electronics

MDPI

Article

Online Adaptive Dynamic Programming-Based Solution of Networked Multiple-Pursuer and Single-Evader Game

Zifeng Gong, Bing He *, Chen Hu, Xiaobo Zhang and Weijie Kang

Department of Nuclear Engineering, PLA Rocket Force University of Engineering, Xi'an 710025, China
* Correspondence: celadongong@mau.edu.mk; Tel.: +86-195-2603-7392

Abstract: This paper presents a new scheme for the online solution of a networked multi-agent pursuit–evasion game based on an online adaptive dynamic programming method. As a multi-agent in the game can form an Internet of Things (IoT) system, by incorporating the relative distance and the control energy as the performance index, the expression of the policies when the agents reach the Nash equilibrium is obtained and proved by the minmax principle. By constructing a Lyapunov function, the capture conditions of the game are obtained and discussed. In order to enable each agent to obtain the policy for reaching the Nash equilibrium in real time, the online adaptive dynamic programming method is used to solve the game problem. Furthermore, the parameters of the neural network are fitted by value function approximation, which avoids the difficulties of solving the Hamilton-Jacobi–Isaacs equation, and the numerical solution of the Nash equilibrium is obtained. Simulation results depict the feasibility of the proposed method for use on multi-agent pursuit–evasion games.

Keywords: multi-agent pursuit–evasion game; differential game; adaptive dynamic programming; policy iteration; value function approximation

Citation: Gong, Z.; He, B.; Hu, C.; Zhang, X.; Kang, W. Online Adaptive Dynamic Programming-Based Solution of Networked Multiple-Pursuer and Single-Evader Game. *Electronics* **2022**, *11*, 3583. https://doi.org/10.3390/electronics11213583

Academic Editors: Srikanta Patnaik, Shengqing Li and Jianqi Li

Received: 8 October 2022
Accepted: 30 October 2022
Published: 2 November 2022

1. Introduction

Recently, the differential game of multi-agents has won the favor of many scholars for its critical application prospects [1–4]. Among them, games of multi-pursuer and single-evader have been widely considered in many guidance and interception problems. In many scenarios, the agents may involve large-scale or complex dynamic systems, which make the decisions of agents difficult to resolve [5], and the constraint of energy consumption is often considered with the application of renewable energy. Although many methods can solve differential games, most of the existing algorithms cannot solve them online, or require a lot of empirical data. Therefore, this paper studies the problem of an online multi-pursuer single-evader game and resolves the decision of the agents through the method of integral reinforcement learning.

The pursuit–evasion game is a kind of common differential game, which is usually used in competitive games, the optimization of IoT resources, military attacks, and so on [6,7] Among them, the simplest scenario is the single-pursuer single-evader game. This game is a zero-sum game, in which the pursuer and the evader have mutually exclusive interests, and no other agents participate in the interference [8]. However, for a game problem involving multiple agents, the benefit of the game will become complex [9]. The difficulty of solving the multi-agent pursuit–evasion game is closely related to the size of the communication network between individuals and the complexity of the game model [10]. Among them, ref. [11] makes a detailed derivation and solution for the capture game problem of agents. Cappello [12] divided 12 pursuit–evasion game of multi-agents into three cases, and the solution of its Nash equilibrium is given. In addition, when the number of agents in the game is quite large, Peng [13] adopts a distributed network so that the decision-making of other individuals is not affected after the failure or damage

of each individual, which means the system has scalability, self-organization, and strong robustness [14].

For an actual multi-agent pursuit–evasion (MPE) game, the core is to solve the decision of each agent according to the setting of the value function. Nowadays, there are many methods with which to solve the strategies of agents in the game. In Refs. [15,16], the scholars solved the pursuit and evasion game problem of multi-agents with an analytical method, obtained the Nash equilibrium solution of each multi-agent, and proved the existence condition of the Nash equilibrium. However, for a complex game system, the process of obtaining an analytical solution consumes a lot of computational costs and time costs. In Ref. [17], the reinforcement learning method was used in the Nash equilibrium game for solving the decisions of agents, and the numerical solution equivalent to the analytical solution was obtained. In addition, recently there have been many off-line algorithms that were able to simplify the solution of multi-agent pursuit and evasion games [18–20]. The off-line algorithm for the pursuit–evasion problem in the differential game is becoming more and more mature. However, off-line methods are not competent enough to realize the flexibility of emergency changes in implementing the online confrontation.

Therefore, how to solve the problem of the MPE game online has become a heated topic in the academic community. In 2002, Murray et al. first proposed the iterative adaptive dynamic programming (ADP) algorithm for continuous systems [21] and first adopted the policy iterative algorithm in Ref. [22]. At this time, the ADP algorithm can only be used as an off-line iteration. Vamvoudakis et al. proposed an online adaptive method based on policy iteration [23–25] to solve the optimal control problem of continuous nonlinear systems and theoretically proved the stability of this online adaptive algorithm. This online adaptive method has also been studied in discrete systems. Using the ADP method to solve differential game problems has also begun to develop in the direction of online learning. An online adaptive control scheme based on policy iteration for multi-person non-zero-sum differential game problems was proposed [26]. Owing some information about a system may not be known, Wei [27] considered both linear and nonlinear systems to compute an online learning method in optimizing control with unknown information about the system matrix, and an event-triggered ADP method with multiple triggering conditions was developed for multi-player non-zero-sum (NZS) games [28]. In Ref. [29], a novel data-based ADP method was presented to solve the optimal consensus tracking control problem for discrete-time (DT) multi-agent systems (MASs) with multiple time delays. However, in the process of policy iteration, the excitation signal may drop to very low over time, making the approximation algorithm difficult to operate; thus, Karg [30] focused on the fulfillment of the persistent excitation condition for signals which result from transformations by means of polynomials. Moreover, Li [31] discussed the feasibility of this method in a distributed system. For the multi-agent PE game problem with uncertain system parameters, it is still difficult to solve because of the complexity of the scale of the actual systems. Furthermore, solving the MPE game problem in real time without knowing all the information about the systems has become a valuable research topic.

In this paper, the ADP method is used to solve the networked MPE game problem with multiple pursuers and a single evader in real time. In order to realize the implementation and solution of the game, we divide the game process into minimal time intervals through integral reinforcement learning, and then we obtain the policy of each agent by a policy iteration method. Through continuous iterations, the game converges to the Nash equilibrium, and the policy of each agent will converge to its Nash equilibrium policy. In order to eliminate the difficulty of solving the HJI equation in a multi-agent game, as the state information of each agent can be perceived mutually under the IoT system, the value function approximation method is used, and finally the numerical solution of each agent's Nash equilibrium policy is obtained. Using the state information provided by the IoT system, we can obtain the Nash equilibrium policies that consider both distance and energy control in the game. Simulation experiments are used to verify the effectiveness of the method.

The main contributions of this paper are listed below:

1. The relative distance and the control energy are incorporated as the performance index, and the Nash equilibrium of an MPE game is obtained by using the minmax principle, in which the Lyapunov function is constructed to verify whether capture scene occurs;
2. The multiple iterative interval is divided in the whole game process by conducting the integral reinforcement learning model, and the policy of each iterative interval is obtained by the policy iteration method, which is proved to converge to the Nash equilibrium solution;
3. The online ADP method is adopted to overcome the difficulty of solving the HJB equation, and a set of approximation functions are established by using the method of value function approximation. The numerical solution of the policies of the agents is obtained.

In simple terms, the original contribution of this paper is to establish an MPE game model by building a weighted value function and creating a basis function to fit the value function using the ADP method. We obtain the Nash equilibrium solution of the game by learning the neural network parameters. The capture conditions are also analyzed by using the Lyapunov function method.

The remainder of this paper is organized as follows: Section 2 constructs the physical model of the multi-agent pursuit–evasion game. Section 3 discusses the Nash equilibrium solution and capture conditions of the game problem. Section 4 discusses the integral reinforcement learning method, i.e., the policy iteration and value function approximation, to obtain the policies without solving the HJI equation directly. Section 5 demonstrates the simulation of a practical problem. Section 6 is the conclusion of the paper.

2. Formulation of the Game

The multi-agent game system is based on the communication network among agents, integrating a data storage module, navigation system, electric actuator, decision evaluation, and an updating system. Its main structure is shown in Figure 1. Each pursuer is equivalent to equipment in the network. It executes the policy through the electric actuator, and transmits position, speed, and other status information to the communication network from the navigation system. Agents can interact with the status via the network layer. All information is transmitted to the platform layer through the communication system, evaluated by the decision evaluation system, and new policies are generated through the decision updating system; the new policies are transmitted to the electric actuator of the agents via the communication network.

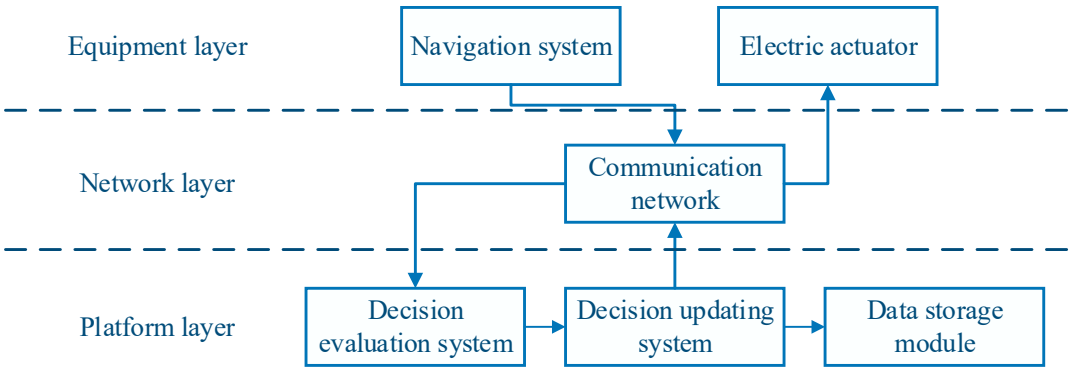


Figure 1. Structure of the MPE game system.

