Data-Based Reinforcement Learning for Nonzero-Sum Games With Unknown Drift Dynamics

Qichao Zhang, *Member, IEEE*, and Dongbin Zhao, *Senior Member, IEEE*

***Abstract*—This paper is concerned about the nonlinear optimization problem of nonzero-sum (NZS) games with unknown drift dynamics. The data-based integral reinforcement learning (IRL) method is proposed to approximate the Nash equilibrium of NZS games iteratively. Furthermore, we prove that the data-based IRL method is equivalent to the model-based policy iteration algorithm, which guarantees the convergence of the proposed method. For the implementation purpose, a single-critic neural network structure for the NZS games is given. To enhance the application capability of the data-based IRL method, we design the updating laws of critic weights based on the offline and online iterative learning methods, respectively. Note that the experience replay technique is introduced in the online iterative learning, which can improve the convergence rate of critic weights during the learning process. The uniform ultimate boundedness of the critic weights are guaranteed using the Lyapunov method. Finally, the numerical results demonstrate the effectiveness of the data-based IRL algorithm for nonlinear NZS games with unknown drift dynamics.**

***Index Terms*—Integral reinforcement learning (IRL), neural network (NN), nonzero-sum (NZS) games, off-policy, single-critic, unknown drift dynamics.**

I. Introduction

As a main branch of operational research, game theory has been widely used to solve the optimal problem for nonlinear systems with multiplayers [1]. Differential games [2], which focus on the continuous-time game system, have attracted significantly increasing attention in various fields, such as economics [3], marketing [4], computational intelligence [5], control theory [6], [7], and so on. Based on the roles and tasks of players, we can divide differential games into three categories: 1) fully cooperative (FC) games [8]; 2) zero-sum (ZS) games [9]; and 3) nonzero-sum (NZS) games [10]. For the FC game, players are completely cooperative to pursue the team interest and fulfill an overall task. For the ZS game, players are completely competitive to pursue their own interest and compete with each other. For the NZS game, there are both cooperative and competitive players, which want to pursue their individual interest. To obtain the Nash equilibrium [11] of NZS games, we have to solve the coupled Hamilton–Jacobi (HJ) equations. However, it is difficult to solve the HJ equations analytically for nonlinear NZS games due to the nonlinear inherent property.

Recently, reinforcement learning (RL) and adaptive dynamic programming (ADP) [12] based on neural network (NN) technique [13] have been investigated for approximating the Nash equilibrium of differential games [14]–[17]. According to the knowledge of the system dynamics, RL can be divided into model-based RL and model-free RL. For the model-based RL, an online synchronous policy iteration (PI) algorithm was given in [18] and [19], where actor and critic NNs were updated based on the full knowledge of system dynamics. A single network adaptive critic structure instead of actor–critic structure was presented in [20], which was used to solve the optimal control synthesis of nonlinear system. Zhang *et al.* [21] proposed a single-network ADP structure without an initial stabilizing control policy to approach the solution of the HJ equations. It should be mentioned that the integral RL (IRL) technique [22] can be considered as a data-based RL using partially knowledge of system dynamics, where the knowledge of the drift dynamics is not required. The IRL has been investigated for the linear and nonlinear ZS games in [23] and [24]. Luo *et al*. [25] combined the IRL and off-policy scheme for the partially unknown ZS game based on offline iterative learning. Based on the data-based RL with unknown drift dynamics, an online iterative algorithm was employed to solve the coupled algebraic Riccati equations for linear NZS games in [26]. For the nonlinear NZS games with unknown drift dynamics, Kamalapurkar *et al.* [27] proposed a novel actor–critic-identifier structure, where the identifier was used to identify the unknown drift dynamics.

The model-free RL can be divided into two categories: 1) identifier-based RL and 2) data-driven (data-based) RL [28], [29]. In [30]–[32], the actor–critic or single-critic structures with an identifier were constructed for nonlinear NZS games, where the NN identifier was designed to model the unknown system dynamics. For the first time, Zhang *et al.* [33] made a significant breakthrough

Manuscript received September 16, 2017; revised January 30, 2018 and April 24, 2018; accepted April 25, 2018. Date of publication May 16, 2018; date of current version May 7, 2019. This work was supported in part by the National Key Research and Development Plan under Grant 2016YFB0101000, and in part by the National Natural Science Foundation of China under Grant 61573353 and Grant 61533017. This paper was recommended by Associate Editor K.-S. Hwang. *(Corresponding author: Dongbin Zhao.)*

The authors are with the State Key Laboratory of Management and Control for Complex Systems, Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China, and also with School of Artificial Intelligence, University of Chinese Academy of Sciences, Beijing 100049, China (e-mail: zhangqichao2014@ia.ac.cn; dongbin.zhao@ia.ac.cn).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TCYB.2018.2830820

in the optimal robust tracking control for unknown general continuous-time nonlinear systems. The main advantages of the proposed method lie in that only the availability of input/output data is required instead of an exact system model, and meanwhile the tracking error converges to zero asymptotically in an optimal way. The results in this paper provides a solid foundation for using ADP in the field of optimal tracking control of the unknown general nonlinear systems. However, the training of identifier was usually time-consuming and introduced the detrimental identification error inevitably. The data-driven RL is based on the IRL technique and off-policy scheme [34], [35], which has been used in uncertain systems [36], ZS games or H_∞ control [37]–[39], FC games [8], and so on. Note that only one iteration equation for data-driven RL is established rather than two phases in the PI algorithm, where the collected system data is required instead of the knowledge of system dynamics. In [40], the unknown NZS game $\dot{x} = f(x) + \sum_{j=1}^{N} g(x)u_j$ was investigated based on the data-driven RL with actor–critic structure using the online iterative learning scheme. It should be mentioned that the nonlinear dynamics functions $g_j(x)$ for each player u_j were assumed to be the same in [40], which is a special case for the general nonlinear NZS games.

Motivated by the above observations and literature research, this paper proposes a data-based IRL algorithm for general nonlinear NZS games $\dot{x} = f(x) + \sum_{j=1}^{N} g_j(x)u_j(t)$ with unknown drift dynamics based on a single-critic structure. Note that the online iterative learning using current data is usually time-consuming and low efficiency for many practical systems, such as intelligent driving [41], power systems [42], and so on. To improve the application ability for various practical systems, the NN-based offline iterative learning and online iterative learning with experience replay (ER) algorithms are adopted for the data-based IRL, respectively.

The main contributions of this paper are listed as follows.

1) This paper extends the IRL technique for linear NZS games [26] and nonlinear NZS games having a common input-to-state dynamics for all players [40] to the general nonlinear NZS games. A data-based IRL scheme is proposed for nonlinear NZS games without the knowledge of the system drift dynamics. Furthermore, the proposed scheme is proved to be equivalent to the model-based PI, which guarantees the convergence of the proposed algorithm.
2) Different with the online learning based on the actor–critic structure provided in [40], the NN-based offline and online iterative learning algorithms based on single-critic structure are proposed for the designed data-based IRL scheme, which can extend the applicability of the data-based IRL scheme. In addition, the convergence analysis for the offline iterative learning and the online iterative learning with ER are proposed, respectively.

The rest of this paper is organized as follows. Section II introduces the problem formulation of N-player NZS games and the model-based PI algorithm. The data-based IRL for nonlinear NZS games with unknown drift dynamics and its implementation based on the single-critic network structure are proposed in Section III. In Sections IV and V, the offline and online iterative learning algorithms are presented with the convergence analysis, respectively. Simulation results and the conclusion are presented in Sections VI and VII. In order to compare the algorithm performances, the data-based IRL with actor–critic structure is provided in the Appendix.

II. Preliminary

A. Problem Statement

In this paper, we study the general N-player NZS differential games

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

where $x \in R^n$ denotes the state vector, $u_j \in R^{m_j}$ denotes the control vector for player j, and the system nonlinear dynamics $f(\cdot) \in R^n$, $g_j(\cdot) \in R^{n\times m_j}$ are both smooth. Note that the input-to-state dynamics $g_j(x)$ is known continuous vector. Let the set of all players be $\mathbf{N} = \{1, \ldots, N\}$, and the supplementary set of player i be $u_{-i} = \{u_j \mid j \in \{1, \ldots, i-1, i+1, \ldots, N\}\}$.

For the optimal control problem with partially unknown dynamics, the following assumption is commonly used, as in [25] and [26].

Assumption 1: The system drift dynamics $f(x)$ is unknown and Lipschitz continuous on a compact set $\Omega \subseteq R^n$ with $f(0) = 0$.

Define the cost functions associated with player i as

$$\begin{aligned} J_i(x_0, u_i, u_{-i}) &= \int_0^\infty \left(Q_i(x) + \sum_{j=1}^{N} u_j^T R_{ij} u_j \right) dt \\ &= \int_0^\infty r_i(x, u_i, u_{-i}) dt, \ \ i \in \mathbf{N} \end{aligned} \tag{2}$$

where $r_i(x, u_i, u_{-i}) = Q_i(x) + \sum_{j=1}^{N} u_j^T R_{ij} u_j$ with $Q_i(x) = x^T Q_i x$, $Q_i \geq 0$, and $R_{ii} \geq 0$ are positive symmetric matrices, $R_{ij} > 0$ are positive semidefinite symmetric, and $x_0 = x(0)$ denotes the initial state. To simplify the expression, we use x and u_i to represent $x(t)$ and $u_i(t)$ in the following.

