IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received May 6, 2021, accepted June 11, 2021, date of publication June 22, 2021, date of current version June 30, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3091605

Realizing Midcourse Penetration With Deep Reinforcement Learning

LIANG JIANG, YING NAN, AND ZHI-HAN LI
Academy of Astronautics, Nanjing University of Aeronautics and Astronautics, Nanjing 211106, China
Corresponding author: Liang Jiang (nuaajl@nuaa.edu.cn)

This work was supported in part by the Aviation Science Foundation of China under Grant 201929052002.

ABSTRACT A midcourse maneuver controller is obtained using deep reinforcement learning to maintain the survivability of a ballistic missile. First, the midcourse is abstracted as a Markov decision process (MDP) with an unknown system state equation. Then, a controller formed by the Dueling Double Deep Q (D3Q) neural network is used to approximate the state-action value function of the MDP. In order to make the controller's intelligence improved by deep reinforcement learning, the state space, action space, and instant reward function of the MDP are customized. The controller uses a real-time situation as input and outputs the ignition states of pulse motors. Offline training shows that deep reinforcement learning can achieve the optimal strategy's convergence after approximately 65 hours. Online tests demonstrate the controller's ability to avoid an interceptor intelligently and to account for an entry error. In scenarios with multiple random factors, the controller achieved a penetration probability of 100% and a mean re-entry error of less than 5000 m.

INDEX TERMS Deep reinforcement learning, ballistic missile, midcourse penetration, neural network.

I. INTRODUCTION

Ballistic missiles have a long flight time in midcourse and a fixed trajectory. Therefore, various countries regard midcourse interception as the core strategy for missile defense systems [1]–[3]. Improving the midcourse penetration capability of a ballistic missile is meaningful for maintaining its strategic deterrence. Under the premise of ensuring strike accuracy, using multi-pulse maneuvers to maximize penetration capability has become a research topic of interest [4].

Early research on midcourse penetration strategies involved procedural maneuvers. In [5] and [6], the lateral procedural maneuver in a horizontal plane was studied. The ballistic missile starts the ignition program of the lateral pulse motors at the expected time to realize the orbital maneuver. There are many options for this procedural maneuver, such as a sinusoidal maneuver, square wave maneuver, and snake maneuver. A midcourse penetration strategy was proposed in [7] using an axial impulse maneuver and provided a precise trajectory design method. This penetration strategy does not require lateral pulse motors. In order to eliminate re-entry errors caused by the midcourse maneuver, remaining pulse motors were used to regress a preset ballistic[8]. Procedural maneuver penetration only needs to inject a pulse ignition program into the onboard computer before launch, which does not occupy computing resources and is easy to implement. A penetration strategy around the detection zone has emerged with the advancement of planning technology and information fusion technology. The concept is to design a trajectory before launch that can evade the enemy's detection zone. Developing a solution to this ballistic problem is a complex nonlinear programming problem involving multiple constraints and multiple stages. Based on the assumption that the earth is flattening, infinitely high cylindrical and semi ellipsoidal detection zones were established in [9]. Then, the trajectory was optimized based on the constraints of waypoint and detection zone. Based on the multi-interval pseudospectrum method, a method is proposed in [10] for adaptively adjusting the interval density with curvature and error criteria. The adaptive pseudo-spectrum method was developed for the multi-interval pseudo-spectrum. A rapid trajectory optimization algorithm was also proposed for the whole course under the condition of multiple constraints and multiple detection zones. Many studies were carried out based on differential game theory to determine how active evasion interceptors can penetrate key areas of enemy air

The associate editor coordinating the review of this manuscript and approving it for publication was Emre Koyuncu.

defenses. Research on evasion maneuvers for a warhead with continuous orbital maneuver capability was conducted by [11] using differential games. The guidance law in [12] was constructed using the state-dependent Riccati equation (SDRE) based on the accurate model of a penetration spacecraft and an interceptor. This approach obtained superior combat effectiveness when compared with classic differential game theory.

By investigating the above literature, we can summarize some problems restricting the implementation of midcourse penetration: (1) It's easy to oppose a semantic strategy defined artificially. With the advancement of interception guidance [13], [14], the procedural maneuvers studied in [5] and [6] may be recognized by interceptors, and it cannot guarantee effectiveness when fighting against advanced interceptors. (2) Insufficient adaptability in the battlefield. In light of the wide distribution of potential re-entry points, the circumvention methods studied in [9]and [10] cannot ensure the planned circumvention trajectory will meet energy constraints. In addition, there may not be a trajectory that can evade all detection zones in the enemy's key air defense area. Moreover, it may be impossible to know enemy detection zones clearly before launch because of limited intelligence capability. (3) The calculation is complex and challenging to apply in engineering. Most active evasion methods come from game theory [11], [12]. However, the differential game theory is time-consuming and performs poorly in real-time. Moreover, the model error can be introduced during linearization [15], and onboard computers cannot achieve high-frequency corrections. From the above presentation, the existing midcourse penetration methods derived from traditional methods, such as multiconstraint programming algorithms or differential game theory, present significant obstacles in engineering applications. Recent new technology offers a solution to these limitations.

Since it is a model-free algorithm, reinforcement learning (RL) is effective for solving decision-making problems. It has gained a lot of traction in the control field since it is entirely based on data and does not require any model knowledge. Due to the limitations of traditional reinforcement learning, early research cannot handle high-dimensional and continuous battlefield state information [17]. In recent years, deep neural networks (DNNs) have demonstrated the ability to approximate an arbitrary function [18] and have unparalleled advantages in the feature extraction of high-dimensional data. Deep reinforcement learning (DRL), which combines the advantages of DNNs and RL, is a method close to general artificial intelligence, and its strategies can exceed human empirical cognition [19], [20]. After training, a DNN can quickly output control commands in milliseconds [21] and has good generalization ability to unknown environments. Therefore, DRL has a promising application in ballistic missile penetration.

The main contributions of this paper are as follows:

(1) To the best of our knowledge, this is the first study to explore the application of DRL in the field of midcourse maneuver control, and the effectiveness of this technology is proven by simulation, which may inspire the research community.
(2) An RL implementation scheme is designed to fit the nature of midcourse penetration.

In the second section, the Markov decision process (MDP) with an unknown system model is used to describe the midcourse penetration. In the third section, a particular controller named the Dueling Double Deep Q (D3Q) neural network and the implementation of RL to train the controller are introduced. A thorough analysis of the D3Q training process is presented in the fourth section, along with many online tests. Finally, the conclusion is presented in Section 5.

II. PROBLEM FORMULATION

This paper aims to seek a controller that can actively evade an interceptor. Neither the ballistic missile nor the interceptor have a fixed initial flight state. Under the restriction of the ignition number, pulse motors are used to evade the interceptor as much as possible. During evasion, ignitions should be conserved in order to minimize the re-entry errors at the next stage.

This paper makes the following assumptions:

(1) Only one interceptor is used against a ballistic missile in the vertical plane. This situation has not been solved well in the existing literature. We hope to be fair to both sides in terms of quantity. Otherwise, either side can win by quantity advantage, which will lead to meaningless research on penetration. Therefore, as a common practice, this paper only focuses on this situation.
(2) The interceptor always performs head-on interception. Head-on interception is the most common interception approach, which means that the velocity component of the interceptor and the target have opposite directions in line of sight during the rendezvous. Benefiting from this, the interceptor can intercept a high-speed target at low speed [22], [23].
(3) On the missile, pulse motors are attached to each side, and the reaction force always passes through the mass center. This installation scheme is utilized on the latest ballistic missiles serviced in the army [24].
(4) The missile's body axis and its velocity are coincident. Since this paper is studying penetration through orbital maneuvers, we ignore the attitude to simplify the model and require that the reaction force be perpendicular to the velocity direction vector.

