mathematics

MDPI

Article

Optimal Asymptotic Tracking Control for Nonzero-Sum Differential Game Systems with Unknown Drift Dynamics via Integral Reinforcement Learning

Chonglin Jing [1], Chaoli Wang [1,*], Hongkai Song [2], Yibo Shi [1] and Longyan Hao [1]

[1] Department of Control Science and Engineering, University of Shanghai for Science and Technology, Shanghai 200093, China; 191550045@st.usst.edu.cn (C.J.); 221240072@st.usst.edu.cn (Y.S.); 191550063@st.usst.edu.cn (L.H.)

[2] Equipment Assets Management Office, Shanghai Jian Qiao University, Shanghai 201306, China; 19116@gench.edu.cn

* Correspondence: clwang@usst.edu.cn

Abstract: This paper employs an integral reinforcement learning (IRL) method to investigate the optimal tracking control problem (OTCP) for nonlinear nonzero-sum (NZS) differential game systems with unknown drift dynamics. Unlike existing methods, which can only bound the tracking error, the proposed approach ensures that the tracking error asymptotically converges to zero. This study begins by constructing an augmented system using the tracking error and reference signal, transforming the original OTCP into solving the coupled Hamilton–Jacobi (HJ) equation of the augmented system. Because the HJ equation contains unknown drift dynamics and cannot be directly solved, the IRL method is utilized to convert the HJ equation into an equivalent equation without unknown drift dynamics. To solve this equation, a critic neural network (NN) is employed to approximate the complex value function based on the tracking error and reference information data. For the unknown NN weights, the least squares (LS) method is used to design an estimation law, and the convergence of the weight estimation error is subsequently proven. The approximate solution of optimal control converges to the Nash equilibrium, and the tracking error asymptotically converges to zero in the closed system. Finally, we validate the effectiveness of the proposed method in this paper based on MATLAB using the ode45 method and least squares method to execute Algorithm 2.

Keywords: nonzero-sum games; optimal asymptotic tracking control; integral reinforcement learning; neural network

MSC: 93C10

Citation: Jing, C.; Wang, C.; Song, H.; Shi, Y.; Hao, L. Optimal Asymptotic Tracking Control for Nonzero-Sum Differential Game Systems with Unknown Drift Dynamics via Integral Reinforcement Learning. *Mathematics* **2024**, *12*, 2555. https://doi.org/10.3390/math12162555

Academic Editor: António Lopes

Received: 9 July 2024
Revised: 8 August 2024
Accepted: 16 August 2024
Published: 18 August 2024

1. Introduction

In the processes of manufacturing, military action, economic activities, and other purposeful human activities, it is necessary to apply a certain control to a controlled system and process to make a certain performance index reach the optimal value [1,2]; such a control effect is called optimal control, which is the most basic and core subject of modern control theory. The central issue is determining how to select a control law based on the system's dynamics to ensure the system operates according to specified technical requirements, thereby optimizing a particular performance index of the system in a defined sense [3]. For example, the use of the minimum amount of fuel or the minimum time to accurately launch space rockets and satellites into the predetermined orbit is a typical optimal control problem (OCP). In the past few decades, optimal control has received much research attention and has been widely used in the fields of aerospace [4], industrial production [5], and power systems [6].

The OCP of linear systems or nonlinear systems can be solved by constructing and solving the Riccati equation or the Hamilton–Jacobi–Bellman (HJB) equation [7]. However, the HJB equation is a nonlinear partial differential equation (NPDE) that is extremely challenging to solve analytically due to the "curse of dimensionality" as the dimensionality increases. Therefore, many scholars have developed an adaptive dynamic programming (ADP) technique that can solve the HJB equation [8,9]. In [10], the convergence and error bound analysis of value iteration (VI) ADP for continuous-time (CT) nonlinear systems were studied. However, most of the previously developed OCP discussed above is for systems affected by a single input parameter or a single agent. In fact, not only are multi-agent systems attracting attention from academics [11–13], but many practical systems are controlled by multi-input controllers, such as micro smart grid systems [14] and wireless communication systems [15], where each control input can be thought of as a player, and each player minimizes its own cost functions by influencing the system state. In this case, each player's optimal problem is coupled to the other players' optimal problems; therefore, the optimal solution does not exist in the general sense, which promotes the formulation of alternative optimality criteria.

For these multi-input systems, game theory provides an approach to a solution [16–18]. Nash equilibrium refers to a combination strategy. The combination strategy consists of the optimal strategy of all players; that is, under the condition of a given strategy of the other players, no individually motivated players choose other strategies, so no one is motivated to break this equilibrium [19]. Therefore, in some game-based control methods, the Nash equilibrium is often used to provide the concept of solutions.

Game theory has been successful in the simulation of strategic behavior in which each player's outcome depends on their own actions and those of all the other players. Each player influences the state of the system by selecting its own control policy to minimize its own predetermined performance goals independent of the other players. Differential games are an important field of game theory and have been used in different fields [20–22]. Differential games can be classified into zero-sum games, cooperative games, and NZS games based on the different tasks and roles of the participants. The objective of NZS games is to find a set of optimal control strategies that minimize the individual performance index function and ensure the stability of the NZS game systems, ultimately producing a Nash equilibrium. The Nash equilibrium can be obtained by solving coupled HJ equations [23]. However, the HJ equation is also an NPDE.

Recently, numerous scholars have investigated approximate dynamic programming (ADP) and reinforcement learning (RL) using an NN to approximate the Nash equilibrium [24–27]. RL can be classified into model-free RL and model-based RL based on the dynamic model of the system. The difference between the two approaches is whether a system model is required in the solution process. For model-based RL, Werbos [24] was the first to propose the use of ADP to tackle the discrete-time OCP, including two algorithms, VI and policy iteration (PI). However, compared to the PI algorithm, the convergence speed of the VI algorithm is slower, and the control strategy obtained at each iteration cannot ensure system stability. In [25], an online critical NN weight-tuning algorithm combining PI and recursive LS is proposed to solve the optimal control problem for players in nonlinear systems with nonzero-sum games. In [26], Zhang proposed a single-layer critic NN instead of a dual critic–actor NN, which solves the Nash equilibrium of NZS game systems. Vrabie [27] proposed an IRL method to solve the HJB equation with unknown drift dynamics. The IRL method is based on the integration time interval, PI technique, and RL concept to obtain the value function and has become a common method for solving the HJB equation. However, these methods still need to assume some knowledge of the model. This has motivated the development of model-free learning design methods.

Model-free RL can be classified into two categories: identifier-based RL and data-based RL. For identifier-based RL, Liu [28] proposed a critic–identifier structure to tackle the OCP for NZS games with completely unknown dynamics. In this approach, an identifier NN and a critic NN were used for approximating the unknown dynamics system

and value function, respectively. However, identifier training is usually time-consuming and inevitably introduces harmful identifier errors. Data-based RL methods are used to solve discrete-time nonlinear NZS game systems [29,30] and CT nonlinear NZS games systems [31–33]. Compared to the identifier-based RL method, this method avoids the introduction of identification error.

To our best knowledge, previous studies have focused on the regulation problem, and there have been few studies on the OTCP of NZS game systems. However, for practical systems, it is common to have the state or output of the system trace a given reference (desired) signal. For the OTCP, the traditional approach entails a two-stage process: optimal feedback tracking control and steady-state control [34]. To avoid such a classic two-step control design and reduce the computational cost, we will tackle the OTCP through an augmented system that only needs a one-step design. Currently, the conventional approach to solving the OTCP involves constructing an augmented system. This transforms the original OTCP into a related optimal regulation problem, which is subsequently tackled using the existing methods for such problems. The solution to the OTCP, namely, the Nash equilibrium, can thus be obtained through this process. In [35], an identifier–critic NN based on RL and NZS game theory was proposed to address the OTCP for nonlinear multi-input systems. However, in that paper, they used NN identification, which inevitably introduced identification errors, and the discount factor was not considered in its value function. Wen [36] solved the OTCP for discrete-time linear two-player NZS game systems by using model-free RL, in which the value function takes the discount factor into account. In [37], a new adaptive critic design was proposed to approximate the online Nash equilibrium solution for the robust trajectory tracking control of NZS games for continuous-time uncertain nonlinear systems. Zhao [38] solved the OTCP of NZS games of nonlinear CT systems through RL. However, in that paper, the tracking error is bounded, which is not ideal. In this paper, an offline IRL algorithm based on a single-layer critic NN is proposed to address the OTCP of N-player NZS games with nonlinear CT systems.

Compared to the existing literature, the innovations of this paper are primarily reflected in the following aspects:

1. To the best of our knowledge, no offline learning algorithm has been used to tackle the OTCP of nonlinear CT NZS differential game systems.
2. In this paper, the discount factor is considered in the cost function, which relaxes the requirement of the reference signal and does not need to require the reference signal to be an asymptotically stable signal.
3. In this paper, only the critic NN is considered to avoid the identification errors and to reduce the computational burden.
4. The offline IRL algorithm designed in this paper enables the weight error of the NN to converge to zero and the approximate solution to converge to a Nash equilibrium. In addition, the stability of the tracking error in the closed system is asymptotically guaranteed.

The subsequent sections of this paper will proceed as follows. In Section 2, an augmented system is developed to convert the OTCP into an optimal regulation problem, and a model-based PI algorithm is introduced. In Section 3, an IRL technique is proposed to approximate the value function, and the equivalence between the proposed method and this model-based policy iteration is proven. Section 4 presents the offline iterative learning algorithm and proves its convergence. Section 5 provides a simulation example. And Section 6 concludes this paper.

The following notations will be used throughout this paper:

Symbols	Meaning of Symbols
$\mathbb{R}$	real number set
$\mathbb{R}^n$	n-dimensional vector
$\mathbb{R}^{n\times m}$	the set of real $n \times m$ matrices
∇	gradient operator
$\lvert\cdot\rvert$	absolute value
$\lVert\cdot\rVert$	2-norm of a matrix or vector
sup	supremum
$C^1(\Omega)$	a function space on Ω with continuous first derivatives

2. Preliminaries

2.1. Problem Description

A class of nonlinear CT NZS differential game systems consisting of N-players is given by

$$\dot{x}(t) = f(x(t)) + \sum_{j=1}^{N} g_j(x(t))u_j(t) \tag{1}$$

where $u_j \in \mathbb{R}^{m_j}$ is the control input for player j, $x \in \mathbb{R}^n$ denotes the measurable system state, and $f(x) \in \mathbb{R}^n$ and $g_j(x) \in \mathbb{R}^{n\times m_j}$ are both smooth nonlinear functions. Assume that $g_j(x)$ is known and Lipschitzcontinuous. u_{-i} is the set of control inputs for all players except player i: $u_{-i} = \{u_1, u_2, \ldots, u_{i-1}, u_{i+1}, \ldots, u_N\}$.