For any admissible policy $u_i \in \Phi(\Omega)$, $i \in \mathbf{N}$ defined in [18], the value function for player i is given by

$$\begin{aligned} V_i(x, u_i, u_{-i}) &= \int_t^\infty \left(Q_i(x(\tau)) + \sum_{j=1}^{N} u_j^T(\tau) R_{ij} u_j(\tau) \right) d\tau \\ &= \int_t^\infty r_i(x(\tau), u_i(\tau), u_{-i}(\tau)) d\tau, \ \ i \in \mathbf{N}. \end{aligned} \tag{3}$$

For the NZS games, it aims to obtain an optimal control policy pair $\{u_i^*, u_{-i}^*\} = \{u_1^*, \ldots, u_i, \ldots, u_N^*\}$ to minimize the value functions associated with each player. The optimal policy pair $\{u_1^*, u_{-i}^*\}$ is called Nash equilibrium such that the corresponding value function (3) will increase if any policy u_i^* changes.

Definition 1 (Nash Equilibrium [43]): The N-player NZS game with N-tuple of optimal control policies $\{u_i^*, u_{-i}^*\}$ is said to have a Nash equilibrium solution, if

$$J_i^*\big(u_1^*, \ldots, u_i^*, \ldots, u_N^*\big) \leq J_i\big(u_1^*, \ldots, u_i, \ldots, u_N^*\big), i \in \mathbf{N}. \tag{4}$$

Algorithm 1 PI for NZS Games

```
1: Start with initial admissible policies $\{u_1^0, u_2^0, \ldots, u_N^0\}$.
2: **Policy Evaluation.** Find $V_i^k(x)$ successively approximated by solving
       $$0 = r_i\left(x, u_i^k, u_{-i}^k\right) + \left(\nabla V_i^{k+1}\right)^T \left(f(x) + \sum_{j=1}^{N} g_j(x)u_j^k\right), \quad V_i^k(0) = 0 \qquad (8)$$
   with the iterative index $k = 0, 1, \cdots$.
3: **Policy Improvement.** Update the control policies simultaneously using
       $$u_i^{k+1}(x) = -\frac{1}{2}R_{ii}^{-1}g_i^T(x)\nabla V_i^{k+1}(x). \qquad (9)$$
4: Let $k = k + 1$, go back to Step 2 and continue.
```

To obtain the Nash equilibrium of the NZS game, the solution of the coupled HJ equations should be approached. The detailed description is summarized in the following lemma.

Lemma 1: Assume that the value function (3) is continuously differentiable. Suppose there exists an N-tuple value set $\{V_1^*, \ldots, V_N^*\}$, which is defined as

$$V_i^*(x) = \min_{u_i} \int_t^\infty \left(Q_i(x(\tau)) + \sum_{j=1}^{N} u_j^T(\tau)R_{ij}u_j(\tau)\right)d\tau, i \in \mathbf{N} \tag{5}$$

satisfies the coupled HJ equations

$$Q_i(x) + \left(\nabla V_i^*\right)^T f(x) - \frac{1}{2}\left(\nabla V_i^*\right)^T \sum_{j=1}^{N} g_j(x)R_{jj}^{-1}g_j^T(x) \times \left(\nabla V_j^*\right) + \frac{1}{4}\sum_{j=1}^{N}\left(\nabla V_j^*\right)^T g_j(x)R_{jj}^{-1}R_{ij}R_{jj}^{-1}g_j^T(x)\nabla V_j^* = 0 \tag{6}$$

with $V_i^*(x) \geq 0$, $V_i(0) = 0$ and $\nabla V_i = (\partial V_i(x)/\partial x)$. Then, the optimal control policy for player i is

$$u_i^*(x) = -\frac{1}{2}R_{ii}^{-1}g_i^T(x)\nabla V_i^*, \quad i \in \mathbf{N}. \tag{7}$$

Note that the Nash equilibrium is composed by the optimal control policy $u_i^*(x)$.

B. Policy Iteration for Solving HJ Equations

To obtain the optimal control policies (7), we have to solve the coupled HJ equations (6), which are nonlinear partial differential equations. In fact, it is difficult to obtain the analytical solution of the HJ equations for nonlinear systems. PI is one of the most common methods to overcome this difficulty, which can be described as follows [18] and [32].

According to [32], the PI algorithm is proved to be the quasi-Newton's method, which means Algorithm 1 will converge to the solution of the HJ equations (6) when the iteration goes to infinity, i.e., $V_i^k(x) \to V_i^*(x)$ and $\{u_i^k(x), u_{-i}^k(x)\} \to \{u_i^*(x), u_{-i}^*(x)\}$ as $i \to \infty$. Note that Algorithm 1 is an infinite iterative process for theoretical analysis. For implementation purpose, a small positive number is usually given in step 4 as the threshold value of a termination condition.

Observe that the iterative equation (8) requires the complete knowledge of system dynamics. For the NZS games with the unknown drift dynamics $f(x)$, Algorithm 1 cannot approximate the solution of the HJ equations directly. For the unknown NZS games, the actor–critic-identifier structure is usually adopted [31], [32]. To avoid the time-consuming identification process, an off-policy IRL method is proposed for a special NZS games with unknown dynamics [34], where the input-to-state dynamics for all players are the same. In fact, the input-to-state dynamics for different players are usually disparate for most of real systems. In the following, we propose a data-based IRL approach to solve the general NZS games, where the system drift dynamics $f(x)$ is unknown and each player is allowed to have its own input-to-state dynamics.

III. Data-Based IRL for NZS Games With Unknown Drift Dynamics

In this section, a data-based IRL method for general NZS games is presented, which avoids the identification of $f(x)$. Furthermore, we give the convergence analysis of the proposed algorithm.

A. Data-Based IRL Method

Given arbitrary admissible control policies $u_j \in \Phi(\Omega), j \in \mathbf{N}$, the NZS differential games (1) can be formulated as

$$\dot{x} = f(x) + \sum_{j=1}^{N} g_j(x)\left(u_j - u_j^k\right) + \sum_{j=1}^{N} g_j(x)u_j^k \tag{10}$$

where u_j^k is obtained by (9). The derivative of $V_i^{k+1}(x)$ in (8) with respect to time for the $\{k+1\}$th iteration along the system trajectory (10) is

$$\begin{aligned}\frac{dV_i^{k+1}(x)}{dt} &= \left(\nabla V_i^{k+1}\right)^T\left(f + \sum_{j=1}^{N} g_j(x)u_j^k\right) \\ &\quad + \left(\nabla V_i^{k+1}\right)^T \sum_{j=1}^{N} g_j(x)\left(u_j - u_j^k\right) \\ &= -\, r_i\left(x, u_i^k, u_{-i}^k\right) + \left(\nabla V_i^{k+1}\right)^T \sum_{j=1}^{N} g_j(x)\left(u_j - u_j^k\right).\end{aligned} \tag{11}$$

According to IRL technique, integrating both sides of (11) on time interval $[t, t + \Delta t]$, we have

$$\begin{aligned}&V_i^{k+1}(x(t)) - V_i^{k+1}(x(t+\Delta t)) + \int_t^{t+\Delta t}\left(\nabla V_i^{k+1}(x(\tau))\right)^T \\ &\quad \sum_{j=1}^{N} g_j(x(\tau))\left(u_j(\tau) - u_j^k(\tau)\right)d\tau \\ &= \int_t^{t+\Delta t} r_i\left(x(\tau), u_i^k(\tau), u_{-i}^k(\tau)\right)d\tau.\end{aligned} \tag{12}$$

Observe that the knowledge of system drift dynamics $f(x)$ is not required in (12). Then, Algorithm 1 is translated to the data-based IRL by replacing equations (8) with (12) for NZS games with unknown drift dynamics.

Compared with the ZS games or H_∞ control in [25], the multiple iterative equations (12) rather than only one iterative equation is considered for the players. Accordingly, the N-tuple value set $\{V_1^*, \dots, V_N^*\}$ should be approximated by solving the N-tuple iterative equations in NZS games rather than a common value function for all players. The difference between the ZS games and NZS games is that the relationship between the players is completely competitive or both competitive and cooperative. It should be mentioned that the value function V_i for player i defined in (3) is associated with all the players's policies. From (5), the player i only concerns to minimize the corresponding value function V_i, where it can compete or cooperate with the other players.

Remark 1: For the tracking problem in [44], ZS games or H_∞ control problem in [34] and [39], or the NZS games with uniform input-to-state dynamics [40], the system dynamics $g(x)$ can be relaxed using a transformation of the control policies during the policy improvement. However, the value function $V_i(x)$ and system dynamics $g_i(x)$ are different for each player in general NZS games. Although the term $\{(\delta V_i^{k+1}(x))^T g_i(x)\}$ can be removed using (9), the term $\{(\delta V_i^{k+1}(x))^T g_j(x), j \neq i\}$ still exists in (12). To relax the input-to-state dynamics $g_i(x)$ for general NZS games without identification process still requires further investigation.

Motivated by [25], the equivalence between iterative equations (8) and (12) is established as follows.

Theorem 1: Let $V_i^{k+1}(x) \in C^1(\Omega)$, $C^1(\Omega)$ denotes a function space on Ω with first derivatives continuous, $V_i^{k+1}(x) \geq 0$, $V_i^{k+1}(0) = 0$. $V_i^{k+1}(x)$ is the solution of (12) if and only if it is the solution of the equation (8).

Proof: According to the derivation of (12), we can conclude that the solution V_i^{k+1} of (8) also satisfies (12). If we can prove V_i^{k+1} is a unique solution of (12), (12) is equivalent to (8).

