SCIENCE CHINA
Information Sciences

• RESEARCH PAPER •

March 2023, Vol. 66 132202:1–132202:16
https://doi.org/10.1007/s11432-022-3543-4

Multi-constrained intelligent gliding guidance via optimal control and DQN

Jianwen ZHU[1*], Hao ZHANG[1], Sibo ZHAO[1] & Weimin BAO[1,2]

[1]*School of Aerospace Science and Technology, Xidian University, Xi'an* 710126*, China;*
[2]*China Aerospace Science and Technology Corporation, Beijing* 100048*, China*

Received 26 April 2022/Revised 22 May 2022/Accepted 17 June 2022/Published online 31 January 2023

Abstract In order to improve the adaptability and robustness of gliding guidance under complex environments and multiple constraints, this study proposes an intelligent gliding guidance strategy based on the optimal guidance, predictor-corrector technique, and deep reinforcement learning (DRL). Longitudinal optimal guidance was introduced to satisfy the altitude and velocity inclination constraints, and lateral maneuvering was used to control the terminal velocity magnitude and position. The maneuvering amplitude was calculated by the analytical prediction of the terminal velocity, and the direction was learned and determined by the deep Q-learning network (DQN). In the direction decision model construction, the state and action spaces were designed based on the flight status and maneuvering direction, and a reward function was proposed using the terminal predicted state and terminal constraints. For DQN training, initial data samples were generated based on the heading-error corridor, and the experience replay pool was managed according to the terminal guidance error. The simulation results show that the intelligent gliding guidance strategy can satisfy various terminal constraints with high precision and ensure adaptability and robustness under large deviations.

Keywords gliding flight, optimal guidance, velocity control, deep reinforcement learning, intelligent decision

Citation Zhu J W, Zhang H, Zhao S B, et al. Multi-constrained intelligent gliding guidance via optimal control and DQN. Sci China Inf Sci, 2023, 66(3): 132202, https://doi.org/10.1007/s11432-022-3543-4

1 Introduction

Hypersonic vehicles have been a research hotspot for the past 15 years. Gliding guidance with the constraints of terminal latitude, longitude, altitude, velocity magnitude, and inclination is a key technology for hypersonic vehicles [1]. Traditional standard trajectory tracking guidance [2], predictor-corrector guidance [3], and quasi-equilibrium gliding guidance [4] are mainly organized based on flight dynamics and control theory and can solve the gliding guidance problem under a single mission and standard condition. However, in the face of various flight missions, large deviations of the environment and vehicle body make the above methods redesign the standard trajectory or adjust the guidance parameters manually, which is inconvenient in actual flight processes. Therefore, improving adaptability and robustness is a key problem to be solved in gliding guidance, and intelligent control is an important technical approach.

Artificial intelligence (AI) based on machine learning is a hot topic in current research. As an algorithm that embodies intelligent decision-making, reinforcement learning (RL) has been recognized by many scholars [5]. RL selects actions that act on the environment, iterations, trials, and errors to obtain the greatest benefits [6]. Q-learning is a typical RL method and has been a preliminary research subject in the field of path planning and parameter determination [7]. Ref. [8] investigated a high-order RL problem for both simulations and real flight tests. In this problem, a quadrotor performs the task of taking pictures of a disaster site, whereas the environment is completely unknown at first. The quadrotor must learn the interest location and the most efficient way to get there. As for the intercept guidance problem, Gaudet et al. [9] employed RL to learn a homing-phase guidance law that is optimal with respect to the missile's

* Corresponding author (email: zhujianwen1117@163.com)

airframe dynamics, sensor and actuator noise, and delays. Aiming at the gliding guidance problem of hypersonic vehicles, Luo et al. [10] established a three-dimensional gliding guidance model and discrete reinforcement learning model, and used a Q-table to determine the lateral maneuver amplitude of velocity control. The above research results explore intelligent guidance using traditional discrete reinforcement learning, which mainly solves decision-making problems with low dimensions of the state space and action space.

As is well known, both flight states and guidance commands are time-continuous variables, so the discretization of states and actions in RL will inevitably affect decision-making accuracy and efficiency. A common improvement method of deep reinforcement learning (DRL) currently uses a deep neural network (DNN) to represent the value function in Q-learning; the input information is the state and action, and the output information is the value function [11, 12]. This method can effectively use the generalization ability of neural networks to solve decision-making problems with high-dimensional input and low-dimensional output [13]. Hovell et al. [13] introduced a novel deep learning guidance method, which consists of a learned guidance strategy that feeds velocity commands to a conventional controller to track. Control theory was combined with DRL to lower the learning burden and facilitate the transfer of the trained system from simulation to reality. Furthermore, Hovell et al. [14] employed DRL to design a spacecraft docking controller when the target aircraft was fixed and rotated in a two-dimensional plane. However, fuel consumption and arrival time constraints were not considered. In [15], RL was utilized to generate reference bank angle commands for directing the aircraft within close proximity of the updraft, and the problem of online trajectory generation was reduced to a simple search in a static "state-action" value table. DRL has also been applied in aerospace applications, mostly in simulations. A simulated fleet of a wildfire surveillance vehicle used DRL to control the flight path of the vehicle [16]. In addition, reinforcement meta-learning is a new type of RL. For the exo-atmospheric interception of maneuvering targets that only have a line-of-sight angle and rate information, reinforcement meta-learning is employed to optimize the policy to adapt to the target acceleration, and the policy has superior performance when compared to the augmented zero-effort misguidance with perfect target acceleration knowledge [17]. Furthermore, Gao et al. [18] designed an intelligent controller via DRL to reduce steady-state error, and the outstanding dynamic performance of the controller was demonstrated by comparing it with a linear matrix inequality controller. The above research results demonstrate the application of DRL in various types of vehicle guidance and control, but rarely involve long-period gliding guidance problems, lack of deep Q-learning network (DQN) initialization design, experience replay pool management, and training process optimization.

Aiming at the gliding guidance problem of long-distance flight with complex missions and environments, the study draws on the "offline training + online use" model, and designs a multi-constrained intelligent guidance strategy based on the optimal guidance, predictor-corrector technique, and DQN. This study uses the analytical optimal guidance to satisfy the terminal altitude and velocity inclination and controls the terminal velocity and position by lateral maneuvering. The maneuvering amplitude was determined by analytical prediction, and the direction was determined by DQN. In DQN training, the reward function is designed by the analytical prediction of terminal states, the initial data samples are generated using the heading-error corridor, and the experience replay pool is managed based on the terminal error. By combining the maneuvering direction decision based on the trained DQN, the optimal guidance, and the maneuvering amplitude solution, multi-constrained gliding guidance is then realized.

2 Intelligent gliding guidance statement

2.1 Intelligent gliding guidance strategy

Gliding guidance needs to control the vehicle to achieve long-distance flight safely and stably while satisfying the requirements of terminal latitude, longitude, altitude, velocity magnitude, and inclination with high precision. The gliding flight is mainly in a complex and changeable near space, and there are large deviations in the flight environment and aerodynamic coefficients. In addition, gliding vehicles are faced with complex flight missions, and even flight missions are changed online. Therefore, ensuring the accuracy of terminal guidance, improving the robustness of complex deviations, and the adaptability of multiple tasks are key issues that must be resolved in gliding guidance. In view of the aforementioned gliding guidance problem, the intelligent guidance strategy shown in Figure 1 is proposed.

