Seeking Nash Equilibrium for Linear Discrete-time Systems via Off-policy Q-learning

Haohan Ni, Yuxiang Ji, Yuxiao Yang, Jianping Zhou

***Abstract*—This paper considers a non-zero-sum game for linear discrete-time systems involving two players. Based on a quadratic value function, we derive coupled algebraic Riccati equations. Then, we propose both on-policy and off-policy Q-learning algorithms, which operate without prior knowledge of the system dynamics, to achieve Nash equilibrium. These algorithms necessitate the inclusion of probing noise to ensure the persistence of excitation. We show that the on-policy Q-learning algorithm may introduce bias to the Nash equilibrium due to the probing noise, while the off-policy Q-learning algorithm maintains an unbiased property. Finally, we offer a numerical example to validate the effectiveness of the presented on-policy and off-policy Q-learning algorithms.**

***Index Terms*—Non-zero-sum game, Q-learning, Nash equilibrium, discrete-time system, probing noise.**

I. Introduction

GAME theory has been widely applied in practical systems such as drones, autonomous driving, human-computer interaction, the medical internet of things, etc. [1–4]. So far, the main focus of research has been on two types of games: zero-sum games and non-zero-sum games. All participants strive to maximize their interests in zero-sum games [5–7]. In many cases, agents pursue team-based goals while simultaneously pursuing individual selfish goals. Cooperation and competition coexist among participants in non-zero-sum games [8, 9]. As a result, non-zero-sum games provide a better theoretical framework for cooperative and non-cooperative worlds.

One of the primary objectives of non-zero-sum games is to identify a set of optimal strategies, known as Nash equilibria [10]. Adaptive dynamic programming [11, 12], which is based on the idea of reinforcement learning, has been extensively adopted to solve optimal control problems. These methods are utilized to develop adaptive control algorithms for acquiring solutions to the algebraic Riccati equations. Notably, for continuous-time systems, Newton-Leibniz's formula is applicable, yielding integral-type Bellman equations. However, in discrete-time situations, the Bellman equations are of the infinite sum difference type, which is more complicated to solve.

Manuscript received March 29, 2024; revised October 8, 2024.

Haohan Ni is a postgraduate student at the School of Computer Science and Technology, Anhui University of Technology, Ma'anshan 243032, China (e-mail: nhhwj1@ahut.edu.cn).

Yuxiang Ji is a postgraduate student at the School of Computer Science and Technology, Anhui University of Technology, Ma'anshan 243032, China (e-mail: yxji@ahut.edu.cn).

Yuxiao Yang is a college student at the School of AI and Advanced Computing, Xi'an Jiaotong-Liverpool University, Suzhou 215000, China (e-mail: yuxiao.yang21@student.xjtlu.edu.cn)

Jianping Zhou is a full professor at the School of Computer Science and Technology, Anhui University of Technology, Ma'anshan 243032, China (corresponding author, e-mail: jpzhou@ahut.edu.cn).

Q-learning serves as a behavior-dependent heuristic dynamic programming technique [13]. For discrete-time systems, Q-learning enables agents to leverage experience data generated by behavior policies to learn target behaviors, showcasing strong exploration capabilities within the state space. Moreover, Q-learning is entirely model-free. In recent years, Q-learning methods have emerged as viable solutions for various optimal control problems [14–17]. However, to our knowledge, there exists a dearth of research concerning the application of such methods in the pursuit of Nash equilibria for non-zero-sum games in discrete-time systems, warranting further investigation.

This paper focuses on a non-zero-sum game for linear discrete-time systems involving two players. Based on a quadratic value function, we derive coupled algebraic Riccati equations. We propose both on-policy and off-policy Q-learning algorithms, which operate without prior knowledge of the system dynamics, to achieve Nash equilibrium. Both algorithms necessitate the inclusion of probing noise to ensure the persistence of excitation. Theoretically, it is established that the on-policy Q-learning algorithm leads to bias due to probing noise, whereas the off-policy Q-learning algorithm remains unaffected. Finally, we provide a numerical example to validate the effectiveness of the algorithms.

The remainder of this paper is structured as follows: In Section II, the non-zero-sum game problem is described, and the coupled algebraic Riccati equations are then derived. In Section III, the on-policy Q-learning algorithm is proposed for the case where the system dynamics are unknown. In Section IV, the off-policy Q-learning algorithm is developed, and its robustness against probing noise is proved. In Section V, simulations are conducted. Finally, our work is concluded in Section VI.

II. Problem Formulation

The notations employed throughout follow established conventions, as specified in [18, 19]. We consider a linear discrete-time system as

$$x_{k+1} = Ax_k + Bu_k + Dv_k \tag{1}$$

where $x_k \in \mathbb{R}^n$ is the system state, $u_k \in \mathbb{R}^{m_1}$ and $v_k \in \mathbb{R}^{m_2}$ are control inputs. $A \in \mathbb{R}^n$, $B \in \mathbb{R}^{m_1}$, and $D \in \mathbb{R}^{m_2}$ are unknown system matrices.

Define the performance indexes of the two players as

$$J_1(x_k) = \min_{u_k} \sum_{k=0}^{\infty} (x_k^T Q_1 x_k + u_k^T R_{11} u_k + v_k^T R_{12} v_k)$$

$$J_2(x_k) = \min_{w_k} \sum_{k=0}^{\infty} (x_k^T Q_2 x_k + u_k^T R_{21} u_k + v_k^T R_{22} v_k)$$

where $Q_i \geq 0, R_{ij} > 0, R_{ji} > 0, i, j = 1, 2$. Note that u_k and v_k represent the two players.

Assumption 1. The system (1) is controllable.

