Journal of Systems Engineering and Electronics
Vol. 33, No. 5, October 2022, pp.1173 – 1185

Hierarchical reinforcement learning guidance with threat avoidance

LI Bohao[1,2,3], WU Yunjie[1,2,3], and LI Guofei[4,*]

1. State Key Laboratory of Virtual Reality Technology and System, Beihang University, Beijing 100191, China; 2. School of Automation Science and Electrical Engineering, Beihang University, Beijing 100191, China; 3. Science and Technology on Aircraft Control Laboratory, Beijing 100191, China; 4. School of Astronautics, Northwestern Polytechnical University, Xi'an 710072, China

Abstract: The guidance strategy is an extremely critical factor in determining the striking effect of the missile operation. A novel guidance law is presented by exploiting the deep reinforcement learning (DRL) with the hierarchical deep deterministic policy gradient (DDPG) algorithm. The reward functions are constructed to minimize the line-of-sight (LOS) angle rate and avoid the threat caused by the opposed obstacles. To attenuate the chattering of the acceleration, a hierarchical reinforcement learning structure and an improved reward function with action penalty are put forward. The simulation results validate that the missile under the proposed method can hit the target successfully and keep away from the threatened areas effectively.

Keywords: guidance law, deep reinforcement learning (DRL), threat avoidance, hierarchical reinforcement learning.

DOI: 10.23919/JSEE.2022.000113

1. Introduction

The guidance law is of importance for the missiles to meet the demand of the military task [1−3]. In view that the external threats may exist to destroy the operation [4], the regular guidance law under such situation cannot satisfy the strict requirements [5,6]. Therefore, it is necessary to make the guidance strategy more flexible and intelligent [7].

Machine learning has the merits of autonomous sensing and adapting to dynamic environments [8−10], in which the deep reinforcement learning (DRL) is a prominent algorithm for its excellent pattern recognition and decision-making performance [11]. Due to such attractive feature, it has been widely employed in target tracking, obstacle avoidance, path planning and other research fields related to guidance [12−14]. However, the existing DRL guidance algorithms mainly focus on the low-speed aircraft such as unmanned aeriel vehicles (UAVs) [15]. A dueling double deep Q-networks algorithm based on global situation information was proposed in [16] to improve the survival probability of UAV. Introducing supervised learning into a two-stage reinforcement learning (RL) framework, [17] realized multi-UAV collision avoidance. An RL reward function for multi-UAV cooperative searching was designed in [18], which can perform the mission effectively in the sea area without prior information. Different from the UAVs, the speed of missile is faster and the direction is more difficult to control [19,20]. An adaptive guidance system using the reinforcement meta-learning with recurrent network was proposed in [21]. A model-based DRL method was presented in [22] to predict the model of the guidance dynamics, and the predicted result was incorporated into a model predictive path integral control framework.

The maneuvering target arouses another challenge for missile guidance with DRL [23−25]. A DRL algorithm with a coarse-to-fine scheme was proposed in [26], which is used to address the aspect ratio variation in target tracking. A multi-agent deep deterministic policy gradient (DDPG) algorithm was proposed in [27], which perform target assignment and path planning simultaneously. A novel DDPG missile guidance law, whose neural network has identical inputs with proportional navigation guidance (PNG), was proposed in [28] with satisfactory robustness. The numerous DRL guidance algorithms just design the reward function according to the distance between the missile and the target, which may lead to unstable training results for the absence of angle information.

Obstacle avoidance should be considered during the guidance law design for self-security, much effort has been devoted to improving the issue [29,30]. A typical

Manuscript received February 24, 2021.

*Corresponding author.

This work was supported by the National Natural Science Foundation of China (62003021; 91212304).

obstacle avoidance problem is how to guide a missile against a maneuvering target while satisfying a circular no-fly zone constraint. The authors in [31] distorted the real space to make the boundary of a circular obstacle become a straight line, then the proportional navigation law is used to steer the missile to the target. An obstacle avoidance guidance algorithm was derived in [32] based on linear quadratic optimal control. An artificial potential field methodology was developed for path planning of cruise missile in [33]. However, these traditional obstacle avoidance algorithms are mostly applicable in stationary scenes. Once the size or position of the obstacle is changed dynamically, both the parameters and trajectories thereupon ought to be readjusted. DRL has the strong decision-making ability which is suitable for the complex environment with obstacle threat. An effective DRL algorithm can avoid various obstacle threat rids of adjusting parameters manually. DRL autonomous navigation of a group of mini-robots in a multi-agent collaborative environment was investigated in [34], and the double Q-learning algorithm was employed to avoid the collision between robots. Transforming the path planning problem into a partially observable Markov decision process, a recurrent deterministic policy gradient was proposed in [35] for navigation in complex environments. For maneuvering target tracking, an improved DDPG algorithm was proposed in [36], which improved the stability and the convergence rate. The previous DRL-based obstacle avoidance methods exploit the distance between the agent and the surface of obstacles as a part of the state space, and the obstacle avoidance penalty is added in the reward function. However, it is unavailable to measure the distance between the missile and the so-called obstacle surface directly in some cases. Besides, obstacle avoidance strategies may lead to serious acceleration chattering which is harmful for the missile actuator during the guidance execution.

The motivation of our research is to improve the DRL guidance performance along with the ability of obstacle avoidance. Inspired by the mentioned issues, we propose an improved hierarchical reinforcement learning (HRL) guidance algorithm based on DDPG.

The main contributions of our work are formulated as follows:

(i) The performance of DDPG guidance is enhanced by employing the line-of-sight (LOS) rate information for the reward function. Such treatment is effective to achieve a stable training result, and conducive for the missile to strike the maneuvering target.

(ii) The threat avoidance for circular no-fly zone is realized by the extended state space. The state space is extended by analyzing the distance between the missile and the edge of the threat area. Via setting the termination criterion related to the state space, the proposed guidance algorithm can avoid the obstacle threat and guarantee self-security.

(iii) The harmful chattering in previous DRL guidance is attenuated by the improved HRL framework. A DDPG-based HRL strategy is presented with two training stages. The chattering of acceleration can be reduced. The target striking and threat avoidance can be achieved by the reward functions relevant to the LOS angle and the extended state space.

The structure of this paper is formulated as follows. Section 2 introduces the dynamic models of guidance and the preliminary. Our main achievements of the research are elaborated in Section 3, including a DRL guidance framework, the environment state space and an HRL algorithm for missile guidance in restrained area. Section 4 gives the simulation results and discussions. Section 5 summarizes this paper and prospects the future research.

2. Preliminary and description

In this section we first give the preliminaries and a brief description including a DRL algorithm——DDPG, which can be used to perform guidance tasks with continuous action space.

2.1 Dynamic model for guidance

The 3D dynamic model of guidance can be divided into two perpendicular 2D models. For simplicity, the dynamic model is constructed according to the relationship between the missile and the target in the horizontal plane. We use polar coordinate to represent the relative position of the missile and the target. The engagement geometry is shown in Fig. 1, where M and T represent the missile and the target, respectively.

