Frontiers of Information Technology & Electronic Engineering
www.jzus.zju.edu.cn; engineering.cae.cn; www.springerlink.com
ISSN 2095-9184 (print); ISSN 2095-9230 (online)
E-mail: jzus@zju.edu.cn

Minimax Q-learning design for H_∞ control of linear discrete-time systems*

Xinxing LI[§1], Lele XI[§2,3], Wenzhong ZHA[‡1], Zhihong PENG[2]

[1]*Information Science Academy, China Electronics Technology Group Corporation, Beijing 100086, China*

[2]*School of Automation, Beijing Institute of Technology, Beijing 100081, China*

[3]*Peng Cheng Laboratory, Shenzhen 518052, China*

E-mail: lixinxing_1006@163.com; xilele.bit@gmail.com; zhawenzhong@126.com; peng@bit.edu.cn

Received Aug. 31, 2020; Revision accepted Jan. 10, 2021; Crosschecked Oct. 21, 2021; Published online Feb. 4, 2022

Abstract: The H_∞ control method is an effective approach for attenuating the effect of disturbances on practical systems, but it is difficult to obtain the H_∞ controller due to the nonlinear Hamilton–Jacobi–Isaacs equation, even for linear systems. This study deals with the design of an H_∞ controller for linear discrete-time systems. To solve the related game algebraic Riccati equation (GARE), a novel model-free minimax Q-learning method is developed, on the basis of an offline policy iteration algorithm, which is shown to be Newton's method for solving the GARE. The proposed minimax Q-learning method, which employs off-policy reinforcement learning, learns the optimal control policies for the controller and the disturbance online, using only the state samples generated by the implemented behavior policies. Different from existing Q-learning methods, a novel gradient-based policy improvement scheme is proposed. We prove that the minimax Q-learning method converges to the saddle solution under initially admissible control policies and an appropriate positive learning rate, provided that certain persistence of excitation (PE) conditions are satisfied. In addition, the PE conditions can be easily met by choosing appropriate behavior policies containing certain excitation noises, without causing any excitation noise bias. In the simulation study, we apply the proposed minimax Q-learning method to design an H_∞ load-frequency controller for an electrical power system generator that suffers from load disturbance, and the simulation results indicate that the obtained H_∞ load-frequency controller has good disturbance rejection performance.

Key words: H_∞ control; Zero-sum dynamic game; Reinforcement learning; Adaptive dynamic programming; Minimax Q-learning; Policy iteration

https://doi.org/10.1631/FITEE.2000446 **CLC number:** TP13

1 Introduction

Reinforcement learning (RL) is an efficient machine learning technique for dealing with sequential decision-making problems when an agent interacts with an external environment, such as Markov decision processes (Sutton and Barto, 1998). The core mechanism of RL is that an agent unceasingly modifies its action, based on the observed stimuli or reward received from the environment, via trial-and-error. Compared with the traditional dynamic programming (DP) technique for handling sequential decision-making problems, RL runs forward in time (i.e., online) and overcomes the curse-of-dimensionality problem, and can find the optimal policy even in a dynamic environment, e.g., dynamic games. It has been shown that RL combines the advantages of optimal and adaptive control

§ These two authors contributed equally to this work

‡ Corresponding author

* Project supported by the National Natural Science Foundation of China (No. U1613225)

ORCID: Xinxing LI, https://orcid.org/0000-0001-6264-2955; Lele XI, https://orcid.org/0000-0002-8024-3349; Wenzhong ZHA, https://orcid.org/0000-0003-2718-5052

(Kiumarsi et al., 2018), which makes it a promising technique for solving optimal control problems and dynamic games. In the control field, RL is also referred to as adaptive dynamic programming (ADP). ADP approaches can be classified into several main schemes: heuristic dynamic programming (HDP), action-dependent HDP (ADHDP), dual heuristic dynamic programming (DHP), ADDHP, globalized DHP (GDHP), and ADGDHP (Prokhorov and Wunsch, 1997). During the last few years, many elegant ADP approaches have been proposed to solve optimal control problems (He and Zhong, 2018; Li HR et al., 2020) and dynamic games (Vamvoudakis et al., 2017; Zhu et al., 2017; Li XX et al., 2019; Valadbeigi et al., 2020).

Due to the uncertainty caused by the environment, most practical systems always suffer from external disturbances. To attenuate the effect of disturbances on the system performance, controllers that can offer robust performance and guarantee stabilization are needed. One of the most effective approaches is the H_∞ control theory, which concentrates on designing controllers to achieve disturbance attenuation in the L_2-gain setting (Doyle et al., 1989; Başar and Bernhard, 1995). It is well known that obtaining an H_∞ controller requires solving the nonlinear Hamilton–Jacobi–Isaacs (HJI) equation. However, obtaining the analytic solution of the HJI equation is impossible. Thus, an approximate solution is always obtained instead (Sakamoto and van der Schaft, 2008). Over the past few years, many ADP methods have been developed to solve continuous-time HJI equations. Luo et al. (2015) proposed a model-free policy iteration (PI) algorithm with one iteration loop for designing an H_∞ controller for nonlinear continuous-time systems, by employing off-policy RL. Modares et al. (2015) developed an online off-policy ADP algorithm for the H_∞ tracking control of continuous-time systems, to name a few.

Compared with H_∞ control of continuous-time systems, H_∞ control of discrete-time systems is more challenging, because the discrete-time HJI equations do not have a closed-loop form (Başar and Bernhard, 1995). To solve discrete-time HJI equations, Mehraeen et al. (2013) proposed an offline PI algorithm with two iteration loops by using Taylor series. To obviate the need for knowledge of the system, Zhang et al. (2014) proposed an online PI algorithm by introducing a neural network (NN) identification scheme. Further, a completely model-free GDHP approach was presented without the need for the NN identifier (Zhong et al., 2018). By employing off-policy RL, Kiumarsi et al. (2017) proposed a model-free ADP method that can learn the H_∞ controller for linear discrete-time systems online. Q-learning serves as another powerful tool for handling discrete-time H_∞ control problems. The first Q-learning method with guaranteed convergence was proposed by Watkins and Dayan (1992) to solve Markov decision processes by employing the temporal-difference (TD) learning technique. Then, minimax Q-learning and Nash-Q learning were developed for zero-sum and nonzero-sum stochastic games with finite state and action spaces, respectively (Littman, 2001). Over the past few years, many efficient Q-learning approaches have been developed for optimal control (Wei QL et al., 2017; Luo et al., 2018; Wei YF et al., 2019; Yan et al., 2019) and H_∞ control (Al-Tamimi et al., 2007; Rizvi and Lin, 2018; Valadbeigi et al., 2020). In Al-Tamimi et al. (2007), a value-iteration-based Q-learning algorithm with convergence guarantee was presented to solve the discrete-time zero-sum game problem, but this algorithm suffers from the excitation noise bias problem, because the injected probing noises in the policy evaluation step will cause excitation noise bias (Kiumarsi et al., 2017). On the basis of this state feedback Q-learning, output feedback Q-learning methods that overcome the excitation noise bias problem have been proposed (Rizvi and Lin, 2018; Valadbeigi et al., 2020). Generally speaking, most of the existing Q-learning methods for H_∞ control of linear discrete-time systems are based on value iteration. Meanwhile, theoretical foundations for policy-iteration-based Q-learning are relatively lacking in the literature. Although the convergence analyses of minimax Q-learning and policy iteration for stochastic games were given in Littman (2001) and Hansen et al. (2003), respectively, these results do not hold for H_∞ control of discrete-time systems with continuous state and action spaces.

Inspired by off-policy RL and adaptive control, we develop a novel policy-iteration-based minimax Q-learning method for H_∞ control of linear discrete-time systems, with guaranteed convergence. The main contributions of this study are summarized as follows:

1. The proposed policy-iteration-based minimax

Q-learning method, which employs an off-policy RL technique, learns the H_∞ controller online using only the state samples generated by the behavior polices, without querying the system model or causing any excitation noise bias.

2. Different from existing Q-learning methods (Al-Tamimi et al., 2007; Rizvi and Lin, 2018; Valadbeigi et al., 2020; Luo et al., 2021), we develop a novel policy improvement scheme by borrowing the idea of a stochastic gradient algorithm. The newly improved control policies can be obtained via online learning without the need for calculating the inverse of the value matrix after performing policy evaluation. Moreover, this policy improvement scheme applies to H_∞ control of nonlinear discrete-time systems.

3. Unlike TD-based minimax Q-learning for stochastic games (Littman, 2001), our minimax Q-learning method is based on policy iteration. In addition, we give a rigorous convergence analysis of offline policy iteration for H_∞ control of linear discrete-time systems by proving its equivalence to Newton's method for solving the game algebraic Riccati equation (GARE), and on this basis, we prove that the proposed policy-iteration-based minimax Q-learning method converges to the exact saddle solution under an appropriate learning rate and certain persistence of excitation (PE) conditions.

Notations: $\mathbb{R}^n$ denotes the n-dimensional Euclidean space. $\mathbb{R}^{n\times m}$ is the set of real $n\times m$ matrices. $\otimes$ stands for the Kronecker product. $\mathrm{vec}(\cdot)$ is the vectorization operator that stacks each column of a matrix into a one-column vector. For vector $\boldsymbol{x}\in\mathbb{R}^n$, the Kronecker product quadratic polynomial basis vector of $\boldsymbol{x}$ is defined as $\boldsymbol{\sigma}(\boldsymbol{x})=[x_1^2,\dots,x_1x_n,x_2^2,\dots,x_2x_n,\dots,x_{n-1}x_n,x_n^2]^{\mathrm{T}}$. The Frobenius norm for matrix $\boldsymbol{A}\in\mathbb{R}^{n\times m}$ is defined as $\|\boldsymbol{A}\|=(\mathrm{tr}(\boldsymbol{A}^{\mathrm{T}}\boldsymbol{A}))^{1/2}$, where $\mathrm{tr}(\cdot)$ represents the trace of a matrix. For a real symmetric matrix $\boldsymbol{E}\in\mathbb{R}^{n\times n}$, $\lambda_{\min}(\boldsymbol{E})$, $\lambda_{\max}(\boldsymbol{E})$, and $\rho(\boldsymbol{E})$ denote the minimum eigenvalue, maximum eigenvalue, and spectral radius of $\boldsymbol{E}$, respectively.

2 Problem statement

In this section, we give the formulation of the worst-case controller design problem, that is, the H_∞ optimal control of linear discrete-time systems. Consider the following linear discrete-time system with two types of inputs:

$$\boldsymbol{x}_{k+1}=\boldsymbol{A}\boldsymbol{x}_k+\boldsymbol{B}\boldsymbol{u}_k+\boldsymbol{D}\boldsymbol{w}_k, \tag{1}$$

where $\boldsymbol{x}_k\in\mathbb{R}^n$ is the system state vector, $\boldsymbol{u}_k\in\mathbb{R}^{m_1}$ is the control input vector, and $\boldsymbol{w}_k\in\mathbb{R}^{m_2}$ is an external disturbance input vector belonging to the square-summable space $L_2(0,\infty)$, i.e., $\sum_{k=0}^{\infty}\boldsymbol{w}_k^{\mathrm{T}}\boldsymbol{w}_k<\infty$ (thus, $\boldsymbol{w}_k$ has finite energy). The plant matrix $\boldsymbol{A}\in\mathbb{R}^{n\times n}$ and the input matrices $\boldsymbol{B}\in\mathbb{R}^{n\times m_1}$ and $\boldsymbol{D}\in\mathbb{R}^{n\times m_2}$ are assumed to be unknown.

The aim of H_∞ control is to find the optimal control policy $\boldsymbol{u}^*$ such that system (1) is asymptotically stable with $\boldsymbol{w}_k=\boldsymbol{0}$ and the following disturbance attenuation condition

$$\sum_{k=0}^{\infty}\left(\boldsymbol{x}_k^{\mathrm{T}}\boldsymbol{S}\boldsymbol{x}_k+\boldsymbol{u}_k^{\mathrm{T}}\boldsymbol{R}\boldsymbol{u}_k\right)\le\gamma^2\sum_{k=0}^{\infty}\boldsymbol{w}_k^{\mathrm{T}}\boldsymbol{w}_k$$

is satisfied, where $\boldsymbol{S}$ and $\boldsymbol{R}$ are user-defined positive definite matrices, and $\gamma>0$ is a prescribed constant disturbance attenuation level. To make sure that the problem is solvable, we make controllability and observability assumptions on $(\boldsymbol{A},\boldsymbol{B})$ and $(\boldsymbol{A},\sqrt{\boldsymbol{S}})$, respectively.