Consider a game system with a couple of pursuers and a single evader, and where each agent in the game system has its goal to achieve, which can be defined as follows:

Definition 1. *In an MPE game with multiple pursuers and a single evader, the evader tries to escape from being captured by every pursuer, while each pursuer tries to capture the single evader. The performance index of each pursuer is to minimize the distance between the evader and its own control energy, and the performance index of the evader is to maximize the distance between each pursuer and to minimize its own control energy.*

The conditions of both the pursuers and the evader change in real time; thus, the system is a multi-agent differential game. Here, the dynamics of each agent are expressed as a set of differential equations. Consider a team of N agents as the pursuers, each of which follows the dynamics:

$$\dot{x}_{pi} = Ax_{pi} + B_i u_{pi} \tag{1}$$

The dynamics of the single evader can be expressed as

$$\dot{x}_e = Ax_e + B_e u_e \tag{2}$$

where x_{pi}, u_{pi}, x_e, and u_e refer to the state variables and controls of the i-th pursuer and the evader, respectively. A, B_i, and B_e are the system matrices. Let x_{pi} represents the position and velocity of the i-th pursuer in different dimensions, respectively, and u_{pi} contains the accelerations of the i-th pursuer along those dimensions. Similarly, x_e and u_e are those of the single evader to fulfill the control model.

Let δ_i is the difference between the state variables of the pursuer i and the evader, which is expressed as

$$\delta_i = x_{pi} - x_e \tag{3}$$

For the multi-agent PE game problem we discussed, each pursuer tries to minimize the distance to the evader, while the evader tries to maximize the distance to every pursuer. Substitute Equations (1) and (2) into Equation (3) to get the time derivative of δ_i as

$$\dot{\delta}_i = A\delta_i + B_i u_{pi} - B_e u_e \tag{4}$$

For each element, there exists a performance index. We establish the integral performance index of agent i as

$$J_i(\delta_i, u_{pi}, u_e) = \frac{1}{2}\int_0^\infty (\delta_i^\top Q_i \delta_i + u_{pi}^\top R_{pi} u_{pi} - u_e^\top R_e u_e) d\tau \tag{5}$$

where Q_i is a positive-definite matrix and R_{pi} and R_e are two symmetric positive definite matrices. u_{pi} and u_e stand for the policies of the i-th pursuer and evader, respectively. $\delta_i^\top Q_i \delta_i$ is the weighted term of the relative state variable, which is used to restrict the relative position between the i-th pursuer and the evader. $u_{pi}^\top R_{pi} u_{pi}$ and $u_e^\top R_e u_e$ are the total amounts of energy consumption of the controls of the i-th pursuer and the evader, respectively, which are used to realize the constraints on the control variables.

Since each pursuer will participate in the PE game, which will affect the final policy of the evader, here, we define the overall index function J as the weighted sum of each index function J_i in Equation (5) for $i = 1, \ldots, N$,

$$J = \frac{1}{N}\sum_{i=1}^{N} J_i \tag{6}$$

The value of the game is evaluated as follows when the i-th pursuer and the evader employ certain policies:

$$V_i(\delta_i) = \frac{1}{2}\int_0^\infty (\delta_i^\top Q_i \delta_i + u_{pi}^\top R_{pi} u_{pi} - u_e^\top R_e u_e) d\tau \tag{7}$$

For pursuer i and the component of the evader, when both of which are acquiring the optimal policies, then we have:

$$V_i^*(\delta_i) = \min_{u_{pi}} \max_{u_e} J_i = \min_{u_{pi}} \max_{u_e} \frac{1}{2} \int_0^\infty (\delta_i^\top Q_i \delta_i + u_{pi}^\top R_{pi} u_{pi} - u_e^\top R_e u_e) d\tau \tag{8}$$

By summing N value functions in the game, the overall value function can be obtained as follows:

$$\begin{aligned} V^*(\delta) &= \min_{u_{p1},\ldots,u_{pN}} \max_{u_e} J_i \\ &= \min_{u_{p1},\ldots,u_{pN}} \max_{u_e} \tfrac{1}{2} \textstyle\int_0^\infty (\frac{1}{N} \sum_{i=1}^N \delta_i^\top Q_i \delta_i + \frac{1}{N} \sum_{i=1}^N u_{pi}^\top R_{pi} u_{pi} - u_e^\top R_e u_e) d\tau \end{aligned} \tag{9}$$

where δ is the set of $\delta_1, \ldots, \delta_N$.

The aim of the issue is to find out the appropriate control to satisfy Equation (9) for each agent. However, there are a couple of difficulties faced when calculating the numeral result of the control policies. In the actual system, the analytical solution of each individual strategy is difficult to obtain directly, or occupies too many computing resources and too much time, so the focus of this paper is to iteratively obtain the Nash equilibrium solution by using the idea of reinforcement learning.

The difficulty faced by this paper lies in solving the Nash equilibrium solution of multi-agent systems in pursuit–evasion games. We popularize the scenario of the zero-sum game and consider the scenario of multiple pursuers. We use the integral reinforcement learning method to solve the policy of each agent, as is shown in Section 4.

3. Solution of the MPE Game Problem

According to the physical model of the PE game problem, the Nash equilibrium solution to the problem is obtained by using the minimax principle. The capture conditions of the PE game are discussed by using the Lyapunov function method.

For multi-agent PE games, each individual's decision will affect the system. The Nash equilibrium solution of the game can be obtained by using the minmax principle, that is, the set of decisions where all parties reach the optimal situation at the same time.

The multi-agent game model is derived from the game of two agents. Among them, the differential game of two agents is a group of differential games, which is established based on bilateral optimal control theory. For the differential game of multiple agents, each index function corresponds to a group of optimization, the policy of which is obtained by the minmax principle. In the connected graph, the evader is linked with multiple pursuers, so it needs to consider the policies of each interconnected chasing pursuer. Suppose there is a game with a set of policies to reach the Nash equilibrium and the evader is linked with a number of N pursuers, then we call $(u_{p1}^*, \ldots, u_{pn}^*, u_e^*)$ the game-theoretic saddle point.

The differential expression in Equation (7) is equivalent to the Bellman equation of the game. According to Equations (1) and (2), the Bellman equation of pursuer i and the evader can be obtained as:

$$\begin{aligned} H(\ldots) &= \tfrac{1}{2} (\tfrac{1}{N} \sum_{i=1}^N \delta_i^\top Q_i \delta_i + \tfrac{1}{N} \sum_{i=1}^N u_{pi}^\top R_{pi} u_{pi} - u_e^\top R_e u_e) + \sum_{i=1}^N (\tfrac{\partial V}{\partial \delta_i})^\top \dot{\delta}_i \\ &= \tfrac{1}{2} (\tfrac{1}{N} \sum_{i=1}^N \delta_i^\top Q_i \delta_i + \tfrac{1}{N} \sum_{i=1}^N u_{pi}^\top R_{pi} u_{pi} - u_e^\top R_e u_e) + \sum_{i=1}^N (\tfrac{\partial V}{\partial \delta_i})^\top (A\delta_i + B_i u_{pi} - B_e u_e) \end{aligned} \tag{10}$$

where $H(\ldots)$ is the Hamilton function of the PE game, u_{pi} and u_e are the admissible control policies of the i-th pursuer and the evader, and ∇V stands for $\frac{\partial V}{\partial \delta_i}$.

In order to find the optimal policies of the game, according to the stationary condition of the minmax principle, the following equations should be satisfied:

$$\frac{\partial H}{\partial u_{pi}} = 0 \tag{11}$$

$$\frac{\partial H}{\partial u_{e}} = 0 \tag{12}$$

Moreover, the second derivative of the Hamilton function to the control of each element should satisfy the following conditions:

$$\frac{\partial^2 H}{\partial u_{pi}^2} > 0 \tag{13}$$

$$\frac{\partial^2 H}{\partial u_{e}^2} < 0 \tag{14}$$

Thus, the optimal policies of the pursuers and evader are found as

$$u_{pi}^* = -NR_{pi}^{-1}B_i^\top(\frac{\partial V^*}{\partial \delta_i}) \tag{15}$$

$$u_{e}^* = -NR_{e}^{-1}B_e^\top(\frac{\partial V^*}{\partial \delta_i}) \tag{16}$$

For infinite time invariant systems, the solution of the game is determined by (15) and (16), where the function V_i is the solution of the Hamilton–Jacobi–Isaacs (HJI) equation of the game as follows:

$$\frac{1}{2}(\frac{1}{N}\sum_{i=1}^{N}\delta_i^\top Q_i\delta_i + \frac{1}{N}\sum_{i=1}^{N}u_{pi}^\top R_{pi}u_{pi} - u_e^\top R_e u_e) + \sum_{i=1}^{N}(\frac{\partial V}{\partial \delta_i})^\top(A\delta_i + B_i u_{pi} - B_e u_e) = 0 \tag{17}$$

We need to further prove the above conclusions. That is, for the MPE game with multiple pursuers and a single evader, when the policy of each agent reaches Equations (15) and (16), the game reaches the Nash equilibrium.