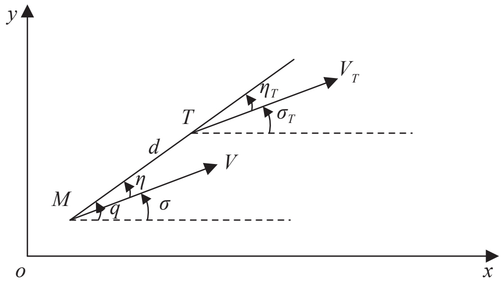


Fig. 1 Geometry of the engagement scenario

The missile-target engagement kinematics can be described by

$$\begin{cases} \dot{d} = V_T\cos\eta_T - V\cos\eta \\ d\dot{q} = V\sin\eta - V_T\sin\eta_T \\ q = \sigma + \eta \\ q = \sigma_T + \eta_T \end{cases} \tag{1}$$

where d denotes the distance between the missile and the target, q represents the LOS angle. V, η and σ denote the speed, the heading error and the flight-path angle of the missile. V_T, η_T and σ_T denote the speed, the heading error and the flight-path angle of the target.

2.2 DDPG and guidance

The framework of DDPG guidance is shown in Fig. 2, where S_t, A_t, and R_t denote the state, action and reward in timestep t respectively. The framework mainly contains environment description and the DDPG algorithm.

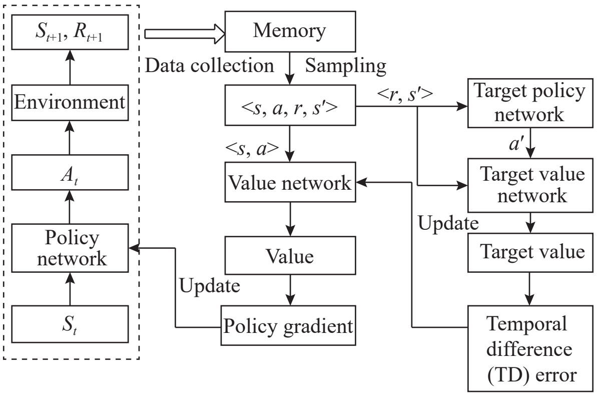


Fig. 2 Flow chart of DDPG

2.2.1 Environment description

There are several modules for environment description.

The first module is to sense the information of the missile and the target, and generate the environment state. Assuming that only the normal accelerations can be exploited on the missile, which means the speed is constant. Then the state at timestep t can be expressed as $S_t = \{d_t, q_t, \eta_t, \eta_{Tt}\}$.

The second one is to receive the action and change the environment state. Action A_t is output by policy network, which generates the normal acceleration command. The missile receives the command and changes the state to S_{t+1}.

The third one is to produce the reward. The reward functions reflect characteristics and physical meanings of different DRL tasks.

2.2.2 DDPG

DDPG is the key factor of our guidance framework, which receives the environment description data, estimates the state value, modifies the policy network, and outputs accelerations to control the velocity direction of missile.

DDPG collects a set of data $\{S_t, A_t, R_{t+1}, S_{t+1}\}$ from environment description and save it in the memory during each timestep. We can sample a batch of date $<s, a, r, s'>$ from memory when training the DDPG.

Define G_t as the sum of the discounted rewards after timestep t:

$$G_t = \sum_{\tau=0}^{+\infty} \gamma^{\tau} R_{t+\tau} \tag{2}$$

where $\gamma \in [0, 1]$ is the discount rate.

The policy network is the map from state to action, which can be defined as a function:

$$a = \pi(s;\theta) \tag{3}$$

where θ refers to the parameters of the policy network.

The task of training is to find a set of parameters θ to make $\mathrm{E}_\pi[G_0]$ maximum which denotes the expected value of G_0 when the agent follows policy π. The value network is used to evaluate the expected value of G_t. It can be defined as

$$q_\pi(s,a;\omega) = \mathrm{E}_\pi[G_t | S_t = s, A_t = a] \tag{4}$$

where ω is the parameters of the value network.

The most important step of training is to update the networks. The first is the value network. For stability, DDPG uses target networks to update the value functions, in each step:

$$y = r + \gamma q_\pi(s', a'; \omega')_{a'=\pi(s';\theta')} \tag{5}$$

where ω' and θ' denote the parameters of target value network and target policy network respectively. With TD learning, the parameters ω can be updated by

$$\omega \leftarrow \omega + \alpha[y - q_\pi(s,a;\omega)]\nabla_\omega q_\pi(s,a;\omega). \tag{6}$$

The second is policy network. Since the target of training is to find a set of parameters θ that make $\mathrm{E}_\pi[G_0]$ maximum, we calculate the gradient of $\mathrm{E}_\pi[G_0]$ at first, which is shown as

$$\nabla_\theta \mathrm{E}_\pi[G_0] = \sum_{t=0}^{+\infty} \mathrm{E}[\gamma^t \nabla_\theta \pi(s;\theta) \nabla_a q(s,a)]. \tag{7}$$

The parameters θ can be updated according to

$$\theta \leftarrow \theta + \beta\gamma^t \nabla_\theta \pi(s;\theta) \nabla_a q(s,a). \tag{8}$$

The steps of DDPG can be seen in Algorithm 1.

Algorithm 1 DDPG

1. Initialize the policy network $\pi(s)$ and the value net-

work $q(s,a)$ with parameters θ and ω; Initialize the target policy network $\pi'(s)$ and the target value network $q'(s,a)$ with parameters $\theta' \leftarrow \theta$ and $\omega' \leftarrow \omega$; Initialize the learning rate of target network ε, batch size N, memory R;

2. Take actions according to the state $A = \pi(S)$;

3. Execute the action A, receive the reward R, acquire new state S';

4. Save $\{S, A, R, S'\}$to the memory;

5. Sample a batch size of N data $\{(s,a,r,s')\}^N$ from memory randomly;

6. Update the policy network and value network:

$$y = r + \gamma q_\pi(s', a'; \omega')_{a'=\pi(s';\theta')},$$

$$\omega \leftarrow \omega + \alpha[y - q_\pi(s,a;\omega)]\nabla_\omega q_\pi(s,a;\omega),$$

$$\theta \leftarrow \theta + \beta\gamma^t \nabla_\theta \pi(s;\theta)\nabla_a q(s,a).$$

7. Update the target networks:

$$\omega' \leftarrow \varepsilon\omega + (1-\varepsilon)\omega',$$

$$\theta' \leftarrow \varepsilon\theta + (1-\varepsilon)\theta'.$$

8. Back to Step 2 until the iterations reach the maximum number of training.

3. Proposed method

In this section, LOS-based reward functions are proposed for missile guidance to strike the maneuvering target. An extended state space and a new termination criterion are adopted for threat avoidance. An HRL algorithm is proposed to reduce the acceleration chattering.

3.1 Reward functions

The kernel of RL is to use the reward to estimate the expected value of G_t and get the optimal policy. The option of the reward function is closely related to the quality of training results.

To reduce the distance between the missile and the target, a transition reward $r_{\rm tr}$ is chosen as

$$r_{\rm tr} = \frac{d_{t-1} - d_t}{\Delta t} \tag{9}$$

where d_t is the distance between the missile and the target in time step t. The more the distance reduced, the more the reward obtained. The success of the mission is evaluated by judging whether the distance is less than a threshold $d_{\min}$. A terminal reward $r_{\rm te}$ is given by

$$r_{\rm te} = \begin{cases} k, & d_t < d_{\min} \\ 0, & \text{otherwise} \end{cases} \tag{10}$$

where k is a positive constant.

3.1.1 Heading error reward

Note that (9) and (10) do not make full use of angle information, it is difficult to be applied against the maneuvering target. The reward function can be improved by introducing heading error reward to facilitate the training result more stable.

The heading error reward r_{g1} is presented by

$$r_{g1} = \exp\left(-\frac{|\eta_t|}{\pi}\right) - 1. \tag{11}$$

The reward (11) aims at making the velocity vector of the missile oriented to the target. r_{g1} is the maximum if the heading error of the missile velocity vector becomes zero. Once η is larger than π/2, the missile will fly away from the target.