Assumption 1 ([39]). *For the OTCP, we need the following basic assumptions:*

(a) *The drift dynamics system $f(x)$ is unknown and Lipschitz-continuous on a compact set $\Omega \in \mathbb{R}^n$ with $f(0) = 0$.*

(b) *$g_j(x)$ is bounded by a constant b_{gj}, i.e., $||g_j(x)|| \leqslant b_{gj}$.*

Remark 1. *Assumption 1 (a) is a standard assumption that guarantees that the solution $x(t)$ of system (1) is unique for any finite initial condition. For Assumption 1 (b), although this assumption is somewhat strict, in practice, there are still many systems that meet such a condition, such as robot systems.*

The bounded reference signal is generated by a Lipschitz-continuous command generator

$$\dot{r}(t) = f_d(r(t)) \tag{2}$$

where $f_d(0) = 0$, $r(t) \in \mathbb{R}^n$ denotes the reference signal. Note that the reference dynamics only need to be stable in the Lyapunov sense and are not required to be asymptotically stable. Sine and cosine waves are some examples of such signals.

The purpose of tracking control is to achieve $x(t)$ following the $r(t)$. Then, the tracking error is given by

$$e_r(t) = x(t) - r(t). \tag{3}$$

Define the cost function of player i as

$$\begin{aligned} J_i(e_r(t), u_1, u_2, \ldots, u_N) = \int_t^{\infty} e^{-\lambda(\eta - t)} (e_r^T(\eta) Q_i e_r(\eta) \\ + \sum_{j=1}^{N} u_j^T(\eta) R_{ij} u_j(\eta))\, d\eta, i \in \mathbb{N} \end{aligned} \tag{4}$$

where $\mathbb{N} = \{1, 2, \ldots, N\}$, $Q_i = Q_i^T \geqslant 0, R_{ii} = R_{ii}^T > 0, R_{ij} = R_{ij}^T \geqslant 0$, and $\lambda > 0$ is the discount factor.

Take the derivative of Equation (3):

$$\dot{e}_r(t) = f(r(t) + e_r(t)) + \sum_{j=1}^{N} g_j(r(t) + e_r(t))u_j(t) - f_d(r(t)). \tag{5}$$

The objective of the OTCP is to determine the optimal control inputs $\{u_1^*, u_2^*, \ldots, u_N^*\}$ that ensure $e_r(t)$ asymptotically converges to zero, and the predetermined cost function (4) for each player i is minimized.

Next, we introduce the augmented state including tracking error $e_r(t)$ and reference signal $r(t)$ expressed by $\mu(t) = [e_r^T(t), r^T(t)]^T \in \mathbb{R}^{2n}$, and the corresponding augmented system can be obtained by using Equations (2) and (5):

$$\dot{\mu}(t) = \mathcal{F}(\mu(t)) + \sum_{j=1}^{N} \mathcal{G}_j(\mu(t))u_j(t) \tag{6}$$

where

$$\mathcal{F}(\mu(t)) = \begin{bmatrix} f(r(t) + e_r(t)) - f_d(r(t)) \\ f_d(r(t)) \end{bmatrix}, \mathcal{G}_j(\mu(t)) = \begin{bmatrix} g_j(r(t) + e_r(t)) \\ 0 \end{bmatrix}.$$

Redefine the cost function of player i as

$$\begin{aligned} \bar{J}_i(\mu(t), u_1, u_2, \ldots, u_N) = \int_t^{\infty} e^{-\lambda(\eta - t)} (\mu^T(\eta)\bar{Q}_i\mu(\eta) \\ + \sum_{j=1}^{N} u_j^T(\eta)R_{ij}u_j(\eta))\, d\eta \end{aligned} \tag{7}$$

where $\bar{Q}_i = \begin{bmatrix} Q_i & 0_{n\times n} \\ 0_{n\times n} & 0_{n\times n} \end{bmatrix}$.

Definition 1 ([39]). *(Admissible control.) The feedback control policy $u_i = u_i(\mu) \in \Phi(\Omega)$ is admissible with respect to (7) on $\Omega \in \mathbb{R}^n$ if $u_i(\mu)$ is continuous on Ω, $u_i(0) = 0$, $u_i(\mu)$ stabilizes the tracking error dynamics (5) on Ω, and (7) is finite $\forall \mu \in \Omega$.*

Remark 2. *As observed from (6), because the augmented system states contain $r(t)$, this state is uncontrollable. However, because the reference signal is assumed to be bounded, the admissible control policy means that $\mu(t)$ is bounded.*

For the simplicity of description, we denote admissible control as $u_i = u_i(\mu)$. Given admissible control u_i, the value functions for player i are given by the following:

$$V_i(\mu(t)) = \int_t^{\infty} e^{-\lambda(\eta - t)} \left(\mu^T(\eta)\bar{Q}_i\mu(\eta) + \sum_{j=1}^{N} u_j^T R_{ij} u_j \right) d\eta, i \in \mathbb{N}. \tag{8}$$

The goal of the optimal regulation problem is to find a set of admissible control sequences $\{u_1^*, u_2^*, \ldots, u_N^*\}$ that minimizes the value functions (8) for each player. $\{u_1^*, u_2^*, \ldots, u_N^*\}$ also represents the Nash equilibrium of NZS games.

Definition 2 ([40]). *(Nash equilibrium.) An N-tuple of control policies $\{u_1^*, u_2^*, \ldots, u_N^*\}$ is called the Nash equilibrium of an N-player game if the following N inequalities are satisfied:*

$$\bar{J}_i^* = \bar{J}_i(u_1^*, u_2^*, \ldots, u_i^*, \ldots, u_N^*) \leqslant \bar{J}_i(u_1^*, u_2^*, \ldots, u_i, \ldots, u_N^*), i \in \mathbb{N}. \tag{9}$$

Remark 3. *The value function (8) must use a discount factor $\lambda > 0$ because $r(t)$ does not go to zero and then the input cost $\sum_{j=1}^{N} u_j^T R_{ij} u_j$ does not go to zero either, and therefore, the performance function is unbounded.*

Remark 4. *The cost function (4) for the OTCP of system (1) has been transformed into a cost function (7) for the related problem of partial optimal regulation (i.e., only adjust the tracking error) by building an augmented system (6). Therefore, we can solve the OTCP of system (1) by using the method of dealing with the optimal regulation problem.*

Assume that the value function $V_i(\mu(t)) \in C^1(\Omega)$, where $C^1(\Omega)$ is a function space on Ω with continuous first derivatives for $i \in \mathbb{N}$. By differentiating V_i along the system trajectories (6), we can write Equation (8) as follows:

$$\begin{aligned} \dot{V}_i(\mu) = \int_t^{\infty} \frac{\partial}{\partial t} e^{-\lambda(\eta - t)} \left(\mu^T \bar{Q}_i \mu + \sum_{j=1}^{N} u_j^T R_{ij} u_j \right) d\eta \\ - U_i(\mu, u_1, u_2, \ldots, u_N), i \in \mathbb{N}. \end{aligned} \tag{10}$$

Equation (10) can be written as

$$0 = U_i(\mu, u_1, u_2, \ldots, u_N) - \lambda V_i + \nabla V_i^T \left(\mathcal{F}(\mu) + \sum_{j=1}^{N} \mathcal{G}_j(\mu) u_j \right), i \in \mathbb{N} \tag{11}$$

where $V_i(0) = 0$, $\nabla V_i = \frac{\partial V_i}{\partial \mu}$, ∇V_i^T is the transpose of ∇V_i, and $U_i(\mu, u_1, u_2, \ldots, u_N) = \mu^T \bar{Q}_i \mu + \sum_{j=1}^{N} u_j^T R_{ij} u_j$.

Define the Hamiltonian functions:

$$\begin{aligned} H_i(\mu, \nabla V_i, u_1, u_2, \ldots, u_N) = U_i(\mu, u_1, u_2, \ldots, u_N) \\ - \lambda V_i + \nabla V_i^T \left(\mathcal{F}(\mu) + \sum_{j=1}^{N} \mathcal{G}_j(\mu) u_j \right), i \in \mathbb{N}. \end{aligned} \tag{12}$$

The optimal value functions V_i^* can be given:

$$V_i^*(\mu(t)) = \min_{u_i} \int_t^{\infty} e^{-\lambda(\eta - t)} \left(\mu^T(\eta) \bar{Q}_i \mu(\eta) + \sum_{j=1}^{N} u_j^T R_{ij} u_j \right) d\eta, i \in \mathbb{N}. \tag{13}$$

Using the stationarity conditions $\frac{\partial H_i}{\partial u_i} = 0$ [41], the optimal control inputs can be obtained:

$$u_i^*(\mu) = -\frac{1}{2} R_{ii}^{-1} \mathcal{G}_i^T(\mu) \nabla V_i^*, i \in \mathbb{N}. \tag{14}$$

Substituting Equation (14) into Equation (11), the N coupled HJ equations are obtained:

$$\begin{aligned} 0 = &(\nabla V_i^*)^T \mathcal{F}(\mu) + \mu^T \bar{Q}_i \mu - \lambda V_i^* - \frac{1}{2} (\nabla V_i^*)^T \sum_{j=1}^{N} \mathcal{G}_j(\mu) R_{jj}^{-1} \mathcal{G}_j^T(\mu) \nabla V_j^* \\ &+ \frac{1}{4} \sum_{j=1}^{N} (\nabla V_j^*)^T \mathcal{G}_j(\mu) \left(R_{jj}^{-1} \right)^T R_{ij} R_{jj}^{-1} \mathcal{G}_j^T(\mu) \nabla V_j^*, V_i(0) = 0, i \in \mathbb{N}. \end{aligned} \tag{15}$$

It is clear from Equation (14) that $V_i^*(\mu)$ must be known if $u_i^*(\mu)$ is to be obtained. That is, solving the OTCP for NZS games is ultimately a question of solving the coupled HJ Equation (15). However, because the coupled HJ equation is an NPDE, it is very difficult to solve it directly. Next, we will apply IRL to try to address the coupled HJ equations for augmented systems (6).

2.2. Policy Iteration Solution for NZS Games

It is essential to recognize that solving the coupled HJ equation (15) requires the information of all the other players' policies. Thus, Equation (15) is difficult to solve. Next, we try to obtain the solution with the PI technique.