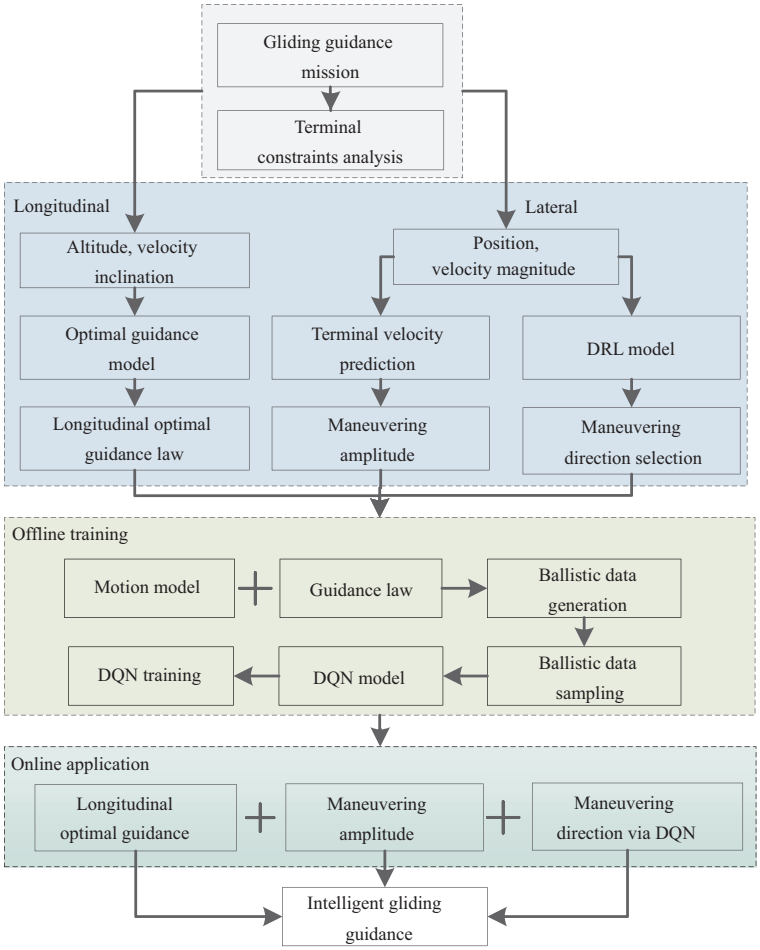

Figure 1 (Color online) Block diagram of intelligent gliding guidance strategy.

For the guidance strategy, the optimal guidance, predictor-corrector technique, and DRL are combined to achieve the guidance goal. The analytical optimal guidance method studied in the previous research was used to control the velocity direction to meet the constraints of terminal altitude and velocity inclination. In addition, lateral maneuvering is used to control the velocity and position, where the maneuvering amplitude is calculated by analytical prediction, and the direction is learned and determined by the DRL.

Maneuvering direction decision via DRL is the core section of this paper, which needs to take the multi-dimensional continuous actual flight states as the inputs and the finite-dimensional maneuvering direction as the output. Therefore, this paper introduces DRL to realize intelligent decision-making. The DRL decision-making model construction includes a state space based on real-time flight status, an action space with maneuvering direction, and a reward function that integrates the terminal position accuracy and velocity error. Using the ballistic data obtained from numerical integration, the DQN parameters were trained using the gradient descent method.

Finally, the optimal guidance, predictor-corrector technique, and trained DQN were comprehensively used to generate intelligent gliding guidance commands.

2.2 Optimal gliding guidance

In a previous study, we established a performance index with minimum energy consumption and designed optimal parameters in the longitudinal and lateral directions that can satisfy the constraints of the terminal latitude, longitude, altitude, and velocity inclination. The guidance law, expressed as the

required overload, is given as follows [19]:

$$\begin{cases} u_y^* = \frac{g}{v^2}\left(C_h L_R - C_\theta\right) + 1, \\ u_z^* = \frac{v^2(\sigma_{\rm LOS} - \sigma_v)}{g(L_{Rf} - L_R)}, \end{cases} \tag{1}$$

where $u_y^* = n_y$ and $u_z^* = n_z$ are the optimal guidance laws, also called the required overloads, in the longitudinal and lateral directions, respectively. g is the Earth's gravitational acceleration at the current altitude, v is the velocity magnitude, L_R is the range from the initial point to the current position, and L_{Rf} is the total range of the gliding phase. σ_v and $\sigma_{\rm LOS}$ are the velocity azimuth angle and line-of-sight (LOS) angle measured from the north in a clockwise direction, respectively. The LOS angle can be computed according to the current position $P(\lambda, \phi)$ and target $T(\lambda_f, \phi_f)$.

$$\tan\sigma_{\rm LOS} = \frac{\sin\left(\lambda_f - \lambda\right)}{\cos\phi\tan\phi_f - \sin\phi\cos\left(\lambda_f - \lambda\right)}, \tag{2}$$

where λ is the longitude, ϕ is the latitude, and the subscript f represents the terminal value. The optimal guidance requirements C_h and C_θ are obtained based on optimal control theory [19].

$$\begin{cases} C_h = \frac{6((L_R - L_{Rf})(\theta_f + \theta) - 2h + 2h_f)}{k^2(L_R - L_{Rf})^3}, \\ C_\theta = \frac{2(L_R L_{Rf}(\theta - \theta_f) - L_{Rf}^2(2\theta + \theta_f) + L_R^2(2\theta_f + \theta) + 3(L_{Rf} + L_R)(h_f - h))}{k^2(L_R - L_{Rf})^3}, \end{cases} \tag{3}$$

where h is the altitude, h_f is the terminal altitude constraint, θ is the velocity inclination constraint, and θ_f is the terminal velocity inclination constraint. Based on the longitudinal and lateral overload commands, the angle of attack α^* and bank angle υ^* can be calculated as follows:

$$\begin{cases} \alpha^* = C_L^{-1}\left(\frac{2mg}{\rho v^2 S_m}, \sqrt{n_y^2 + n_z^2}\right), \\ \upsilon^* = \arctan\left(n_y / n_z\right), \end{cases} \tag{4}$$

where ρ is the atmospheric density, m is the vehicle mass, S_m is the reference area, and C_L^{-1} is the inverse function calculation, specifically for inverse interpolation using the aerodynamic lift coefficient.

3 Maneuvering amplitude of velocity control

3.1 Terminal velocity prediction

Based on the vehicle's current position (λ, ϕ) and terminal position (λ_f, ϕ_f), the remaining flight range can be calculated:

$$L_{\rm Rgo} = R_e \arccos\left(\sin\phi\sin\phi_f + \cos\phi\cos\phi_f\cos\left(\lambda_f - \lambda\right)\right), \tag{5}$$

where R_e is the average radius of Earth. By combining the current flight velocity v and its derivative $\dot{v}$, the remaining time $T_{\rm go}$ is obtained as follows:

$$T_{\rm go} \approx \frac{-v\cos\theta + \sqrt{v^2\cos^2\theta + 2\dot{v}L_{\rm Rgo}}}{\dot{v}}. \tag{6}$$

At relatively low altitudes, the atmospheric density and aerodynamic lift are sufficiently large; thus, the gliding flight condition can be satisfied. Consequently, the longitudinal forces of the vehicle are balanced; that is, the aerodynamic lift acting in the positive direction is equal to the gravity acting in the negative direction.

$$L \approx mg, \tag{7}$$

where L is the aerodynamic lift. For a gliding flight, the lift-to-drag ratio $R_{\rm L/D}$ is large, and the range of change is small. Therefore, $R_{\rm L/D}$ can be set as a constant in a certain guidance cycle, and the aerodynamic drag can be expressed as

$$D = \frac{L}{R_{\rm L/D}} \approx \frac{mg}{R_{\rm L/D}}, \tag{8}$$

where L is the aerodynamic drag. Based on the simplified aerodynamic drag in (8), the velocity differential can be transformed into

$$\dot{v} = -\frac{D}{m} - g\sin\theta = -\frac{g}{R_{\mathrm{L/D}}} - g\sin\theta. \tag{9}$$

When the vehicle satisfies the gliding flight condition, the velocity inclination angle and its derivative are small; therefore, the right side of differential equation (9) can be regarded as a constant. The definite integral in (9) and the predicted terminal velocity can be obtained as follows:

$$\begin{aligned} \int_{t_c}^{t_f} \dot{v}\mathrm{d}t &= \int_{t_c}^{t_c+T_{\mathrm{go}}} \left(-\frac{g}{R_{\mathrm{L/D}}} - g\sin\theta\right)\mathrm{d}t \\ &\Rightarrow v_{fp} = v - \left(\frac{g}{R_{\mathrm{L/D}}} + g\sin\theta\right)T_{\mathrm{go}}, \end{aligned} \tag{10}$$

where t_c is the current flight time.