Before proving this conclusion, we need to use the basic property of the Hamilton function in a multilateral optimal control problem, which is embodied in Lemma 1.

Lemma 1. *Let V^* satisfy the HJI Equation (17), making the Hamilton function $H(\delta, \nabla V^*, u_{p1}, \ldots, u_{pN}, u_e^*) = 0$. Then (10) becomes:*

$$\begin{aligned} H(\delta, \nabla V^*, u_{p1}, \ldots, u_{pN}, u_e \quad) = &\sum_{i=1}^{N}(\tfrac{\partial V^*}{\partial \delta_i})^\top(B_i(u_{pi} - u_{pi}^*) + B_e(u_e^* - u_e)) + \tfrac{1}{2N}\sum_{i=1}^{N}u_{pi}^\top R_{pi}u_{pi} \\ &-\tfrac{1}{2N}\sum_{i=1}^{N}u_{pi}^{*\top}R_{pi}u_{pi}^* + \tfrac{1}{2}u_e^{*\top}R_e u_e^* - \tfrac{1}{2}u_e^\top R_e u_e. \end{aligned} \tag{18}$$

Proof of Lemma 1. Upon adding and subtracting the terms $u_p^{*\top}R_p u_p^*$, $u_e^{*\top}R_e u_e^*$, $\nabla V^\top B u_p^*$, and $\nabla V^\top B u_e^*$, the Hamiltonian (10) yields:

$$\begin{aligned} H(\ldots) &= \sum_{i=1}^{N} \delta_i^\top Q_i \delta_i + \sum_{i=1}^{N} \left(\tfrac{\partial V}{\partial \delta_i}\right)^\top (A\delta_i + B_i u_{pi}^* - B_e u_e^*) + \tfrac{1}{2N} \sum_{i=1}^{N} u_{pi}^{*\top} R_{pi} u_{pi}^* - \tfrac{1}{2} u_e^{*\top} R_e u_e^* \\ &+ \sum_{i=1}^{N} \left(\tfrac{\partial V}{\partial \delta_i}\right)^\top (B_i(u_{pi} - u_{pi}^*) + B_e(u_e^* - u_e)) + \tfrac{1}{2N} \sum_{i=1}^{N} u_{pi}^\top R_{pi} u_{pi} - \tfrac{1}{2} u_e^\top R_e u_e \\ &- \tfrac{1}{2N} \sum_{i=1}^{N} u_{pi}^{*\top} R_{pi} u_{pi}^* + \tfrac{1}{2} u_e^{*\top} R_e u_e^* \\ &= H(\delta, \nabla V, u_{p1}^*, \ldots, u_{pN}^*, u_e^*) + \sum_{i=1}^{N} \left(\tfrac{\partial V}{\partial \delta_i}\right)^\top (B_i(u_{pi} - u_{pi}^*) + B_e(u_e^* - u_e)) \\ &+ \tfrac{1}{2N} \sum_{i=1}^{N} u_{pi}^\top R_{pi} u_{pi} - \tfrac{1}{2N} \sum_{i=1}^{N} u_{pi}^{*\top} R_{pi} u_{pi}^* + \tfrac{1}{2} u_e^{*\top} R_e u_e^* - \tfrac{1}{2} u_e^\top R_e u_e. \end{aligned} \tag{19}$$

When the value function V attains the Nash equilibrium value V^*, we have:

$$\begin{aligned} H(\nabla V^*, u_{p1}, \ldots, u_{pN}, u_e) &= H(\nabla V^*, u_{p1}^*, \ldots, u_{pN}^*, u_e^*) + \sum_{i=1}^{N} \left(\tfrac{\partial V^*}{\partial \delta_i}\right)^\top (B_i(u_{pi} - u_{pi}^*) + B_e(u_e^* - u_e)) \\ &+ \tfrac{1}{2N} \sum_{i=1}^{N} u_{pi}^\top R_{pi} u_{pi} - \tfrac{1}{2N} \sum_{i=1}^{N} u_{pi}^{*\top} R_{pi} u_{pi}^* + \tfrac{1}{2} u_e^{*\top} R_e u_e^* - \tfrac{1}{2} u_e^\top R_e u_e. \end{aligned} \tag{20}$$

From the HJI function (17), we can find that the Hamilton function of the game $H(\nabla V^*, u_{p1}^*, \ldots, u_{pN}^*, u_e^*) = 0$ when the value function attains the optimal value, which completes the proof. □

The Hamilton function can be transformed using Lemma 1. As the control variables u_{pi}, u_e, and the conditions at the Nash equilibrium $H(\nabla V^*, u_{p1}^*, \ldots, u_{pN}^*, u_e^*)$ are concluded, it is easier to prove the Nash equilibrium for the MPE problem mentioned in Theorem 1.

Theorem 1. *Consider the kinetic equations of vehicles (1) and (2) with the value function as given in (7). Let V^* be a positive definite smooth solution of the HJI Equation (17). Then, $(u_{p1}^*, \ldots, u_{pN}^*, u_e^*)$given by (15) and (16) becomes the game-theoretic saddle point and V^* becomes the Nash equilibrium value of the MPE game.*

Proof of Theorem 1. In order to prove that $(u_{p1}^*, \ldots, u_{pN}^*, u_e^*)$ is the game-theoretic saddle point of the game, we need to show that the best action for the evader to maximize the value function (15) is u_e^* when all the pursuers execute the policies as given in (21). On the other hand, the best action for the i-th pursuer to minimize the value function (16) is u_{pi}^* when the other agents execute the policy as given in (22), which indicates the following:

$$u_{pi}^* = \text{argmin} V_{u_{p1},\ldots,u_{pi},\ldots,u_{pN},u_e^*}(\delta(t)) \tag{21}$$

$$u_e^* = \text{argmax} V_{u_{p1},\ldots,u_{pN},u_e}(\delta(t)) \tag{22}$$

which are equivalent to:

$$V_{u_{p1}^*,\ldots,u_{pN}^*,u_e}(\delta(t)) \le V_{u_{p1}^*,\ldots,u_{pN}^*,u_e^*}(\delta(t)) \le V_{u_{p1}^*,\ldots,u_{pi},\ldots,u_{pN}^*,u_e^*}(\delta(t)) \tag{23}$$

where $V_{u_{pi},u_e}(\delta(t))$ is the corresponding solution of the Hamiltonian (18). Define $V(\delta(t_0))$ as the initial value of the game. Assume that a capture occurs in the interval $t \in [t_0, \infty]$, which implies $\lim_{x \to +\infty} V_{u_p,u_e}(\delta(t)) = 0$. Adding this item to Equation (7), we get:

$$V = \min_{u_{p1},\ldots,u_{pN}} \max_{u_e} \frac{1}{2} \int_0^\infty \left(\frac{1}{N} \sum_{i=1}^{N} \delta_i^\top Q_i \delta_i + \frac{1}{N} \sum_{i=1}^{N} u_{pi}^\top R_{pi} u_{pi} - u_e^\top R_e u_e\right) d\tau + \int_{t_0}^\infty \dot{V}_{u_{p1},\ldots,u_{pN},u_e} d\tau + V(\delta(t_0)) \tag{24}$$

From Equation (24), it is obvious that $V_{u^*_{p1},\ldots,u^*_{pN},u^*_e}(\delta(t)) = V^*(\delta(t_0))$. Upon using Lemma 1, (24) becomes:

$$\begin{aligned} V_{u_{p1},\ldots,u_{pN},u_e}(\delta(t_0)) &= \int_{t_0}^{\infty}(\sum_{i=1}^{N}(\frac{\partial V}{\partial \delta_i})^{\top}(B_i(u_{pi}-u^*_{pi})+B_e(u^*_e-u_e)) \\ &+\frac{1}{2N}\sum_{i=1}^{N}u_{pi}^{\top}R_{pi}u_{pi}-\frac{1}{2N}\sum_{i=1}^{N}u_{pi}^{*\top}R_{pi}u^*_{pi}+\frac{1}{2}u_e^{*\top}R_eu^*_e-\frac{1}{2}u_e^{\top}R_eu_e)d\tau+V(\delta(t_0)). \end{aligned} \tag{25}$$