The following Algorithm 1 is actually an infinite iterative process that is only suitable for theoretical analysis in this paper. For practical systems, it is common to set a termination condition on the value function in step 4. According to [28], the convergence of Algorithm 1 is proven, i.e., $V_i^k(\mu) \to V_i^*(\mu)$ and $\{u_i^k(\mu), u_{-i}^k(\mu)\} \to \{u_i^*(\mu), u_{-i}^*(\mu)\}$ as $k \to \infty$.

It is clear that Equation (16) still requires a full system model because Algorithm 1 does not provide a solution for the HJ equation with unknown drift dynamics. References [35,42] used the identifier technique to solve unknown NZS games. To avoid the identification process, we adopt the IRL method to tackle NZS games with multiple inputs, where $\mathcal{F}(\mu)$ is unknown.

Algorithm 1 Model-based PI for solving the HJ equation

1: Start with an initial policies $\{u_1^0, u_2^0, \ldots, u_N^0\} \in \Phi(\Omega)$, and set $k = 0$.

2: According to the control policies of the N-tuple $\{u_1^k, u_2^k, \ldots, u_N^k\}$, find the N-tuple of value functions $\{V_1^{k+1}(\mu), V_2^{k+1}(\mu), \ldots, V_N^{k+1}(\mu)\}$ successively approximated by solving

$$\begin{aligned} (\nabla V_i^{k+1})^T \left[\mathcal{F}(\mu) + \sum_{j=1}^{N} \mathcal{G}_j(\mu) u_j^k\right] - \lambda V_i^{k+1} + U_i(\mu, u_i^k, u_{-i}^k) = 0, \\ V_i^{k+1}(0) = 0, i \in \mathbb{N}. \end{aligned} \tag{16}$$

3: Revise the N-tuple of control policies as follows

$$u_i^{k+1}(\mu) = -\frac{1}{2} R_{ii}^{-1} \mathcal{G}_i^T(\mu) \nabla V_i^{k+1}(\mu) \tag{17}$$

4: Let $k = k + 1$, and return to Step 2.

3. IRL Method for NZS Games

In this section, we adopt an IRL method to tackle NZS games and prove the convergence of the IRL method.

3.1. IRL Method

Inspired by [43], we can rewrite system (6) as follows:

$$\dot{\mu} = \mathcal{F}(\mu) + \sum_{j=1}^{N} \mathcal{G}_j(\mu)\left(u_j - u_j^k\right) + \sum_{j=1}^{N} \mathcal{G}_j(\mu) u_j^k \tag{18}$$

where $\forall u_j \in \Phi(\Omega), j \in \mathbb{N}$, u_j^k represents the kth iteration of the jth control input.

Let $V_i^{k+1}(\mu)$ be the solution of Equation (16). The time derivative of $V_i^{k+1}(\mu)$ along the system trajectory (18) is

$$\begin{aligned} \frac{dV_i^{k+1}(\mu)}{dt} &= (\nabla V_i^{k+1})^T \left[\mathcal{F}(\mu) + \sum_{j=1}^{N} \mathcal{G}_j(\mu) u_j^k + \sum_{j=1}^{N} \mathcal{G}_j(\mu)(u_j - u_j^k)\right] \\ &= \lambda V_i^{k+1} - U_i(\mu, u_i^k, u_{-i}^k) + (\nabla V_i^{k+1})^T \sum_{j=1}^{N} \mathcal{G}_j(\mu)(u_j - u_j^k). \end{aligned} \tag{19}$$

According to the IRL technique, taking integrals on both sides of Equation (19) over the time interval $[t, t+\Delta t]$,

$$\begin{aligned} V_i^{k+1}(\mu(t+\Delta t)) - V_i^{k+1}(\mu(t)) = &\int_t^{t+\Delta t} (\nabla V_i^{k+1}(\mu(\eta)))^T \sum_{j=1}^{N} \mathcal{G}_i(\mu(\eta))(u_j - u_j^k) d\eta \\ &- \int_t^{t+\Delta t} U_i\left(\mu(\eta), u_i^k, u_{-i}^k\right) d\eta + \int_t^{t+\Delta t} \lambda V_i^{k+1}(\mu(\eta)) d\eta. \end{aligned} \tag{20}$$

From Equation (20), it is evident that dynamics knowledge $\mathcal{F}(\mu)$ is not needed. Therefore, by replacing Equation (16) in Algorithm 1 with Equation (20), the NZS games with unknown $\mathcal{F}(\mu)$ are solved. Next, we will prove that Equation (16) is equivalent to Equation (20).

Theorem 1. *Let $V_i^{k+1}(\mu) \in C^1(\Omega)$, $V_i^{k+1}(\mu) \geqslant 0$, and $V_i^{k+1}(0) = 0$. $V_i^{k+1}(\mu)$ is the solution of Equation (20) if and only if $V_i^{k+1}(\mu)$ is the solution of Equation (16).*

Proof of Theorem 1. From the derivation of Equation (20), it is obvious that if V_i^{k+1} is the solution of Equation (16), then V_i^{k+1} satisfies Equation (20). If we can prove that Equation (20) has only one solution, then Equation (20) is equivalent to Equation (16). We use the contradiction method to derive that Equation (20) has only one solution. Before embarking on the proof of contradiction, let us derive the following fact:

$$\begin{aligned} &\lim_{\Delta t \to 0} \frac{1}{\Delta t} \int_t^{t+\Delta t} h(\eta) d\eta \\ &\quad = \lim_{\Delta t \to 0} \frac{1}{\Delta t} \left(\int_0^{t+\Delta t} h(\eta) d\eta - \int_0^t h(\eta) d\eta \right) \\ &\quad = \frac{d}{dt} \int_0^t h(\eta) d\eta \\ &\quad = h(t). \end{aligned} \tag{21}$$

From Equation (20), we can obtain

$$\begin{aligned} &\frac{dV_i^{k+1}(\mu(t))}{dt} \\ &\quad = \lim_{\Delta t \to 0} \frac{1}{\Delta t} \left(V_i^{k+1}(\mu(t+\Delta t)) - V_i^{k+1}(\mu(t)) \right) \\ &\quad = \lim_{\Delta t \to 0} \int_t^{t+\Delta t} (\nabla V_i^{k+1}(\mu(\eta)))^T \sum_{j=1}^{N} \mathcal{G}_i(\mu(\eta))(u_j - u_j^k) d\eta \\ &\quad - \lim_{\Delta t \to 0} \int_t^{t+\Delta t} U_i\left(\mu(\eta), u_i^k, u_{-i}^k\right) d\eta + \lim_{\Delta t \to 0} \int_t^{t+\Delta t} \lambda V_i^{k+1}(\mu(\eta)) d\eta. \end{aligned} \tag{22}$$

By using fact (21), Equation (22) can be written as Equation (19). Suppose that there is another solution $Z_i(\mu(t))$ of Equation (20) with $Z_i(\mu(t)) \geqslant 0$ and $Z_i(0) = 0$. So, $Z_i(\mu(t))$ also satisfies Equation (20), i.e.,

$$\frac{dZ_i(\mu(t))}{dt} = \lambda Z_i(\mu(t)) - U_i(\mu(t), u_i^k, u_{-i}^k) + \nabla Z_i^T(\mu(t)) \sum_{j=1}^{N} \mathcal{G}_j(\mu(t))(u_j - u_j^k). \tag{23}$$

From Equations (20) and (23), we can obtain

$$\begin{aligned}\frac{d}{dt}\left(V_i^{k+1}(\mu(t)) - Z_i(\mu(t))\right) &= \lambda(V_i^{k+1}(\mu(t)) - Z_i(\mu(t))) \\ &+\left((\nabla V_i^{k+1}(\mu(t)))^T - \nabla Z_i^T(\mu(t))\right)\sum_{j=1}^{N}\mathcal{G}_j(\mu(t))\left(u_j - u_j^k\right)\end{aligned} \tag{24}$$

then

$$\begin{aligned}&\frac{d}{dt}(V_i^{k+1}(\mu(t)) - Z_i(\mu(t))) - \lambda(V_i^{k+1}(\mu(t)) - Z_i(\mu(t))) \\ &\quad= \left((\nabla V_i^{k+1})^T(\mu(t)) - \nabla Z_i^T(\mu(t))\right)\sum_{j=1}^{N}\mathcal{G}_j(\mu(t))\left(u_j - u_j^k\right).\end{aligned} \tag{25}$$

Multiplying $e^{-\lambda t}$ on both sides of Equation (25), we can rewrite Equation (25) as follows:

$$\begin{aligned}\frac{d}{dt}\left(e^{-\lambda t}\left(V_i^{k+1}(\mu(t)) - Z_i(\mu(t))\right)\right) &= e^{-\lambda t}\left((\nabla V_i^{k+1})^T(\mu(t)) - \nabla Z_i^T(\mu(t))\right) \\ &\times\sum_{j=1}^{N}\mathcal{G}_j(\mu(t))\left(u_j - u_j^k\right).\end{aligned} \tag{26}$$

Equation (26) always holds $\forall u_j \in \Phi(\Omega)$. When we select $u_j = u_j^k$, then

$$\frac{d}{dt}\left(e^{-\lambda t}\left(V_i^{k+1}(\mu(t)) - Z_i(\mu(t))\right)\right) \equiv 0, \forall \mu(t) \in \Omega \tag{27}$$

Thus,

$$e^{-\lambda t}\left(V_i^{k+1}(\mu(t)) - Z_i(\mu(t))\right) \equiv c, \forall \mu(t) \in \Omega \tag{28}$$

where c is a real constant.

Now, considering the condition $V_i^{k+1}(0) = 0$ and $Z_i(0) = 0$, it follows that $c = e^{-\lambda t}\left(V_i^{k+1}(0) - Z_i(0)\right) = 0$. According to $e^{-\lambda t} > 0$, it can be deduced that $V_i^{k+1}(\mu(t)) = Z_i(\mu(t))\ \forall \mu(t) \in \Omega$. This contradicts the existence of another solution. Then, Equation (20) has only a solution, which means that its solution is equivalent to that of Equation (16). Thus, the proof is complete. □

From Theorem 1, it follows that the solution of Equation (16) is equivalent to the solution of Equation (20), so the convergence of the IRL iterative method (20) is guaranteed. That is, the equivalence between Algorithm 1 and the IRL method is proven, which ensures the convergence of the IRL method.

