Hindawi
International Journal of Aerospace Engineering
Volume 2022, Article ID 2021693, 15 pages
https://doi.org/10.1155/2022/2021693

Research Article

A Deep Q-Network-Based Collaborative Control Research for Smart Ammunition Formation

Jian Shen,[1] **Benkang Zhang**,[1] **Qingyu Zhu**,[2] **and Pengyun Chen**[1]

[1]*College of Mechatronics Engineering, North University of China, Taiyuan 030051, China*
[2]*AVIC China Aero-Polytechnology Establishment, Beijing 100028, China*

Correspondence should be addressed to Benkang Zhang; sz202101054@st.nuc.edu.cn

Received 16 February 2022; Accepted 9 June 2022; Published 22 June 2022

Academic Editor: Zhiguang Song

The smart ammunition formation (SAF) system model usually has the characteristics of complexity, time variation, and nonlinearity. With the consideration of random factors, such as sensor error and environmental disturbance, the system model cannot be modeled accurately. To deal with this problem, this paper investigated an intelligent deep Q-network- (DQN-) based control algorithm for the SAF collaborative control, which deals with the high dynamics and uncertainty in the SAF flight environment. In the environment description of the SAF, we built a dynamic model to represent the system joint states, which referred to the smart ammunition's velocity, the trajectory inclination angle, the ballistic deflection angle, and the relative position between different formation nodes. Next, we describe the SAF collaborative control process as a Markov decision process (MDP) with the application of the reinforcement learning (RL) technique. Then, the basic framework ε-imitation action-selecting strategy and the algorithm details were developed to address the SAF control problem based on the DQN scheme. Finally, the numerical simulation was carried out to verify the effectiveness and portability of the DQN-based algorithm. The average total reward curve showed a reasonable convergence, and the relative kinematic relationship among the formation nodes met the requirements of the controller design. It illustrated that the DQN-based algorithm obtained a novel performance in the SAF collaborative control.

1. Introduction

SAF is an important piece of equipment to realize the collaborative combat framework of the network center. Since the U.S. military put forward the concept of network-centric operations (NCO), unmanned aerial vehicles (UAVs) have been one of the essential links in future collaborative warfare [1]. Unmanned combat aerial vehicles include UAVs, unmanned airships, hypersonic vehicles, smart ammunition, and guided munitions. They mainly perform battlefield information detection, military mapping, damage assessment, and collaborative operations. Nowadays, the military-strategic defense technology of various countries is advancing by leaps and bounds, and the multilevel, all-around, and three-dimensional air defense and antismart ammunition systems make the penetration capability and attack effect of guided weapons drop sharply [2].

At present, the research on collaborative guidance and control technology of the SAF is in a sped-up development stage [3]. Similar UAV formation collaborative control technology has achieved many results after years of research. However, due to the particularity of the SAF, the relevant research results on UAV formation cannot be directly applied to the collaborative control of the SAF [4]. SAF is an important embodiment of the militarization application of multiagent systems. Compared with UAVs and other agents, SAF has higher movement speed, which makes the collaborative control method of multiple SAF require higher real-time performance and less formation communication. In addition, it is difficult for SAF to realize circling and keep stationary in comparison with UAVs [5].

Because of the significance of engineering implementation, there is much research on the collaborative control of the SAF. Zeng et al. [6] studied the collaborative guidance

of multi-unit smart ammunition attacking fixed targets. The guidance strategy combining (PNG) guidance law with state error feedback and acceleration command was adopted to realize the simultaneous attack of the smart ammunition group on the target. And simulation verified the effectiveness of the guidance and control algorithm. Wang and Wu [7] proposed a new distributed collaborative guidance law, which realized the consistency of attack angle and time of the leader-follower multiple smart ammunition formation and ensured zero miss distance. Compared with the existing research results, the guidance law proposed in this paper had fewer restrictions and was easier to implement in engineering. Song et al. [8] studied the simultaneous attack of muili-unit smart ammunition under the condition of communication uncertainty composed of a random network and additional noise. A robust control framework composed of a cyclic fading network and the collaborative algorithm was proposed, and simulation verified the multiple smart ammunition collaborative control algorithm under the condition of communication uncertainty. Wu et al. [9] developed a missile formation algorithm and deduced the time-varying formation constraints in three-dimensional space. The formation control under time-varying position constraints was transformed into a constrained optimization problem. The three-degrees-of-freedom simulation results of the missiles obtained by constraint optimization showed that the formation strategy proposed was feasible in missile formation control under complex time-varying constraints. Saar and Noa [10] illustrated that the formation control approaching selection was affected by the research field, the formation control coordination scheme, the sensing capabilities, and the information assumption. In this literature, it showed that different approach combinations would provide the researcher with suitable knowledge regarding both the benefits and deficiencies. Kun et al. [11] divided the collaborative control and communication methods of intelligent swarms into the following four categories: task assignment-based methods, bioinspired methods, distributed sensor fusion, and reinforcement learning-based methods. Based on the basic ideas and introduction of different methods, the future development stages of intelligent group cooperative control and communication are forecasted, and the problems and challenges are put forward. Mohamed et al. [12] presented a summary of the main applications related to aerial swarm systems and the associated research works. They introduced a proposed abstraction of an aerial swarm system architecture to help developers to understand the main required modules. Zhang et al. [13] investigated a novel cooperative control system based on the sliding mode variable structure control theory for multimissile formation flight. According to numerical simulations, the method proposed in this article could achieve similar relative position errors under the condition of uncontrollable speed. And the robustness, versatility, and formation adaptability of the method were confirmed by simulation results.

For the collaborative control of the SAF, each type of ammunition belongs to a nonlinear system, and the flight attitude control of the SAF needs to be realized through the nonlinear control method [14]. However, in practical applications, the characteristics of the SAF cannot be accurately described by the modeling methods. The accurate aircraft system model usually has complexity, time variation, and nonlinearity. Random factors such as sensor error and environmental disturbance often make it difficult to model accurately. This seriously limits the application of traditional control methods. As an alternative method, the application of the model-free reinforcement learning method to solve the above contradiction has attracted increasing attention [15]. The control design method based on deep reinforcement learning (DRL) technology can realize the collaborative control of the SAF without relying on an accurate system model. The DRL-based collaborative control has been adopted increasingly in multiagent areas, such as UAVs, multirobot systems, unmanned surface vessels (USVs), and satellite formation. Du et al. [16] proposed a multiagent reinforcement learning- (MARL-) based approach to solving the collaborative pursuit problem for UAVs. This approach enabled the pursuer UAVs to capture unauthorized UAVs more quickly in urban airspace under poor communication conditions. Through the designed learning process, the pursuers could ultimately learn effective pursuit strategy and collision resolution strategy in the meantime. Extensive experimental results showed the superiority of the proposed method in terms of higher capturing probability. Wang et al. [17] presented the graph reinforcement learning multiagent formation control model with obstacle avoidance under restricted communication. The authors used the characteristics of graph, attention, and multiple long-short-term memories to promote cooperation behavior. The model was shown to perform a satisfying strategy under dynamic obstacles. Sui et al. [18] proposed a new method based on DRL to solve the problem of formation control with collision avoidance. The learning-based policy was extended to the field of formation control, which involved a two-stage training framework: imitation learning and reinforcement learning. Many representative simulations were conducted, and it deployed the method on an experimental platform. It validated the effectiveness and practicability of the proposed method through both the simulation and experiment results. Ma et al. [19] investigated the target encirclement control problem of multirobot systems via DRL. The method mentioned in this literature provided a distributed control architecture for each robot in continuous action space. The behavioral output at each time step was determined by its independent network. Robots and the moving target could be trained simultaneously. The calculation results validated the effectiveness of the proposed algorithm. Jan et al. [20] developed a multi-UAV fleet control system based on the DRL algorithm. A deep convolutional neural network with a linear output layer was chosen as a control policy and trained for aerial surveillance and base defense with five UAVs. The control policy performed well in the real drone flight test. Zhao et al. [21] addressed the problem of path following for underactuated USV formation via a changed DRL with random braking. With the aid of DRL, the proposed system could adjust the formation automatically and flexibly. Simulation verified the effectiveness and superiority of the mentioned formation and path-following control strategies. Smith et al. [22] created a framework for solving highly nonlinear satellite formation control problems by using model-free policy optimization DRL methods. The proposed DRL framework could solve complex satellite formation flying problems and provide

key insights into achieving optimal and robust formation control using reinforcement learning. Wang et al. [23] proposed a distributed DRL algorithm for USV formation. This algorithm could enhance the adaptability and extendibility of the formations to increase the number of USVs or change formation shape arbitrarily. The effectiveness of the algorithms has been verified and validated through several computer-based simulations.

Considering the modeling features in the SAF collaborative control issue, a DRL-based algorithm will be the ideal choice for dealing with the high dynamics and uncertainty in its flight environment. Aiming at the characteristics of the SAF flight environment, a safe cooperative control DRL-based algorithm, DQN, is studied in this work. First, a SAF environment is illustrated in Section 2. The dynamic model represents the system joint state of intelligent ammunition, which relates to the leader and followers' velocity, the trajectory inclination, the ballistic deflection angle, and the relative position in the SAF. Second, the SAF collaborative control process is described as an MDP model, and the state space, action space, and reward function will be discussed in this part. Third, the basic framework of the SAF control problem based on DQN will be discussed in Section 3. The action-selecting strategy ε-imitation, Q-network construction, and algorithm details are proposed. Finally, several numerical simulation tests are conducted to verify the convergence and portability of the DQN-based algorithm proposed in this paper.