Let $\varepsilon(V)$ be the integral term in Equation (25). In order to prove Equation (23), we just need to verify that $\varepsilon(V_{u_p,u^*_e}) \geq 0$ and $\varepsilon(V_{u^*_p,u_e}) \leq 0$. Using (25) we get:

$$\begin{aligned} \varepsilon\left(V_{u^*_{p1},\ldots,u^*_{pN},u_e}\right) &= \int_t^{\infty}(\sum_{i=1}^{N}(\frac{\partial V}{\partial \delta_i})^{\top}B_e(u^*_e-u_e)+\frac{1}{2}u_e^{*\top}R_eu^*_e-\frac{1}{2}u_e^{\top}R_eu_e)d\tau \\ &= \int_t^{\infty}-u_e^{*\top}R_e(u^*_e-u_e)+\frac{1}{2}u_e^{*\top}R_eu^*_e-\frac{1}{2}u_e^{\top}R_eu_e)d\tau \\ &= -\frac{1}{2}\int_t^{\infty}(u^*_e-u_e)^{\top}R_e(u^*_e-u_e)d\tau \leq 0 \end{aligned} \tag{26}$$

$$\begin{aligned} \varepsilon\left(V_{u^*_{p1},\ldots,u_{pi},\ldots,u^*_{pN},u^*_e}\right) &= \int_t^{\infty}((\frac{\partial V}{\partial \delta_i})^{\top}B_i(u_{pi}-u^*_{pi})-\frac{1}{2N}u_p^{*\top}R_pu^*_p+\frac{1}{2N}u_p^{\top}R_pu_p)d\tau \\ &= \int_t^{\infty}(-\frac{1}{N}u_{pi}^{*\top}R_{pi}(u_{pi}-u^*_{pi})-\frac{1}{2N}u_{pi}^{*\top}R_{pi}u^*_{pi}+\frac{1}{2N}u_{pi}^{\top}R_{pi}u_{pi})d\tau \\ &= \frac{1}{2N}\int_t^{\infty}(u^*_{pi}-u_{pi})^{\top}R_{pi}(u^*_{pi}-u_{pi})d\tau \geq 0, \end{aligned} \tag{27}$$

which completes the proof. □

Remark 1. *From Theorem 1, we know that when the game reaches the Nash equilibrium, the value function does not decrease no matter how the i-th pursuer unilaterally changes its policy. On the other hand, no matter how the single evader unilaterally changes its policy, the value function does not increase. As soon as the policy set of the game reaches the game-theoretic saddle point, any unilateral change in the policies by either agent is in contrary to its benefits, and the other side of the game reaps the reward in this process. Intuitively speaking, when the game reaches the Nash equilibrium, if any pursuer unilaterally changes its policy, the evader becomes more difficult to capture. However, the evader can be captured easily if it unilaterally changes the policy at the saddle point.*

So far, we can summarize the Algorithm 1 as follows:

Algorithm 1: The optimality of Nash equilibrium policies for each agent

Step 1: Obtain the system Nash equilibrium expression
The Hamiltonian function (10) is obtained according to the system parameters, and the expression of multi-agent Nash equilibrium policies (15) and (16) is obtained through the minimax conditions (11)~(14).
Step 2: Construct difference form Hamilton function
The Nash equilibrium function term (19) is constructed in the Hamiltonian function, and the differential Hamiltonian function (20) is obtained to facilitate the comparison of the properties of Nash equilibrium policies and other policies.
Step 3: Proof of Nash equilibrium of game
The rationality of inequality (23) is proved according to the difference method. By comparing the positive and negative of the integral term, i.e., Equations (26) and (27), it is found that the Nash equilibrium can realize the minmax strategy, which is the property of the value function (9).

The process of proving the optimality of Nash equilibrium is shown in Figure 2.

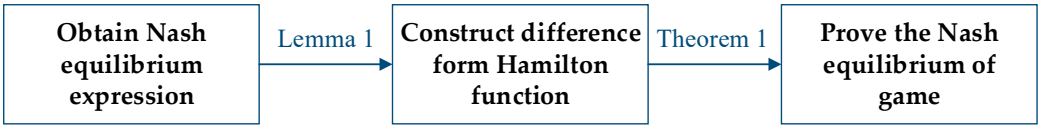


Figure 2. Proof of the optimality of Nash equilibrium policies for each agent.

In this problem, each pursuer shares states and control information, so there is a cooperative relationship between each pursuer, which affects the decisions of each agent. According to Equation (6), the value function V is composed of the sum of $V_1, \ldots, V_N$. Due to the coupling between individuals, $\nabla V_k = \sum_{i=1}^{N} \frac{\partial V_k}{\partial \delta_i}$, $i, k = 1, \ldots, N$, so we have:

$$\nabla V = \sum_{k=1}^{N} \nabla V_k = \sum_{k=1}^{N} \sum_{i=1}^{N} \frac{\partial V_k}{\partial \delta_i}. \tag{28}$$

When there is no communication between each pursuer, the game model is equivalent to N groups of single-pursuer and single-evader game problem, and there is no coupling between individuals, which means $\frac{\partial V_k}{\partial \delta_i} = 0$. If $k \neq i$, then the sum of $V_1, \ldots, V_N$ is meaningless, and V_k needs to be calculated separately. ∇V_k in Equation (28) changes to:

$$\nabla V_k = \frac{\partial V_k}{\partial \delta_k}. \tag{29}$$

As this paper considers the case where communication exists between each individual, coupling between agents is contained in the following parts.

In an MPE game problem, whether the pursuers can capture all the evaders or not is very noteworthy information. If so, the MPE problem is likely to become a finite-time game as an interception problem. Next, we demonstrate the conditions under which the capture scenario occurs in the game.

In Theorem 1, we assume that a finite-time capture occurs in the interval $t \in [t_0, +\infty]$ to ensure the existence of the Nash equilibrium. Theorem 2 gives the sufficient conditions for the occurrence of the capture.

Theorem 2. *Let the pursuers satisfy the dynamics as given by Equation (1) and the evader satisfy Equation (2). Moreover, let (15) and (16) be the control policies of the pursuers and the single evader, respectively, where function $V(\delta)$ is the solution of the HJI Equation (17). Then, a capture of the MPE game occurs in the sense that dynamic (4) is asymptotically stable.*

Proof of Theorem 2. In order to prove the property of the stability, we can take the value function as a Lyapunov function. Since $V(\delta)$ is the solution of the HJI Equation (15), and we can acquire that $V(\delta) \geq 0$ and $V(\delta(t_0)) = 0$, then the derivative of the function $\dot{V}(\delta)$ can be expressed as follows:

$$\begin{aligned} \dot{V} = \sum_{i=1}^{N} \left(\tfrac{\partial V}{\partial \delta_i}\right)^{\top} (A\delta_i + B_i u_{pi} - B_e u_e) &= -\tfrac{1}{2}\left(\tfrac{1}{N} \sum_{i=1}^{N} \delta_i^{\top} Q_i \delta_i + \tfrac{1}{N} \sum_{i=1}^{N} u_{pi}^{\top} R_{pi} u_{pi} - u_e^{\top} R_e u_e\right) \\ &= -\tfrac{1}{2N} \sum_{i=1}^{N} \delta_i^{\top} Q_i \delta_i - \tfrac{N}{2} \sum_{i=1}^{N} \left(\tfrac{\partial V}{\partial \delta_i}\right)^{\top} (B_i R_{pi}^{-1} B_i^{\top} - B_e R_e^{-1} B_e^{\top}) \tfrac{\partial V}{\partial \delta_i}. \end{aligned} \tag{30}$$

From Equation (30), the derivative of the Lyapunov function $\dot{V}$ keeps negative only if $B_i R_{pi} B_i^{\top} - B_e R_e^{-1} B_e^{\top} \geq 0$, $i = 1, \ldots, N$. That is, if game system (A, B_i) and (A, B_e) is stabilizable, $(A, \sqrt{Q_i})$ is observable, and $B_i R_{pi} B_i^{\top} - B_e R_e^{-1} B_e^{\top} \geq 0$ for $i = 1, \ldots, N$ holds, then the dynamic of the MPE game is asymptotically stable, and all pursuers have the potential to catch up with the evader. On the other hand, if $B_i R_{pi} B_i^{\top} - B_e R_e^{-1} B_e^{\top} < 0$, then the Lyapunov stability condition is not satisfied, and the state variables of system (4) may tend to diverge. The divergence of the distance between two agents will make it impossible for the capture to occur. At this time, the pursuers cannot capture the evader. □

Remark 2. *In particular, when $B_i = B_e = B$ for all $i = 1, \ldots, N$, it can be predicted that the distance between the pursuer i and the evader will approach 0 as time passes as $R_{pi}^{-1} - R_e^{-1}$ is positive. On the contrary, if $R_{pi}^{-1} - R_e^{-1}$ is not positive, pursuer i may not be able to catch the evader. In value function (7), $u_{pi}^{\top} R_p u_{pi}$ and $u_e^{\top} R_e u_e$ represent the total amounts of energy consumption of*

the control for pursuer i and the evader. For pursuer i and the evader, R_{pi} and R_e matrices represent their soft constraints on the control utility, which act as the control penalty. The larger the control energy weight of the evader or the smaller the control energy weight of the pursuer, the easier the capture scenario is to trigger.

4. Solution Using Adaptive Dynamic Programming

In Section 3, we obtain the Nash equilibrium policy of each agent in the MPE game. However, it is challenging to solve the accurate result of the policy for actual systems. For the numerical solution of the policy, Pontani and Conway used a genetic algorithm to calculate the off-line control policies of the agents in a zero-sum differential game [11]. However, for the online game problem of a continuous system, the off-line algorithm may not monitor the states and controls of all the agents. Therefore, based on the concept of reinforcement learning, this section solves the implementation policy through the policy iteration (PI) method. In addition, it is difficult to directly obtain the derivative of the value function with respect to the state quantity, which needs to be fitted by an approximate algorithm called the value function approximation (VFA). When the PI method loops to convergence, the Nash equilibrium policies of all agents are obtained.

4.1. Policy Iteration

Since the value function of the MPE game above has an integral form, the value function can be decomposed by dividing the upper and lower bounds of the integral. By forming the terms of integral reinforcement learning, the policy iteration (PI) method is executed to solve the game.

Define an infinite-horizon integral cost associated with the control input as:

$$V(\delta(t)) = \int_t^{\infty} \Gamma(\delta(\tau), u_{p1}(\tau), \ldots, u_{pN}(\tau), u_e(\tau)) d\tau \tag{31}$$

For $\Gamma(\delta(\tau), u_p(\tau), u_e(\tau)) = \delta^\top Q\delta + u_p^\top R_p u_p - u_e^\top R_e u_e$.