Now, we prove V_i^{k+1} is the unique solution of (12) by contradiction. Suppose that there is another solution $\hbar_i(x)$ of (12) with $\hbar_i(x) \geq 0$ and $\hbar_i(0) = 0$. Thus, $\hbar_i(x)$ also satisfies (11), i.e.,

$$\frac{d\hbar_i(x)}{dt} = -r_i\Big(x, u_i^k, u_{-i}^k\Big) + \nabla\hbar_i^T \sum_{j=1}^{N} g_j(x)\Big(u_j - u_j^k\Big). \tag{13}$$

Substituting (13) from (11), we have

$$\frac{d}{dt}\Big(V_i^{k+1}(x) - \hbar_i(x)\Big) = \Big(\Big(\nabla V_i^{k+1}\Big)^T - \nabla\hbar_i^T\Big) \times \sum_{j=1}^{N} g_j(x)\Big(u_j - u_j^k\Big) \tag{14}$$

for $u_j \in \Phi(\Omega), j \in \mathbf{N}$. As the admissible control policy u_j can be arbitrarily for $u_j \in \Phi(\Omega)$, we choose $u_j = u_j^k$. Then we can obtain

$$\frac{d}{dt}\Big(V_i^{k+1}(x) - \hbar_i(x)\Big) = 0 \tag{15}$$

which means that the term $V_i^{k+1}(x) - \hbar_i(x)$ equals to a real constant for $\forall x \in \Omega$. According to the boundary conditions $V_i^{k+1}(0) = 0, \hbar_i(0) = 0$, we can deduce that $V_i^{k+1}(x) - \hbar_i(x) = 0$, i.e., $V_i^{k+1}(x) = \hbar_i(x)$ for $\forall x \in \Omega$. Then, $\{V_1^{k+1}, \dots, V_N^{k+1}\}$ is the unique solution set of (12) for all players, which means (12) is equivalent to (8). This completes the proof. ■

According to [32], Algorithm 1 can approximate the optimal value functions and control policies. Based on Theorem 1, we know that the data-based IRL method for partially unknown NZS games is equivalent to Algorithm 1 for completely known NZS games. That is to say, the convergence of the proposed method for NZS games with unknown drift dynamics is guaranteed.

Remark 2: According to Theorem 1, the data-based IRL is equivalent to Algorithm 1. Based on the property of quasi-Newton's method, the data-based IRL is also a local optimization scheme, which is similar with the method in [32], [39], and [40].

B. Off-Policy IRL Algorithm With Single-Critic Structure

For the implementation purpose, a single-critic NN approximation is introduced to approach the solution of (12). According to the Weirstrass high-order approximation theorem, a smooth function can be uniformly approximated on a compact set by NN

$$V_i^k(x) = w_{i,k}^T \phi_i(x) + \varepsilon_{i,k} \tag{16}$$

where $\phi_i{:}R^n \to R^{K_i}$ is linear independent basis function vector, $w_{i,k} \in R^{K_i}$ is the unknown weight vector, K_i is the number of hidden neurons, and $\varepsilon_{i,k}$ denotes the reconstruction error for $i \in \mathbf{N}$. It is shown in [45] that as $K_i \to \infty$, the reconstruction error $\varepsilon_{i,k}$ converges to zero.

According to (16), we rewrite the iteration equation (12) as

$$\begin{aligned}(\phi_i(x + \Delta t) - \phi_i(x))^T w_{i,k+1} &\\ - \int_t^{t+\Delta t} \sum_{j=1}^{N} \Big(g_j(x)\Big(u_j(\tau) - u_j^k(\tau)\Big)\Big)^T \nabla\phi_i^T(x) w_{i,k+1} d\tau &\\ + \int_t^{t+\Delta t} Q_i(x) + \sum_{j=1}^{N} \Big(\Big(u_j^k(\tau)\Big)^T R_{ij} u_j^k(\tau)\Big) d\tau &\\ = \zeta_{i,k+1}(x(t)) &\end{aligned} \tag{17}$$

where

$$\begin{aligned}\zeta_{i,k+1}(x(t)) = {}& \varepsilon_{i,k+1}(x(t)) - \varepsilon_{i,k+1}(x(t + \Delta t)) \\ & + \int_t^{t+\Delta t} \sum_{j=1}^{N} \Big(g_j(x)\Big(u_j(\tau) - u_j^k(\tau)\Big)\Big)^T \\ & \times \nabla\varepsilon_{i,k+1}(x) d\tau.\end{aligned}$$

Denote $\hat{w}_{i,k}$ as the estimations of the unknown weight vector $w_{i,k}$. Then the output of the critic NN approximation is

$$\hat{V}_i^k(x) = \hat{w}_{i,k}^T \phi_i(x). \tag{18}$$

Based on (9), the approximate control policies are

$$\hat{u}_i^k(x) = -\frac{1}{2} R_{ii}^{-1} g_i^T(x) \nabla\phi_i^T(x) \hat{w}_{i,k}, \quad i \in \mathbf{N}. \tag{19}$$

Remark 3: Since the input-to-state dynamics $g_j(x)$ are known in this paper, we can obtain the approximated control policies (19) based on the critic NN approximation (18) directly. Hence, the single-critic structure rather than the actor–critic structure is adopted, which can save the computational burden and eliminate the approximation error resulting from action NNs. To compare the effectiveness of two structures in Section VI, the off-policy IRL with actor–critic structure is given in the Appendix.

Using $\hat{V}_i^{k+1}(x)$ to replace $V_i^{k+1}(x)$ in (12), due to the existence of the truncation error of the estimated solution, the residual error for the player i is obtained as

$$\begin{aligned} & e_i^{k+1}(x(\tau), u_i(\tau), u_{-i}(\tau)) \\ & = (\phi_i(x(t)) - \phi_i(x(t+\Delta t)))^T \hat{w}_{i,k+1} \\ & \quad + \int_t^{t+\Delta t} \sum_{j=1}^{N} \Big(g_j(x) \Big(u_j(\tau) - u_j^k(\tau) \Big) \Big)^T \nabla \phi_i^T(x) \hat{w}_{i,k+1} d\tau \\ & \quad - \int_t^{t+\Delta t} Q_i(x) d\tau - \int_t^{t+\Delta t} \sum_{j=1}^{N} \Big(\Big(u_j^k(\tau) \Big)^T R_{ij} u_j^k(\tau) \Big) d\tau \\ & \triangleq e_i^{k+1}(t). \end{aligned} \tag{20}$$

Let

$$\begin{aligned} \rho_i(x(t), u_i(t), u_{-i}(t)) & \triangleq (\phi_i(x(t)) - \phi_i(x(t+\Delta t)))^T \\ & \quad + \int_t^{t+\Delta t} \sum_{j=1}^{N} \Big(g_j(x) \Big(u_j(\tau) - u_j^k(\tau) \Big) \Big)^T \\ & \quad \times \nabla \phi_i^T(x) d\tau \\ \pi_i(x(t)) & \triangleq \int_t^{t+\Delta t} Q_i(x) + \sum_{j=1}^{N} \Big(\Big(u_j^k(\tau) \Big)^T R_{ij} u_j^k(\tau) \Big) d\tau. \end{aligned} \tag{21}$$

Note that

$$\begin{aligned} & \int_t^{t+\Delta t} \sum_{j=1}^{N} \Big(g_j(x) \Big(u_j(\tau) - u_j^k(\tau) \Big) \Big)^T \nabla \phi_i^T(x) \hat{w}_{i,k+1} d\tau \\ & = \int_t^{t+\Delta t} \sum_{j=1}^{N} \Big(u_j^T(\tau) g_j^T(x) \Big) \nabla \phi_i^T(x) \hat{w}_{i,k+1} d\tau \\ & \quad + \frac{1}{2} \int_t^{t+\Delta t} \sum_{j=1}^{N} \Big(\hat{w}_{i,k}^T \nabla \phi_j(x) g_j(x) R_{jj}^{-1} g_j^T(x) \Big) \\ & \quad \times \nabla \phi_i^T(x) \hat{w}_{i,k+1} d\tau \end{aligned}$$

and

$$\begin{aligned} & \int_t^{t+\Delta t} \sum_{j=1}^{N} \Big(\Big(u_j^k(\tau) \Big)^T R_{ij} u_j^k(\tau) \Big) d\tau \\ & = \frac{1}{4} \int_t^{t+\Delta t} \sum_{j=1}^{N} \Big(\hat{w}_{j,k}^T \nabla \phi_j(x) g_j(x) R_{jj}^{-1} R_{ij} \\ & \qquad \times R_{jj}^{-1} g_j^T(x) \nabla \phi_j^T(x) \hat{w}_{j,k} \Big) d\tau. \end{aligned}$$

Let

$$\begin{aligned} D_{i,j}(x) & \triangleq \nabla \phi_j(x) g_j(x) R_{jj}^{-1} g_j^T(x) \nabla \phi_i^T(x) \\ E_{i,j}(x) & \triangleq \nabla \phi_j(x) g_j(x) R_{jj}^{-1} R_{ij} R_{jj}^{-1} g_j^T(x) \nabla \phi_j^T(x) \\ \eta_1(x(t)) & \triangleq (\phi_i(x(t)) - \phi_i(x(t+\Delta t)))^T \\ \eta_2(x(t), u_i, u_{-i}) & \triangleq \int_t^{t+\Delta t} \left(\sum_{j=1}^{N} u_j^T(\tau) g_j^T(x) \right) \nabla \phi_i^T(x) d\tau \\ \eta_3(x(t)) & \triangleq \begin{bmatrix} \int_t^{t+\Delta t} D_{i1}(x) d\tau \\ \vdots \\ \int_t^{t+\Delta t} D_{iN}(x) d\tau \end{bmatrix} \\ \eta_4(x(t)) & \triangleq \begin{bmatrix} \int_t^{t+\Delta t} E_{i,1}(x) d\tau & 0 & 0 \\ 0 & \ddots & \vdots \\ 0 & \cdots & \int_t^{t+\Delta t} E_{i,N}(x) d\tau \end{bmatrix} \\ \eta_5(x(t)) & \triangleq \int_t^{t+\Delta t} Q_i(x) d\tau. \end{aligned}$$

Then, one can get

$$\begin{aligned} \rho_i(x(t), u_i(t), u_{-i}(t)) & = \eta_1(x(t)) + \eta_2(x(t), u_i, u_{-i}) \\ & \quad + \frac{1}{2} \hat{W}_k^T \eta_3(x(t)) \\ \pi_i(x(t)) & = \frac{1}{4} \hat{W}_k^T \eta_4(x(t)) \hat{W}_k + \eta_5(x(t)) \end{aligned}$$

where $\hat{W}_k = [\hat{w}_{1,k}^T, \ldots, \hat{w}_{N,k}^T]^T$.