2. Environment and MDP Framework

2.1. Environment of SAF

2.1.1. Motion Equation of Smart Ammunition. Based on the assumption of instantaneous balance and without considering the rotation of the projectile around the center of mass, we establish the motion equations of the center of mass of the smart ammunition in the form shown in the following equation [24].

$$\begin{cases} m\dfrac{dV}{dt} = P\cos\alpha_b\cos\beta_b - X_b - mg\sin\theta, \\ mV\dfrac{d\theta}{dt} = P(\sin\alpha_b\cos\gamma_V + \cos\alpha_b\sin\beta_b\sin\gamma_V) + Y_b\cos\gamma_V - Z_b\sin\gamma_V - mg\cos\theta, \\ -mV\cos\theta\dfrac{d\psi_V}{dt} = P(\sin\alpha_b\sin\gamma_V - \cos\alpha_b\sin\beta_b\cos\gamma_V) + Y_b\sin\gamma_V + Z_b\cos\gamma_V, \\ \dfrac{dx}{dt} = V\cos\theta\cos\psi_V, \\ \dfrac{dy}{dt} = V\sin\theta, \\ \dfrac{dz}{dt} = -V\cos\theta\sin\psi_V, \\ \dfrac{dm}{dt} = -m_t, \\ \alpha_b = -\dfrac{m_z^{\delta z}}{m_z^{\alpha}}\delta_{zb}, \\ \beta_b = -\dfrac{m_y^{\delta y}}{m_y^{\beta}}\delta_{yb}, \end{cases} \tag{1}$$

where m is the mass of the ammunition; m_t is the reduction of propellant mass per unit time; V is the velocity of the center of mass; P is the thrust; α_b is the balanced angle of attack; β_b is the balanced sideslip angle; δ_{zb} and δ_{yb} are the balanced rudder deflection angles; γ_V is the rolling angle; and X_b, Y_b, and Z_b are the balanced resistance, lift, and lateral force, respectively. g is the gravity acceleration, θ is the trajectory inclination angle, ψ_V is the ballistic deflection angles, and $m_z^{\delta z}$ and m_z^{α} are the derivatives of the pitch moment coefficient regarding to δ_z and α, respectively. $m_y^{\delta y}$ and m_y^{β} are the derivatives of the yaw moment coefficient with respect to δ_y and β, respectively.

2.1.2. SAF System. SAF adopts the leader-follower structure, with one leader and several followers forming a formation unit. In the control design requirements of the formation, even if the leader maneuvers, the follower must keep a

relatively safe distance from the leader and limit the chances of parameters such as the trajectory inclination angle, the ballistic deflection angle, and the velocity of the follower to a small range, thus ensuring the approximate stability of the formation [25].

The leader-follower control method is an important technical approach to studying the consistency of system collaborative control [26]. The leader is a special individual. As the leader of the formation, it is not affected by other individuals and guides the movement track of the formation. However, the follower does not need to sense the target information of the formation but only gets the information from the leader [27]. Consequently, the disadvantage of the leader-follower control method is that the leader and the follower are relatively independent, and it is difficult to get the tracking error feedback from the follower.

To describe the relative position of the leader-follower SAF configuration, a coordinate system with the follower projectile as a reference is established, as shown in Figure 1 [28].

In Figure 1, X_L, Y_L, and Z_L are the coordinates of the inertial system of the leader projectile. X_F, Y_F, and Z_F are the coordinates of the inertial system of the follower projectile. x_F, y_F, and z_F are the relative distances between the leader and the follower projectile in the velocity coordinate system referenced by the follower. V_L and V_F are the speed of the leader and the follower, respectively. θ_L and θ_F are the trajectory inclination angles of the leader and the follower, respectively. ψ_{VL} and ψ_{VF} are the ballistic deflection angles of the leader and the follower, respectively.

It is assumed that the control parameters, such as the velocity, the trajectory inclination angle, and the ballistic deflection angle of the leader, can be measured. A simplified model is adopted in the formation control problem. Considering that the motion of the formation is a first-order system, the motions of the leader and the follower are independent of each other [29]. The leader's motion equations are shown as

$$\begin{cases} \dfrac{dV_L}{dt} = \dfrac{1}{\tau_{VL}}(V_{LC} - V_L) + \eta_V, \\ \dfrac{d\theta_L}{dt} = \dfrac{1}{\tau_{\theta L}}(\theta_{LC} - \theta_L) + \eta_\theta, \\ \dfrac{d\psi_{VL}}{dt} = \dfrac{1}{\tau_{\psi VL}}(\psi_{VLC} - \psi_{VL}) + \eta_{\psi_V}, \\ \dfrac{dX_L}{dt} = V_L \cos\theta_L \cos\psi_{VL} + \eta_x, \\ \dfrac{dY_L}{dt} = V_L \sin\theta_L + \eta_y, \\ \dfrac{dZ_L}{dt} = -V_L \cos\theta_L \sin\psi_{VL} + \eta_z. \end{cases} \tag{2}$$

Like the leader's kinematic model in Equation (2), the follower's model can be shown as

$$\begin{cases} \dfrac{dV_F}{dt} = \dfrac{1}{\tau_{VF}}(V_{FC} - V_F) + \eta_V, \\ \dfrac{d\theta_F}{dt} = \dfrac{1}{\tau_{\theta F}}(\theta_{FC} - \theta_F) + \eta_\theta, \\ \dfrac{d\psi_{VF}}{dt} = \dfrac{1}{\tau_{\psi VF}}(\psi_{VFC} - \psi_{VF}) + \eta_{\psi_V}, \\ \dfrac{dX_F}{dt} = V_F \cos\theta_F \cos\psi_{VF} + \eta_x, \\ \dfrac{dY_F}{dt} = V_F \sin\theta_F + \eta_y, \\ \dfrac{dZ_F}{dt} = -V_F \cos\theta_F \sin\psi_{VF} + \eta_z, \end{cases} \tag{3}$$

where τ_{VL}, τ_{VF}, $\tau_{\theta L}$, $\tau_{\theta F}$, $\tau_{\psi VL}$, and $\tau_{\psi VF}$ are the time-related constants which are related to the, velocity, trajectory inclination angle, and deflection angle of the leader and the follower. V_{LC} and V_{FC} are the speed instructions for the leader and the follower, respectively. θ_{LC} and θ_{FC} are the trajectory inclination commands of the leader and the follower, respectively. ψ_{VLC} and ψ_{VFC} are the ballistic deflection angle commands of the leader and the follower, respectively. η_V, η_θ, η_{ψ_V}, η_x, η_y, and η_z are the disturbances for every state variable. They all obey the normal distribution, and the mean values and variance are shown in Table 1.

According to Equations (2) and (3), the SAF system states, which show the relative relationship between the leader and the follower, can be shown as [30]

$$\begin{cases} S_1 = \theta_F - \theta_L, \\ S_2 = \psi_{VF} - \psi_{VL}, \\ \begin{bmatrix} S_3 \\ S_4 \\ S_5 \end{bmatrix} = \begin{bmatrix} X_F - X_L \\ Y_F - Y_L \\ Z_F - Z_L \end{bmatrix} = R(\theta, \psi_V) \begin{bmatrix} x_F \\ y_F \\ z_F \end{bmatrix}. \end{cases} \tag{4}$$

The conversion matrix in (4) is

$$R(\theta, \psi_V) = \begin{bmatrix} \cos\theta_F \cos\psi_{VF} & -\sin\theta_F & \cos\theta_F \sin\psi_{VF} \\ \sin\theta_F \cos\psi_{VF} & \cos\theta_F & \sin\theta_F \sin\psi_{VF} \\ -\sin\psi_{VF} & 0 & \cos\psi_{VF} \end{bmatrix}, \tag{5}$$

where S_1, S_2 are the differences between the trajectory inclination angle and the ballistic deflection angle between the leader and the follower, respectively. S_3, S_4, S_5 are the differences of the relative position in the x, y, z direction, respectively, between the leader and the follower. During the real flight, the control command of the leader will be adjusted according to the battlefield situation. To make the model more suitable for the dynamic input uncertainties, the control commands will be constant or generated randomly by user functions, as will be shown in Section 4.

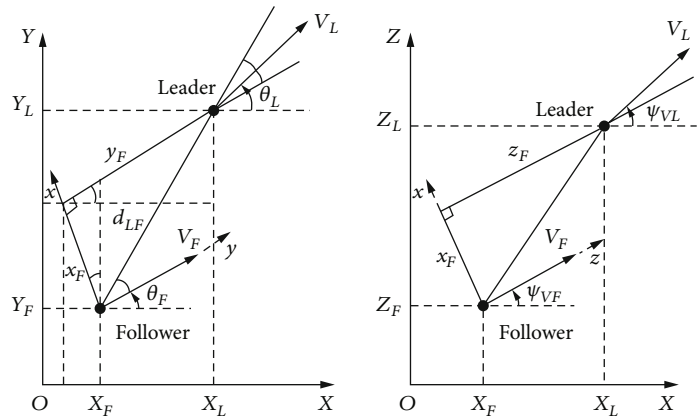

Figure 1: Relation between the leader and the follower ammunition in the inertial coordinate system.