Selecting T as a time period, Equation (31) can be expanded as follows:

$$V(\delta(t)) = \int_t^{t+T} \Gamma d\tau + \int_{t+T}^{\infty} \Gamma d\tau = \int_t^{t+T} \Gamma d\tau + V(\delta(t+T)) \tag{32}$$

The integrand $\Gamma(\delta(\tau), u_{p1}(\tau), \ldots, u_{pN}(\tau), u_e(\tau))$ is known as the integral reinforcement of the pursuer in the time interval $[t, t+T]$. Note that T is not a state or control variable, but a hyper-parameter in the algorithm. Choosing different time intervals can affect the final solution and efficiency of the algorithm. Divide the whole time period of the game into multiple short time intervals. Assume that $[t, t+T]$ is the ith time interval. In this time interval, the pursuers and evader adopt policies as $u_{pi}^{(j)}$, $i = 1, \ldots, N$ and $u_e^{(j)}$, respectively. Then, Equation (32) becomes:

$$V^{(j)}(\delta(t)) = \int_t^{t+T} \Gamma(u_{p1}^{(j)}, \ldots, u_{pN}^{(j)}, u_e^{(j)}) d\tau + V^{(j)}(\delta(t+T)) \tag{33}$$

The controls for the next time interval can be obtained by using the following in (33):

$$\begin{aligned} u_{pi}^{(j+1)} &= -NR_{pi}^{-1} B_i^\top \nabla V^{(j)} \\ u_e^{(j+1)} &= -NR_e^{-1} B_e^\top \nabla V^{(j)}. \end{aligned} \tag{34}$$

Note that only the state information and control information of both sides is needed in the solving progress. System matrix A does not participate in the above operation. In actual applications, it is likely to solve the game of systems with unknown parameters, such as in the modeling of an unconventional aircraft. Therefore, this algorithm can also be used for obtaining an online solution effectively.

Equations (33) and (34) compose one cycle of policy iteration. The PI method can make the MPE game converge to the Nash equilibrium as the cycle continues. The following theorem proves the convergence of the PI method.

Theorem 3. *For the given MPE game, set u_{pi}^0 and u_e^0,$i = 1, \ldots, N$ as the admissible initial control policies. Then, the value $V(\delta)$ and policies u_{pi} and u_e,$i = 1, \ldots, N$ converge to the Nash equilibrium with $V^*(\delta)$, u_{pi}^* , and u_e^* , $i = 1, \ldots, N$, which optimize each player in the game.*

Proof of Theorem 3. Let $V_{u_{pi}^{(j)}, u_e^{(j)}}$ be the value when the ith pursuer and evader adopt the policies $(u_p^{(j)}, u_e^{(j)})$, and j is the iteration counter. Subtracting the value from (25) when the pursuers and the evader adopt $(u_{pi}^{(j+1)}, u_e^*)$ and $(u_{pi}^{(j)}, u_e^*)$, we get the following:

$$V_{u_{p1}^*, \ldots, u_{pi}^{(j+1)}, \ldots, u_{pN}^*, u_e^*}(\delta(t)) - V_{u_{p1}^*, \ldots, u_{pi}^{(j)}, \ldots, u_{pN}^*, u_e^*}(\delta(t)) = \int_t^\infty -(u_{pi}^{(j+1)} - u_{pi}^{(j)})^\top R_{pi}(u_{pi}^{(j+1)} - u_{pi}^{(j)}) \leq 0 \tag{35}$$

which means that the function group $V_{u_{pi}^{(j)}, u_e^*}(\delta(t))$ decreases monotonically. Similarly, we can see that the function group $V_{u_{pi}^*, u_e^{(j)}}(\delta(t))$ increases monotonically. According to Dini's theorem and the uniqueness of the value function (7), the value of the game $V_{u_{pi}^{(j)}, u_e^{(j)}}$ converges uniformly to $V_{u_{pi}^*, u_e^*}$.

In Equation (35), since $V_{u_{p1}^{(j)}, \ldots, u_{pN}^{(j)}, u_e^{(j)}}(\delta(t))$ is differentiable, the first derivative of δ, $\nabla V_{u_{p1}^{(j)}, \ldots, u_{pN}^{(j)}, u_e^{(j)}}$, converges to $\nabla V_{u_{p1}^{(*)}, \ldots, u_{pN}^{(*)}, u_e^{(*)}}$ as the value function converges. Thus, the policies of every player $\nabla V_{u_{p1}^{(*)}, \ldots, u_{pN}^{(*)}, u_e^{(*)}}$ converge to the Nash equilibrium policy set $(u_{p1}^{(*)}, \ldots, u_{pN}^{(*)}, u_e^{(*)})$ as the value function converges, which completes the proof. □

Based on Theorem 3, the continuous MPE game problem can be solved online via the PI method, which brings the solution to converge definitely to the Nash equilibrium as the iteration cycle operates.

Remark 3. *For the continuous time MPE problem, the use of the PI method to solve the Nash equilibrium parameters of the game will not lead to changes in its convergence, and the policy of each agent will eventually approach the analytical solution. In addition, the algorithm is still available for time-varying systems. If A changes suddenly, as long as the current controller stabilizes the new A, it is equivalent to solve the Nash equilibrium in the new state. Using Theorem 3, we can see that the game will converge to the Nash equilibrium corresponding to the final form of the system.*

Remark 4. *In the process of the PI method, the accurate information of system matrix A is not required when solving the policies of the agents, which means that for systems with some unknown specific structure, the policies can still be obtained and make the game converge to the Nash equilibrium. In the cycle of iteration, the state variables δ_t and δ_{t+T} and control policies $u_{pi}^{(j)}$ and $u_e^{(j)}$ for $i = 1, \ldots, N$ need to be known at every iteration step.*

4.2. Value Function Approximation

In the actual application scenario, solving the HJI equation is often a tough process analytically, for it might not have any analytical solution in an MPE game. Thus, we apply an approximative method to solve the equation by utilizing a structure to approximate the solution of the value function.

Assume that a finite set of basis functions $\phi_j(\delta)$ can be determined that approximate the value function V. Note that the functions are linearly independent, and the value

function is usually composed of polynomials of the state variables. The value function V can be approximately represented as

$$V(\delta(t)) = \sum_{k=1}^{L} w_k \phi_k(\delta) = w_L^\top \varphi_L(\delta) \tag{36}$$

where L is the number of basis functions in approximation, and $\varphi_L(\delta)$ is the vector of the corresponding basis function, which is composed of multiple state variables of the agents. w_L is a vector including unknown weights to be determined, and $w_k, (k = 1, \ldots, L)$ are the elements of vector w_L.

Using the VFA method for the cost function, the HJI equation can be expressed as

$$w_L^\top \varphi_L(\delta(t)) = \int_t^{t+T} \Gamma d\tau + w_L^\top \varphi_L(\delta(t+T)) \tag{37}$$

The focus is to solve the unknown weight parameter w_L in Equation (36). The initial value of the parameters is determined before iteration starts, which is likely to obtain residual error before the weight parameters converge to their analytical value. From Equation (37), the residual error can be defined as

$$\xi(\delta(t), T) = \int_t^{t+T} \Gamma d\tau + w_L^\top (\varphi_L(\delta(t+T)) - \varphi_L(\delta(t))) \tag{38}$$

The residual error indicates the difference between the actual weight parameters and the parameters in the solving process, which can be viewed as a temporal difference residual error for the game system.

In the VFA algorithm, we try to fit the value function through the basis function method and learn the weight parameters by building a neural network. At this time, the weight parameters are also called neural network parameters.

At each iteration step, in order to obtain the weight parameters $w_L^{(j)}$ in the VFA that approximates the value function $V^{(j)}$, the least-square method is used during each iteration step. When the game converges to the Nash equilibrium, the residual will converge to 0. In the process of convergence, the absolute value of the residuals will gradually decrease. Hence, the weight parameters $w_L^{(j)}$ of the VFA are adapted in a way to minimize the following quadratic integral residual:

$$S = \int \xi^2(\delta(t), T) d\delta \tag{39}$$

The quadratic integral residual S in Equation (39) represents the cumulative error of the algorithm, and it reaches the minimum value when its first partial derivative over the weight parameter $w_L^{(j)}$ comes to 0, that is

$$\int \frac{d\xi(\delta(t), T)}{dw_L^{(j)}} \xi(\delta(t), T) d\delta = 0 \tag{40}$$

Substituting (38) into (40) and assuming $\rho = \int_t^{t+T} \Gamma d\tau$, we have

$$w_L^{(j)} = \Phi^{-1} \Theta \tag{41}$$

where

$$\Phi = \int (\varphi_L(\delta(t+T)) - \varphi_L(\delta(t))) \cdot (\varphi_L(\delta(t+T)) - \varphi_L(\delta(t)))^\top d\delta$$

and

$$\Theta = \int (\varphi_L(\delta(t+T)) - \varphi_L(\delta(t))) \, \rho \, d\delta$$

When the residual function S reaches the minimum value, it means that the existing basis function can completely fit the value function by updating the neural network parameters obtained in Equation (41). According to Theorem 3, the value function approximated by the basis function can converge to V^*. Thus, the policies of every player $\nabla V_{u_{p1}^{(*)},\ldots,u_{pN}^{(*)},u_e^{(*)}}$ converge to the Nash equilibrium policy set $(u_{p1}^{(*)},\ldots,u_{pN}^{(*)},u_e^{(*)})$ as the value function converges. Since then, the ADP algorithm is used to avoid the difficulty of solving the composite HJI equation, and the Nash equilibrium policies consistent with the analytical method are obtained.