3.1.2 LOS rate reward

Heading error reward may lead to the relative speed direction tending to the LOS, and cannot attack the target from omni direction. To overcome this issue, the LOS rate reward is given by

$$r_{g2} = \exp\left(-\frac{|\dot{q}|}{\pi}\right) - 1. \tag{12}$$

The reward (12) aims at making the LOS move parallel. No matter what maneuver the target makes, the components of missile velocity and target velocity perpendicular to the LOS are equal.

In combination of those guidance rewards, we achieve the total reward function:

$$r = \lambda_1 r_{\rm tr} + \lambda_2 r_{\rm te} + \lambda_3 r_g \tag{13}$$

where λ_1, λ_2 and λ_3 represent the weights.

3.2 Threat avoidance

3.2.1 Adaptive proportional navigation

The obstacle threat forms a risky area which can be represented as a circular area (O, rad), where O and rad denote the threat center and the threat radius, respectively.

The threat avoidance process of the adaptive PNG algorithm is shown in Fig. 3, where M denotes the missile, V denotes the missile velocity. MA and MB are tangent with the edge of the risky area. When the threat center is detected by the missile detects, the threat level will be evaluated with the premise that the velocity direction is unchanged. Then the adaptive PNG algorithm performs the avoidance task by choosing MA or MB as the threat-avoiding direction, and guiding the missile into the flight safety zone.

In the adaptive PNG algorithms, threat avoidance and target attacking are two separated sub-tasks. This characteristic may lead to the loss of guidance information. The algorithm ignores the state of target and performs the same threat avoidance action when the relative velocity and position between missile-threat are unchanged.

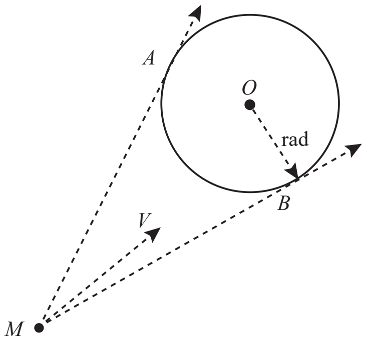


Fig. 3 Threat avoidance of adaptive PNG

However, threat avoidance and target attacking are treated as an indivisible task in DRL guidance. All the received information is involved in the state space and maps to the action of missile by the policy network. Therefore, DRL considers all the received information in the process of making decisions, which facilitates the missile getting a global optimal solution.

3.2.2 Extended state space

To achieve the threat avoidance, we must extend the state space and add threat information into it. In many DRL obstacle-avoidance system, there are sensors to measure the distance between the agent and the surface of obstacles as in Fig. 4, where d_0 is the distance between the missile and the target, d_i ($i \in [1, n]$) denotes the minimum distance between agent and surface in the direction of sensor i.

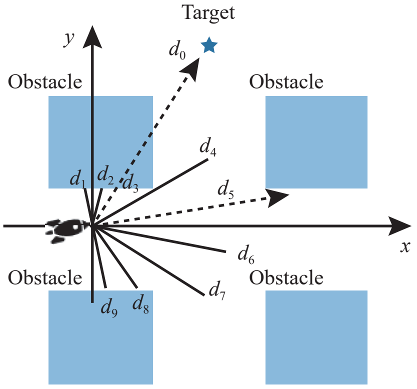


Fig. 4 Schematic diagram of obstacle avoidance

Inspired by obstacle avoidance, we use the distances between the missile and the edges of risky areas to extend the state space. These distances can be calculated by the geometrical relationship between the missile and the threats. The geometry is presented in Fig. 5, where M denotes the missile, V denotes the velocity. O and rad denote the threat center and the threat radius. MA denotes the distance between the missile and the edges of risky areas, which can be named as risky distance. The risky angle χ denotes the angle between velocity and risky distance. MB is tangent with the edge of risky areas at B, and ψ denotes the escape angle.

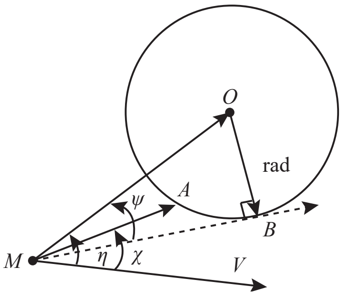


Fig. 5 Relative position of missile and threat

The risky distance d_{MA} can be calculated by

$$d_{MA} = \begin{cases} d_{MO}\cos(\eta-\chi) - \text{rad}\cdot \\ \qquad \cos\left(\arcsin\dfrac{d_{MO}\sin(\eta-\chi)}{\text{rad}}\right), & \psi \geqslant |\eta-\chi| \\ d_{\max}, \quad \psi < |\eta-\chi| \end{cases} \tag{14}$$

where $d_{\max}$ is the maximum of risky distance.

As the risky angle is selected, we can calculate the risky distance by (14). We select several risky angles within the range of [−π/2, π/2]. The extended state space can be defined as

$$S = \{d_0, q, \eta, \eta_T, d_1, d_2, \cdots, d_n\} \tag{15}$$

where d_0 is the distance between the missile and the target, d_i ($i \in [1, n]$) denotes the minimum risky-distance for different threats in risky angle χ_i.

We add the terminal condition and change (10) as

$$r_{\text{te}} = \begin{cases} k_1, & d_{0_t} \leqslant d_{\min} \\ k_2, & d_{i_t} \leqslant 0, i \in \{1, 2, \cdots, n\} \\ 0, & \text{otherwise} \end{cases}. \tag{16}$$

Using the extended state space and the changed reward function, we can improve the DDPG guidance algorithm, which could perform the guidance mission with threat avoidance.

3.3 HDDPG guidance

An improved hierarchical DDPG (HDDPG) algorithm is presented in this subsection to attenuate the acceleration chattering.

The proposed DDPG guidance algorithm could achieve expected performance in the task of target attacking and threat avoidance. However, due to the local optimum phenomenon caused by the tradeoff of hitting effect and threat avoidance, the serious chattering exists in the normal acceleration of the missile. The threat avoidance requires the missile to stay away from the threat center, while the hitting effect requires the missile to reach the target with the minimum miss distance. The priority varies with respect to the dynamic environment relevant to the hitting effect and threat avoidance. Some drastic transitions may result in harmful chartering.

To make the acceleration output smoothly, a penalty is presented to restrain the sudden variation of acceleration, which is given by

$$r_{\mathrm{re}} = -\exp(a_t^2) + 1. \tag{17}$$

Hence, the total reward function is changed as

$$r = \lambda_1 r_{\mathrm{tr}} + \lambda_2 r_{\mathrm{te}} + \lambda_3 r_g + \lambda_4 r_{\mathrm{re}}. \tag{18}$$

However, the training results is not stable enough with the new reward function (18). The main difficulty is that it is hard to decide the weight of λ_4. If the value of λ_4 is too small, the improvement is not obvious. When the value of λ_4 is too large, the guidance effect will be influenced.

To improve the training stability and weaken the acceleration chattering, we propose an HRL framework based on DDPG. The flow chart of HDDPG is shown in Fig. 6, where the memory and target networks are omitted for simplification.