According to Başar and Bernhard (1995), the H_∞ optimal control problem can be equivalently translated into a two-player zero-sum linear quadratic dynamic game, i.e., the following minimax optimization problem:

$$\begin{aligned}&(\boldsymbol{u}_k^*,\boldsymbol{w}_k^*)\\&=\min_{\boldsymbol{u}_k}\max_{\boldsymbol{w}_k}\sum_{i=k}^{\infty}\left(\boldsymbol{x}_i^{\mathrm{T}}\boldsymbol{S}\boldsymbol{x}_i+\boldsymbol{u}_i^{\mathrm{T}}\boldsymbol{R}\boldsymbol{u}_i-\gamma^2\boldsymbol{w}_i^{\mathrm{T}}\boldsymbol{w}_i\right)\\&=\max_{\boldsymbol{w}_k}\min_{\boldsymbol{u}_k}\sum_{i=k}^{\infty}\left(\boldsymbol{x}_i^{\mathrm{T}}\boldsymbol{S}\boldsymbol{x}_i+\boldsymbol{u}_i^{\mathrm{T}}\boldsymbol{R}\boldsymbol{u}_i-\gamma^2\boldsymbol{w}_i^{\mathrm{T}}\boldsymbol{w}_i\right)\end{aligned} \tag{2}$$

subject to

$$\boldsymbol{x}_{k+1}=\boldsymbol{A}\boldsymbol{x}_k+\boldsymbol{B}\boldsymbol{u}_k^*+\boldsymbol{D}\boldsymbol{w}_k^*. \tag{3}$$

$V(\boldsymbol{x}_k)=\sum_{i=k}^{\infty}\left(\boldsymbol{x}_i^{\mathrm{T}}\boldsymbol{S}\boldsymbol{x}_i+\boldsymbol{u}_i^{\mathrm{T}}\boldsymbol{R}\boldsymbol{u}_i-\gamma^2\boldsymbol{w}_i^{\mathrm{T}}\boldsymbol{w}_i\right)$ is the value function corresponding to admissible control policies $\boldsymbol{u}$ and $\boldsymbol{w}$. The goal of the zero-sum dynamic game is to find the feedback saddle solution $(\boldsymbol{u}^*,\boldsymbol{w}^*)$ such that the following inequality

$$V(\boldsymbol{x}_k,\boldsymbol{u}_k^*,\boldsymbol{w}_k)\le V^*(\boldsymbol{x}_k,\boldsymbol{u}_k^*,\boldsymbol{w}_k^*)\le V(\boldsymbol{x}_k,\boldsymbol{u}_k,\boldsymbol{w}_k^*),\forall k \tag{4}$$

is satisfied for arbitrary admissible control policies $\boldsymbol{u}$ and $\boldsymbol{w}$. From inequality (4), we know that no

player will deviate from $(\boldsymbol{u}^*, \boldsymbol{w}^*)$, because a unilateral change of strategy will cause a loss of revenue for both players.

According to Bellman's principle of optimality, the feedback saddle solution $(\boldsymbol{u}^*, \boldsymbol{w}^*)$ should satisfy the following Bellman optimality equation:

$$V^*(\boldsymbol{x}_k)=\boldsymbol{x}_k^{\mathrm{T}}\boldsymbol{S}\boldsymbol{x}_k+(\boldsymbol{u}_k^*)^{\mathrm{T}}\boldsymbol{R}\boldsymbol{u}_k^*-\gamma^2(\boldsymbol{w}_k^*)^{\mathrm{T}}\boldsymbol{w}_k^*+V^*(\boldsymbol{x}_{k+1}), \tag{5}$$

where $\boldsymbol{x}_{k+1} = \boldsymbol{A}\boldsymbol{x}_k + \boldsymbol{B}\boldsymbol{u}_k^* + \boldsymbol{D}\boldsymbol{w}_k^*$. From Başar and Bernhard (1995), we can represent the value function $V^*(\boldsymbol{x}_k)$ as a quadratic form of the state, i.e., $V^*(\boldsymbol{x}_k) = \boldsymbol{x}_k^{\mathrm{T}}\boldsymbol{P}\boldsymbol{x}_k$, where $\boldsymbol{P}$ is the positive semi-definite value matrix. Substituting $V^*(\boldsymbol{x}_k) = \boldsymbol{x}_k^{\mathrm{T}}\boldsymbol{P}\boldsymbol{x}_k$ into Eq. (5) gives the feedback gains corresponding to the saddle solution:

$$\begin{aligned}\boldsymbol{K}_1^* =& \\ &\left(\boldsymbol{B}^{\mathrm{T}}\boldsymbol{P}\boldsymbol{D}(\gamma^2\boldsymbol{I}-\boldsymbol{D}^{\mathrm{T}}\boldsymbol{P}\boldsymbol{D})^{-1}\boldsymbol{D}^{\mathrm{T}}+\boldsymbol{P}\boldsymbol{B}\boldsymbol{R}+\boldsymbol{B}^{\mathrm{T}}\boldsymbol{P}\boldsymbol{B}\right)^{-1}\\ &\times\left(\boldsymbol{B}^{\mathrm{T}}\boldsymbol{P}\boldsymbol{A}+\boldsymbol{B}^{\mathrm{T}}\boldsymbol{P}\boldsymbol{D}(\gamma^2\boldsymbol{I}-\boldsymbol{D}^{\mathrm{T}}\boldsymbol{P}\boldsymbol{D})^{-1}\boldsymbol{D}^{\mathrm{T}}\boldsymbol{P}\boldsymbol{A}\right),\end{aligned} \tag{6}$$

$$\begin{aligned}\boldsymbol{K}_2^* =& \\ &\left(-\boldsymbol{D}^{\mathrm{T}}\boldsymbol{P}\boldsymbol{B}(\boldsymbol{R}+\boldsymbol{B}^{\mathrm{T}}\boldsymbol{P}\boldsymbol{B})^{-1}\boldsymbol{B}^{\mathrm{T}}\boldsymbol{P}\boldsymbol{D}+\boldsymbol{D}^{\mathrm{T}}\boldsymbol{P}\boldsymbol{D}-\gamma^2\boldsymbol{I}\right)^{-1}\\ &\times\left(\boldsymbol{D}^{\mathrm{T}}\boldsymbol{P}\boldsymbol{A}-\boldsymbol{D}^{\mathrm{T}}\boldsymbol{P}\boldsymbol{B}(\boldsymbol{R}+\boldsymbol{B}^{\mathrm{T}}\boldsymbol{P}\boldsymbol{B})^{-1}\boldsymbol{B}^{\mathrm{T}}\boldsymbol{P}\boldsymbol{A}\right);\end{aligned} \tag{7}$$

hence, $\boldsymbol{u}_k^* = -\boldsymbol{K}_1^*\boldsymbol{x}_k$ and $\boldsymbol{w}_k^* = -\boldsymbol{K}_2^*\boldsymbol{x}_k$. Substituting $\boldsymbol{u}_k^*$ and $\boldsymbol{w}_k^*$ into Eq. (5) then yields the compact form of the GARE

$$\begin{aligned}\boldsymbol{P} =& \boldsymbol{A}^{\mathrm{T}}\boldsymbol{P}\boldsymbol{A}+\boldsymbol{S}-\begin{bmatrix}\boldsymbol{A}^{\mathrm{T}}\boldsymbol{P}\boldsymbol{B} & \boldsymbol{A}^{\mathrm{T}}\boldsymbol{P}\boldsymbol{D}\end{bmatrix}\\ &\times\begin{bmatrix}\boldsymbol{R}+\boldsymbol{B}^{\mathrm{T}}\boldsymbol{P}\boldsymbol{B} & \boldsymbol{B}^{\mathrm{T}}\boldsymbol{P}\boldsymbol{D}\\ \boldsymbol{D}^{\mathrm{T}}\boldsymbol{P}\boldsymbol{B} & \boldsymbol{D}^{\mathrm{T}}\boldsymbol{P}\boldsymbol{D}-\gamma^2\boldsymbol{I}\end{bmatrix}^{-1}\begin{bmatrix}\boldsymbol{B}^{\mathrm{T}}\boldsymbol{P}\boldsymbol{A}\\ \boldsymbol{D}^{\mathrm{T}}\boldsymbol{P}\boldsymbol{A}\end{bmatrix}.\end{aligned} \tag{8}$$

To guarantee a unique feedback saddle solution, the following inequalities

$$\boldsymbol{I}-\gamma^{-2}\boldsymbol{D}^{\mathrm{T}}\boldsymbol{P}\boldsymbol{D}>0, \tag{9}$$

$$\boldsymbol{I}+\boldsymbol{B}^{\mathrm{T}}\boldsymbol{P}\boldsymbol{B}>0 \tag{10}$$

should be satisfied (Başar and Bernhard, 1995). Furthermore, the disturbance attenuation level γ should be selected such that $\gamma \geq \gamma^* > 0$ is satisfied, where $\gamma^* > 0$ is the infimum of γ.

From Eqs. (6)–(8), we know that obtaining $\boldsymbol{K}_1^*$ and $\boldsymbol{K}_2^*$ requires solving the GARE, which is a nonlinear matrix equation. Moreover, Eqs. (6)–(8) are dependent on full knowledge of $\boldsymbol{A}$, $\boldsymbol{B}$, and $\boldsymbol{D}$, which are assumed to be unknown in this study. In the following sections, we will develop a minimax Q-learning algorithm to learn online $\boldsymbol{K}_1^*$ and $\boldsymbol{K}_2^*$ without querying the system models $\boldsymbol{A}$, $\boldsymbol{B}$, and $\boldsymbol{D}$.

3 Offline policy iteration for zero-sum linear quadratic dynamic games

Before deriving the online minimax Q-learning algorithm, we first introduce the model-based offline PI algorithm deduced from Algorithm 1 in Kiumarsi et al. (2017). The offline PI algorithm lays the foundation for the following minimax Q-learning algorithm. The offline PI algorithm, employing a successive approximation technique, indirectly solves the nonlinear GARE (8) by constructing a sequence of linear matrix equations. The detailed algorithm is given in Algorithm 1.

Algorithm 1 Model-based offline policy iteration algorithm

1: Start with a set of initially stabilizing feedback gains $(\boldsymbol{K}_1^1, \boldsymbol{K}_2^1)$ // Initialization

2: For the given stabilizing feedback gains $(\boldsymbol{K}_1^l, \boldsymbol{K}_2^l)$, solve for the corresponding value matrix $\boldsymbol{P}^{l+1}$ via the following matrix equation: // Policy evaluation

$$\begin{aligned}\boldsymbol{P}^{l+1} =& \boldsymbol{S}+(\boldsymbol{K}_1^l)^{\mathrm{T}}\boldsymbol{R}\boldsymbol{K}_1^l-\gamma^2(\boldsymbol{K}_2^l)^{\mathrm{T}}\boldsymbol{K}_2^l+(\boldsymbol{A}-\boldsymbol{B}\boldsymbol{K}_1^l\\ &-\boldsymbol{D}\boldsymbol{K}_2^l)^{\mathrm{T}}\boldsymbol{P}^{l+1}(\boldsymbol{A}-\boldsymbol{B}\boldsymbol{K}_1^l-\boldsymbol{D}\boldsymbol{K}_2^l)\end{aligned} \tag{11}$$

3: Update the control policy and disturbance policy using the following equation: // Policy improvement

$$\begin{pmatrix}\boldsymbol{K}_1^{l+1}\\ \boldsymbol{K}_2^{l+1}\end{pmatrix}=\zeta(\boldsymbol{P}^{l+1})\begin{bmatrix}\boldsymbol{B}^{\mathrm{T}}\boldsymbol{P}^{l+1}\boldsymbol{A}\\ \boldsymbol{D}^{\mathrm{T}}\boldsymbol{P}^{l+1}\boldsymbol{A}\end{bmatrix}, \tag{12}$$

where $\zeta(\boldsymbol{P}^{l+1})$ is defined as follows:

$$\zeta(\boldsymbol{P}^{l+1})=\begin{bmatrix}\boldsymbol{R}+\boldsymbol{B}^{\mathrm{T}}\boldsymbol{P}^{l+1}\boldsymbol{B} & \boldsymbol{B}^{\mathrm{T}}\boldsymbol{P}^{l+1}\boldsymbol{D}\\ \boldsymbol{D}^{\mathrm{T}}\boldsymbol{P}^{l+1}\boldsymbol{B} & \boldsymbol{D}^{\mathrm{T}}\boldsymbol{P}^{l+1}\boldsymbol{D}-\gamma^2\boldsymbol{I}\end{bmatrix}^{-1}$$

4: Stop if $\left\|\boldsymbol{K}_i^{l+1}-\boldsymbol{K}_i^l\right\| \leq \varepsilon$ $(i=1,2)$, where ε is a threshold; otherwise, set $l=l+1$ and go to step 2

The policy evaluation step (Eq. (11)) is used to evaluate the performance of the given control policy $\boldsymbol{u}^l = -\boldsymbol{K}_1^l\boldsymbol{x}$ and the disturbance policy $\boldsymbol{w}^l = -\boldsymbol{K}_2^l\boldsymbol{x}$. After policy evaluation, a new control policy and the disturbance policy are obtained via the certainty equivalence principle; that is, the obtained value matrix $\boldsymbol{P}^{l+1}$ is regarded as the optimal value matrix $\boldsymbol{P}$.

In the following theorem (Theorem 1), we will prove that $(\boldsymbol{K}_1^l, \boldsymbol{K}_2^l)$ will converge to $(\boldsymbol{K}_1^*, \boldsymbol{K}_2^*)$ under any initially stabilizing feedback gains $(\boldsymbol{K}_1^1, \boldsymbol{K}_2^1)$.