A. ANALYSIS OF THE MIDCOURSE PENETRATION

Fig. (1) shows the movement of a ballistic missile M and an interceptor I in Earth-centered Earth-fixed (ECEF) coordinates. The coordinates and velocity of the spacecraft are known, as well as the position of the expected re-entry point T. Both M and I use pulse motors to perform orbital maneuvers. The reaction forces with amplitude fixed as $\|\boldsymbol{F}\|$ perpendicular to the lower and upper sides of the body are defined as F_{d} and F_{u}, respectively. λ_{S} is the angle between

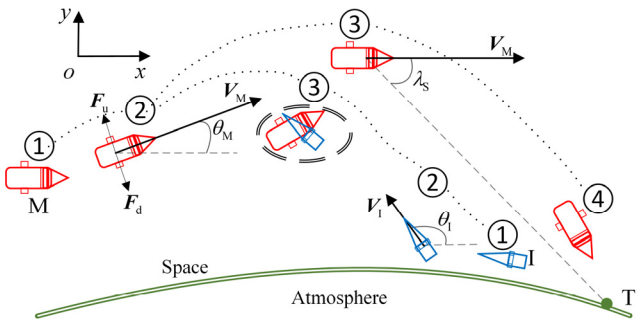

FIGURE 1. Penetration in ECEF coordinates.

the velocity direction of M and the sightline of M and T. The ballistic inclinations of M and I are denoted by θ_{M} and θ_{I}, respectively. The flight velocities are denoted by V_{M} and V_{I}, respectively.

The midcourse penetration can be divided into four stages: Stage ① : A ballistic missile changes orbit using lateral [4] or axial [7] pulse motors to evade the interceptor. Due to the limitation of the detection range, the interceptor glides towards the preset interception point without any control during this stage. Stage ②: The ballistic missile enters the interceptor's lock range. The interceptor began to correct its orbit based on the guidance information [26]. There are two situations in Stage ③: The interceptor destroys the ballistic missile or is evaded. Stage ④: The ballistic missile uses the remaining pulse motors to optimize the re-entry error. The complete process of midcourse penetration is shown in Fig. (1).

From the above analysis, it can be seen that whether the ballistic missile can evade the interceptor in stage ③ depends on the orbital maneuvers adopted in stages ① and ②. Stages ① and ② also determine the difficulty of orbit correction in stage ④. Therefore, the maneuver control of the ballistic missile in penetration belongs to a typical sequential decision problem [27], and the MDP can model this kind of problem [28].

Most of the time, ballistic missiles and interceptors are in an inertial sliding motion after entering space. Under the control of multiple pulses, they follow the same vertical plane motion [25]:

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{v}_x \\ \dot{v}_y \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} v_x \\ v_y \\ -\dfrac{\mu x}{(x^2+y^2)^{\frac{3}{2}}} \\ -\dfrac{\mu y}{(x^2+y^2)^{\frac{3}{2}}} \\ \dfrac{\dot{v}_y\dot{x} - \dot{y}\dot{v}_x}{\dot{x}^2+\dot{y}^2} \end{bmatrix} + \|\boldsymbol{F}\| \begin{bmatrix} 0 \\ 0 \\ -\sin\theta \\ \cos\theta \\ 0 \end{bmatrix} u \qquad (1)$$

where (x, y) is the coordinate vector of an spacecraft in ECEF coordinates. (V_x, V_y) is the velocity vector of the spacecraft; μ is the Earth's gravitational constant. The control input $u \in \{-1, 0, 1\}$ represents the ignition control of the orbital pulse motor. When $u = 1$, the orbital pulse engine mounted on the upper side is ignited and generates force $\boldsymbol{F}_{\mathrm{u}}$, which decreases V_x and increases V_y. When $u = -1$, V_x is increased and V_y is decreased by $\boldsymbol{F}_{\mathrm{d}}$. Obviously, $\boldsymbol{F}_{\mathrm{u}} = -\boldsymbol{F}_{\mathrm{d}}$ and $\|\boldsymbol{F}_{\mathrm{u}}\| = \|\boldsymbol{F}_{\mathrm{d}}\| = \|\boldsymbol{F}\|$. When $u = 0$, neither side's motors fire.

B. MARKOV DECISION PROCESS WITH SYSTEM EQUATION UNKNOWN

The ballistic missile determines when and how to perform maneuvers based on multidimensional information. The deterministic MDP with continuous state and discrete action can be defined by a five-tuple $\langle \mathcal{S}, \mathcal{A}, T, R, \gamma \rangle$ [29]. Among them, $\mathcal{S}$ is a multidimensional continuous state space. $\mathcal{A}$ is a set of available actions. T is a state transition function: $\mathcal{S} \times \mathcal{A} \to \mathcal{S}$, which describes the deterministic state transition. That is, after an action $a \in \mathcal{A}$ is taken in the state $\boldsymbol{s} \in \mathcal{S}$, the state changes from $\boldsymbol{s} \in \mathcal{S}$ to $\boldsymbol{s}' \in \mathcal{S}$. R is an instant reward function: $\mathcal{S} \times \mathcal{A} \times \mathcal{S} \to \mathbb{R}$ represents an instant reward obtained from the state transition. $\gamma \in [0, 1]$ is a constant discount factor used to balance the importance of instant reward and forward cumulative reward.

For the time-discrete MDP, the controller forms a command sequence $\tau = \{a_0, a_1, \cdots, a_n\}$ based on the current state $\boldsymbol{s}_t$ at each time point. The controller starts from state $\boldsymbol{s}_0$. After a command a_t is issued, the system obtains a new state according to T and simultaneously obtains an instant reward r_t by R. The cumulative reward obtained by the controller under the command sequence τ is:

$$G(\boldsymbol{s}_0, \tau) = \sum_{t=0}^{\infty} \gamma^t r_t \qquad (2)$$

The optimal controller outputs the optimal command sequence $\tau^* = \arg\max_{\tau}\{G(\boldsymbol{s}_0, \tau)\}$ according to the initial state $\boldsymbol{s}_0$, and this sequence is the optimal strategy.

Assuming that the time-discrete MDP starts from $\boldsymbol{s}_t$, its cumulative reward is as shown in Eq. (2). Then, its maximum cumulative reward is [30], [31]:

$$G^*(\boldsymbol{s}_t) = \max_{a_t} \left\{ R(\boldsymbol{s}_t, a_t, \boldsymbol{s}_{t+1}) + \gamma G^*(\boldsymbol{s}_{t+1}) \right\} \qquad (3)$$

However, there is a confrontation between the ballistic missile and the interceptor, which forms a system with an unknown state equation. An analytical method based on Eq. (3) cannot be used to calculate the optimal control because the movement of the interceptor is controlled by its guidance law which is unknown for the ballistic missile.

The expected value function of the cumulative reward based on state $\boldsymbol{s}_i$ and the expected value function of $(\boldsymbol{s}_i, a_i)$ are introduced as shown below:

$$V^{\pi}(\boldsymbol{s}_t) = E\left[\sum_{k=0}^{\infty} \gamma^k r_{t+k} | \boldsymbol{s}_t, \pi \right] \qquad (4)$$

$$Q^{\pi}(\boldsymbol{s}_t, a_t) = E\left[\sum_{k=0}^{\infty} \gamma^k r_{t+k} | \boldsymbol{s}_t, a_t, \pi \right] \qquad (5)$$

$V^{\pi}(\boldsymbol{s}_t)$ indicates the expected cumulative reward that controller π can obtain in the current state $\boldsymbol{s}_t$. $Q^{\pi}(\boldsymbol{s}_t, a_t)$ indicates the expected cumulative reward under controller π after executing a_t in state $\boldsymbol{s}_t$.