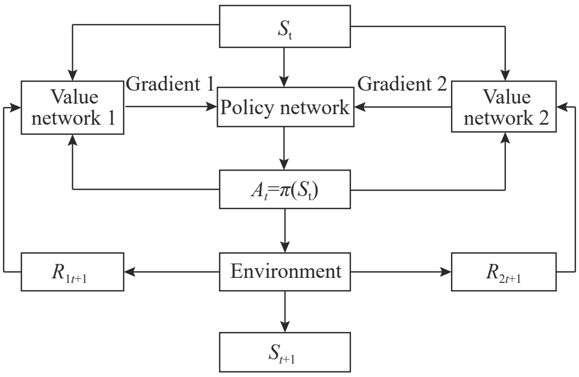


Fig. 6 Flow chart of HDDPG

There are two layers of DDPG in the presented framework, both share the same policy network. The reward function of the first DDPG is

$$r = \lambda_1 r_{\mathrm{tr}} + \lambda_2 r_{\mathrm{te}} + \lambda_3 r_g, \tag{19}$$

and the reward function of the second DDPG is

$$r = \lambda_2 r_{\mathrm{te}} + \lambda_4 r_{\mathrm{re}}. \tag{20}$$

Both two layers of DDPG update their value network according to their reward functions respectively. The policy network is shared, and it is trained in the framework of the first DDPG until the total reward (19) is convergent. The policy network learns target hitting and threat avoidance in this stage. To make the acceleration smooth, the training of policy network proceeds in the framework of the second DDPG until the total reward (20) is convergent. Algorithm 2 provides the explicit steps of the proposed HDDPG algorithm.

Algorithm 2 HDDPG

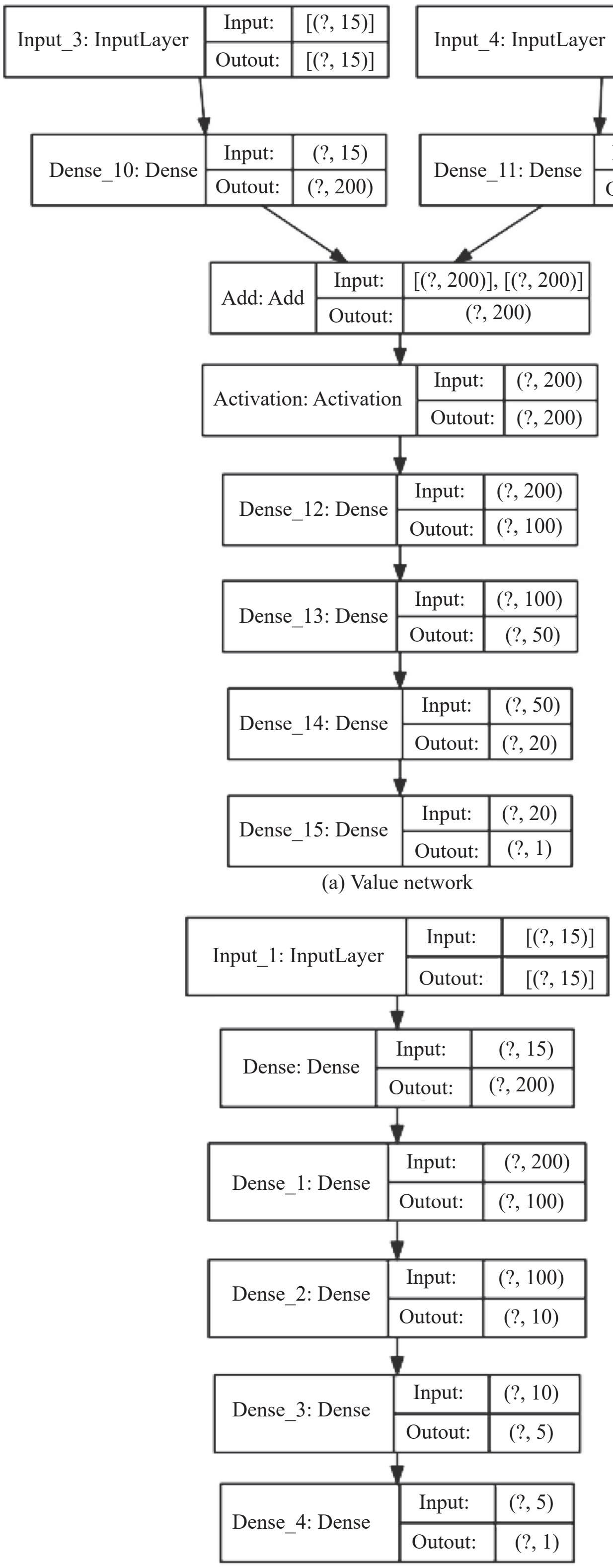
```
1. Initialize the policy network $\pi(s)$, the value network 1 $q_1(s,a)$and the value network 2 $q_2(s,a)$ with parameters $\theta$, $\omega_1$ and $\omega_2$; Initialize the target policy network $\pi'(s)$, the target value network 1 $q_1{}'(s,a)$, the target value network 2 $q_2{}'(s,a)$ with parameters $\theta' \leftarrow \theta$, $\omega_1{}' \leftarrow \omega_1$, $\omega_2{}' \leftarrow \omega_2$; Initialize the learning rate of target network $\varepsilon$, batch size $N$, memory $R$.
2. Take actions according to the state $A = \pi(S)$.
3. Execute the action $A$, receive the reward $R_1$, $R_2$ and acquire new state $S'$.
4. Save $\{S, A, R_1, R_2, S'\}$ to the memory.
5. Sample a batch size of $N$ data $\{(s,a,r_1,r_2,s')\}^N$ from memory randomly.
6. Update the policy network and the value network:
    $y_1 = r_1 + \gamma q_{1_\pi}(s', a'; \omega_1{}')_{a'=\pi(s';\theta')}$,
    $\omega_1 \leftarrow \omega_1 + \alpha[y_1 - q_{1_\pi}(s,a;\omega_1)]\nabla_{\omega_1} q_{1\pi}(s,a;\omega_1)$,
    $y_2 = r_2 + \gamma q_{2_\pi}(s', a'; \omega_2{}')_{a'=\pi(s';\theta')}$,
    $\omega_2 \leftarrow \omega_2 + \alpha[y_2 - q_{2_\pi}(s,a;\omega_2)]\nabla_{\omega_2} q_{2\pi}(s,a;\omega_2)$.
    If the guidance strategy is not convergen, update the policy network according to
    $\theta \leftarrow \theta + \beta\gamma^t \nabla_\theta \pi(s;\theta) \nabla_a q_1(s,a)$.
    If the guidance strategy is convergent, update the policy network according to
    $\theta \leftarrow \theta + \beta y^t \nabla_\theta \pi(s;\theta) \nabla_a q_2(s,a)$.
7. Update the target networks:
    $\omega_1{}' \leftarrow \varepsilon\omega_1 + (1-\varepsilon)\omega_1{}'$,
    $\omega_2{}' \leftarrow \varepsilon\omega_2 + (1-\varepsilon)\omega_2{}'$,
    $\theta' \leftarrow \varepsilon\theta + (1-\varepsilon)\theta'$.
8. Back to Step 2 until the iterations reach the maximum number of training.
```

4. Simulation

The simulation is carried out to validate the proposed algorithm in this section. We perform the simulation in various environments. The changes of index in the training process are given as well.

4.1 Simulation settings

For simplification, the value networks have a same network structure. The network structures of value network and policy network are shown in Fig. 7.