Therefore, we can rewrite (20) as

$$e_i^{k+1}(t) = \rho_i(x(t),\ u_i(t), u_{-i}(t)) \hat{w}_{i,k+1} - \pi_i(x(t)). \tag{22}$$

It should be mentioned that (22) is the key for the proposed data-based IRL algorithm with unknown drift dynamics. In the next section, the offline and online iterative learning schemes are proposed to approach the ideal critic weight $w_{i,k}$ using $\hat{w}_{i,k}$ by minimizing the square error $(1/2)(e_i^k)^T e_i^k$.

IV. Offline Iterative Learning

A. Offline Iterative Learning Algorithm

For the developed offline learning algorithm, we use the least-square (LS) scheme to update the estimated critic weight vectors. Define a strictly increasing time sequence $\{t_m\}_{m=0}^{q}$, where q denotes the number of collected samples. Define the sample set as $M_i = \{(x_m, u_{i,m}, u_{-i,m})\}_{m=0}^{q}$. For description simplicity, denote $\rho_{i,m} = \rho_i(x_m, u_{i,m}, u_{-i,m})$ and $\pi_{i,m} = \pi_i(x_m)$. To ensure the convergence of the estimated weight vector $\hat{w}_{i,k+1}$, the following persistency of excitation (PE) assumption is given.

Assumption 2: Let $\rho_{i,m}$ be persistently existed. If there exist $q_0 > 0$ and $\delta > 0$ such that for all $q \le q_0$, one can get

$$\frac{1}{q} \sum_{k=0}^{q-1} \rho_{i,m} \rho_{i,m}^T \ge \delta I_{i,m}$$

with the identity matrix $I_{i,m}$.

For the offline learning algorithm, we aim to update the estimated weight vector $\hat{w}_{i,k+1}$ by minimizing the square

Algorithm 2 Offline Iterative Learning for NZS Games

```
1: For NZS games, select any initial admissible control policies $\{u_i, u_{-i}\}$. Collect the available system data $(x_m, u_i, u_{-i})$ for sample set $M$, then compute $\eta_1(x_m)$, $\eta_2(x_m, u_i, u_{-i})$, $\eta_3(x_m)$, $\eta_4(x_m)$ and $\eta_5(x_m)$;
2: For each player $i$, select initial critic NN weight vector $\hat{w}_{i,0}$. Let the iteration index $k = 0$;
3: Compute $P_i$ and $\Pi_i$, and update $\hat{w}_{i,k+1}$ for each player using (23);
4: Let $k = k+1$, if $\|\hat{w}_{i,k+1} - \hat{w}_{i,k}\|^2 \le \epsilon$ ($\epsilon$ is a small positive number), stop iteration and $\hat{w}_{i,k}$ is employed to obtain the control policy (19), else go back to Step 3 and continue.
```

error $(1/2)(e_{i,m}^{k+1})^T e_{i,m}^{k+1}$. Based on the Monte Carlo integration method in [25], the updating law of the estimated weight vector is given by

$$\hat{w}_{i,k+1} = [P_i^T P_i]^{-1} P_i^T \Pi_i \tag{23}$$

where

$$P_i = \big[\rho_{i,0}, \dots, \rho_{i,q-1}\big]^T; \quad \Pi_i = [\pi_{i,0}, \dots, \pi_{i,q-1}]^T.$$

According to the updating law (23), we present the offline iterative learning algorithm for the data-based IRL in Algorithm 2. It includes the measurement phase and offline learning phase, where the measurement phase of step 1 is used to collect the available system data and the offline learning phase of steps 2–4 is used to approach the ideal weight vectors. Next, the control policies (19) based on the approximate ideal weight vectors can be applied to the real-time NZS games.

Remark 4: Note that the PE assumption is to guarantee the existence of the inverse matrix $[P_i^T \quad P_i]^{-1}$. For the real implementation, exploration noises, such as random noises, sinusoidal function with different frequencies, and so on, are usually added to the given control inputs to satisfy the PE condition. Meanwhile, a large size of sample set M_i can also guarantee the richness of sample data. However, the PE condition is not easy to check. How to choose the number of samples q and the exploration noises are generally experience-based.

B. Convergence Analysis for Offline Iterative Learning

For the NN-based offline iterative learning algorithm, it is necessary to analyze the convergence of the critic weights, which is given in Theorem 2.

Theorem 2: Suppose that Assumption 2 holds and V_i^{k+1} is the solution of the iteration equation (12). For $\forall\beta > 0$, there exists an integer $K_i^* > 0$ such that $K_i > K_i^*$, then:

1) $\sup_{x\in\Omega} |\hat{V}_i^{k+1}(x) - V_i^{k+1}(x)| < \beta$;

2) $\sup_{x\in\Omega} |\hat{V}_i^{k+1}(x) - V_i^{*}(x)| < \beta$.

Proof: 1) Define the weight estimation error as $\tilde{w}_{i,k+1} = \hat{w}_{i,k+1} - w_{i,k+1}$. From (23), we have

$$P_i^T P_i \tilde{w}_{i,k+1} = P_i^T \Pi_i - P_i^T P_i w_{i,k+1}. \tag{24}$$

Multiplying $\tilde{w}_{i,k+1}^T$ on both sides of (24) yields

$$P_i^T P_i \|\tilde{w}_{i,k+1}\|^2 = \tilde{w}_{i,k+1}^T P_i^T \big(\Pi_i - P_i w_{i,k+1}\big). \tag{25}$$

According to Assumption 2, we have $P_i^T P_i \|\tilde{w}_{i,k+1}\|^2 \ge \delta q I_{i,m} \|\tilde{w}_{i,k+1}\|^2$.

According to the definition of P_i, Π_i, (17) and (21), we have

$$\begin{aligned} &P_i^T\big(\Pi_i - P_i w_{i,k+1}\big) \\ &= \sum_{m=0}^{q-1}\Bigg[\rho_{i,m}^T\Bigg(\Big(\phi_i(x(t_{m+1})) - \phi_i(x(t_m))\Big)^T w_{i,k+1} \\ &\quad - \int_{t_m}^{t_{m+1}} \sum_{j=1}^{N}\Big(g_j(x)\Big(u_j - u_j^k\Big)\Big)^T \nabla\phi_i^T(x) w_{i,k+1} d\tau \\ &\quad + \int_{t_m}^{t_{m+1}} Q_i(x) + \sum_{j=1}^{N}\Big(\Big(u_j^k\Big)^T R_{ij} u_j^k\Big) d\tau\Bigg)\Bigg] \\ &= \sum_{m=0}^{q-1}\big(\rho_{i,m}^T \zeta_{i,k+1}(x_m)\big) \end{aligned}$$

where $\zeta_{i,k+1}(x_m)$ denotes the residual error for the time interval $[t_m, t_{m+1}]$ instead of $[t, t+\Delta t]$ for (17).

Based on (25), we have

$$\begin{aligned} \delta q \big\|\tilde{w}_{i,k+1}\big\|^2 &\le \big\|\tilde{w}_{i,k+1}\big\| \sum_{m=0}^{q-1} \big\|\rho_{i,m}^T\big\| \big|\zeta_{i,k+1}(x_m)\big| \\ &\le \big\|\tilde{w}_{i,k+1}\big\| \sum_{m=0}^{q-1} \big\|\rho_{i,m}^T\big\| \zeta_{i,\max} \end{aligned} \tag{26}$$

where $\zeta_{i,\max}$ denotes the bound of $\zeta_{i,k+1}$. Note that $\lim_{K_{i,k+1}\to\infty} \zeta_{i,k+1}(x_m) = 0$. Based on (26), we have $\lim_{K_{i,k+1}\to\infty} \tilde{w}_{i,k+1} = 0$.

As

$$\hat{V}_i^{k+1} - V_i^{k+1} = \tilde{w}_{i,k+1}^T \phi_i(x) - \varepsilon_{i,k+1}. \tag{27}$$

As $\lim_{K_{i,k+1}\to\infty} \varepsilon_{i,k+1} = 0$, we can get

$$\lim_{K_{i,k+1}\to\infty} \hat{V}_i^{k+1} = V_i^{k+1}. \tag{28}$$

That is to say, there exists an integer $K_i^* > 0$ for $\forall x \in \Omega$, $\beta > 0$ such that if $K_i > K_i^*$, then

$$\Big|\hat{V}_i^{k+1}(x) - V_i^{k+1}(x)\Big| < \beta.$$

2) According to [45, Ths. 3 and 4], the result of $\sup_{x\in\Omega} |\hat{V}_i^{k+1}(x) - V_i^*(x)| < \beta$ can be proven directly. Some similar proof steps are omitted for avoidance of repetition.

This completes the proof. ■

Remark 5: According to the conclusion of Theorem 2, we known that the critic weight estimation error converges to zero as the number of hidden neurons goes to infinite. For arbitrarily $K_i > K_i^*$, the critic weight estimation error $\tilde{w}_{i,k+1}$

satisfies UUB. Then, we can get the control policies (19) will converge to the Nash equilibrium based on [30, Th. 3].

V. Online Iterative Learning

A. Online Iterative Learning Algorithm

For the online learning algorithm, we use the gradient descent method to update the estimated critic weight vectors. For the ER technique, the current data and past data are both used to approach the critic NNs' weights. As the estimated critic weight vectors are learned continuously, we can replace $w_{i,k+1}$, e_i^{k+1} with w_i, e_i, respectively.