3.2 Maneuvering amplitude calculation in velocity control

The purpose of velocity control is to make the predicted remaining velocity $\Delta v_p = v - v_{fp}$ equal the required remaining velocity $\Delta v_r = v - v_f$. Let k_v denote the ratio of the remaining velocity:

$$k_v = \frac{\Delta v_r}{\Delta v_p} = \frac{v - v_f}{v - v_{fp}}. \tag{11}$$

The goal of terminal velocity control is

$$\lim_{L_R \to L_{Rf}} k_v = 1. \tag{12}$$

Because the performance index in (1) is the minimum energy consumption, which means that the terminal velocity is the highest. Therefore, terminal velocity control reduces the maximum terminal velocity to a given constraint value by increasing the velocity loss. To achieve velocity control, the coefficient k_v is introduced to correct the current aerodynamic drag acceleration A_{Dc}.

$$A_{Dr} = k_v A_{Dc}. \tag{13}$$

It is known that the aerodynamic drag acceleration is related to the angle of attack, Mach number, atmospheric density, vehicle mass, and reference area, whereas the angle of attack is the most direct way to obtain the drag acceleration A_{Dr}. Therefore, the required angle of attack α_{Dr} for velocity control can be obtained by inverse interpolation using the drag coefficient C_D.

$$\alpha_{Dr} = C_D^{-1}\left(\frac{2m}{\rho v^2 S_m}, A_{Dr}\right). \tag{14}$$

The offline obtained standard aerodynamic coefficient C_D must have deviations; therefore, the angle of attack calculated based on the standard drag coefficient C_D will affect the guidance accuracy. To this end, we estimate the drag coefficient $\hat{C_D}$ using the online measured drag acceleration and recalculate the angle of attack α_{Dr}^* via $\hat{C_D}$ to enhance robustness.

$$\begin{cases} \hat{C}_D = \frac{2mA_{Dc}}{\rho v^2 S_m}, \\ \alpha_{Dr}^* = \hat{C}_D^{-1}\left(\frac{2m}{\rho v^2 S_m}, A_{Dr}\right), \end{cases} \tag{15}$$

where $\hat{C_D}$ is the estimated drag coefficient calculated by online measured aerodynamic drag acceleration A_{Dc}, α_{Dr}^* is the angle of attack required for velocity control obtained by inverse interpolation using drag coefficient and online flight states. α_{Dr}^* can be directly output to the vehicle control system in theory. However, this operation is a unilateral destruction of the original optimal guidance law, which will inevitably affect the gliding ballistic characteristics and guidance accuracy. Therefore, it is better to unify α_{Dr}^* and the overload command in the original optimal guidance. Based on the angle of attack α_{Dr}^*, the required total overload N_{total} including velocity control can be obtained as

$$N_{\mathrm{total}} = \frac{\rho v^2 S_m C_L(\alpha_{Dr}^*, v)}{2mg}. \tag{16}$$

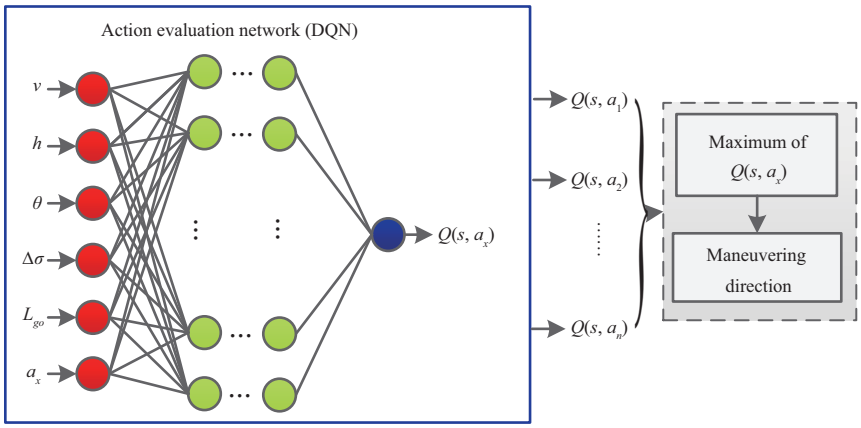

Figure 2 (Color online) Intelligent decision of maneuvering direction via DQN.

According to the velocity control strategy, the longitudinal guidance command is still the optimal overload command, whereas the lateral overload must be adjusted as

$$n_z = \sqrt{N_{\text{total}}^2 - n_y^{*2}}, \tag{17}$$

where n_z is magnitude of lateral overload.

4 DRL for maneuvering direction decision

Eq. (17) gives the magnitude of the lateral overload command used for terminal velocity control. However, another key factor is the sign of the lateral overload; that is, the maneuvering direction needs to be determined.

$$n_z = \sqrt{N_{\text{total}}^2 - n_y^{*2}} \cdot \text{sign}\left(n_z\right), \tag{18}$$

where $\text{sign}(x)$ represents the sign of variable x. The $\text{sign}(x)$ needs to take the actual flight status as an input to satisfy the terminal velocity constraints and eliminate the heading error. This study uses DRL to learn the maneuvering direction decision strategy; therefore, we construct the DRL framework model, including the DQN structure, state and action space, and the reward function. The intelligent decision logic for the maneuvering direction via DQN is shown in Figure 2.

4.1 DRL framework model construction

The basis of DRL is RL and DNN [11]. RL is the process of obtaining a set of expected cumulative benefits from J_π and strategy π, as shown in (19) through the accumulation of a large amount of data and parameter optimization [6].

$$J_\pi = \mathbb{E}_\pi \left[\sum_{t=0}^{n} \gamma_t P(s_t, a_t, s_{t+1}) f_R(s_t, a_t, s_{t+1}) \right], \tag{19}$$

where J_π is the expected cumulative benefits from the initial state s_0 to the terminal state s_f through n steps under the action of the strategy π; $\mathbb{E}_\pi[\cdot]$ is the mathematical expectation calculation; s_t and a_t are the state and action at t steps, and $f_R(s_t, a_t, s_{t+1})$ is the corresponding reward; $P(s_t, a_t, s_{t+1})$ is the state transition probability; $\gamma \in [0, 1]$ is the discount factor. In addition, the decision of the maneuvering direction satisfies the Markov decision process (MDP):

$$\text{MDP} = (\boldsymbol{S}, \boldsymbol{A}, P, f_R, \gamma). \tag{20}$$

Action space $\boldsymbol{A}$ is one of the key parts of RL, in which the action needs to impact the flight state to control the gliding flight by adjusting the action. The DQN method is used to make decisions on the maneuvering direction; therefore, the action space can be designed as a set of maneuvering overload signs.

$$\boldsymbol{A} = \{-1, 1\}. \tag{21}$$

State space $\boldsymbol{S}$ refers to a collection of flight states that can accurately describe gliding flights and guidance tasks. The flight states of three-dimensional gliding flight include velocity, velocity inclination, azimuth angle, longitude, latitude, and altitude. In addition, gliding guidance tasks are characterized by terminal constraints. To reduce input parameters and enhance versatility, we integrate flight status and terminal constraints into relative states and then construct a concise and efficient state space.

$$\boldsymbol{S} = \{v, h, \theta, \Delta\sigma, L_{\mathrm{Rgo}}\}. \tag{22}$$

The remaining range-to-go L_{Rgo} includes the current and target latitude and longitude, and the heading error $\Delta\sigma$ includes the position and azimuth information.

$$\Delta\sigma = \sigma_v - \sigma_{\mathrm{LOS}}. \tag{23}$$

The states in (22) are time-continuous variables that need to be discretized in traditional RL. The more intensive the state discretization, the higher the learning accuracy; however, this will reduce the learning efficiency. For gliding vehicles flying in large airspaces, the range of changes of each state is relatively large; for example, the remaining range L_{Rgo} can reach several thousand kilometers. Therefore, the traditional state-space construction method has a contradiction between high-precision rapid decision-making and a sharp increase in dimensionality. Therefore, the DRL method is used to realize maneuvering direction decision-making.

As shown in Figure 2, the DQN constructed in this study includes an input layer, two middle layers, and an output layer. Each middle layer contains 200 neurons and uses the ReLU "tansig" activation function. The input and output parameters are as follows:

$$\begin{cases} \boldsymbol{D}_{\mathrm{input}} = \{v, h, \theta, \Delta\sigma, L_{\mathrm{go}}, a_x\}, \\ \boldsymbol{D}_{\mathrm{output}} = Q(s, a_x), \end{cases} \tag{24}$$

where a_x is a certain action in action space $\boldsymbol{A}$ and $Q(s, a_x)$ represents the reward value of the input flight states and action. To improve training efficiency, the sample data were normalized during training.

$$\begin{cases} v_{\mathrm{normal}} = \frac{v - v_f}{v_0 - v_f}, \ \theta_{\mathrm{normal}} = \theta \times \frac{180}{\pi}, \\ h_{\mathrm{normal}} = \frac{h - h_f}{h_0 - h_f}, \ \Delta\sigma_{\mathrm{normal}} = \Delta\sigma \times \frac{180}{\pi}, \\ L_{\mathrm{gonormal}} = \frac{L_{\mathrm{go}} - L_{\mathrm{go}f}}{L_{\mathrm{go}_0} - L_{\mathrm{go}f}}, \end{cases} \tag{25}$$

where the subscript "f" represents the terminal constraint, and the subscript "0" indicates the initial glide state.