Definition 1. (Nash Equilibrium) [20] The policies $\{u^*, v^*\}$ are said to constitute a Nash equilibrium for the two-player game if the following inequalities hold:

$$J_1^*(u^*, v^*) \leq J_1^*(u, v^*)$$
$$J_2^*(u^*, v^*) \leq J_2^*(u^*, v).$$

Definition 2. [21] A r-vector sequence h = [h_1 ... h_r] is said to be persistently exciting over an interval [k + 1, k + r] if

$$\sum_{i=k+1}^{k+1} h_i h_i^T \geq \beta I \tag{2}$$

holds for some constant $\beta > 0$.

Note that if $r < q$, (2) cannot be satisfied. Define a value function as

$$V_i(x_k) = \sum_{j=0}^{\infty} (x_j^T Q_i x_j + u_j^T R_{i1} u_j + v_j^T R_{i2} v_j), \ i = 1, 2. \tag{3}$$

By using (3), one has

$$V_i(x_k) = x_k^T Q_i x_k + u_k^T R_{i1} u_k + v_k^T R_{i2} v_k + \sum_{j=k+1}^{\infty} (x_j^T Q_i x_j + u_j^T R_{i1} u_j + v_j^T R_{i2} v_j) \tag{4}$$

which yields the coupled Bellman equation:

$$V_i(x_k) = x_k^T Q_i x_k + u_k^T R_{i1} u_k + v_k^T R_{i2} v_k + V_i(x_{k+1}). \tag{5}$$

Furthermore, the goal is to identify the saddle point (u^*, v^*) such that

$$V_i^*(x_k) = \min_{u_k, v_k} \{x_k^T Q_i x_k + u_k^T R_{i1} u_k + v_k^T R_{i2} v_k + V_i(x_{k+1})\}. \tag{6}$$

From [22], (3) can be presented as the following quadratic form:

$$V_i(x_k) = x_k^T P_i x_k, \ i = 1, 2.$$

By utilizing the formula above, (5) can be expressed in the following form:

$$x_k^T P_i x_k = x_k^T Q_i x_k + u_k^T R_{i1} u_k + v_k^T R_{i2} v_k + x_{k+1}^T P_i x_{k+1}.$$

The Hamiltonian function is defined as

$$H_i(x_k, u_k, v_k) = x_k^T Q_i x_k + u_k^T R_{i1} u_k + v_k^T R_{i2} v_k + x_{k+1}^T P_i x_{k+1} - x_k^T P_i x_k.$$

The optimal control inputs $u_k^* = K^* x_k$ and $v_k^* = L^* x_k$ should satisfy simultaneously $\partial H_1(x_k, u_k, v_k)/\partial u_k = 0$ and $\partial H_2(x_k, u_k, v_k)/\partial v_k = 0$. Therefore, one has

$$\begin{aligned} K^* = & -[R_{11} + B^T P_1^* B - B^T P_1^* D(R_{22} + D^T P_2^* D)^{-1} \\ & \times D^T P_2^* B]^{-1} [B^T P_1^* A - B^T P_1^* D \\ & \times (R_{22} + D^T P_2^* D)^{-1} D^T P_2^* A] \end{aligned} \tag{7}$$

$$\begin{aligned} L^* = & -[R_{22} + D^T P_2^* D - D^T P_2^* B(R_{11} + B^T P_1^* B)^{-1} \\ & \times B^T P_1^* D]^{-1} [D^T P_2^* A - D^T P_2^* B \\ & \times (R_{11} + B^T P_1^* B)^{-1} B^T P_1^* A] \end{aligned} \tag{8}$$

where P_1^*, P_2^* satisfie the coupled algebraic Riccati equations

$$\begin{aligned} P_i^* = & \begin{bmatrix} K^* \\ L^* \end{bmatrix}^T \begin{bmatrix} R_{i1} + B^T P_i^* B & B^T P_i^* D \\ D^T P_i^* B & R_{i2} + D^T P_i^* D \end{bmatrix} \begin{bmatrix} K^* \\ L^* \end{bmatrix} \\ & + \begin{bmatrix} K^* \\ L^* \end{bmatrix}^T \begin{bmatrix} B^T P_i^* A \\ D^T P_i^* A \end{bmatrix} + \begin{bmatrix} B^T P_i^* A \\ D^T P_i^* A \end{bmatrix}^T \begin{bmatrix} K^* \\ L^* \end{bmatrix} \\ & + A^T P_i^* A + Q_i, i = 1, 2. \end{aligned} \tag{9}$$

Remark 1. It is noted that (7), (8), and (9) involve the system parameters. When these parameters are unknown or uncertain, the desired control gains K^* and L^* cannot be determined by (7), (8), and (9).

III. On-policy Q-learning for non-zero-sum game

In this section, we first provide the expression of the Q-function. Then, we derive the on-policy Q-learning algorithm for solving the two-player non-zero-sum game. Finally, we develop a theorem demonstrating that the introduction of probing noise may cause bias from the desired solution.

Based on (4), the Q-function is defined as

$$Q_i(x_k, u_k, v_k) = x_k^T Q_i x_k + u_k^T R_{i1} u_k + v_k^T R_{i2} v_k + x_{k+1}^T P_i x_{k+1}, \ i = 1, 2. \tag{10}$$

Similar to (6), we have

$$Q_i(x_k, u_k^*, v_k^*) = \min_{u_k, v_k} \{x_k^T Q_i x_k + u_k^T R_{i1} u_k + v_k^T R_{i2} v_k + x_{k+1}^T P_i x_{k+1}\}.$$

Using the system (1), the Q-function (10) can be written as

$$Q_i(x_k, u_k, v_k) = \begin{bmatrix} x_k \\ u_k \\ v_k \end{bmatrix}^T H_i \begin{bmatrix} x_k \\ u_k \\ v_k \end{bmatrix} \tag{11}$$

where

$$\begin{aligned} H_i = & \begin{bmatrix} H_{xxi} & H_{xui} & H_{xvi} \\ H_{uxi} & H_{uui} & H_{uvi} \\ H_{vxi} & H_{vui} & H_{vvi} \end{bmatrix} \\ = & \begin{bmatrix} Q_i + A^T P_i A & A^T P_i B & A^T P_i D \\ B^T P_i A & R_{i1} + B^T P_i B & B^T P_i D \\ D^T P_i A & D^T P_i B & R_{i2} + D^T P_i D \end{bmatrix}. \end{aligned}$$