Theorem 1: Optimal Control of MDP based on expected value function [19]

The cumulative reward of an MDP is defined as Eq. (2). The expected value function of the cumulative reward based on the state-action pair $Q^{\pi}(\boldsymbol{s}_t, a_t)$ of a controller $\pi(\boldsymbol{s}_t)$ is known. Updating $\pi(\boldsymbol{s}_t)$ through the iteration rule shown in the following equation can approximate the maximum expected value of the cumulative reward:

$$\pi'(\boldsymbol{s}_t) = \arg\max_{a_t} Q^{\pi}(\boldsymbol{s}_t, a_t) \tag{6}$$

III. SOLUTION OF THE CONTROLLER

Section 2 converts the midcourse maneuver control of a ballistic missile to an MDP with an unknown state equation. It is crucial to estimate $Q^{\pi}(\boldsymbol{s}_t, a_t)$ accurately in order to obtain an optimal MDP controller. The controller needs to process as much data as possible and then choose the next action. The input space of the controller has high dimensionality and continuous characteristics, and its action space has discrete characteristics. In recent years, deep Q networks (DQNs) have been widely used to approximate $Q^{\pi}(\boldsymbol{s}_t, a_t)$ of high-dimensional state systems [32]–[34]. To achieve better strategy stability and faster learning speed, this section combines duel and double network concepts to improve the structure of a traditional DQN. Finally, D3Q and its reinforcement learning algorithm are obtained.

A. REINFORCEMENT LEARNING DESIGN

To approximate the value function $Q^{\pi}(\boldsymbol{s}_t, a_t)$ through reinforcement learning, the state space, action space and instant reward function of reinforcement learning are designed as follows.

As mentioned in Section 2.a, the mainstream orbital maneuver for ballistic missiles and interceptors is intermittent ignition using pulse motors, which most of the time during the flight satisfies Eq.(1). Therefore, the controller can predict the future flight state (coordinate, velocity) based on the current coordinate and velocity. The state space of the controller is $\mathcal{S} = \mathcal{S}_{\mathrm{M}} \times \mathcal{S}_{\mathrm{I}}$, where $\mathcal{S}_{\mathrm{M}} \in \mathbb{R}^4$ and $\mathcal{S}_{\mathrm{I}} \in \mathbb{R}^4$ are the state space of the ballistic missile and the interceptor, respectively. The dimension of $\mathcal{S}$ is reduced by substituting it with the relative state space $\mathcal{S}_{\mathrm{MI}} \in \mathbb{R}^4$, thereby speeding up learning. When the evasion is successful, the controller needs to mask the interceptor information, so a scalar is used to indicate the evasion state $\mathcal{S}_{\mathrm{B}} = \{0, 1\}$. Element 0 indicates that the evasion has not yet been completed, and element 1 indicates that the evasion has been completed. It is necessary for the controller to know how many pulse motors are available in order to determine the remaining maneuverability. It is also necessary to know the relative position to the expected re-entry point $\mathcal{S}_{\mathrm{MT}} \in \mathbb{R}^2$ and the real-time velocity of the missile $\mathcal{S}_{\mathrm{M}\_v} \in \mathbb{R}^2$.

Finally, the state space in this paper is designed as a 10-dimensional space $\mathcal{S} = \mathcal{S}_{\mathrm{MI}} \times \mathcal{S}_{\mathrm{B}} \times \mathcal{S}_{\mathrm{P}} \times \mathcal{S}_{\mathrm{MT}} \times \mathcal{S}_{\mathrm{M}\_V}$, and each element is

$$\left(s_{\mathrm{MI}\_x}, s_{\mathrm{MI}\_y}, s_{\mathrm{MI}\_V_x}, s_{\mathrm{MI}\_V_y}, s_{\mathrm{B}}, s_{\mathrm{p}}, s_{\mathrm{MT}\_x}, s_{\mathrm{MT}\_y}, s_{\mathrm{M}\_Vx}, s_{\mathrm{MT}\_Vy}\right).$$

As explained in Section 2, pulse motors are installed on the side of a spacecraft. In the early stage, as the missile and the interceptor are far apart from each other, relative position changes slowly. Therefore, there is no need for the controller to output commands with high frequency. The action space is $\mathcal{A} = \{-1, 0, 1\}$. Element 0 means either side of the missile does not ignite at the current, element 1 means motors on the lower side of the missile produce a force F_{u}, and element 2 means motors on the upper side are ignited to produce a force F_{d}. $|F_{\mathrm{u}}|$ and $|F_{\mathrm{d}}|$are constants but have opposite directions, and the directions are perpendicular to the missile's body axis. At each control moment, the controller selects an alternative command in $\mathcal{A}$ to control the ignition state according to input state $\boldsymbol{s} \in \mathcal{S}$. Actions at each time point constitute a time sequence. Action space has only three elements and looks simplistic, while penetration strategy space has a great deal of variety. The penetration strategy space is $\mathcal{L} = \times\mathcal{A}_{i=1\ldots k}$ if the controller needs to make k decisions. If, for example, 1 Hz is the control frequency and 200 s is the time span of midcourse penetration, then $\mathcal{L}$ contains 3^{100} alternative strategies.

The state space is sparse. Therefore, the value function $Q^{\pi}(\boldsymbol{s}_t, a_t)$ will converge more quickly in reinforcement learning.

The instant reward function is a crucial component of the trial-and-error approach as it guides the controller in learning the optimal strategy [33], [36]. Diverse instant reward functions indicate different behavioral tendencies. The instant reward function affects the quality of the strategy learned by reinforcement learning.

According to the goal of this paper, the instant reward function plays two roles: (1) rewards the evasion. (2) rewards the acceptable re-entry error. An ideal instant reward function is:

$$R(\boldsymbol{s}_t) = R_1(\boldsymbol{s}_t) + R_2(\boldsymbol{s}_t) \tag{7}$$

$$R_1(\boldsymbol{s}_t) = \begin{cases} r_{\mathrm{c}_1} & f_{r_1}(\boldsymbol{s}_t) = 0 \\ 0 & else \end{cases} \tag{8}$$

$$R_2(\boldsymbol{s}_t) = \begin{cases} r_{\mathrm{c}_2} & f_{r_1}(\boldsymbol{s}_t) = 0, f_{r_2}(\boldsymbol{s}_t) = 0 \\ 0 & else \end{cases} \tag{9}$$

where $r_{\mathrm{c}_1}, r_{\mathrm{c}_2} \in \mathbb{R}^+$. $f_{r_1}(\cdot)$ is the criterion for a successful evading. $f_{r_2}(\cdot)$ is the condition of acceptable re-entry error. This reward function only makes rewards based on the instantaneous state and does not guide the movement (state trajectory). Therefore, the instant reward function biases only a few specific states that satisfy $f_{r_1}(\cdot)$ and $f_{r_2}(\cdot)$ without disturbing other movements. From a theoretical view, the controller can explore the entire strategy space $\mathcal{L}$ through reinforcement learning.

However, during the random exploration process, the controller enters $f_{r_1}(\cdot)$ and $f_{r_2}(\cdot)$ is almost impossible. Moreover, since the reward for a few instantaneous states is too sparse, reinforcement learning requires massive state trajectories. Therefore, powerful computer resources are needed. To solve this problem, it is necessary to change the instant reward

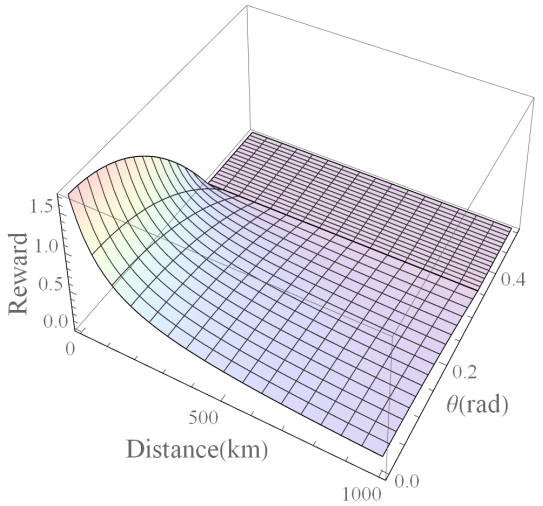

FIGURE 2. The instant reward function used in this paper with $\lambda_{th} = 0.35$, $c_1 = 7$, $c_2 = 1$, $c_3 = 0.85$, $c_4 = 2$, $c_5 = 0.01$.

function based on the instantaneous states to a function based on the process.