Before giving the convergence proof, we define two useful mappings. Consider the space $\mathbb{R}^{n\times n}$ composed of all $n\times n$ dimensional real matrices. We can easily verify that $\mathbb{R}^{n\times n}$ forms a Banach space under the Frobenius norm. We now define a mapping $\mathcal{F}$: $\mathbb{R}^{n\times n}\to\mathbb{R}^{n\times n}$:

$$\mathcal{F}(\boldsymbol{P}^l)=\boldsymbol{A}^{\mathrm{T}}\boldsymbol{P}^l\boldsymbol{A}+\boldsymbol{S}-\boldsymbol{P}^l-\Theta(\boldsymbol{P}^l)\zeta(\boldsymbol{P}^l)(\Theta(\boldsymbol{P}^l))^{\mathrm{T}}, \tag{13}$$

where $\Theta(\boldsymbol{P}^l)=[\boldsymbol{A}^{\mathrm{T}}\boldsymbol{P}^l\boldsymbol{B}\quad \boldsymbol{A}^{\mathrm{T}}\boldsymbol{P}^l\boldsymbol{D}]$ and

$$\zeta(\boldsymbol{P}^l)=\begin{bmatrix}\boldsymbol{R}+\boldsymbol{B}^{\mathrm{T}}\boldsymbol{P}^l\boldsymbol{B} & \boldsymbol{B}^{\mathrm{T}}\boldsymbol{P}^l\boldsymbol{D}\\ \boldsymbol{D}^{\mathrm{T}}\boldsymbol{P}^l\boldsymbol{B} & \boldsymbol{D}^{\mathrm{T}}\boldsymbol{P}^l\boldsymbol{D}-\gamma^2\boldsymbol{I}\end{bmatrix}^{-1}.$$

From the definition of $\mathcal{F}$, we know that $\boldsymbol{P}$ is the zero-point of mapping $\mathcal{F}$. Now we define a new mapping based on $\mathcal{F}$. The new mapping $\mathcal{T}$: $\mathbb{R}^{n\times n}\to\mathbb{R}^{n\times n}$ is given as follows:

$$\mathcal{T}(\boldsymbol{P}^l)=\boldsymbol{P}^l-(\mathcal{F}'_{\boldsymbol{P}^l})^{-1}(\boldsymbol{P}^l), \tag{14}$$

where $\mathcal{F}'_{\boldsymbol{P}^l}$ is the Fréchet derivative of $\mathcal{F}$ taken with respect to $\boldsymbol{P}^l$. Clearly, Eq. (14) is exactly Newton's method for obtaining the zero-point of $\mathcal{F}$, or equivalently the fixed-point of Eq. (8). Directly calculating the Fréchet derivative is always impossible, so we calculate the Gâteaux derivative instead.

Definition 1 (Gâteaux derivative) Let Ξ: $\mathbb{U}(\boldsymbol{V})\in\mathbb{X}\to\mathbb{Y}$ be a mapping from Banach space $\mathbb{X}$ to Banach space $\mathbb{Y}$, where $\mathbb{U}(\boldsymbol{V})$ is a neighborhood of $\boldsymbol{V}$. The mapping Ξ is Gâteaux differentiable at $\boldsymbol{V}$ if and only if there exists a bounded linear operator $\mathcal{G}:\mathbb{X}\to\mathbb{Y}$ such that $\Xi(\boldsymbol{V}+s\boldsymbol{W})-\Xi(\boldsymbol{V})=s\mathcal{G}(\boldsymbol{W})+o(s), s\to 0$ for all $\boldsymbol{W}$ with $\|\boldsymbol{W}\|=1$ and all real numbers s in some neighborhood of zero, where $\lim_{s\to 0}o(s)/s=0$. The linear operator $\mathcal{G}$ is called the Gâteaux derivative of Ξ at $\boldsymbol{V}$; thus, $\mathcal{G}$ is calculated as

$$\mathcal{G}(\boldsymbol{W})=\lim_{s\to 0}\frac{\Xi(\boldsymbol{V}+s\boldsymbol{W})-\Xi(\boldsymbol{V})}{s}. \tag{15}$$

Note that the Fréchet derivative at $\boldsymbol{V}$ equals the Gâteaux derivative $\mathcal{G}$, if the Gâteaux derivative $\mathcal{G}$ exists in some neighborhood of $\boldsymbol{V}$ and $\mathcal{G}$ is continuous at $\boldsymbol{V}$. Now we turn to calculating the Fréchet derivative of $\mathcal{F}$ at $\boldsymbol{P}^l$ according to the following lemma:

Lemma 1 Let $\mathcal{F}$ be a mapping defined in Eq. (13). Then the Fréchet derivative of $\mathcal{F}$ at $\boldsymbol{P}^l$ is given by

$$\begin{aligned}&\mathcal{F}'_{\boldsymbol{P}^l}(\boldsymbol{M})\\ &=\boldsymbol{A}^{\mathrm{T}}\boldsymbol{M}\boldsymbol{A}-\boldsymbol{M}-\Theta(\boldsymbol{M})\zeta(\boldsymbol{P}^l)(\Theta(\boldsymbol{P}^l))^{\mathrm{T}}-\Theta(\boldsymbol{P}^l)\\ &\times\zeta(\boldsymbol{P}^l)(\Theta(\boldsymbol{M}))^{\mathrm{T}}+\Theta(\boldsymbol{P}^l)\zeta(\boldsymbol{P}^l)\vartheta(\boldsymbol{M})\zeta(\boldsymbol{P}^l)(\Theta(\boldsymbol{P}^l))^{\mathrm{T}},\end{aligned} \tag{16}$$

where $\Theta(\boldsymbol{M})=[\boldsymbol{A}^{\mathrm{T}}\boldsymbol{M}^{\mathrm{T}}\boldsymbol{B}\quad \boldsymbol{A}^{\mathrm{T}}\boldsymbol{M}^{\mathrm{T}}\boldsymbol{D}]$ and

$$\vartheta(\boldsymbol{M})=\begin{bmatrix}\boldsymbol{B}^{\mathrm{T}}\boldsymbol{M}\boldsymbol{B} & \boldsymbol{B}^{\mathrm{T}}\boldsymbol{M}\boldsymbol{D}\\ \boldsymbol{D}^{\mathrm{T}}\boldsymbol{M}\boldsymbol{B} & \boldsymbol{D}^{\mathrm{T}}\boldsymbol{M}\boldsymbol{D}\end{bmatrix}.$$

Proof First, we calculate the Gâteaux derivative $\mathcal{G}$ at $\boldsymbol{P}^l$. Note that $(\boldsymbol{I}+\boldsymbol{X})^{-1}=\boldsymbol{I}-\boldsymbol{X}+\boldsymbol{X}^2-\boldsymbol{X}^3+\cdots$ holds for any $\boldsymbol{X}\in\mathbb{R}^{n\times n}$ if $\rho(\boldsymbol{X})<1$ is satisfied. Select s such that $s<\rho^{-1}(\zeta(\boldsymbol{P}^l)\vartheta(\boldsymbol{M}))$ is met. We then obtain

$$\begin{aligned}&\mathcal{F}_{\boldsymbol{P}^l}(\boldsymbol{P}^l+s\boldsymbol{M})\\ &=\boldsymbol{A}^{\mathrm{T}}\boldsymbol{P}^l\boldsymbol{A}-\boldsymbol{P}^l-s\boldsymbol{A}^{\mathrm{T}}\boldsymbol{M}\boldsymbol{A}+\boldsymbol{S}-s\boldsymbol{M}+\boldsymbol{\Delta}_1(s)\\ &\quad-\Theta(\boldsymbol{P}^l)\zeta(\boldsymbol{P}^l)(\Theta(\boldsymbol{P}^l))^{\mathrm{T}}-s\Theta(\boldsymbol{M})\zeta(\boldsymbol{P}^l)\\ &\quad\times(\Theta(\boldsymbol{P}^l))^{\mathrm{T}}-s\Theta(\boldsymbol{P}^l)\zeta(\boldsymbol{P}^l)(\Theta(\boldsymbol{M}))^{\mathrm{T}}+s\Theta(\boldsymbol{P}^l)\\ &\quad\times\zeta(\boldsymbol{P}^l)\vartheta(\boldsymbol{M})\zeta(\boldsymbol{P}^l)(\Theta(\boldsymbol{P}^l))^{\mathrm{T}},\end{aligned} \tag{17}$$

where $\boldsymbol{\Delta}_1(s)$ is the higher-order term on s; in other words, $\lim_{s\to 0}\boldsymbol{\Delta}_1(s)/s=\boldsymbol{0}_{n\times n}$. Combining Eqs. (13) and (17) we know that the Gâteaux derivative $\mathcal{G}$ equals the right-hand side of Eq. (16) according to the definition of the Gâteaux derivative (15). Clearly, $\mathcal{G}$ is continuous with respect to $\boldsymbol{M}$, because $\mathcal{G}$ is a linear function of $\boldsymbol{M}$, and the system matrices are constant. Therefore, the Fréchet derivative of $\mathcal{F}$ at $\boldsymbol{P}^l$ equals the Gâteaux derivative $\mathcal{G}$. The proof is completed.

Employing the result from Lemma 1, we can now prove that Algorithm 1 is equivalent to Newton's method for calculating the zero-point of Eq. (8) in the Banach space $\mathbb{R}^{n\times n}$.

Theorem 1 Let $\mathcal{T}$ be the mapping defined in Eq. (14). Then iteration between Eqs. (11) and (12) is equivalent to the following Newton method:

$$\boldsymbol{P}^{l+1}=\mathcal{T}(\boldsymbol{P}^l)=\boldsymbol{P}^l-(\mathcal{F}'_{\boldsymbol{P}^l})^{-1}(\boldsymbol{P}^l). \tag{18}$$

Proof To prove Eq. (18), we just need to prove the equivalent form which is given by $\mathcal{F}'_{\boldsymbol{P}^l}(\boldsymbol{P}^{l+1})=\mathcal{F}'_{\boldsymbol{P}^l}(\boldsymbol{P}^l)-\mathcal{F}(\boldsymbol{P}^l)$. From Eqs. (14) and (17), we obtain

$$\begin{aligned}&\mathcal{F}'_{\boldsymbol{P}^l}(\boldsymbol{P}^l)=\mathcal{F}(\boldsymbol{P}^l)-\boldsymbol{S}-\Theta(\boldsymbol{P}^l)\\ &\times\zeta(\boldsymbol{P}^l)(\Theta(\boldsymbol{P}^l))^{\mathrm{T}}+\Theta(\boldsymbol{P}^l)\zeta(\boldsymbol{P}^l)\vartheta(\boldsymbol{P}^l)\zeta(\boldsymbol{P}^l)(\Theta(\boldsymbol{P}^l))^{\mathrm{T}},\end{aligned} \tag{19}$$

with $\vartheta(\boldsymbol{P}^l)$ defined as follows:

$$\vartheta(\boldsymbol{P}^l)=\begin{bmatrix}\boldsymbol{B}^{\mathrm{T}}\boldsymbol{P}^l\boldsymbol{B} & \boldsymbol{B}^{\mathrm{T}}\boldsymbol{P}^l\boldsymbol{D}\\ \boldsymbol{D}^{\mathrm{T}}\boldsymbol{P}^l\boldsymbol{B} & \boldsymbol{D}^{\mathrm{T}}\boldsymbol{P}^l\boldsymbol{D}\end{bmatrix}.$$

Substituting Eq. (12) into Eq. (11) gives

$$\begin{aligned}-\boldsymbol{S}=&\boldsymbol{A}^{\mathrm{T}}\boldsymbol{P}^{l+1}\boldsymbol{A}-\boldsymbol{P}^{l+1}-\Theta(\boldsymbol{P}^{l+1})\zeta(\boldsymbol{P}^l)\\&\times(\Theta(\boldsymbol{P}^l))^{\mathrm{T}}-\Theta(\boldsymbol{P}^l)\zeta(\boldsymbol{P}^l)(\Theta(\boldsymbol{P}^{l+1}))^{\mathrm{T}}\\&+\Theta(\boldsymbol{P}^l)\zeta(\boldsymbol{P}^l)\big(\zeta(\boldsymbol{P}^{l+1})\big)^{-1}\zeta(\boldsymbol{P}^l)(\Theta(\boldsymbol{P}^l))^{\mathrm{T}},\end{aligned}\tag{20}$$

where $\Theta(\boldsymbol{P}^{l+1})=[\boldsymbol{A}^{\mathrm{T}}\boldsymbol{P}^{l+1}\boldsymbol{B}\quad\boldsymbol{A}^{\mathrm{T}}\boldsymbol{P}^{l+1}\boldsymbol{D}]$. Combining Eqs. (19) and (20) then results in

$$\begin{aligned}&\mathcal{F}'_{\boldsymbol{P}^l}(\boldsymbol{P}^l)-\mathcal{F}(\boldsymbol{P}^l)-\mathcal{F}'_{\boldsymbol{P}^l}(\boldsymbol{P}^{l+1})\\=&-\Theta(\boldsymbol{P}^l)\zeta(\boldsymbol{P}^l)(\Theta(\boldsymbol{P}^l))^{\mathrm{T}}+\Theta(\boldsymbol{P}^l)\zeta(\boldsymbol{P}^l)\vartheta(\boldsymbol{P}^l)\zeta(\boldsymbol{P}^l)\\&\times(\Theta(\boldsymbol{P}^l))^{\mathrm{T}}+\Theta(\boldsymbol{P}^l)\zeta(\boldsymbol{P}^l)\Big(\big(\zeta(\boldsymbol{P}^l)\big)^{-1}-\vartheta(\boldsymbol{P}^l)\Big)\\&\times\zeta(\boldsymbol{P}^l)(\Theta(\boldsymbol{P}^l))^{\mathrm{T}}=0.\end{aligned}\tag{21}$$

The proof is completed.

According to Theorem 1, we come to a conclusion that $\boldsymbol{P}^l$ and $(\boldsymbol{K}_1^l,\boldsymbol{K}_2^l)$ will converge to $\boldsymbol{P}$ and $(\boldsymbol{K}_1^*,\boldsymbol{K}_2^*)$, respectively, as the iteration number l tends to infinity.