3.2. Single-Layer Critic NN

A single-layer critic NN is utilized for approximating the solution to Equation (20). According to the Weierstrass approximation theorem, the approximate form of the value function $V_i^{k+1}(\mu)$ and its gradient $\nabla V_i^{k+1}(\mu)$ can be given as follows:

$$V_i^{k+1}(\mu) = \omega_{i,k+1}^T\psi_i(\mu) + \sigma_{i,k+1} \tag{29}$$

$$\nabla V_i^{k+1}(\mu) = \nabla\psi_i^T(\mu)\omega_{i,k+1} + \nabla\sigma_{i,k+1}, i \in \mathbb{N} \tag{30}$$

where $\psi_i : \mathbb{R}^{2n} \to \mathbb{R}^{K_i}$ are linearly independent activation functions, K_i denotes the number of hidden neurons, $\omega_{i,k+1} \in \mathbb{R}^{K_i}$ are the unknown ideal weights, and $\sigma_{i,k+1}$ are the approximation errors. It is shown in [8] that as $K_i \to \infty$, the approximation error $\sigma_{i,k+1}$ converges to zero.

Assumption 2 ([39]).

(1) The approximation error $\sigma_{i,k+1}(\mu)$ and its gradient $\nabla\sigma_{i,k+1}(\mu)$ are bounded on Ω, specifically, $||\sigma_{i,k+1}(\mu)|| \leqslant b_{\sigma_i}$ and $||\nabla\sigma_{i,k+1}(\mu)|| \leqslant b_{\sigma\mu_i}$, where b_{σ_i} and $b_{\sigma\mu_i}$, $i \in \mathbb{N}$, are positive constants.

(2) The activation functions $\psi_i(\mu)$ and their gradients $\nabla\psi_i(\mu)$ are bounded, i.e., $||\psi_i(\mu)|| \leqslant b_{\psi_i}$ and $||\nabla\psi_i(\mu)|| \leqslant b_{\psi\mu_i}$, with b_{ψ_i} and $b_{\psi\mu_i}$, $i \in \mathbb{N}$, being positive constant.

Remark 5. *For Assumption 2 (1), it is known that as the number of neurons $K_i \to \infty$, the error $\sigma_{i,k+1}(\mu) \to 0$. In addition, for fixed K_i, there exist $||\sigma_{i,k+1}(\mu)|| \leqslant b_{\sigma_i}$ and $||\nabla\sigma_{i,k+1}(\mu)|| \leqslant b_{\sigma\mu_i}$. For Assumption 2 (2), this condition is mild in practice because many activation functions, such as the sigmoid function and tanh function, satisfy Assumption 2 (2).*

According to Equation (29), Equation (20) can be written as

$$\begin{aligned}
&(\psi_i(\mu(t+\Delta t)) - \psi_i(\mu(t)))^T\omega_{i,k+1} \\
&\quad - \int_t^{t+\Delta t}\sum_{j=1}^{N}(\mathcal{G}_j(\mu(\eta)(u_j(\eta) - u_j^k(\eta)^T))\nabla\psi_i^T(\mu)\omega_{i,k+1}d\eta \\
&+ \int_t^{t+\Delta t}(\mu\bar{Q}_i\mu + \sum_{j=1}^{N}((u_j^k(\eta))^T R_{ij}u_i^k(\eta)))d\eta - \lambda\int_t^{t+\Delta t}(\psi_i(\mu(\eta)))^T w_{i,k+1}d\eta \\
&= e_{i,k+1}(\mu(t))
\end{aligned} \tag{31}$$

where $e_{i,k+1}(\mu(t))$ is the error from the NN approximation error:

$$\begin{aligned}
&e_{i,k+1}(\mu(t)) = \sigma_{i,k+1}(\mu(t)) - \sigma_{i,k+1}(\mu(t+\Delta t)) \\
&+ \int_t^{t+\Delta t}\sum_{j=1}^{N}(\mathcal{G}_j(u_j(\eta) - u_j^k(\eta))^T\nabla\sigma_{i,k+1}(\mu(\eta)))d\eta + \lambda\int_t^{t+\Delta t}\sigma_{i,k+1}(\mu(\eta))d\eta.
\end{aligned} \tag{32}$$

Denote by $\hat{\omega}_{i,k+1}$ the estimations of $\omega_{i,k+1}$. Thus, $V_i^{k+1}(\mu)$ can be approximated as

$$\hat{V}_i^{k+1}(\mu) = \hat{\omega}_{i,k+1}^T\psi_i(\mu), i \in \mathbb{N}. \tag{33}$$

Based on Equation (17), the approximate control policies are

$$\hat{u}_i^{k+1}(\mu) = -\frac{1}{2}R_{ii}^{-1}\mathcal{G}_i^T(\mu)\nabla\psi_i^T(\mu)\hat{\omega}_{i,k+1}, i \in \mathbb{N}. \tag{34}$$

Remark 6. *Because the input dynamics $G_i(\mu)$ are known, we directly use the critic NN approximation (19) to obtain the approximated optimal control (20). Therefore, the single-layer critic structure is adopted instead of the actor–critic structure, reducing the computational cost and avoiding approximation errors from the action NN.*

Due to the estimation error of Equation (29), $\hat{V}_i^k(\mu)$ is replaced by $V_i^k(\mu)$ in Equation (20). Therefore, the residual error for player i is given by

$$\begin{aligned}\hat{e}_{i,k+1}&(\mu(t),u_i(t),u_{-i}(t))\\ &=(\psi_i(\mu(t))-\psi_i(\mu(t+\Delta t)))^T\hat{\omega}_{i,k+1}\\ &+\int_t^{t+\Delta t}\sum_{j=1}^{N}(\mathcal{G}_j(\mu(\eta))(u_j(\eta)-u_j^k(\eta)))^T\nabla\psi_i^T(\mu(\eta))\hat{\omega}_{i,k+1}d\eta\\ &-\int_t^{t+\Delta t}\bar{Q}_i(\mu(\eta))d\eta-\int_t^{t+\Delta t}\sum_{j=1}^{N}((u_j^k(\eta))^TR_{ij}u_j^k(\eta))d\eta\\ &+\lambda\int_t^{t+\Delta t}\psi_j(\mu(\eta))\hat{\omega}_{i,k+1}d\eta.\end{aligned}\tag{35}$$

Note that

$$\begin{aligned}&\int_t^{t+\Delta t}\sum_{j=1}^{N}(\mathcal{G}_j(\mu(\eta))(u_j(\eta)-u_j^k(\eta)))^T\nabla\psi_i^T(\mu(\eta))\hat{\omega}_{i,k+1}d\eta\\ &=\int_t^{t+\Delta t}\sum_{j=1}^{N}(u_j^T(\eta)\mathcal{G}_j^T(\mu(\eta)))\nabla\psi_i^T(\mu(\eta))d\eta\\ &+\frac{1}{2}\int_t^{t+\Delta t}\sum_{j=1}^{N}(\hat{\omega}_{j,k}^T\nabla\psi_j(\mu(\eta))\mathcal{G}_j(\mu(\eta))R_{jj}^{-1}\mathcal{G}_j^T(\mu(\eta)))\nabla\psi_i^T(\mu(\eta))\hat{\omega}_{i,k+1}d\eta\end{aligned}\tag{36}$$

and

$$\begin{aligned}\int_t^{t+\Delta t}\sum_{j=1}^{N}((u_j^k(\eta))^TR_{ij}u_j^k(\eta))d\eta=\frac{1}{4}\int_t^{t+\Delta t}\sum_{j=1}^{N}&(\hat{\omega}_{j,k}^T\nabla\psi_j(\mu(\eta))\mathcal{G}_j(\mu(\eta))R_{jj}^{-1}\\ &\times R_{ij}R_{jj}^{-1}\mathcal{G}_j^T(\mu(\eta))\nabla\psi_j^T(\mu(\eta))\hat{\omega}_{j,k})d\eta.\end{aligned}\tag{37}$$

For notation simplicity, define

$$\begin{aligned}D_{i,j}(\mu)&=\nabla\psi_j(\mu)\mathcal{G}_j(\mu)R_{jj}^{-1}R_{ij}R_{jj}^{-1}\mathcal{G}_j^T(\mu)\nabla\psi_j^T(\mu)\\ E_{i,j}(\mu)&=\nabla\psi_j(\mu)\mathcal{G}_j(\mu)R_{jj}^{-1}\mathcal{G}_j^T(\mu)\nabla\psi_i^T(\mu)\\ \zeta_{1,i}(\mu(t))&=(\psi_i(\mu(t))-\psi_i(\mu(t+\Delta t)))^T\\ \zeta_{2,i}(\mu(t),u_i,u_{-i})&=\int_t^{t+\Delta t}\sum_{j=1}^{N}(u_j^T(\eta)\mathcal{G}_j^T(\mu(\eta)))\nabla\psi_i^T(\mu(\eta))d\eta\\ \zeta_{3,i}(\mu(t))&=\begin{bmatrix}\int_t^{t+\Delta t}E_{i,1}(\mu(\eta))d\eta\\ \vdots\\ \int_t^{t+\Delta t}E_{i,N}(\mu(\eta))d\eta\end{bmatrix}\\ \zeta_{4,i}(\mu(t))&=\int_t^{t+\Delta t}\lambda\psi_i^T(\mu(\eta))d\eta\\ \zeta_{5,i}(\mu(t))&=\int_t^{t+\Delta t}\bar{Q}_i(\mu(\eta))d\eta\\ \zeta_{6,i}(\mu(t))&=\begin{bmatrix}\int_t^{t+\Delta t}D_{i,1}(\mu(\eta))d\eta & 0 & 0\\ 0 & \ddots & \vdots\\ 0 & \dots & \int_t^{t+\Delta t}D_{i,N}(\mu(\eta))d\eta.\end{bmatrix}\end{aligned}$$

Equation (35) can be written as

$$\hat{e}_{i,k+1}(\mu(t),u_i(t),u_{-i}(t))=\rho_i(\mu(t),u_i(t),u_{-i}(t))\hat{\omega}_{i,k+1}-s_i(\mu(t))\tag{38}$$

where

$$\begin{aligned}\rho_i(\mu(t), u_i(t), u_{-i}(t)) &= \zeta_{1,i}(\mu(t)) + \zeta_{2,i}(\mu(t), u_i(t), u_{-i}(t)) \\ &\quad + \frac{1}{2}\hat{W}_k^T \zeta_{3,i}(\mu(t)) + \zeta_{4,i}(\mu(t)) \\ s_i(\mu(t)) &= \zeta_{5,i}(\mu(t)) + \frac{1}{4}\hat{W}_k^T \zeta_{6,i}(\mu(t))\hat{W}_k \\ \hat{W}_k &= [\hat{\omega}_{1,k}^T, \ldots, \hat{\omega}_{N,k}^T]^T\end{aligned}$$

Consider the objective function

$$E_{i,k+1} = \frac{1}{2}\hat{e}_{i,k+1}^2. \tag{39}$$

In the following section, an algorithm is proposed to update the weights $\hat{\omega}_{i,k+1}$ by minimizing $E_{i,k+1}$ and to prove the convergence of the algorithm.