By applying $\partial Q_1(x_k, u_k, v_k)/\partial u_k = 0$ and $\partial Q_2(x_k, u_k, v_k)/\partial v_k = 0$ to (11), we can obtain the optimal policies

$$\begin{aligned} K^* = & -[Huu_1 - Huv_1 Hvv_2^{-1} Hvu_2]^{-1} \\ & \times [Hux_1 - Huv_1 Hvv_2^{-1} Hvx_2] \\ L^* = & -[Hvv_2 - Hvu_2 Huu_1^{-1} Huv_1]^{-1} \\ & \times [Hvx_2 - Hvv_2 Huu_1^{-1} Hux_1]. \end{aligned}$$

From the above deductions, the expressions of optimal control gains are obtained. Note that the system dynamics parameters A, B, and D are unnecessary, as only H_1 and H_2 are required. Next, the problem is transformed to solve for the matrices H_1 and H_2 without reliance on the system parameters. In Algorithm 1, the inclusion of probing noises is essential to achieve the persistence of excitation. The actual control inputs applied to the system are $\widehat{u}_k^j = u_k^j + e_k$ and $\widehat{v}_k^j = v_k^j + w_k$.

However, the introduction of probing noises may lead to bias, as demonstrated below. Inspired by [23], we can present the following theorem.

Algorithm 1 On-policy Q-learning Algorithm

Step 1: Initialize and select initial policies that can ensure the system stability and let iteration index j = 0.

Step 2: Evaluate policies K^j and L^j by solving

$$\begin{aligned} Q_i^{j+1}(x_k,u_k,v_k) &= x_k^T Q_i x_k + u_k^T R_{i1} u_k + v_k^T R_{i2} v_k \\ &\quad + x_k^T (A+BK^j+DL^j)^T \\ &\quad \times \begin{bmatrix} I & (K^j)^T & (L^j)^T \end{bmatrix} \\ &\quad \times H_i^{j+1} \begin{bmatrix} I & (K^j)^T & (L^j)^T \end{bmatrix}^T \\ &\quad \times (A+BK^j+DL^j)x_k \end{aligned} \tag{12}$$

for H_i, i = 1, 2.

Step 3: Update the iterative feedback gains K^j and L^j according

$$\begin{aligned} K^{j+1} = & -[Huu_1^j - Huv_1^j (Hvv_2^j)^{-1} Hvu_2^j]^{-1} \\ & \times [Hux_1^j - Huv_1^j (Hvv_2^j)^{-1} Hvx_2^j] \\ L^{j+1} = & -[Hvv_2^j - Hvu_2^j (Huu_1^j)^{-1} Huv_1^j]^{-1} \\ & \times [Hvx_2^j - Hvv_2^j (Huu_1^j)^{-1} Hux_1^j]. \end{aligned}$$

Step 4: Let $j = j+1$.

Step 5: Stop if $\|K^j - K^{j+1}\| \le \varepsilon_1$ and $\|L^j - L^{j+1}\| \le \varepsilon_2$, where ε_i, i = 1,2, are a predetermined error bound; otherwise, go to Step 2.

Theorem 1. Let H_i^{j+1} be the solution to (12) with $e_k = w_k = 0$ and $\widehat{H}_i^{j+1}$ be the solution to (12) with $e_k \neq 0$ and $w_k \neq 0$. Then, $H_i^{j+1} \neq \widehat{H}_i^{j+1}$.

Proof: Rewrite (12) as

$$\begin{aligned} &\begin{bmatrix} x_k \\ K^j x_k \\ L^j x_k \end{bmatrix}^T H_i^{j+1} \begin{bmatrix} x_k \\ K^j x_k \\ L^j x_k \end{bmatrix} \\ &= (u_k^j)^T R_{i1} u_k^j + (v_k^j)^T R_{i2} v_k^j + x_k^T Q_i x_k + x_{k+1}^T \\ &\quad \times \begin{bmatrix} I \\ K^j \\ L^j \end{bmatrix}^T H_i^{j+1} \begin{bmatrix} I \\ K^j \\ L^j \end{bmatrix} x_{k+1}. \end{aligned} \tag{13}$$

By using (1) in (13), we have

$$\begin{aligned} &\begin{bmatrix} x_k \\ K^j x_k \\ L^j x_k \end{bmatrix}^T H_i^{j+1} \begin{bmatrix} x_k \\ K^j x_k \\ L^j x_k \end{bmatrix} \\ &= (u_k^j)^T R_{i1} u_k^j + (v_k^j)^T R_{i2} v_k^j + x_k^T Q_i x_k + (Ax_k + Bu_k^j \\ &\quad + Dv_k^j)^T \begin{bmatrix} I \\ K^j \\ L^j \end{bmatrix}^T H_i^{j+1} \begin{bmatrix} I \\ K^j \\ L^j \end{bmatrix} (Ax_k + Bu_k^j + Dv_k^j). \end{aligned} \tag{14}$$

