

Received December 26, 2019, accepted January 20, 2020, date of publication January 31, 2020, date of current version February 17, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2970760

# $H_\infty$ Control for Discrete-Time Multi-Player Systems via Off-Policy Q-Learning

JINNA LI<sup>1,2</sup>, (Member, IEEE), AND ZHENFEI XIAO<sup>1</sup>

<sup>1</sup>School of Information and Control Engineering, Liaoning Shihua University, Liaoning 113001, China

<sup>2</sup>State Key Laboratory of Synthetical Automation for Process Industries, Northeastern University, Shenyang 110819, China

Corresponding author: Jinna Li (lijinna\_721@126.com)


This work was supported in part by the National Natural Science Foundation of China under Grant 61673280, in part by the Open Project of Key Field Alliance of Liaoning Province under Grant 2019-KF-03-06, and in part by the Project of Liaoning Shihua University under Grant 2018XJJ-005.

**ABSTRACT** This paper presents a novel off-policy game Q-learning algorithm to solve  $H_\infty$  control problem for discrete-time linear multi-player systems with completely unknown system dynamics. The primary contribution of this paper lies in that the Q-learning strategy employed in the proposed algorithm is implemented in an off-policy policy iteration approach other than on-policy learning, since the off-policy learning has some well-known advantages over the on-policy learning. All of players struggle together to minimize their common performance index meanwhile defeating the disturbance that tries to maximize the specific performance index, and finally they reach the Nash equilibrium of game resulting in satisfying disturbance attenuation condition. For finding the solution of the Nash equilibrium,  $H_\infty$  control problem is first transformed into an optimal control problem. Then an off-policy Q-learning algorithm is put forward in the typical adaptive dynamic programming (ADP) and game architecture, such that control policies of all players can be learned using only measured data. More importantly, the rigorous proof of no bias of solution to the Nash equilibrium by using the proposed off-policy game Q-learning algorithm is presented. Comparative simulation results are provided to verify the effectiveness and demonstrate the advantages of the proposed method.

**INDEX TERMS**  $H_\infty$  control, off-policy Q-learning, game theory, Nash equilibrium.

## I. INTRODUCTION

The  $H_\infty$  control is a robust control method which is aimed at designing the controllers to attenuate the negative effects in performance of dynamical systems caused by external disturbances meanwhile guarantee the stability of systems if no disturbance exists [1]–[3]. This issue can be handled using the zero-sum game theory, that is, solving a game Bellman equation in the zero-sum game framework results in getting the  $H_\infty$  controller policies [4], [5]. The truth of more complex and large-scale systems with multiple subsystems and multiple controllers in practical engineering applications makes anti-interference control of multi-player systems valuable and more complicated, thereby attracting increasing attention of researchers to  $H_\infty$  control for multi-player or multi-agent systems [6]–[9].

The associate editor coordinating the review of this manuscript and approving it for publication was Zhiguang Feng .

By reviewing the existing results on  $H_\infty$  control for dynamical systems, it is not difficult to find that most of researchers are concerned about model-based  $H_\infty$  controller design using the variety of methods, such as linear matrix inequality (LMI) [10]–[12], zero-sum game [13]–[17] and pole assignment [18]–[20], etc. The requirement of these methods that the dynamics of systems should be accurately known a priori prevents them from applications for systems with inaccurate or even completely unknown models. Reinforcement learning (RL) that can deal with controller design or decision-making problem in uncertain or unknown environment is an alternative tool to solve  $H_\infty$  control for systems without the information of system dynamics. For discrete-time (DT) systems, Al-Tamimi et al. [21] proposed a model-free Q-learning algorithm for the linear zero-sum game with the application to  $H_\infty$  control. Kiumarsi et al. [22] used an off-policy RL method to get a model-free solution to the  $H_\infty$  control of linear systems. Kim and Lewis [23] developed a model-free  $H_\infty$  control algorithm for unknown

linear systems by using RL method based on an actor-critic structure. Rizvi and Lin [24] presented an output feedback Q-learning algorithm for the linear quadratic zero-sum game with  $H_\infty$  control problem. For continuous-time (CT) systems, Modares et al. [25] dealt with the  $H_\infty$  tracking controller design for nonlinear systems. Jiang et al. [26] presented a novel data-driven RL approach to solve the  $H_\infty$  control problem for nonlinear systems with completely unknown dynamics and constrained control input. Luo et al. [27] solved the data-driven  $H_\infty$  control problem of nonlinear distributed parameter systems by using the off-policy learning method and an off-policy RL learning algorithm was proposed for  $H_\infty$  control design [28]. Kiumarsi et al. [29] designed a model-free  $H_\infty$  optimal tracking controller for affine nonlinear systems with input constraints. Fu and Chai [30] proposed an online adaptive algorithm for learning the Nash equilibrium solution for nonlinear zero-sum game problem. It is worth pointing out that the designed  $H_\infty$  controllers in the aforementioned literature are only for single-player systems. On the other hand, the existing reports on multi-player games [31]–[36] usually ignore the negative effects caused by disturbances on performance of systems. What we focus here is how to attenuate the influence of disturbances for multi-player systems using only measured data even in the completely unknown environments.

Most related results are  $H_\infty$  control of multi-agent and multi-player systems [6]–[9]. In [7] agents have their individual dynamics and anti-interference problem has been investigated for continuous-time multi-player systems in [6], [8], [9]. The main difference of multi-player systems from multi-agent systems is all players in multi-player systems are capable of accessing the state of the overall systems, which makes us try to find a different manipulation from multi-agent systems to design  $H_\infty$  controllers. Like [6], [8], [9], the model-free  $H_\infty$  controller design will be taken into account for multi-player systems in this paper, while the difference of nature of discrete-time sampling from continuous-time processes makes it more complicated to solve  $H_\infty$  control problem from the discrete-time system perspective, and multiple players and completely unknown dynamics of players increase this difficulty. Moreover, in view of the advantages of off-policy learning over on-policy learning shown in our previous result [37] wherein the off-policy Q-learning method was proposed for multi-player systems without the consideration of disturbance, developing an off-policy game Q-learning algorithm to solve  $H_\infty$  control problem for discrete-time linear multi-player systems using only measured data becomes our target. To our best knowledge, this problem has not been reported up to now. Besides, the anti-interference control has many practical applications, such as Van der Pol's oscillator systems [6], F-16 aircraft systems of DT or CT [14], [21], [22], [35], rotational/translational actuator (RTAC) nonlinear benchmark problems [38], and industrial operational control [39], etc.

In the ADP architecture, this paper proposes a novel off-policy game Q-learning algorithm for finding multiple controllers that not only guarantee the stability of multi-player systems, but also the disturbance is limited within a disturbance attenuation bound. The contributions of this paper are summarized as follows:

- 1) Unlike the on-policy algorithm [21], [23], [36], [39]–[41] and the off-policy RL method [27], [28] which only consider single-player systems, this paper employs the idea of off-policy Q-learning and extends it to multi-player systems for handling  $H_\infty$  control of multi-player systems. Compared with the existing  $H_\infty$  control algorithms of multi-agent systems [7] and multi-player systems [6], [8], [9], there are two differences, one is model-free  $H_\infty$  controller is designed in this paper for discrete-time dynamics of systems rather than continuous-time dynamics of systems [6]–[9] and the other is that all players in the focused systems cooperatively work hard for achieving their common objective, i.e. minimizing the specific performance and defeating external disturbances, which is not like the way in [6], [8], [9] where players competed each other for optimizing its own performance.
- 2) Under the premise of ensuring the PE condition, probing noises have to be added in the behavior control policy for each player when learning the target policies. The unbiasedness of solution to Nash equilibrium for zero-sum multi-player games using the off-policy game Q-learning algorithm is rigorously proven for the first time.

The structure of this paper is shown below. In Section II, the  $H_\infty$  control problem of multi-player linear DT systems is proposed and the conversion of  $H_\infty$  control into zero-sum game is presented. Section III focuses on solving the problem of multi-player zero-sum games with external disturbance. Moreover, the on-policy game Q-learning algorithm is proposed and the solution to the Nash equilibrium using this kind of algorithm is proven to be biased. In Section IV, an off-policy game Q-learning algorithm is developed together with the proof that the Nash equilibrium solution learned by the proposed algorithm is unbiased. The Section V is the simulation experiments that are used to verify the effectiveness and contributions of the proposed method. Finally, a brief conclusion of this paper is given in Section VI.

*Notations 1:*  $\mathbb{R}^p$  denotes the  $p$  dimensional Euclidean space.  $\mathbb{R}^{p \times q}$  is the set of all real  $p$  by  $q$  matrices. Positive definite matrix is assumed that in the case that  $Q$  is a square matrix of order  $n$  and  $x$  is any non-zero vector,  $x^T Q x > 0$ . If  $x^T Q x \geq 0$ , it is a semi-positive definite matrix.  $\|\cdot\|$  denotes the vector norm. The superscript  $T$  is used for the transpose.  $\otimes$  stands for the Kronecker product.  $vec(L)$  is used to turn any matrix  $L$  into a single column vector.

## II. PROBLEM STATEMENT

In this section, the  $H_\infty$  control problem for linear DT multi-player systems is proposed first. And then it is converted into a zero-sum game problem that all players work together for minimizing the specific performance index, while the disturbance makes the performance worst on the opposite. Thirdly, the value function and the Q-function defined in terms of the performance index are proven to be quadratic.

### A. $H_\infty$ CONTROL PROBLEM

Consider the following linear DT multi-player system subject to exogenous disturbance

$$x_{k+1} = Ax_k + \sum_{i=1}^n B_i u_{ik} + Ed_k \quad (1)$$

where  $x_k = x(k) \in \mathbb{R}^p$  is the system state with initial state  $x_0$ ,  $u_{ik} = u_i(k) \in \mathbb{R}^{m_i}$  ( $i = 1, \dots, n$ ) are the control inputs and  $d_k = d(x_k) \in \mathbb{R}^q$  is the external disturbance input.  $A \in \mathbb{R}^{p \times p}$ ,  $B_i \in \mathbb{R}^{p \times m_i}$ ,  $E \in \mathbb{R}^{p \times q}$  and  $k$  is the sampling time instant.

*Definition 1* [22], [38]: System (1) has  $L_2$ -gain less than or equal to  $\gamma$  if

$$\sum_{k=0}^{\infty} (x_k^T Q x_k + \sum_{i=1}^n u_{ik}^T R_i u_{ik}) \leq \gamma^2 \sum_{k=0}^{\infty} \|d_k\|^2 \quad (2)$$

for all  $d_k \in L_2[0, \infty)$ .  $Q \geq 0$ ,  $R_i > 0$  and  $\gamma \geq 0$  is a prescribed constant disturbance attenuation level.

The  $H_\infty$  control is to find a feedback control policy  $u$  such that the system (1) with  $d_k = 0$  is asymptotically stable and it satisfies the disturbance attenuation condition (2). As claimed in [1], the  $H_\infty$  control problem can be equivalently expressed as

$$J(x_0, U, d_k) = \min_U \max_{d_k} \sum_{k=0}^{\infty} (x_k^T Q x_k + \sum_{i=1}^n u_{ik}^T R_i u_{ik} - \gamma^2 \|d_k\|^2) \quad (3)$$

where  $U = \{u_{1k}, u_{2k}, \dots, u_{nk}\}$ , which means the set  $U$  is composed of  $n$  players with each of them is a controller. As one can know from (3), the objective of these players  $U$  is to fight with the disturbance for minimizing the performance in (3), while the disturbance  $d_k$  could also be viewed as a player that tries to maximize (3). This is a typical zero-sum game problem.

*Definition 2 (Saddle Point Solution [6], [42]):* A set of policies  $(u_{1k}^*, u_{2k}^*, \dots, u_{nk}^*, d_k^*)$  is called the game theoretical saddle point if it satisfies the form

$$\begin{aligned} J(x_0, u_{1k}^*, u_{2k}^*, \dots, u_{nk}^*, d_k^*) &= \min_U \max_{d_k} J(x_0, U, d_k) \\ &= \max_{d_k} \min_U J(x_0, U, d_k) \end{aligned} \quad (4)$$

which indicates the Nash equilibrium condition holds, that is

$$J(x_0, U^*, d_k) \leq J(x_0, U^*, d_k^*) \leq J(x_0, U, d_k^*)$$

where  $U^* = \{u_{1k}^*, u_{2k}^*, \dots, u_{nk}^*\}$ .