4.2 Reward function design

The reward function is a key section of maneuvering direction decision making, which is used to calculate the reward value of the vehicle after the execution of the action. The reward value directly determines the scientific nature of the action judgment and DQN training efficiency, especially for long-range gliding guidance. In this study, the purpose of the maneuvering direction decision is to satisfy the terminal position and velocity magnitude, which means that terminal state errors can be used to calculate the return value and evaluate the current action. Consequently, the reward function is designed as follows:

$$f_R = -1000\frac{|v_{fp} - v_f|}{\sqrt{\mu_e / R_e}} - 1000\frac{|P_{\mathrm{error}}|}{R_e}, \tag{26}$$

where v_{fp} is the predicted terminal velocity, P_{error} is the terminal position error, μ_e is the gravitational constant, R_e is the mean radius of Earth. Both v_{fp} and P_{error} were calculated by numerical integration from the current flight states to the terminal conditions. In the prediction process, the guidance commands are generated by longitudinal optimal guidance, whereas the lateral guidance is obtained by terminal velocity prediction and current DQN parameters. The reward value calculation logic is illustrated in Figure 3. In (26), the closer the predicted value is to the terminal constraint, the greater the reward. As shown in Figure 4, the reward value varies linearly with terminal errors, and the maximum reward is 100.

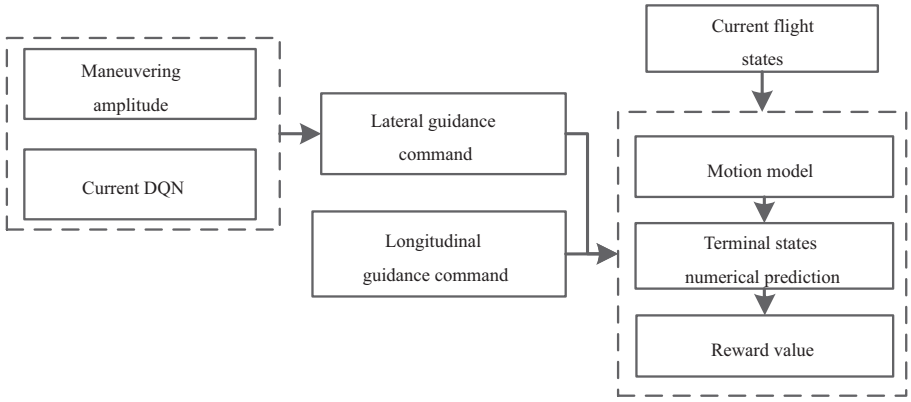

Figure 3 Block diagram of reward value calculation.

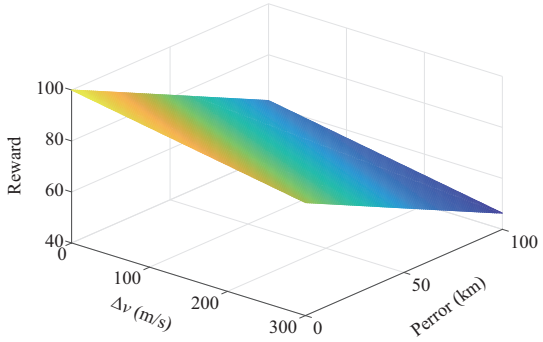

Figure 4 (Color online) Reward values under different velocities and position errors.

5 DQN training for maneuvering direction decision

The DQN training needs to construct data samples, set training goals, and train network parameters. Considering the particularity of long-distance gliding flight, this section uses the heading-error corridor to initialize the network and proposes an experience replay pool management and training strategy suitable for long-distance flight.

5.1 DRL algorithm structure

A DQN is a multilayer DNN that can approximate the action value function. There are two networks with the same structure but different parameters in the DRL: the evaluation network and the target network.

First, we use the same method to initialize the evaluation network and the target network. In each training cycle, take the action evaluation network whose internal parameter is $\boldsymbol{w}$, calculate the estimated value $Q(s,a;\boldsymbol{w})$ of all actions in the current state, and output the action a_t with the maximum value. Secondly, act a_t on the vehicle, calculate the reward value r_t, and get the next state s_{t+1}. The experience tuple $e_t=(s_t,a_t,r_t,s_{t+1})$ is stored in a database $\mathcal{D}=e_1,\ldots,e_N$. $\mathcal{D}=e_1,\ldots,e_N$ contains the experience information of multiple gliding trajectories, such that it is called an experience replay pool. Finally, sample experience data $e\sim\mathcal{D}$ randomly, construct a loss function based on the target Q-value $Q_T(s_{t+1},a_{t+1};\boldsymbol{w}_T)$ and evaluated Q-value $Q(s_t,a_t;\boldsymbol{w})$, and then use gradient descent to update the network parameters. Based on the current states, action and reward value, the target value for training can be calculated based on the target network [11,12].

$$y_t=\begin{cases} r_t,\ \ s_{t+1}=s_f, \\ r_t+\gamma\max_{a_{t+1}}Q_T\left(s_{t+1},a_{t+1};w_T\right),\ \ s_{t+1}\neq s_f. \end{cases} \tag{27}$$

The loss function based on the target value and estimated value is [11]

$$L_t\left(\boldsymbol{w}\right)=\mathbb{E}_\pi\left[\frac{1}{2}\left(y_t-Q\left(s_t,a_t;\boldsymbol{w}\right)\right)^2\right]. \tag{28}$$

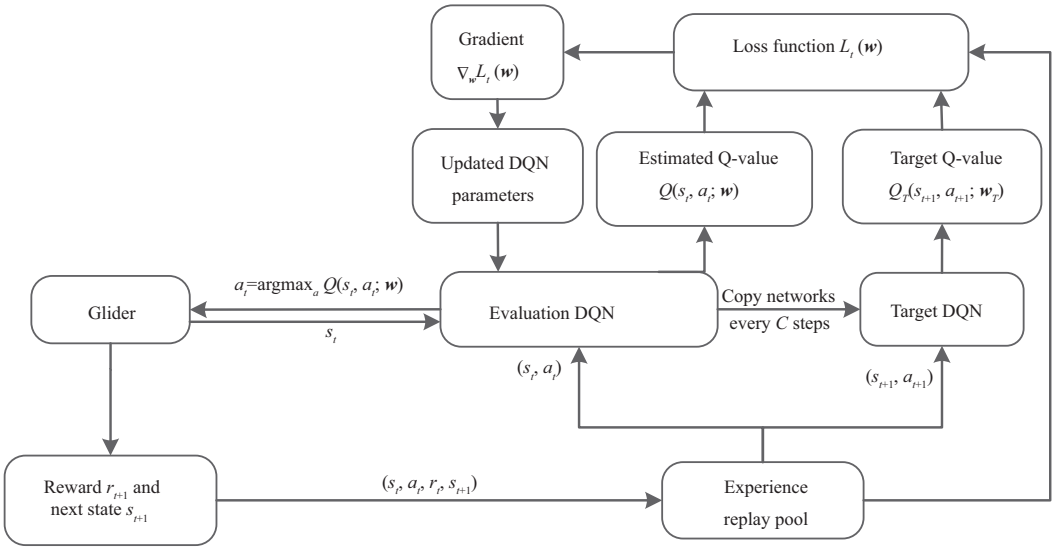

Figure 5 DQN algorithm with two networks.

The gradient of $L_T(\boldsymbol{w})$ relative to each network parameter $\boldsymbol{w}$ is [11]

$$\nabla_w L_t(\boldsymbol{w}) = \mathbb{E}_\pi \left[\left(y_t - Q(s_t, a_t; \boldsymbol{w}) \right) \nabla_w Q(s_t, a_t; \boldsymbol{w}) \right] \tag{29}$$

with above process repeated continuously, and the network parameter is updated using (29). We obtained a set of network parameters that can make maneuvering-direction decisions. When the parameters of the evaluation network were updated C times, the weight of the current evaluation network was copied to the target network. In the next C evaluation network parameter update, the target network is used to generate a target value based on (27), as the learning target for the evaluation network. The DQN algorithm with the two networks is illustrated in Figure 5.