After introducing the probing noises e_k and w_k, (13) becomes the following

$$\begin{aligned} &\begin{bmatrix} x_k \\ K^j x_k \\ L^j x_k \end{bmatrix}^T \widehat{H}_i^{j+1} \begin{bmatrix} x_k \\ K^j x_k \\ L^j x_k \end{bmatrix} \\ &= (u_k^j + e_k)^T R_{i1} (u_k^j + e_k) + (v_k^j + w_k)^T R_{i2} (v_k^j + w_k) \\ &\quad + x_k^T Q_i x_k + x_{k+1}^T \begin{bmatrix} I \\ K^j \\ L^j \end{bmatrix}^T \widehat{H}_i^{j+1} \begin{bmatrix} I \\ K^j \\ L^j \end{bmatrix} x_{k+1} \end{aligned} \tag{15}$$

where

$$x_{k+1} = Ax_k + B(u_k^j + e_k) + D(v_k^j + w_k).$$

Then, (15) is rewritten as

$$\begin{aligned} &\begin{bmatrix} x_k \\ K^j x_k \\ L^j x_k \end{bmatrix}^T \widehat{H}_i^{j+1} \begin{bmatrix} x_k \\ K^j x_k \\ L^j x_k \end{bmatrix} \\ &= (u_k^j)^T R_{i1} u_k^j + (v_k^j)^T R_{i2} v_k^j + x_k^T Q_i x_k + (Ax_k + Bu_k^j \\ &\quad + Dv_k^j)^T \begin{bmatrix} I \\ K^j \\ L^j \end{bmatrix}^T \widehat{H}_i^{j+1} \begin{bmatrix} I \\ K^j \\ L^j \end{bmatrix} (Ax_k + Bu_k^j + Dv_k^j) \\ &\quad + G(x_k,u_k,v_k,e_k,w_k) \end{aligned} \tag{16}$$

where

$$\begin{aligned} &G(x_k,u_k,v_k,e_k,w_k) \\ &= (Be_k + Dv_k^j)^T \begin{bmatrix} I \\ K^j \\ L^j \end{bmatrix}^T \widehat{H}_i^{j+1} \begin{bmatrix} I \\ K^j \\ L^j \end{bmatrix} (Ax_k + Bu_k^j \\ &\quad + Dv_k^j) + (Ax_k + Bu_k^j + Dv_k^j)^T \begin{bmatrix} I \\ K^j \\ L^j \end{bmatrix}^T \widehat{H}_i^{j+1} \begin{bmatrix} I \\ K^j \\ L^j \end{bmatrix} \\ &\quad \times (Be_k + Dv_k^j) + (Be_k + Dv_k^j)^T \begin{bmatrix} I \\ K^j \\ L^j \end{bmatrix}^T \widehat{H}_i^{j+1} \begin{bmatrix} I \\ K^j \\ L^j \end{bmatrix} \\ &\quad \times (Be_k + Dv_k^j) + e_k^T R_{i1} u_k^j + (u_k^j)^T R_{i1} e_k + e_k^T R_{i1} e_k \\ &\quad + w_k^T R_{i2} v_k^j + (v_k^j)^T R_{i2} w_k + w_k^T R_{i2} w_k. \end{aligned}$$

Since $G(x_k,u_k,v_k,e_k,w_k) \neq 0$ when $e_k \neq 0$ and $w_k \neq 0$, (15) contains additional terms compared to (13). Consequently, $H_i^{j+1} \neq \widehat{H}_i^{j+1}$. ■

Remark 2. Algorithm 1 exhibits biased characteristics because $u_k^j + e_k$ and $v_k^j + w_k$ are applied in the system dynamics to generate the data, while u_k^j and v_k^j are for value function evaluation. As a result, the control policies learned from the data differ from those used for evaluation.

IV. Off-policy Q-learning for non-zero-sum game

In this section, the off-policy Q-learning algorithm is developed to avoid the bias caused by probing noise. First, two auxiliary inputs are introduced to derive the new system equation and establish a new algorithm framework. Next, the Kronecker product is used to separate the unknown parameter part from the system data part. Then, the value function is iterated using the least squares method to find an approximate solution, eliminating the need for system dynamics. Finally, the unbiasedness of this algorithm is proven.

Introducing auxiliary variables $u_k^j = K^j x_k$ and $w_k^j = L^j x_k$ into system (1) yields

$$x_{k+1} = \bar{A}x_k + B(u_k - K^j x_k) + D(v_k - L^j x_k) \tag{17}$$

where $\bar{A} = A + BK^j + DL^j$. u_k and w_k are referred to as behavioral strategies for generating data, whereas u_k^j and w_k^j are termed target strategies that require improvement.

Algorithm 2 Off-policy Q-learning Algorithm

Step 1: Initialize and select the initial policies that can ensure the system stability.
Step 2: Evaluate policies K^j and L^j by solving (21) for H_i^{j+1}, $i = 1, 2$.
Step 3: Update the iterative feedback gains K^j and L^j according to (22) and (23).
Step 4: Let $j = j + 1$.
Step 5: Stop if $\|K^j - K^{j+1}\| \leq \varepsilon_1$ and $\|L^j - L^{j+1}\| \leq \varepsilon_2$, where ε_i, $i = 1, 2$, are a predetermined error bound; otherwise, go to Step 2.

Referring to (11) and the quadratic expression of $V(x_k)$, we can derive

$$P_i^{j+1} = \begin{bmatrix} I \\ K^j \\ L^j \end{bmatrix}^T H_i^{j+1} \begin{bmatrix} I \\ K^j \\ L^j \end{bmatrix}, \ i = 1, 2.$$

By using (12), (17) yields

$$\begin{aligned} & Q_i^{j+1}(x_k, u_k, v_k) - [x_{k+1} - B(u_k - K^j x_k) - D \\ & \times (v_k - L^j x_k)]^T \begin{bmatrix} I \\ K^j \\ L^j \end{bmatrix}^T H_i^{j+1} \begin{bmatrix} I \\ K^j \\ L^j \end{bmatrix} [x_{k+1} - B \\ & \times (u_k - K^j x_k) - D(v_k - L^j x_k)] \\ & = x_k^T Q_i x_k + (u_k^j)^T R_{i1} u_k^j + (v_k^j)^T R_{i2} v_k^j, \ i = 1, 2. \end{aligned} \tag{18}$$