In such zero-sum game (3), the saddle-point solution exists if and only if there exists a function  $V(x_k)$  satisfying the following Hamilton-Jacobi-Isaacs (HJI) equation [35], [43].

$$\begin{aligned} &V^*(x_k) \\ &= \min_U \max_{d_k} \sum_{l=k}^{\infty} (x_l^T Q x_l + \sum_{i=1}^n u_{il}^T R_i u_{il} - \gamma^2 \|d_l\|^2) \\ &= \min_U \max_{d_k} \left( x_k^T Q x_k + \sum_{i=1}^n u_{ik}^T R_i u_{ik} - \gamma^2 \|d_k\|^2 + V^*(x_{k+1}) \right) \\ &= x_k^T Q x_k + \sum_{i=1}^n u_{ik}^{*T} R_i u_{ik}^* - \gamma^2 \|d_k^*\|^2 \\ &\quad + V^*(Ax_k + \sum_{i=1}^n B_i u_{ik}^* + Ed_k^*) \end{aligned} \quad (5)$$

where  $V^*(x_k)$  is viewed as the optimal value function.

The arguments provided above has illustrated that  $H_\infty$  control is closely related to zero-sum game (3). The ultimate target of designing  $H_\infty$  control policies in this paper can be achieved by seeking the saddle-point solution to the zero-sum game.

Similar to [21], [31], [44], the optimal Q-function referring to (3) can be defined as

$$Q^*(x_k, U, d_k) = x_k^T Q x_k + \sum_{i=1}^n u_{ik}^T R_i u_{ik} - \gamma^2 \|d_k\|^2 + V^*(x_{k+1}) \quad (6)$$

Thus, the following relation holds

$$V^*(x_k) = \min_U \max_{d_k} Q^*(x_k, U, d_k) = Q^*(x_k, U^*, d_k^*) \quad (7)$$

### B. QUADRATIC FORM PROOF OF VALUE FUNCTION AND Q-FUNCTION

*Definition 3 (Admissible Control Policies [21], [33]):* Suppose  $d_k = 0$ , the control policies  $u_1(x_k), u_2(x_k), \dots, u_n(x_k)$  are defined as admissible with respect to (3) on  $\Omega \in \mathbb{R}^p$ , denoted by  $u_i(x_k) \in \Psi(\Omega)$  if  $u_1(x_k), u_2(x_k), \dots, u_n(x_k)$  are continuous on  $\Omega$ ,  $u_1(0) = 0, u_2(0) = 0, \dots, u_n(0) = 0$ ,  $u_1(x_k), u_2(x_k), \dots, u_n(x_k)$  stabilize (1) on  $\Omega$  and (3) is finite  $\forall x_0 \in \Omega$ .

Lemma 1 is given to show the quadratic forms of the optimal value function and the optimal Q-function associated with performance index (3).

*Lemma 1:* Assume that there are admissible control policies  $u_i = -K_i x_k$  and disturbance policy  $d_k = -K x_k$ , the quadratic forms of the optimal value function and the optimal Q-function can be expressed as:

$$V^*(x_k) = x_k^T P^* x_k \quad (8)$$

and

$$Q^*(x_k, U, d_k) = z_k^T H^* z_k \quad (9)$$

where  $P^*$  and  $H^*$  are positive definite matrices. And

$$z_k = [x_k^T \quad u_{1k}^T \quad u_{2k}^T \quad \dots \quad u_{nk}^T \quad d_k^T]^T \quad (10)$$

Proof:

$$\begin{aligned}
 V^*(x_k) &= \min_U \max_{d_l} \sum_{l=k}^{\infty} (x_l^T Q x_l + \sum_{i=1}^n u_{il}^T R_i u_{il} - \gamma^2 d_l^T d_l) \\
 &= \min_{K_i(i=1,2,\dots,n)} \max_K \sum_{l=k}^{\infty} \left[ x_l^T Q x_l + \sum_{i=1}^n (-K_i x_l)^T R_i \right. \\
 &\quad \left. \times (-K_i x_l) - \gamma^2 (-K_i x_l)^T (-K_i x_l) \right] \\
 &= \min_{K_i(i=1,2,\dots,n)} \max_K \sum_{l=k}^{\infty} x_l^T \left[ Q + \sum_{i=1}^n (K_i)^T R_i (K_i) \right. \\
 &\quad \left. - \gamma^2 K^T K \right] x_l \\
 &= \min_{K_i(i=1,2,\dots,n)} \max_K \sum_{l=0}^{\infty} x_{l+k}^T \left[ Q + \sum_{i=1}^n (K_i)^T R_i (K_i) \right. \\
 &\quad \left. - \gamma^2 K^T K \right] x_{l+k} \tag{11}
 \end{aligned}$$

where  $x_{l+k} = (A - \sum_{i=1}^n B_i K_i - EK)^l x_k = G^l x_k$ . Further, one has

$$\begin{aligned}
 V^*(x_k) &= \min_{K_i(i=1,2,\dots,n)} \max_K \sum_{l=0}^{\infty} x_k^T (G^l)^T \\
 &\quad \times \left[ Q + \sum_{i=1}^n (K_i)^T R_i (K_i) - \gamma^2 K^T K \right] (G^l) x_k \tag{12}
 \end{aligned}$$

then, one has

$$V^*(x_k) = x_k^T P^* x_k \tag{13}$$

where

$$P^* = \min_{K_i(i=1,2,\dots,n)} \max_K \sum_{l=0}^{\infty} (G^l)^T \left[ Q + \sum_{i=1}^n (K_i)^T R_i (K_i) - \gamma^2 K^T K \right] (G^l)$$

Then, one has

$$\begin{aligned}
 Q^*(x_k, U, d_k) &= x_k^T Q x_k + \sum_{i=1}^n u_{ik}^T R_i u_{ik} - \gamma^2 d_k^T d_k + V^*(x_{k+1}) \\
 &= x_k^T Q x_k + \sum_{i=1}^n u_{ik}^T R_i u_{ik} - \gamma^2 d_k^T d_k + x_{k+1}^T P^* x_{k+1} \\
 &= x_k^T Q x_k + \sum_{i=1}^n u_{ik}^T R_i u_{ik} - \gamma^2 d_k^T d_k \\
 &\quad + (Ax_k + \sum_{i=1}^n B_i u_{ik} - Ed_k)^T P^* (Ax_k + \sum_{i=1}^n B_i u_{ik} - Ed_k) \\
 &= z_k^T H^* z_k \tag{14}
 \end{aligned}$$

where

$$\begin{aligned}
 H^* &= \begin{bmatrix} H_{xx}^* & H_{xu_1}^* & H_{xu_2}^* & \dots & H_{xu_n}^* & H_{xd}^* \\ H_{xu_1}^{*,T} & H_{u_1 u_1}^* & H_{u_1 u_2}^* & \dots & H_{u_1 u_n}^* & H_{u_1 d}^* \\ H_{xu_2}^{*,T} & H_{u_1 u_2}^{*,T} & H_{u_2 u_2}^* & \dots & H_{u_2 u_n}^* & H_{u_2 d}^* \\ \vdots & \vdots & \vdots & \dots & \vdots & \vdots \\ H_{xu_n}^{*,T} & H_{u_1 u_n}^{*,T} & H_{u_2 u_n}^{*,T} & \dots & H_{u_n u_n}^* & H_{u_n d}^* \\ H_{xd}^{*,T} & H_{u_1 d}^{*,T} & H_{u_2 d}^{*,T} & \dots & H_{u_n d}^{*,T} & H_{dd}^* \end{bmatrix} \\
 &= \begin{bmatrix} A^T P^* A + Q & \dots & A^T P^* B_n & \dots & A^T P^* E \\ (A^T P^* B_1)^T & \dots & B_1^T P^* B_n & \dots & B_1^T P^* E \\ (A^T P^* B_2)^T & \dots & B_2^T P^* B_n & \dots & B_2^T P^* E \\ \vdots & \dots & \vdots & \dots & \vdots \\ (A^T P^* B_n)^T & \dots & B_n^T P^* B_n + R_n & \dots & B_n^T P^* E \\ (A^T P^* E)^T & \dots & (B_n^T P^* E)^T & \dots & -\gamma^2 I + E^T P^* E \end{bmatrix} \tag{15}
 \end{aligned}$$

By (13) and (14), one can get

$$P^* = M^T H^* M \tag{16}$$

where

$$M = [I \quad -K_1^T \quad \dots \quad -K_n^T \quad -K^T]^T$$

*Remark 1:* Following the idea in [21], where the quadratic forms of value function and Q-function for linear single-player systems are proven, the rigorous proof that the value function and Q-function defined for zero-sum game of multi-player systems are quadratic is presented in this paper.

### III. SOLVING MULTI-PLAYER ZERO-SUM GAME

In this section, the theoretical solution of the zero-sum game for multi-player systems is first obtained according to the Bellman equation of Q-function. Then the on-policy game Q-learning algorithm is provided to solve this problem. Finally, it is proved that the Nash equilibrium solution obtained by the on-policy game Q-learning algorithm is biased.

#### A. THEORETICAL SOLUTION

Now we are in the position to solve HJI equation based on game theory. By Lemma 1, referring to HJI equation (5) yields the optimal Q-function based Bellman equation below:

$$z_k^T H z_k = x_k^T Q x_k + \sum_{i=1}^n u_{ik}^T R_i u_{ik} - \gamma^2 \|d_k\|^2 + z_{k+1}^T H z_{k+1} \tag{17}$$

The optimal control policy  $u_i^*$  of each player  $i$  and the worst-case disturbance  $d_k^*$  should satisfy  $\frac{\partial Q^*(x_k, U, d_k)}{\partial u_i} = 0$  and  $\frac{\partial Q^*(x_k, U, d_k)}{\partial d_k} = 0$ . Therefore, one has

$$u_i^*(k) = -K_i^* x_k \tag{18}$$

$$d_k^* = -K^* x_k \tag{19}$$

where

$$K_i^* = H_{u_i u_i}^{-1} \left[ H_{x u_i}^T - (H_{u_i u_1} K_1 + \dots + H_{u_i u_{i-1}} K_{i-1} + H_{u_i u_{i+1}} K_{i+1} + \dots + H_{u_i u_n} K_n + H_{u_i d} K) \right] \quad (20)$$

$$K^* = H_{dd}^{-1} \left[ H_{xd}^T - (H_{du_1} K_1 + H_{du_2} K_2 + \dots + H_{du_n} K_n) \right] \quad (21)$$

Substituting  $K_i^*$  in (20) and  $K^*$  in (21) into (17) yields the optimal Q-function based game Riccati equation (GRE).

$$(z_k)^T H^* z_k = x_k^T Q x_k + \sum_{i=1}^n (u_i^*)^T R_i u_i^* - \gamma^2 \|d_k^*\|^2 + (z_{k+1})^T H^* z_{k+1} \quad (22)$$

One thing to note is that Al-Tamimi et al. [21] and Vamvoudakis et al. [35] have proved that the following  $K_i^*$  ( $i = 1, 2, \dots, n$ ) and  $K^*$  can keep system (1) stable when  $d_k = 0$  and achieve Nash equilibrium.

$$K_i^* = (H_{u_i u_i}^*)^{-1} \left[ (H_{x u_i}^*)^T - (H_{u_i u_1}^* K_1^* + \dots + H_{u_i u_{i-1}}^* K_{i-1}^* + H_{u_i u_{i+1}}^* K_{i+1}^* + \dots + H_{u_i u_n}^* K_n^* + H_{u_i d}^* K^*) \right] \quad (23)$$

$$K^* = (H_{dd}^*)^{-1} \left[ (H_{xd}^*)^T - (H_{du_1}^* K_1^* + H_{du_2}^* K_2^* + \dots + H_{du_n}^* K_n^*) \right] \quad (24)$$

Note that solving (23) and (24), that is, solving the zero-sum game problem defined by (6), is to find the optimal control policies satisfying the disturbance attenuation condition (2).

*Remark 2:* Since the optimal control policies and disturbance policy in (23) and (24) are coupled, they are difficult to be solved. Therefore, an on-policy game Q-learning algorithm is going to be presented to overcome this difficulty resulting in obtaining the control laws  $u_i^*(k) = -K_i^* x_k$  and disturbance policy  $d_k^* = -K^* x_k$ .

### B. ON-POLICY GAME Q-LEARNING ALGORITHM

The on-policy RL algorithms in [5], [45]–[47] are extended to solve  $H_\infty$  control problem for multi-player systems, and thus the on-policy game Q-learning Algorithm 1 is proposed.

*Remark 3:* In Algorithm 1, the training data  $z_k$  are generated by the iterative control policies (26) and iterative disturbance policies (27). In this sense, Algorithm 1 is indeed on-policy learning [39]–[41]. Moreover, the control policy gains  $K_i^{j+1}$  are updated by solving matrix  $H^{j+1}$  in Q-function. As  $j \rightarrow \infty$ ,  $H^{j+1}$  converges to the optimal value, and then  $K_i^{j+1}$  converge to the optimal value. This conclusion can be made and proven using the similar way to [21], [44].