4. Offline Iterative Learning

4.1. Offline Algorithm for Updating the Weights

The LS method will be used for updating $\hat{\omega}_{i,k+1}$. $\{t_m\}_{m=0}^p$ represents a strictly increasing time series, where p represents the number of samples and is a sufficiently large integer. $M_i = \{(\mu_m, u_{i,m}, u_{-i,m})\}_{m=0}^p$ denotes the sample set, where $\mu_m = \mu(t_m)$ is the state at time t_m and $u_{i,m} = u_i(t_m)$ and $u_{-i,m} = u_{-i}(t_m)$ represent the control input at time t_m with $m = 0, 1, \ldots, p$. For simplicity, let $\rho_{i,m} = \rho_i(\mu_m, u_{i,m}, u_{-i,m})$ and $s_{i,m} = s_i(\mu_m, u_{i,m}, u_{-i,m})$, where

$$\begin{aligned}\rho_{i,m} &= \zeta_{1,i}(\mu(t_m)) + \zeta_{2,i}(\mu(t_m), u_{i,m}, u_{-i,m}) + \frac{1}{2}\hat{W}_k^T \zeta_{3,i}(\mu(t_m)) + \zeta_4(\mu(t_m)) \\ s_{i,m} &= \zeta_{5,i}(\mu(t_m)) + \frac{1}{4}\hat{W}_k^T \zeta_{6,i}(\mu(t_m))\hat{W}_k.\end{aligned} \tag{40}$$

The following persistence of excitation (PE) condition is used for ensuring the convergence of $\hat{\omega}_{i,k+1}$.

Assumption 3. *There exist $p_0 > 0$ and $\beta > 0$ such that for all $p \geqslant p_0$, we have*

$$\frac{1}{p}\sum_{m=0}^{p-1} \rho_{i,m}^T \rho_{i,m} - \beta I_{i,K_i} \geqslant 0 \tag{41}$$

where $I_{i,K_i} \in \mathbb{R}^{K_i \times K_i}$ denotes the identity matrix.

Based on [8,43], the updating law of $\hat{\omega}_{i,k+1}$ is given by

$$\hat{\omega}_{i,k+1} = \left[P_i^T P_i\right]^{-1} P_i^T S_i \tag{42}$$

where

$$P_i = [\rho_{i,0}^T, \ldots, \rho_{i,p-1}^T]^T, S_i = [s_{i,0}, \ldots, s_{i,p-1}]^T.$$

An offline algorithm is presented based on the weight updating law (42). In Algorithm 2, we can see that steps 1–2 are a measurement process that is used to collect real data. Steps 3–4 are an offline learning process, which is used to approximate real weights.

Algorithm 2 NN-based offline learning for updating weights

1: For each player i, let $\{u_i^0, u_{-i}^0\} \in \Phi(\Omega)$ and initial weight $\hat{\omega}_{i,0}$, set $k = 0$, a small constant $\epsilon > 0$;
2: Then, collect the data $(\mu_m, u_{i,m}, u_{-i,m})$ for M_i, and compute $\zeta_{1,i}(\mu(t_m))$, $\zeta_{2,i}(\mu(t_m), u_{i,m}, u_{-i,m})$, $\zeta_{3,i}(\mu(t_m))$, $\zeta_{4,i}(\mu(t_m))$, $\zeta_{5,i}(\mu(t_m))$, and $\zeta_{6,i}(\mu(t_m))$;
3: Compute P_i and S_i and update $\hat{\omega}_{i,k+1}$ with Equation (42);
4: If $||\hat{\omega}_{i,k+1} - \hat{\omega}_{i,k}||^2 \leqslant \epsilon$, Stop iterating and bring $\hat{\omega}_{i,k+1}$ back into Equation (34) for optimal control input; else, let $k = k + 1$, and return to Step 3.

Remark 7. *In on-policy learning algorithms [26,38], approximate control policies (not real policies) are usually used for generating data and then learning the value function. This means that during the strategy learning process, "incorrect" data are employed, leading to the accumulation of errors. According to reference [43], Algorithm 2 can be regarded as an off-policy learning algorithm. In this algorithm, control u_i can be arbitrarily selected on $\Phi(\Omega)$, and ensures error-free data generation, thereby preventing cumulative errors.*

Remark 8. *As seen from (42), updating the weight requires the inverse of $[P_i^T P_i]^{-1}$, necessitating the PE condition to ensure the invertibility of this matrix. Thus, in practical applications, it becomes essential to add detection noise, such as random noise or sine waves of different frequencies, to make the given control input meet the PE assumption.*

4.2. Convergence Analysis for the Offline Algorithm

To show the effectiveness of the updating law (42), the following theorem is given.

Theorem 2. *For $i \in \mathbb{N}$, assume that V_i^{k+1} is the solution of Equation (20) and Assumption 3 holds. $\forall \mu(t) \in \Omega$, $\forall \delta > 0$, there $\exists K_i^* \in \mathbb{N}_+$ such that*

$$(1)\sup_{\mu\in\Omega}\left|\hat{V}_i^{k+1}(\mu) - V_i^{k+1}(\mu)\right| < \delta \tag{43}$$

$$(2)\sup_{\mu\in\Omega}\left|\hat{V}_i^{k+1}(\mu) - V_i^{*}(\mu)\right| < \delta \tag{44}$$

for $K_i > K_i^*$ (K_i is the number of neurons) and the approximate optimal tracking control $\{\hat{u}_1, \hat{u}_2, \ldots, \hat{u}_N\}$ in Equation (34) will converge to the Nash equilibrium, i.e., the tracking error dynamics system can be stabilized.

Proof of Theorem 2. A similar proof has already been provided in references [32,44]. To avoid repetition, we omit some similar proof steps.

From Theorem 2, V_i^{k+1} is the solution of iteration Equation (20). Then, with the same procedure used in Theorem 3.1 of reference [44] and Theorem 2 of reference [32], result (43) can be proven.

In other words, there exists $K_i^* > 0, \forall \mu \in \Omega, \delta > 0$ such that if $K_i > K_i^*$, then

$$\left|\hat{V}_i^{k+1}(\mu) - V_i^{k+1}(\mu)\right| < \delta. \tag{45}$$

According to Theorem 4 of reference [8], the result of $\sup_{\mu\in\Omega}|\hat{V}_i^{k+1}(\mu) - V_i^{*}(\mu)| < \delta$ can be proven directly.

The optimal input error of tracking control can be obtained by using Equations (14) and (34):

$$\begin{aligned} e_{ui} &= u_i^* - \hat{u}_i^k \\ &= -\frac{1}{2}R_{ii}^{-1}\mathcal{G}_i^T(\mu)\nabla V_i^*(\mu) + \frac{1}{2}R_{ii}^{-1}\mathcal{G}_i^T(\mu)\nabla\psi_i^T(\mu)\hat{\omega}_{i,k} \\ &= \frac{1}{2}R_{ii}^{-1}\mathcal{G}_i^T(\mu)(\nabla\hat{V}_i^k(\mu) - \nabla V_i^*(\mu)), i \in \mathbb{N}. \end{aligned} \tag{46}$$

Because ψ_i is linearly independent, using [27] Theorem 2, we know that $\sup_{\mu\in\Omega}|\nabla\hat{V}_i^k(\mu) - \nabla V_i^*(\mu)| < \delta$. We know from Assumption 1 (b) that $g_j(x)$ is bounded, so it is clear that $\mathcal{G}_j(\mu(t))$ is also bounded. Therefore, the errors e_{ui} will eventually converge to zero. In other words, $\hat{u}_i^k$ will converge to u_i^*, the N-tuple control inputs $\{\hat{u}_1, \hat{u}_2, \ldots, \hat{u}_N\}$ constitute a Nash equilibrium for the NZS games, and the tracking error dynamics (5) will be asymptotically stable. □

Remark 9. *From Theorem 2, we can easily obtain the following conclusions. The critic weight error converges to zero. The optimal control inputs $\{u_1^*, u_2^*, \ldots, u_N^*\}$ can enable $x(t)$ to track the reference signal $r(t)$, and $\hat{u}_i$ converges to u_i^*; then, the N-tuple control outputs $\{\hat{u}_1, \hat{u}_2, \ldots, \hat{u}_N\}$ can guarantee that the stability of the tracking error of the closed systems will be asymptotically stable.*

5. Simulation Results

We will verify the feasibility of the IRL method through a numerical simulation example. Consider the nonlinear CT differential game with two players as below [35]:

$$\dot{x} = f(x) + g_1(x)u_1 + g_2(x)u_2$$

where

$$f(x) = \begin{bmatrix} x_2 \\ -x_2 - 0.5x_1 - 0.25x_2(\sin(4x_1) + 2)^2 + 0.25x_2(\cos(2x_1) + 2)^2 \end{bmatrix}$$
$$g_1(x) = \begin{bmatrix} 0 \\ \cos(2x_1) + 2 \end{bmatrix}, g_2(x) = \begin{bmatrix} 0 \\ \sin(4x_1^2) + 2 \end{bmatrix}$$

$u_1, u_2 \in \mathbb{R}$ are the control inputs and $x = [x_1, x_2]^T \in \mathbb{R}^2$ is the system state.