Though Algorithm 1 provides a feasible scheme for solving zero-sum linear quadratic dynamic games by operating on the reduced-order linear matrix equations, Eqs. (11) and (12) still depend on the system model, which makes Algorithm 1 sensitive to the drift in system dynamics and the inaccuracy in system modeling.

4 Online minimax Q-learning method based on off-policy reinforcement learning

To develop an intelligent algorithm that can learn the saddle solution online without querying the information of the system model, in this section, we establish an online minimax Q-learning method by borrowing an idea from off-policy RL and adaptive control. We construct the minimax Q-learning method on the basis of Algorithm 1.

4.1 Derivation of the online minimax Q-learning algorithm

Let $\boldsymbol{u}^l=-\boldsymbol{K}_1^l\boldsymbol{x}$ and $\boldsymbol{w}^l=-\boldsymbol{K}_2^l\boldsymbol{x}$ be the given admissible policies at the l^{th} iteration in Algorithm 1. We define the following Q-function corresponding to $\boldsymbol{u}^l$ and $\boldsymbol{w}^l$:

$$\begin{aligned}&Q^{l+1}(\boldsymbol{x}_k,\boldsymbol{u}_k,\boldsymbol{w}_k)\\=&\boldsymbol{x}_k^{\mathrm{T}}\boldsymbol{S}\boldsymbol{x}_k+\boldsymbol{u}_k^{\mathrm{T}}\boldsymbol{R}\boldsymbol{u}_k-\gamma^2\boldsymbol{w}_k^{\mathrm{T}}\boldsymbol{w}_k+\boldsymbol{x}_{k+1}^{\mathrm{T}}\boldsymbol{P}^{l+1}\boldsymbol{x}_{k+1},\end{aligned}\tag{22}$$

where $\boldsymbol{u}$ and $\boldsymbol{w}$ are the behavior polices adopted at time k. Thus, the state at time $k+1$ is determined by $\boldsymbol{x}_{k+1}=\boldsymbol{A}\boldsymbol{x}_k+\boldsymbol{B}\boldsymbol{u}_k+\boldsymbol{D}\boldsymbol{w}_k$. From time $k+1$ on, one follows the target polices $\boldsymbol{u}^l$ and $\boldsymbol{w}^l$. According to the definition of the Q-function, we know that $Q^{l+1}(\boldsymbol{x}_k,\boldsymbol{u}_k,\boldsymbol{w}_k)$ contains two types of policies, namely, the behavior policies $\boldsymbol{u}$ and $\boldsymbol{w}$ applied to system (1) and the target polices $\boldsymbol{u}^l$ and $\boldsymbol{w}^l$ which are expected to converge to the saddle solution. In particular, $Q^{l+1}(\boldsymbol{x}_{k+1},\boldsymbol{u}_{k+1}^l,\boldsymbol{w}_{k+1}^l)=\boldsymbol{x}_{k+1}^{\mathrm{T}}\boldsymbol{P}^{l+1}\boldsymbol{x}_{k+1}$; therefore, Eq. (22) can be rewritten as the following Bellman equation:

$$\begin{aligned}Q^{l+1}(\boldsymbol{x}_k,\boldsymbol{u}_k,\boldsymbol{w}_k)=&\boldsymbol{x}_k^{\mathrm{T}}\boldsymbol{S}\boldsymbol{x}_k+\boldsymbol{u}_k^{\mathrm{T}}\boldsymbol{R}\boldsymbol{u}_k-\gamma^2\boldsymbol{w}_k^{\mathrm{T}}\boldsymbol{w}_k\\&+Q^{l+1}(\boldsymbol{x}_{k+1},\boldsymbol{u}_{k+1}^l,\boldsymbol{w}_{k+1}^l).\end{aligned}\tag{23}$$

We can now use Eq. (23) to calculate Q^{l+1} instead of solving Eq. (11) directly for $\boldsymbol{P}^{l+1}$; clearly, Eq. (23) requires no information for $\boldsymbol{A}$, $\boldsymbol{B}$, and $\boldsymbol{D}$. From Eq. (22), we can represent Q^{l+1} as a quadratic form of the state and inputs, or equivalently, Q^{l+1} can be expressed as $Q^{l+1}(\boldsymbol{x}_k,\boldsymbol{u}_k,\boldsymbol{w}_k)=\boldsymbol{W}_{\mathrm{c},l+1}^{\mathrm{T}}\boldsymbol{\sigma}_k$, where $\boldsymbol{\sigma}_k$ is the Kronecker product quadratic polynomial basis vector corresponding to $[\boldsymbol{x}_k^{\mathrm{T}},\,\boldsymbol{u}_k^{\mathrm{T}},\,\boldsymbol{w}_k^{\mathrm{T}}]^{\mathrm{T}}$, i.e., $\boldsymbol{\sigma}_k=\boldsymbol{\sigma}([\boldsymbol{x}_k^{\mathrm{T}},\boldsymbol{u}_k^{\mathrm{T}},\boldsymbol{w}_k^{\mathrm{T}}]^{\mathrm{T}})$. Hence, Eq. (23) can be rewritten as follows:

$$\begin{aligned}&\boldsymbol{W}_{\mathrm{c},l+1}^{\mathrm{T}}\boldsymbol{\sigma}_k\\=&\boldsymbol{x}_k^{\mathrm{T}}\boldsymbol{S}\boldsymbol{x}_k+\boldsymbol{u}_k^{\mathrm{T}}\boldsymbol{R}\boldsymbol{u}_k-\gamma^2\boldsymbol{w}_k^{\mathrm{T}}\boldsymbol{w}_k+\boldsymbol{W}_{\mathrm{c},l+1}^{\mathrm{T}}\boldsymbol{\sigma}_{k+1,l},\end{aligned}\tag{24}$$

where $\boldsymbol{\sigma}_{k+1,l}$ is the Kronecker product quadratic polynomial basis vector of $\big[\boldsymbol{x}_{k+1}^{\mathrm{T}},\,(\boldsymbol{u}_{k+1}^l)^{\mathrm{T}},\,(\boldsymbol{w}_{k+1}^l)^{\mathrm{T}}\big]^{\mathrm{T}}$, that is, $\boldsymbol{\sigma}_{k+1,l}=\boldsymbol{\sigma}\Big(\big[\boldsymbol{x}_{k+1}^{\mathrm{T}},\,(\boldsymbol{u}_{k+1}^l)^{\mathrm{T}},\,(\boldsymbol{w}_{k+1}^l)^{\mathrm{T}}\big]^{\mathrm{T}}\Big)$. Next, we aim to obtain the true weight $\boldsymbol{W}_{\mathrm{c},l+1}$ online in real time, using only the data samples generated by the behavior polices. This is essentially a prediction problem in RL and can be solved by TD learning techniques; from the perspective of adaptive control, it becomes an online parameter identification problem. Let $\hat{\boldsymbol{W}}_{\mathrm{c},l+1}(i)$ be the estimate of $\boldsymbol{W}_{\mathrm{c},l+1}$ at time k, with $i\le k$. Replacing $\boldsymbol{W}_{\mathrm{c},l+1}$ with $\hat{\boldsymbol{W}}_{\mathrm{c},l+1}(i)$ in Eq. (24) gives

the estimation error:

$$\begin{aligned} e_k =& \boldsymbol{x}_k^{\mathrm{T}}\boldsymbol{S}\boldsymbol{x}_k + \boldsymbol{u}_k^{\mathrm{T}}\boldsymbol{R}\boldsymbol{u}_k - \gamma^2\boldsymbol{w}_k^{\mathrm{T}}\boldsymbol{w}_k \\ &+ \hat{\boldsymbol{W}}_{\mathrm{c},l+1}^{\mathrm{T}}(i)(\boldsymbol{\sigma}_{k+1,l} - \boldsymbol{\sigma}_k). \end{aligned} \tag{25}$$

Recursive least squares (RLS) (Ioannou and Fidan, 2006) can now be used to estimate $\boldsymbol{W}_{\mathrm{c},l+1}$ online in real time:

$$\hat{\boldsymbol{W}}_{\mathrm{c},l+1}(i+1) = \hat{\boldsymbol{W}}_{\mathrm{c},l+1}(i) - \frac{\boldsymbol{P}_l(i)\bar{\boldsymbol{\sigma}}_k e_k}{1+\bar{\boldsymbol{\sigma}}_k^{\mathrm{T}}\boldsymbol{P}_l(i)\bar{\boldsymbol{\sigma}}_k}, \tag{26a}$$

$$\boldsymbol{P}_l(i+1) = \boldsymbol{P}_l(i) - \frac{\boldsymbol{P}_l(i)\bar{\boldsymbol{\sigma}}_k\bar{\boldsymbol{\sigma}}_k^{\mathrm{T}}\boldsymbol{P}_l(i)}{1+\bar{\boldsymbol{\sigma}}_k^{\mathrm{T}}\boldsymbol{P}_l(i)\bar{\boldsymbol{\sigma}}_k}, \tag{26b}$$

where $\bar{\boldsymbol{\sigma}}_k = \boldsymbol{\sigma}_{k+1,l} - \boldsymbol{\sigma}_k$ and $\boldsymbol{P}_l(1) = \gamma\boldsymbol{I}$ (γ is a large positive constant). During the learning process, the tuning index i is increased with the time index k; in other words, $\hat{\boldsymbol{W}}_{\mathrm{c},l+1}(i)$ is tuned online by using the state data generated by the behavior policies. According to Ioannou and Fidan (2006), $\hat{\boldsymbol{W}}_{\mathrm{c},l+1}(i)$ will converge to $\boldsymbol{W}_{\mathrm{c},l+1}$, if $\bar{\boldsymbol{\sigma}}_k$ satisfies the following PE condition:

$$\varepsilon_0\boldsymbol{I} \le \sum_{i=1}^{l-1}\bar{\boldsymbol{\sigma}}_{k+i}\bar{\boldsymbol{\sigma}}_{k+i}^{\mathrm{T}} \le \varepsilon_1\boldsymbol{I},\ \forall k \ge 0,$$

where l is a positive integer. The PE condition requires that the system state be persistently exciting for a long enough period of time.

To meet the PE condition above, we can inject some exploration noise into the behavior polices $\boldsymbol{u}$ and $\boldsymbol{w}$. Note that the injected exploration noise will not cause any excitation noise bias, although the excitation noise bias problem cannot be eliminated in on-policy methods (Kiumarsi et al., 2017). For the sake of explanation, we should confirm the fact that the Q-function Q^{l+1} is essentially a mapping from the state-input space to $\mathbb{R}$; thus, we can use the behavior policies and the state samples generated by them to identify Q^{l+1} at each policy evaluation step. The exploration noises can be selected as harmonic signals containing sufficient frequencies or random noises. Because there exist no systematic methods for choosing exploration noises, one can choose them by trial-and-error.

After obtaining the Q-function Q^{l+1}, we carry out the policy improvement step by solving the following minimax optimization problem:

$$\begin{aligned} \left(\boldsymbol{u}_k^{l+1}, \boldsymbol{w}_k^{l+1}\right) =& \min_{\tilde{\boldsymbol{u}}_k}\max_{\tilde{\boldsymbol{w}}_k} Q^{l+1}(\boldsymbol{x}_k, \tilde{\boldsymbol{u}}_k, \tilde{\boldsymbol{w}}_k) \\ =& \min_{\tilde{\boldsymbol{u}}_k}\max_{\tilde{\boldsymbol{w}}_k}\left(\boldsymbol{x}_k^{\mathrm{T}}\boldsymbol{S}\boldsymbol{x}_k + \tilde{\boldsymbol{u}}_k^{\mathrm{T}}\boldsymbol{R}\tilde{\boldsymbol{u}}_k \right. \\ & \left. -\gamma^2\tilde{\boldsymbol{w}}_k^{\mathrm{T}}\tilde{\boldsymbol{w}}_k + \boldsymbol{x}_{k+1}^{\mathrm{T}}\boldsymbol{P}^{l+1}\boldsymbol{x}_{k+1}\right), \end{aligned} \tag{27}$$

where $\boldsymbol{x}_{k+1} = \boldsymbol{A}\boldsymbol{x}_k + \boldsymbol{B}\tilde{\boldsymbol{u}}_k + \boldsymbol{D}\tilde{\boldsymbol{w}}_k$. Denote $\boldsymbol{u}_k^{l+1} = -\boldsymbol{K}_1^{l+1}\boldsymbol{x}_k$ and $\boldsymbol{w}_k^{l+1} = -\boldsymbol{K}_2^{l+1}\boldsymbol{x}_k$. Obviously, Eq. (27) can be reformulated as

$$\left(\bar{\boldsymbol{K}}_1^{l+1}, \bar{\boldsymbol{K}}_2^{l+1}\right) = \min_{\boldsymbol{K}_1}\max_{\boldsymbol{K}_2} Q^{l+1}\left(\boldsymbol{x}_k, \boldsymbol{K}_1^{\mathrm{T}}\boldsymbol{x}_k, \boldsymbol{K}_2^{\mathrm{T}}\boldsymbol{x}_k\right), \tag{28}$$