According to (22), let the residual errors at interval $[t_d, t_{d+1}]$ be

$$e_i(t_d) = \rho_i(t_d)\hat{w}_i + \pi_i(t_d). \tag{29}$$

It is aimed to update the estimated weight vector $\hat{w}_{i,k+1}$ by minimizing the square error

$$E_i = \frac{1}{2}(e_i(t))^T e_i(t) + \frac{1}{2}\sum_{d=1}^{l}(e_i(t_d))^T e_i(t_d).$$

Condition 1: Define $D_i = [\rho_i(t_d), \rho_i(t_{d+1}), \ldots, \rho_i(t_{d+l})]$ as the recorded data matrix for each player i. There are as many linearly independent elements as the number of corresponding critic NN's hidden neurons for the recorded data matrix D_i, such that $\operatorname{rank}(D_i) = K_i$.

The estimated weight vector of the critic NN is updated using the gradient descent scheme and ER, which is given by

$$\begin{aligned}\dot{\hat{w}}_i = -\alpha_i \Bigg[& \frac{\rho_i^T(t)}{\big(1+\rho_i^T(t)\rho_i(t)\big)^2}\big(\rho_i\hat{w}_i + \pi_i(t)\big) \\ & + \sum_{d=1}^{l}\frac{\rho_i^T(t_d)}{\big(1+\rho_i^T(t_d)\rho_i(t_d)\big)^2}\big(\rho_i(t_d)\hat{w}_i + \pi_i(t_d)\big)\Bigg].\end{aligned} \tag{30}$$

B. Convergence Analysis for Online IRL With ER

The following theorem demonstrates the UUB of the weights estimation error of critic NNs using Lyapunov method.

Theorem 3: If recorded data D_i for each critic NN satisfy Condition 1, the critic weights estimation error $\tilde{w}_{i,k+1}$ is UUB and the system state x is asymptotically stable.

Proof: Choose Lyapunov function candidate as

$$L_i = \frac{1}{2\alpha_i}\tilde{w}_i^T\tilde{w}_i.$$

Its time derivative is

$$\dot{L}_i = \frac{1}{\alpha_i}\tilde{w}_i^T\dot{\tilde{w}}_i. \tag{31}$$

Note that

$$\begin{aligned}\dot{\tilde{w}}_i &= \dot{\hat{w}}_i \\ \rho_i(t)\hat{w}_i + \pi_i(t) &= \rho_i(t)\tilde{w}_i + \pi_i(t) + \rho_i(t)w_i \\ &= \rho_i(t)\tilde{w}_i - \zeta_i(t).\end{aligned}$$

Then, (31) can be written as

$$\begin{aligned}\dot{L}_i = \tilde{w}_i^T \Bigg[& \frac{\rho_i^T(t)}{\big(1+\rho_i^T(t)\rho_i(t)\big)^2}(\rho_i(t)\tilde{w}_i - \zeta_i(t)) \\ & + \sum_{d=1}^{l}\frac{\rho_i^T(t_d)}{\big(1+\rho_i^T(t_d)\rho_i(t_d)\big)^2}(\rho_i(t_d)\tilde{w}_i - \zeta_i(t_d))\Bigg] \\ = & \left[\bar{\rho}_i^T\bar{\rho}_i + \sum_{d=1}^{l}\bar{\rho}_i^T(t_d)\bar{\rho}_i(t_d)\right]\|\tilde{w}_i\|^2 \\ & - \left[\frac{\bar{\rho}_i^T(t)}{m(t)}\zeta_i(t) + \sum_{d=1}^{l}\frac{\bar{\rho}_i^T(t_d)}{m(t_d)}\zeta_i(t_d)\right]|\tilde{w}_i|\end{aligned} \tag{32}$$

where $\bar{\rho}_i = \rho_i/(1+\rho_i^T\rho_i)$ and $m = 1+\rho_i^T\rho_i$. Denote that $H_i = \bar{\rho}_i^T(t)\bar{\rho}_i(t) + \sum_{d=1}^{l}\bar{\rho}_i^T(t_d)\bar{\rho}_i(t_d)$ and $\zeta_B = [(\bar{\rho}_i^T(t))/m(t)]\zeta_i(t) + \sum_{d=1}^{l}[(\bar{\rho}_i^T(t_d))/m(t_d)]\zeta_i(t_d)$. If Condition 1 is satisfied, then $H_i > 0$. Therefore, $\dot{L}_i$ is negative definite provided that

$$\|\tilde{w}_{i,k+1}\| > \frac{(l+1)\zeta_{i,\max}}{\lambda_{\min}(H_i)}. \tag{33}$$

For the NZS games, define Lyapunov function candidate as (3) with the feedback control policy (7). Take the time derivative to obtain

$$\dot{V}_i = -Q_i(x) - \sum_{j=1}^{N}u_j^T R_{ij}u_j < 0.$$

That is to say, $V_i(x)$ for each player i is a Lyapunov function. The closed-loop system is asymptotically stable. This completes the proof. ■

Remark 6: Similarly with Remark 5, suppose that the hypotheses of Theorem 3 holds, the obtained control policies (19) can converge to the approximate Nash equilibrium solution of the NZS games.

Remark 7: Observe that the Condition 1 for ER is used to guarantee the matrix H_i to be positive in the online iterative learning. It is similar with the PE assumption in the offline iterative learning. In fact, the Condition 1 is a PE-like condition, which can be checked online easily [30]. During the learning process, the ER can improve the convergence rate under the persistent exciting input signals. Compared with the on-policy ER, the ER performs a more positive role in the off-policy scheme [44].

Remark 8: According to Remark 2, the data-based IRL is a local optimization method. For the proposed offline and online iterative learning algorithms, the value function $\hat{V}_i^0(x)$ computed by the initial weights of critic NNs should be located in a neighborhood of the solution of the HJ equations (6). The initial weights of critic NNs are choose based on experience in most of the off-policy RL algorithms [8], [25], [34], [38], [39].

VI. Simulation Study

Consider the two-player nonlinear NZS differential games as follows [18]:

$$\dot{x} = f(x) + g(x)u + k(x)w \tag{34}$$

where

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

$$g(x) = \begin{bmatrix} 0 \\ \cos(2x_1)+2 \end{bmatrix}, \quad k(x) = \begin{bmatrix} 0 \\ \sin\left(4x_1^2\right)+2 \end{bmatrix}$$

$x = [x_1, x_2]^T \in R^2$ is the state vector, and $u, w \in R$ are the control inputs.

The computer processor is Intel Core i5-4570 CPU @3.20 GHz, and the simulation platform is MATLAB R2014a. Select $Q_1(x) = 2x^T x$, $Q_2(x) = x^T x$, $R_{11} = R_{12} = 2I$, and $R_{21} = R_{22} = I$. Note that I denotes an identity matrix with appropriate dimensions. Based on [18], we can get the optimal value functions $V_1^*(x) = 0.5x_1^2 + x_2^2$ and $V_2^*(x) = 0.25x_1^2 + 0.5x_2^2$. For the data-based IRL algorithm, we choose the activation functions of the two NN approximators as

$$\phi_{c1}(x) = \phi_{c2}(x) = \left[x_1^2 \; x_1x_2 \; x_2^2\right]^T.$$

Then, we can obtain the ideal weights

$$w_{c1} = [0.5 \; 0.0 \; 1.0]^T; \quad w_{c2} = [0.25 \; 0.0 \; 0.5]^T.$$

A. Offline Iterative Learning

In this section, we compare the effects of the offline iterative learning with single-critic structure and with actor–critic structure. Let the initial state vector be $x_0 = [2, -2]^T$. The hyper-parameters setting is given as follows. Set the initial probing control inputs $u_1' = 0.7e^{-0.006t}\sin(t)^2\cos(t) + \sin(2t)^2\cos(0.1t) + \sin(-1.2t)^2\cos(0.5t) + \sin(t)^5 + 0.5(x_1 + x_2)(\cos(2x_1) + 2)$ and $u_2' = 0.7e^{-0.006t}\sin(t)^2\cos(t) + \sin(2t)^2\cos(0.1t) + \sin(1.12i)^2 + \cos(2.4t)\sin(2.4t)^3 + (x_1 + x_2)(\sin(4x_1) + 2)$. The integral time interval is 0.1 s. We choose the length index $q = 200$. That is to say, the online data collection phase will last 20 s. For the single-critic structure in Section IV, the initial weights of the critic NNs are initialized randomly in [0, 1]. For the actor–critic structure in the Appendix, the initial weights of critic NNs are the same as the ones with single-critic structure. The activation functions of the actor NNs in (35) are chosen as $\varphi_1(x) = [x_1^2 \; x_1x_2 \; x_2^2 \; x_1^3 \; x_2^3]$, and the initial weights of actor NNs are chosen randomly in [0, 1]. The iterative termination condition is $\|\hat{w}_{i,k+1} - \hat{w}_{i,k}\|^2 \leq$ with $\epsilon = 10^{-6}$.

The convergence curves of w_{ci} are given in Figs. 1 and 2. It can be seen that the estimated weights of actor–critic and single-critic structures can both converge to

$$\hat{w}_{c1} = [0.4896 \quad 0.091 \quad 1.0113]^T$$
$$\hat{w}_{c2} = [0.2216 \quad 0.053 \quad 0.5158]^T$$

which are approximate ideal values. However, the number of iterations for the single-critic structure is 4, while the one for the actor–critic structure is 7. In addition, the recorded computation time for the offline learning is 5.0121 s for single-critic structure and 8.0635 s for the actor–critic structure, where less computational burden is required for the single-critic structure. The similar approached results are shown based on the online learning scheme in [18]. Compared with [18], the knowledge of drift dynamics is not required in the proposed data-based IRL algorithm. The trajectories of system state, the control inputs u and w are shown in Figs. 3 and 4, respectively. We can see the system state is stable under the obtained optimal controllers.