$R(s_t)$ is redesigned as follows:

$$R(s_t) = f(\lambda_S, l) = \begin{cases} c_4 \cdot (c_3 - c_1 \cdot \lambda_S^2) \cdot \dfrac{1}{c_5 \cdot l + c_2}, \\ \qquad |\lambda_S| < \lambda_{th} \\ 0, \quad |\lambda_S| \geq \lambda_{th} \end{cases} \tag{10}$$

In the above equation, the angle between the velocity of the ballistic missile and the sight-line (missile to expected reentry point) forms λ_S (as shown in Fig. (1)). λ_S determines the re-entry error. l is the slant distance between the missile and the expected re-entry point. $c_1 \sim c_5 \in \mathbb{R}^+$ are constants. λ_{th} is a threshold. The instant reward function used in this paper is shown in Fig. (2).

As shown in Fig. (2), when distance l is significant, the reward is not sensitive to l, which can relax the penetration strategy in penetration stage ① ② because any strategy obtains a close reward. However, varying strategies in stage ① ② will lead to a vast difference in stage ④, which helps guide the controller to evade an interceptor successfully. When l is small, λ_S greatly affects the cumulative reward, which provide guidance to the controller in obtaining a better re-entry point error.

B. STRUCTURE OF THE CONTROLLER

The deep neural network structure used in this paper is shown in Fig. (3). The traditional DQN structure lacks a description of the relative potential value of each alternative action, so the output strategy is unstable when facing complex problems, and the training takes a long time. To address this problem, a dueling architecture is introduced [37]. Suppose the parameter set of the deep neural network is $\boldsymbol{\phi}$, and $Q^\pi(s_t, a_t; \boldsymbol{\phi})$ is defined as two parts:

$$Q^\pi(s_t, a_t; \boldsymbol{\phi}) = V^\pi(s_t; \boldsymbol{\phi}) + A^\pi(s_t, a_t; \boldsymbol{\phi}) - \sum_{a'} A^\pi(s_t, a'; \boldsymbol{\phi}) / |A| \tag{11}$$

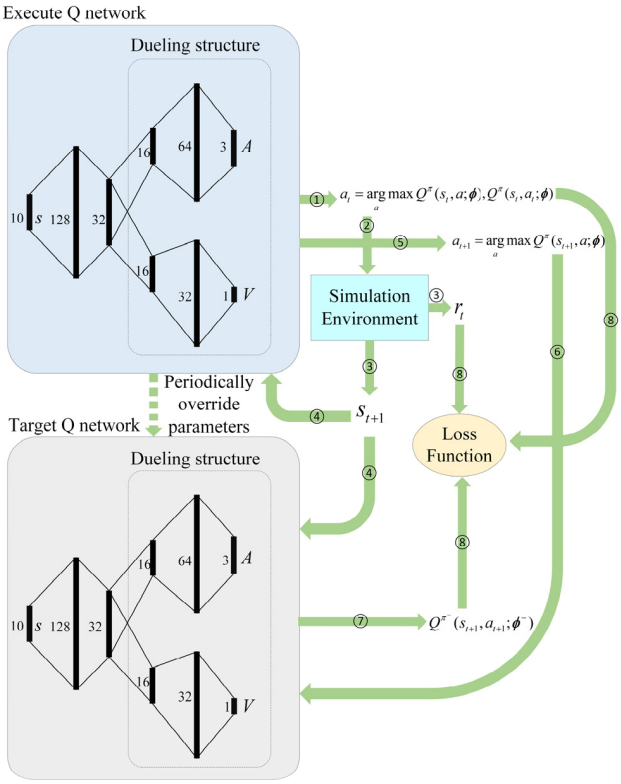

FIGURE 3. The structure and loss function of D3Q.

where $V^\pi(s_t; \boldsymbol{\phi})$ is the expected value function of the cumulative reward mentioned in Section 2.b. $A^\pi(s_t, a_t; \boldsymbol{\phi})$ is an advantage function used to characterize the potential of each alternative action. By the definition, $E[Q^\pi(s_t, a_t; \boldsymbol{\phi})] = V^\pi(s_t; \boldsymbol{\phi})$, so the last term in Eq. (11) can guarantee stability.

By introducing a target network, we are able to reduce the volatility caused by un-converged parameters. Suppose the parameter set of the target DQN is $\boldsymbol{\phi}^-$, and the parameter set of the execution DQN is $\boldsymbol{\phi}$. The target DQN $Q^{\pi^-}(s_t, a_t; \boldsymbol{\phi}^-)$ and the instant reward $r(s_t, a_t, s_{t+1})$ form the target value. The execution DQN is used to approximate the target value. That is, only the execution network parameters $\boldsymbol{\phi}$ are updated for each training. After a certain amount of training, override $\boldsymbol{\phi}$ is override with $\boldsymbol{\phi}^-$, as shown in Fig. (3). The principle of D3Q is to reduce the correlation between the target value and the selected action, avoiding overestimation of the state-action value.

C. TRAINING ALGORITHM

Much progress towards artificial intelligence has been made using supervised learning systems that are trained to replicate the decisions of human experts. However, expert data sets are often expensive, unreliable or simply unavailable, which is not a problem for reinforcement learning. The purpose of reinforcement learning is that D3Q interacts with the environment (simulation program of midcourse penetration) to obtain a memory pool and is trained based on the memory pool.

The approximate goal of D3Q is the nonlinear mapping: $\boldsymbol{s}_t \rightarrow Q^{\pi}(\boldsymbol{s}_t, a_t)$. Let $\hat{Q}^{\pi}(\boldsymbol{s}_t, a_t)$ be the real value. The r_t is the instant reward under $(\boldsymbol{s}_t, a_t)$. According to the definition of Eq. (5), the approximation value of D3Q in the current state is:

$$\tilde{y}_t = r_t + \lambda \max_a \{\hat{Q}^{\pi}(\boldsymbol{s}_{t+1}, a)\} \tag{12}$$

D3Q introduces the concept of execution networks and target networks. $\hat{Q}^{\pi}(\boldsymbol{s}_t, a_t)$ is replaced by $Q^{\pi^-}(\boldsymbol{s}_t, a_t; \boldsymbol{\phi}^-)$, and the action in the next state $\boldsymbol{s}_{t+1}$ is selected by the execution network $Q^{\pi}(\boldsymbol{s}_{t+1}, a; \boldsymbol{\phi})$, so the target value (approximation value) of D3Q is rewritten as:

$$\tilde{y}_t = r_t + \lambda Q^{\pi^-}(\boldsymbol{s}_{t+1}, \arg\max_a \left\{Q^{\pi}(\boldsymbol{s}_{t+1}, a; \boldsymbol{\phi})\right\}; \boldsymbol{\phi}^-) \tag{13}$$

During the training process, the difference between the output of D3Q and the target value $\tilde{y}_t$ is evaluated through the loss function $L(\boldsymbol{\phi})$ as follows:

$$L(\boldsymbol{\phi}) = \left(\max_a \left\{Q^{\pi}(\boldsymbol{s}_t, a; \boldsymbol{\phi})\right\} - \tilde{y}_t\right)^2 \tag{14}$$

To slow down the overfitting, the regularization term $\Gamma(\boldsymbol{\phi})$ needs to be considered in the loss function. $\Gamma(\boldsymbol{\phi})$ is the sum of the modulus of all parameters in D3Q. Eq. (14) is then modified as:

$$L(\boldsymbol{\phi}) = \left(\max_a \left\{Q^{\pi}(\boldsymbol{s}_t, a; \boldsymbol{\phi})\right\} - \tilde{y}_t\right)^2 + w\Gamma(\boldsymbol{\phi}) \tag{15}$$

where w is a constant weight and $w \in \mathbb{R}^+$.