### C. BIAS ANALYSIS OF SOLUTION LEARNED BY THE ON-POLICY GAME Q-LEARNING ALGORITHM

For systems with single player, the existing results have proven that biased solutions to iterative Bellman equation can

### Algorithm 1 On-Policy Game Q-Learning for the Zero-Sum Game

- 1: Given an n-tuple initial admissible controller gains for  $K_1^0, K_2^0, \dots, K_n^0$  and disturbance policy gain  $K^0$ . Let  $j = 0$  and  $j$  represents the number of iterations, and  $i$  denotes the player  $i$  ( $i = 1, 2, \dots, n + 1$ ). Set  $i = 1$ ;
- 2: Evaluate policies by solve  $H^{j+1}$

$$z_k^T H^{j+1} z_k = x_k^T Q x_k + \sum_{i=1}^n (u_{ik}^j)^T R_i u_{ik}^j - \gamma^2 \|d_k^j\|^2 + z_{k+1}^T H^{j+1} z_{k+1} \quad (25)$$

$$\text{where } z_{k+1} = \left[ x_{k+1}^T \quad u_{i,k+1}^{j,T} (i = 1, 2, \dots, n) \quad d_{k+1}^{j,T} \right]^T.$$

- 3: Update the control and disturbance policy:

$$u_{ik}^{j+1} = -K_i^{j+1} x_k \quad (26)$$

$$d_k^{j+1} = -K^{j+1} x_k \quad (27)$$

where

$$K_i^{j+1} = (H_{u_i u_i}^{j+1})^{-1} \left[ (H_{x u_i}^{j+1})^T - (H_{u_i u_1}^{j+1} K_1^j + H_{u_i u_{i-1}}^{j+1} K_{i-1}^j + H_{u_i u_{i+1}}^{j+1} K_{i+1}^j + \dots + H_{u_i u_n}^{j+1} K_n^j + H_{u_i d}^{j+1} K^j) \right] \quad (28)$$

$$K^{j+1} = (H_{dd}^{j+1})^{-1} \left[ (H_{xd}^{j+1})^T - (H_{du_1}^{j+1} K_1^j + H_{du_2}^{j+1} K_2^j + \dots + H_{du_n}^{j+1} K_n^j) \right] \quad (29)$$

- 4: If  $i < n + 1$ , then  $i = i + 1$  and go back to Step 2. Otherwise  $j = j + 1$ ,  $i = 1$ , and go to Step 5;
- 5: Stop when

$$\|H^{j-1} - H^j\| \leq \varepsilon$$

with a small constant  $\varepsilon$  ( $\varepsilon > 0$ ). Otherwise go back to Step 2.

be generated caused by adding probing noise. The sequel is going to prove that this kind of biasedness of solutions still exists when solving the optimal Q-function based Bellman equation using the on-policy game Q-learning algorithm for multi-player systems.

To satisfy the PE condition in Algorithm 1, probing noises are added to  $u_{ik}^j$ . Thus, the actual control inputs applied to the system for collecting data are

$$\hat{u}_{ik}^j = u_{ik}^j + e_{ik} \quad (30)$$

with  $e_{ik} = e_i(k)$  being probing noises and  $u_{ik}^j$  given by (26). Theorem 1 will prove that there exists the bias of solutions to (25).



*Theorem 1:* Rewrite iterative Q-function based Bellman equation (25) as

$$\begin{aligned} & x_k^T (M^j)^T H^{j+1} M^j x_k \\ &= x_k^T Q x_k + \sum_{i=1}^n (u_{ik}^j)^T R_i u_{ik}^j \\ & \quad - \gamma^2 d_k^T d_k + x_{k+1}^T (M^j)^T H^{j+1} M^j x_{k+1} \end{aligned} \quad (31)$$

where

$$M^j = \begin{bmatrix} I & -(K_1^j)^T & \dots & -(K_n^j)^T & -(K^j)^T \end{bmatrix}^T$$

Let  $H^{j+1}$  be the solution (31) with  $e_{ik} = 0$  and  $\hat{H}^{j+1}$  be the solution to (31) with  $e_{ik} \neq 0$ . Then,  $H^{j+1} \neq \hat{H}^{j+1}$ .

*Proof:* Using (30) with  $e_i \neq 0$  in (31), the Bellman equation becomes the following

$$\begin{aligned} & x_k^T (M^j)^T \hat{H}^{j+1} M^j x_k \\ &= x_k^T Q x_k + \sum_{i=1}^n (u_{ik}^j + e_{ik})^T R_i (u_{ik}^j + e_{ik}) - \gamma^2 d_k^T d_k \\ & \quad + x_{k+1}^T (M^j)^T \hat{H}^{j+1} M^j x_{k+1} \end{aligned} \quad (32)$$

where

$$x_{k+1} = A x_k + \sum_{i=1}^n B_i (u_{ik}^j + e_{ik}) + E d_k \quad (33)$$

Further, (32) is rewritten as

$$\begin{aligned} & x_k^T (M^j)^T \hat{H}^{j+1} M^j x_k \\ &= x_k^T Q x_k + \sum_{i=1}^n (u_{ik}^j + e_{ik})^T R_i \\ & \quad \times (u_{ik}^j + e_{ik}) - \gamma^2 d_k^T d_k \\ & \quad + \left( A x_k + \sum_{i=1}^n B_i (u_{ik}^j + e_{ik}) - E d_k \right)^T (M^j)^T \hat{H}^{j+1} \\ & \quad \times M^j \left( A x_k + \sum_{i=1}^n B_i (u_{ik}^j + e_{ik}) - E d_k \right) \\ &= x_k^T Q x_k + \sum_{i=1}^n (u_{ik}^j)^T R_i u_{ik}^j - \gamma^2 d_k^T d_k \\ & \quad + x_{k+1}^T (M^j)^T \hat{H}^{j+1} M^j x_{k+1} + 2 \sum_{i=1}^n e_{ik}^T R_i u_{ik}^j \\ & \quad + \sum_{i=1}^n e_{ik}^T (B_i^T (M^j)^T \hat{H}^{j+1} M^j B_i + R_i) e_{ik} \\ & \quad + 2 \sum_{i=1}^n e_{ik}^T B_i^T (M^j)^T \hat{H}^{j+1} M^j x_{k+1} \end{aligned} \quad (34)$$

It can be seen that what we learned in Algorithm 1 needs to satisfy (34) other than (25). By comparing (25) with (34), one can find  $H^{j+1} \neq \hat{H}^{j+1}$ , which indicates that the control policies updated by (28) and (29) may be inaccurate and produce bias if probing noise is added when implementing Algorithm 1. This completes the proof. ■

*Remark 4:* It can be noted that in Algorithm 1, the system must update  $u_{ik}^{j+1} = -K_i^{j+1} x_k$  to generate data, which is a typical feature of the on-policy reinforcement learning algorithm, which has to produce deviation of solution during the learning process.

*Remark 5:* One can also notice that the disturbance input must be updated in the prescribed manner (27) and applied to the system. However, this is not possible in practical applications, because the disturbance is generally independent and random.

In order to avoid these shortcomings of the on-policy game Q-learning algorithm mentioned in Remark 3 and Remark 4, behavior inputs and behavior disturbance policy are going to be introduced in this paper when learning the saddle point for multi-player systems subject to disturbance, which indicates we shall investigate the off-policy game Q-learning to solve the zero-sum game problem. Therefore, the off-policy game Q-learning method will be proposed in the next section.

*Remark 6:* Compared with [22], we extend its standard  $H_\infty$  control problem to multi-player zero-sum game systems, and the rigorous proof of biased solution has been provided in this paper.

#### IV. OFF-POLICY GAME Q-LEARNING TECHNIQUE

In this section, we propose an off-policy game Q-learning algorithm to solve the zero-sum game problem, such that the  $H_\infty$  controllers of multi-player systems can be found even when no information of system dynamics is available. Moreover, it is proved that this algorithm will not produce deviation of solution even though probing noises are added to the behavior control policies. The structure of the off-policy game Q-learning for the multi-player  $H_\infty$  control problem is shown in Fig. 1.

##### A. DERIVATION OF OFF-POLICY GAME Q-LEARNING ALGORITHM

In this part, we focus on the formulas derivation and presentation of the off-policy game Q-learning algorithm.

From (25), one has

$$\begin{aligned} (M^j)^T H^{j+1} M^j &= (M^j)^T \Lambda M^j \\ & \quad + \left( A - \sum_{i=1}^n B_i K_i^j - E K^j \right)^T (M^j)^T H^{j+1} M^j \\ & \quad \times \left( A - \sum_{i=1}^n B_i K_i^j - E K^j \right) \end{aligned} \quad (35)$$

where

$$\Lambda = \text{diag}(Q, R_1, R_2, \dots, R_n, -\gamma^2 I)$$

Introducing auxiliary variables  $u_i^j = -K_i^j x_k$  and  $d_k^j = -K^j x_k$  to system (1) yields

$$x_{k+1} = A_c x_k + \sum_{i=1}^n B_i (u_{ik} - u_{ik}^j) + E (d_k - d_k^j) \quad (36)$$

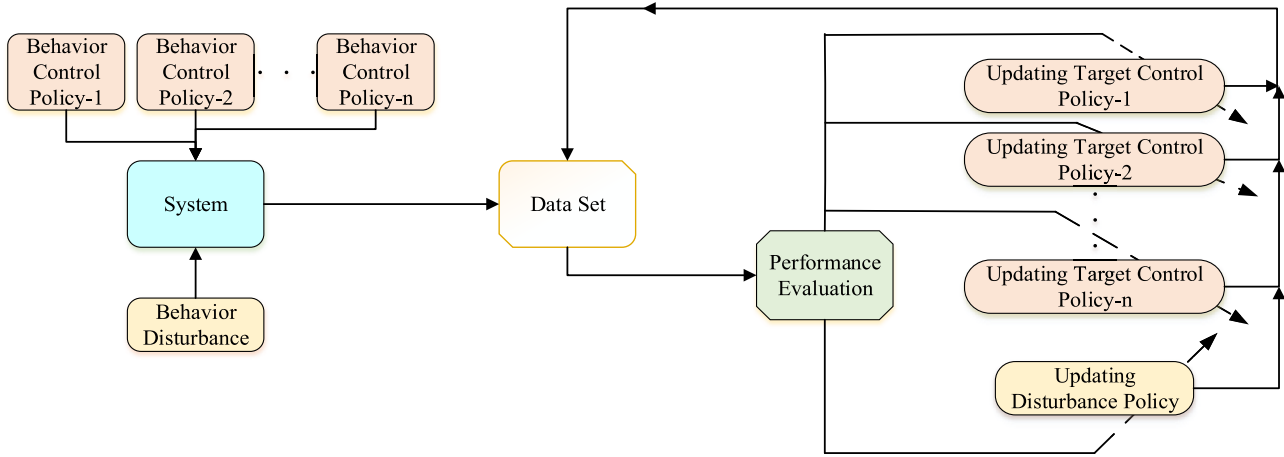


FIGURE 1. The structure of the off-policy game Q-learning for  $H_\infty$  control.

where  $A_c = A - \sum_{i=1}^n B_i K_i^j - EK^j$ ,  $u_i$  and  $d_k$  are called the behavior control policies and the behavior disturbance policy which are used to generate data, while  $u_{ik}^j$  and  $d_k^j$  are called the target control policies and the target disturbance policy which need to be learned. Along the system trajectory is (36), one has

$$\begin{aligned} & Q^{j+1}(x_k, U, d_k) - x_k^T A_c^T (M^j)^T H^{j+1} M^j A_c x_k \\ &= x_k^T (M^j)^T H^{j+1} M^j x_k \\ & \quad - \left( x_{k+1} - \sum_{i=1}^n B_i (u_{ik} - u_{ik}^j) - E(d_k - d_k^j) \right)^T (M^j)^T \\ & \quad \times H^{j+1} M^j \left( x_{k+1} - \sum_{i=1}^n B_i (u_{ik} - u_{ik}^j) - E(d_k - d_k^j) \right) \\ &= x_k^T (M^j)^T \Lambda M^j x_k \end{aligned} \quad (37)$$

In view that  $P^{j+1}$  and  $H^{j+1}$  are related as shown in (15) and (16), then the following holds

$$\begin{aligned} & x_k^T (M^j)^T H^{j+1} M^j x_k - x_{k+1}^T (M^j)^T H^{j+1} M^j x_{k+1} \\ & + 2 \left( Ax_k + \sum_{i=1}^n B_i u_{ik} + Ed_k \right)^T P^{j+1} \sum_{i=1}^n B_i (u_{ik} - u_{ik}^j) \\ & + 2 \left( Ax_k + \sum_{i=1}^n B_i u_{ik} + Ed_k \right)^T P^{j+1} E(d_k - d_k^j) \\ & - \sum_{i=1}^n (u_{ik} - u_{ik}^j)^T B_i^T P^{j+1} \sum_{i=1}^n B_i (u_{ik} - u_{ik}^j) \\ & - 2 \sum_{i=1}^n (u_{ik} - u_{ik}^j)^T B_i^T P^{j+1} E(d_k - d_k^j) \\ & - (d_k - d_k^j)^T E^T P^{j+1} E(d_k - d_k^j) \\ &= x_k^T (M^j)^T \Lambda M^j x_k \end{aligned} \quad (38)$$