(a) Value network

(b) Policy network

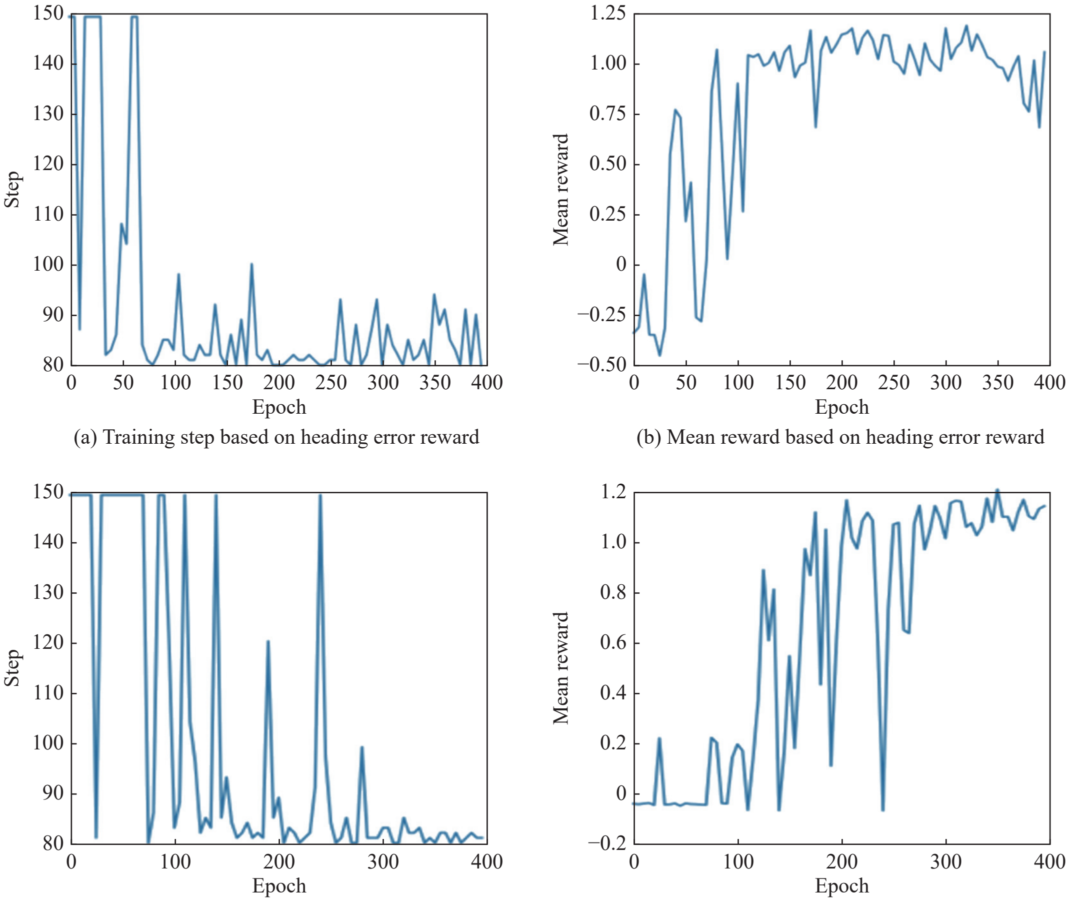
Fig. 7 Policy and value networks of HDDPG

We set the parameters as V = 600 m/s, V_T = 200 m/s, the sampling time-interval Δt = 0.1 s, the acceleration range of missile sets as [−40g, 40g] where g is the gravitational acceleration. The reward function is instantiated as: k_1=100, k_2=−10, $d_{\max}$=2 000 m, $d_{\min}$=50 m, γ = 0.99, ε = 0.01, λ_1=0.000 1, λ_2=1, λ_3=1 and λ_4=0.1. The capacity of memory is set to 3 000. The batch size is set to 64. The root mean square-prop optimizer is employed to learn the network parameters with a learning rate of 0.000 1 for policy network and a learning rate of 0.001 for value network. The exploration noise is set to Var(−0.1, 0.1). The number of train epochs in the first stage of HRL is n_1 = 400, and the number of train epochs in the second stage is n_2=200.

4.2 Simulation without threat avoidance

The effectiveness of the proposed guidance reward functions is compared. Since the index of training steps and rewards in each episode determine the effect of policy network, the training process can be reflected by the steps and the mean reward in each training epoch.

Fig. 8 and Fig. 9 show the training process of DDPG guidance and the corresponding simulation result without threats avoidance.



(a) Training step based on heading error reward

(b) Mean reward based on heading error reward

(c) Training step based on LOS rate reward

(d) Mean reward based on LOS rate reward

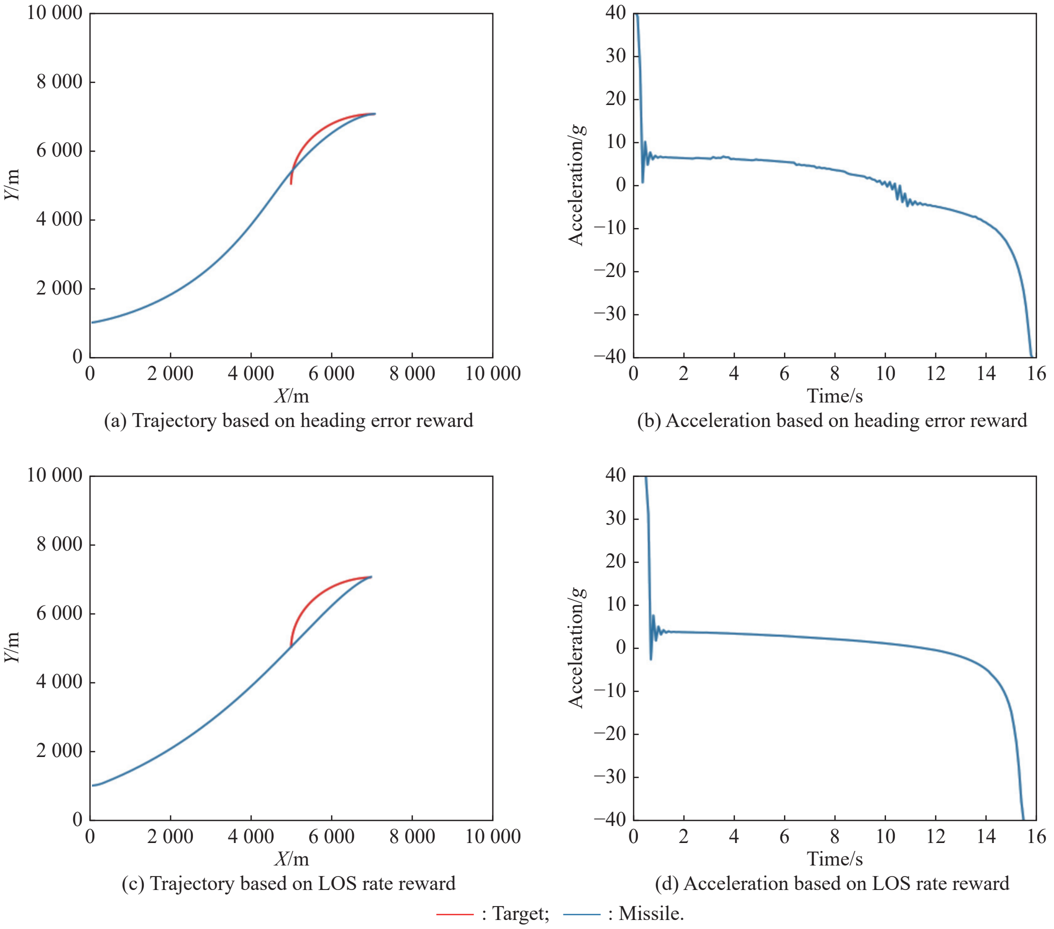
Fig. 8 Training process of DDPG guidance

The purpose of DRL is to improve the cumulative reward through continuous learning to obtain the maximum cumulative reward. From the index of reward in Fig. 8, we can find that the reward in the two methods both reach the high values. Steps in each epoch indicates weather the target is being hit, and the index of steps in Fig. 8 shows that the hit rate increases gradually.