The reference signals are given by the following commands:

$$\dot{r}(t) = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} r(t).$$

Select the initial state $x_0 = [2, 0]^T$, $r_0 = [-0.1168, 0.2763]^T$, $Q_1 = diag[1, 1]$, $Q_2 = diag[2, 2]$, $\lambda = 0.1$, $\sigma = 10^{-4}$, $R_{11} = R_{12} = 2$, and $R_{21} = R_{22} = 1$. Set the initial probing control input $u_1 = u_2 = 1.4(sin(8t)^2cos(2t) + sin(20t)^4cos(7t))$. We set the interval of integration as 0.05 and the number of samples collected as $p = 100$. The augmented system states are $\mu(t) = [\mu_1, \mu_2, \mu_3, \mu_4]^T = [e_1, e_2, r_1, r_2]^T$, and select the following activation functions

$$\psi_1(\mu(t)) = \psi_2(\mu(t)) = [e_1^2, e_1e_2, e_1r_1, e_1r_2, e_2^2, e_2r_1, e_2r_2, r_1^2, r_1r_2, r_2^2]^T$$

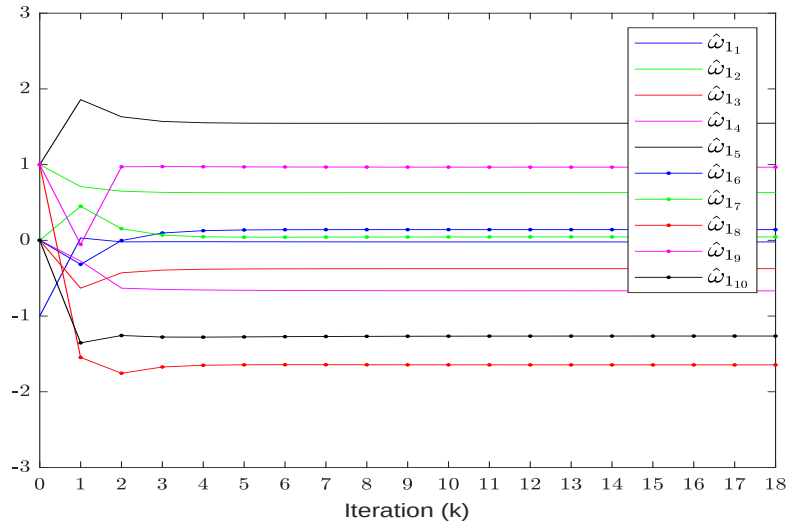
and the initial NN weights

$$\omega_{1,0} = [-1, 1, 0, 0, 1, 0, 0, 1, 1, 0]^T, \omega_{2,0} = [-1, -1, 1, -1, 0, -1, 1, 1, 0, 0]^T.$$

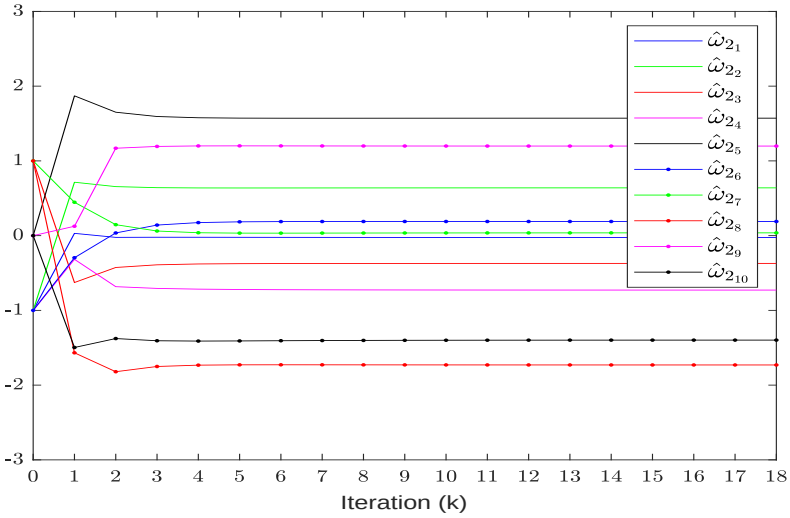
In order to verify the effectiveness of the proposed method, we will compare it with the method in [38] under the same conditions. To save space, this article presents a comparison of only some of the important results. Figure 1 shows the convergence curve of evaluating the weights of the critic NN, which finally converges to

$$\hat{\omega}_1 = [-0.0213, 0.6304, -0.3745, -0.6673, 1.5467, 0.1415, 0.0455, -1.6452, 0.9663, 1.2640]^T$$

$$\hat{\omega}_2 = [-0.0247, 0.6392, -0.3716, -0.7285, 1.5715, 0.1891, 0.0372, -1.7302, 1.1982, -1.3978]^T.$$



(**a**) Weights of critic NN for player 1.

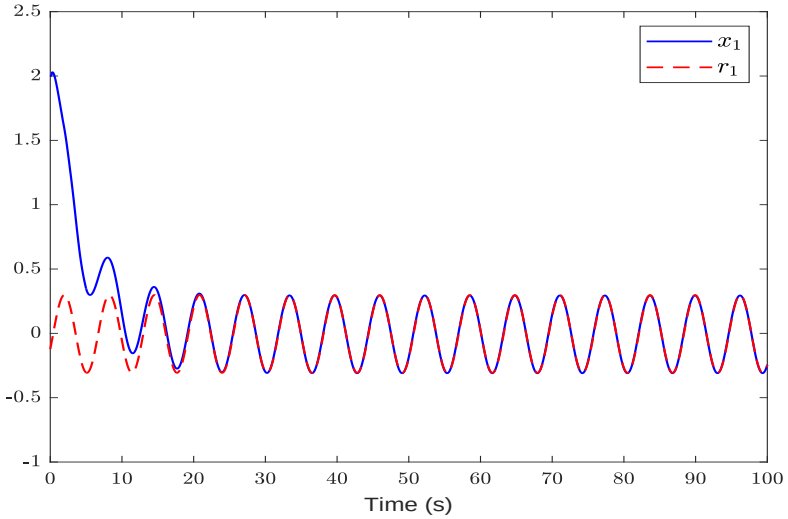


(**b**) Weights of critic NN for player 2.

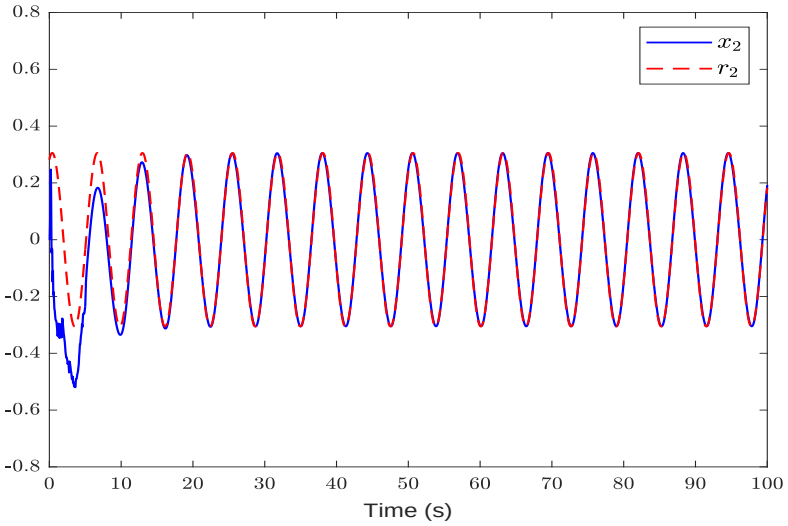
Figure 1. Critic NN weight convergence curve.

Figure 2 shows the system state $x(t)$ and reference trajectory $r(t)$ of the proposed method in this paper. It can be seen that the system state $x(t)$ can track the reference trajectory $r(t)$ after 20 s. Figure 3 shows the system state $x(t)$ and reference trajectory $r(t)$ of the method given in [38]. It can be seen that the reference trajectory $r(t)$ can be tracked in the system state $x(t)$ only after 90 s. In Figure 4a,b are the evolution curves of the tracking error of the proposed method in this paper and that of the proposed method in [38], respectively. From Figure 4, it is easy to see that the tracking error convergence speed of the proposed method in this paper is faster than that used in [38]. Figure 5 shows a comparison between the value function obtained by the proposed method and that obtained by the method used in [38]. It is easy to see that the value function of the proposed method in this paper is smaller both at the initial moment and the final moment, that is, the optimal control obtained in this paper is better than that obtained by the comparison method. The control inputs of the proposed method are compared with that of the comparison method in Figure 6. Figure 7 is an evolution curve approximating the HJ equation. For Equation (15), optimal control can make the left end of Equation (15) equal to zero, but for a non-optimal control it is not necessarily possible to make the left end of Equation (15) equal to zero. In this paper, approximate optimal control is used to approximate optimal control, so it is necessary to bring the obtained approximate optimal control back to the

right end of Equation (15) to verify whether it is equal to zero. By observing Figure 7, it can be seen that the optimal approximate control obtained by this method makes the left end of Equation (15) equal to zero, i.e., the optimal approximate control $\hat{u}_i^k$ converges to the optimal control u_i^*.

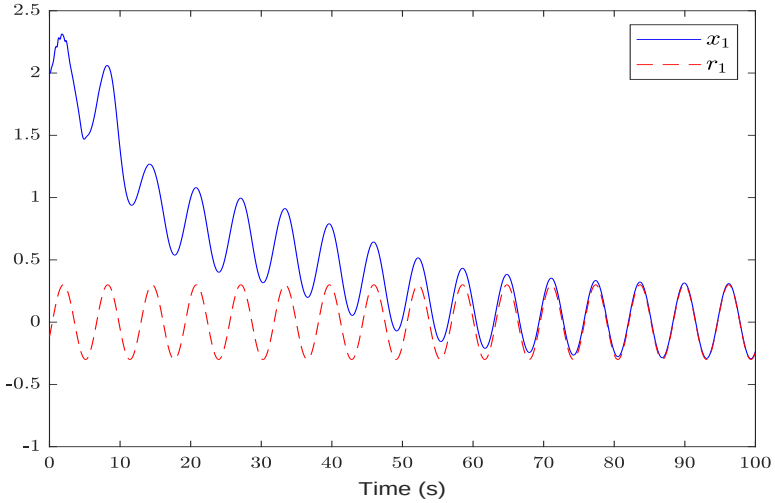


(**a**) The system state x_1 and the reference trajectory r_1.

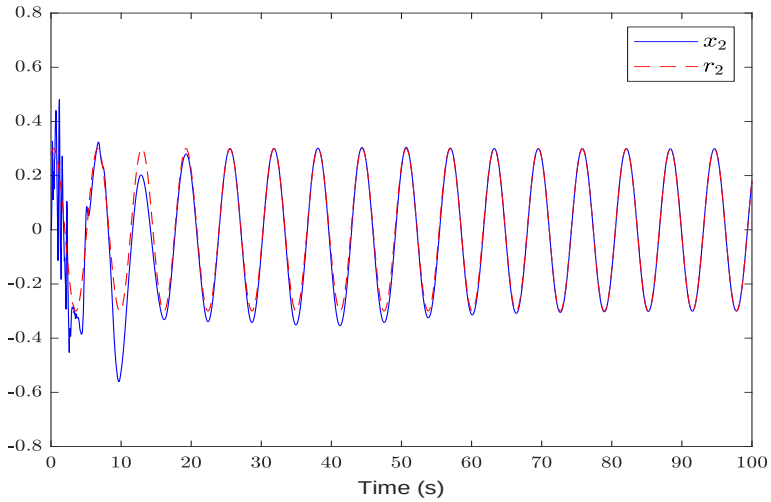


(**b**) The system state x_2 and the reference trajectory r_2.