The general method of DRL is given above. For long-distance and long-time gliding flight, we are most concerned with the satisfaction of terminal constraints. Therefore, during gliding flight, some actions with small return value will not affect the success or failure of guidance, but will reduce the learning efficiency and accuracy. To this end, in the RL iterative process, we employ the guidance accuracy after the calculation of the whole trajectory to manage the experience pool data. The improved DQN training algorithm is shown in Algorithm 1. In Algorithm 1, DRL needs to calculate the complete gliding trajectory repeatedly during the iterative training process, and the trajectory data will be stored in the experience pool for DQN parameter training if the terminal accuracy satisfies the requirements, that is $\Delta s_f < \epsilon$. In the improved method, the DQN parameter is trained using the success gliding trajectory data. The terminal error range ϵ can be set larger in the initial training stage to explore more actions, and then reduces continuously to ensure training convergence and guidance accuracy.

5.2 Network parameter initialization via heading error

Initialization is the first step in DQN training. The gliding flight is long, and the gliding guidance needs to judge the maneuvering direction at each time, so that the decision-making space for glide guidance is huge. The traditional random initialization method leads to low learning efficiency and slow convergence rate. Therefore, this study comprehensively uses the heading-error corridor and random methods to generate trajectory data to initialize the network parameters.

It is known that the premise of velocity control is to ensure that the terminal position constraint, that is, to control the heading error within a certain range. Therefore, this study determines the lateral overload sign based on the heading-error corridor to ensure terminal guidance accuracy and reduce the bank angle reversal times. The heading-error corridor is shown in Figure 6.

In Figure 6, L_{R1}, L_{R2}, L_{R3} and L_{Rf} are the range nodes of the corridor and the boundary values, and all of the above parameters are set manually. The corridor boundary value in Figure 6 can be described

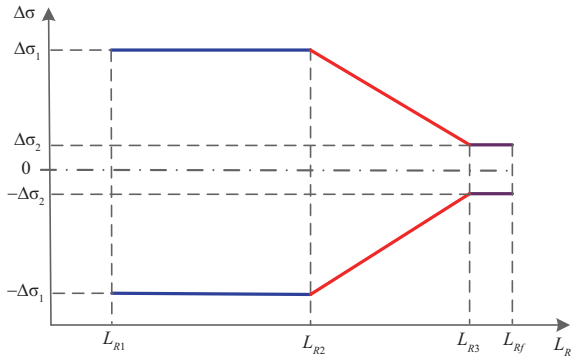

Figure 6 (Color online) Heading-error corridor.

Algorithm 1 Intelligent gliding guidance training process

Initialize experience replay memory $\mathcal{D}$ to capability N;
Initialize evaluation and target network $\boldsymbol{w} = \boldsymbol{w}_T$;
Set training parameters;
for episode = 1 to M **do**
  Initialize flight states $s_0 = \{v_0, h_0, \theta_0, \Delta\sigma_0, L_{go0}\}$;
  while $L_R < L_{Rf}$ **do**
    With ϵ-greedy strategy, select action $a_t = \begin{cases} \text{random}, & \epsilon \text{ probablity}, \\ \max_a Q(s_t, a; \boldsymbol{w}), & 1-\epsilon \text{ probablity}; \end{cases}$
    Execute action a_t, obtain reward r_t and next state s_{t+1};
    Construct array $e_t = \{s_t, a_t, r_t, s_{t+1}\}$;
    Store array e_t in experience pool $\mathcal{D}$ (if $\mathcal{D}$ is full, delete the old array);
  end while
  if $\Delta s_f < \epsilon$ **then**
    Sample $\{e_1, e_2, \ldots, e_{Ne}\}$ from $\mathcal{D}$ randomly;
    For $j = 1, 2, \ldots, N_e$, set $y_i = \begin{cases} r_{j+1}, & s_{j+1} = s_f, \\ r_{j+} + \gamma \max_{a_{j+1}} Q(s_{j+1}, a_{j+1}; \boldsymbol{w}_T), & s_{j+1} \neq s_f; \end{cases}$
    Calculate loss function $L_j(w) = \Sigma_j (y_j - Q(s_j, a_j; w))^2$;
    Train the network parameters $\boldsymbol{w}$ using the gradient descent method;
  end if
  Reset target network parameters $\boldsymbol{w}_T = \boldsymbol{w}$ every C steps.
end for

by the following piecewise linear function:

$$\Delta\sigma_{\max}(L_R) = \begin{cases} \Delta\sigma_1, & L_{R1} < L_R < L_{R2}, \\ \Delta\sigma_1 + \frac{\Delta\sigma_2 - \Delta\sigma_1}{L_{R3} - L_{R2}}(L_R - L_{R2}), & L_{R2} < L_R < L_{R3}, \\ \Delta\sigma_2, & L_R > L_{R3}. \end{cases} \tag{30}$$

For the heading-error corridor, shown in Figure 6 and (30), the error range is large enough to reduce the bank angle reversal times when the vehicle is far from the target. As the vehicle approaches the target, the error range decreases linearly and maintains a small value to ensure position accuracy.

The logic of lateral overload sign determination for terminal velocity control is that the lateral overload sign remains unchanged when $\Delta\sigma$ is within the error corridor; the sign will be negative to reduce the heading error if $\Delta\sigma$ exceeds the upper boundary of the error corridor. In contrast, the lateral overload command is positive if $\Delta\sigma$ exceeds the lower boundary.

$$\text{sign}(n_z)_{\text{corridor}} = \begin{cases} -1, & L_{R1} < L_R < L_{R2}, \\ 1, & L_{R2} < L_R < L_{R3}, \\ \text{sign}(n_{z0}), & L_R > L_{R3}, \end{cases} \tag{31}$$

where n_{z0} represents the lateral overload command at a previous moment. Here, we employ the maneuvering sign in (31) and a random method to determine the maneuvering sign.

$$\text{sign}(n_z) = \begin{cases} \text{random}, & \epsilon \text{ probablity}, \\ \text{sign}(n_z)_{\text{corridor}}, & 1-\epsilon \text{ probablity}, \end{cases} \tag{32}$$

Table 1 Simulation and DQN training conditions

Simulation condition	Value	DQN training parameter	Value
Initial velocity	6500 m/s	Learning episodes	1000
Initial velocity inclination	0°	Guidance period	2 s
Initial position	(0°E, 0°N)	Discount factor	$\gamma = 0.9$
Initial altitude	65 km	Target network update period	$C = 3$
Initial heading error	0°	Data sampling interval	200 km, 20 km (last 5% of total range)
Terminal altitude	30 km	Learning rate	0.005
Terminal velocity inclination	0°	Size of experience pool	104
Target position	(95°E, 10°N)	Sampling size for each training	2000

where $\epsilon_n = 0.5$. By substituting the maneuvering sign determined using (32) into (18) and combining it with longitudinal optimal guidance, the gliding trajectory can be calculated. With the entire glide trajectory, we can store the ballistic data, maneuvering sign, and reward value, and train the DQN parameters.

6 Simulations and analysis of guidance performance

CAV-H was used to verify the guidance performance [20]. The initial conditions, terminal altitude, and DQN training parameters are listed in Table 1.

The control capability constraints were as follows: maximum angle of attack, 20°; maximum rate, 5°/s; maximum bank angle, 70°; and maximum rate, 20°/s. We employ the ϵ-greedy strategy to train the DQN parameters, in which $\epsilon = 0.5$, in the initial 1/3 total range; $\epsilon = 0.3$ in the middle 1/3 total range; and $\epsilon = 0.1$ in the last 1/3 total range. In the construction of the experience pool, the terminal velocity error range Δv_f was gradually reduced from 100 to 20 m/s, and the position error range ΔP_f was reduced linearly from 100 km to 100 m. Because the velocity prediction error decreases as the remaining distance decreases, the velocity control is set in the last 60%–90% of the total range. In the terminal 1% range, only optimal guidance is provided to ensure position and velocity inclination precision.

6.1 Nominal performance test

The terminal parameters were set as follows: longitude, 95°; latitude, 10°; velocity inclination, 0°; and velocity, 2800 m/s. The training results are shown in Figure 7. It can be seen from the training results that the intelligent gliding guidance strategy and DQN training method can achieve perfect convergence effects.