where $\bar{\boldsymbol{K}}_1^{l+1} = -(\boldsymbol{K}_1^{l+1})^{\mathrm{T}}$ and $\bar{\boldsymbol{K}}_2^{l+1} = -(\boldsymbol{K}_2^{l+1})^{\mathrm{T}}$ ($\bar{\boldsymbol{K}}_1^{l+1} \in \mathbb{R}^{n\times m_1}$, $\bar{\boldsymbol{K}}_2^{l+1} \in \mathbb{R}^{n\times m_2}$). To obtain $\left(\bar{\boldsymbol{K}}_1^{l+1}, \bar{\boldsymbol{K}}_2^{l+1}\right)$, one needs to select a set of points $\boldsymbol{x}_k$ for training. Considering that $\boldsymbol{A}$, $\boldsymbol{B}$, and $\boldsymbol{D}$ are unknown, we cannot solve Eq. (27) or (28) directly. To overcome this, we create a novel online gradient-based policy improvement scheme. The main idea is that we use two behavior polices to generate a sequence of state samples online. Once a new state sample is generated, we update the estimated values of $\boldsymbol{K}_1$ and $\boldsymbol{K}_2$ simultaneously using the newly generated state sample, which means that the updating of the estimated values and the generation of state samples are concurrent. Let $(\hat{\boldsymbol{K}}_{1,l+1}(j), \hat{\boldsymbol{K}}_{2,l+1}(j))$ be the estimate of $(\bar{\boldsymbol{K}}_1^{l+1}, \bar{\boldsymbol{K}}_2^{l+1})$ at $\boldsymbol{x}_k$, where $j \le k$. Then $\hat{\boldsymbol{K}}_{1,l+1}(j)$ and $\hat{\boldsymbol{K}}_{2,l+1}(j)$ are tuned via the following normalized gradient method:

$$\begin{aligned} \hat{\boldsymbol{K}}_{1,l+1}(j+1) =& \hat{\boldsymbol{K}}_{1,l+1}(j) - \frac{\beta}{\left(1+\boldsymbol{x}_k^{\mathrm{T}}\boldsymbol{x}_k\right)^2} \\ &\times\frac{\partial}{\partial\hat{\boldsymbol{K}}_{1,l+1}(j)}Q^{l+1}\left(\boldsymbol{x}_k, \hat{\boldsymbol{K}}_{1,l+1}^{\mathrm{T}}(j)\boldsymbol{x}_k, \hat{\boldsymbol{K}}_{2,l+1}^{\mathrm{T}}(j)\boldsymbol{x}_k\right), \end{aligned} \tag{29a}$$

$$\begin{aligned} \hat{\boldsymbol{K}}_{2,l+1}(j+1) =& \hat{\boldsymbol{K}}_{2,l+1}(j) + \frac{\beta}{\left(1+\boldsymbol{x}_k^{\mathrm{T}}\boldsymbol{x}_k\right)^2} \\ &\times\frac{\partial}{\partial\hat{\boldsymbol{K}}_{2,l+1}(j)}Q^{l+1}\left(\boldsymbol{x}_k, \hat{\boldsymbol{K}}_{1,l+1}^{\mathrm{T}}(j)\boldsymbol{x}_k, \hat{\boldsymbol{K}}_{2,l+1}^{\mathrm{T}}(j)\boldsymbol{x}_k\right), \end{aligned} \tag{29b}$$

where $\Phi_a(k) = (1+\boldsymbol{x}_k^{\mathrm{T}}\boldsymbol{x}_k)^2$ is the normalized term, and β is a small positive learning rate. During the learning process, the state samples are generated by behavior policies $\boldsymbol{u}'$ and $\boldsymbol{w}'$ (that is, $\boldsymbol{x}_{k+1} = \boldsymbol{A}\boldsymbol{x}_k + \boldsymbol{B}\boldsymbol{u}'_k + \boldsymbol{D}\boldsymbol{w}'_k$), and the tuning index j is also increased with the time index k. From Eqs. (29a) and (29b), we know that the controller and the disturbance perform gradient descent and gradient ascent, respectively. In the following theorem, we

will prove that $(\hat{\boldsymbol{K}}_{1,l+1}(j), \hat{\boldsymbol{K}}_{2,l+1}(j))$ converges to $(\bar{\boldsymbol{K}}_1^{l+1}, \bar{\boldsymbol{K}}_2^{l+1})$ exponentially, if $\bar{\boldsymbol{x}}_k = \boldsymbol{x}_k/(1+\boldsymbol{x}_k^{\mathrm{T}}\boldsymbol{x}_k)$ is persistently exciting and β is small enough.

We now give the complete online minimax Q-learning algorithm.

Compared with Algorithm 1, Algorithm 2 is completely model-free, and thus robust to the drift in system dynamics and the inaccuracy in system modeling. In addition, both policy evaluation and policy improvement are carried out in an online adaptive way by using the state samples generated by the behavior polices. Note that this novel online policy improvement scheme provides a potential choice for other problems, e.g., optimal control and a nonzero-sum game of discrete-time systems.

Algorithm 2 Online minimax Q-learning algorithm

1: Start with a set of initially stabilizing feedback gains $(\boldsymbol{K}_1^1, \boldsymbol{K}_2^1)$ // Initialization
2: For the given stabilizing feedback gains $(\bar{\boldsymbol{K}}_1^1, \bar{\boldsymbol{K}}_2^1)$, run Eqs. (26a) and (26b) until $\hat{\boldsymbol{W}}_{\mathrm{c},l+1}(i+1)$ converges to $\boldsymbol{W}_{\mathrm{c},l+1}$ // Policy evaluation
3: Using the obtained $\boldsymbol{W}_{\mathrm{c},l+1}$, run Eqs. (29a) and (29b) simultaneously until $(\hat{\boldsymbol{K}}_{1,l+1}(j), \hat{\boldsymbol{K}}_{2,l+1}(j))$ converges to $(\bar{\boldsymbol{K}}_1^{l+1}, \bar{\boldsymbol{K}}_2^{l+1})$ // Policy improvement
4: Stop if $\left\|\boldsymbol{K}_i^{l+1} - \boldsymbol{K}_i^l\right\| \le \varepsilon$ $(i=1,2)$, where ε is a threshold; otherwise, set $l = l+1$ and go to step 2

4.2 Convergence analysis of the proposed online minimax Q-learning algorithm

In the following theorem, we will give the convergence analysis of the proposed minimax Q-learning method. Before deriving the main theorem, we first provide two lemmas that will be used in the following convergence analysis. The first lemma is taken from Ioannou and Fidan (2006), which is given as follows:

Lemma 2 (Ioannou and Fidan, 2006) Consider a time-varying linear discrete-time system $\boldsymbol{y}_{k+1} = \boldsymbol{C}(k)\boldsymbol{y}_k$. Suppose that there exists a positive definite symmetric constant matrix $\boldsymbol{M}$ such that

$$\boldsymbol{C}^{\mathrm{T}}(k)\boldsymbol{M}\boldsymbol{C}(k) - \boldsymbol{M} = -\boldsymbol{N}(k)\boldsymbol{N}^{\mathrm{T}}(k)$$

for some matrix sequence $\{\boldsymbol{N}(k)\}$ and all k. If $(\boldsymbol{A}(k), \boldsymbol{N}(k))$ is also uniformly completely observable (UCO), i.e., there exist constants $\alpha > 0$, $\gamma > 0$, and $l > 0$ such that for all k,

$$0 < \alpha\boldsymbol{I} \le \sum_{i=0}^{l-1} \boldsymbol{\Phi}^{\mathrm{T}}(k+i,k)\boldsymbol{N}(k+i) \times \boldsymbol{N}^{\mathrm{T}}(k+i)\boldsymbol{\Phi}(k+i,k) \le \gamma\boldsymbol{I} < \infty,$$

where $\boldsymbol{\Phi}(k+i,k) = \boldsymbol{C}(k+i-1)\boldsymbol{C}(k+i-2)\ldots\boldsymbol{C}(k+1)\boldsymbol{C}(k)$ is the transition matrix of the linear system, then $\boldsymbol{y}(k)$ will converge to the origin exponentially.

Before stating the next lemma, we introduce one useful property of the Kronecker product on the matrix eigenvalue. Suppose that $\boldsymbol{A}$ and $\boldsymbol{B}$ are square matrices of sizes n and m, respectively. Let $\lambda_1, \lambda_2, \ldots, \lambda_n$ be the eigenvalues of $\boldsymbol{A}$ and $\mu_1, \mu_2, \ldots, \mu_m$ be those of $\boldsymbol{B}$. Then the eigenvalues of $\boldsymbol{A} \otimes \boldsymbol{B}$ are $\lambda_i\mu_j$ $(i = 1, 2, \ldots, n,\ j = 1, 2, \ldots, m)$.

Lemma 3 Consider a time-varying linear discrete-time system given by $\boldsymbol{z}_{k+1} = (\boldsymbol{I} - 2\eta(\boldsymbol{\theta}_k\boldsymbol{\theta}_k^{\mathrm{T}}) \otimes \boldsymbol{E})\boldsymbol{z}_k$, where $\{\boldsymbol{\theta}_k\}$ is a sequence of bounded column vectors, η is a positive constant, and $\boldsymbol{E}$ is a positive definite matrix. Let $\boldsymbol{\theta}_k$ be persistently exciting and η be small enough. Then $\boldsymbol{z}_k$ will converge to the origin exponentially.

Proof Let $\boldsymbol{W}(k) = (\boldsymbol{I} - 2\eta(\boldsymbol{\theta}_k\boldsymbol{\theta}_k^{\mathrm{T}}) \otimes \boldsymbol{E})$. Employing the Kronecker product, $\boldsymbol{W}(k)$ can be rewritten as $\boldsymbol{W}(k) = (\boldsymbol{I} - 2\eta\bar{\boldsymbol{\theta}}_k\bar{\boldsymbol{\theta}}_k^{\mathrm{T}})$, with $\bar{\boldsymbol{\theta}}_k = \boldsymbol{\theta}_k \otimes \boldsymbol{E}^{1/2}$. We first prove that $\bar{\boldsymbol{\theta}}_k$ is also persistently exciting. As $\boldsymbol{\theta}_k$ is persistently exciting, there exist $\alpha_1 > 0$, $\alpha_2 > 0$, and $l > 0$ such that $\alpha_1\boldsymbol{I} \le \sum_{i=0}^{l-1} \boldsymbol{\theta}_{k+i}\boldsymbol{\theta}_{k+i}^{\mathrm{T}} \le \alpha_2\boldsymbol{I}$. Considering that $\bar{\boldsymbol{\theta}}_{k+i}\bar{\boldsymbol{\theta}}_{k+i}^{\mathrm{T}} = (\boldsymbol{\theta}_{k+i}\boldsymbol{\theta}_{k+i}^{\mathrm{T}}) \otimes \boldsymbol{E}$, we have $\sum_{i=0}^{l-1} \bar{\boldsymbol{\theta}}_{k+i}\bar{\boldsymbol{\theta}}_{k+i}^{\mathrm{T}} = (\sum_{i=0}^{l-1} \boldsymbol{\theta}_{k+i}\boldsymbol{\theta}_{k+i}^{\mathrm{T}}) \otimes \boldsymbol{E}$. Using the property of the Kronecker product on the matrix eigenvalue, we obtain $\lambda_{\min}(\boldsymbol{E})\alpha_1\boldsymbol{I} \le \sum_{i=0}^{l-1} \bar{\boldsymbol{\theta}}_{k+i}\bar{\boldsymbol{\theta}}_{k+i}^{\mathrm{T}} \le \lambda_{\max}(\boldsymbol{E})\alpha_2\boldsymbol{I}$. Let $\boldsymbol{M} = \frac{1}{2\eta}\boldsymbol{I}$ and $\boldsymbol{N}(k) = \bar{\boldsymbol{\theta}}_k(2 - 2\eta\bar{\boldsymbol{\theta}}_k^{\mathrm{T}}\bar{\boldsymbol{\theta}}_k)^{1/2}$. Then we have

$$\boldsymbol{W}^{\mathrm{T}}(k)\boldsymbol{M}\boldsymbol{W}(k) - \boldsymbol{M} = -\boldsymbol{N}(k)\boldsymbol{N}^{\mathrm{T}}(k). \tag{30}$$

Next, we prove that $(\boldsymbol{W}(k),\ \boldsymbol{N}(k))$ is UCO. Consider the following system:

$$\begin{cases} \boldsymbol{x}_{k+1} = \boldsymbol{W}(k)\boldsymbol{x}_k = (\boldsymbol{I} - 2\eta\bar{\boldsymbol{\theta}}_k\bar{\boldsymbol{\theta}}_k^{\mathrm{T}})\boldsymbol{x}_k, \\ \boldsymbol{y}_k = \boldsymbol{N}^{\mathrm{T}}(k)\boldsymbol{x}_k = \left(2 - 2\eta\bar{\boldsymbol{\theta}}_k^{\mathrm{T}}\bar{\boldsymbol{\theta}}_k\right)^{1/2}\bar{\boldsymbol{\theta}}_k^{\mathrm{T}}\boldsymbol{x}_k. \end{cases} \tag{31}$$

Clearly, system (31) is equivalent to the following system:

$$\begin{cases} \boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \boldsymbol{u}_k, \\ \boldsymbol{y}_k = \left(2 - 2\eta\bar{\boldsymbol{\theta}}_k^{\mathrm{T}}\bar{\boldsymbol{\theta}}_k\right)^{1/2}\bar{\boldsymbol{\theta}}_k^{\mathrm{T}}\boldsymbol{x}_k, \end{cases} \tag{32}$$

with output feedback

$$\boldsymbol{u}_k=\frac{-2\eta\bar{\boldsymbol{\theta}}_k\boldsymbol{y}_k}{(2-2\eta\bar{\boldsymbol{\theta}}_k^{\mathrm{T}}\bar{\boldsymbol{\theta}}_k)^{1/2}}.$$

So, we can prove that system (32) is UCO instead. Because $\boldsymbol{\theta}_k$ is bounded, $\bar{\boldsymbol{\theta}}_k$ is also bounded; thus, there exists $a>0$ such that $\bar{\boldsymbol{\theta}}_k^{\mathrm{T}}\bar{\boldsymbol{\theta}}_k\le a$ is satisfied for all k. Let $\eta\le 1/(2a)$. We have $1\le 2-2\eta\bar{\boldsymbol{\theta}}_k^{\mathrm{T}}\bar{\boldsymbol{\theta}}_k\le 2$; therefore,

$$\bar{\alpha}_1\boldsymbol{I}\le\sum_{i=0}^{l-1}\boldsymbol{N}(k+i)\boldsymbol{N}^{\mathrm{T}}(k+i)\le\bar{\alpha}_2\boldsymbol{I},\tag{33}$$

where $\bar{\alpha}_1=\lambda_{\min}(\boldsymbol{E})\alpha_1$ and $\bar{\alpha}_2=2\lambda_{\max}(\boldsymbol{E})\alpha_2$. Clearly, the transition matrix of system (32) is $\boldsymbol{I}$, and Eq. (30) and inequality (33) imply that system (32) is UCO, or equivalently, $(\boldsymbol{W}(k),\boldsymbol{N}(k))$ is UCO. Using the result from Lemma 2, we know that $\boldsymbol{z}_k$ will converge to the origin exponentially.