Use $L(\boldsymbol{\phi})$ and the gradient descent method to adjust the parameters of the execution network controlled by a learning rate α:

$$\boldsymbol{\phi}_{\text{new}} = \boldsymbol{\phi}_{\text{old}} + \alpha \nabla_{\boldsymbol{\phi}} L(\boldsymbol{\phi}_{\text{old}}) \tag{16}$$

Due to the complexity of the midcourse penetration problem, it is difficult to obtain an accurate loss function through one pair of state-action rewards. Therefore, a batch gradient descent with sample size N_{b} is used to calculate the loss function:

$$\bar{L}(\boldsymbol{\phi}_{\text{old}}) = \frac{1}{N_{\text{b}}} \sum_{n=1}^{N_{\text{b}}} \left(\max_a \left(Q^{\pi}(\boldsymbol{s}_{n,t}, a; \boldsymbol{\phi}_{\text{old}})\right) - \tilde{y}_{n,t}\right)^2 + w\Gamma(\boldsymbol{\phi}) \tag{17}$$

The reinforcement learning of D3Q is shown in Algorithm 1.

Algorithm 1 Reinforcement Learning of D3Q

Δt: Kinematics integral step size;
γ: Reward discount factor;
α: Initial learning rate;
n: Target Q network update period;
ε: Random exploration probability;
T_{c}: D3Q control period;
w: Regular term weight;
N: Capacity of sample pool D;
N_{b}: Capacity of batch temporary storage area C;
$\boldsymbol{\phi}$: The initial parameters of the execution network Q;
$\boldsymbol{\phi}^-$: The initial parameters of the target network Q^-.
Start:
for *Episode* $= 1, M$ **do**
  Randomly initialize state of the ballistic missile $\boldsymbol{s}_{M_0}$, state of the interceptor $\boldsymbol{s}_{I_0}$, and expected re-entry point $\boldsymbol{s}_T$.;
  for $t = 1{:}\ \Delta T{:}T$ **do**
    if *t is the time of control* **then**
      Select an action a_i from A with the probability of ε ;
      or;
      Select an action a_i which is the current optimal action $a_i = \arg\max_a \{Q^{\pi}(\boldsymbol{s}_i, a; \boldsymbol{\phi})\}$
    end
    The ballistic missile moves according to the received command a_i, and the interceptor moves under the control of its own guidance law to get a new situation $\boldsymbol{s}_{t+1}$ and an instant reward r_i.;
    If the sample pool D is full **then** delete oldest sample;
    If $s_{\text{p}} > 0$ **then** store the quaternion $\{\boldsymbol{s}_t, a_t, \boldsymbol{s}_{t+1}, r_t\}$ as a sample in D;
    Randomly sample N_{b} samples from D and put them into C;
    Calculate the target value of the sample in C by Eq. (13), and calculate the batch loss function by Eq.(17). ;
    Perform batch gradient descent on each parameter in D3Q according to Eq. (16);
    Every n parameter iterations, update the parameters of the target network $Q\ \theta \rightarrow \theta^-$
  end
  update the ε
end

Because training samples are drawn from the memory pool, the quality of the memory pool determines the learning speed and the convergence of strategy. The traditional generation method of the memory pool is that the controller uses probability ε to perform random actions to explore new strategies and uses probability $(1-\varepsilon)$ to obtain deterministic actions from the neural network. Memory pools are traditionally created by randomly action with a probability ε and using it to experiment new strategies, while a probability $(1-\varepsilon)$ is used to produce deterministic actions calculated from the neural network. Each time the controller completes an interaction with the environment, the result is obtained from this interaction. When new data enter after the memory pool has filled, the earliest status-action-reward pairs are removed from the memory pool. As the research background of this paper is a ballistic missile flying in the midcourse, the execution times of its action (ignition of pulse motors) are limited. Thus, most of the time, the ballistic missile is in

an inactive state. When the number of ignitions is exhausted, the arbitrary output of D3Q has no effect on the final cumulative reward. Therefore, the learning goal of D3Q is to approach $Q^{\pi}(s_t, a_t; \boldsymbol{\phi})$ when the remaining pulse ignition times greater than 0. Therefore, unlike the traditional memory pool generation method, the memory pool only contains the state-action reward pairs whose remaining pulse ignition times are greater than 0.

In view of the MDP abstracting from the midcourse penetration, an approximator of $Q^{\pi}(s_t, a_t; \boldsymbol{\phi})$ is constructed based on the D3Q network proposed in Section 3.b. The scheme designed in Section 3.a is used for reinforcement learning. The parameters of the D3Q network are updated iteratively through the algorithm proposed in Section 3.c. Finally, the midcourse maneuver controller can be achieved.

IV. TRAINING AND TESTING

Section 3 introduces how to solve the maneuver controller established in Section 2. This section verifies the effectiveness of deep reinforcement learning in solving the midcourse penetration problem through numerical simulation. An analysis of the deep reinforcement learning process of D3Q is presented, and its convergence is demonstrated. Besides, many tests are performed.

A. SETTING

The scenario is set to a space with a distance of 800 km in the horizontal direction and 100 km in the vertical direction. In this space, the ballistic missile only remains a warhead part. The warhead in the simulation scenario refers to 'Minuteman-3'. The parameters of the interceptor refer to the lightweight exoatmospheric projectile-kinetic kill vehicle [38], as shown in Table 1.

The coordinates of spacecraft are defined in the ECEF. The nominal initial flight state (coordinate, velocity) of the ballistic missile is [0 km, 6600 km, 6 km/s, 0 km/s]. The nominal initial coordinate of the interceptor is [800 km, 6470 km]. The nominal expected re-entry point position is [800 km, 6525 km]. To prevent D3Q from overfitting the scenario and lacking generalization ability, random fluctuations are added to the nominal initial flight state when the scenario is initialized during the training. These random fluctuations follow uniform distributions $U(\cdot)$.

As mentioned in [35], Upon entering space, the interceptor is in an inertial flight state, and its sliding direction is determined by the preset interception point. This paper adopts the method of curing the speed in the x direction and adjusting the speed in the y direction to aim at the preset intercept point. When the slant distance of the interceptor and the ballistic missile reaches the threshold (200 km in this paper), the sensors on the interceptor can provide guidance information, and the interceptor uses the lateral pulse to correct the terminal interception error. Since the simple design of the terminal guidance law (proportional guidance law)in this paper (proportional guidance law), the kill radius of the interceptor is expanded to 80 m.

TABLE 1. Spacecraft parameters.

Parameters	Interceptor	Ballistic missile
Mass(kg)	16.7	150.0
Number of pulses	50	50
Ignition time of single pulse(s)	0.1	1.0
Specific impulse of pulse motor(s)	280.0	280.0
Flow of pulse motor(kg·s^{-1})	1.0	2.0
Initial x(km)	0	800
Initial y(km)	6600	6470
Initial Vx(km/s)	$U(5.7, 6.3)$	$U(-3.92, -5.88)$
Initial Vy(km/s)	0	Alignment algorithm
Expected x of re-entry point (km)	800	/
Expected y of re-entry point (km)	$U(6450, 6600)$	/

TABLE 2. Training hyperparameters.