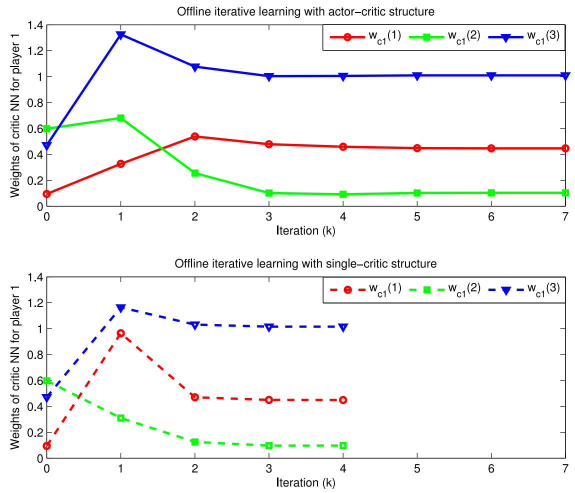


Fig. 1. Weights w_{c1} of critic NN for player 1.

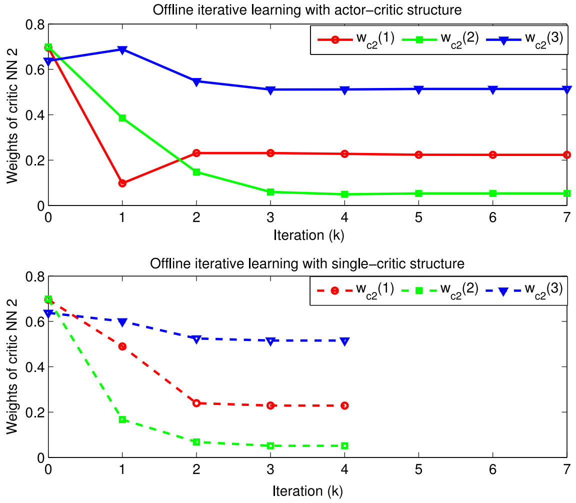


Fig. 2. Weights w_{c2} of critic NN for player 2.

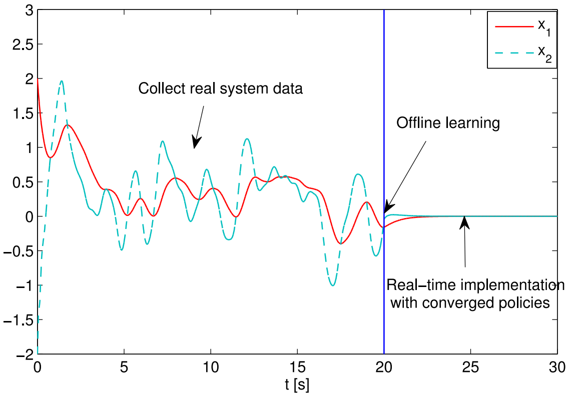


Fig. 3. Trajectories of system state.

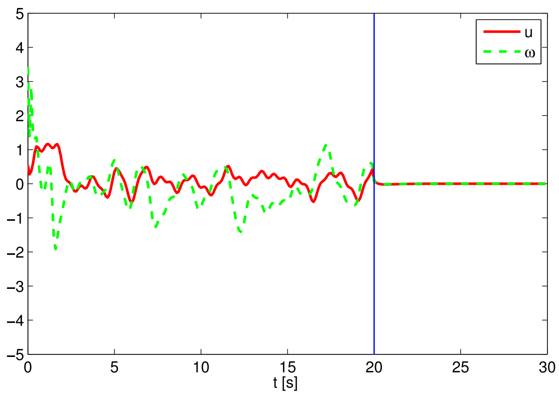


Fig. 4. Trajectories of control inputs.

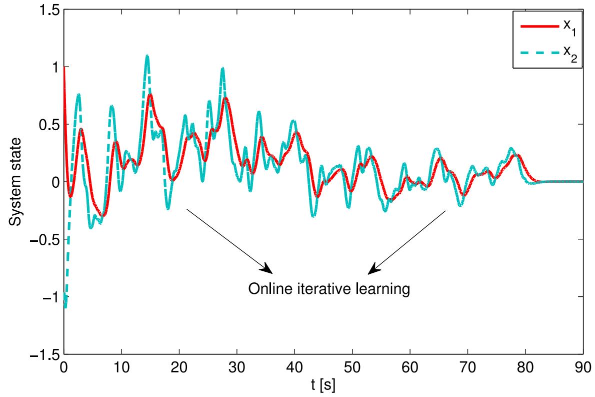


Fig. 5. Trajectories of system state.

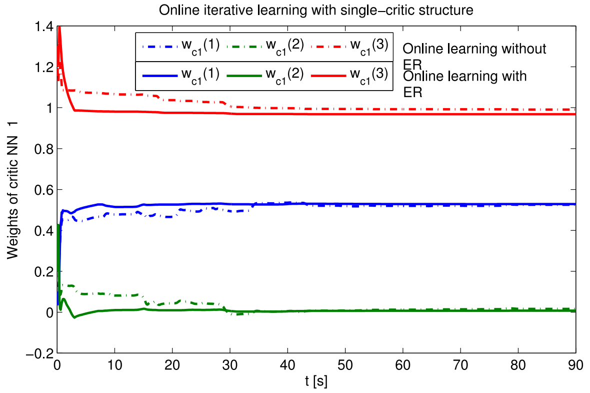


Fig. 6. Weights w_{c1} of critic NN for player 1.

B. Online Iterative Learning

Set the initial state vector as $x_0 = [1, -1]^T$. The hyperparameters setting is given as follows. Let the initial probing control inputs $u_1' = e^{-0.2t}\sin(t)^2\cos(t) + \sin(2t)^2\cos(0.1t) + \sin(-1.2t)^2\cos(0.5t) + \sin(t)^5 + 0.5(x_1 + x_2)(\cos(2x_1) + 2)$ and $u_2' = e^{-0.2t}\sin(t)\cos(t)+\sin(3t)^2\cos(0.1t)+\sin(1.12i)^2+\cos(2.4t)\sin(2.4t)^3+(x_1+x_2)(\sin(4x_1)+2)$. Let the experience set size be $l = 20$ and the integral time interval be 0.1 s. Note that the initial probing control inputs are removed at 80 s. The learning rates are $\alpha_1 = 2$ and $\alpha_2 = 4$.

For the online adaptation law (30) with single-critic structure, the activation functions of critic NNs are the same with

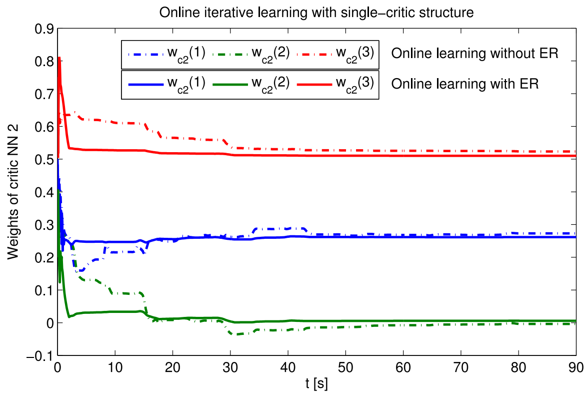


Fig. 7. Weights w_{c2} of critic NN for player 2.

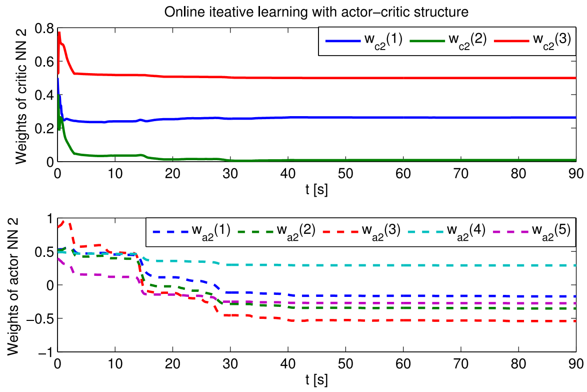


Fig. 8. Weights of actor–critic NNs for player 1.

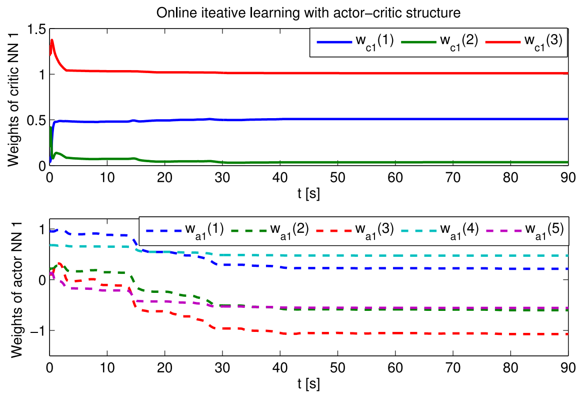


Fig. 9. Weights of actor–critic NNs for player 2.

the offline iterative learning algorithm. The trajectories of system state is shown in Fig. 5. The system states are converged rapidly after 80 s. The estimated weights for player 1 and player 2 for the online learning with ER technique are

$$\hat{w}_{c1} = [0.5156\ 0.0114\ 0.9906]^T$$
$$\hat{w}_{c2} = [0.2592\ 0.0111\ 0.4901]^T.$$

Compared with the online learning without ER technique, we can see that the convergence rate of critic NNs is improved effectively in Figs. 6 and 7.

For the online adaptation law (39) with actor–critic structure, the activation functions of critic and actor NNs are the same with the offline iteration learning algorithm. The weights of actor–critic NNs for player 1 and 2 are shown in Figs. 8 and 9, respectively. Compared with Figs. 6 and 7, the convergence curves of critic NNs are similar. However, the recorded computation time for the online learning with ER is 43.5612 s for single-critic structure and 48.7362 s for the actor–critic structure. Thus, the simulation results prove the effectiveness of the proposed online data-based IRL method with single-critic structure.