We are now ready to state and prove the following theorem:

Theorem 2 Let $\bar{\boldsymbol{\sigma}}_k$ and $\bar{\boldsymbol{x}}_k$ be persistently exciting and the learning rate β satisfy

$$0<\beta<\min\left(-\frac{\lambda}{r(k_0)\theta}\ln\lambda,\frac{1}{2a_1},\frac{1}{2a_2}\right).\tag{34}$$

Then, in Algorithm 2, $(\hat{\boldsymbol{K}}_{1,l+1}(j),\hat{\boldsymbol{K}}_{2,l+1}(j))$ will converge to $(\bar{\boldsymbol{K}}_1^{l+1},\bar{\boldsymbol{K}}_2^{l+1})$ exponentially; further, as $l\to\infty$, $((\bar{\boldsymbol{K}}_1^{l})^{\mathrm{T}},(\bar{\boldsymbol{K}}_2^{l})^{\mathrm{T}})$ will converge to the saddle feedback gains $(-\boldsymbol{K}_1^*,-\boldsymbol{K}_2^*)$.

Proof First, we prove that $(\hat{\boldsymbol{K}}_{1,l+1}(j),\hat{\boldsymbol{K}}_{2,l+1}(j))$ will converge to $(\bar{\boldsymbol{K}}_1^{l+1},\bar{\boldsymbol{K}}_2^{l+1})$, if $\bar{\boldsymbol{x}}_k$ is persistently exciting and the learning rate β satisfies inequality (34). From Eq. (28), we know that $Q^{l+1}(\boldsymbol{x}_k,\boldsymbol{K}_1^{\mathrm{T}}\boldsymbol{x}_k,\boldsymbol{K}_2^{\mathrm{T}}\boldsymbol{x}_k)$ reaches a saddle point at $(\bar{\boldsymbol{K}}_1^{l+1},\bar{\boldsymbol{K}}_2^{l+1})$, observing that Q^{l+1} is convex in $\boldsymbol{K}_1$ and $\boldsymbol{K}_2$. The first-order necessary condition implies

$$\frac{\partial}{\partial\boldsymbol{K}_1}Q^{l+1}\left(\boldsymbol{x}_k,\boldsymbol{K}_1^{\mathrm{T}}\boldsymbol{x}_k,(\bar{\boldsymbol{K}}_2^{l+1})^{\mathrm{T}}\boldsymbol{x}_k\right)\Big|_{\boldsymbol{K}_1=\bar{\boldsymbol{K}}_1^{l+1}}=0,\tag{35a}$$

$$\frac{\partial}{\partial\boldsymbol{K}_2}Q^{l+1}\left(\boldsymbol{x}_k,(\bar{\boldsymbol{K}}_1^{l+1})^{\mathrm{T}}\boldsymbol{x}_k,\boldsymbol{K}_2^{\mathrm{T}}\boldsymbol{x}_k\right)\Big|_{\boldsymbol{K}_2=\bar{\boldsymbol{K}}_2^{l+1}}=0.\tag{35b}$$

From the definition of $Q^{l+1}(\boldsymbol{x}_k,\boldsymbol{K}_1^{\mathrm{T}}\boldsymbol{x}_k,\boldsymbol{K}_2^{\mathrm{T}}\boldsymbol{x}_k)$, Eqs. (35a) and (35b) can be rewritten as follows:

$$\beta\boldsymbol{x}_k\big(2\boldsymbol{R}(\boldsymbol{K}_1^{l+1})^{\mathrm{T}}\boldsymbol{x}_k+2\boldsymbol{B}^{\mathrm{T}}\boldsymbol{P}^{l+1}\tilde{\boldsymbol{x}}_{k+1}\big)^{\mathrm{T}}=0,\tag{36a}$$

$$\beta\boldsymbol{x}_k\big(-2\gamma^2(\boldsymbol{K}_2^{l+1})^{\mathrm{T}}\boldsymbol{x}_k+2\boldsymbol{D}^{\mathrm{T}}\boldsymbol{P}^{l+1}\tilde{\boldsymbol{x}}_{k+1}\big)^{\mathrm{T}}=0,\tag{36b}$$

where $\tilde{\boldsymbol{x}}_{k+1}=\boldsymbol{A}\boldsymbol{x}_k+\boldsymbol{B}(\bar{\boldsymbol{K}}_1^{l+1})^{\mathrm{T}}\boldsymbol{x}_k+\boldsymbol{D}(\bar{\boldsymbol{K}}_2^{l+1})^{\mathrm{T}}\boldsymbol{x}_k$. Similarly, Eqs. (29a) and (29b) can be given by

$$\begin{aligned}\hat{\boldsymbol{K}}_{1,l+1}(j+1)=&\hat{\boldsymbol{K}}_{1,l+1}(j)-\beta\frac{\boldsymbol{x}_k}{\Phi_a(k)}\\&\times\Big(2\boldsymbol{R}\hat{\boldsymbol{K}}_{1,l+1}^{\mathrm{T}}(j)\boldsymbol{x}_k+2\boldsymbol{B}^{\mathrm{T}}\boldsymbol{P}^{l+1}\hat{\boldsymbol{x}}_{k+1}\Big)^{\mathrm{T}},\end{aligned}\tag{37a}$$

$$\begin{aligned}\hat{\boldsymbol{K}}_{2,l+1}(j+1)=&\hat{\boldsymbol{K}}_{2,l+1}(j)+\beta\frac{\boldsymbol{x}_k}{\Phi_a(k)}\\&\times\Big(-2\gamma^2\hat{\boldsymbol{K}}_{2,l+1}^{\mathrm{T}}(j)\boldsymbol{x}_k+2\boldsymbol{D}^{\mathrm{T}}\boldsymbol{P}^{l+1}\hat{\boldsymbol{x}}_{k+1}\Big)^{\mathrm{T}},\end{aligned}\tag{37b}$$

where

$$\hat{\boldsymbol{x}}_{k+1}=\boldsymbol{A}\boldsymbol{x}_k+\boldsymbol{B}\hat{\boldsymbol{K}}_{1,l+1}^{\mathrm{T}}(j)\boldsymbol{x}_k+\boldsymbol{D}\hat{\boldsymbol{K}}_{2,l+1}^{\mathrm{T}}(j)\boldsymbol{x}_k.$$

Define the following estimation errors: $\tilde{\boldsymbol{K}}_{1,l+1}=\hat{\boldsymbol{K}}_{1,l+1}-\bar{\boldsymbol{K}}_1^{l+1}$ and $\tilde{\boldsymbol{K}}_{2,l+1}=\hat{\boldsymbol{K}}_{2,l+1}-\bar{\boldsymbol{K}}_2^{l+1}$. Combining Eqs. (36a) and (36b) with Eqs. (37a) and (37b) gives the following error dynamics:

$$\begin{aligned}\tilde{\boldsymbol{K}}_{1,l+1}(j+1)=&\tilde{\boldsymbol{K}}_{1,l+1}(j)-2\beta\bar{\boldsymbol{x}}_k\bar{\boldsymbol{x}}_k^{\mathrm{T}}\tilde{\boldsymbol{K}}_{1,l+1}(j)\boldsymbol{R}\\&-2\beta\bar{\boldsymbol{x}}_k\bar{\boldsymbol{x}}_k^{\mathrm{T}}\boldsymbol{K}_{1,l+1}(j)\boldsymbol{B}^{\mathrm{T}}\boldsymbol{P}^{l+1}\boldsymbol{B}\\&-2\beta\bar{\boldsymbol{x}}_k\bar{\boldsymbol{x}}_k^{\mathrm{T}}\tilde{\boldsymbol{K}}_{2,l+1}(j)\boldsymbol{D}^{\mathrm{T}}\boldsymbol{P}^{l+1}\boldsymbol{B},\end{aligned}\tag{38a}$$

$$\begin{aligned}\tilde{\boldsymbol{K}}_{2,l+1}(j+1)=&\tilde{\boldsymbol{K}}_{2,l+1}(j)-2\beta\gamma^2\bar{\boldsymbol{x}}_k\bar{\boldsymbol{x}}_k^{\mathrm{T}}\tilde{\boldsymbol{K}}_{2,l+1}(j)\\&+2\beta\bar{\boldsymbol{x}}_k\bar{\boldsymbol{x}}_k^{\mathrm{T}}\boldsymbol{K}_{2,l+1}(j)\boldsymbol{D}^{\mathrm{T}}\boldsymbol{P}^{l+1}\boldsymbol{D}\\&+2\beta\bar{\boldsymbol{x}}_k\bar{\boldsymbol{x}}_k^{\mathrm{T}}\tilde{\boldsymbol{K}}_{1,l+1}(j)\boldsymbol{B}^{\mathrm{T}}\boldsymbol{P}^{l+1}\boldsymbol{D},\end{aligned}\tag{38b}$$

where $\bar{\boldsymbol{x}}_k=\boldsymbol{x}_k/(1+\boldsymbol{x}_k^{\mathrm{T}}\boldsymbol{x}_k)$. Using the fact that $\mathrm{vec}(\boldsymbol{XYZ})=(\boldsymbol{Z}^{\mathrm{T}}\otimes\boldsymbol{X})\mathrm{vec}(\boldsymbol{Y})$ holds for any matrices $\boldsymbol{X}$, $\boldsymbol{Y}$, and $\boldsymbol{Z}$ with appropriate dimensions, we can rewrite Eqs. (38a) and (38b) in the following compact form:

$$\begin{aligned}\begin{bmatrix}\mathrm{vec}(\tilde{\boldsymbol{K}}_{1,l+1}(j+1))\\\mathrm{vec}(\tilde{\boldsymbol{K}}_{2,l+1}(j+1))\end{bmatrix}=&\begin{bmatrix}\boldsymbol{A}_1(k)&\boldsymbol{A}_2(k)\\\boldsymbol{A}_3(k)&\boldsymbol{A}_4(k)\end{bmatrix}\\&\times\begin{bmatrix}\mathrm{vec}(\tilde{\boldsymbol{K}}_{1,l+1}(j))\\\mathrm{vec}(\tilde{\boldsymbol{K}}_{2,l+1}(j))\end{bmatrix},\end{aligned}\tag{39}$$

where

$$\begin{cases}\boldsymbol{A}_1(k)=\boldsymbol{I}-2\beta\bar{\boldsymbol{x}}_{k1}\bar{\boldsymbol{x}}_{k1}^{\mathrm{T}},\\\boldsymbol{A}_2(k)=-2\beta(\boldsymbol{B}^{\mathrm{T}}\boldsymbol{P}^{l+1}\boldsymbol{D})\otimes(\bar{\boldsymbol{x}}_k\bar{\boldsymbol{x}}_k^{\mathrm{T}}),\\\boldsymbol{A}_3(k)=2\beta(\boldsymbol{D}^{\mathrm{T}}\boldsymbol{P}^{l+1}\boldsymbol{B})\otimes(\bar{\boldsymbol{x}}_k\bar{\boldsymbol{x}}_k^{\mathrm{T}}),\\\boldsymbol{A}_4(k)=\boldsymbol{I}-2\beta\bar{\boldsymbol{x}}_{k2}\bar{\boldsymbol{x}}_{k2}^{\mathrm{T}},\\\bar{\boldsymbol{x}}_{k1}(k)=\bar{\boldsymbol{x}}_k\otimes\sqrt{\boldsymbol{R}+\boldsymbol{B}^{\mathrm{T}}\boldsymbol{P}^{l+1}\boldsymbol{B}},\\\bar{\boldsymbol{x}}_{k2}=\bar{\boldsymbol{x}}_k\otimes\sqrt{\gamma^2\boldsymbol{I}-\boldsymbol{D}^{\mathrm{T}}\boldsymbol{P}^{l+1}\boldsymbol{D}}.\end{cases}$$

According to the Kronecker product, we know that both $\bar{\boldsymbol{x}}_{k1}$ and $\bar{\boldsymbol{x}}_{k2}$ are also persistently exciting if