By comparation in Fig. 9, we find the LOS rate method has the smoother track. The acceleration curve of the heading error method has higher volatility.

To illustrate the superiority of LOS rate reward further, we set the target at the coordinate of (8 000, 5 000) and test the algorithm by feeding 50 random predefined initial conditions of the missile. Fig. 10 shows the simulation trajectories for those conditions.

As is shown in the trajectories in Fig. 10, every missile hits the target and satisfies the requirement of the minimum miss distance. However, the result here implies that the LOS rate-based guidance algorithm has less chattering and better adaptability.



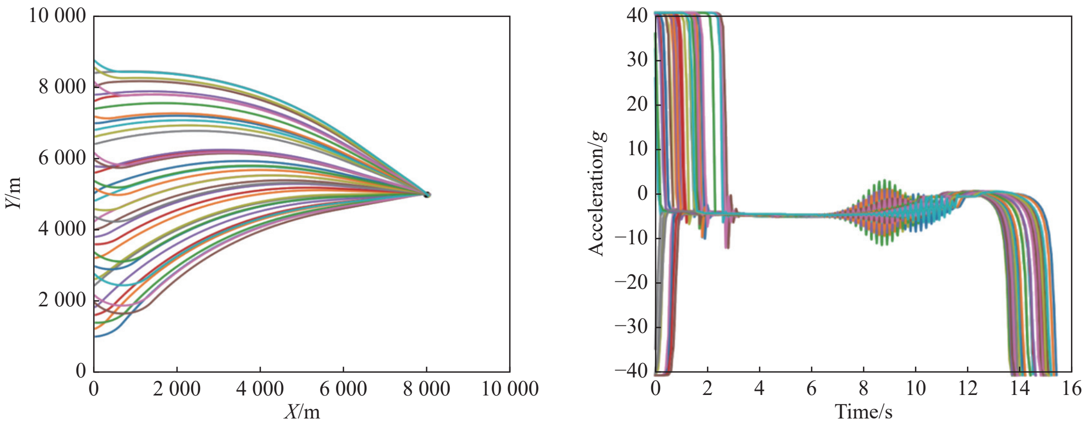
(a) Trajectory based on heading error reward

(b) Acceleration based on heading error reward

(c) Trajectory based on LOS rate reward

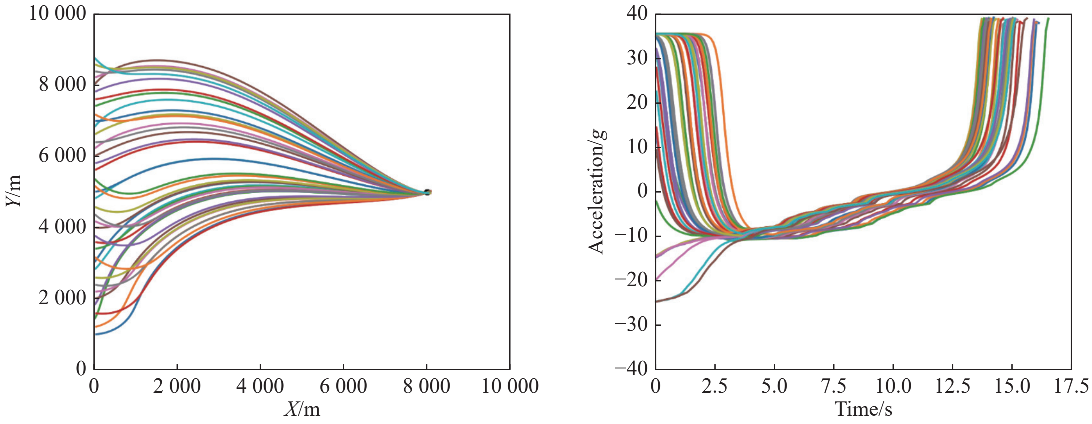
(d) Acceleration based on LOS rate reward

Fig. 9 DDPG guidance for maneuvering target



(a) Trajectory based on heading error reward

(b) Acceleration based on heading error reward



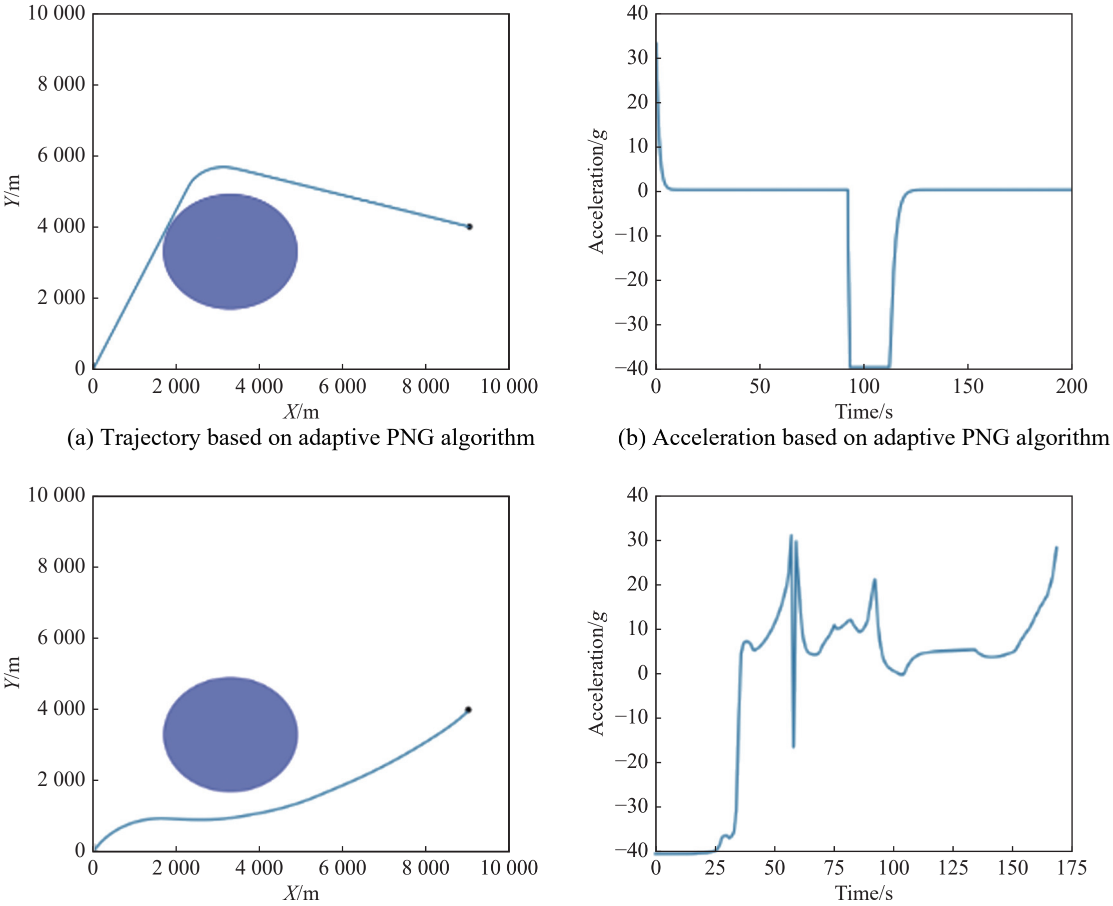
(c) Trajectory based on LOS rate reward

(d) Acceleration based on LOS rate reward

Fig. 10 DDPG guidance for stationary target

4.3 Simulation with threat avoidance

The effectiveness of presented DDPG guidance is demonstrated in this subsection. The DDPG guidance with the extended state space is compared with an adaptive PNG algorithm which accomplishes threat avoidance by setting an avoidance vector. Fig. 11 shows the simulation results of these two methods.