Remark 5. *The value function iteration is embedded in the policy iteration algorithm, i.e., n batches of least-square sample points need to be selected in each iteration process. The number of sample points n in each interval should be more than L (the number of the reinforcement learning parameters to be determined). Otherwise, the reinforcement learning parameter w_L cannot be regressed, resulting in the termination of the algorithm iteration. Moreover, the policies of both sides are no longer obtainable.*

5. Numerical Simulation

In this section, we simulate an MPE game with multiple pursuers and a single evader. In the simulation process, the system of agents adopts a class of second-order form, that is, the kind of dynamic model which takes acceleration as control. The trajectory and speed of the agents change in real time.

Consider the MPE game problem as:

$$\begin{cases} \dot{s}_{pix} = v_{pix} \\ \dot{s}_{piy} = v_{piy} \\ \dot{v}_{pix} = a_{pix} \\ \dot{v}_{piy} = a_{piy} \end{cases} \tag{42}$$

$$\begin{cases} \dot{s}_{ex} = v_{ex} \\ \dot{s}_{ey} = v_{ey} \\ \dot{v}_{ex} = a_{ex} \\ \dot{v}_{ey} = a_{ey} \end{cases} \tag{43}$$

where s_{pix}, s_{piy}, v_{pix}, and v_{piy} are the state variables of the i-th pursuer, which represent its position and velocity along two directions. Similarly, s_{ex}, s_{ey}, v_{ex}, and v_{ey} are the state variables of the single evader, which represent its position and velocity along those two directions. (a_{pix}, a_{piy}) and (a_{ex}, a_{ey}) are the accelerator couples of the i-th pursuer and the single evader, which serve as the policies of the i-th pursuer and the single evader, respectively.

Subtract the model of the i-th pursuer (42) and evader (43) to obtain the difference between their state variables $\delta = [d_x, \Delta v_x, d_y, \Delta v_y]$, where d_x and d_y are the distance projections in the X and Y directions, respectively. The dynamic of the subtracted model is:

$$\begin{cases} \dot{d}_{ix} = \Delta v_{ix} \\ \Delta\dot{v}_{ix} = a_{pix} - a_{ex} \\ \dot{d}_{iy} = \Delta v_{iy} \\ \Delta\dot{v}_{iy} = a_{piy} - a_{ey} \end{cases} \tag{44}$$

The distance between the two agents is expressed as a function of the difference of the state variables of Equation (44) as:

$$d = \sqrt{(s_{pix} - s_{ex})^2 + (s_{piy} - s_{ey})^2} = \sqrt{d_{ix}^2 + d_{iy}^2}$$

In addition, in order to confirm whether the pursuers can capture the evader, set the capture radius as l. When the distance between two agents is less than l, then a successful capture takes place, and the MPE game comes to an end.

Since the benefits of agents are determined by the distance between them, and the velocity is irrelevant, matrix Q in the value function (7) is set as $Q = diag(1, 1, 0, 0)$.

The set of the approximation basis functions is composed of the Kronecker product quadratic polynomial elements $\{\delta_i * \delta_j\}_{i,j=1,2,3,4}$. To reduce the computational cost of VFA, eliminate the combinations where there is no coupling relationship between state variables and construct the basis function group $\varphi_{iL}(\delta) = [\delta_{i1}^2, \delta_{i1}\delta_{i3}, \delta_{i2}^2, \delta_{i2}\delta_{i4}, \delta_{i3}^2, \delta_{i4}^2]$.

The listed initial values are considered in the following simulation experiment: $x_{p10} = [3; 1; 2; 0]$, $x_{p20} = [10; 0; 0; 3]$; $x_{p30} = [2; 10; 3; 5]$, $x_{e0} = [8; 8; 1; 2]$, $R_{p1} = R_{p2} = R_{p3} = 0.1$, $R_e = 1$, $a_{p10} = a_{p20} = a_{p30} = 0$, $a_{e0} = 0$, $w_0 = [1.5; 1.2; 1.5; 1.5; 1.5; 1.2]$.

Set the capture radius as $l = 0.05\ m$ to start the MPE game simulation, and the trajectories of the agents vary as in Figure 3. Table 1 shows the state of agents at the end of the game.

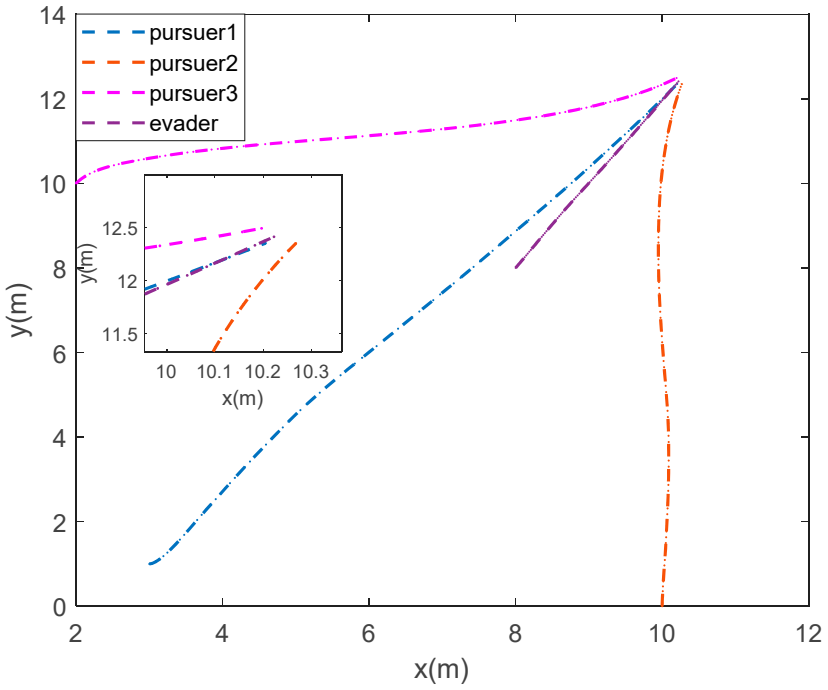


Figure 3. Trajectories of the agents in MPE game from beginning to capture.

Table 1. State variables of each agent in the game.

	Pursuer 1	**Pursuer 2**	**Pursuer 3**	**Evader**
Coordinate (m)	(10.2406, 12.4123)	(10.2820, 12.4131)	(10.2355, 12.5222)	(10.2419, 12.4489)
Distance (m)	0.0366	0.0538	0.0735	-

After the game starts, the evader accelerates to the direction where no pursuer exists, and the three pursuers adjust their policies according to the position of the evader. The evader can also adjust its policy to avoid being captured, but it is still captured by pursuer 1 due to greater energy constraints.

As the three pursuers have the same performance and follow value function (8), they can maintain good coordination in pursuit and finally capture the evader. The evader takes all factors of the state of the three pursuers into consideration and makes the decision to accelerate away from the directions of the pursuers.

In the process of the game, the gradient of the distance becomes steeper first and then gentler, which indicates that the pursuers narrow the distance as much as possible in the beginning, but in the later stage, due to the reduction of the distance, their energy control factors gradually occupy the main place. The agents realize the decision of pursuing and evading under the condition that their energy consumptions are as small as possible.

The capture happens at $t_c = 4.12\ s$, when the coordinates of pursuer 1 and evader are $(10.2406, 12.4123)$ and $(10.2419, 12.4489)$, respectively. The distance between pursuer 1 and evader is demonstrated in Figure 4.

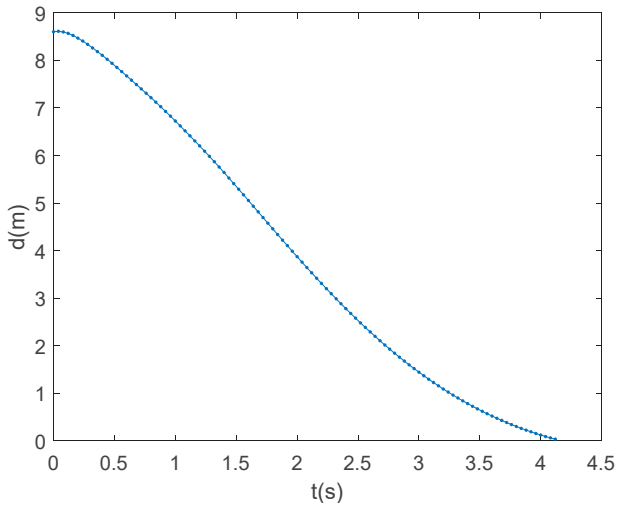
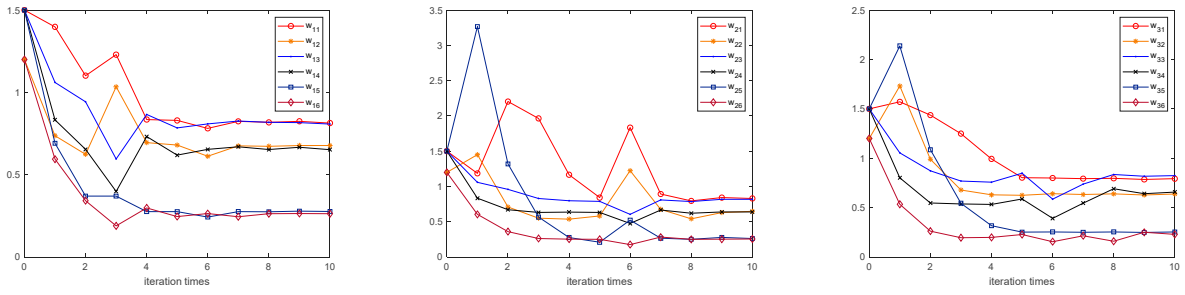


Figure 4. Distance between pursuer 1 and the evader.