Further one has

$$\begin{aligned} & x_k^T (M^j)^T H^{j+1} M^j x_k - x_{k+1}^T (M^j)^T H^{j+1} M^j x_{k+1} \\ & + 2x_k^T \begin{bmatrix} H_{xu_1}^{j+1} & H_{xu_2}^{j+1} & \dots & H_{xu_n}^{j+1} \end{bmatrix} \sum_{i=1}^n (u_{ik} + K_i^j x_k) \\ & + 2 \sum_{i=1}^n u_{ik}^T G^{j+1} \sum_{i=1}^n (u_{ik} + K_i^j x_k) \\ & + 2d_k^T (H_{ud}^{j+1})^T \sum_{i=1}^n (u_{ik} + K_i^j x_k) \\ & + 2x_k^T (H_{xd}^{j+1}) (d_k + K^j x_k) + 2 \sum_{i=1}^n u_{ik}^T H_{u_i d}^{j+1} (d_k + K^j x_k) \\ & - \sum_{i=1}^n (u_{ik} + K_i^j x_k)^T G^{j+1} \sum_{i=1}^n (u_{ik} + K_i^j x_k) \\ & - 2 \sum_{i=1}^n (u_{ik} + K_i^j x_k)^T H_{u_i d}^{j+1} (d_k + K^j x_k) \\ & - (d_k + K^j x_k)^T (H_{dd}^{j+1} + \gamma^2 I) (d_k + K^j x_k) \\ &= x_k^T (M^j)^T \Lambda M^j x_k \end{aligned} \quad (39)$$

where

$$G^{j+1} = \begin{bmatrix} H_{u_1 u_1}^{j+1} - R_1 & H_{u_1 u_2}^{j+1} & \dots & H_{u_1 u_n}^{j+1} \\ (H_{u_1 u_2}^{j+1})^T & H_{u_2 u_2}^{j+1} - R_2 & \dots & H_{u_2 u_n}^{j+1} \\ (H_{u_1 u_3}^{j+1})^T & (H_{u_2 u_3}^{j+1})^T & \dots & H_{u_3 u_n}^{j+1} \\ \vdots & \vdots & \dots & \vdots \\ (H_{u_1 u_n}^{j+1})^T & (H_{u_2 u_n}^{j+1})^T & \dots & H_{u_n u_n}^{j+1} - R_n \end{bmatrix}$$

Manipulating (39) can get the following form

$$\hat{\theta}^j(k) \hat{L}^{j+1} = \sum_{k=0}^{\infty} \hat{\rho}_k \quad (40)$$

where

$$\hat{\rho}_k = x_k^T Q x_k + \sum_{i=1}^n u_{ik}^T R_i u_{ik} - \gamma^2 d_k^T d_k$$

$$\hat{L}^{j+1} = \left[ (\text{vec}(\hat{L}_{rz}^{j+1}))^T, \dots, (\text{vec}(\hat{L}_{n+1,n+1}^{j+1}))^T \right]^T$$

$$\hat{\theta}^j(k) = \left[ \hat{\theta}_{rz}^j, \dots, \hat{\theta}_{n+1,n+1}^j \right]$$

with  $r = 0, 1, 2, \dots, n+1, z = r, r+1, r+2, \dots, n+1$ . Besides,

$$\hat{\theta}_{00}^j = x_k^T \otimes x_k^T - x_{k+1}^T \otimes x_{k+1}^T$$

$$\hat{L}_{00}^{j+1} = H_{xx}^{j+1}$$

$$\hat{\theta}_{ss}^j = -(K_s^j x_{k+1})^T \otimes (K_s^j x_{k+1})^T + u_s^T \otimes u_s^T$$

$$\hat{L}_{ss}^{j+1} = H_{u_s u_s}^{j+1}$$

$$\hat{\theta}_{s+1,s+1}^j = -(K^j x_{k+1})^T \otimes (K^j x_{k+1})^T + d_k^T \otimes d_k^T$$

$$\hat{L}_{s+1,s+1}^{j+1} = H_{dd}^{j+1}$$

$$\hat{\theta}_{0s}^j = 2x_{k+1}^T \otimes (K_s^j x_{k+1})^T + 2x_k^T \otimes u_s^T$$

$$\hat{L}_{0s}^{j+1} = H_{xu_s}^{j+1}$$

$$\hat{\theta}_{0s+1}^j = 2x_{k+1}^T \otimes (K^j x_{k+1})^T + 2x_k^T \otimes d_k^T$$

$$\hat{L}_{0s+1}^{j+1} = H_{xd}^{j+1}$$

$$\hat{\theta}_{st}^j = -2(K_s^j x_{k+1})^T \otimes (K_t^j x_{k+1})^T + 2u_s^T \otimes u_t^T$$

$$\hat{L}_{st}^{j+1} = H_{u_s u_t}^{j+1}$$

$$\hat{\theta}_{s,s+1}^j = -2(K_s^j x_{k+1})^T \otimes (K^j x_{k+1})^T + 2u_s^T \otimes d_k^T$$

$$\hat{L}_{s,s+1}^{j+1} = H_{u_s d}^{j+1}$$

with  $s \neq t$  and  $s, t = 1, 2, \dots, n$ .

Based on the above part,  $K_1^{j+1}, K_2^{j+1}, \dots, K_n^{j+1}$  and  $K^{j+1}$  can be expressed as the form of  $\hat{L}^{j+1}$

$$K_i^{j+1} = (\hat{L}_{ii}^{j+1})^{-1} \left( (\hat{L}_{0i}^{j+1})^T - \left[ (\hat{L}_{i1}^{j+1})^T K_1^j + \dots + (\hat{L}_{(i,i-1)}^{j+1})^T K_{i-1}^j + \hat{L}_{(i,i+1)}^{j+1})^T K_{i+1}^j + \dots + (\hat{L}_{in}^{j+1})^T K_n^j + (\hat{L}_{i,n+1}^{j+1})^T K^j \right] \right) \quad (41)$$

$$K^{j+1} = (\hat{L}_{n+1,n+1}^{j+1})^{-1} \left( (\hat{L}_{0,n+1}^{j+1})^T - \left[ (\hat{L}_{n+1,1}^{j+1})^T K_1^j + (\hat{L}_{n+1,2}^{j+1})^T K_2^j + \dots + (\hat{L}_{n+1,n}^{j+1})^T K_n^j \right] \right) \quad (42)$$

**Theorem 2:**  $(H^{j+1}, K_i^{j+1}, K^{j+1})$  are the solution of (40), (41) and (42) if and only if they are the solution of (25), (28) and (29).

*Proof:* We can see from the formula derivation that if  $(H^{j+1}, K_i^{j+1}, K^{j+1})$  are solutions to (25), (28) and (29), then  $(H^{j+1}, K_i^{j+1}, K^{j+1})$  will also satisfy (40), (41) and (42). Now we need to prove that the solutions to (40), (41) and (42) are the same as the solutions to (25), (28) and (29).

It can be seen that the (40) is equivalent to the (38), so their solutions are also the same. Subtracting (38) from (37),

one gets (25), Thus one knows that the solutions of (40), (41) and (42) is equal to (25), (28) and (29). This completes the proof. ■

---

**Algorithm 2** Off-Policy Game Q-Learning for the Zero-Sum Game

---

- 1: Data collection: Collect system data  $x_k$  and store them in (40) by using (36);
  - 2: Initialize the admissible control policies of multiple players  $K_1^0, K_2^0, K_3^0, \dots, K_n^0$  and disturbance policy gain(the  $n+1$  player)  $K^0$ . Set the iteration index  $j=0$  and  $i=1$  represents player  $i(i=1, 2, \dots, n+1)$ ;
  - 3: Performing the off-policy game Q-learning: use the recursive least-square method to solve the  $\hat{L}^{j+1}$  in (40), and then  $K_i^{j+1}$  and  $K^{j+1}$  can be updated by (41) and (42);
  - 4: If  $i < n+1$ , then  $i = i+1$  and go back to Step 3. Otherwise  $j = j+1, i = 1$  and go to Step 5;
  - 5: Stop when  $\|K_i^j - K_i^{j-1}\| \leq \varepsilon$  ( $i = 1, 2, \dots, n+1$ ), the optimal control policy is obtained. Otherwise,  $i = 1$ , and go back to Step 3.
- 

*Remark 7:* One thing to be noticed is that Algorithm 1 and Algorithm 2 have the same solution, which was proven in Theorem 2. Meanwhile, it concludes from Theorem 2 that, if  $\hat{L}^{j+1}$  can be solved correctly, then the control policies  $K_i$  will converge to the optimal value since  $K_i$  learned by Algorithm 1 converge to the optimal values, that is, when  $j \rightarrow \infty, u_i^j \rightarrow u_i^*$ .

*Remark 8:* The game Q-learning in Algorithm 2 is indeed an off-policy Q-learning approach, since the target control policies are updated but not applied to the real systems. In Algorithm 2, arbitrary behavior control policies  $u_i$  and behavior disturbance policy  $d_k$  that can make the system stable are used to generate data, thereby overcoming the shortcoming of insufficient system exploration, which is also the most basic characteristic of the off-policy learning [22], [31], [35], [44], [48]. While the policies  $u_i^j = -K_i^j x_k$  and  $d_k^j = -K^j x_k$  are the target policies and updated using measured data.

**B. NO BIAS ANALYSIS OF SOLUTION LEARNED BY OFF-POLICY GAME Q-LEARNING ALGORITHM**

In Section III, we have proved that Algorithm 1 will have an impact on the learning results when probing noise is added. Next, we will prove the superiority of Algorithm 2 over Algorithm 1, that is, the probing noise will not affect the system, and the Nash equilibrium solution learned is without deviation.

*Theorem 3:* Add probing noises to the behavior control policies in Algorithm 2. Let  $H^{j+1}$  be the solution to (37) with  $e_i = 0$  and  $\hat{H}^{j+1}$  be the solution to (37) with  $e_i \neq 0$ , then  $\hat{H}^{j+1} = H^{j+1}$ .

*Proof:* Probing noises are added to the behavior control policies  $u_i + e_i$ , the off-policy game Q-learning



equation (37) is

$$\begin{aligned} & \hat{x}_k^T (M^j)^T \hat{H}^{j+1} M^j \hat{x}_k \\ &= \hat{x}_k^T (M^j)^T \Lambda M^j \hat{x}_k \\ &+ \hat{x}_k^T \left( A - \sum_{i=1}^n B_i K_i^j - EK^j \right)^T (M^j)^T \\ &\times \hat{H}^{j+1} M^j \left( A - \sum_{i=1}^n B_i K_i^j - EK^j \right) \hat{x}_k \end{aligned} \quad (43)$$

Notice that if adding probing noises into system (36), then it becomes

$$\hat{x}_{k+1} = A_c \hat{x}_k + \sum_{i=1}^n B_i (u_{ik} + e_{ik} + K_i^j \hat{x}_k) + E(d_k + K^j \hat{x}_k) \quad (44)$$

In this case, (37) becomes

$$\begin{aligned} & \hat{x}_k^T (M^j)^T \hat{H}^{j+1} M^j \hat{x}_k \\ &- \left( \hat{x}_{k+1} - \sum_{i=1}^n B_i (u_{ik} + e_{ik} + K_i^j \hat{x}_k) - E(d_k + K^j \hat{x}_k) \right)^T \\ &\times (M^j)^T \hat{H}^{j+1} M^j \\ &\times \left( \hat{x}_{k+1} - \sum_{i=1}^n B_i (u_{ik} + e_{ik} + K_i^j \hat{x}_k) - E(d_k + K^j \hat{x}_k) \right) \\ &= \hat{x}_k^T (M^j)^T \Lambda M^j \hat{x}_k \end{aligned} \quad (45)$$

Substituting (44) into (45), (45) becomes (43). So the solution to (43) is the same as (37). From the proof of Theorem 2, on can find that the solution to (40) is equal to that to (37). Therefore, it is impossible for the off-policy game Q-learning algorithm to produce bias when adding probing noises. The unbiasedness of the off-policy game Q-learning method is proved. ■

*Remark 9:* Different from [22], where the off-policy RL method is used to solve standard  $H_\infty$  control problem for systems with single controller. What this paper has done is not only to extend the result in [22] to the case of  $H_\infty$  control for multi-player systems, but no bias of solution using the off-policy game Q-learning and the biased solution using the on-policy game Q-learning have been rigorously proven even though complex derivation caused by multiple players in the systems.

*Remark 10:* Compared with [31]–[33], [37], [44] that ignored the negative impact of disturbance on performance of multi-agent systems, this is the first time that the off-policy game Q-learning Algorithm is proposed and applied to  $H_\infty$  control of multi-player systems.

*Remark 11:* The proposed off-policy game Q-learning Algorithm 2 does not require any knowledge of the system dynamics, such that the control policies, which can guarantees  $L_2$  gain less than  $\gamma$  or equal to  $\gamma$  and the closed-loop multi-player system is asymptotically stable when  $d_k \equiv 0$ , can be found for unknown multi-player systems

unlike [14], [17] which require the accurate system model to be known.