(a) Trajectory based on adaptive PNG algorithm

(b) Acceleration based on adaptive PNG algorithm

(c) Trajectory based on DDPG guidance

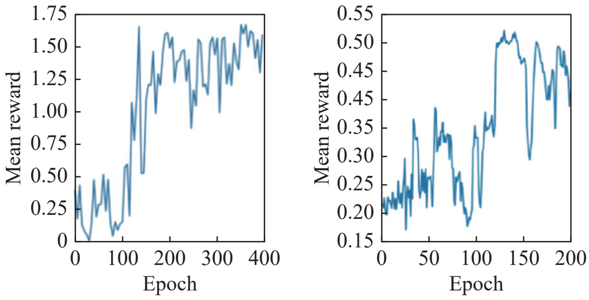
(d) Acceleration based on DDPG guidance

Fig. 11 DDPG guidance for threat avoidance

The adaptive PNG algorithm performs the target attacking task after the threat avoidance. The information of relative velocity and position between missile-target is not used during the procedure of threat avoidance. In contrast, the proposed DDPG guidance algorithm considering much more global information to enhance decision making. The acceleration of the missile is obtained by virtue of the policy network and the state space, which includes the information of both the target and the obstacle threat. Our DDPG guidance can perform the attacking and threat avoidance tasks synchronously. As revealed from Fig. 11(b), a more efficient and safer trajectory is generated for the missile to reach the target.

4.4 Simulation for HDDPG

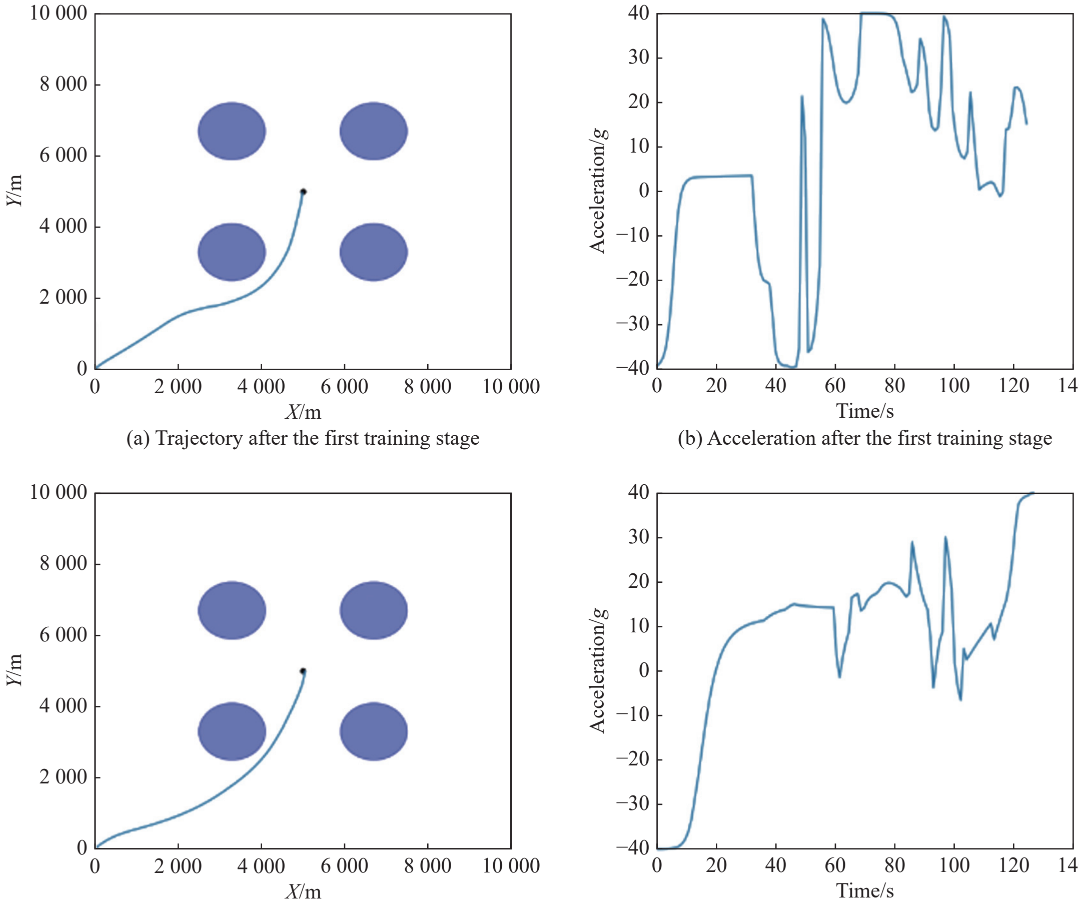
In this subsection, the effectiveness of the HDDPG guidance is verified. Fig. 12 and Fig. 13 show the training process of HDDPG guidance and the corresponding simulation result in threat existing environment.



(a) The first training stage of HDDPG

(b) The second training stage of HDDPG

Fig. 12 Training process of HDDPG



(a) Trajectory after the first training stage

(b) Acceleration after the first training stage

(c) Trajectory after the second training stage

(d) Acceleration after the second training stage

Fig. 13 HDDPG guidance

We can find from Fig. 12 that due to the extended state space and the proposed reward functions, those two stages in HDDPG both have a stable training process. Fig. 12(a) indicates that our algorithm learns target attacking and threat avoidance in the training prosses, gradually. However, Fig. 13(a) shows that the threat avoidance function leads to serious chattering in acceleration. In contrast, Fig. 13(b) reveals that the proposed HDDPG algorithm can attenuate the chattering and make the acceleration smoother due to the action penalty in the second stage of HDDPG.

5. Conclusions

In this paper, we propose a DDPG based HRL algorithm for missile guidance in consideration of threat avoidance. A more stable training result is obtained by the improved LOS reward function. The threat avoidance is achieved with the aid of extending the state space. A HDDPG framework is presented to attenuate the chattering in acceleration. The simulation results validate the effectiveness of the proposed algorithm. It is worth noting that some extensions can be done in future works. For instance, the smaller the missile-target distance is, the higher requirement of control accuracy becomes. DRL has a strong ability of independent decision-making but is weak in precise control, which may lead to large acceleration output at the attack instant. An alternative way to settle this problem is improved by combination of the general guidance law method.

References

[1] JI Y, LIN D F, WANG W, et al. Three-dimensional terminal angle constrained robust guidance law with autopilot lag consideration. Aerospace Science and Technology, 2019, 86: 160–176.

[2] RYOO C K, CHO H, TAHK M J. Time-to-go weighted optimal guidance with impact angle constraints. IEEE Trans. on Control Systems Technology, 2006, 14(3): 483–492.

[3] JEON I S, LEE J I, TAHK M J. Impact-time-control guidance law for anti-ship missiles. IEEE Trans. on Control Systems Technology, 2006, 14(2): 260–266.

[4] DONG Y E, SHI M M, SUN Z W. Satellite proximate interception vector guidance based on differential games. Chinese Journal of Aeronautics, 2018, 31(6): 1352–1361.