As the iteration goes, the neural network parameters converge as the games reach the Nash equilibrium, which means the policies of both the pursuers and the evader convergence to the optimal value. The distance between pursuer 2 and evader at the time of capture is 0.0366 m. Note that the control policies of the agents are updated at each interval $T = 0.4\ s$.

From the beginning of the game to when the capture happens, 10 cycles of the policy iteration are completed. During this period, the parameters of the neural network converge gradually, and usually become stable at some fixed values from the fourth iteration cycle onward. We verify the effectiveness of the algorithm by obtaining the analytical solution of the network parameters, which can be seen in Figure 5. The coordinate and time of capture is shown in Table 2.



(**a**) (**b**) (**c**)

Figure 5. Variation of neural network parameters with the number of iterations. (**a**) neural network parameters w_{1i}, $i = 1, \cdots, 6$; (**b**) neural network parameters w_{2i}, $i = 1, \cdots, 6$; (**c**) neural network parameters w_{3i}, $i = 1, \cdots, 6$.

Table 2. Capture conditions of the MPE game when time interval is set as T = 0.4 s and T = 0.2 s.

	Capture Time (s)	Capture Coordinate (m)	Travel Distance of Evader (m)
T = 0.4 s	4.12	(10.2419, 12.4489)	4.9818
T = 0.2 s	3.84	(10.1025, 12.1622)	4.6631

By contrast, as long as the conditions in Theorem 3 are met, different initial values may affect the speed and pattern of the convergence of neural network parameters, but they will eventually converge to the analytical values.

The selection of the time interval T for each iteration step may also influence the solution of the MPE game. The selection of the iteration period is mainly based on the computing performance of the agents. Now we choose a different iteration time interval as $T = 0.2\ s$ to recompute the game problem. Keeping other initial states and parameters of the agents unchanged, we conduct the MPE game and get the trajectories between the pursuers and evader as illustrated in Figure 6, and the distances as shown in Figure 7.

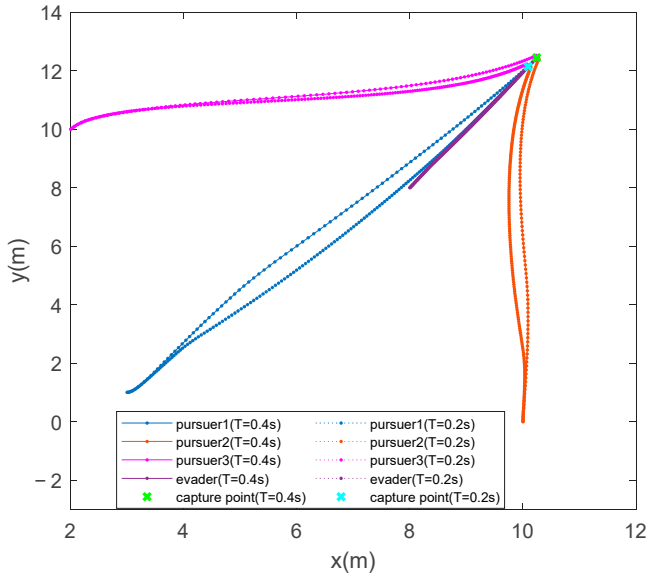


Figure 6. Trajectories of the agents in MPE game when time interval is set as T = 0.4 s and T = 0.2 s.

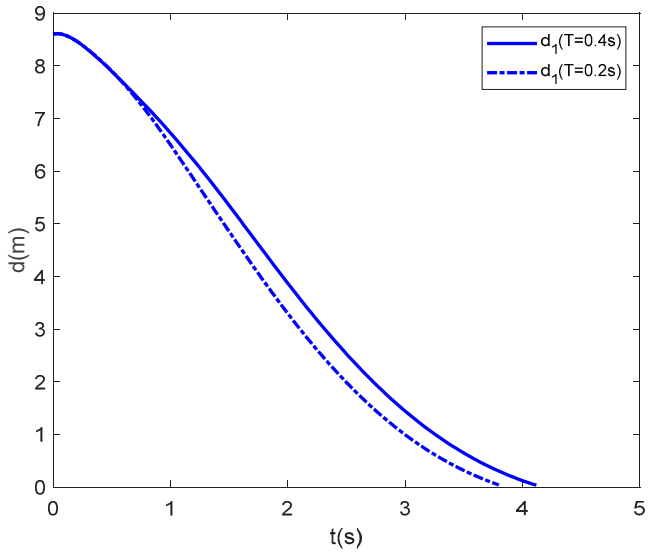


Figure 7. Distance between pursuer 1 and evader when time interval is set as T = 0.4 s and T = 0.2 s.

We can see in Figure 6 that a shorter iteration time interval brings closer a result of reaching the Nash equilibrium. However, the shorter the iteration period, the more iterations are needed in the whole game process, which means that the calculation cost will increase with the number of iterations, though the accuracy will be better at this time.

When a shorter time interval of iterations is selected, the pursuers can catch up with the evader faster. This is mainly because the evader in the simulation has weaker maneuverability, and its policy takes minor changes as the time interval varies. For the pursuers with strong maneuverability, the improvement is obvious, so the capture distance is nearer and the capture time is shorter. A shorter iteration time interval means that the frequency of policy updates increases, and the VFA algorithm is implemented in a shorter time, which can complete optimization faster. In general, while considering the computing performance of the agents, refining the time interval of iterations can enable agents to obtain Nash equilibrium policies faster.

In order to test the stability of the algorithm, we now consider the possible errors in agents. If an error occurs for pursuer 2, which means pursuer 2 cannot receive the information from others to execute the policy obtained by the algorithm, then it can only execute the initially defined policy, while pursuer 1 and pursuer 3 are in a normal state. By executing the above simulation, the trajectory shown in Figure 8 can be obtained.

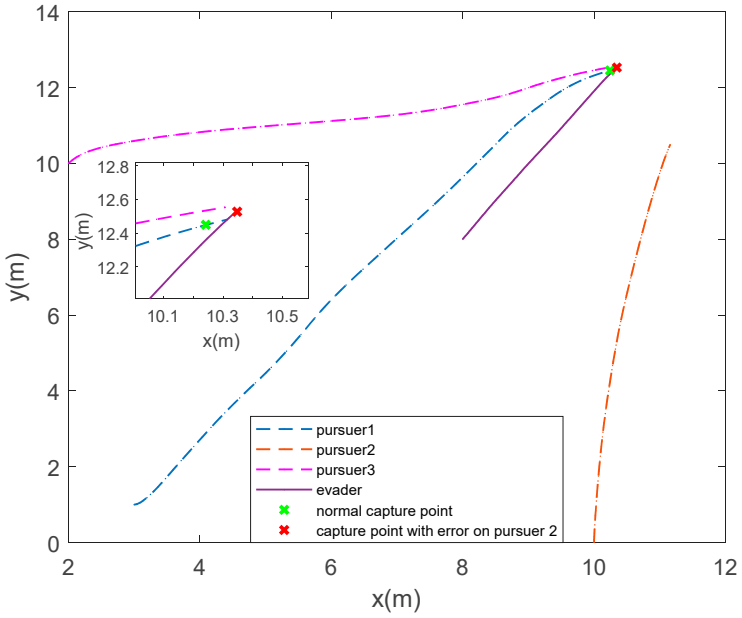


Figure 8. Trajectories of the agents in MPE game when an error occurs for pursuer 2.

It can be seen that when an error occurs for pursuer 2, the other pursuers can still capture the evader. At this time, the capture location is slightly farther than normal, and the capture time is slightly longer at 4.32 s. The distance between the three pursuers and the evader is shown in Figure 9, and the distances when there is no error for the pursuers are also shown as a comparison. The capture conditions are shown in Table 3.

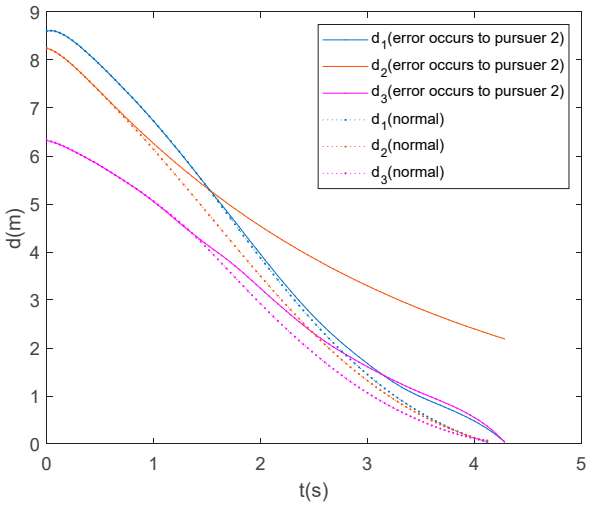


Figure 9. Distance between three pursuers and an evader in normal operation and in the case when pursuer 2 fails.

Table 3. Capture conditions of normal status and in the case when an error occurs for pursuer 2.