$\bar{\boldsymbol{x}}(k)$ is persistently exciting. Define the following matrices:

$$\bar{\boldsymbol{A}}(k)=\begin{bmatrix}\boldsymbol{A}_1(k) & \boldsymbol{0}\\ \boldsymbol{0} & \boldsymbol{A}_4(k)\end{bmatrix},$$

$$\boldsymbol{\Delta}(k)=\begin{bmatrix}\boldsymbol{0} & \boldsymbol{A}_2(k)\\ \boldsymbol{A}_3(k) & \boldsymbol{0}\end{bmatrix}.$$

Let the learning process start at k_0, i.e., $j=k-k_0+1$. Let

$$\boldsymbol{Z}(j)=\left[\mathrm{vec}(\tilde{\boldsymbol{K}}_{1,l+1}(j))^{\mathrm{T}}\ \mathrm{vec}(\tilde{\boldsymbol{K}}_{2,l+1}(j))^{\mathrm{T}}\right]^{\mathrm{T}}.$$

Using the result from Lemma 3, we know that the following time-varying system

$$\boldsymbol{Z}(j+1)=\bar{\boldsymbol{A}}(j+k_0-1)\boldsymbol{Z}(j) \tag{40}$$

is exponentially stable if the learning rate β is selected such that $\beta\le\min(1/(2a_1),1/(2a_2))$ is satisfied, where $a_1=\sup(\bar{\boldsymbol{x}}_{k1}^{\mathrm{T}}\bar{\boldsymbol{x}}_{k1})$ and $a_2=\sup(\bar{\boldsymbol{x}}_{k2}^{\mathrm{T}}\bar{\boldsymbol{x}}_{k2})$. Therefore, there exist $\gamma(k_0)>0$ and $\lambda\in(0,1)$ such that $\left\|\boldsymbol{\Phi}_{\bar{\boldsymbol{A}}(j+k_0-1)}(j,1)\right\|\le\gamma(k_0)\lambda^{j-1}$, where $\boldsymbol{\Phi}_{\bar{\boldsymbol{A}}(j+k_0-1)}(j,1)$ is the state transition matrix of system (40). Clearly, $\bar{\boldsymbol{x}}(k),\boldsymbol{B},\boldsymbol{D}$, and $\boldsymbol{P}^{l+1}$ are bounded; thus, there exists a positive constant θ such that $\|\boldsymbol{\Delta}(k)\|\le\beta\theta$ is satisfied for all k. We now rewrite Eq. (39) as

$$\boldsymbol{Z}(j+1)=(\bar{\boldsymbol{A}}(j+k_0-1)+\boldsymbol{\Delta}(j+k_0-1))\boldsymbol{Z}(j). \tag{41}$$

Then $\boldsymbol{Z}(j)$ can be determined as

$$\begin{aligned}\boldsymbol{Z}(j)=&\boldsymbol{\Phi}_{\bar{\boldsymbol{A}}(j+k_0-1)}(j,1)\boldsymbol{Z}(1)\\&+\sum_{i=1}^{j-1}\boldsymbol{\Phi}_{\bar{\boldsymbol{A}}(j+k_0-1)}(j,i+1)\boldsymbol{\Delta}(i+k_0-1)\boldsymbol{Z}(i).\end{aligned} \tag{42}$$

Taking the Frobenius norm on both sides of Eq. (42), we have

$$\begin{aligned}&\|\boldsymbol{Z}(j)\|\le\left\|\boldsymbol{\Phi}_{\bar{\boldsymbol{A}}(j+k_0-1)}(j,1)\right\|\|\boldsymbol{Z}(1)\|\\&+\sum_{i=1}^{j}\left\|\boldsymbol{\Phi}_{\bar{\boldsymbol{A}}(j+k_0-1)}(j,i+1)\right\|\|\boldsymbol{\Delta}(i+k_0-1)\|\|\boldsymbol{Z}(i)\|\\&\le\gamma(k_0)\lambda^{j-1}\|\boldsymbol{Z}(1)\|+\sum_{i=1}^{j-1}\gamma(k_0)\\&\times\lambda^{j-i-1}\|\boldsymbol{\Delta}(i+k_0-1)\|\|\boldsymbol{Z}(i)\|.\end{aligned} \tag{43}$$

Let $\boldsymbol{T}(j)=\boldsymbol{Z}(j)\lambda^{-(j+1)}$. Inequality (43) can be rewritten as

$$\begin{aligned}\|\boldsymbol{T}(j)\|\le&\gamma(k_0)\|\boldsymbol{T}(1)\|\\&+\sum_{i=1}^{j-1}\frac{\gamma(k_0)}{\lambda}\|\boldsymbol{\Delta}(i+k_0-1)\|\|\boldsymbol{T}(i)\|.\end{aligned} \tag{44}$$

Employing the Gronwall inequality, inequality (44) gives

$$\begin{aligned}\|\boldsymbol{T}(j)\|\le&\gamma(k_0)\|\boldsymbol{T}(1)\|\\&\times\prod_{i=1}^{j-1}\left(1+\frac{\gamma(k_0)}{\lambda}\|\boldsymbol{\Delta}(i+k_0-1)\|\right).\end{aligned} \tag{45}$$

Taking logarithms on both sides of inequality (45) yields

$$\begin{aligned}\ln(\|\boldsymbol{T}(j)\|)\le&\ln(\gamma(k_0)\|\boldsymbol{T}(1)\|)\\&+\sum_{i=1}^{j-1}\ln\left(1+\frac{\gamma(k_0)}{\lambda}\|\boldsymbol{\Delta}(i+k_0-1)\|\right).\end{aligned} \tag{46}$$

Note that $\ln(1+x)\le x$ holds for any $x\ge0$, and from inequality (46), we can further obtain

$$\|\boldsymbol{T}(j)\|\le\gamma(k_0)\|\boldsymbol{T}(1)\|\exp\left(\sum_{i=1}^{j-1}\frac{\gamma(k_0)}{\lambda}\|\boldsymbol{\Delta}(i+k_0-1)\|\right). \tag{47}$$

Substituting $\boldsymbol{T}(j)=\boldsymbol{Z}(j)\lambda^{-(j+1)}$ into inequality (47) yields

$$\begin{aligned}\|\boldsymbol{Z}(j)\|\le&\gamma\lambda^{j-1}\|\boldsymbol{Z}(1)\|\\&\times\exp\left(\sum_{i=1}^{j-1}\frac{\gamma(k_0)}{\lambda}\|\boldsymbol{\Delta}(i+k_0-1)\|\right).\end{aligned} \tag{48}$$

Let $\boldsymbol{\Phi}_{\bar{\boldsymbol{A}}(j)+\boldsymbol{\Delta}(j)}(j,1)$ be the transition matrix of Eq. (39). Clearly,

$$\begin{aligned}&\left\|\boldsymbol{\Phi}_{\bar{\boldsymbol{A}}(j)+\boldsymbol{\Delta}(j)}(j,1)\right\|\\=&\left\|\boldsymbol{\Phi}_{\bar{\boldsymbol{A}}(j)+\boldsymbol{\Delta}(j)}(j,1)\boldsymbol{I}_{n(m_1+m_2)}\right\|\\=&\left\|\left[\boldsymbol{\Phi}_{\bar{\boldsymbol{A}}(j)+\boldsymbol{\Delta}(j)}(j,1)\boldsymbol{e}_1,\boldsymbol{\Phi}_{\bar{\boldsymbol{A}}(j)+\boldsymbol{\Delta}(j)}(j,1)\boldsymbol{e}_2,\ldots,\right.\right.\\&\left.\left.\boldsymbol{\Phi}_{\bar{\boldsymbol{A}}(j)+\boldsymbol{\Delta}(j)}(j,1)\boldsymbol{e}_{n(m_1+m_2)}\right]\right\|,\end{aligned} \tag{49}$$

where $\boldsymbol{e}_k$ $(k=1,2,\ldots,n(m_1+m_2))$ is the k^{th} column of $\boldsymbol{I}_{n(m_1+m_2)}$. According to inequality (48), we obtain

$$\begin{aligned}&\left\|\boldsymbol{\Phi}_{\bar{\boldsymbol{A}}(j)+\boldsymbol{\Delta}(j)}(j,1)\right\|\\\le&\sqrt{n(m_1+m_2)}\gamma(k_0)\lambda^{j-1}\\&\times\exp\left(\sum_{i=1}^{j-1}\frac{\gamma(k_0)}{\lambda}\|\boldsymbol{\Delta}(i+k_0-1)\|\right)\\\le&\sqrt{n(m_1+m_2)}\gamma(k_0)\lambda^{j-1}\exp\left(\frac{\gamma(k_0)}{\lambda}\beta\theta(j-1)\right)\\=&\sqrt{n(m_1+m_2)}\gamma(k_0)\left(\exp\left(\frac{\gamma(k_0)}{\lambda}\beta\theta+\ln\lambda\right)\right)^{j-1}.\end{aligned} \tag{50}$$

Let the learning rate β be selected satisfying inequality (34). Then we have $\frac{\gamma(k_0)}{\lambda}\beta\theta + \ln\lambda < 0$, which means that system (39) is also exponentially stable. Therefore, $(\hat{\boldsymbol{K}}_{1,l+1}(j), \hat{\boldsymbol{K}}_{2,l+1}(j))$ will converge to $(\bar{\boldsymbol{K}}_1^{l+1}, \bar{\boldsymbol{K}}_2^{l+1})$ exponentially. If, further, $\bar{\boldsymbol{\sigma}}_k$ is persistently exciting, $\hat{\boldsymbol{W}}_{\mathrm{c},l+1}(i)$ will converge to $\boldsymbol{W}_{\mathrm{c},l+1}$. Using the result from Theorem 1, we know that $((\bar{\boldsymbol{K}}_1^{l+1})^{\mathrm{T}}, (\bar{\boldsymbol{K}}_2^{l+1})^{\mathrm{T}})$ will converge to the saddle feedback gains $(-\boldsymbol{K}_1^*, -\boldsymbol{K}_2^*)$, if $\bar{\boldsymbol{\sigma}}_k$ and $\bar{\boldsymbol{x}}_k$ are persistently exciting and the learning rate β satisfies inequality (34). The proof is completed.

5 Simulation study

In this section, we will use Algorithm 2 to design an H_∞ load-frequency controller for an electrical power system generator that suffers from load disturbance.

Consider the following fourth-order discrete-time electrical power system:

$$\boldsymbol{x}_{k+1} = \boldsymbol{A}\boldsymbol{x}_k + \boldsymbol{B}u_k + \boldsymbol{D}w_k, \tag{51}$$

where $\boldsymbol{x}_k = [x_{k1},\ x_{k2},\ x_{k3},\ x_{k4}]^{\mathrm{T}}$ (x_{k1} denotes the incremental frequency deviation, x_{k2} the incremental change in generator output, x_{k3} the incremental change in governor position, and x_{k4} the incremental change in integral control), and w_k is the load disturbance. The initial state is set to be $\boldsymbol{x}_0 = [4,\ 3,\ -1.5,\ 2.5]^{\mathrm{T}}$. The system matrices are given as follows:

$$\boldsymbol{A} = \begin{bmatrix} 0.9704 & 0.6629 & 0.0849 & -0.0446 \\ -0.0762 & 0.6724 & 0.1584 & -0.1462 \\ -0.3954 & -0.1663 & 0.2367 & -0.7403 \\ 0.0594 & 0.0212 & 0.0019 & 0.9993 \end{bmatrix},$$

$$\boldsymbol{B} = \begin{bmatrix} 0.0446 \\ 0.1462 \\ 0.7403 \\ 0.0007 \end{bmatrix},\ \boldsymbol{D} = \begin{bmatrix} -0.7924 \\ 0.0230 \\ 0.1893 \\ -0.0239 \end{bmatrix}.$$

The value function is given by

$$V(\boldsymbol{x}_k) = \sum_{i=k}^{\infty} (\boldsymbol{x}_i^{\mathrm{T}}\boldsymbol{x}_i + u_i^{\mathrm{T}}u_i - 9w_i^{\mathrm{T}}w_i);$$

that is, the disturbance attenuation level is set to be $\gamma = 3$.