Table 1: Parameters of the leader and follower ammunition.

	Leader	Follower 1	Follower 2
Initial position (m)	(0, 8000, 0)	(-500, 8500, 0)	(-1000, 9000, 0)
V_0 (m/s)	200	200	200
θ_0 (deg)	0.0	5.0	-5.0
ψ_{V_0} (deg)	0.0	0.0	0.0

2.2. MDP Model for the SAF Collaborative Control. From what has been discussed above, it can be found that the control problem for the SAF is essentially a multiple step decision-making problem, of which the core is to choose the proper control commands of the ammunition velocity, the trajectory inclination, and the ballistic deflection angle and timing to implement and release the sequential decision. This paper proposes an intelligent and efficient control method to cope with the control problem for the SAF collaborative control. The projectile operation is redefined to be an MDP framework. The basic MDP framework is shown in Figure 2.

The discrete MDP can be represented by a quintuple array $\{S, A, R, P, J\}$. S is the state space divided according to the relative position and attitude of the leader and the follower, and A is the action space composed of the control instructions of the follower's velocity, the angles of trajectory inclination, and the ballistic deflection. R is the return of corresponding states and actions. R is the transition probability between states, and J is the optimization aim function of the control decision. Discrete MDP has the following characteristics:

$$\begin{aligned} &p\left(s_{t+1} = s_j | s_t = s_i, a_t = a_k, s_{t-1}, a_{t-1}, \cdots, s_0, a_0\right) \\ &\quad = p\left(s_{t+1} = s_j | s_t = s_i, a_t = a_k\right) = p_{ij}(a_k) \quad \forall s_i, s_j \in S, a_k \in A, \forall t \geq 0, \end{aligned} \tag{6}$$

where $p_{ij}(a_k)$ is the probability that the state s_i will transition to s_j when the action a_k is taken in the state.

In the discrete MDP model, the range and accuracy of discrete parameters in state space S will directly affect the learning effect of the formation controller. According to the factors to be considered in the battle process of the SAF, the state space parameters of the formation MDP model to be selected include the relative position and angles between the leader and the follower. The other four parameters A, R, P, J of the MDP model are mainly constructed according to the task target. Action space A contains the action, such as the velocity, the angles of trajectory inclination, and the ballistic deflection. The reward function R is constructed by the safe distance values between the real-time positions of different formation members. The transition probability P depends on the actual ballistic position of the smart ammunition after the action is executed. The aim function J is set to the total return value. Set as the action selection strategy, J^* is the optimal return value, including

$$J* = \max_{\pi} J = \max_{\pi} E\left(\sum_{t=0}^{\infty} \gamma^t r_t\right), \tag{7}$$

where $\gamma \in (0, 1)$ is the return discount factor and r_t is the return value at time t.

2.2.1. State Space. The state of the smart ammunition system can be represented by a multidimensional array. In the formation collaborative control problem under the leader-follower topology, the relative relationship between the leader and the follower (such as distance and heading difference) has a crucial impact on the formulation of the control strategy. The system state can represent the state space to characterize the relative spatial position and pose relationship between the leader and the follower ammunition. The control command of the leader is determined by the flight control system according to the relative position relationship between the leader and the follower in practical engineering applications. In this work, the formation of the collaborative control architecture is the main content, so the control instructions of guided smart ammunition are simplified as Equation (4). To make the model get the adaptability of various inputs, the random function is used to generate the leader's control command in the DQN training process to simulate the uncertainty of system input. According to Equation (4), the state space of the SAF's MDP scheme can be defined as $S = \{S_1, S_2, S_3, S_4, S_5\}$.

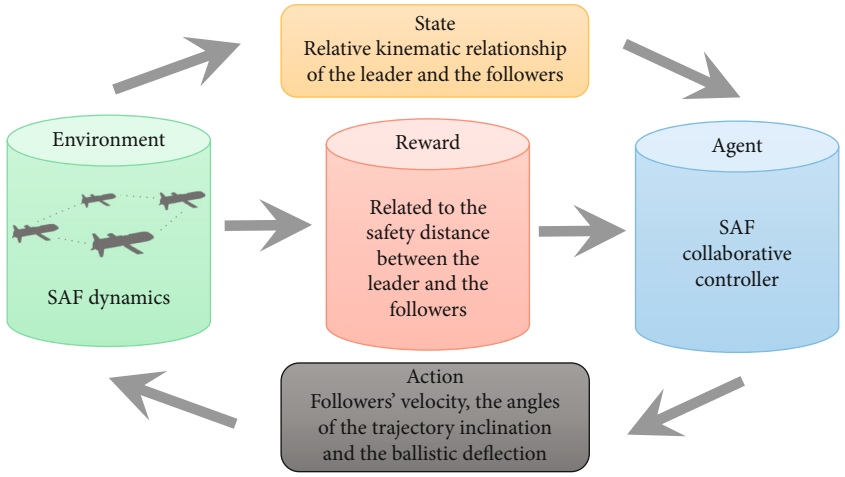

Figure 2: The MDP framework of the SAF collaborative control.

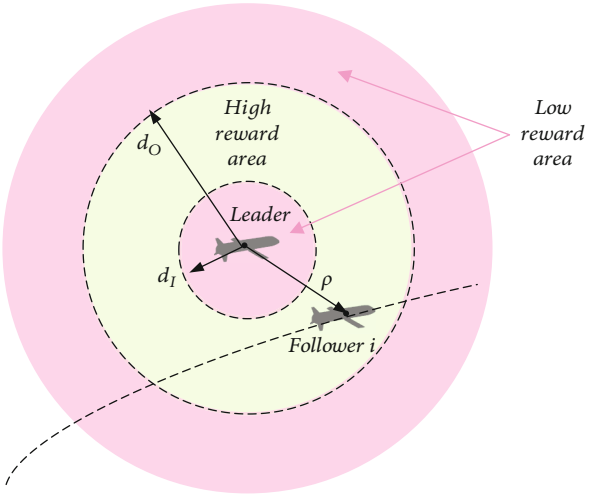

Figure 3: UAV formation collision avoidance scheme.

2.2.2. Action Space. The control of smart ammunition is realized by changing the velocity, the trajectory inclination angle, and the ballistic deflection angle. The control strategy updates the control command in a certain period, and the bottom closed-loop control is completed by the self-driving instrument within the interval. Considering the maximum acceleration of the smart ammunition and avoiding the violent change of control command affecting the safe flight of ammunition, the action space contains the velocity, the trajectory inclination angle, the ballistic deflection angle, and the ballistic deflection angle command of the follower. On the one hand, the followers should be as close as possible to the movement state of the leader. On the other hand, it should avoid the instability of the projectile body, which will cause unsafe flight.

The action space A for the followers can be shown as

$$A = \begin{cases} \{-V_{\max}, \quad 0, \quad +V_{\max}\}, \\ \{-\theta_{\max}, \quad 0, \quad +\theta_{\max}\}, \\ \{-\psi_{V\ \max}, \quad 0, \quad +\psi_{V\ \max}\}, \end{cases} \tag{8}$$

where $V_{\max}, \theta_{\max}, \psi_{V\ \max}$ represent the maximum action candidates for the followers' velocity, the angles of trajectory inclination, and the ballistic deflection, respectively.

The desired action for the next time step can be illustrated as

$$V_d = \begin{cases} V_{bd}, & V + a_V > V_{bd}, \\ -V_{bd}, & V + a_V < -V_{bd}, \\ V + a_V, & \text{otherwise}, \end{cases}$$

$$\theta_d = \begin{cases} \theta_{bd}, & \theta + a_\theta > \theta_{bd}, \\ -\theta_{bd}, & \theta + a_\theta < -\theta_{bd}, \\ \theta + a_\theta, & \text{otherwise}, \end{cases} \tag{9}$$

$$\psi_{Vd} = \begin{cases} \psi_{Vbd}, & \psi_V + a_\psi > \psi_{Vbd}, \\ -\psi_{Vbd}, & \psi_V + a_\psi < -\psi_{Vbd}, \\ \psi_V + a_\psi, & \text{otherwise}, \end{cases}$$

where a_V, a_θ, a_ψ are the chosen actions according to the control requirements of the followers. $V_{bd}, \theta_{bd}, \psi_{Vbd}$ are the thresholds of the follower's velocity, the trajectory inclination angle, and the ballistic deflection angle, respectively.

2.2.3. Reward Function. With the need for configuration maintenance, every node in the formation ought to hold a safe distance from its neighbors. If there is not enough spacing, the collision may happen among these adjacent nodes. And if the distance is very long, the time delay of communication may cause other failures [31]. According to the desired high reward and safe distance range (d_O, d_I) from the leader to the follower, a scheme of collision avoidance and reward evaluation is shown in Figure 3. Every node will get a reward value from the leader depending on the distance to its neighbors. Members in the formation will adjust their states based on these reward values.