	Capture Time (s)	Capture Coordinate (m)	Travel Distance of Evader (m)
Normal	4.12	(10.2419, 12.4489)	4.9818
Error occurs for pursuer 2	4.32	(10.3427, 12.5269)	5.0972

Therefore, when there is an error on pursuer 2, the pursuers do not strictly implement the Nash equilibrium policies at this time, so the capture requires a longer distance, and the capture time is increased slightly compared with the normal state. Pursuers 1 and 3 try their best to achieve their optimal policies, while chaser 2 gradually leaves the game due to its error, which also confirms that the Nash equilibrium policies are optimal for each agent in the MPE game.

This simulation experiment evaluated the sensitivity of the initial value to the algorithm, and found that the change in sampling interval would affect the convergence of the neural network parameters. If the sampling interval is too short, it may lead to over-fitting, which affects the accuracy of the algorithm. In addition, the selection of the initial value of the state does not affect the stability of the algorithm, and the game will still gradually converge to the Nash equilibrium in the process of iteration.

6. Conclusions

The solution of an MPE game with networked multiple-pursuer and single-evader is discussed in this paper, and the expression of the policy of each agent is obtained by using the minmax strategy. It is proved that when the MPE game reaches the Nash equilibrium, the policy of each agent will converge to its Nash equilibrium policy, i.e., the optimal policy of the agent. The adaptive dynamic programming method is used for online policy iteration, and the VFA is adopted, utilizing the data provided through the IoT system to avert the difficulties in solving the complicated HJI equation. It was shown that the game reaches the Nash equilibrium condition after multiple iterations.

In this paper, the online ADP method is used to solve the MPE game problem. For a game with an integral value function, which is similar to Equation (7), the PI method proposed in Section 4 can be used to find its solution. However, for game problems with terminal or compounded value functions, the ADP method is difficult to solve, which limits the use of such methods. In addition, this paper did not take into account the case of complex constraints in the game process, and subsequent research should focus on the constraints of state variables and control variables.

In the future, we will consider the pursuit–evasion game scenarios of more complicated systems. It is worth studying scenarios when there are more evaders and pursuers in one game system. In addition, the algorithm still has room for improvement with regard to computing large-scale neural network parameters.

Author Contributions: Conceptualization, B.H. and C.H.; methodology, Z.G.; software, Z.G.; validation, Z.G., X.Z. and W.K.; formal analysis, Z.G.; investigation, Z.G.; resources, X.Z.; data curation, W.K.; writing—original draft preparation, Z.G.; writing—review and editing, Z.G., C.H. and W.K.; visualization, Z.G.; supervision, W.K. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Liu, F.; Dong, X.; Li, Q.; Ren, Z. Robust multi-agent differential games with application to cooperative guidance. *Aerosp. Sci. Technol.* **2021**, *111*, 106568. [CrossRef]
2. Makkapati, V.R.; Sun, W.; Tsiotras, P. Optimal evading strategies for two-pursuer/one-evader problems. *J. Guid. Control Dyn.* **2018**, *41*, 851–862. [CrossRef]
3. Zhang, H.; Ren, H.; Mu, Y.; Han, J. Optimal consensus control design for multiagent systems with multiple time delay using adaptive dynamic programming. *IEEE Trans. Cybern.* **2021**. [CrossRef] [PubMed]
4. Yuan, Z.; Wu, T.; Wang, Q.; Yang, Y.; Li, L.; Zhang, L. T3omvp: A transformer-based time and team reinforcement learning scheme for observation-constrained multi-vehicle pursuit in urban area. *Electronics* **2022**, *11*, 1339. [CrossRef]
5. Shi, M.D.Y.; Sun, Z. Satellite proximate pursuit-evasion game with different thrust configurations. *Aerosp. Sci. Technol.* **2020**, *99*, 105715.1–105715.10.
6. Isaacs, R. *Games of Pursuit*; Rand: Santa Monica, CA, USA, 1951.
7. Li, C.; Li, S.; Zhang, A.; Yang, L.; Zio, E.; Pecht, M.; Gryllias, K. A Siamese hybrid neural network framework for few-shot fault diagnosis of fixed-wing unmanned aerial vehicles. *J. Comput. Des. Eng.* **2022**, *9*, 1511–1524. [CrossRef]
8. Shima, T. Optimal cooperative pursuit and evasion strategies against a homing missile. *J. Guid. Control Dyn.* **2011**, *34*, 414–425. [CrossRef]
9. Li, D.; Cruz, J.; Chen, G.; Kwan, C.; Chang, M.-H. A Hierarchical Approach to Multi-Player Pursuit-Evasion Differential Games. In Proceedings of the 44th IEEE Conference on Decision and Control, 2005 and 2005 European Control Conference, CDC-ECC '05, Seville, Spain, 15 December 2005.
10. Friedman, A. *Differential Games*; Courier Corporation: Chicago, IL, USA, 2013.
11. Pontani, M.; Conway, B.A. Optimal interception of evasive missile warheads: Numerical solution of the differential game. *J. Guid. Control Dyn.* **2008**, *31*, 1111–1122. [CrossRef]
12. Cappello, D.M.T. Distributed differential games for control of multi-agent systems. *IEEE Trans. Control Netw. Syst.* **2022**, *9*, 635–646. [CrossRef]
13. Peng, B.; Stancu, A.; Dang, S.; Ding, Z. Differential graphical games for constrained autonomous vehicles based on viability theory. *IEEE Trans. Cybern.* **2022**, *52*, 8897–8910. [CrossRef]
14. Faruqi, F.A. *Differential Game Theory with Applications to Missiles and Autonomous Systems Guidance*; John Wiley & Sons: Hoboken, NJ, USA, 2017.
15. Lopez, V.G.; Lewis, F.L.; Wan, Y.; Sanchez, E.N.; Fan, L. Solutions for multiagent pursuit-evasion games on communication graphs: Finite-time capture and asymptotic behaviors. *IEEE Trans. Autom. Control* **2020**, *65*, 1911–1923. [CrossRef]
16. Zhang, Z.; Yan, W.; Li, H. Distributed optimal control for linear multi-agent systems on general digraphs. *IEEE Trans. Autom. Control* **2020**, *66*, 322–328. [CrossRef]
17. Wang, Y.; Dong, L.; Sun, C. Cooperative control for multi-player pursuit-evasion games with reinforcement learning. *Neurocomputing* **2020**, *412*, 101–114. [CrossRef]
18. Talebi, S.P.; Werner, S. Distributed kalman filtering and control through embedded average consensus information fusion. *IEEE Trans. Autom. Control* **2019**, *64*, 4396–4403. [CrossRef]
19. Ansart, A.; Juang, J.-C.; Ramachandran, K.G. Robust formation maintenance methods under general topology pursuit of multi-agents systems. *Electronics* **2021**, *10*, 1970. [CrossRef]
20. Li, C.; Li, S.; Zhang, A.; He, Q.; Liao, Z.; Hu, J. Meta-learning for few-shot bearing fault diagnosis under complex working conditions. *Neurocomputing* **2021**, *439*, 197–211. [CrossRef]

21. Murray, J.J.; Cox, C.J.; Lendaris, G.G.; Saeks, R. Adaptive dynamic programming. *IEEE Trans. Syst. Man Cybern. Part C* **2002**, *32*, 140–153. [CrossRef]
22. Abu-Khalaf, M.; Lewis, F.L. Nearly optimal control laws for nonlinear systems with saturating actuators using a neural network hjb approach. *Automatica* **2005**, *41*, 779–791. [CrossRef]
23. Vrabie, D. Online Adaptive Optimal Control for Continuous-Time Systems. Ph.D. Thesis, University of Texas at Arlington, Arlington, TX, USA, 2010.
24. Vrabie, D.; Lewis, F. Neural network approach to continuous-time direct adaptive optimal control for partially unknown nonlinear systems. *Neural Netw.* **2009**, *22*, 237–246. [CrossRef]
25. Vrabie, D.; Lewis, F. Adaptive dynamic programming algorithm for finding online the equilibrium solution of the two player zero-sum differential game. In Proceedings of the 2010 International Joint Conference on Neural Networks (IJCNN), Barcelona, Spain, 18–23 July 2010; pp. 1–8.
26. Kartal, Y.; Subbarao, K.; Dogan, A.; Lewis, F. Optimal game theoretic solution of the pursuit-evasion intercept problem using on-policy reinforcement learning. *Int. J. Robust Nonlinear Control* **2021**, *31*, 7886–7903. [CrossRef]
27. Wei, Q.; Liu, D. Adaptive dynamic programming for optimal tracking control of unknown nonlinear systems with application to coal gasification. *IEEE Trans. Autom. Sci. Eng.* **2014**, *11*, 1020–1036. [CrossRef]
28. Zw, A.; Qw, B.; Dl, C. Event-triggered adaptive dynamic programming for discrete-time multi-player games. *Inf. Sci.* **2020**, *506*, 457–470.
29. Zhang, H.; Zhang, J.; Yang, G.-H.; Luo, Y. Leader-based optimal coordination control for the consensus problem of multiagent differential games via fuzzy adaptive dynamic programming. *IEEE Trans. Fuzzy Syst.* **2015**, *23*, 152–163. [CrossRef]
30. Karg, P.; Koepf, F.; Braun, C.A.; Hohmann, S. Excitation for adaptive optimal control of nonlinear systems in differential games. *IEEE Trans. Autom. Control* **2022**. [CrossRef]
31. Li, Y.; Tang, Y.; Zhang, R.; Li, N. Distributed reinforcement learning for decentralized linear quadratic control: A derivative-free policy optimization approach. *IEEE Trans. Autom. Control* **2021**. [CrossRef]