## V. SIMULATION RESULTS

In this section, through  $H_\infty$  control simulations of three-player and five-player systems, the effectiveness of the proposed off-policy game Q-learning Algorithm 2 is verified. Moreover, comparative simulation experiments with [32], [33] where disturbance is ignored are carried out to show the advantages of the proposed off-policy game Q-learning algorithm for systems subject to external disturbance. It is assumed that the dynamics  $A$ ,  $B_i$  and  $E$  are completely unknown during the learning process of algorithm.

The state trajectories of the system are assumed to be subject to the following disturbance.

$$d_k = e^{-0.0001k} \sin(2.0k) \quad (46)$$

### A. COMPARISON RESULTS OF ON-POLICY LEARNING WITH OFF-POLICY LEARNING

In this part, a three-player system is used as the object of simulation experiments when implementing on-policy game Q-learning Algorithm 1 and off-policy game Q-learning Algorithm 2.

Consider the following linear DT system with three-player and disturbance input:

$$x_{k+1} = Ax_k + B_1 u_1 + B_2 u_2 + B_3 u_3 + Ed_k \quad (47)$$

where

$$\begin{aligned} A &= \begin{bmatrix} 0.906488 & 0.0816012 & -0.0005 \\ 0.074349 & 0.90121 & -0.000708383 \\ 0 & 0 & 0.132655 \end{bmatrix} \\ B_1 &= \begin{bmatrix} -0.00150808 \\ -0.0096 \\ 0.867345 \end{bmatrix}, \quad B_2 = \begin{bmatrix} 0.00951892 \\ 0.00038373 \\ 0 \end{bmatrix} \\ B_3 &= \begin{bmatrix} -0.00563451 \\ -0.08962 \\ 0.356478 \end{bmatrix}, \quad E = \begin{bmatrix} 0.0123956 \\ 0.068 \\ -0.05673 \end{bmatrix} \end{aligned}$$

Choose  $Q = \text{diag}(5, 5, 5)$  and  $R_1 = R_2 = R_3 = 1$ . The disturbance attenuation factor is selected to be  $\gamma = 1$ . Rewrite (22) as

$$H^* = \Lambda + G^T H^* G \quad (48)$$

where

$$G = \begin{bmatrix} A & B_1 & B_2 & B_3 & E \\ -K_1 A & -K_1 B_1 & -K_1 B_2 & -K_1 B_3 & -K_1 E \\ -K_2 A & -K_2 B_1 & -K_2 B_2 & -K_2 B_3 & -K_2 E \\ -K_3 A & -K_3 B_1 & -K_3 B_2 & -K_3 B_3 & -K_3 E \end{bmatrix}$$

The theoretical solution to (48) can be obtained by using MATLAB. Thus, the optimal Q-function matrix  $H^*$  and the optimal controller gains  $(K_1^*, K_2^*, K_3^*)$  and the worst-case

TABLE 1.  $H_\infty$  controller gains under three probing noises.

	on-policy game Q-learning	off-policy game Q-learning
Case 1	$K_1 = \begin{bmatrix} -0.5666 & -0.7299 & -0.1098 \end{bmatrix}$ $K_2 = \begin{bmatrix} -0.4271 & -0.2753 & -0.0009 \end{bmatrix}$ $K_3 = \begin{bmatrix} 2.2645 & 2.8474 & -0.0329 \end{bmatrix}$ $K = \begin{bmatrix} 2.2926 & 2.6571 & 0.0032 \end{bmatrix}$	$K_1 = \begin{bmatrix} -0.5665 & -0.7300 & -0.1098 \end{bmatrix}$ $K_2 = \begin{bmatrix} -0.4271 & -0.2752 & -0.0009 \end{bmatrix}$ $K_3 = \begin{bmatrix} 2.2643 & 2.8476 & -0.0329 \end{bmatrix}$ $K = \begin{bmatrix} 2.2925 & 2.6572 & 0.0032 \end{bmatrix}$
Case 2	$K_1 = \begin{bmatrix} -0.5635 & -0.7372 & -0.1099 \end{bmatrix}$ $K_2 = \begin{bmatrix} -0.4279 & -0.2738 & -0.0009 \end{bmatrix}$ $K_3 = \begin{bmatrix} 2.2540 & 2.8714 & -0.0328 \end{bmatrix}$ $K = \begin{bmatrix} 2.2844 & 2.6765 & 0.0033 \end{bmatrix}$	$K_1 = \begin{bmatrix} -0.5665 & -0.7300 & -0.1098 \end{bmatrix}$ $K_2 = \begin{bmatrix} -0.4271 & -0.2752 & -0.0009 \end{bmatrix}$ $K_3 = \begin{bmatrix} 2.2643 & 2.8476 & -0.0329 \end{bmatrix}$ $K = \begin{bmatrix} 2.2925 & 2.6572 & 0.0032 \end{bmatrix}$
Case 3	$K_1 = \begin{bmatrix} -0.7767 & -0.1625 & -0.0974 \end{bmatrix}$ $K_2 = \begin{bmatrix} 0.1022 & -1.4765 & -0.0268 \end{bmatrix}$ $K_3 = \begin{bmatrix} 2.8057 & 1.3265 & -0.0663 \end{bmatrix}$ $K = \begin{bmatrix} 1.7373 & 3.7918 & 0.0220 \end{bmatrix}$	$K_1 = \begin{bmatrix} -0.5665 & -0.7300 & -0.1098 \end{bmatrix}$ $K_2 = \begin{bmatrix} -0.4271 & -0.2752 & -0.0009 \end{bmatrix}$ $K_3 = \begin{bmatrix} 2.2643 & 2.8476 & -0.0329 \end{bmatrix}$ $K = \begin{bmatrix} 2.2925 & 2.6572 & 0.0032 \end{bmatrix}$

disturbance policy gain  $K^*$  can be obtained below:

$$H^* = \begin{bmatrix} 47.2688 & 28.7098 & -0.0292 & -0.2510 \\ 28.7098 & 40.9565 & -0.0241 & -0.2922 \\ -0.0292 & -0.0241 & 5.0883 & 0.5774 \\ -0.2510 & -0.2922 & 0.5774 & 4.7762 \\ 0.4329 & 0.2868 & -0.0003 & -0.0025 \\ -2.7108 & -3.4598 & 0.2395 & 1.5791 \\ 2.4375 & 2.8845 & -0.0398 & -0.2704 \\ 0.4329 & -2.7108 & 2.4375 & \\ 0.2868 & -3.4598 & 2.8845 & \\ -0.0003 & 0.2395 & -0.0398 & \\ -0.0025 & 1.5791 & -0.2704 & \\ 1.0044 & -0.0270 & 0.0244 & \\ -0.0270 & 1.9705 & -0.3786 & \\ 0.0244 & -0.3786 & -0.7515 & \end{bmatrix}$$

$$\begin{aligned} K_1^* &= \begin{bmatrix} -0.5665 & -0.7300 & -0.1098 \end{bmatrix} \\ K_2^* &= \begin{bmatrix} -0.4271 & -0.2752 & -0.0009 \end{bmatrix} \\ K_3^* &= \begin{bmatrix} 2.2643 & 2.8476 & -0.0329 \end{bmatrix} \\ K^* &= \begin{bmatrix} 2.2925 & 2.6572 & 0.0032 \end{bmatrix} \end{aligned} \quad (49)$$

Three types of probing noise are used to demonstrate whether the on-policy game Q-learning Algorithm 1 and the off-policy game Q-learning Algorithm 2 would produce bias of solution to Q-function based iterative Bellman equations (25) and (40), when adding probing noises. The following three probing noises are considered.

1) Case 1:

$$e_i = \sum_j^{100} 0.5 * \sin(\text{noise}_{feq}(1, j) * k) \quad (50)$$

2) Case 2:

$$e_i = \sum_j^{100} 6 * \sin(\text{noise}_{feq}(1, j) * k) \quad (51)$$

3) Case 3:

$$e_i = \sum_j^{100} 10 * \sin(\text{noise}_{feq}(1, j) * k) \quad (52)$$

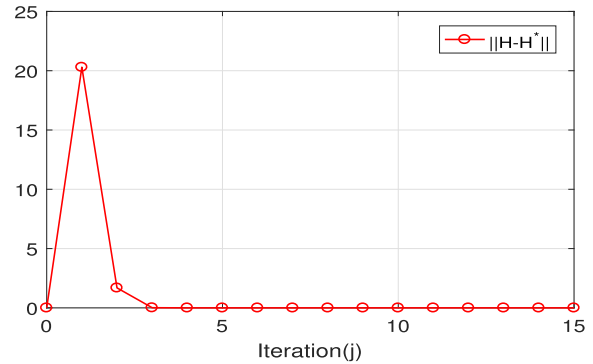


FIGURE 2. Case 1: Convergence of  $H$  when implementing the on-policy game Q-learning.

where

$$\text{noise}_{feq}(1, j) = 500 * \text{rand}(1, 100) - 200 * \text{ones}(1, 100) \quad (53)$$

Table 1 shows the controller gains and the worst-case disturbance policy learned by Algorithm 1 and Algorithm 2 under these three kinds of probing noises. It can be seen that Algorithm 1 is affected by probing noises, and its final controller gain is deviated from the theoretical value. However, Algorithm 2 is not affected by probing noises and can converge to the theoretical optimal value.

Under Case 1, the convergence process of matrix  $H^j$ , the control policy gains and disturbance policy gain  $(K_1^j, K_2^j, K_3^j, K^j)$  can be seen in Fig. 2 and Fig. 3. Fig. 4 shows the system state when implementing Algorithm 1.

Fig. 5 and Fig. 6 show the convergence state of matrix  $H^j$ , the control policy gain and disturbance policy gain  $(K_1^j, K_2^j, K_3^j, K^j)$  under Case 2, and Fig. 7 shows the evolution process of the system trajectory under the learned control policies using Algorithm 1.

Algorithm 1 was implemented under Case 3, it can be seen from Fig. 8 and Fig. 9 that the convergence process of the matrix  $H^j$ , the control policy gains and disturbance policy gain  $(K_1^j, K_2^j, K_3^j, K^j)$ , and Fig. 10 shows the state responses of the system under the controller learned by Algorithm 1, and Fig. 11 shows the system performance  $J$  under the control policies learned by the Algorithm 1.

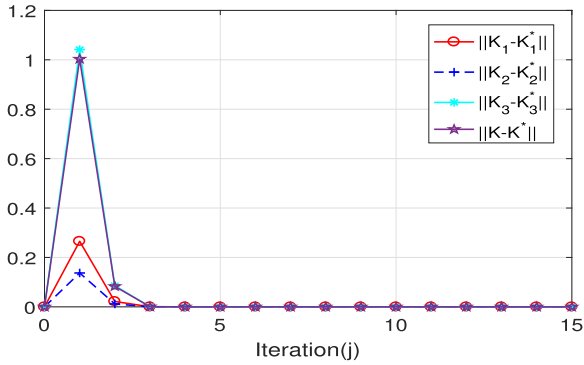


FIGURE 3. Case 1: Convergence of  $K_i (i = 1, 2, 3)$  and  $K$  when implementing the on-policy game Q-learning.

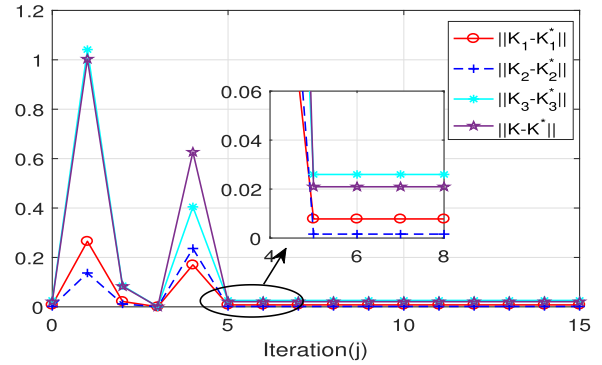


FIGURE 6. Case 2: Convergence of  $K_i (i = 1, 2, 3)$  and  $K$  when implementing the on-policy game Q-learning.

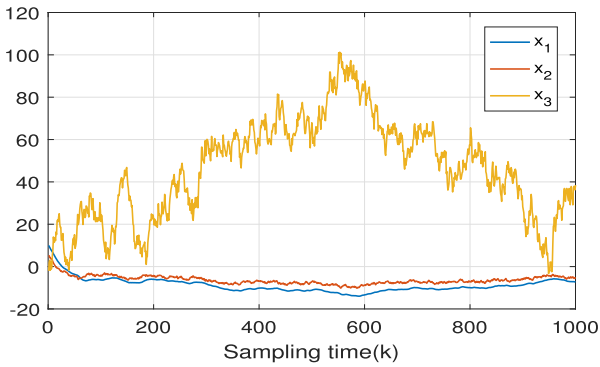


FIGURE 4. Case 1: The states  $x$  of system when implementing the on-policy game Q-learning.