As shown in Figures 7(a) and (b), in the initial training stage, DRL needs to explore more actions, and the neural network converges, so there are large errors in the terminal position, velocity, and altitude, resulting in a small return value. According to the glide guidance strategy, the lateral maneuver for velocity control primarily affects the position error. Therefore, in the process of learning the DQN, the position error is large, convergence rate is slow, and terminal position error is approximately 16 m. In Figure 7(c), because the overload command magnitude is directly calculated from the predicted terminal velocity, the velocity error only exists in the initial five episodes and then converges to zero rapidly. It can be observed in Figure 7(d), that the lateral maneuver has little impact on the altitude. The altitude has a maximum error of 2 km in the initial learning stage and then converges to the constraint value accurately, with an error of approximately 10 m. Figures 7(e) and (f) give the evaluated value and target value of the Q-value, respectively, which are very close, indicating the good convergence effect of the DQN. The offline trained DQN parameters are stored and will be used in online gliding guidance.

In the process of gliding flight, the actual flight status is transferred to the above DQN to obtain the maneuvering direction command for velocity control. Combining the lateral maneuver and longitudinal optimal guidance, the gliding trajectory curve obtained is shown in Figure 8. Because velocity control is realized by lateral maneuvering, there is a large amplitude modification in the bank angle during the velocity control phase, while the angle of attack has a relatively small variation range, as shown in Figure 8(a). Owing to the above control variables, the heading error and ground trajectory have corresponding variations. In Figure 8(b), the heading error is within 15°, and the reduction in the relative distance causes a mutation in the LOS angle; thus, the heading error varies from 0° to 1.942° rapidly at the end of flight. As shown in Figures 8(c)–(f), the terminal position error is 36.848 m, the

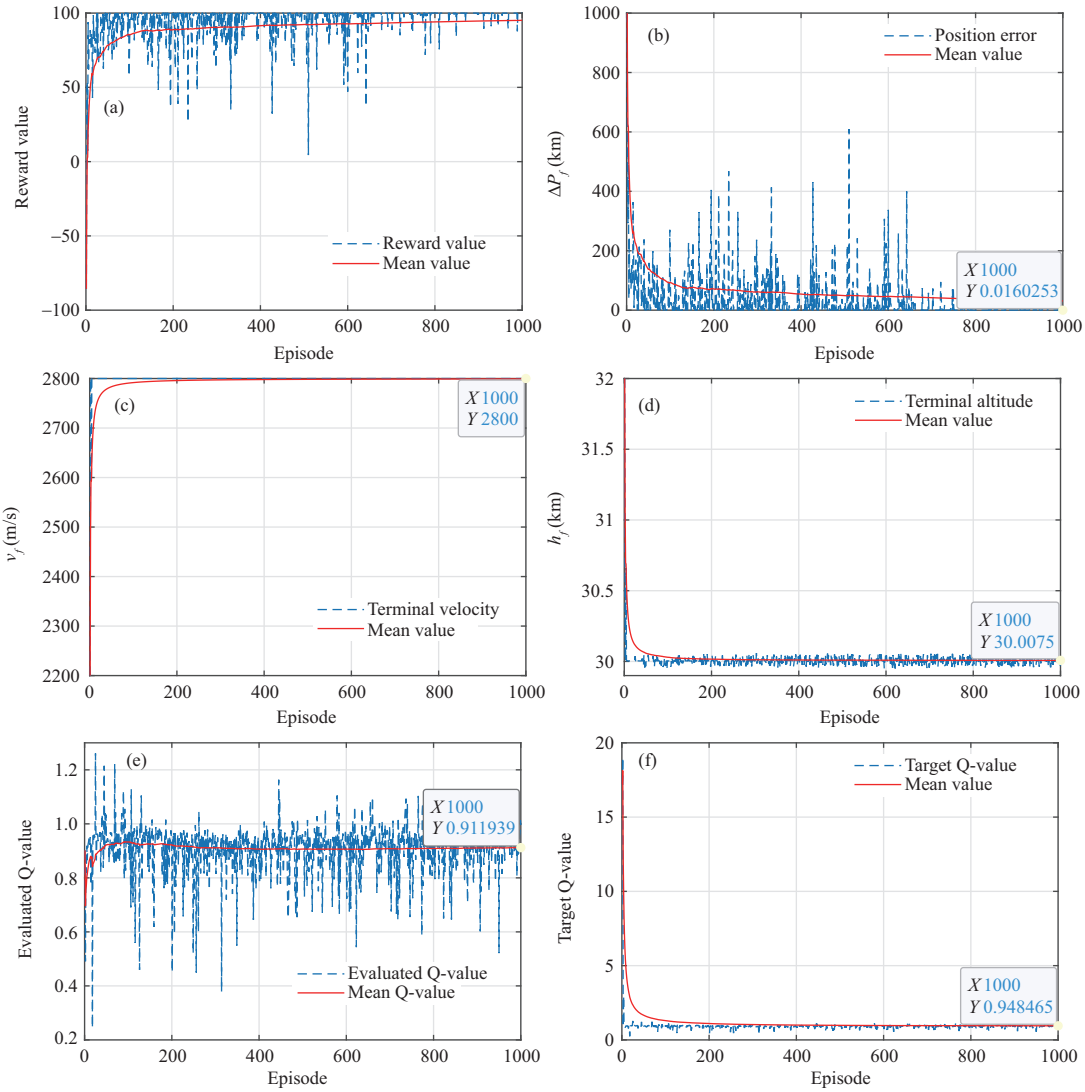

Figure 7 (Color online) Key flight states in DQN training process. (a) Reward value; (b) terminal position error; (c) terminal position error; (d) terminal altitude; (e) evaluated Q-value; (f) target Q-value.

velocity magnitude error is almost zero, the altitude error is 5 m, and the velocity slope angle error is 0.001°. It can be seen from the simulation results that intelligent gliding guidance can control the vehicle to satisfy the terminal constraints with high accuracy under the premise of strong process constraints.

6.2 Adaptability simulation verification

Adaptability to diversified missions is an important aspect of intelligent gliding guidance. To this end, we set different initial point positions and terminal velocity constraints, and employed the DQN parameters obtained in Subsection 6.1, to make online decisions on maneuvering direction under different guidance tasks. The initial position, terminal velocity constraints, and terminal guidance results are listed in Table 2 and Figure 9. It can be seen from the simulation results that the intelligent gliding guidance strategy can satisfy various terminal constraints with high accuracy under different initial positions and terminal velocity constraints without any adjustment of the DQN model and parameters. All terminal position errors were within 70 m, and the velocity errors were almost zero.

6.3 Robustness simulation verification

There are various deviations in the flight environment and vehicle body model. Therefore, to verify the robustness, we added constant deviations of the atmospheric density and aerodynamic coefficients during