Combining (17) and (18), we have

$$\begin{aligned} & \begin{bmatrix} I \\ K^j \\ L^j \end{bmatrix}^T H_i^{j+1} \begin{bmatrix} I \\ K^j \\ L^j \end{bmatrix} - (A + BK^j + DL^j)^T \\ & \times \begin{bmatrix} I \\ K^j \\ L^j \end{bmatrix}^T H_i^{j+1} \begin{bmatrix} I \\ K^j \\ L^j \end{bmatrix} (A + BK^j + DL^j) \\ & = Q_i + (K^j)^T R_{i1} K^j + (L^j)^T R_{i2} L^j, \ i = 1, 2. \end{aligned} \tag{19}$$

Then, (18) can be formulated as follows

$$\begin{aligned} & x_k^T \begin{bmatrix} I \\ K^j \\ L^j \end{bmatrix}^T H_i^{j+1} \begin{bmatrix} I \\ K^j \\ L^j \end{bmatrix} x_k - x_{k+1}^T \begin{bmatrix} I \\ K^j \\ L^j \end{bmatrix}^T H_i^{j+1} \\ & \times \begin{bmatrix} I \\ K^j \\ L^j \end{bmatrix} x_{k+1} + 2x_k^T A^T P_i^{j+1} B(u_k - K^j x_k) \\ & + 2u_k^T B^T P_i^{j+1} B(u_k - K^j x_k) + 2x_k^T A^T P_i^{j+1} D(v_k \\ & - L^j x_k) + 2v_k^T D^T P_i^{j+1} D(v_k - L^j x_k) + 2u_k^T B^T \\ & \times P_i^{j+1} D(v_k - L^j x_k) - (v_k - L^j x_k)^T D^T P_i^{j+1} D(v_k \\ & - L^j x_k) - (u_k - K^j x_k)^T B^T P_i^{j+1} B(u_k - K^j x_k) \\ & - 2(u_k - K^j x_k)^T B^T P_i^{j+1} D(v_k - L^j x_k) \\ & + 2v_k^T D^T P_i^{j+1} B(u_k - K^j x_k) \\ & = x_k^T Q_i x_k + (u_k^j)^T R_{i1} u_k^j + (v_k^j)^T R_{i2} v_k^j, \ i = 1, 2. \end{aligned} \tag{20}$$

By employing Kronecker product and least squares operation, (20) is elaborated as follows

$$\theta^j X_i^{j+1} = \rho_i^j, \ i = 1, 2 \tag{21}$$

where

$$\begin{aligned} X_i^{j+1} &= [vec(X_{1i}^{j+1}) \ vec(X_{2i}^{j+1}) \ ... \ vec(X_{6i}^{j+1})] \\ X_{1i}^{j+1} &= Hxx_i^{j+1}, \ X_{2i}^{j+1} = Hxu_i^{j+1}, \ X_{3i}^{j+1} = Hxv_i^{j+1} \\ X_{4i}^{j+1} &= Hvu_i^{j+1}, \ X_{5i}^{j+1} = Huu_i^{j+1}, \ X_{6i}^{j+1} = Hvv_i^{j+1} \\ \rho_i^j &= x_k^T Q_i x_k + (u_k^j)^T R_{i1} u_k^j + (v_k^j)^T R_{i2} v_k^j \\ \theta^j &= [\theta_1^j \ \theta_2^j \ \theta_3^j \ \theta_4^j \ \theta_5^j \ \theta_6^j] \\ \theta_1^j &= x_k \otimes x_k - x_{k+1} \otimes x_{k+1} \\ \theta_2^j &= 2x_k \otimes u_k - 2x_{k+1} \otimes (K^j x_{k+1}) \\ \theta_3^j &= 2x_k \otimes v_k - 2x_{k+1} \otimes (L^j x_{k+1}) \\ \theta_4^j &= v_k \otimes u_k - (K^j x_{k+1}) \otimes (L^j x_{k+1}) \\ \theta_5^j &= u_k \otimes u_k - (K^j x_{k+1}) \otimes (K^j x_{k+1}) \\ \theta_6^j &= v_k \otimes v_k - (L^j x_{k+1}) \otimes (L^j x_{k+1}). \end{aligned}$$

In this way, the updated control gains can be obtained as

$$\begin{aligned} K^j = & (Huu_1 - Hvu_1(Hvv_2)^{-1}Hvu_2)^{-1} \\ & \times (Hvu_1(Hvv_2)^{-1}Hxv_2 - Hxu_1) \end{aligned} \tag{22}$$

$$\begin{aligned} L^j = & (Hvv_2 - Hvu_2(Huu_1)^{-1}Hvu_1)^{-1} \\ & \times (Hvu_2(Huu_1)^{-1}Hxu_1 - Hxv_2). \end{aligned} \tag{23}$$

Remark 3. Algorithm 2 is a model-free algorithm designed for the two-player non-zero-sum game. It is worth noting that no system parameters are required when updating the control gains using (22) and (23).

Theorem 2. The obtained $\widehat{H}_i^{j+1}$ matrix is unaffected and remains unchanged if nonzero probing noises are added to the control strategies in Algorithm 2.

Proof: After the probing noises e_k and w_k are added to the u_k and v_k, respectively, (17) takes the form:

$$\widehat{x}_{k+1} = \bar{A}\widehat{x}_k + B(u_k + e_k - K^j\widehat{x}_k) + D(v_k + w_k - L^j\widehat{x}_k) \tag{24}$$

where $\widehat{x}_k$ is the state after adding noises. Then, (19) can be transformed as

$$\begin{aligned} & \widehat{x}_k^T \begin{bmatrix} I \\ K^j \\ L^j \end{bmatrix}^T \widehat{H}_i^{j+1} \begin{bmatrix} I \\ K^j \\ L^j \end{bmatrix} \widehat{x}_k - [\widehat{x}_{k+1} - B(u_k + e_k \\ & - K^j\widehat{x}_k) - D(v_k + w_k - L^j\widehat{x}_k)]^T \begin{bmatrix} I \\ K^j \\ L^j \end{bmatrix}^T \widehat{H}_i^{j+1} \\ & \times \begin{bmatrix} I \\ K^j \\ L^j \end{bmatrix} [\widehat{x}_{k+1} - B(u_k + e_k - K^j\widehat{x}_k) \\ & - D(v_k + w_k - L^j\widehat{x}_k)] \\ & = \widehat{x}_k^T Q_i \widehat{x}_k + \widehat{x}_k^T (K^j)^T R_{i1} K^j \widehat{x}_k + \widehat{x}_k^T (L^j)^T R_{i2} L^j \widehat{x}_k. \end{aligned} \tag{25}$$