Hyperparameters	Value
λ	0.99
α	10^{-4}
n	1000
ε	0.1
T_c	1 s
w	10^{-4}
N	2048000
N_b	512
Weight initialization	$N(0, 0.02)$
Bias initialization	$N(0, 0.02)$

The parameters of the instant reward are shown in Fig. (2). The training hyperparameters are shown in Table 2. The D3Q adopts a multilayer fully connected layer competition architecture. The number of neurons in each layer is shown in Fig. (3)b. The activation function of the fully connected layer is Leaky-ReLU (the negative slope is 0.1), and the activation function of the output layer is linear.

B. TRAINING ANALYSIS

The CPU of the training platform is an AMD Ryzen5 3600@4.2 GHz. The RAM is 8 GB × 2 DDR4@3733 MHz. A GPU is not selected for the parameter iteration because the D3Q is straightforward and computation is concentrated on the calculation of spacecraft models. Spacecraft models are written in C++ and packaged as a dynamic link library (DLL). The D3Q and training is implemented by Python. Interaction between the D3Q and the virtual scenario is realized by calling the DLL. Through 65 h 17 m of reinforcement learning, 20815 episodes are completed, and D3Q performs 2.525 million iterations. In Fig. (4)∼Fig. (6), the training

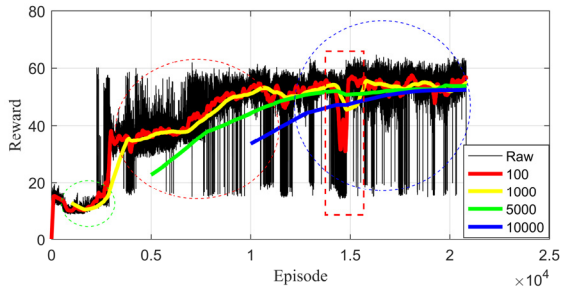

FIGURE 4. Cumulative reward and moving average of each episode in training.

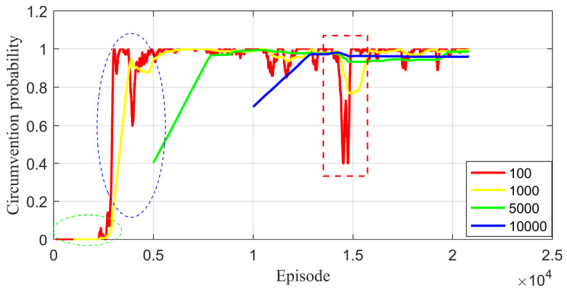

FIGURE 5. Moving average penetration probability of each episode in training.

process is shown, and in Table 3, the training statistics are presented.

It can be seen in Fig. (4) that the training of D3Q can be divided into three stages: the exploration stage (shown by the green dashed circle), the optimization stage of active evasion (shown by the red dashed circle), and the optimization stage of re-entry error (shown by the blue dashed circle). The cumulative rewards in this stage (less than 20) are low since D3Q has no intelligence and moves randomly in the exploration phase. As the trial evolves, D3Q chooses some behaviors that will expand the cumulative reward, swiftly acquiring the intelligence to evade the interceptor. The accumulative reward increases significantly in this stage, demonstrating D3Q can learn excellent strategies quickly and effectively. The cumulative reward of most episodes has a clear advantage over the exploration stage, indicating that this stage has learned to evade the interceptor and then collects a multistep instant reward after evasion. Because D3Q has not yet fully adapted to the random scenario and there is a 10% probability of random action ($\varepsilon = 0.1$), the cumulative rewards of a few episodes are still less than 20. In the optimization stage of re-entry error, the cumulative reward gradually increases. According to the moving average curve of each window size, the cumulative reward exhibits smooth convergence, which is a result of the convergence of the D3Q and the reinforcement learning algorithm.

In Fig. (5), it can be seen that D3Q cannot evade the interceptor at all in the exploration stage (shown by the green dashed circle), but through trial and study in the scenario, some excellent strategies are obtained by an extremely low probability. D3Q learns evasion very quickly after learning

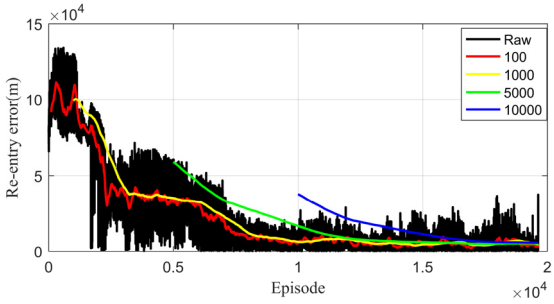

FIGURE 6. Re-entry error and moving average of each episode in training.

TABLE 3. Reinforcement learning statistics.

Window size(episode)	Mean of cumulative reward	Penetration probability	Mean of re-entry error(m)
100	56.73	1.00	3141.52
1000	54.78	0.96	4689.14
5000	53.76	0.96	5357.69
10000	52.49	0.96	5806.32

these strategies(shown by the blue dashed circle). From the moving average curve of penetration probability with window sizes of 100 and 1000, it can be found that a modal collapse phenomenon shown in the red dashed box appears in Fig. (5) (also reflected in Fig. (4)). Because D3Q overemphasizes the importance of the re-entry error in the optimization stage of the re-entry point, a local pole that is not conducive to evading the interceptor is stuck. In subsequent training, D3Q jumps out of this local pole and gains intelligence to balance re-entry error and evasion.

It can be seen in Fig. (6) that D3Q is very stable in the optimization stage of re-entry error. The re-entry error gradually converges from approximately 100 km to 5 km. Table 3 shows that D3Q achieves a 100% penetration probability in the last 100 episodes of training, and the re-entry error is concentrated in a small range (approximately 3 km). Judging from the statistical results of the last 1000, 5000, and 10,000 episodes, D3Q can still maintain a high average cumulative reward and a small variance, and the penetration probability and re-entry error are within acceptable ranges. In the final period of training, D3Q has adapted to the scenario with multiple random factors.

C. TEST ANALYSIS

The D3Q performs penetration tests in virtual scenarios to reveal the specific strategy that it has learned from training. To verify the adaptability of D3Q facing different expected re-entry points, the expected re-entry heights of 6475 km, 6525 km and 6575 km are tested. Since random exploration is no longer needed, ε is set to 0. For each of the three re-entry heights, 1,000 episodes are carried out while keeping random factors in mind to simulate the uncertain nature of a battlefield. Test results are shown in Fig. (7).

It can be seen in subplot (a) and (d) of Fig. (7) that when facing a low expected re-entry point, the strategy adapted by D3Q is to bypass the interceptable zone of the interceptor

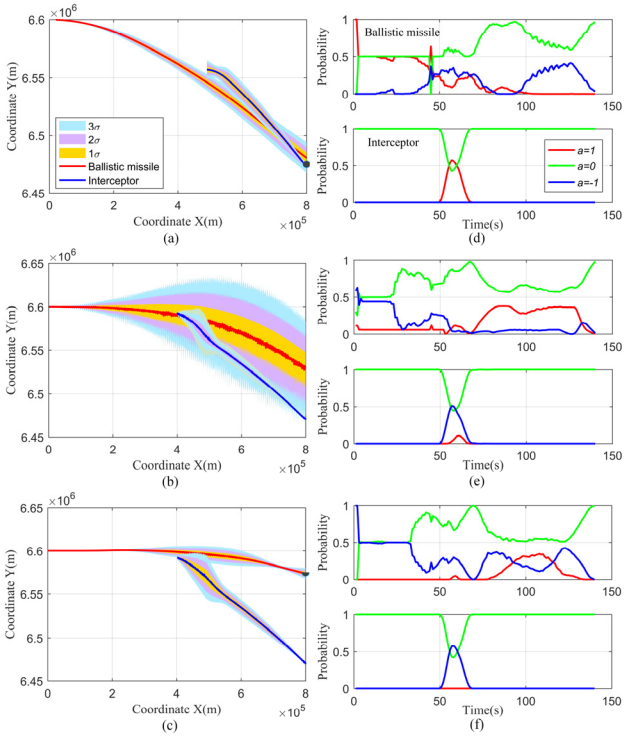

FIGURE 7. Test results of (a)(d) expected re-entry heights of 6475 km; (b)(e) expected re-entry heights of 6525 km; (c)(f) expected re-entry heights of 6575 km.