In reinforcement learning, it is essential to design a reasonable reward function. The cost function of the SAF

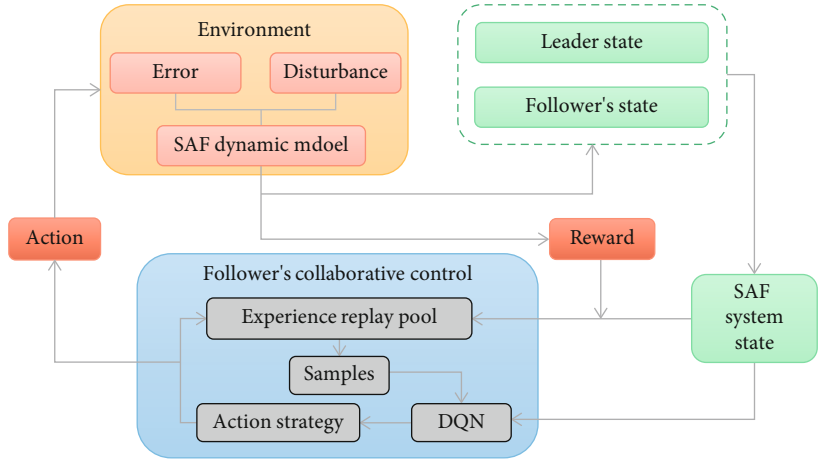

Figure 4: The DQN-based control algorithm framework of the SAF collaborative control.

collaborative control is designed by referring to Ref. [15], and the reward function is defined. The reward function mainly considers the safety distance as what is shown in Figure 3. The reward value limits that the followers are always within the safety distance after the action is performed. The reward function is shown as

$$\begin{cases} r = -\text{Cost}, \\ \text{Cost} = \max\left\{D, \dfrac{d_I|s_2|}{\pi(1+\omega D)}\right\}, \\ D = \max\{d_I - \rho, 0, \rho - d_O\}, \\ \rho = \sqrt{{s_3}^2 + {s_4}^2 + {s_5}^2}, \end{cases} \tag{10}$$

where r is the immediate reward. d_I and d_O are the inner radius and outer radius in Figure 3, respectively. D is the distance between the follower and the circle. ω is the adjust factor which is used to adjust the weight of D. ρ is the distance between the leader and the follower.

3. DQN-Based Control Algorithm

3.1. Basic Framework. In SAF, the follower ammunition receives the system state information from the leader. The control system selects the action by the action selecting strategy and calculates the reward function value through the feedback of the next system state information after executing the action. The advantages and disadvantages of the action strategy are reevaluated by using the real-time return from the smart ammunition, and the cumulative return is maximized. Based on this theoretical framework, the Q-learning algorithm stores and estimates the action-value function of the follower actuator in different states in the MDP model and uses the real-time system state information feedback of the pilot to update the action-value function to solve the optimal sequential decision of the follower actuator iteratively.

Set to the value function estimation $Q(s_t, a_t)$ of the action a_t performed by the follower when it is in the state s_t:

$$Q(s_t, a_t) = E\left(\sum_{t=0}^{\infty} \gamma^t r_t | s_0 = s_t, a_0 = a_t\right), \tag{11}$$

where s_0 is the initial state of the SAF and a_0 is the initial action of the follower.

According to the relevant theories of operation research, $Q(s_t, a_t)$ satisfies the following Bellman equation:

$$\begin{aligned} Q(s_t, a_t) = &\sum_{s_{t+1}} [p(s_t, a_t, s_{t+1}) r(s_t, a_t, s_{t+1})] \\ &+ \gamma \sum_{s_{t+1}, a_{t+1}} [p(s_t, a_t, s_{t+1}) Q(s_{t+1}, a_{t+1})], \end{aligned} \tag{12}$$

where $p(s_t, a_t, s_{t+1})$ is the probability of state s_t transition to s_{t+1} with the follower actions a_t. And $r(s_t, a_t, s_{t+1})$ is the return value of the state s_t transition to s_{t+1} with the follower actions a_t.

Q-learning's optimal strategy $Q(s_t, a_t)$ is to maximize the cumulative return value, so the optimal strategy can be expressed as

$$\pi*(s_t) = \arg \max_{a_t} Q(s_t, a_t). \tag{13}$$

In reinforcement learning, agents constantly interact with the environment in the way of trial and error, to learn an optimal strategy to maximize the cumulative reward they get from the environment [32]. In the Q-learning algorithm, once the Q-value function is determined, the optimal strategy can be determined according to the Q-value function: the agent selects the action with the greedy strategy and selects the action defined by the maximum Q-value at each time step. The Q-learning algorithm is simple to implement and widely used, but it still faces the problem of dimensional disaster. The algorithm usually stores Q-values as tables and is not suitable for reinforcement learning problems in high-dimensional or continuous state space [33].

To solve the above problem, a deep neural network (DNN) which is used as a function approximator to estimate the Q-value has become an alternative [34]. Adapting neural

Input: the max chosen action of the follower's velocity, the trajectory inclination angle, and the ballistic deflection angle, $V_{\max}$, $\theta_{\max}$, $\psi_{V\ \max}$, respectively. The spatial position state of the leader and n followers, (x_L, y_L, z_L), (x_{Fi}, y_{Fi}, z_{Fi}), $i = 1, 2, \cdots, n$. The threshold $(\Delta X_{\text{desire}}, \Delta Y_{\text{desire}}, \Delta Z_{\text{desire}})$.
Output: n followers' action A_F.

```
Initialize the random number, e ∈ (0, 1)
for i = 1, 2, ⋯, n do
  if e > ε then
    A_Fi = A_max = [V_max, θ_max, ψ_V max]
  else
    if x_Fi − x_L > ΔX_desire then
      V_Fi = −V_max
    else if x_Fi − x_L < −ΔX_desire then
      V_Fi = V_max
    else
      V_Fi = 0.0
    end if
    if y_Fi − y_L > ΔY_desire then
      θ_Fi = −θ_max
    else if y_Fi − y_L < −ΔY_desire then
      θ_Fi = θ_max
    else
      θ_Fi = 0.0
    end if
    if z_Fi − z_L > ΔZ_desire then
      ψ_VFi = −ψ_V max
    else if z_Fi − z_L < −ΔZ_desire then
      ψ_VFi = ψ_V max
    else
      ψ_VFi = 0.0
    end if
  end if
end for
```

Algorithm 1: ε-imitation action selection strategy.

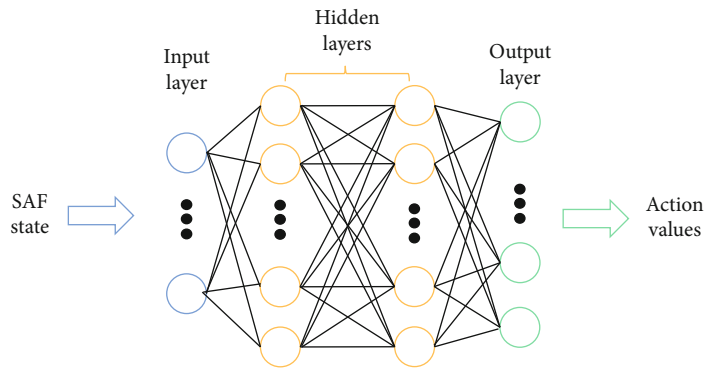

Figure 5: The framework of a Q-network.

network (CNN) and empirical playback technology to introduce a Q-learning algorithm, Minh et al. illustrated a DQN algorithm [35]. A separate target network to generate Q-values was used to reduce the correlation between the predicted Q-value (main network output) and the target Q-value (target network output) and ease the instability of the neural network approximation function to a certain extent.

According to Equation (11), after obtaining the maximum Q-function, we need to derive the optimal policy. Using the recursive mechanism, the Q-function can be updated as [36]

$$Q(s_t, a) = Q(s_t, a) + \lambda\left[r_{t+1} + \gamma \max_a Q(s_{t+1}, a) - Q(s_t, a)\right], \tag{14}$$

where λ is the learning rate.

1: Initialize Q-network randomly with weights θ.
2: Initialize the target network $\widehat{Q}$ with weights $\theta^{-} = \theta$.
3: Initialize experience pool to capacity N, greedy probability ε, size of minibatch samples n_e, discount factor γ, learning rate λ, and update period K of the target network.
4: Initialize the planned operation time T and time interval Δt and calculate N with $T/\Delta t$.
5: **for** episode = $1, 2, \cdots, N_S$ **do**
6: Initialize the SAF state according to the system's initial characteristics.
7: **while** $T_1 < T_{\text{Episode}}$ **do**
8: Select followers' action a_t according to ε-imitation policy.
9: Perform action a_t on the SAF system, calculate Equations (2) and (3) with the fourth-order Runge-Kutta method, get the system state s_{t+1} at next time, and observe reward r_t.
10: Store transition (s_t, a_t, r_t, s_{t+1}) in experience pool.
11: Randomly sample a minibatch of n_e transitions from the experience pool.
12: Train the network and update the parameters using Variable Learning Rate Gradient Descent.
13: Every K steps update $\theta^{-} = \theta$.
14: **end while**
15: **end for**

Algorithm 2: DQN algorithm.

The target Q-value can be shown as

$$y_t^{\text{DQN}} = r_{t+1} + \gamma \max_a Q(s_{t+1}, a, \theta^{-}), \tag{15}$$

where θ^{-} is the parameter of the target network.

The minimization loss function is

$$L(\theta) = E\left[\left(y_t^{\text{DQN}} - Q(s_{t+1}, a_t|\theta)\right)^2\right]. \tag{16}$$

As DQN illustrated, the difference between the main network estimated Q-value and the target network output Q-value will update the main network parameters in real time. Different from the real-time updated parameters of the main network, the target network parameters are updated every K time steps. The main network parameters are copied to the target network to complete the target network parameters every K time step [15].

