
5. Yu, J., Dong, X., Li, Q., Ren, Z., Lv, J.: Cooperative guidance strategy for multiple hypersonic gliding vehicles system. Chin. J. Aeronaut. **33**(3), 990–1005 (2020)
6. Ding, Y., Yue, X., Chen, G., Si, J.: Review of control and guidance technology on hypersonic vehicle. Chin. J. Aeronaut. **35**(7), 1–18 (2022)
7. Zhou, H., Wang, X., Bai, B., Cui, N.: Reentry guidance with constrained impact for hypersonic weapon by novel particle swarm optimization. Aerosp. Sci. Technol. **78**, 205–213 (2018)
8. Bu, X., Qi, Q.: Fuzzy optimal tracking control of hypersonic flight vehicles via single-network adaptive critic design. IEEE Trans. Fuzzy Syst. **30**(1), 270–278 (2022)
9. Yu, W., Chen, W.: Entry guidance with real-time planning of reference based on analytical solutions. Adv. Space Res. **55**(9), 2325–2345 (2015)
10. Zhang, W., Chen, W., Yu, W.: Analytical solutions to three-dimensional hypersonic gliding trajectory over rotating Earth. Acta Astronaut. **179**, 702–716 (2021)
11. Yu, W., Yang, J., Chen, W.: Entry guidance based on analytical trajectory solutions. IEEE Trans. Aerosp. Electron. Syst. **58**(3), 2438–2466 (2022)
12. Lu, P.: Entry guidance: a unified method. J. Guid. Control. Dyn. **37**(3), 713–728 (2014)
13. Lu, P., Brunner, C.W., Stachowiak, S.J., Mendeck, G.F., Tigges, M.A., Cerimele, C.J.: Verification of a fully numerical entry guidance algorithm. J. Guid. Control. Dyn. **40**(2), 230–247 (2017)
14. Zhao, J., Zhou, R., Jin, X.: Reentry trajectory optimization based on a multistage pseudospectral method. Sci. World J. **2014**, 1–13 (2014)
15. Zhao, D.J., Song, Z.Y.: Reentry trajectory optimization with waypoint and no-fly zone constraints using multiphase convex programming. Acta Astronaut. **137**, 60–69 (2017)
16. Zhang, H., Wang, H., Li, N., Yu, Y., Su, Z., Liu, Y.: Time-optimal memetic whale optimization algorithm for hypersonic vehicle reentry trajectory optimization with no-fly zones. Neural Comput. Appl. **32**(7), 2735–2749 (2020)
17. Liang, Z., Ren, Z.: Tentacle-based guidance for entry flight with no-fly zone constraint. J. Guid. Control. Dyn. **41**(4), 996–1005 (2018)
18. Gao, Y., Cai, G., Yang, X., Hou, M.: Improved tentacle-based guidance for reentry gliding hypersonic vehicle with no-fly zone constraint. IEEE Access **7**, 119246–119258 (2019)
19. Zhang, D., Liu, L., Wang, Y.: On-line reentry guidance algorithm with both path and no-fly zone constraints. Acta Astronaut. **117**, 243–253 (2015)
20. Li, Z., Yang, X., Sun, X., Liu, G., Hu, C.: Improved artificial potential field based lateral entry guidance for waypoints passage and no-fly zones avoidance. Aerosp. Sci. Technol. **86**, 119–131 (2019)
21. Li, M., Zhou, C., Shao, L., Lei, H., Luo, C.: An improved predictor-corrector guidance algorithm for reentry glide vehicle based on intelligent flight range prediction and adaptive crossrange corridor. Int. J. Aerosp. Eng. **2022**, 1–18 (2022)
22. Hu, Y., Gao, C., Li, J., Jing, W., Chen, W.: A novel adaptive lateral reentry guidance algorithm with complex distributed no-fly zones constraints. Chin. J. Aeronaut. **35**(7), 128–143 (2022)
23. Yuksek, B., Umut Demirezen, M., Inalhan, G., Tsourdos, A.: Cooperative planning for an unmanned combat aerial vehicle fleet using reinforcement learning. J. Aerosp. Inform. Syst. **18**(10), 739–750 (2021)

24. Silver, D., Lever, G., Heess, N., Degris, T., Wierstra, D., Riedmiller, M.: Deterministic policy gradient algorithms. In: International Conference on Machine Learning, pp. 387–395. PMLR (2014)
25. Lillicrap, T.P., Hunt, J.J., Pritzel, A., Heess, N., Erez, T., Tassa, Y., Silver, D., Wierstra, D.: Continuous control with deep reinforcement learning (2019). arXiv:1509.02971
26. Fujimoto, S., van Hoof, H., Meger, D.: Addressing function approximation error in actor-critic methods (2018). arXiv:1802.09477
27. Barth-Maron, G., Hoffman, M.W., Budden, D., Dabney, W., Horgan, D., T.B., Muldal, A., Heess, N., Lillicrap, T.: Distributed distributional deterministic policy gradients (2018).arXiv:1804.08617