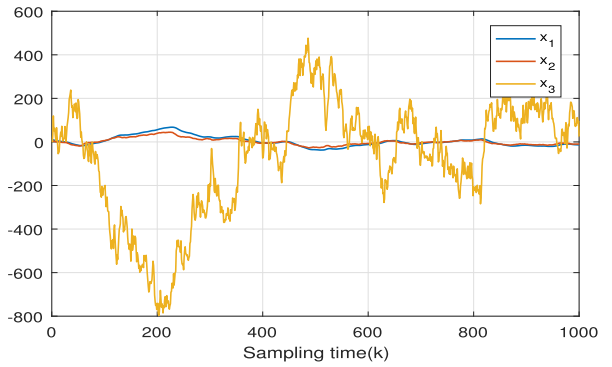


FIGURE 7. Case 2: The states  $x$  of system when implementing the on-policy game Q-learning.

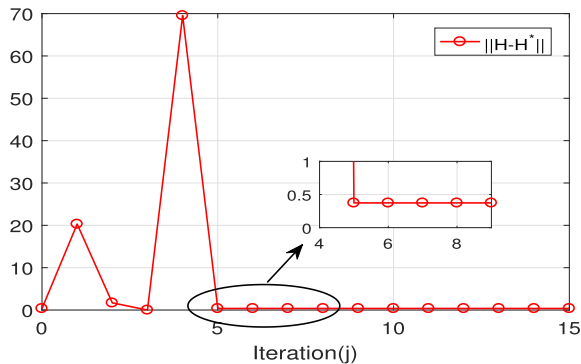


FIGURE 5. Case 2: Convergence of  $H$  when implementing the on-policy game Q-learning.

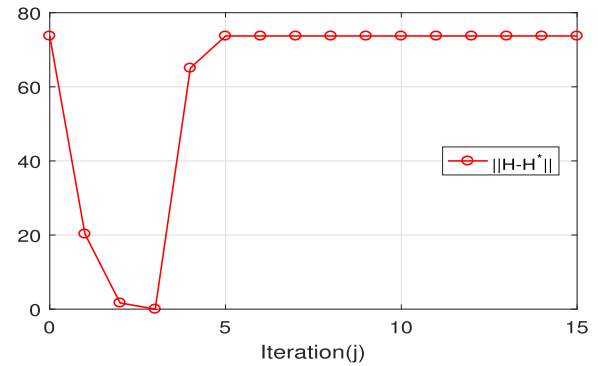


FIGURE 8. Case 3: Convergence of  $H$  when implementing the on-policy game Q-learning.

Now, we start to plot simulation results using the learned controller gains from Algorithm 2. The convergence results of matrix  $H^j$ , the control policy gains and disturbance policy gain  $(K_1^j, K_2^j, K_3^j, K^j)$  under Case 1 of probing noises are shown in Fig. 12 and Fig. 13. Fig. 14 shows the state trajectory of the system, and Fig. 15 shows the performance  $J$  of the system when using Algorithm 2.

The signal to noise ratio is calculated as

$$\frac{\sum_{k=0}^{500} (x_k^T Q x_k + \sum_{i=1}^3 u_i^T R_i u_i)}{\sum_{k=0}^{500} \|d_k\|^2} = 0.2867 < 1$$

It can be seen that the disturbance attenuation condition (2) is satisfied.

As one can see in Fig. 4, 7 and 10, the state of system has been obviously affected by adding probing noises when implementing the on-policy game Q-learning. However, as shown in Fig. 14, the trajectory  $x$  of system can quickly converge to a stable state, and the cost shown in Fig. 15 is small than that when utilizing Algorithm 1.

### B. VERIFICATION OF ANTI-INTERFERENCE

In this part, five-player are used to further verify the validity of Algorithm 2. In addition, anti-interference of the proposed Algorithm 2 for  $H_\infty$  control of multi-player systems and

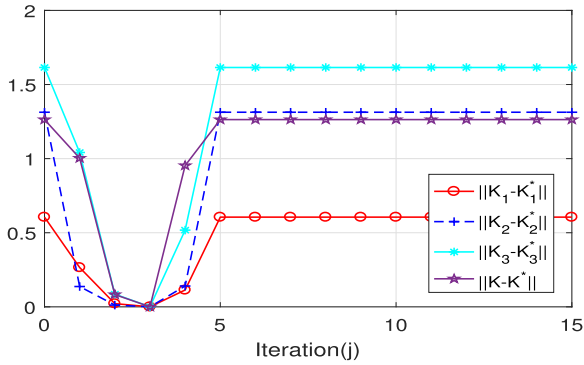


FIGURE 9. Case 3: Convergence of  $K_i$  ( $i = 1, 2, 3$ ) and  $K$  when implementing the on-policy game Q-learning.

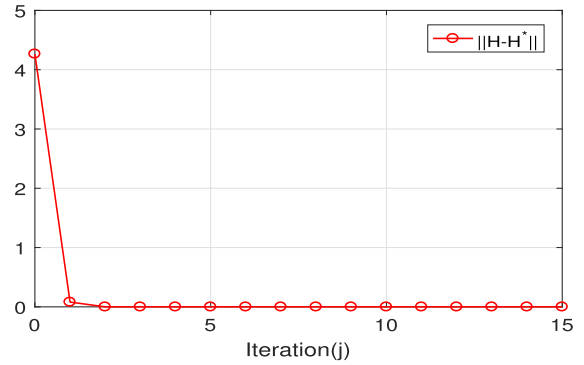


FIGURE 12. Convergence of  $H$  when implementing the off-policy game Q-learning.

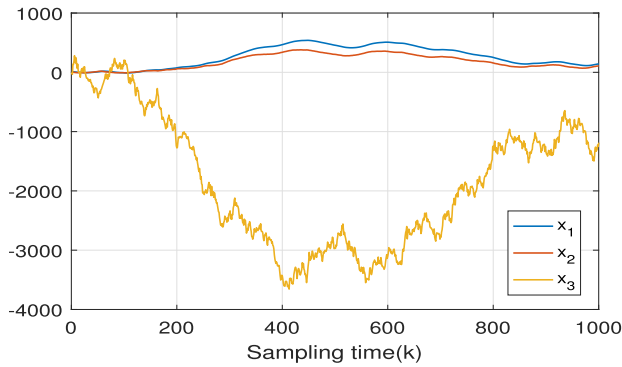


FIGURE 10. Case 3: The states  $x$  of system when implementing the on-policy game Q-learning.

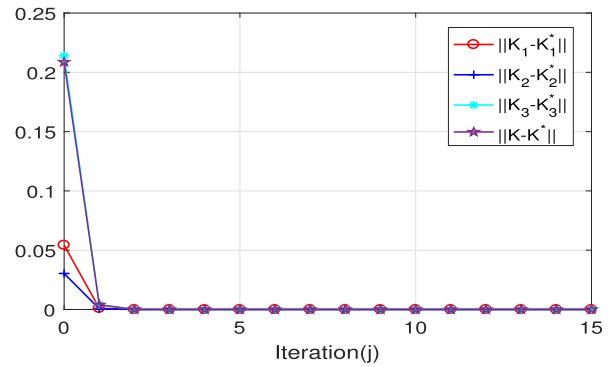


FIGURE 13. Convergence of  $K_i$  ( $i = 1, 2, 3$ ) and  $K$  when implementing the off-policy game Q-learning.

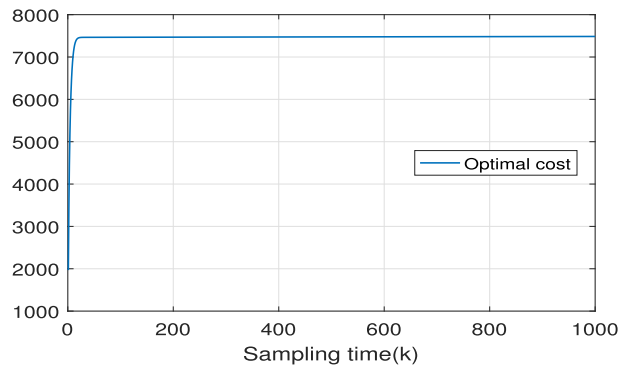


FIGURE 11. Case 3: The cost  $J$  of system when implementing the on-policy game Q-learning.

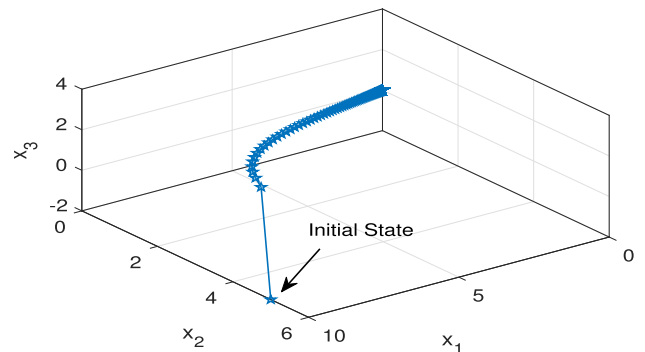


FIGURE 14. The states  $x$  of system when implementing the off-policy game Q-learning.

the advantage over the methods without taking into external disturbance account are demonstrated.

Consider the following linear DT system with five players and disturbance input:

$$x_{k+1} = Ax_k + B_1u_1 + B_2u_2 + B_3u_3 + B_4u_4 + B_5u_5 + Ed_k \quad (54)$$

where

$$A = \begin{bmatrix} 0.906488 & 0.0816012 & -0.0005 \\ 0.074349 & 0.90121 & -0.000708383 \\ 0 & 0 & 0.132655 \end{bmatrix}$$

$$B_1 = \begin{bmatrix} -0.00150808 \\ -0.0096 \\ 0.867345 \end{bmatrix}, \quad B_2 = \begin{bmatrix} 0.00951892 \\ 0.00038373 \\ 0 \end{bmatrix}$$

$$B_3 = \begin{bmatrix} -0.00563451 \\ -0.08962 \\ 0.356478 \end{bmatrix}, \quad B_4 = \begin{bmatrix} 0.1568 \\ 0.006018 \\ -0.18235673 \end{bmatrix}$$

$$B_5 = \begin{bmatrix} -0.125 \\ 0.4 \\ -0.4898 \end{bmatrix}, \quad E = \begin{bmatrix} 0.0123956 \\ 0.068 \\ -0.05673 \end{bmatrix}$$

Choose  $Q = \text{diag}(5, 5, 5)$  and  $R_1 = R_2 = R_3 = R_4 = R_5 = 1$ . The disturbance attenuation factor is selected to

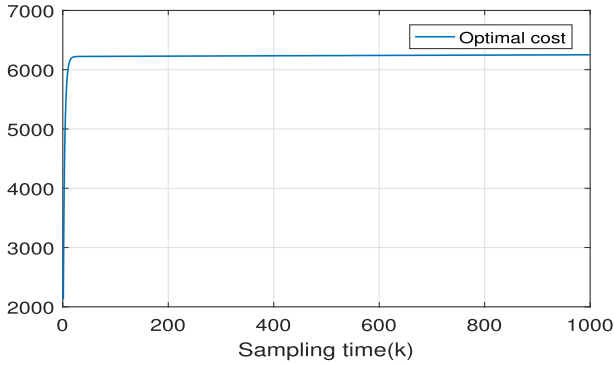


FIGURE 15. The cost  $J$  of system when implementing the off-policy game Q-learning.