In this section, we develop a DQN-based control approach for the SAF collaborative control. As a novel variant of Q-learning, the DQN algorithm combines RL with artificial neural networks [35]. To address the instabilities occurring when the action-value function (Q-function) is approximated using neural networks, the periodically updated separate target Q-network and experience replay mechanism are introduced in the DQN algorithm. Thus, the DQN has been successfully applied to a variety of domains, such as agriculture, communication, medical aspects, social security, transportation, service industry, financial industry, big data processing, and aerospace engineering. The framework of the DQN-based control algorithm is described in Figure 4.

As what is described in Figure 4, the followers are mapped to the agents in RL, which learn the control strategy and update the network parameters in the continuous interaction with the environment. The followers get the state information of the leader and their own state information. This state information forms a joint system state S and is inputted into the DQN. Action selection strategy, ε-imitation (ε represents the exploration rate), selects the follower's velocity, the trajectory inclination angle, and the ballistic deflection angle according to the output of DQN. The action commands of the leader and the followers are inputted into the kinematics model of the SAF, to get the state of the leader and the followers, respectively, the next time. The reward function value R and the system state S' at the next time can also be obtained. (S, A, R, S') generated in the interaction process are maintained in the experience pool. At each time step, random sampling is carried out from the experience pool, and the network parameters of DQN are updated. When the time step of each round reaches a certain number of steps, the current episode ends and the next one starts [15].

3.2. Action Strategy. To improve the learning efficiency of DQN in the training stage, an ε-imitation action selection strategy, which is a combination of ε greedy strategy and imitation strategy, is used to balance learning exploration and utilization [15]. The imitation strategy is that the follower selects its control command according to the relative distance control requirements. The main idea of this strategy is that when the followers select the action from the action space with 1-ε probability, the action is selected based on the desired relative distance in the x, y, z direction between the leader and the follower which has been mentioned in Equation (4). Specifically, when the relative distance between the leader and the follower is beyond the threshold $(\Delta X_{\text{desire}}, \Delta Y_{\text{desire}}, \Delta Z_{\text{desire}})$ in x, y, or z direction, there is an $A_{\max}$ chosen from the action space to reduce the distance. If the relative distance is less than the threshold, the follower should maintain the current states, which means the action is equal to zero. ε-imitation action selection strategy is a benefit for the topology keeping in the SAF flight and reduces the blindness of

Table 2: Parameter set for the DQN.

Parameter	Value	Parameter	Value
Inner radius d_I (m)	600	Total episodes N_s	5×10^4
Outer radius d_O (m)	1000	Each episode's time t_E (s)	120
Adjust factor ω	0.05	Time step Δt (s)	1.0
Return discount factor γ	0.95	Episode number N_{Avg} to calculate the average total reward	100
Update period of the target network K	1000	η_V's mean value and variance $(\bar{\eta}_V, \sigma_V)$	$(V_0, 0.8)$
Exploration probability ε	0.1	η_θ's mean value and variance $(\bar{\eta}_\theta, \sigma_\theta)$	(0.0, 1.0)
Number of followers	2	η_{ψ_V}'s mean value and variance $\left(\bar{\eta}_{\psi_V}, \sigma_{\psi_V}\right)$	(0.0, 1.0)
Capacity of experience replay pool N	10^5	η_x's mean value and variance $(\bar{\eta}_x, \sigma_x)$	(0.0, 1.0)
Mini-batch size n_e	32	η_y's mean value and variance $\left(\bar{\eta}_y, \sigma_y\right)$	(0.0, 1.0)
Learning rate λ	0.01	η_z's mean value and variance $(\bar{\eta}_z, \sigma_z)$	(0.0, 1.0)

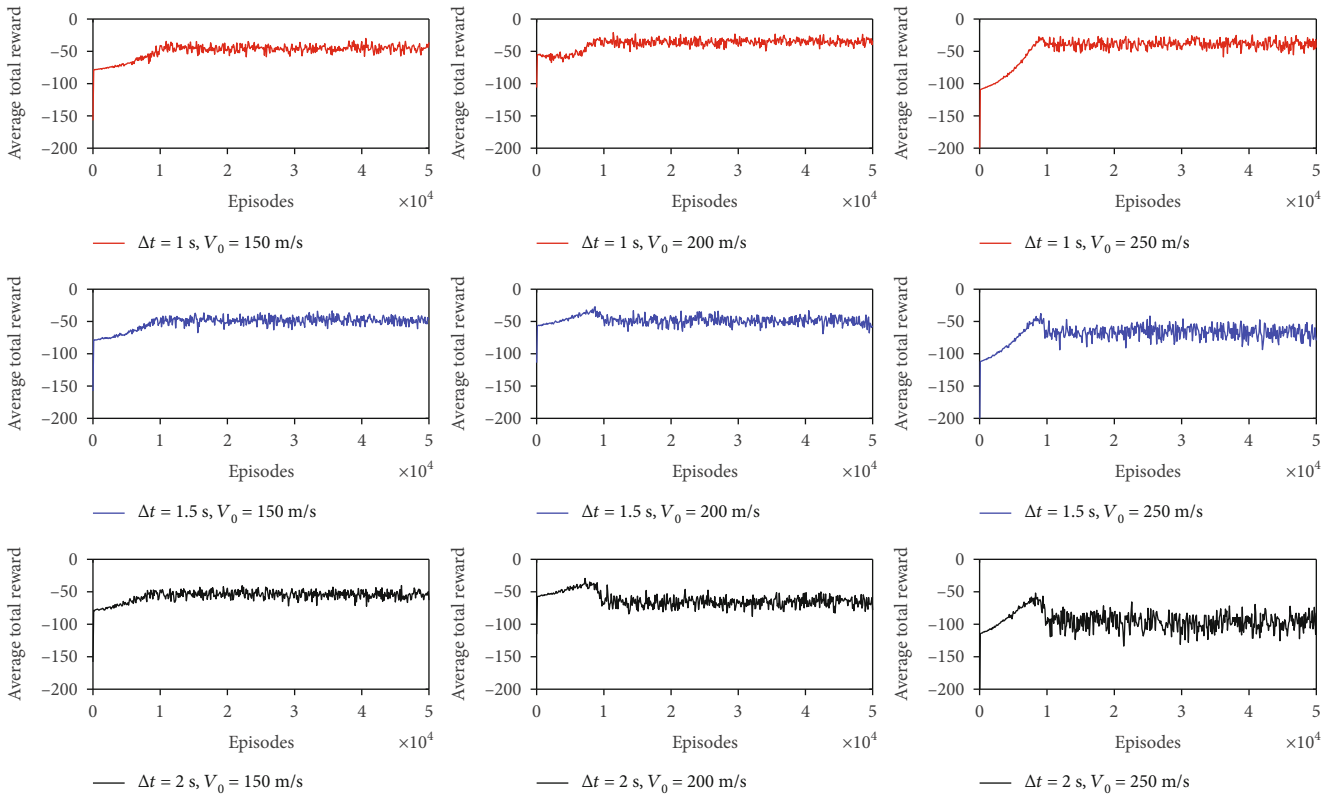

Figure 6: The change of the average total rewards per episode in tests 1-9.

the follower in the initial exploration stage. This also reduces the number of invalid explorations, increases the number of positive samples in the experience pool, and helps to improve the training efficiency.

ε-imitation action selection strategy is illustrated in Algorithm 1.

3.3. DQN Algorithm. In the DQN framework, the Q-function is approximated by a neural network with weight parameters θ, referred to as a Q-network. To evaluate the Q-value, a fully connected Q-network is built as what is shown in Figure 5. At the t step, the SAF state is received by the input layer of the Q-network. Each node in the output layer corresponds to the Q-value of each possible action. The network contains one input layer, two hidden layers, and one output layer. The size of the hidden layer is 40×40, and the training function is set as the Variable Learning Rate Gradient Descent.