downwards. The triple variance (3σ) of ballistic is small, so the strategy is stable. It can also be seen in subplot (d) of Fig. (7) that D3Q has diverse purposes in different time periods. From 0 to 40 s, as the probability of action 1 is much greater than that of action -1, D3Q mainly ignites upper pulse motors to evade the interceptor as soon as possible. From 40∼90 s, the interceptor enters the terminal guidance stage, and D3Q adapts the corresponding ignition control according to the real-time situation. If the estimated missing amount was large, the re-entry point would be corrected in advance. If the estimated missing amount was small, further maneuvers would be taken to expand the missing amount. In the post-90 s, D3Q mainly takes an upward maneuver for terminal correction because in the early stage of penetration, too many downward maneuvers have been used to evade the interceptor. Subplot (c) and (f) of Fig. (7) show that D3Q adapts a strategy of bypassing the interceptable zone upwards when facing a high expected re-entry point. In the early stage, it mainly uses an upward maneuver to bypass the interceptor. A partial downward maneuver is adopted to reduce re-entry error. The above phenomenon shows it is clearly that D3Q can provide penetration strategies when facing different expected re-entry heights.

Subplot (b) of Fig. (7) shows that the strategy is very unstable when D3Q faces a medium height point, and the maneuver of the interceptor is also unstable. For different interceptions, D3Q adapts different penetration strategies. In this situation, the main factor affecting the penetration strategy is no longer the height of the expected re-entry point but the real-time situation of both offense and defense, which reflects D3Q's ability to adapt to the uncertainty of the battlefield. To reveal this intelligence, the expected re-entry point is fixed to 6525 km, and scenarios with different initial energy states are tested. The parameters of the 4 test scenarios are shown in Table 4, and the orbits of all scenarios are shown in Fig. (8).

TABLE 4. Initial parameters in medium height test.

	Initial Vx of M(m/s)	Initial Vy of M(m/s)	Initial Vx of I(m/s)	Initial Vy of I(m/s)
Scenario 1	5700	0	-3920	Alignment algorithm
Scenario 2	5700		-3920	
Scenario 3	6300		-5880	
Scenario 4	6300		-5880	

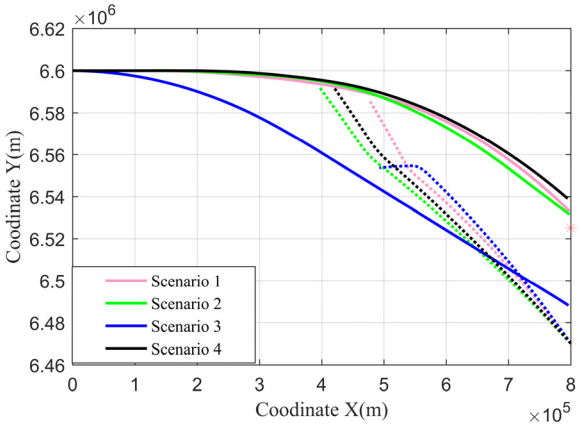

FIGURE 8. Orbits of all scenarios in the medium height re-entry test.

TABLE 5. Statistics of different strategies.

Strategy	Mean of cumulative reward	Evade probability	Mean of reentry error(m)	Mean of Missing amount(m)
Random	17.25	0%	/	37.24
Upward	20.89	100%	80892.20	12374.11
Downward	22.91	100%	99273.58	9705.09
D3Q	53.25	100%	4939.82	5599.20

In scenarios 1, 2, and 4, D3Q bypasses the interceptable zone from the upper side of the interceptor, but in scenario 3, D3Q takes a downward evasion strategy. In scenario 3, the interceptor's Vx is slow, and the velocity of the ballistic missile is faster. Therefore, the interceptor has a bigger inclination during inertial taxi and has a larger interceptable zone for targets passing above. If D3Q still took the strategy of flying upwards, it would need to make a downward maneuver when it is close to the expected re-entry point, which would reduce the relative inclination and therefore enter the interceptable zone.

To better illustrate the intelligence of the D3Q, three penetration strategies (random maneuver, continuous upward maneuver, and continuous downward maneuver) are compared. The results are shown in Table 5.

The penetration probability of the random maneuver strategy is 0. Obviously, it is impossible to evade the interceptor at all. The re-entry error is beyond the acceptable range although continuous upward maneuvering and continuous downward maneuvering can evade the interceptor by 100% probability. D3Q also achieves a 100% evade probability, and the re-entry point error is approximately 5 km, which has significant advantages. Comparing the missing amount of interceptors, when facing D3Q, the missing amount is much smaller than other strategies. This shows that D3Q does not pursue to evade the interceptor blindly but bypasses the interceptable zone at a relatively close distance. This saves many meaningless pulses and reserves considerable maneuverability to optimize the re-entry error.

V. CONCLUSION

A D3Q-based controller is constructed in this paper for maneuver control of a ballistic missile in midcourse penetration. Reinforcement learning is then implemented to make it more intelligent. Compared with traditional methods, the controller and its training algorithm proposed in this paper have the following advantages: (1) Through training in virtual scenarios, available penetration strategies can be obtained independently without relying on any prior human knowledge or labeled training data. (2) D3Q ensures that the ballistic missile can evade an interceptor while taking the re-entry error into account, and it can autonomously respond to a variety of battlefield situations with high intelligence. (3) D3Q is entirely data-driven, simple in structure, and consumes very few computing resources.

REFERENCES

[1] E.-J. Song and M.-J. Tahk, "Three-dimensional midcourse guidance using neural networks for interception of ballistic targets," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 38, no. 2, pp. 404–414, Apr. 2002.

[2] J. J. Dougherty and J. L. Speyer, "Near-optimal guidance law for ballistic missile interception," *J. Guid., Control, Dyn.*, vol. 20, no. 2, pp. 355–361, Mar. 1997.

[3] L.-Y. Li, F.-X. Liu, and Y.-Y. Mei, "Modeling and simulation system design of tactical ballistic missile interception based on UML," in *Proc. 4th Int. Conf. Comput. Inf. Sci.*, Chongqing, China, Aug. 2012, pp. 53–56.

[4] N. M. Qi, Q. H. Zhou, N. G. Cui, and C. Dong, "Penetration trajectory design based on defense system delay and maximum prediction errors in ballistic midcourse," *J. Chin. Inert Technol.*, vol. 24, no. 3, pp. 409–414, 2016.

[5] I. Forte, A. Steinberg, and J. Shinar, "The effects of non-linear kinematics in optimal evasion," *Optim. Control Appl. Methods*, vol. 4, no. 2, pp. 139–152, Apr. 1983.

[6] F. Imado and S. Miwa, "Fighter evasive maneuvers against proportional navigation missile," *J. Aircr.*, vol. 23, no. 11, pp. 825–830, Nov. 1986.

[7] G. M. Zhang, P. Y. Gao, and Q. G. Tang, "The method of the impulse trajectory transfer in a different plane for the ballistic missile penetrating missile defense system in the passive ballistic curve," *J. Astro.*, vol. 29, no. 1, pp. 89–94, 2008.

[8] Q. X. Wu and W. H. Zhang, "Research on midcourse maneuver penetration of ballistic missile," *J. Astro.*, vol. 27, no. 6, pp. 1243–1247, 2006.