Substituting (24) into (25), results in

$$\begin{aligned} & \widehat{x}_k^T \begin{bmatrix} I \\ K^j \\ L^j \end{bmatrix}^T \widehat{H}_i^{j+1} \begin{bmatrix} I \\ K^j \\ L^j \end{bmatrix} \widehat{x}_k \\ & - \widehat{x}_k^T \bar{A} \begin{bmatrix} I \\ K^j \\ L^j \end{bmatrix}^T \widehat{H}_i^{j+1} \begin{bmatrix} I \\ K^j \\ L^j \end{bmatrix} \bar{A}\widehat{x}_k \\ & = \widehat{x}_k^T Q_i \widehat{x}_k + (u_k^j)^T R_{i1} u_k^j + (v_k^j)^T R_{i2} v_k^j. \end{aligned} \tag{26}$$

Since (26) holds for all state trajectories, we have

$$\begin{aligned}&\begin{bmatrix} I \\ K^j \\ L^j \end{bmatrix}^T \widehat{H}_i^{j+1} \begin{bmatrix} I \\ K^j \\ L^j \end{bmatrix} \\ &- \bar{A} \begin{bmatrix} I \\ K^j \\ L^j \end{bmatrix}^T \widehat{H}_i^{j+1} \begin{bmatrix} I \\ K^j \\ L^j \end{bmatrix} \bar{A} \\ &= Q_i + (u_k^j)^T R_{i1} u_k^j + (v_k^j)^T R_{i2} v_k^j.\end{aligned} \tag{27}$$

Note that (27) is equivalent to (19). Since the solution of (19) matches that of (18), adding probing noises during the implementation of the proposed off-policy Q-learning Algorithm 2 does not introduce bias. This completes the proof. ■

Remark 4. Compared to Algorithm 1, Algorithm 2 completely overcomes the defects caused by probing noise. In terms of noise selection, as outlined in [24], either sinusoidal or Gaussian noise can be considered as probing noise.

V. Simulation

This section provides a numerical example to verify the proposed algorithm. We consider two types of noise situations to illustrate the impact of different noises. The linear system (1) with the following system matrices is considered:

$$A = \begin{bmatrix} 0.9065 & 0.0816 & -0.0005 \\ 0.0743 & 0.9012 & -0.0007 \\ 0 & 0 & 0.1327 \end{bmatrix}$$
$$B = \begin{bmatrix} -0.0015 \\ -0.0096 \\ 0.8674 \end{bmatrix} \quad D = \begin{bmatrix} 0.0095 \\ 0.0004 \\ 0 \end{bmatrix}.$$

It is important to note that the system matrices are only used to simulate the system and acquire data, not for the control algorithms. The performance index is considered as (3) with $Q_1 = diag(1,1,1)$, $Q_2 = diag(1,1,1)$, $R_{11} = 1$, $R_{12} = 2$, $R_{21} = 2$, and $R_{22} = 1$. The initial state and the initial gains are chosen as $x_0 = [5 \ -10 \ 0]^T$, $K_0 = [-2 \ -2 \ -1]$, and $L_0 = [0 \ 0 \ 0]$. Now, we analyze the results of the proposed algorithms 1 and 2 under two cases of probing noises. The system dynamics are assumed to be completely unknown.

Inspired by the literature [25], the optimal control strategies are determined through a model-based algorithm. Hence, the gains K^* and L^* are given as

$$K^* = \begin{bmatrix} 0.0805 & 0.0925 & -0.0661 \end{bmatrix}$$
$$L^* = \begin{bmatrix} -0.1395 & -0.1166 & 0.0001 \end{bmatrix}.$$

Case 1. The probing noises are chosen as

$$e_k = sin(1.009k) + cos^2(0.538k) + sin(0.9k)$$
$$w_k = sin(2k) + cos(k)cos(2k) + sin(k) + cos(8k)$$

Fig. 1 depicts the convergence of the control gains obtained from Algorithm 1 during the learning process. It can be observed that the control gains are influenced by noise and do not approach the optimal values. Figs. 2 and 3 depict the evolution of the control gains and the system state trajectories, respectively, based on Algorithm 2. Fig. 2 shows that the control gains converge to the optimal values. The probing noise is removed after 400 time steps. Fig. 3 demonstrates that the state trajectories converge to zero as

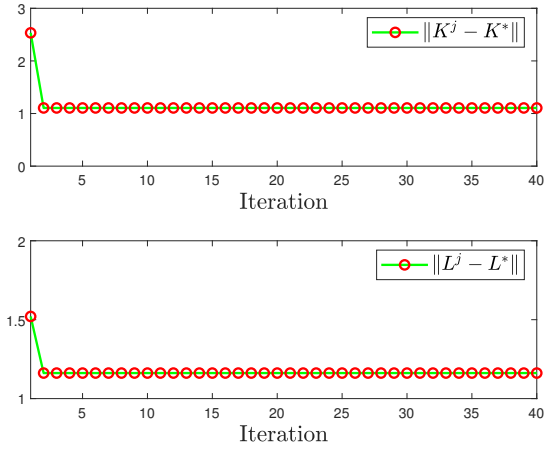

Fig. 1. Convergence of parameters K^j and L^j in on-policy Q-learning for Case 1.