Figure 2. The system state $x(t)$ of the proposed method is compared with the reference trajectory $r(t)$.

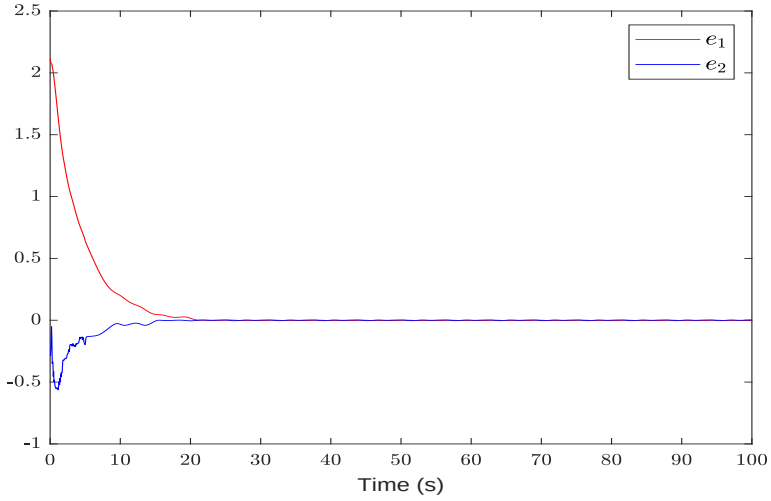


(**a**) The system state x_1 and the reference trajectory r_1.



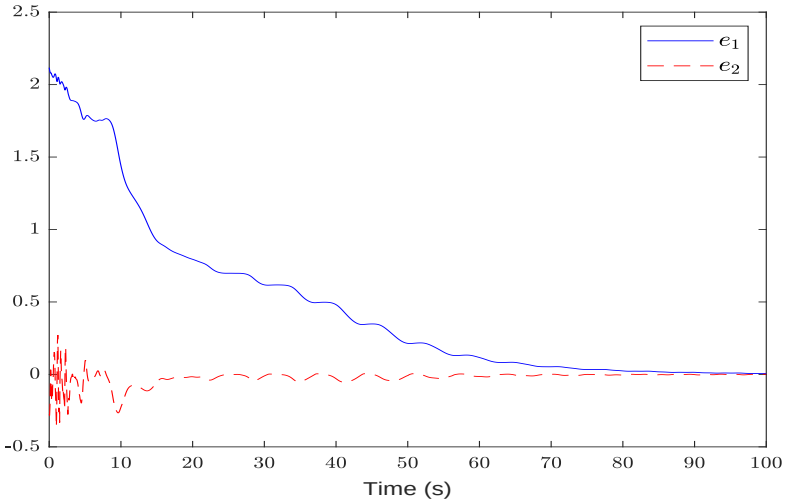
(**b**) The system state x_2 and the reference trajectory r_2.

Figure 3. The system state $x(t)$ of the comparison method is compared with the reference trajectory $r(t)$.



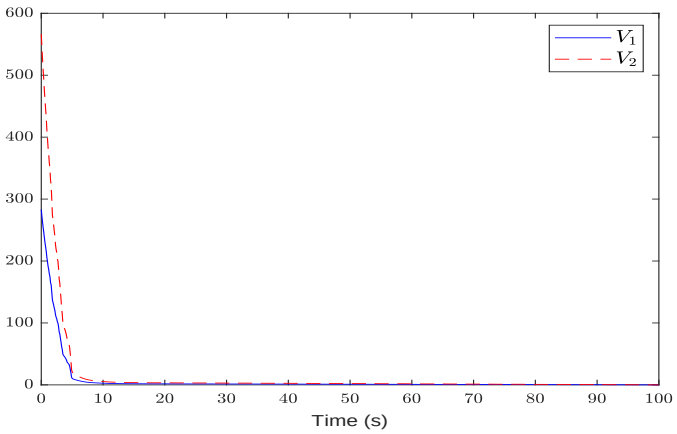
(**a**) The tracking error of the proposed method.

Figure 4. *Cont.*

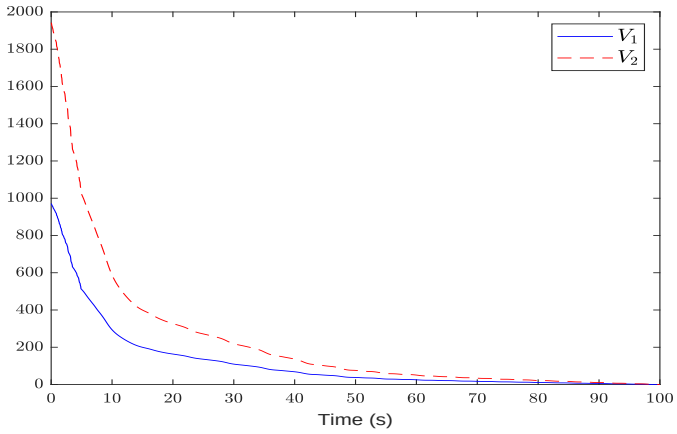


(**b**) The tracking error of the comparison method.

Figure 4. The evolution curve of the tracking error of the proposed method is compared with that of the comparison method.

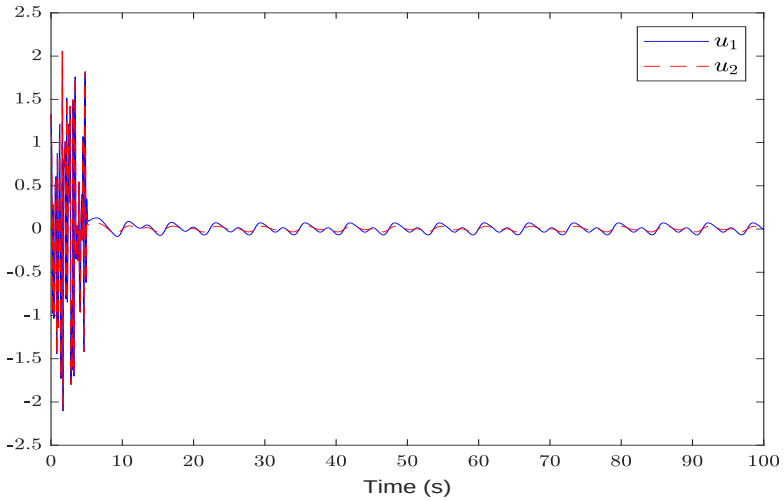


(**a**) The value function of the proposed method.

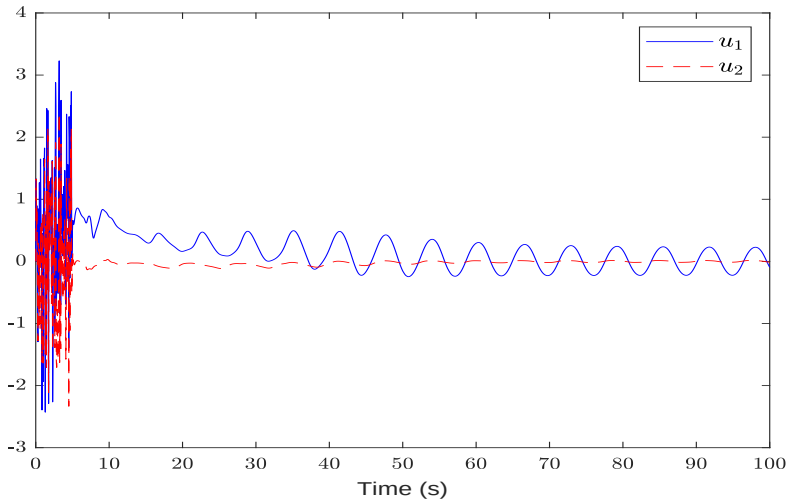


(**b**) The value function of the comparison method.

Figure 5. The comparison between the value function of the proposed method and that of the comparison method.



(**a**) Control inputs for the proposed method.



(**b**) Control inputs for the comparison method.

Figure 6. The control inputs of the proposed method are compared with that of the comparison method.

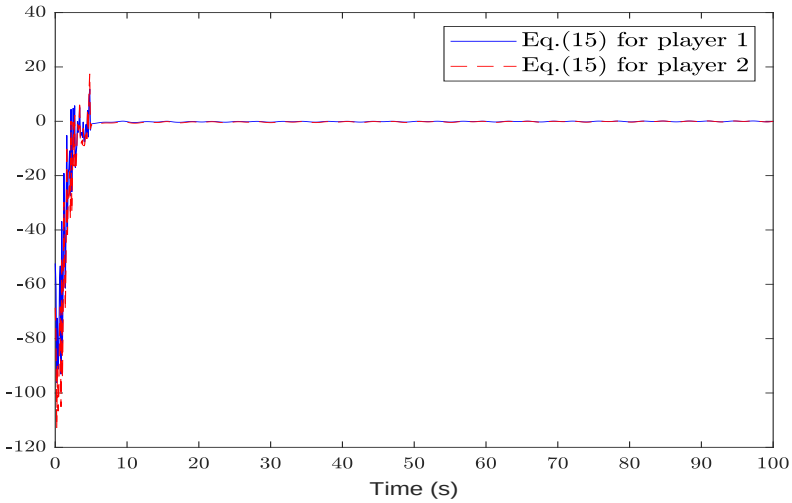


Figure 7. The evolution of (15).

6. Conclusions

To tackle the OTCP for nonlinear CT NZS differential game systems with unknown drift dynamics, an IRL method based on PI is proposed. Because the HJB equation is an NPDE that cannot be solved directly, the single-layer critic NN is used for approximating the value function of each player, and the LS method is used to update the weight of the NN. Due to the stability of the tracking error dynamics system, the approximate solutions converge to a Nash equilibrium, and the convergence of the weights of the NN is strictly proven. Finally, the validity of Algorithm 2 is verified by MATLAB simulation, and the comparison yields faster convergence and shows the higher convergence accuracy of this method.

Author Contributions: Conceptualization, C.J.; methodology, C.J.; validation, C.J. and Y.S.; writing—original draft preparation, C.J.; writing—review and editing, C.J.; visualization, C.W. and L.H.; supervision, C.W. and H.S.; project administration, C.W.; funding acquisition, C.W. All the authors have read and agreed to the published version of this manuscript.