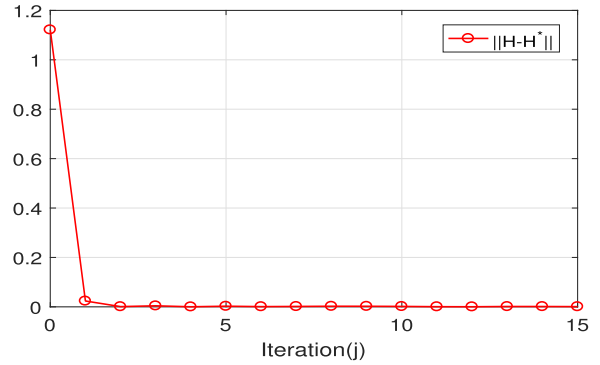


FIGURE 16. Convergence of  $H$  when implementing the off-policy game Q-learning.

be  $\gamma = 1$ . Rewrite (22) as

$$H^* = \Lambda + \begin{bmatrix} A & B_1 & \dots & B_5 & E \\ -K_1A & -K_1B_1 & \dots & -K_1B_5 & -K_1E \\ -K_2A & -K_2B_1 & \dots & -K_2B_5 & -K_2E \\ -K_3A & -K_3B_1 & \dots & -K_3B_5 & -K_3E \\ -K_4A & -K_4B_1 & \dots & -K_4B_5 & -K_4E \\ -K_5A & -K_5B_1 & \dots & -K_5B_5 & -K_5E \end{bmatrix}^T \times H^* \begin{bmatrix} A & B_1 & \dots & B_5 & E \\ -K_1A & -K_1B_1 & \dots & -K_1B_5 & -K_1E \\ -K_2A & -K_2B_1 & \dots & -K_2B_5 & -K_2E \\ -K_3A & -K_3B_1 & \dots & -K_3B_5 & -K_3E \\ -K_4A & -K_4B_1 & \dots & -K_4B_5 & -K_4E \\ -K_5A & -K_5B_1 & \dots & -K_5B_5 & -K_5E \end{bmatrix} \quad (55)$$

Similarly, the theoretical solution to (55) can be obtained using MATLAB software.

$$H^* = \begin{bmatrix} 17.2288 & 4.9395 & -0.0033 & -0.0201 & \\ 4.9395 & 13.7043 & 0.0036 & -0.0142 & \\ -0.0033 & 0.0036 & 5.0882 & 0.5769 & \\ -0.0201 & -0.0142 & 0.5769 & 4.7722 & \\ 0.1267 & 0.0482 & -0.00003 & -0.0002 & \\ -0.4418 & -0.8204 & 0.2368 & 1.5517 & \\ 2.0775 & 0.7744 & -0.1219 & -0.7964 & \\ 0.0509 & 3.0606 & -0.3236 & -2.1330 & \\ 0.4520 & 0.6807 & -0.0375 & -0.2479 & \\ 0.1267 & -0.4418 & 2.0775 & 0.0509 & 0.4520 \\ 0.0482 & -0.8204 & 0.7744 & 3.0606 & 0.6807 \\ -0.00003 & 0.2368 & -0.1219 & -0.3236 & -0.0375 \\ -0.0002 & 1.5517 & -0.7964 & -2.1330 & -0.2479 \\ 1.0013 & -0.0043 & 0.0216 & -0.0008 & 0.0045 \\ -0.0043 & 1.7146 & -0.3948 & -1.1682 & -0.1654 \\ 0.0216 & -0.3948 & 1.5203 & 0.4272 & 0.1239 \\ -0.0008 & -1.1682 & 0.4272 & 3.5806 & 0.3669 \\ 0.0045 & -0.1654 & 0.1239 & 0.3669 & -0.9302 \end{bmatrix}$$

$$K_1^* = [-0.2870 \quad -0.6379 \quad -0.0098]$$

$$K_2^* = [-0.0950 \quad -0.0392 \quad -0.0004]$$

$$K_3^* = [0.2282 \quad 0.1920 \quad -0.0316]$$

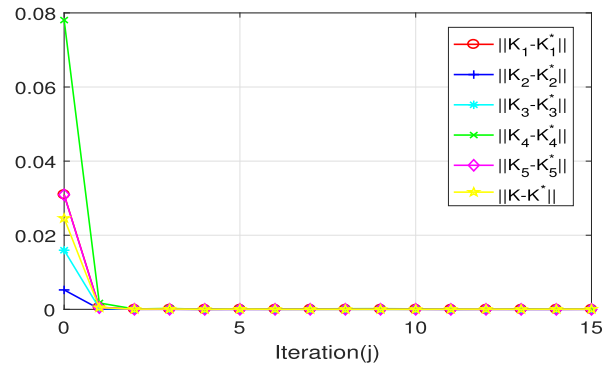


FIGURE 17. Convergence of  $K_i (i = 1, 2, \dots, 5)$  and  $K$  when implementing the off-policy game Q-learning.

$$K_4^* = [-1.4928 \quad -0.4988 \quad 0.0144]$$

$$K_5^* = [0.0330 \quad -1.1483 \quad 0.0188]$$

$$K^* = [0.3356 \quad 0.3481 \quad 0.0012] \quad (56)$$

Firstly, five player are used to systems to verify the effectiveness of the proposed off-policy game Q-learning Algorithm 2. It can be seen from Fig. 16 and Fig. 17 that the learned matrix  $H^j$ , controller gains and disturbance policy gain ( $K_1^j, K_2^j, K_3^j, K_4^j, K_5^j, K^j$ ) are not affected by the probing noise, and they can quickly converge to theoretical optimal values without deviation as shown in Table 2. Fig. 18 and 19 show the state trajectories and performance  $J$  of the system under the learned control policies using off-policy game Q-learning Algorithm 2.

The signal to noise ratio is calculated as

$$\frac{\sum_{k=0}^{500} (x_k^T Q x_k + \sum_{i=1}^5 u_i^T R_i u_i)}{\sum_{k=0}^{500} \|d_k\|^2} = 0.4289 < 1$$

It can be seen that the disturbance attenuation condition (2) is satisfied.

Secondly, we assume  $E = 0$  which means the external disturbance is not taken into account similar to the methods in [31]–[35], and implementing Algorithm 2 yields the optimal controller gains shown in Table 2. Then, simulation comparisons are going to be made under the following three kinds of external disturbances.



TABLE 2. Controller gains when considering the disturbance or not.

	off-policy game Q-learning( $E \neq 0$ )	off-policy Q-learning without consider disturbance( $E = 0$ )
probing noise Case 1/2/3	$K_1 = [-0.2870 \quad -0.6379 \quad -0.0098]$ $K_2 = [-0.0950 \quad -0.0392 \quad -0.0004]$ $K_3 = [0.2282 \quad 0.1920 \quad -0.0316]$ $K_4 = [-1.4928 \quad -0.4988 \quad 0.0144]$ $K_5 = [0.0330 \quad -1.1483 \quad 0.0188]$ $K = [0.3356 \quad 0.3481 \quad 0.0012]$	$K_1 = [-0.2743 \quad -0.6278 \quad -0.0996]$ $K_2 = [-0.0926 \quad -0.0371 \quad -0.0004]$ $K_3 = [0.2144 \quad 0.1790 \quad -0.0317]$ $K_4 = [-1.4563 \quad -0.4669 \quad 0.0147]$ $K_5 = [0.0768 \quad -1.1065 \quad 0.0191]$

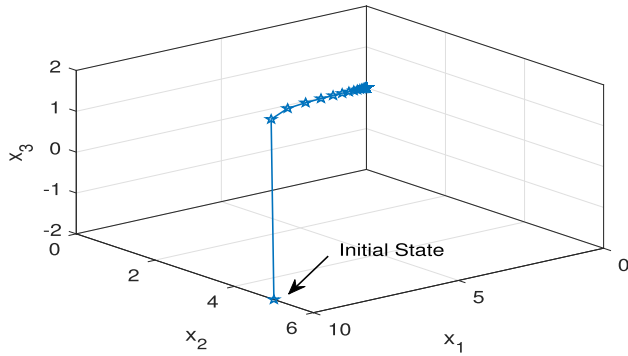


FIGURE 18. The states  $x$  of system when implementing the off-policy game Q-learning.

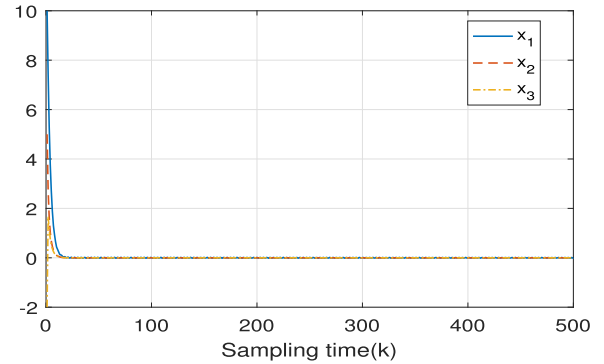


FIGURE 21. Case 1: The states  $x$  of system when implementing the off-policy game Q-learning( $E \neq 0$ ).

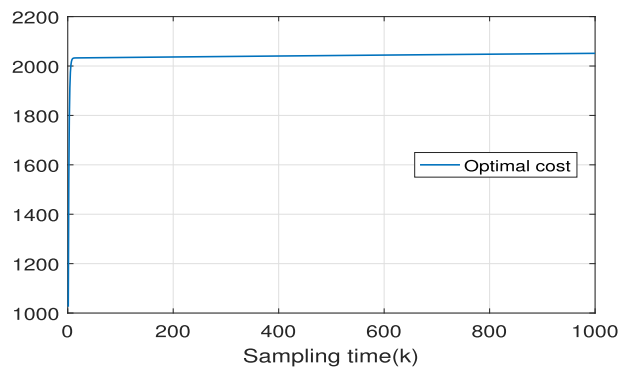


FIGURE 19. The cost  $J$  of system when implementing the off-policy game Q-learning.

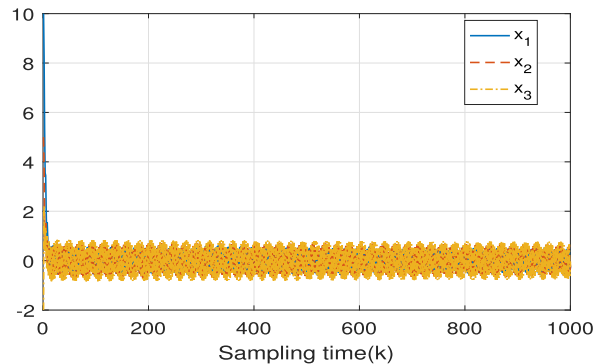


FIGURE 22. Case 2: The states  $x$  of system when implementing the off-policy Q-learning( $E = 0$ ).

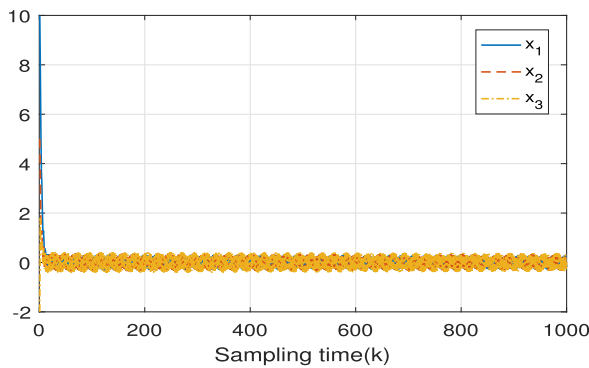


FIGURE 20. Case 1: The states  $x$  of system when implementing the off-policy Q-learning( $E = 0$ ).

1) Case 1:

$$d_k = 0.5e^{-0.001k} \sin(2.0k) \quad (57)$$

2) Case 2:

$$d_k = e^{-0.001k} \sin(2.0k) \quad (58)$$

3) Case 3:

$$d_k = 5e^{-0.001k} \sin(2.0k) \quad (59)$$

Under the three kinds of disturbance, Fig. 20, 22 and 24 plot the state trajectories of the system under the controllers without considering the external disturbances. By comparing the results in Fig. 20, 22 and 24 with those in Fig. 21, 23 and 25, it can be seen that under the same external disturbance, the system state obtained by considering the interference in the learning process always tends to be stable, while the system state obtained without considering the interference will be greatly affected by the external disturbances.

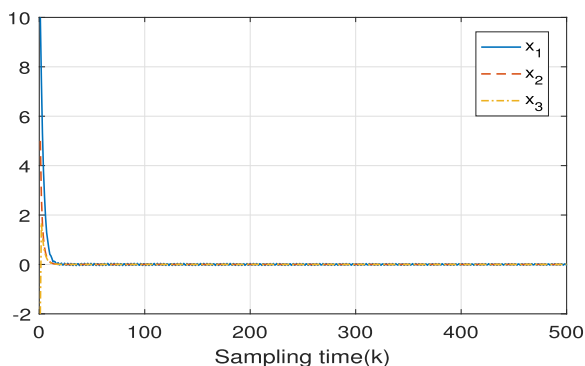


FIGURE 23. Case 2: The states  $x$  of system when implementing the off-policy game Q-learning( $E \neq 0$ ).

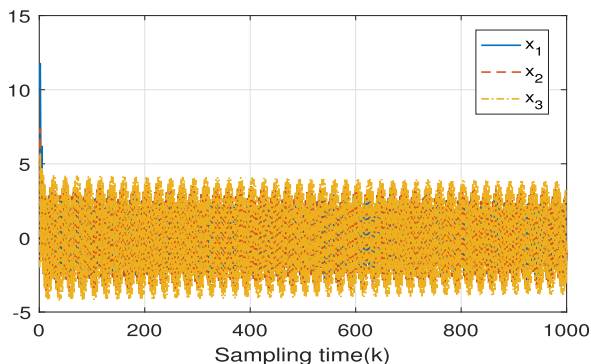


FIGURE 24. Case 3: The states  $x$  of system when implementing the off-policy Q-learning( $E = 0$ ).