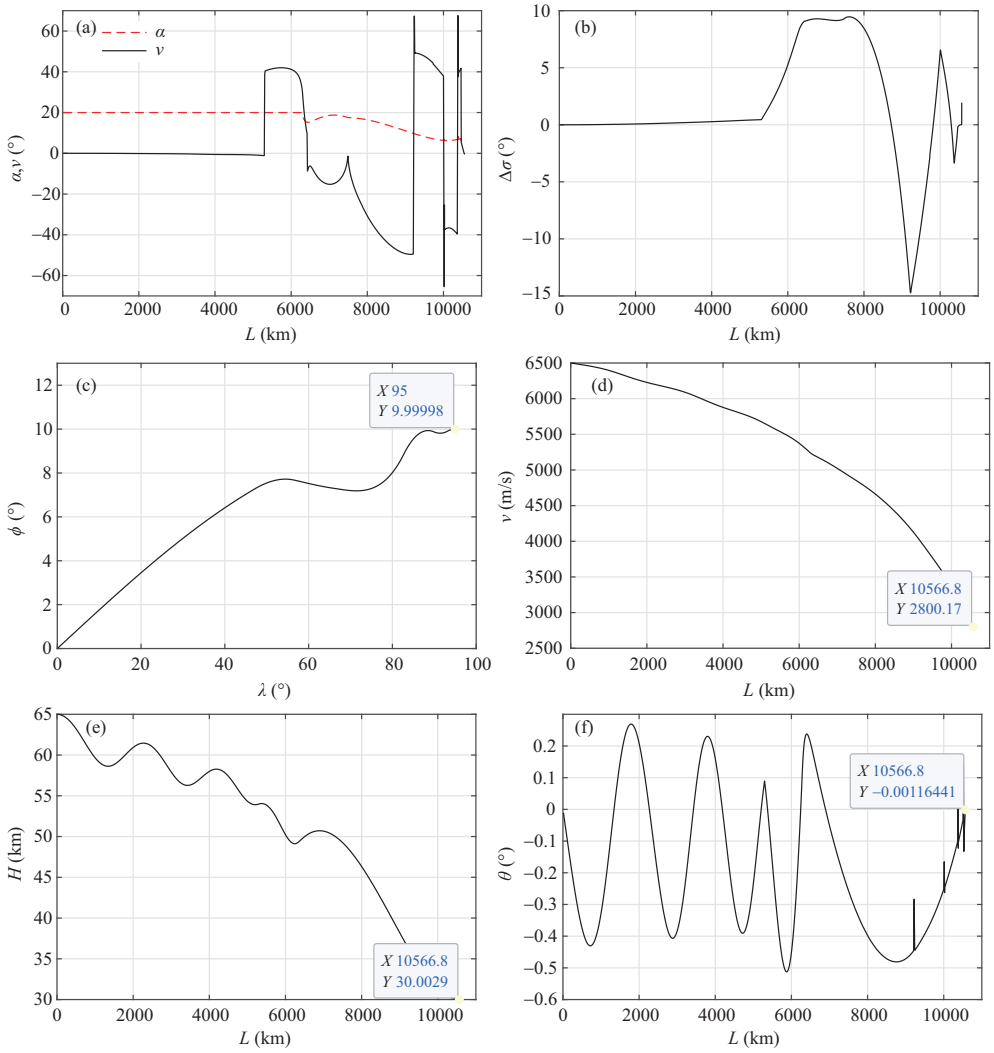

Figure 8 (Color online) Main gliding trajectories of intelligent guidance. (a) Control variables; (b) heading error; (c) ground trajectory; (d) velocity; (e) altitude; (f) velocity inclination.

Table 2 Terminal status in different initial positions

Initial position	Required velocity (m/s)	Position error (m)	Terminal velocity (m/s)	Terminal altitude (km)
(0°E, 0°N)	2800	36.848	2800	29.995
(5°E, 0°N)	2800	65.524	2800	30.003
(5°W, 0°N)	2800	24.803	2800	30.007
(0°E, 3°N)	2800	17.608	2800	29.982
(0°E, 5°N)	2800	26.006	2800	30.05
(0°E, 3°S)	2800	17.051	2800	29.997
(0°E, 5°S)	2800	35.826	2800	30.018

the entire flight course in the ballistic calculation; the deviations are unknown to the guidance system. Taking the initial position as (0°E, 0°N) and the terminal position as (90°E, 10°N) as an example for simulation verification, the terminal guidance results are presented in Table 3. It can be seen from the calculation results that the constant deviation inevitably affects the guidance accuracy, especially for large deviations.

For the atmospheric density deviation, the available overload and control ability of the vehicle increase as the atmospheric density increases. Therefore, when the atmospheric density increases by 20%, the terminal constraints can still be satisfied with high accuracy. Conversely, the control ability decreases

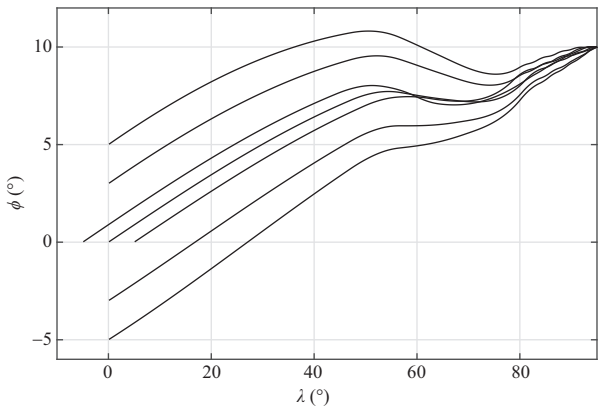

Figure 9 Ground trajectories under different initial positions.

Table 3 Terminal results under constant deviation in the whole course

Deviation item	Deviation value (%)	Required velocity (m/s)	Actual velocity (m/s)	Position error (m)	Terminal altitude (km)
Without deviation	0	2800	2800	27.325	30.016
Atmospheric density	20	2800	2800	37.596	29.97
	10	2800	2800	43.684	30.02
	5	2800	2800	31.814	29.987
	−5	2800	2800	11.354	30
	−10	2800	2800	89.143	30.308
	−20	2800	2800	97.924	30.956
Lift coefficients	20	2800	2800	1263.127	30.002
	10	2800	2800	34.927	29.974
	5	2800	2800	15.493	29.992
	−5	2800	2800.1	11.889	29.976
	−10	2800	2800	132.334	30.001
	−20	2800	2031.6	12.808	29.995
Drag coefficients	20	2000	2000	37.803	29.995
	10	2800	2800	73.592	30.003
	5	2800	2800	43.449	30.002
	−5	2800	2800	48.922	30.041
	−10	2800	2800	19.59	30.013
	−15	2800	2800	144.27	30.61
	−20	2800	2800	1023.1	31.453

as the atmospheric density decreases, and the reduction in aerodynamic drag also increases the terminal velocity when the velocity is uncontrolled. Therefore, the residual velocity to be consumed increases with a decrease in atmospheric density; therefore, the vehicle needs to adopt more serious maneuvering flights to satisfy the terminal velocity of 2800 m/s, which produces greater terminal position and altitude errors.

Increasing the lift coefficient can increase the lift-to-drag ratio and terminal velocity when the velocity is not controlled. Consequently, it can be observed from Table 3 that the vehicle has a sufficient capacity to consume the remaining velocity when the lift coefficient increases. When the lift coefficient increases and the required velocity remains at 2800 m/s, the greater remaining velocity causes more serious maneuvering, resulting in a larger terminal position error. In contrast, the control capability decreases as the lift coefficient decreases, and the uncontrolled terminal velocity is also reduced. When the lift coefficient is reduced by 10%, the terminal position error reaches 132 m. When the deviation further increases to −20%, the vehicle can only fly with optimal guidance without velocity control, and the maximum terminal velocity can only reach 2031.6 m/s. In contrast to the lift coefficient, the control ability decreases as the drag coefficient increases. The guidance accuracy after the drag coefficient deviation was affected by the control ability and residual velocity. Excessive residual velocity leads to a serious maneuvering flight and increases the guidance error.

From the calculation results presented in Table 3 and the above analysis, it can be observed that the

Table 4 Statistical characteristics of terminal parameters under random deviation

	G1 (Intelligent guidance law)		G2 (Guidance law in [10])	
	Mean value	Mean square error	Mean value	Mean square error
Position error (m)	1.331	4.212	12.465	5.113
Velocity (m/s)	2801.181	2.7042	2806.421	1.667
Altitude (km)	30.001	0.003	30.001	0.003
Velocity inclination (°)	−0.0018	0.0186	0.0021	0.0187