[5] MARCHIDAN A, BAKOLAS E. Collision avoidance for an unmanned aerial vehicle in the presence of static and moving obstacles. Journal of Guidance, Control, and Dynamics, 2020, 43(1): 96–110.

[6] XU X G, WEI Z Y, REN Z, et al. Time-varying fault-tolerant formation tracking based cooperative control and guidance for multiple cruise missile systems under actuator failures and directed topologies. Journal of Systems Engineering and Electronics, 2019, 30(3): 587–600.

[7] DARSHAN D, ARCHANA C, DEBAJYOTI M. Artificial intelligence based missile guidance system. Proc. of the 7th International Conference on Signal Processing and Integrated Networks, 2020: 873–878.

[8] JIE Z, LI H D, BIN X. A joint mid-course and terminal course cooperative guidance law for multi-missile salvo attack. Chinese Journal of Aeronautics, 2018, 31(6): 1311–1326.

[9] WANG P, ZHANG X B, ZHU J H. Integrated missile guidance and control: a novel explicit reference governor using a disturbance observer. IEEE Trans. on Industrial Electronics, 2018, 66(7): 5487–5496.

[10] FU S N, LIU X D, ZHANG W J, et al. Multiconstraint adaptive three-dimensional guidance law using convex optimization. Journal of Systems Engineering and Electronics, 2020, 31(4): 791–803.

[11] FANG M, GROEN F C A. Collaborative multi-agent reinforcement learning based on experience propagation. Journal of Systems Engineering and Electronics, 2013, 24(4): 683–689.

[12] SHALUMOV V. Cooperative online guide-launch-guide policy in a target-missile-defender engagement using deep reinforcement learning. Aerospace Science and Technology, 2020, 104: 105996.

[13] YOU S X, DIAO M, GAO L P, et al. Target tracking strategy using deep deterministic policy gradient. Applied Soft Computing, 2020, 95: 106490.

[14] GAUDET B, LINARES R, FURFARO R. Deep reinforcement learning for six degree-of-freedom planetary landing. Advances in Space Research, 2020, 65(7): 1723–1741.

[15] LI Y, QIU X H, LIU X D, et al. Deep reinforcement learning and its application in autonomous fitting optimization for attack areas of UCAVs. Journal of Systems Engineering and Electronics, 2020, 31(4): 734–742.

[16] YAN C, XIANG X J, WANG C. Towards real-time path planning through deep reinforcement learning for a UAV in dynamic environments. Journal of Intelligent & Robotic Systems, 2020, 98(2): 297–309.

[17] WANG D W, FAN T X, HAN T, et al. A two-stage reinforcement learning approach for multi-UAV collision avoidance under imperfect sensing. IEEE Robotics and Automation Letters, 2020, 5(2): 3098–3105.

[18] YUE W, GUAN X H, WANG L Y. A novel searching method using reinforcement learning scheme for multi-UAVs in unknown environments. Applied Sciences, 2019, 9(22): 4964.

[19] LI G F, WU Y, XU P. Adaptive fault-tolerant cooperative guidance law for simultaneous arrival. Aerospace Science and Technology, 2018, 82: 243–251.

[20] LI G F, WU Y, XU P. Fixed-time cooperative guidance law with input delay for simultaneous arrival. International Journal of Control, 2021, 94(6): 1664–1673.

[21] GAUDET B, LINARES R, FURFARO R. Adaptive guidance and integrated navigation with reinforcement meta-learning. Acta Astronautica, 2020, 169: 180–190.

[22] LIANG C, WANG W H, LIU Z H, et al. Range-aware impact angle guidance law with deep reinforcement meta-learning. IEEE Access, 2020, 8: 152093–152104.

[23] HU Q L, HAN T, XIN M. Sliding-mode impact time guidance law design for various target motions. Journal of Guidance, Control, and Dynamics, 2019, 42(1): 136–148.

[24] ZHANG W J, FU S N, LI W, et al. An impact angle constraint integral sliding mode guidance law for maneuvering targets interception. Journal of Systems Engineering and Electronics, 2020, 31(1): 168–184.

[25] LI G F, LI Q, WU Y J, et al. Leader-following cooperative guidance law with specified impact time. Science China: Technological Sciences, 2020, 63(11): 2349–2356.

[26] ZHANG W, SONG K, RONG X W, et al. Coarse-to-fine UAV target tracking with deep reinforcement learning. IEEE Trans. on Automation Science and Engineering, 2018, 16(4): 1522–1530.

[27] QIE H, SHI D X, SHEN T L, et al. Joint optimization of multi-UAV target assignment and path planning based on multi-agent reinforcement learning. IEEE Access, 2019, 7: 146264–146272.

[28] HONG D, KIM M, PARK S. Study on reinforcement learning-based missile guidance law. Applied Sciences, 2020, 10(18): 6567.

[29] CHENG L, LU H, LEI T, et al. Path planning for anti-ship missile using tangent based dubins path. Proc. of the 2nd International Conference on Intelligent Autonomous Systems, 2019: 175–180.

[30] GUO H, FU W X, FU B, et al. Smart homing guidance stra-

tegy with control saturation against a cooperative target-defender team. Journal of Systems Engineering and Electronics, 2019, 30(2): 366–383.

[31] YU W B, CHEN W C. Guidance law with circular no-fly zone constraint. Nonlinear Dynamics, 2014, 78(3): 1953–1971.

[32] WEISS M, SHIMA T. Linear quadratic optimal control-based missile guidance law with obstacle avoidance. IEEE Trans. on Aerospace and Electronic Systems, 2018, 55(1): 205–214.

[33] FAN S P, QI Q, LU K F, et al. Autonomous collision avoidance technique of cruise missiles based on modified artificial potential method. Transaction of Beijing Institute of Technology, 2018, 38(8): 828–834.

[34] CHAYSRI P, BLEKAS K, VLACHOS K. Multiple mini-robots navigation using a collaborative multiagent reinforcement learning framework. Advanced Robotics, 2020, 34(13): 902–916.

[35] WANG C, WANG J, SHEN Y, et al. Autonomous navigation of UAVs in large-scale complex environments: a deep reinforcement learning approach. IEEE Trans. on Vehicular Technology, 2019, 68(3): 2124–2136.

[36] LI B H, WU Y. Path planning for UAV ground target tracking via deep reinforcement learning. IEEE Access, 2020, 8: 29064–29074.

Biographies

LI Bohao was born in 1990. He received his B.E. degree from Lanzhou University, Lanzhou, China, in 2012, and M.S. degree in Lanzhou University of Technology, Lanzhou, China, in 2017. He is currently pursuing his Ph.D. degree in navigation, guidance and control with Beihang University, Beijing, China. His research interests include deep learning, deep reinforcement learning, and guidance.

E-mail: libh08@buaa.edu.cn

WU Yunjie was born in 1969. She received her Ph.D. degree in navigation guidance and control from Beihang University in 2006. Now, she is a professor in the School of Automation Science and Electrical Engineering, Beihang University, Beijing, China. Her research interests include system simulation, intelligent control, servo control, aircraft guidance and control technology.

E-mail: wyjmip@ buaa.edu.cn

LI Guofei was born in 1991. He received his Ph.D. degree from the School of Automation Science and Electrical Engineering, Beihang University, Beijing, China, in 2020. From 2020 to 2021, he was a postdoctoral fellow of Zhuoyue Program in the School of Cyber Science and Technology, Beihang University, Beijing, China. Now, he is an associate professor in the School of Astronautics, Northwestern Polytechnical University, Xi’an, China. His research interests include cooperative guidance, servo system control, and nonlinear control.

E-mail: liguofei1@126.com