The DQN algorithm is used to realize the formation coordination control of the SAF. The training process is

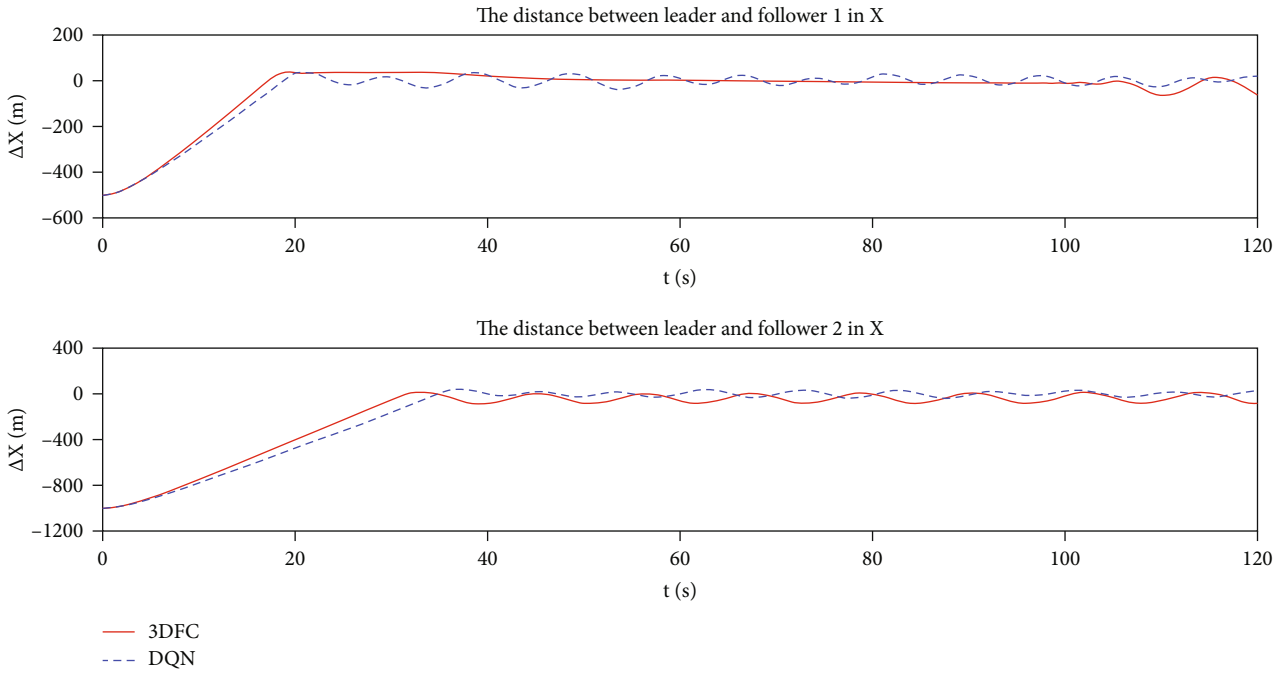

Figure 7: The relative distance ΔX between the leader and the followers.

shown in Figure 4, where the main implementation steps of the coordination control algorithm based on DQN are proposed in Algorithm 2 [34].

4. Results and Discussion

The simulations were performed in Visual Studio 2019 with a set of C++ codes. The SAF contains one leader and two followers which is the same as the configuration illustrated in Ref. [24]. To verify the effectiveness and novelty of the DQN-based control algorithm proposed in this work, the performance of the controller will be compared with a three-dimensional formation controller (3DFC) in Ref. [24]. The parameters referred to the SAF dynamics are shown as follows.

(1) Time parameter:

$$\tau_{VL} = \tau_{VF} = 3.0, \tau_{\theta L} = \tau_{\theta F} = 3.0, \tau_{\psi VL} = \tau_{\psi VF} = 3.0. \quad (17)$$

(2) Desired relative position:

$$\Delta X_{\text{desire}} = 0, \Delta Y_{\text{desire}} = 0, \Delta Z_{\text{desire}} = 800. \quad (18)$$

(3) Control command of the leader's velocity, the trajectory inclination angle, and the ballistic deflection angle:

$$\begin{aligned} V_{LC} &= 200.0, \\ \theta_{LC} &= 0.0, \\ \psi_{VLC} &= 0.05 \sin(0.1t). \end{aligned} \quad (19)$$

(4) Control range of the follower's velocity, the trajectory inclination angle, and the ballistic deflection angle:

$$\begin{aligned} 165 < V_{FC} < 235, \\ -10^\circ < \theta_{FC} < 10^\circ, \\ -5^\circ < \psi_{VFC} < 5^\circ. \end{aligned} \quad (20)$$

The physical parameters of the leader and the followers are shown in Table 2.

The detailed parameter settings for the DQN are given in Table 1 [15, 24].

4.1. Sensitivity Analysis of Parameters. To evaluate the effectiveness of the DQN-based algorithm adopted in this work, an average total reward R_{Avg} was built as a criterion. R_{Avg} is defined as [15]

$$\begin{aligned} N_E &= \frac{t_E}{\Delta t}, \\ R_{\text{Avg}} &= \frac{1}{N_{\text{Avg}} N_E} \sum_{n=1}^{N_{\text{Avg}}} \sum_{t=1}^{N_E} r_{n,t}, \end{aligned} \quad (21)$$

where r is the immediate reward in Equation (10).

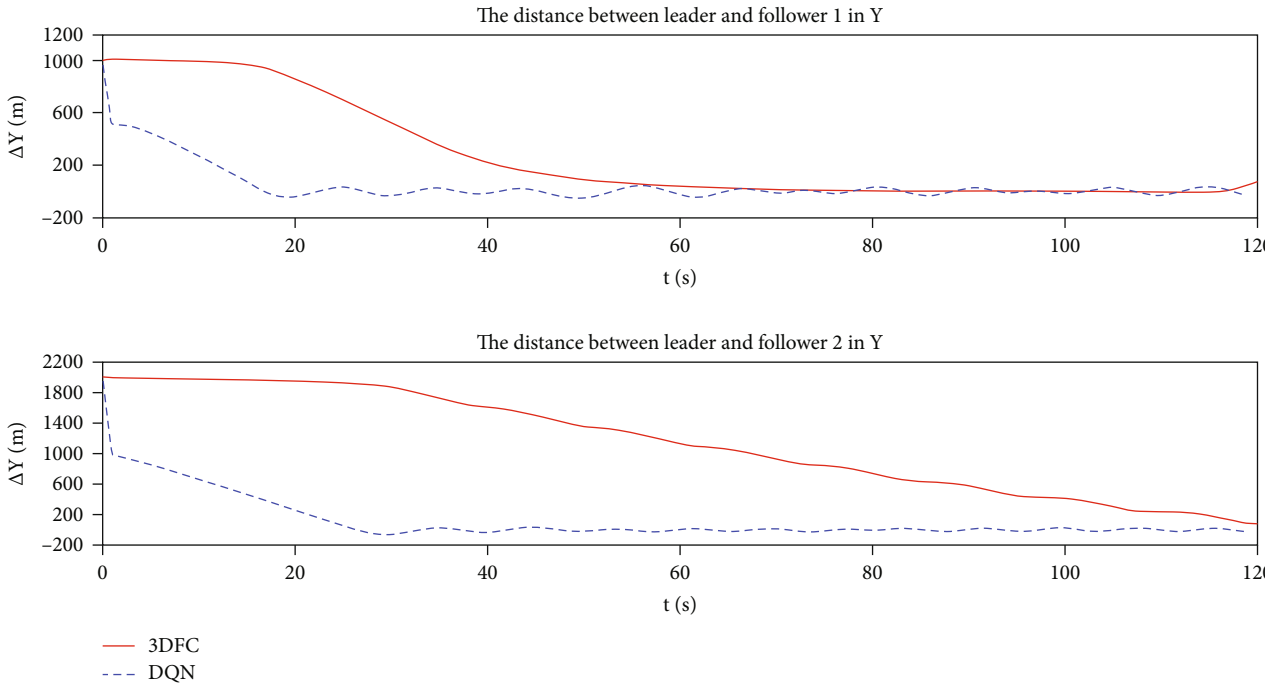

Figure 8: The relative distance ΔY between the leader and the followers.

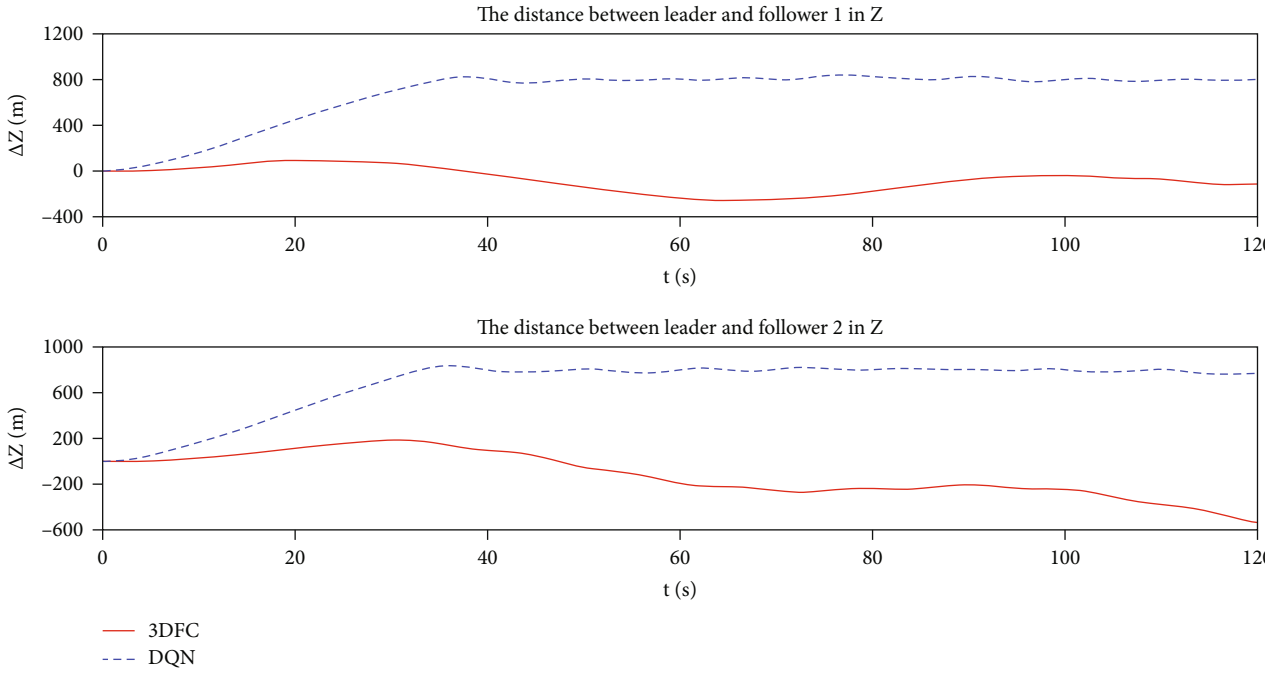

Figure 9: The relative distance ΔZ between the leader and the followers.