[9] K. N. Zhang, H. Zhou, and W. C. Chen, "Trajectory planning for hypersonic vehicle with multiple constraints and multiple maneuvering penetrations trategies," *J. Ballistics*, vol. 24, no. 3, pp. 85–90, 2012.

[10] X. Zhao, W. Qin, X. Zhang, B. He, and X. Yan, "Rapid full-course trajectory optimization for multi-constraint and multi-step avoidance zones," *J. Solid Rocket Technol.*, vol. 42, no. 2, pp. 245–252, 2019.

[11] T. Yang, "Research on maneuver penetration strategy of ballistic missile based on differential game theory," M.S. thesis, Dept. Electron. Eng., National Univ. Defense Technol., Changsha, China, 2015.

[12] S. M. Sun and B. Z. Zhou, "Differential strategy and its application in ballistic missile maneuver penetration," in *Proc. 22nd Flight Mech. Flight Test Acad. Conf. Chinese*, Chengdu, China, 2006, pp. 207–211.

[13] Y. Xian, H. P. Tian, J. Wang, and J. Q. Shi, "Research on intelligent maneuver penetration of missile based on differential game theory," *Fli. Dyn.*, vol. 32, no. 1, pp. 70–83, 2014.

[14] R. Bardhan and D. Ghose, "Nonlinear differential games-based impact-angle-constrained guidance law," *J. Guid. Control Dyn.*, vol. 38, no. 3, pp. 1–19, 2015.

[15] C. Guo, X.-G. Liang, and J.-W. Wang, "Multi-model soft switching tracking control for near-space interceptor based on the disturbance observer," *Proc. Inst. Mech. Eng., I, J. Syst. Control Eng.*, vol. 230, no. 10, pp. 1077–1092, Nov. 2016.

[16] C. Guo, C. Song, Y.-J. Zhao, and J.-W. Wang, "Nonlinear model predictive control for near-space interceptor based on finite time disturbance observer," *Int. J. Aeronaut. Space Sci.*, vol. 19, no. 4, pp. 945–961, Dec. 2018.

[17] J. L. Sun and C. S. Liu, "An overview on the adaptive dynamic programming based missile guidance law," *J. Autom Sinica.*, vol. 43, no. 7, pp. 1101–1113, 2017.

[18] J. W. Chen, Y. H. Cheng, and B. Jiang, "Mission constrained spacecraft attitude control system on orbit reconfiguration algorithm," *J. Astro*, vol. 38, no. 9, pp. 989–997, 2017.

[19] R. S. Sutton and A. G. Barto, "Reinforcement learning," *Bradford Book*, vol. 15, no. 7, pp. 665–685, 1998.

[20] G. E. Hinton, L. Deng, D. Yu, G. E. Dahl, A.-R. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. N. Sainath, and B. Kingsbury, "Deep neural networks for acoustic modeling in speech recognition," *IEEE Signal Process. Mag.*, vol. 29, no. 6, pp. 82–97, Oct. 2012.

[21] X. Zou, R. Yang, C. Yin, Z. Nie, and H. Wang, "Deploying tactical communication node vehicles with AlphaZero algorithm," *IET Commun.*, vol. 14, no. 9, pp. 1392–1396, Jun. 2020.

[22] G. Cuccu, J. Togelius, and P. Cudré-Mauroux, "Playing atari with few neurons," *Auto. Agents Multi-Agent Syst.*, vol. 35, no. 2, pp. 1–23, Oct. 2021.

[23] C. Cheng and K. K. Parhi, "Fast 2D convolution algorithms for convolutional neural networks," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 67, no. 5, pp. 1678–1691, May 2020.

[24] D. T. Yu, H. Wang, L. S. Li, and H. Y. Wang, "Attack area modeling of kinetic kill vehicle head-on interception with energy constraint," *J. Astro*, vol. 38, no. 7, pp. 704–713, 2017.

[25] J. H. Xiong, S. J. Tang, J. Guo, and D. L. Zhu, "Design of variable structure guidance law for head-on interception based on variable coefficient strategy," *Acta Armamentarii*, vol. 35, no. 1, pp. 134–139, 2014.

[26] Z. Qiang, X. Hou, and Y. Cai, "Strategy and simulation on the energy management steering maneuver of THAAD interceptor," *Tactical Missile Technol.*, vol. 46, no. 1, pp. 46–52, 2015.

[27] Z. Zhang, H. Wang, and W. X. Jing, "Cooperative tracking and guidance algorithm for exoatmospheric multiple interceptors," *J. Astro*, vol. 7, no. 40, pp. 785–793, 2019.

[28] Y. Xian and W.-H. Si, "Research on midcourse maneuvering penetration guidance law of ballistic missile based on genetic algorithm," in *Proc. IEEE Int. Conf. Intell. Comput. Intell. Syst. (ICIS)*, Shanghai, China, Nov. 2009, pp. 188–191.

[29] Z. C. Huang, Y. S. Zhang, H. Chai, and Y. Liu, "Research on interception along trace based on standard missile-3," *Comput. Meas. Control*, vol. 26, no. 23, pp. 137–140, 2018.

[30] R. S. Sutton and A. G. Barto, "Reinforcement learning: An introduction," *IEEE Trans. Neural Netw.*, vol. 9, no. 5, p. 1054, Sep. 1998.

[31] O. Alagoz, H. Hsu, A. J. Schaefer, and M. S. Roberts, "Markov decision processes: A tool for sequential decision making under uncertainty," *Med. Decis. Making*, vol. 30, no. 4, pp. 474–483, Jul. 2010.

[32] P. Hernandez-Leal, B. Kartal, and M. E. Taylor, "A survey and critique of multiagent deep reinforcement learning," *Auto. Agents Multi-Agent Syst.*, vol. 33, no. 6, pp. 750–797, Nov. 2019.

[33] S. Br, D. Turner, and K. Mohanty, "Multi agent deep Q-network approach for online job shop scheduling in flexible manufacturing," in *Proc. ICMSMM*, Chennai, India, 2020, pp. 78–86.

[34] H. Sasaki, T. Horiuchi, and S. Kato, "Experimental study on behavior acquisition of mobile robot by deep Q-network," *J. Adv. Comput. Intell. Intell. Inform.*, vol. 21, no. 5, pp. 840–848, 2017.

[35] S. Yoon and K.-J. Kim, "Deep q networks for visual fighting game AI," in *Proc. IEEE Conf. Comput. Intell. Games (CIG)*, New York, NY, USA, Aug. 2017, pp. 306–308.
[36] G. Guo and Y. P. Wang, "Signal priority control for trams using deep reinforcement learning," *J. Autom Sinica*, vol. 45, no. 12, pp. 2366–2377, 2019.
[37] G. Sun, K. Xiong, and G. O. Boateng, "Resource slicing and customization in RAN with dueling deep Q-network," *J. Netw. Comput. Appl.*, vol. 157, no. 3, pp. 1–13, 2020.
[38] Q. X. Wu, Z. Y. Wang, and W. H. Zhang, "System optimization design of maneuver penetration motor in the midcourse of ballistic missile," *J. Solid Rock Technol*, vol. 30, no. 4, pp. 278–281, 286, 2007.

LIANG JIANG is currently pursuing the Ph.D. degree with the Academy of Astronautics, Nanjing University of Aeronautics and Astronautics. His research interests include multi-agent deep reinforcement learning and flight simulation.

YING NAN received the Ph.D. degree from Northwestern Polytechnical University, in 1993. He is currently a Professor with the Nanjing University of Aeronautics and Astronautics. His main research interests include flight dynamics and control, aircraft design, and flight simulation.

ZHI-HAN LI is currently pursuing the Ph.D. degree with the Academy of Astronautics, Nanjing University of Aeronautics and Astronautics. His research interests include trajectory planning and flight simulation.