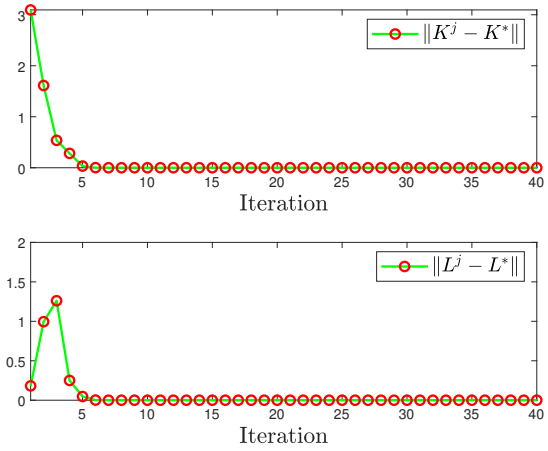

Fig. 2. Convergence of parameters K^j and L^j in off-policy Q-learning for Case 1.

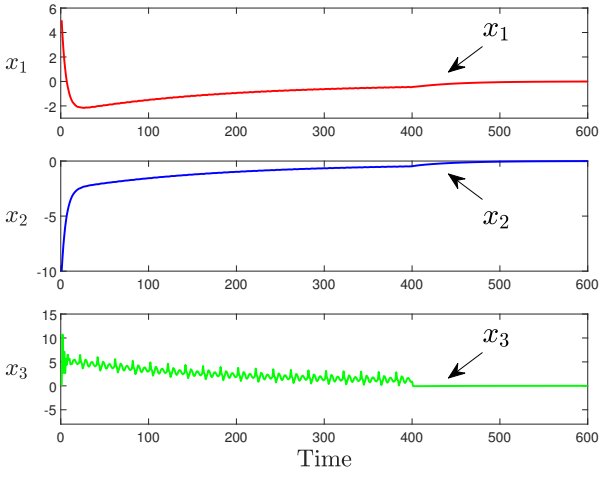

Fig. 3. System state trajectories in off-policy Q-learning for Case 1.

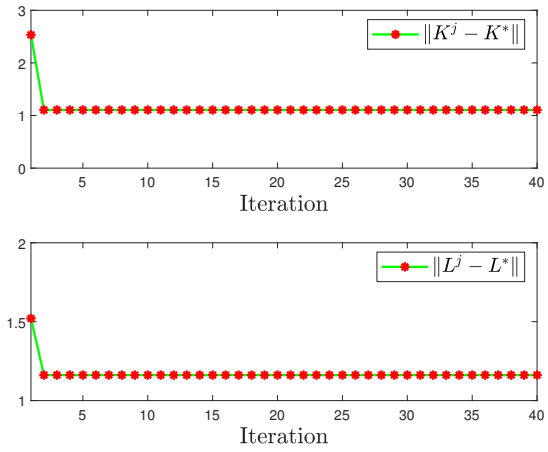

Fig. 4. Convergence of parameters K^j and L^j in on-policy Q-learning for Case 2.

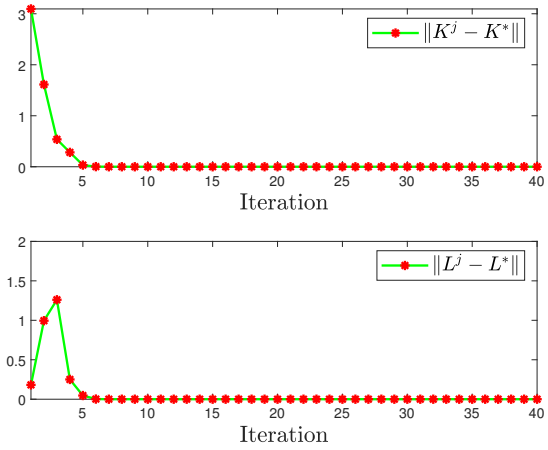

Fig. 5. Convergence of parameters K^j and L^j in off-policy Q-learning for Case 2.

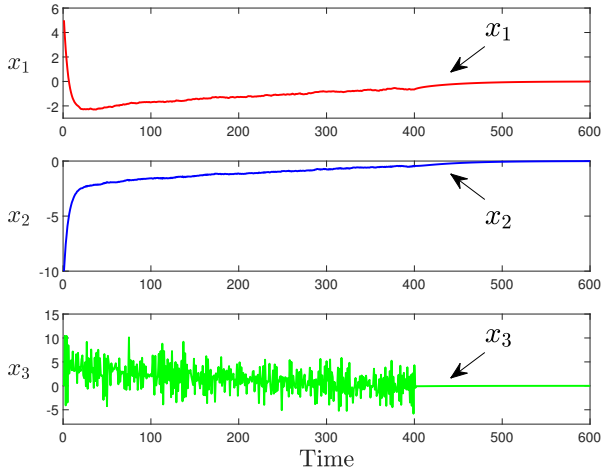

Fig. 6. System state trajectories in off-policy Q-learning for Case 2.

expected.

Case 2. The probing noises are chosen as

$$e_k = 2(randn(1) - 1)\ w_k = 2(randn(1) - 1)$$

where randn(·) is a MATLAB function that generates random numbers from a standard normal distribution, which has a mean of 0 and a standard deviation of 1.

Figs. 4, 5, and 6 show simulations of Case 2. Similar to Case 1, Fig. 5 shows that the control gains obtained from Algorithm 1 do not converge to the optimal solutions, while Figs. 5 and 6 illustrate the desired convergence of the control gains and state trajectories based on Algorithm 2.

VI. Conclusion

This paper considered the non-zero-sum game for linear discrete-time systems. Based on a quadratic value function, we derived coupled algebraic Riccati equations in (9). Then, we proposed both on-policy and off-policy Q-learning algorithms to achieve Nash equilibrium, which are model-free algorithms. To ensure the persistence of excitation, we introduced probing noise into the control input. We presented two theorems demonstrating that the on-policy Q-learning algorithm may introduce bias to the Nash equilibrium due to probing noise, whereas the off-policy Q-learning algorithm maintains an unbiased property. Finally, simulation results validated the effectiveness of the algorithms.