Here, nine tests are conducted to investigate the influences of the time interval Δt and initial speed V_0 on the optimal control strategy of the SAF. Concretely, each episode time t_E is still set to be 120 s, but the time interval Δt will be 1 s, 1.5 s, and 2 s. Moreover, the initial speed V_0 is assumed to be 150 m/s, 200 m/s, and 250 m/s. The other parameter settings are the same as those in Tables 1 and 2.

Figure 6 shows the changes in the average total rewards per episode with the increase of the training episodes in tests 1-9. It shows that the average total rewards converge after 50,000 episodes of training in all nine tests. In the case of $\Delta t = 1$ s, the average total rewards are all around -50 after the curves showing convergence. However, after reaching the peak values, the reward values decrease more clearly with the increase of the initial speed. Additionally, it shows that

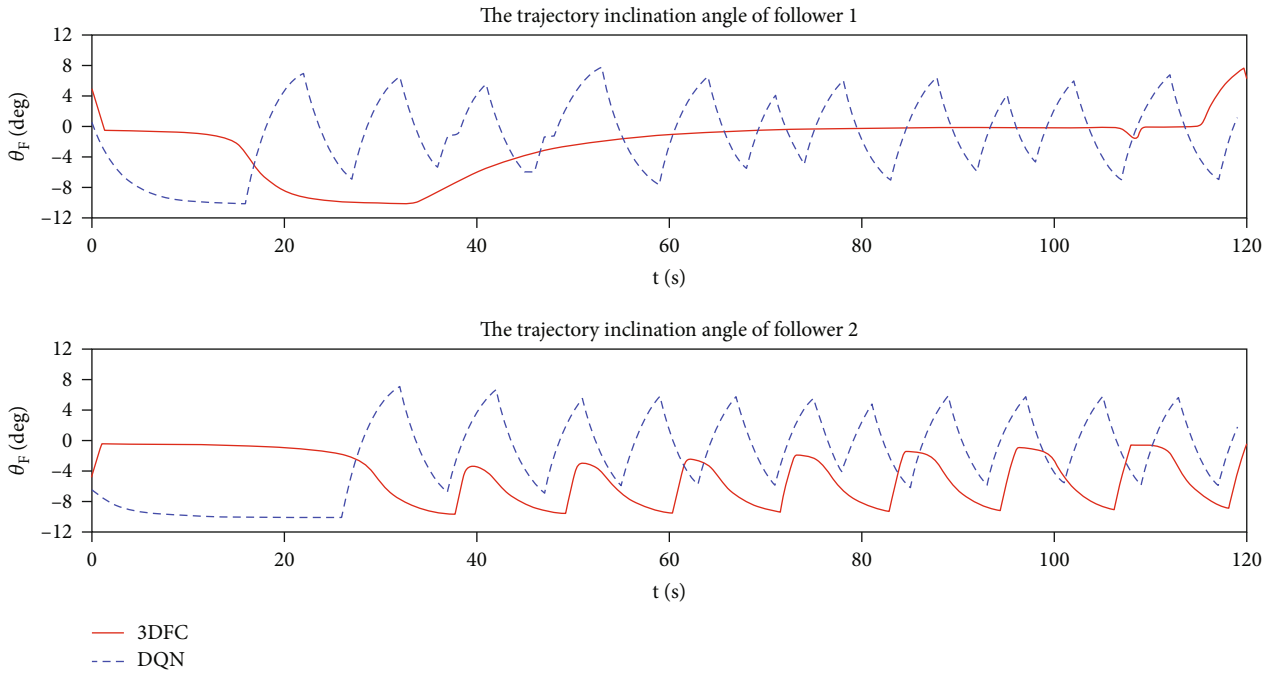

Figure 10: The trajectory inclination angle of the followers.

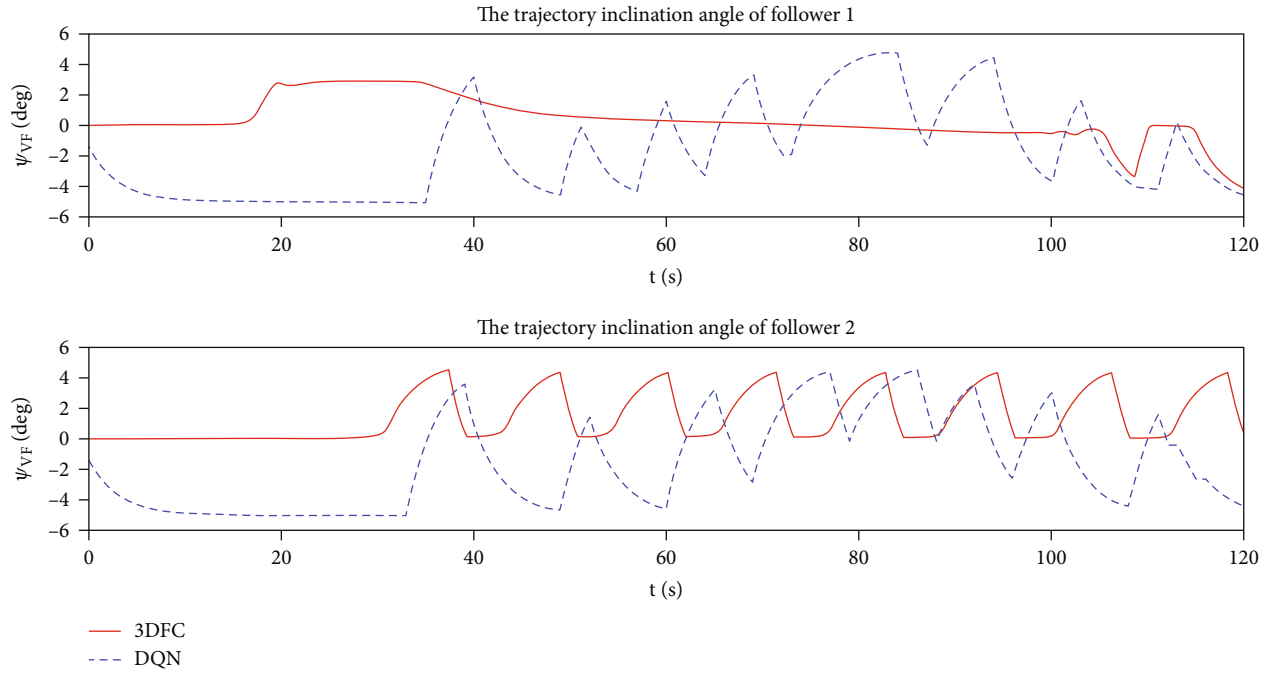

Figure 11: The ballistic deflection angle of the followers.

the curves converge earlier with the initial speed increasing. For other tests of different time intervals, they also show the same trend with the $\Delta t = 1$ s case. For the cases of different time intervals with the same initial speed, the reward values decline more greatly as the rise of the time interval. Considering the above results, to ensure the efficiency of calculation and consider the working overload of the smart ammunition, the time interval is set as 1 s, and the initial speed is 200 m/s for the following tests.

4.2. Evaluation of DQN-Based Algorithm. Figures 7, 8, and 9 show the relative distance-time history curves of the leader and the follower projectile in the X, Y, and Z directions, respectively. The initial states and simulation parameters are all the same for 3DFC and the DQN-based algorithm. Figure 7 shows a similar trend of the relative distance in the X direction for both followers. With 3DFC or the DQN algorithm, the distance ΔX of the SAF meets the requirements ΔX_{desire} and remains until the calculation is

ended. In the Y direction, as is presented in Figure 8, it is slower for 3DFC to reach the desired distance ΔY_{desire} in comparison with DQN. In Figure 9, DQN acts a better performance than 3DFC, and the distances in the Z direction meet the requirement during the simulation time of 30-40 seconds.

Figures 10 and 11 show the comparisons of the trajectory inclination angle, and the ballistic deflection angle of different followers with 3DFC and the DQN-based algorithms. With the help of ε-imitation action selection strategy, there are more homogeneous changes of different angle states with the DQN-based controller. Additionally, the changing amplitudes are greater according to ε-imitation.

5. Conclusions

In this study, we aimed to develop a novel method to overcome the uncertainty, nonlinearity, system error, and disturbances in the process of SAF modeling. Based on a DQN algorithm, an intelligent control scheme was presented for the SAF collaborative control task. The environment was described to propose the system joint states, which referred to the smart ammunition's velocity, the trajectory inclination angle, the ballistic deflection angle, and the relative position in the formation. Then, an MDP model was adapted to represent the SAF collaborative control process, which combined the RL basic framework. After that, the detailed DQN algorithm was introduced, which included the basic framework, ε-imitation action selecting a strategy, and the algorithm description. Finally, a simulation experiment was implemented to verify the validity and applicability of the DQN control scheme mentioned in this paper.