VII. Conclusion

In this paper, we propose the data-based IRL algorithm for the nonlinear NZS games with unknown drift dynamics. For the implementation purpose, NN-based offline iterative learning and online iterative learning algorithms based on the single-critic structure are proposed, where the convergence analysis are given respectively. Finally, numerical results are given to demonstrate the effectiveness of the developed algorithms.

The approaches to tradeoff exploration and exploitation in RL fall into two main classes: 1) on-policy and 2) off-policy. For on-policy methods such as Sarsa, Monte Carlo policy gradient, etc., the value function is estimated based on the evaluating policy while using it to control systems and generate data simultaneously. Although on-policy methods can offer nearly unbiased estimates of the policy gradient, they only use the on-policy samples during the learning process which means that a large number of samples is required. Therefore, on-policy methods are usually sample intensive. Meanwhile, the evaluating policy has to complete the exploration and exploitation tasks, which usually makes the on-policy learning process time-consuming. In off-policy methods such as *Q*-learning, off-policy actor–critic, etc., the behavior policy used to generate data may be unrelated to the evaluating policy that is improved. An advantage of this separation is that the evaluating policy may be deterministic, while the behavior policy can continue to sample all possible actions [46]. All of the samples including past samples can be used efficiently. However, convergence of such methods is difficult to guarantee with nonlinear function approximators [47]. In this paper, we prove the proposed data-based IRL for NZS games is equivalent to the model-based PI algorithm, which guarantees the convergence of the proposed method.

The learning approaches to train the weights of NNs can be divided into two major categories: 1) online learning and 2) offline learning. For online learning methods, the generated sample at each time step is used to train at once rather than collecting a static dataset in offline learning methods. Note that most of the on-policy methods are implemented online [25]. In fact, the online off-policy IRL can be considered as a batch online learning, where the samples generated during each integral interval are a batch data. Online learning could adapt to the new or unseen data. Meanwhile, a small memory footprint is used without storing a large static data set for online learning. However, online learning is impractical for many expensive or dangerous control systems, such as unmanned driving, industrial control systems, and so on. For the offline learning, a static data set is collected beforehand, and the collected data set can be utilized repeatedly for different hyperparameter settings. Although it is safe and effective, the offline learning is not sensitive enough to the real-time process especially for the new or unseen data. Combined with the above discussion, we propose the offline and online iterative learning algorithms to improve the applicability of the data-based IRL scheme.

Appendix
Off-Policy IRL With Actor–Critic Structure

To compare with the single-critic structure, we give the off-policy IRL with actor–critic structure in the Appendix. The offline and online iterative learning algorithms are proposed, respectively. Note that the approximated value functions is described in (18). To approach the control policy, the actor NN approximation for each player i is designed as

$$\hat{u}_i^k(x) = \hat{w}_{a_i,k}^T \varphi_i(x) \tag{35}$$

where $\hat{w}_{a_i,k}^T \in R^{K_i^a \times m_i}$ based on the critic NN (18) and actor NN (35) for the player i, the residual error in (12) is

$$\begin{aligned} e_{i,ac}^{k+1}(t) = {} & (\phi_i(x(t)) - \phi_i(x(t+\Delta t)))^T \hat{w}_{i,k+1} \\ & - \int_t^{t+\Delta t} 2\varphi_i^T(x)\hat{w}_{a_i,k+1} R_{ii}\Big(u_i(\tau) - u_i^k(\tau)\Big) d\tau \\ & + \int_t^{t+\Delta t} \sum_{j=1,j\neq i}^{N} \Big(g_j(x)\Big(u_j(\tau) - u_j^k(\tau)\Big)\Big)^T \\ & \times \nabla\phi_i^T(x)\hat{w}_{i,k+1} d\tau \\ & - \int_t^{t+\Delta t} Q_i(x) d\tau \\ & - \int_t^{t+\Delta t} \sum_{j=1}^{N} \Big(\Big(u_j^k(\tau)\Big)^T R_{ij} u_j^k(\tau)\Big) d\tau. \end{aligned} \tag{36}$$

According, (36) can be rewritten as

$$e_{i,ac}^{k+1}(t) = \rho_{i,ac}^k(x(t), u_i(t), u_{-i}(t))\hat{W}_{i,k+1} - \pi_i(x(t)) \tag{37}$$

where $\hat{W}_{i,k+1} = [\hat{w}_{i,k+1}^T, \mathbf{vec}(\hat{w}_{a_i,k+1})^T]^T \in R^{K_i + K_i^a \times m_i}$ is named the estimated weighting function vector

$$\rho_{i,ac}(x(t), u_i(t), u_{-i}(t)) = \begin{bmatrix} (\phi_i(x(t)) - \phi_i(x(t+\Delta t)))^T + \\ \int_t^{t+\Delta t} \sum_{j=1,j\neq i}^{N} \Big(g_j(x)\Big(u_j(\tau) - u_j^k(\tau)\Big)\Big)^T \nabla\phi_i^T(x) d\tau \\ \int_t^{t+\Delta t} -2R_{ii}\big(u_i(\tau) - u_i^k(\tau)\big) \otimes \varphi_i(x) d\tau \end{bmatrix}.$$

Similarly, the offline and online iterative learning schemes can be proposed to approach the ideal critic and actor weights by minimizing the square error $(1/2)(e_{i,ac}^k)^T e_{i,ac}^k$.

Remark 9: Note that the dimension of the vector $\rho_{i,ac}$ in (37) is $K_i + K_i^a \times m_i$ for the actor–critic structure while the dimension of ρ_i in (22) is K_i for the single critic structure. Evidently, higher dimensional data is required for the actor–critic structure, which increases the computational burden.

For the offline iterative learning, we can obtain the solution to the corresponding LS problem yield

$$\hat{W}_{i,k+1} = \left[P_{i,ac}^T P_{i,ac}\right]^{-1} P_{i,ac}^T \Pi_i \tag{38}$$

where $P_{i,ac} = [\rho_{i,ac}^0, \ldots, \rho_{i,ac}^{q-1}]^T$.

Similar with Algorithm 2, the real system data is collected to compute $P_{i,ac}$ and Π_i, then the estimated weighting function vector is updated based on (38) until it converged to a small positive number.

For the online iterative learning, the gradient-based adaptation law with ER for $\hat{W}_i$ is given by

$$\begin{aligned}\dot{\hat{W}}_i = -\,\alpha_i\Bigg[&\frac{\rho_{i,ac}^T(t)}{\left(1+\rho_{i,ac}^T(t)\rho_i(t)\right)^2}\left(\rho_{i,ac}\hat{W}_i+\pi_i(t)\right)\\ &+\sum_{d=1}^{l}\frac{\rho_{i,ac}^T(t_d)}{\left(1+\rho_{i,ac}^T(t_d)\rho_i(t_d)\right)^2}\left(\rho_{i,ac}(t_d)\hat{W}_i+\pi_i(t_d)\right)\Bigg].\end{aligned} \tag{39}$$

The convergence analysis for the offline IRL and online IRL with actor–critic structure is similar with Theorem 2 and 3, so we omit it here.

References

[1] P. Morris, *Introduction to Game Theory*. New York, NY, USA: Springer, 2012.

[2] A. Friedman, *Differential Games*. Mineola, NY, USA: Courier Corporat., 2013.

[3] S. Clemhout and H. Y. Wan, "Differential games—Economic applications," *Handbook Game Theory Econ. Appl.*, vol. 2, pp. 801–825, 1994.

[4] S. Jørgensen and G. Zaccour, *Differential Games in Marketing*, vol. 15. New York, NY, USA: Springer, 2012.

[5] D. Zhao, Y. Dai, and Z. Zhang, "Computational intelligence in urban traffic signal control: A survey," *IEEE Trans. Syst., Man, Cybern. C, Appl. Rev.*, vol. 42, no. 4, pp. 485–494, Jul. 2012.

[6] Q. Zhang, D. Zhao, and Y. Zhu, "Event-triggered H_∞ control for continuous-time nonlinear system via concurrent learning," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 47, no. 7, pp. 1071–1081, Jul. 2017.

[7] Y. Zhu, D. Zhao, X. Yang, and Q. Zhang, "Policy iteration for H_∞ optimal control of polynomial nonlinear systems via sum of squares programming," *IEEE Trans. Cybern.*, vol. 48, no. 2, pp. 500–509, Feb. 2018, doi: 10.1109/TCYB.2016.2643687.

[8] Q. Zhang, D. Zhao, and Y. Zhu, "Data-driven adaptive dynamic programming for continuous-time fully cooperative games with partially constrained inputs," *Neurocomputing*, vol. 238, pp. 377–386, May 2017.

[9] Q. Wei, R. Song, and P. Yan, "Data-driven zero-sum neuro-optimal control for a class of continuous-time unknown nonlinear systems with disturbance using ADP," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 27, no. 2, pp. 444–458, Feb. 2016.

[10] A. W. Starr and Y.-C. Ho, "Nonzero-sum differential games," *J. Optim. Theory Appl.*, vol. 3, no. 3, pp. 184–206, 1969.

[11] J. Nash, "Non-cooperative games," *Ann. Math.*, vol. 54, no. 2, pp. 286–295, 1951.

[12] Y. Zhu and D. Zhao, "Comprehensive comparison of online ADP algorithms for continuous-time optimal control," *Artif. Intell. Rev.*, vol. 49, no. 4, pp. 531–547, 2018.

[13] H. Zhang, Z. Wang, and D. Liu, "A comprehensive review of stability analysis of continuous-time recurrent neural networks," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 25, no. 7, pp. 1229–1262, Jul. 2014.

[14] B. Luo, D. Liu, T. Huang, and D. Wang, "Model-free optimal tracking control via critic-only Q-learning," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 27, no. 10, pp. 2134–2144, Oct. 2016.