Funding: This work was funded by the National Natural Science Foundation of China under grants (62173054 and 62173232).

Data Availability Statement: The data are contained within the article.

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

References

1. Jiang, Y.; Jiang, Z. Robust Adaptive Dynamic Programming for Large-Scale Systems with an Application to Multimachine Power Systems. *IEEE Trans. Circuits Syst. Express Briefs* **2012**, *59*, 693–697. [CrossRef]
2. Bian, T.; Jiang, Y.; Jiang, Z.P. Decentralized Adaptive Optimal Control of Large-Scale Systems with Application to Power Systems. *IEEE Trans. Ind. Electron.* **2015**, *62*, 2439–2447. [CrossRef]
3. Kirk, D.E. *Optimal Control Theory: An Introduction*; Dover Publications: New York, NY, USA, 2004.
4. Rodrigues, L. Affine Quadratic Optimal Control and Aerospace Applications. *IEEE Trans. Aerosp. Electron. Syst.* **2021**, *57*, 795–805. [CrossRef]
5. Lu, K.; Yu, H.; Liu, Z.; Han, S.; Yang, J. Inverse Optimal Adaptive Control of Canonical Nonlinear Systems with Dynamic Uncertainties and Its Application to Industrial Robots. *IEEE Trans. Ind. Inf.* **2024**, *20*, 5318–5327. [CrossRef]
6. Alfred, D.; Czarkowski, D.; Teng, J. Reinforcement Learning-Based Control of a Power Electronic Converter. *Mathematics* **2024**, *12*, 671. [CrossRef]
7. Mu, C.; Zhen, N.; Sun, C.; He, H. Data-Driven Tracking Control with Adaptive Dynamic Programming for a Class of Continuous-Time Nonlinear Systems. *IEEE Trans. Cybern.* **2017**, *47*, 1460–1470. [CrossRef]
8. 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]
9. Lv, Y.; Na, J.; Yang, Q.; Wu, X.; Guo, Y. Online adaptive optimal control for continuous-time nonlinear systems with completely unknown dynamics. *Int. J. Control* **2015**, *89*, 99–112. [CrossRef]
10. Xiao, G.; Zhang, H. Convergence Analysis of Value Iteration Adaptive Dynamic Programming for Continuous-Time Nonlinear Systems. *IEEE Trans. Cybern.* **2024**, *54*, 1639–1649. [CrossRef]
11. Wang, G. Distributed control of higher-order nonlinear multi-agent systems with unknown non-identical control directions under general directed graphs. *Automatica* **2019**, *110*, 108559. [CrossRef]
12. Chen, B.; Hu, J.; Zhao, Y.; Ghosh, B.K. Finite-time observer based tracking control of uncertain heterogeneous underwater vehicles using adaptive sliding mode approach. *Neurocomputing* **2022**, *481*, 322–332. [CrossRef]
13. Wang, G. Consensus Algorithm for Multiagent Systems with Nonuniform Communication Delays and Its Application to Nonholonomic Robot Rendezvous. *IEEE Trans. Control Netw. Syst.* **2023**, *10*, 1496–1507. [CrossRef]
14. Bidram, A.; Davoudi, A.; Lewis, F.L.; Guerrero, J.M. Distributed Cooperative Secondary Control of Microgrids Using Feedback Linearization. *IEEE Trans. Power Syst.* **2013**, *28*, 3462–3470. [CrossRef]
15. Kodagoda, K.R.S.; Wijesoma, W.S.; Teoh, E.K. Fuzzy speed and steering control of an AGV. *IEEE Trans. Control Syst. Techn.* **2002**, *10*, 112–120. [CrossRef]
16. Song, R.; Wei, Q.; Zhang, H.; Lewis, F.L. Discrete-Time Non-Zero-Sum Games with Completely Unknown Dynamics. *IEEE Trans. Cybern.* **2021**, *51*, 2929–2943. [CrossRef]
17. Karg, P.; Kopf, F.; Braun, C.A.; Hohmann, S. Excitation for Adaptive Optimal Control of Nonlinear Systems in Differential Games. *IEEE Trans. Autom. Control* **2023**, *68*, 596–603. [CrossRef]
18. Li, H.; Wei, Q. Initial Excitation-Based Optimal Control for Continuous-Time Linear Nonzero-Sum Games. *IEEE Trans. Syst. Man Cybern. Syst.* **2024**, 1–12. [CrossRef]

19. Nash, J.F. Non-cooperative Games. In *Classics in Game Theory*; Princeton University Press: Princeton, NJ, USA, 1951.
20. Clemhout, S.; Wan, H.Y. Differential games-Economic applications. *Handb. Game Theory Econ. Appl.* **1994**, *2*, 801–825.
21. Zhang, Z.; Xu, J.; Fu, M. Q-Learning for Feedback Nash Strategy of Finite-Horizon Nonzero-Sum Difference Games. *IEEE Trans. Cybern.* **2022**, *52*, 9170–9178. [CrossRef]
22. Savku, E. A Stochastic Control Approach for Constrained Stochastic Differential Games with Jumps and Regimes. *Mathematics* **2023**, *11*, 3043. [CrossRef]
23. Case, J.H. Toward a Theory of Many Player Differential Games. *SIAM J. Control* **1969**, *7*, 179–197. [CrossRef]
24. Werbos, P.J. Approximate dynamic programming for real-time control and neural modeling. In *Handbook of Intelligent Control Neural Fuzzy & Adaptive Approaches*; Van Nostrand Reinhold: New York, NY, USA, 1992, pp. 493–525.
25. Song, R.; Yang, G. Online solving Nash equilibrium solution of N-player nonzero-sum differential games via recursive least squares. *Soft Comput.* **2023**, *27*, 16659–16673. [CrossRef]
26. Zhang, H.; Cui, L.; Luo, Y. Near-Optimal Control for Nonzero-Sum Differential Games of Continuous-Time Nonlinear Systems Using Single-Network ADP. *IEEE Trans. Cybern.* **2013**, *43*, 206–216. [CrossRef]
27. 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]
28. Liu, D.; Li, H.; Wang, D. Online Synchronous Approximate Optimal Learning Algorithm for Multi-Player Non-Zero-Sum Games with Unknown Dynamics. *IEEE Trans. Syst. Man Cybern. Syst.* **2014**, *44*, 1015–1027. [CrossRef]
29. Zhang, H.; Jiang, H.; Luo, C.; Xiao, G. Discrete-Time Nonzero-Sum Games for Multiplayer Using Policy-Iteration-Based Adaptive Dynamic Programming Algorithms. *IEEE Trans. Cybern.* **2017**, *47*, 3331–3340. [CrossRef]
30. Wei, Q.; Zhu, L.; Song, R.; Zhang, P.; Liu, D.; Xiao, J. Model-Free Adaptive Optimal Control for Unknown Nonlinear Multiplayer Nonzero-Sum Game. *IEEE Trans. Neural Netw. Learn. Syst.* **2022**, *33*, 879–892. [CrossRef]
31. Jiang, H.; Zhang, H.; Zhang, K.; Cui, X. Data-driven adaptive dynamic programming schemes for non-zero-sum games of unknown discrete-time nonlinear systems. *Neurocomputing* **2018**, *275*, 649–658. [CrossRef]
32. Zhang, Q.; Zhao, D. Data-Based Reinforcement Learning for Nonzero-Sum Games with Unknown Drift Dynamics. *IEEE Trans. Cybern.* **2019**, *49*, 2874–2885. [CrossRef]
33. Qin, C.; Shang, Z.; Zhang, Z.; Zhang, D.; Zhang, J. Robust Tracking Control for Non-Zero-Sum Games of Continuous-Time Uncertain Nonlinear Systems. *Mathematics* **2022**, *10*, 1904. [CrossRef]
34. Kiumarsi, B.; Lewis, F.L. Actor–Critic-Based Optimal Tracking for Partially Unknown Nonlinear Discrete-Time Systems. *IEEE Trans. Neural Netw. Learn. Syst.* **2015**, *26*, 140–151. [CrossRef]
35. Lv, Y.; Ren, X.; Na, J. Adaptive Optimal Tracking Controls of Unknown Multi-input Systems based on Nonzero-Sum Game Theory. *J. Frankl. Inst.* **2019**, *356*, 8255–8277. [CrossRef]
36. Wen, Y.; Zhang, H.; Su, H.; Ren, H. Optimal tracking control for non-zero-sum games of linear discrete-time systems via off-policy reinforcement learning. *Optim. Control Appl. Methods* **2020**, *41*, 1233–1250. [CrossRef]
37. Qin, C.; Qiao, X.; Wang, J.; Zhang, D.; Hou, Y.; Hu, S. Barrier-Critic Adaptive Robust Control of Nonzero-Sum Differential Games for Uncertain Nonlinear Systems with State Constraints. *IEEE Trans. Syst. Man Cybern. Syst.* **2024**, *54*, 50–63. [CrossRef]
38. Zhao, J. Neural networks-based optimal tracking control for nonzero-sum games of multi-player continuous-time nonlinear systems via reinforcement learning. *Neurocomputing* **2020**, *412*, 167–176. [CrossRef]
39. Modares, H.; Lewis, F.L. Optimal tracking control of nonlinear partially-unknown constrained-input systems using integral reinforcement learning. *Automatica* **2014**, *50*, 1780–1792. [CrossRef]
40. Başar, T.; Olsder, G.J. *Dynamic Noncooperative Game Theory*, 2nd ed.; SIAM: Philadelphia, PA, USA, 1999.
41. Amvoudakis, K.G.V.; Lewis, F.L. Multi-player non-zero-sum games: Online adaptive learning solution of coupled Hamilton–Jacobi equations. *Automatica* **2011**, *47*, 1556–1569. [CrossRef]
42. Kamalapurkar, R.; Klotz, J.R.; Dixon, W.E. Concurrent learning-based approximate feedback-Nash equilibrium solution of N-player nonzero-sum differential games. *IEEE/CAA J. Autom. Sin.* **2014**, *1*, 239–247. [CrossRef]
43. Luo, B.; Wu, H.N.; Huang, T. Off-policy reinforcement learning for H_∞ control design. *IEEE Trans. Cybern.* **2015**, *45*, 65–76. [CrossRef]
44. Jiang, Y.; Jiang, Z.P. Robust Adaptive Dynamic Programming and Feedback Stabilization of Nonlinear Systems. *IEEE Trans. Neural Netw. Learn. Syst.* **2014**, *25*, 882–893. [CrossRef]