According to the simulation results, the DQN-based algorithm acts as a novel performance in the SAF collaborative control. The average total reward curve shows a reasonable convergence, and the relative kinematic relationship among the formation nodes meets the requirements of the collaborative controller design. The future work will extend to the high-fidelity loop simulation hardware system, which will be constructed to verify the effectiveness and portability of the DQN-based algorithm. The control strategy training in the numerical simulation environment can be directly transferred to the hardware in the loop simulation system without too much parameter adjustment.

Data Availability

All data included in this study are available upon request via contact with the corresponding author.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (grant numbers 62003314 and 51909245), Aeronautical Science Foundation of China (grant number 2019020U0002), and Youth Science and Technology Research Fund, Shanxi Province Applied Basic Research Project (grant number 201901D211244).

References

[1] E. J. Zhao and M. W. Sun, "Review on the key technology of collaborative engagement for multiple flight vehicles," *Tactical Smart Ammunition Technology*, vol. 202, no. 4, pp. 175–182, 2020.

[2] S. T. Wu, *Collaborative Guidance & Control of Smart Ammunition Autonomous Formation*, National Defense Industry Press, Beijing, 1st edition, 2015.

[3] K. J. Zhang, H. S. Lin, and B. Xia, "New development of smart ammunition cluster intelligent penetration technology," *Tactical smart ammunition technology*, vol. 191, no. 5, pp. 1–5, 2018.

[4] J. B. Zhao and S. X. Yang, "Review of multi-smart ammunition collaborative guidance," *Acta Aeronauticaet Astronautica Sinica*, vol. 38, no. 1, pp. 20256–020256, 2017.

[5] Z. J. Jiang and W. L. Yang, "Analysis and effects of U.S. accelerating the development of smart ammunition cluster combat capacity," *Tactical Smart Ammunition Technology*, vol. 202, no. 4, pp. 189–192, 2020.

[6] Z. Jie, H. D. Li, and X. Bin, "Cooperative salvo attack using guidance law of multiple missiles," *Journal of Advanced Computational Intelligence and Smart Informatics*, vol. 19, no. 2, pp. 301–306, 2015.

[7] X. L. Wang and H. L. Wu, "Distributed cooperative guidance of multiple anti-ship missiles with arbitrary impact angle constraint," *Aerospace Science and Technology*, vol. 46, pp. 299–311, 2015.

[8] L. Song, Y. A. Zhang, D. Huang, and S. Fu, "Cooperative simultaneous attack of multi-missiles under unreliable and noisy communication network: a consensus scheme of impact time," *Aerospace Science and Technology*, vol. 47, pp. 31–41, 2015.

[9] J. J. Wu, F. T. Zhu, and C. Song, "Optimal discretization of feedback control in missile formation," *Aerospace Science and Technology*, vol. 67, pp. 456–472, 2017.

[10] S. Cohen and N. Agmon, "Recent advances in formations of multiple robots," *Current Robotics Reports*, vol. 2, no. 2, pp. 159–175, 2021.

[11] K. Hou, Y. J. Yang, X. R. Yang, and J. Z. Lai, "Cooperative control and communication of intelligent swarms: a survey," *Control Theory Technology*, vol. 18, no. 2, pp. 114–134, 2020.

[12] A. Mohamed, G. Samet, J. Hassan, and S. S. Jeff, "Aerial swarms: recent applications and challenges," *Current Robotics Reports*, vol. 2, no. 3, pp. 309–320, 2021.

[13] Z. Zhang, K. Zhang, and Z. Han, "A novel cooperative control system of multi-missile formation under uncontrollable speed," *IEEE Access*, vol. 9, pp. 9753–9770, 2021.

[14] Y. Zhen, J. Q. Yuan, Q. X. Chi, and M. R. Hao, "Research on application of DRL method in aircraft control," *Tactical Smart Ammunition Technology*, vol. 4, pp. 112–118, 2020.

[15] X. J. Xiang, C. Yan, C. Wang, and D. Yin, "Coordination control method for fixed-wing UAV formation through DRL," *Acta Aeronauticaet Astronautica Sinica*, vol. 42, no. 4, pp. 524009–524009, 2021.

[16] W. B. Du, T. Guo, J. Chen, B. Y. Li, G. X. Zhu, and X. B. Cao, "Cooperative pursuit of unauthorized UAVs in urban airspace

via multi-agent reinforcement learning," *Transportation Research Part C: Emerging Technologies*, vol. 128, article 103122, 2021.

[17] H. M. Wang, T. H. Qiu, Z. Liu, Z. Q. Pu, and J. Q. Yi, "Multi-agent formation control with obstacles avoidance under restricted communication through graph reinforcement learning," *IFAC-Papers OnLine*, vol. 53, no. 2, pp. 8150–8156, 2020.

[18] Z. Z. Sui, Z. Q. Pu, J. Q. Yi, and S. G. Wu, "Formation control with collision avoidance through deep reinforcement learning using model-guided demonstration," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 32, no. 6, pp. 2358–2372, 2021.

[19] J. C. Ma, H. M. Lu, J. H. Xiao, Z. W. Zeng, and Z. Q. Zheng, "Multi-robot target encirclement control with collision avoidance via deep reinforcement learning," *Journal of Intelligent & Robotic Systems*, vol. 99, no. 2, pp. 371–386, 2020.

[20] J. Tožička, B. Szulyovszky, C. G. De, V. Sarwal, U. Wani, and M. Gribulis, *Application of Deep Reinforcement Learning to UAV Fleet Control*, Springer, Cham, 2018.

[21] Y. J. Zhao, Y. Ma, and S. L. Hu, "USV formation and path-following control via deep reinforcement learning with random braking," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 32, no. 12, pp. 5468–5478, 2021.

[22] S. Brenton, A. Rasit, A. Joshua, and R. Boyce, "Propulsionless planar phasing of multiple satellites using deep reinforcement learning," *Advances in Space Research*, vol. 67, no. 11, pp. 3667–3682, 2021.

[23] S. W. Wang, F. Ma, X. P. Yan, P. Wu, and Y. C. Liu, "Adaptive and extendable control of unmanned surface vehicle formations using distributed deep reinforcement learning," *Applied Ocean research*, vol. 110, article 102590, 2021.

[24] J. Shen, Q. Y. Zhu, L. Xu, G. G. Chen, X. L. Tian, and X. L. Yan, "Research on dynamic simulation and collaborative control of smart ammunition formation," in *2020 IEEE International Conference on Mechatronics and Automation (ICMA)*, pp. 1488–1492, Beijing, China, October 2020.

[25] L. B. Zong, F. Xie, and S. Y. Qin, "Intelligent optimizing control of formation flight for UAVs based on MAS," *Acta Aeronauticaet Astronautica Sinica*, vol. 29, no. 5, pp. 1326–1333, 2008.

[26] Y. Li, F. Xu, G. Q. Xie, and X. L. Huang, "Survey of development and application of multi-agent technology," *Computer Engineering and Application*, vol. 54, no. 9, pp. 13–21, 2018.

[27] F. Liu, Y. L. Li, and H. Y. Yang, "Based on multi-source interference consistency of the leader-follower multi-agent systems," *Information and Control*, vol. 47, no. 1, pp. 111–118, 2018.

[28] P. B. Ma and J. Ji, "Three-dimensional formation control of multi-smart ammunition," *Acta Aeronautica Sinica*, vol. 31, no. 8, pp. 1660–1666, 2010.

[29] J. Shen, Q. Y. Zhu, X. G. Wang, and P. Y. Chen, "Typical fault estimation and dynamic analysis of a leader-follower unmanned aerial vehicle formation," *International Journal of Aerospace Engineering*, vol. 2021, Article ID 6656422, 16 pages, 2021.

[30] B. L. Wang, S. G. Li, X. Z. Gao, and T. Xie, "UAV swarm confrontation using hierarchical multiagent reinforcement learning," *International Journal of Aerospace Engineering*, vol. 2021, Article ID 3360116, 12 pages, 2021.

[31] L. T. Jiang, R. X. Wei, Q. R. Zhang, and D. Wang, "Anti-collision control of UAVs based on swarm intelligence mechanism," *Acta Aeronauticaet Astronautica Sinica*, vol. 41, no. S2, pp. 111–118, 2020.

[32] Z. C. Wei, F. Liu, Y. Zhang, J. Xu, J. J. Ji, and Z. W. Lyu, "A Q-learning algorithm for task scheduling based on improved SVM in wireless sensor networks," *Computer Networks*, vol. 161, pp. 138–149, 2019.

[33] X. Q. Wen and Z. A. Xu, "Wind turbine fault diagnosis based on ReliefF-PCA and DNN," *Expert Systems with Applications*, vol. 178, article 115016, 2021.

[34] W. Liu, S. Su, T. Tang, and X. Wang, "A DQN-based intelligent control method for heavy haul trains on long steep downhill section," *Transportation Research Part C Emerging Technologies*, vol. 129, no. 10, article 103249, 2021.

[35] V. Mnih, K. Kavukcuoglu, D. Silver et al., "Human-level control through deep reinforcement learning," *Nature*, vol. 518, no. 7540, pp. 529–533, 2015.

[36] H. J. Huang, Y. C. Yang, H. Wang, Z. G. Ding, H. Sari, and F. Adachi, "Deep reinforcement learning for UAV navigation through massive MIMO technique," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 1, pp. 1117–1121, 2020.