[15] Y. Zhu, D. Zhao, H. He, and J. Ji, "Event-triggered optimal control for partially unknown constrained-input systems via adaptive dynamic programming," *IEEE Trans. Ind. Electron.*, vol. 64, no. 5, pp. 4101–4109, May 2017.

[16] D. Wang, H. He, and D. Liu, "Improving the critic learning for event-based nonlinear H_∞ control design," *IEEE Trans. Cybern.*, vol. 47, no. 10, pp. 3417–3428, Oct. 2017, doi: 10.1109/TCYB.2017.2653800.

[17] Y. Yang, D. Wunsch, and Y. Yin, "Hamiltonian-driven adaptive dynamic programming for continuous nonlinear dynamical systems," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 28, no. 8, pp. 1920–1940, Aug. 2017.

[18] K. G. Vamvoudakis and F. L. Lewis, "Multi-player non-zero-sum games: Online adaptive learning solution of coupled Hamilton–Jacobi equations," *Automatica*, vol. 47, no. 8, pp. 1556–1569, 2011.

[19] S. Yasini, M. B. N. Sitani, and A. Kirampor, "Reinforcement learning and neural networks for multi-agent nonzero-sum games of nonlinear constrained-input systems," *Int. J. Mach. Learn. Cybern.*, vol. 7, no. 6, pp. 967–980, 2016.

[20] R. Padhi, N. Unnikrishnan, X. Wang, and S. N. Balakrishnan, "A single network adaptive critic (SNAC) architecture for optimal control synthesis for a class of nonlinear systems," *Neural Netw.*, vol. 19, no. 10, pp. 1648–1660, 2006.

[21] H. Zhang, L. Cui, and Y. Luo, "Near-optimal control for nonzero-sum differential games of continuous-time nonlinear systems using single-network ADP," *IEEE Trans. Cybern.*, vol. 43, no. 1, pp. 206–216, Feb. 2013.

[22] D. Vrabie and F. Lewis, "Neural network approach to continuous-time direct adaptive optimal control for partially unknown nonlinear systems," *Neural Netw.*, vol. 22, no. 3, pp. 237–246, 2009.

[23] H.-N. Wu and B. Luo, "Neural network based online simultaneous policy update algorithm for solving the HJI equation in nonlinear H_∞ control," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 23, no. 12, pp. 1884–1895, Dec. 2012.

[24] D. Vrabie and F. Lewis, "Adaptive dynamic programming for online solution of a zero-sum differential game," *J. Control Theory Appl.*, vol. 9, no. 3, pp. 353–360, 2011.

[25] B. Luo, H.-N. Wu, and T. Huang, "Off-policy reinforcement learning for H_∞ control design," *IEEE Trans. Cybern.*, vol. 45, no. 1, pp. 65–76, Jan. 2015.

[26] D. Vrabie and F. Lewis, "Integral reinforcement learning for online computation of feedback Nash strategies of nonzero-sum differential games," in *Proc. IEEE Conf. Dec. Control*, Atlanta, GA, USA, 2010, pp. 3066–3071.

[27] R. Kamalapurkar, J. R. Klotz, and W. E. Dixon, "Concurrent learning-based approximate feedback-Nash equilibrium solution of N-player nonzero-sum differential games," *IEEE/CAA J. Automatica Sinica*, vol. 1, no. 3, pp. 239–247, Jul. 2014.

[28] C. Mu, D. Wang, and H. He, "Novel iterative neural dynamic programming for data-based approximate optimal control design," *Automatica*, vol. 81, pp. 240–252, Jul. 2017.

[29] Z. Ni, H. He, X. Zhong, and D. V. Prokhorov, "Model-free dual heuristic dynamic programming," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 26, no. 8, pp. 1834–1839, Aug. 2015.

[30] D. Zhao, Q. Zhang, D. Wang, and Y. Zhu, "Experience replay for optimal control of nonzero-sum game systems with unknown dynamics," *IEEE Trans. Cybern.*, vol. 46, no. 3, pp. 854–865, Mar. 2016.

[31] M. Johnson, R. Kamalapurkar, S. Bhasin, and W. E. Dixon, "Approximate *N*-player nonzero-sum game solution for an uncertain continuous nonlinear system," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 26, no. 8, pp. 1645–1658, Aug. 2015.

[32] D. Liu, H. Li, and D. Wang, "Online synchronous approximate optimal learning algorithm for multi-player non-zero-sum games with unknown dynamics," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 44, no. 8, pp. 1015–1027, Aug. 2014.

[33] H. Zhang, L. Cui, X. Zhang, and Y. Luo, "Data-driven robust approximate optimal tracking control for unknown general nonlinear systems using adaptive dynamic programming method," *IEEE Trans. Neural Netw.*, vol. 22, no. 12, pp. 2226–2236, Dec. 2011.

[34] R. Song, F. L. Lewis, Q. Wei, and H. Zhang, "Off-policy actor–critic structure for optimal control of unknown systems with disturbances," *IEEE Trans. Cybern.*, vol. 46, no. 5, pp. 1041–1050, May 2016.

[35] H. Modares, F. L. Lewis, and Z.-P. Jiang, "H_∞ tracking control of completely unknown continuous-time systems via off-policy reinforcement learning," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 26, no. 10, pp. 2550–2562, Oct. 2015.

[36] Y. Jiang and Z.-P. Jiang, "Robust adaptive dynamic programming and feedback stabilization of nonlinear systems," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 25, no. 5, pp. 882–893, May 2014.

[37] H. Li, D. Liu, and D. Wang, "Integral reinforcement learning for linear continuous-time zero-sum games with completely unknown dynamics," *IEEE Trans. Autom. Sci. Eng.*, vol. 11, no. 3, pp. 706–714, Jul. 2014.

[38] Y. Zhu, D. Zhao, and X. Li, "Iterative adaptive dynamic programming for solving unknown nonlinear zero-sum game based on online data," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 28, no. 3, pp. 714–725, Mar. 2017, doi: 10.1109/TNNLS.2016.2561300.

[39] B. Luo, T. Huang, H.-N. Wu, and X. Yang, "Data-driven H_∞ control for nonlinear distributed parameter systems," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 26, no. 11, pp. 2949–2961, Nov. 2015.

[40] R. Song, F. L. Lewis, and Q. Wei, "Off-policy integral reinforcement learning method to solve nonlinear continuous-time multiplayer nonzero-sum games," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 28, no. 3, pp. 704–713, Mar. 2017.

[41] D. Zhao, Z. Xia, and Q. Zhang, "Model-free optimal control based intelligent cruise control with hardware-in-the-loop demonstration [research frontier]," *IEEE Comput. Intell. Mag.*, vol. 12, no. 2, pp. 56–69, May 2017.

[42] Y. Jiang and Z.-P. Jiang, "Robust adaptive dynamic programming with an application to power systems," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 24, no. 7, pp. 1150–1156, Jul. 2013.

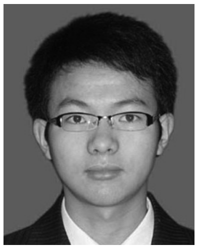
[43] T. Basar *et al.*, *Dynamic Noncooperative Game Theory*, vol. 200. Philadelphia, PA, USA: SIAM, 1995.

[44] Y. Zhu, D. Zhao, and X. Li, "Using reinforcement learning techniques to solve continuous-time non-linear optimal tracking problem without system dynamics," *IET Control Theory Appl.*, vol. 10, no. 12, pp. 1339–1347, Aug. 2016.

[45] M. Abu-Khalaf and F. L. Lewis, "Nearly optimal control laws for nonlinear systems with saturating actuators using a neural network HJB approach," *Automatica*, vol. 41, no. 5, pp. 779–791, 2005.

[46] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*, vol. 1. Cambridge, MA, USA: MIT Press, 1998.

[47] S. Gu, T. Lillicrap, Z. Ghahramani, R. E. Turner, and S. Levine, "Q-Prop: Sample-efficient policy gradient with an off-policy critic," in *Proc. Int. Conf. Learn. Represent.*, Toulon, France, Apr. 2017, pp. 1–13.



Qichao Zhang (M'17) received the B.S. degree in automation from Northeastern Electric Power University, Jilin, China, in 2012, the M.S. degree in control theory and control engineering from Northeast University, Shenyang, China, in 2014, and the Ph.D. degree in control theory and control engineering from the State Key Laboratory of Management and Control for Complex Systems, Institute of Automation, Chinese Academy of Sciences, Beijing, China, in 2017.

He is currently an Assistant Professor with the State Key Laboratory of Management and Control for Complex Systems, Institute of Automation, Chinese Academy of Sciences. His current research interests include reinforcement learning, game theory, and multiagent systems.

Dongbin Zhao (M'06–SM'10) received the B.S., M.S., Ph.D. degrees from the Harbin Institute of Technology, Harbin, China, in 1994, 1996, and 2000, respectively.

He was a Postdoctoral Fellow with Tsinghua University, Beijing, China, from 2000 to 2002. He has been a Professor with the Institute of Automation, Chinese Academy of Sciences, Beijing, China, since 2002, and a Professor with the University of Chinese Academy of Sciences, Beijing. From 2007 to 2008, he was also a Visiting Scholar with the University of Arizona, Tucson, AZ, USA. He has published four books, and over 60 international journal papers. His current research interests include computational intelligence, adaptive dynamic programming, deep reinforcement learning, robotics, intelligent transportation systems, and smart grids.

Dr. Zhao has been an Associate Editor of the IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS since 2012, and the *IEEE Computation Intelligence Magazine* since 2014. He is the Chair of the Beijing Chapter, and was the Chair of Adaptive Dynamic Programming and Reinforcement Learning Technical Committee from 2015 to 2016, and Multimedia Subcommittee of IEEE Computational Intelligence Society from 2015 to 2016. He works as several guest editors of renowned international journals. He is involved in organizing several international conferences.