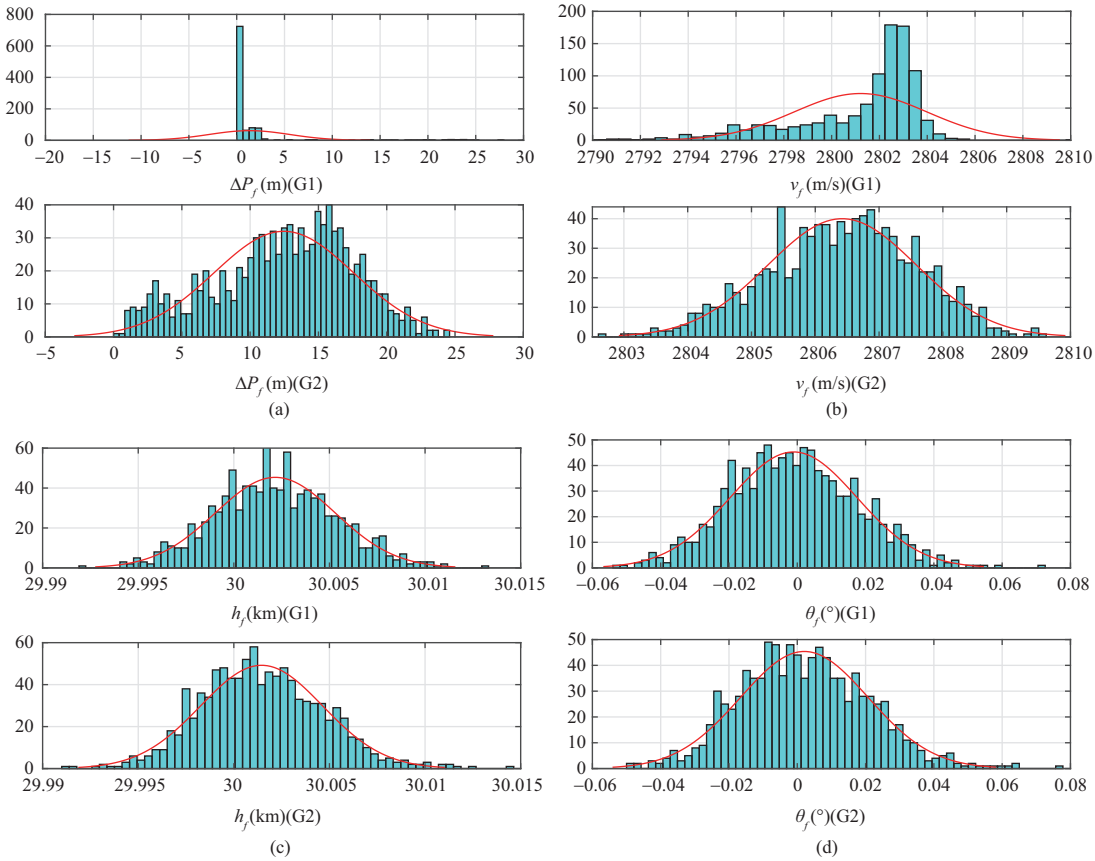

Figure 10 (Color online) Distribution of terminal parameters under random deviation. (a) Position error; (b) velocity; (c) altitude; (d) velocity inclination.

constant deviation can directly affect the residual velocity, actual control ability, and guidance precision. With sufficient control ability, the intelligent gliding guidance strategy can use the same set of DQN parameters to satisfy multiple terminal constraints without any adjustment of the guidance models and methods.

In the actual flight process, in addition to the full-range constant deviation under the above extreme conditions, random deviations also occur. For this reason, the random deviation of the standard normal distribution was added to the atmospheric density and aerodynamic coefficients to verify robustness. The mean square error is $3\sigma = 30\%$ when the altitude is greater than 40 km and reduces to $3\sigma = 15\%$ at low altitudes. The statistical results and distribution of terminal parameters of 1000 simulated shootings for intelligent guidance (G1) and the guidance law in [10] (G2), are listed in Table 4 and Figure 10. The calculation results show that the intelligent gliding guidance method can effectively deal with the influence of external deviations. The overload command of the intelligent gliding guidance is directly calculated according to the terminal velocity, and the direction is determined by the DQN. The 3σ statistical results show that the terminal position error is within 5 m, which is better than the result of G2. In addition, the velocity error of G1 was within 4 m/s, whereas the error of G2 was approximately 8 m/s. Because velocity control is mainly embodied in lateral guidance, the terminal altitude and velocity inclination error of G1 and G2 are basically the same. The altitude deviation was less than 5 m, the velocity inclination

deviation was less than 0.05°.

7 Conclusion

This study proposes a multi-constrained intelligent gliding guidance strategy based on the optimal guidance, predictor-corrector technique, and DRL. Analytical optimal guidance was used to satisfy the constraints of terminal latitude, longitude, altitude, and velocity inclination. Lateral maneuvering is used to control the terminal velocity, in which the amplitude is calculated by analytical prediction and the direction is determined by the DQN. The key points of this study are as follows. First, we proposed an intelligent decision-making model of maneuvering direction based on DQN, which mainly includes the design of time continuous state space and finite-dimensional action space, and the construction of a reward function based on terminal state prediction. The second is the efficient training of DQN parameters, which includes an initial sample design based on the heading-error corridor and the experience pool management for long-period gliding decision-making. The following conclusion was drawn through theoretical derivation and simulation verification.

(1) Comprehensive utilization of optimal guidance, predictor-corrector technique, and DRL methods can satisfy terminal constraints with high precision and reduce the complexity and dimension of DQN models.

(2) Maneuvering direction selection is an intelligent decision-making problem that takes continuous high-dimensional flight status as input and finite-dimensional maneuvering sign as output, which can be effectively solved by a DQN.

(3) The training efficiency of the DQN can be improved effectively with the help of the construction of the initial sample using the heading-error corridor and the experience replay pool management according to the terminal status during training.

(4) Owing to the strong generalization ability of the DQN, intelligent guidance can still achieve different tasks and resist different forms of process interference without model and parameter modification.

Robustness to uncertainties is key to gliding guidance. Increasing the uncertainty of the environment and vehicle body in DQN training is an effective way to enhance robustness, and it is one of the primary objectives of future research.

References

1 Yu J L, Dong X W, Li Q D, et al. Cooperative guidance strategy for multiple hypersonic gliding vehicles system. Chin J Aeronaut, 2020, 33: 990–1005
2 Lahanier H, Serre L. Trajectory and guidance scheme design for free flight test of hypersonic vehicle. In: Proceedings of the AIAA Guidance, Navigation, and Control Conference, 2017
3 Joshi A, Sivan K, Amma S S. Predictor-corrector reentry guidance algorithm with path constraints for atmospheric entry vehicles. J Guid Control Dyn, 2007, 30: 1307–1318
4 Zhu J, Zhang S. Adaptive optimal gliding guidance independent of QEGC. Aerospace Sci Tech, 2017, 71: 373–381
5 Mnih V, Kavukcuoglu K, Silver D, et al. Human-level control through deep reinforcement learning. Nature, 2015, 518: 529–533
6 Kober J, Bagnell J A, Peters J. Reinforcement learning in robotics: a survey. Int J Robot Res, 2013, 32: 1238–1274
7 Bhopale P, Kazi F, Singh N. Reinforcement learning based obstacle avoidance for autonomous underwater vehicle. J Mar Sci Appl, 2019, 18: 228–238
8 Junell J L, van Kampen E J, de Visser C C, et al. Reinforcement learning applied to a quadrotor guidance law in autonomous flight. In: Proceedings of AIAA Guidance, Navigation, and Control Conference, 2015
9 Gaudet B, Furfaro R. Missile homing-phase guidance law design using reinforcement learning. In: Proceedings of AIAA Guidance, Navigation, and Control Conference, 2012
10 Luo Z, Li X S, Wang L X, et al. Multiconstrained gliding guidance based on optimal and reinforcement learning method. Math Problems Eng, 2021, 2021: 1–12
11 Yang J, You X, Wu G, et al. Application of reinforcement learning in UAV cluster task scheduling. Future Generation Comput Syst, 2019, 95: 140–148
12 Chai R, Tsourdos A, Savvaris A, et al. Six-DOF spacecraft optimal trajectory planning and real-time attitude control: a deep neural network-based approach. IEEE Trans Neural Netw Learn Syst, 2019, 31: 5005–5013
13 Hovell K, Ulrich S. On deep reinforcement learning for spacecraft guidance. In: Proceedings of AIAA Scitech 2020 Forum, 2020
14 Hovell K, Ulrich S. Deep reinforcement learning for spacecraft proximity operations guidance. J Spacecraft Rockets, 2021, 58: 254–264
15 Woodbury T D, Dunn C, Valasek J. Autonomous soaring using reinforcement learning for trajectory generation. In: Proceedings of the 52nd Aerospace Sciences Meeting, 2014
16 Julian K D, Kochenderfer M J. Distributed wildfire surveillance with autonomous aircraft using deep reinforcement learning. J Guid Control Dyn, 2019, 42: 1768–1778
17 Gaudeta B, Furfaroa R, Linares R. Reinforcement meta-learning for angle-only intercept guidance of maneuvering targets. In: Proceedings of the AIAA SciTech 2020 Forum, 2020
18 Gao W, Zhou X, Pan M, et al. Acceleration control strategy for aero-engines based on model-free deep reinforcement learning method. Aerospace Sci Tech, 2022, 120: 107248
19 Zhu J, Liu L, Tang G, et al. Highly constrained optimal gliding guidance. Proc Inst Mech Eng Part G-J Aerospace Eng, 2015, 229: 2321–2335
20 Phillips T H. A common aero vehicle (CAV) model, description, and employment guide. Schafer Corp AFRL and AFSPC, 2003, 27: 1–12