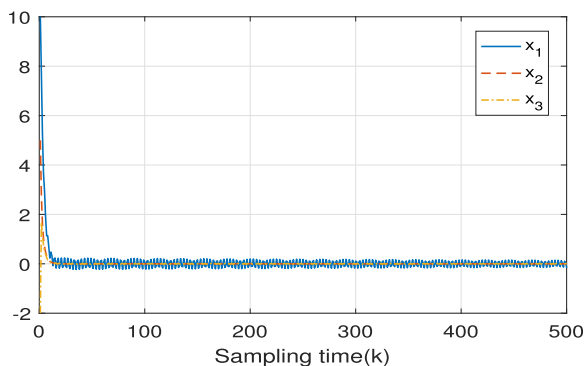


FIGURE 25. Case 3: The states  $x$  of system when implementing the off-policy game Q-learning( $E \neq 0$ ).

## VI. CONCLUSION

In this paper, a novel off-policy game Q-learning algorithm is proposed to solve the  $H_\infty$  control problem for multi-player linear DT systems. The proposed algorithm does not need to know the dynamics model of the system in advance and is complete data-driven. From rigorous theoretical proof and the simulation results, the probing noise added to the system will not affect the Nash equilibrium solution learned by the proposed algorithm, which means the learning results can converge to the optimal value without deviation. Finally, the effectiveness of the proposed method is verified by the simulation results.

## REFERENCES

- [1] A. van der Schaft, “ $L_2$ -gain analysis of nonlinear systems and nonlinear state feedback  $H_\infty$  control,” *IEEE Trans. Autom. Control*, vol. 37, no. 6, pp. 770–784, Jun. 1992.
- [2] A. Isidori, “ $H_\infty$  control via measurement feedback for affine nonlinear systems,” *Int. J. Robust Nonlinear Control*, vol. 4, no. 4, pp. 553–574, 1994.
- [3] C. Wang, Z. Zuo, Z. Qi, and Z. Ding, “Predictor-based extended-state-observer design for consensus of MASs with delays and disturbances,” *IEEE Trans. Cybern.*, vol. 49, no. 4, pp. 1259–1269, Apr. 2019.
- [4] T. Basar,  *$H_\infty$  Optimal Control and Related Minimax Design Problems*. Boston, MA, USA: Birkhäuser, 2008.
- [5] A. Al-Tamimi, M. Abu-Khalaf, and F. L. Lewis, “Adaptive critic designs for discrete-time zero-sum games with application to  $H_\infty$  control,” *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 37, no. 1, pp. 240–247, Feb. 2007.
- [6] H. Jiang, H. Zhang, J. Han, and K. Zhang, “Iterative adaptive dynamic programming methods with neural network implementation for multi-player zero-sum games,” *Neurocomputing*, vol. 307, pp. 54–60, Sep. 2018.
- [7] Q. Jiao, H. Modares, S. Xu, F. L. Lewis, and K. G. Vamvoudakis, “Multi-agent zero-sum differential graphical games for disturbance rejection in distributed control,” *Automatica*, vol. 69, pp. 24–34, Jul. 2016.
- [8] Y. Lv and X. Ren, “Approximate Nash solutions for multiplayer mixed-zero-sum game with reinforcement learning,” *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 49, no. 12, pp. 2739–2750, Dec. 2019.
- [9] R. Song, Q. Wei, and B. Song, “Neural-network-based synchronous iteration learning method for multi-player zero-sum games,” *Neurocomputing*, vol. 242, pp. 73–82, Jun. 2017.
- [10] L. El Ghaoui, F. Oustry, and M. Aitrami, “A cone complementarity linearization algorithm for static output-feedback and related problems,” *IEEE Trans. Autom. Control*, vol. 42, no. 8, pp. 1171–1176, Aug. 1997.
- [11] J. Geromel, C. De Souza, and R. Skelton, “Static output feedback controllers: Stability and convexity,” *IEEE Trans. Autom. Control*, vol. 43, no. 1, pp. 120–125, Jan. 1998.
- [12] Y.-Y. Cao, J. Lam, and Y.-X. Sun, “Static output feedback stabilization: An ILMI approach,” *Automatica*, vol. 34, no. 12, pp. 1641–1645, Dec. 1998.
- [13] Y. Jiang and Z.-P. Jiang, “Computational adaptive optimal control for continuous-time linear systems with completely unknown dynamics,” *Automatica*, vol. 48, no. 10, pp. 2699–2704, Oct. 2012.
- [14] K. G. Vamvoudakis and F. Lewis, “Online solution of nonlinear two-player zero-sum games using synchronous policy iteration,” *Int. J. Robust Nonlinear Control*, vol. 22, no. 13, pp. 1460–1483, Sep. 2012.
- [15] P. Werbos, “Approximate dynamic programming for realtime control and neural modelling,” in *Handbook of Intelligent Control: Neural, Fuzzy, and Adaptive Approaches*. New York, NY, USA: Van Nostrand Reinhold, 1992, pp. 493–525.
- [16] D. Vrabie, “Online adaptive optimal control for continuous-time systems,” Univ. Texas Arlington, Arlington, TX, USA, Tech. Rep. Vrabie\_uta\_2502D\_10530, 2010.
- [17] K. G. Vamvoudakis and F. L. Lewis, “Online actor–critic algorithm to solve the continuous-time infinite horizon optimal control problem,” *Automatica*, vol. 46, no. 5, pp. 878–888, 2010.
- [18] J. Kautsky, N. K. Nichols, and P. Van Dooren, “Robust pole assignment in linear state feedback,” *Int. J. Control*, vol. 41, no. 5, pp. 1129–1155, May 1985.
- [19] F. Amini and M. Z. Samani, “A wavelet-based adaptive pole assignment method for structural control,” *Comput.-Aided Civil Infrastruct. Eng.*, vol. 29, no. 6, pp. 464–477, Jul. 2014.
- [20] R. Byers and S. G. Nash, “Approaches to robust pole assignment,” *Int. J. Control*, vol. 49, no. 1, pp. 97–117, Jan. 1989.
- [21] A. Al-Tamimi, F. L. Lewis, and M. Abu-Khalaf, “Model-free Q-learning designs for linear discrete-time zero-sum games with application to  $H_\infty$  control,” *Automatica*, vol. 43, no. 3, pp. 473–481, 2007.
- [22] B. Kiumarsi, F. L. Lewis, and Z. Jiang, “ $H_\infty$  control of linear discrete-time systems: Off-policy reinforcement learning,” *Automatica*, vol. 78, pp. 144–152, Apr. 2017.
- [23] J. H. Kim and F. L. Lewis, “Model-free  $H_\infty$  control design for unknown linear discrete-time systems via Q-learning with LMI,” *Automatica*, vol. 46, no. 8, pp. 1320–1326, 2010.
- [24] S. A. A. Rizvi and Z. Lin, “Output feedback Q-learning for discrete-time linear zero-sum games with application to the h-infinity control,” *Automatica*, vol. 95, pp. 213–221, 2018.

- [25] H. Modares, F. L. Lewis, and Z.-P. Jiang, " $H_\infty$  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, Jun. 2015.
- [26] H. Jiang, H. Zhang, Y. Luo, and X. Cui, " $H_\infty$  control with constrained input for completely unknown nonlinear systems using data-driven reinforcement learning method," *Neurocomputing*, vol. 237, pp. 226–234, 2017.
- [27] B. Luo, T. Huang, H. Wu, and X. Yang, "Data-driven  $H_\infty$  control for nonlinear distributed parameter systems," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 26, no. 11, pp. 2949–2961, Nov. 2015.
- [28] B. Luo, H.-N. Wu, and T. Huang, "Off-policy reinforcement learning for  $H_\infty$  control design," *IEEE Trans. Cybern.*, vol. 45, no. 1, pp. 65–76, Jan. 2015.
- [29] B. Kiumarsi, W. Kang, and F. L. Lewis, " $H_\infty$  control of nonaffine aerial systems using off-policy reinforcement learning," *Unmanned Syst.*, vol. 4, no. 01, pp. 51–60, 2016.
- [30] Y. Fu and T. Chai, "Online solution of two-player zero-sum games for continuous-time nonlinear systems with completely unknown dynamics," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 27, no. 12, pp. 2577–2587, Dec. 2016.
- [31] J. Li, H. Modares, T. Chai, F. L. Lewis, and L. Xie, "Off-policy reinforcement learning for synchronization in multiagent graphical games," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 28, no. 10, pp. 2434–2445, Oct. 2017.
- [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] 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, Aug. 2011.
- [34] K. G. Vamvoudakis, F. L. Lewis, and G. R. Hudas, "Multi-agent differential graphical games: Online adaptive learning solution for synchronization with optimality," *Automatica*, vol. 48, no. 8, pp. 1598–1611, Aug. 2012.
- [35] K. G. Vamvoudakis, H. Modares, B. Kiumarsi, and F. L. Lewis, "Game theory-based control system algorithms with real-time reinforcement learning: How to solve multiplayer games online," *IEEE Control Syst.*, vol. 37, no. 1, pp. 33–52, Feb. 2017.
- [36] K. G. Vamvoudakis, "Q-learning for continuous-time graphical games on large networks with completely unknown linear system dynamics," *Int. J. Robust. Nonlinear Control*, vol. 27, no. 16, pp. 2900–2920, Nov. 2017.
- [37] J. Li, Z. Xiao, and P. Li, "Discrete-time multi-player games based on off-policy Q-learning," *IEEE Access*, vol. 7, pp. 134647–134659, 2019.
- [38] H. Modares, F. L. Lewis, and M.-B. N. Sistani, "Online solution of nonquadratic two-player zero-sum games arising in the  $H_\infty$  control of constrained input systems," *Int. J. Adapt. Control Signal Process.*, vol. 28, nos. 3–5, pp. 232–254, 2014.
- [39] J. Li, J. Ding, T. Chai, and F. L. Lewis, "Nonzero-sum game reinforcement learning for performance optimization in large-scale industrial processes," *IEEE Trans. Cybern.*, to be published.
- [40] K. G. Vamvoudakis, "Q-learning for continuous-time linear systems: A model-free infinite horizon optimal control approach," *Syst. Control Lett.*, vol. 100, pp. 14–20, Feb. 2017.
- [41] B. Kiumarsi, F. L. Lewis, H. Modares, A. Karimpour, and M.-B. Naghibi-Sistani, "Reinforcement-learning for optimal tracking control of linear discrete-time systems with unknown dynamics," *Automatica*, vol. 50, no. 4, pp. 1167–1175, Apr. 2014.
- [42] H. Modares and F. L. Lewis, "Optimal tracking control of nonlinear partially-unknown constrained-input systems using integral reinforcement learning," *Automatica*, vol. 50, no. 7, pp. 1780–1792, Jul. 2014.
- [43] A. Al-Tamimi, F. Lewis, and M. Abu-Khalaf, "Discrete-time nonlinear HJB solution using approximate dynamic programming: Convergence proof," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 38, no. 4, pp. 943–949, Aug. 2008.
- [44] J. Li, T. Chai, F. L. Lewis, Z. Ding, and Y. Jiang, "Off-Policy interleaved Q-learning: Optimal control for affine nonlinear discrete-time systems," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 30, no. 5, pp. 1308–1320, May 2019.
- [45] Q. Wei, F. L. Lewis, Q. Sun, P. Yan, and R. Song, "Discrete-time deterministic Q-learning: A novel convergence analysis," *IEEE Trans. Cybern.*, vol. 47, no. 5, pp. 1224–1237, May 2017.
- [46] J. Y. Lee, J. B. Park, and Y. H. Choi, "Integral Q-learning and explored policy iteration for adaptive optimal control of continuous-time linear systems," *Automatica*, vol. 48, no. 11, pp. 2850–2859, Nov. 2012.
- [47] 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.
- [48] R. Munos, T. Stepleton, A. Harutyunyan, and M. Bellemare, "Safe and efficient off-policy reinforcement learning," in *Proc. Adv. Neural Inf. Process. Syst.*, 2016, pp. 1054–1062.



**JINNA LI** (Member, IEEE) received the M.S. and Ph.D. degrees from Northeastern University, Shenyang, China, in 2006 and 2009, respectively. From April 2009 to April 2011, she held a post-doctoral position with the Laboratory of Industrial Control Networks and Systems, Shenyang Institute of Automation, Chinese Academy of Sciences. From June 2014 to June 2015, she was a Visiting Scholar granted by the China Scholarship Council with the Energy Research Institute, Nanyang Technological University, Singapore. From September 2015 to June 2016, she was a Domestic Young Core Visiting Scholar granted by the Ministry of Education of China with the State Key Laboratory of Synthetical Automation for Process Industries, Northeastern University. From January 2017 to July 2017, she was a Visiting Scholar with the School of Electrical and Electronic Engineering, The University of Manchester, U.K. She is currently a Full Professor with the School of Information and Control Engineering, Liaoning Shihua University. Her current research interests include neural networks, reinforcement learning, optimal operational control, distributed optimization control, and data-based control.



**ZHENFEI XIAO** received the B.S. degree in electrical engineering and automation from the Great Wall College, China University of Geosciences, Baoding, China, in 2018. He is currently pursuing the M.S. degree in control theory and control engineering with the Liaoning Shihua University, Fushun, China. His current research interests include neural networks, reinforcement learning, and data-based control.

• • •