References

[1] M.-A. Messous, S.-M. Senouci, H. Sedjelmaci, and S. Cherkaoui, "A game theory based efficient computation offloading in an UAV network," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 5, pp. 4964–4974, 2019.

[2] P. Hang, C. Lv, Y. Xing, C. Huang, and Z. Hu, "Human-like decision making for autonomous driving: A noncooperative game theoretic approach," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 4, pp. 2076–2087, 2021.

[3] X. Wang, X. Zheng, W. Chen, and F.-Y. Wang, "Visual Human–computer interactions for intelligent vehicles and intelligent transportation systems: The state of the art and future directions," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 51, no. 1, pp. 253–265, 2021.

[4] Z. Ning, P. Dong, X. Wang, X. Hu, L. Guo, B. Hu, Y. Guo, T. Qiu, and R. Y. K. Kwok, "Mobile edge computing enabled 5G health monitoring for internet of medical things: A ecentralized game theoretic approach," *IEEE Journal on Selected Areas in Communications*, vol. 39, no. 2, pp. 463–478, 2021.

[5] C. Wu, X. Li, W. Pan, J. Liu, and L. Wu, "Zero-sum game-based optimal secure control under actuator attacks," *IEEE Transactions on Automatic Control*, vol. 66, no. 8, pp. 3773–3780, 2020.

[6] M. Wang and M. Wang, "Study on parameter correction of spring particle model based on generative adversarial network," *Engineering Letters*, vol. 29, no. 4, pp. 1494–1501, 2021.

[7] H. Ren, B. Jiang, and Y. Ma, "Zero-sum differential game-based fault-tolerant control for a class of affine nonlinear systems," *IEEE Transactions on Cybernetics*, vol. 54, no. 2, pp. 1272–1282, 2024.

[8] S. Wu, "Linear-quadratic non-zero sum backward stochastic differential game with overlapping information," *IEEE Transactions on Automatic Control*, vol. 68, no. 3, pp. 1800–1806, 2023.

[9] H. Su, H. Zhang, H. Jiang, and Y. Wen, "Decentralized event-triggered adaptive control of discrete-time nonzero-sum games

over wireless sensor-actuator networks with input constraints," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 31, no. 10, pp. 4254–4266, 2020.

[10] Y. Feng, "Game study on the evolution of subsidy strategies for on-site construction waste recycling management," *Engineering Letters*, vol. 31, no. 2, pp. 794–805, 2023.

[11] 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, 2012.

[12] T. Bian and Z.-P. Jiang, "Reinforcement learning and adaptive optimal control for continuous-time nonlinear systems: A value iteration approach," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 33, no. 7, pp. 2781–2790, 2021.

[13] Z. Xia, M.-J. Hu, W. Dai, H. Yan, and X. Ma, "Q-learning based multi-rate optimal control for process industries," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 70, no. 6, pp. 2006–2010, 2023.

[14] Y.-e. Hou, W. Gu, C. Wang, K. Yang, and Y. Wang, "A selection hyper-heuristic based on Q-learning for school bus routing problem," *IAENG International Journal of Applied Mathematics*, vol. 52, no. 4, pp. 817–825, 2022.

[15] S. Song, M. Zhu, X. Dai, and D. Gong, "Model-free optimal tracking control of nonlinear input-affine discrete-time systems via an iterative deterministic Q-learning algorithm," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 35, no. 1, pp. 999–1012, 2024.

[16] H. Xuan, J. Lu, N. Li, and L. Wang, "Novel virtual network function service chain deployment algorithm based on Q-learning," *IAENG International Journal of Computer Science*, vol. 50, no. 2, pp. 736–744, 2023.

[17] B. Luo, Y. Yang, and D. Liu, "Policy iteration Q-learning for data-based two-player zero-sum game of linear discrete-time systems," *IEEE Transactions on Cybernetics*, vol. 51, no. 7, pp. 3630–3640, 2020.

[18] X.-H. Chang, J. Wang, and X. Zhao, "Peak-to-peak filtering for discrete-time singular systems," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 68, no. 7, pp. 2543–2547, 2021.

[19] J. Zhou, J. Dong, S. Xu, and C. K. Ahn, "Input-to-state stabilization for Markov jump systems with dynamic quantization and multimode injection attacks," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 54, no. 4, pp. 2517–2529, 2024.

[20] T. Basar and H. Selbuz, "Properties of nash solutions of a two-stage nonzero-sum game," *IEEE Transactions on Automatic Control*, vol. 21, no. 1, pp. 48–54, 1976.

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

[22] F. L. Lewis, D. Vrabie, and K. G. Vamvoudakis, "Reinforcement learning and feedback control: Using natural decision methods to design optimal adaptive controllers," *IEEE Control Systems Magazine*, vol. 32, no. 6, pp. 76–105, 2012.

[23] L. Zhang, J. Fan, W. Xue, V. G. Lopez, J. Li, T. Chai, and F. L. Lewis, "Data-driven $\mathcal{H}_\infty$ optimal output feedback control for linear discrete-time systems based on off-policy Q-learning," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 34, no. 7, pp. 3553–3567, 2021.

[24] H. Xu, S. Jagannathan, and F. L. Lewis, "Stochastic optimal control of unknown linear networked control system in the presence of random delays and packet losses," *Automatica*, vol. 48, no. 6, pp. 1017–1030, 2012.

[25] X. Xin, Y. Tu, V. Stojanovic, H. Wang, K. Shi, S. He, and T. Pan, "Online reinforcement learning multiplayer non-zero sum games of continuous-time Markov jump linear systems," *Applied Mathematics and Computation*, vol. 412, pp. 126537, 2022.