Employing Algorithm 1, we obtain the optimal feedback gains for the controller and the disturbance:

$$\boldsymbol{K}_1^* = [2.1739, 3.7564, 0.8402, 1.4448],$$
$$\boldsymbol{K}_2^* = [1.2517, 1.6753, 0.2920, 1.1466].$$

Now we apply the minimax Q-learning method developed in Section 4 to solve for the H_∞ load-frequency controller. Note that the system matrices $\boldsymbol{A}$, $\boldsymbol{B}$, and $\boldsymbol{D}$ are not needed to design the controller. They are used only to simulate the system. The initial admissible feedback gains are selected as $\boldsymbol{K}_1^1 = \boldsymbol{K}_2^1 = [0,\ 0,\ 0,\ 0]$. The learning rate is selected as $\beta = 0.1$. The threshold to stop the algorithm is set to be $\varepsilon = 10^{-4}$. The state samples used for RLS tuning at each policy evaluation step are generated by the behavior policies $u_k^1 = 0.5\sum_{i=1}^{50}\sin(\omega_i k)$ and $u_k^2 = 0.6\sum_{i=1}^{50}\sin(\omega_i k)$. The state samples used for gradient tuning at each policy improvement step are generated by the behavior policies $u_k^3 = 0.2\sum_{i=1}^{50}\sin(\omega_i k)$ and $u_k^4 = 0.15\sum_{i=1}^{50}\sin(\omega_i k)$, where ω_i is an integer randomly generated from $[-50,\ 50]$. In fact, one can choose from a variety of behavior policies as long as the behavior policies are such that both $\bar{\boldsymbol{\sigma}}_k$ and $\bar{\boldsymbol{x}}_k$ are persistently exciting. At each policy evaluation step, we carry out 5000 tuning steps, while 3000 tuning steps are carried out at each policy improvement step. After eight iterations, convergence of Algorithm 2 is attained, and the convergent values are given as follows:

$$\bar{\boldsymbol{K}}_1^8 = [-2.1729, -3.7555, -0.8398, -1.4445]^{\mathrm{T}},$$
$$\bar{\boldsymbol{K}}_2^8 = [-1.2507, -1.6746, -0.2916, -1.1451]^{\mathrm{T}}.$$

Clearly, $((\bar{\boldsymbol{K}}_1^8)^{\mathrm{T}}, (\bar{\boldsymbol{K}}_2^8)^{\mathrm{T}})$ is very close to the saddle feedback gains $(-\boldsymbol{K}_1^*, -\boldsymbol{K}_2^*)$. Figs. 1 and 2 show the evolution of $\bar{\boldsymbol{K}}_1^l$ and $\bar{\boldsymbol{K}}_2^l$, respectively. Fig. 3 indicates that the obtained H_∞ load-frequency controller $u_k = (\bar{\boldsymbol{K}}_1^8)^{\mathrm{T}}\boldsymbol{x}_k$ stabilizes system (51) under disturbance $w_k = 5\exp(-0.16k)$. Fig. 4 indicates that the obtained H_∞ load-frequency controller $u_k = (\bar{\boldsymbol{K}}_1^8)^{\mathrm{T}}\boldsymbol{x}_k$ still stabilizes system (51) even under the worst-case disturbance $w_k = (\bar{\boldsymbol{K}}_2^8)^{\mathrm{T}}\boldsymbol{x}_k$.

For comparison, we apply the value-iteration-based Q-learning method (Al-Tamimi et al., 2007; Rizvi and Lin, 2018; Valadbeigi et al., 2020) to solve for the H_∞ load-frequency controller. The initial value matrix is chosen as $\boldsymbol{H} = \boldsymbol{I}_{6\times 6}$, and the initial feedback gains for the controller and the disturbance are selected as $\boldsymbol{K}_1^1 = [0,\ 0,\ 0,\ 0]$ and $\boldsymbol{K}_2^1 = [0,\ 0,\ 0,\ 0]$, respectively. After 50 iterations, convergence of the value-iteration-based Q-learning method is attained, with the convergent values given as follows:

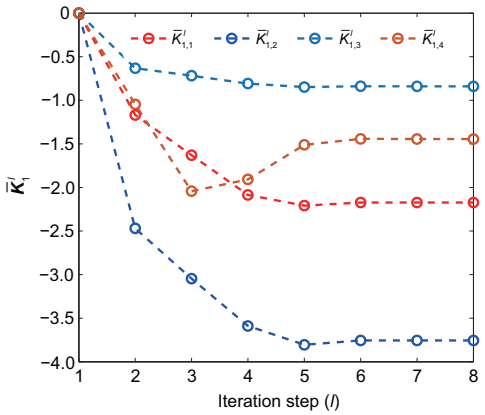

Fig. 1 Evolution of the controller feedback gain $\bar{K}_1^l$ in the policy-iteration-based minimax Q-learning method, where $\bar{K}_1^l = [\bar{K}_{1,1}^l, \bar{K}_{1,2}^l, \bar{K}_{1,3}^l, \bar{K}_{1,4}^l]^{\mathrm{T}}$ (References to color refer to the online version of this figure)

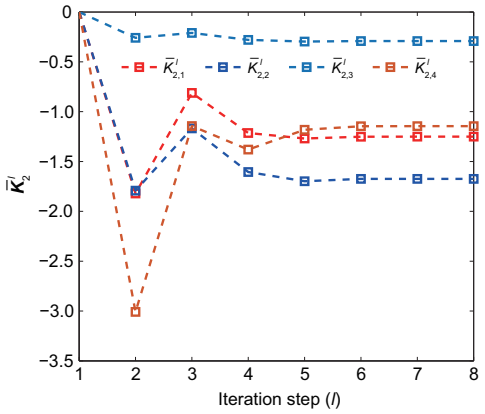

Fig. 2 Evolution of the disturbance feedback gain $\bar{K}_2^l$ in the policy-iteration-based minimax Q-learning method, where $\bar{K}_2^l = [\bar{K}_{2,1}^l, \bar{K}_{2,2}^l, \bar{K}_{2,3}^l, \bar{K}_{2,4}^l]^{\mathrm{T}}$ (References to color refer to the online version of this figure)

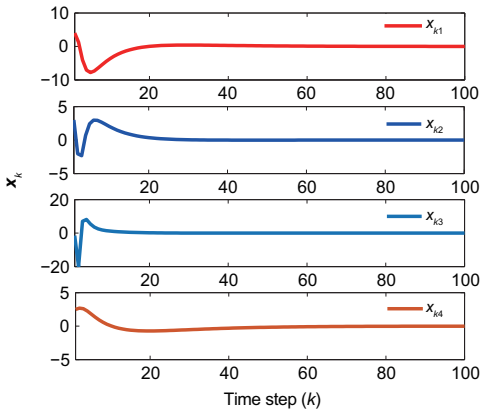

Fig. 3 State evolution of system (51) by implementing $u_k = (\bar{K}_1^8)^{\mathrm{T}} x_k$ under disturbance $w_k = 5\exp(-0.16k)$

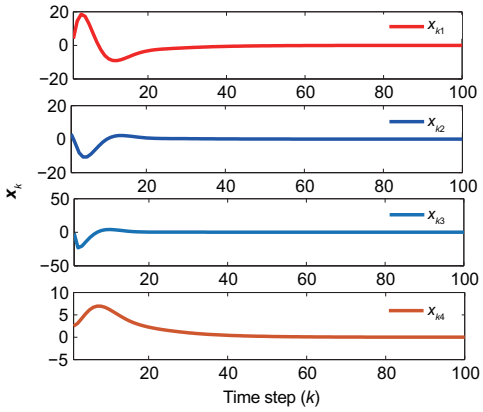

Fig. 4 State evolution of system (51) by implementing $u_k = (\bar{K}_1^8)^{\mathrm{T}} x_k$ under the worst-case disturbance $w_k = (\bar{K}_2^8)^{\mathrm{T}} x_k$

$$\boldsymbol{K}_1^{50} = [-2.1729, -3.7550, -0.8399, -1.4427],$$
$$\boldsymbol{K}_2^{50} = [-1.2512, -1.6746, -0.2919, -1.1456].$$

Clearly, $(\boldsymbol{K}_1^{50}, \boldsymbol{K}_2^{50})$ is also close to the saddle feedback gains $(-\boldsymbol{K}_1^*, -\boldsymbol{K}_2^*)$. Figs. 5 and 6 show the convergence of $\boldsymbol{K}_1^l$ and $\boldsymbol{K}_2^l$, respectively. It is observed that both the policy-iteration-based minimax Q-learning method and the value-iteration-based Q-learning method converge to the saddle solution. Obviously, compared with the value-iteration-based Q-learning method, it takes far fewer steps of iteration for the policy-iteration-based minimax Q-learning method to converge.

6 Conclusions

The H_∞ control problem for linear discrete-time systems has been investigated in this paper. A policy-iteration-based minimax Q-learning method has been developed to learn the H_∞ controller online by using the state samples generated by the behavior policies, without querying the system model. By employing a normalized gradient method, a novel policy improvement scheme has been proposed. The rigorous convergence analysis of the proposed minimax Q-learning method has been established under some persistence of excitation conditions and learning rate constraints. In addition, the excitation noise bias problem has been overcome. The simulation results demonstrated the good disturbance rejection capacity of the obtained H_∞ controller. In future work, we will explore Q-learning approaches for H_∞ control of nonlinear discrete-time systems.

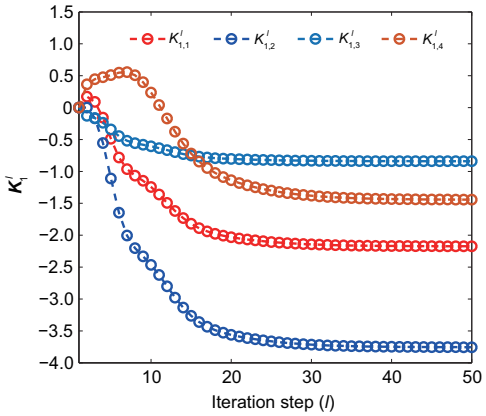

Fig. 5 Evolution of the controller feedback gain K_1^l in the value-iteration-based Q-learning method, where $K_1^l=[K_{1,1}^l,K_{1,2}^l,K_{1,3}^l,K_{1,4}^l]$ (References to color refer to the online version of this figure)

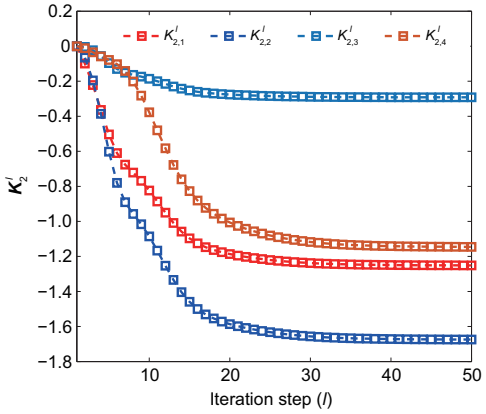

Fig. 6 Evolution of the disturbance feedback gain K_2^l in the value-iteration-based Q-learning method, where $K_2^l=[K_{2,1}^l,K_{2,2}^l,K_{2,3}^l,K_{2,4}^l]$ (References to color refer to the online version of this figure)

Contributors

Xinxing LI and Lele XI designed the research, conducted the investigation, and drafted the paper. Wenzhong ZHA and Zhihong PENG supervised the research, helped organize the paper, and revised and finalized the paper.

Compliance with ethics guidelines

Xinxing LI, Lele XI, Wenzhong ZHA, and Zhihong PENG declare that they have no conflict of interest.

References

Al-Tamimi A, Lewis FL, Abu-Khalaf M, 2007. Model-free Q-learning designs for linear discrete-time zero-sum games with application to H-infinity control. *Automatica*, 43(3):473-481. https://doi.org/10.1016/j.automatica.2006.09.019

Başar T, Bernhard P, 1995. H$^\infty$-Optimal Control and Related Minimax Design Problems (2nd Ed.). Springer, Boston, USA.

Doyle JC, Glover K, Khargonekar PP, et al., 1989. State-space solutions to standard H_2 and H_∞ control problems. *IEEE Trans Autom Contr*, 34(8):831-847. https://doi.org/10.1109/9.29425

Hansen TD, Miltersen PB, Zwick U, 2003. Strategy iteration is strongly polynomial for 2-player turn-based stochastic games with a constant discount factor. *J ACM*, 60(1): Article 1. https://doi.org/10.1145/2432622.2432623

He HB, Zhong XN, 2018. Learning without external reward. *IEEE Comput Intell Mag*, 13(3):48-54. https://doi.org/10.1109/MCI.2018.2840727

Ioannou PA, Fidan B, 2006. Adaptive Control Tutorial. SIAM, Philadelphia, USA.

Kiumarsi B, Lewis FL, Jiang ZP, 2017. H$_\infty$ control of linear discrete-time systems: off-policy reinforcement learning. *Automatica*, 78:144-152. https://doi.org/10.1016/j.automatica.2016.12.009

Kiumarsi B, Vamvoudakis KG, Modares H, et al., 2018. Optimal and autonomous control using reinforcement learning: a survey. *IEEE Trans Neur Netw Learn Syst*, 29(6):2042-2062. https://doi.org/10.1109/TNNLS.2017.2773458

Li HR, Zhang QC, Zhao DB, 2020. Deep reinforcement learning-based automatic exploration for navigation in unknown environment. *IEEE Trans Neur Netw Learn Syst*, 31(6):2064-2076. https://doi.org/10.1109/TNNLS.2019.2927869

Li XX, Peng ZH, Jiao L, et al., 2019. Online adaptive Q-learning method for fully cooperative linear quadratic dynamic games. *Inform Sci*, 62:222201. https://doi.org/10.1007/s11432-018-9865-9

Littman ML, 2001. Value-function reinforcement learning in Markov games. *Cogn Syst Res*, 2(1):55-66. https://doi.org/10.1016/S1389-0417(01)00015-8

Luo B, Wu HN, Huang TW, 2015. Off-policy reinforcement learning for H_∞ control design. *IEEE Trans Cybern*, 45(1):65-76. https://doi.org/10.1109/TCYB.2014.2319577

Luo B, Yang Y, Liu DR, 2018. Adaptive Q-learning for data-based optimal output regulation with experience replay. *IEEE Trans Cybern*, 48(12):3337-3348. https://doi.org/10.1109/TCYB.2018.2821369

Luo B, Yang Y, Liu DR, 2021. Policy iteration Q-learning for data-based two-player zero-sum game of linear discrete-time systems. *IEEE Trans Cybern*, 51(7):3630-3640. https://doi.org/10.1109/TCYB.2020.2970969

Mehraeen S, Dierks T, Jagannathan S, et al., 2013. Zero-sum two-player game theoretic formulation of affine nonlinear discrete-time systems using neural networks. *IEEE Trans Cybern*, 43(6):1641-1655. https://doi.org/10.1109/TSMCB.2012.2227253

Modares H, Lewis FL, Jiang ZP, 2015. H_∞ tracking control of completely unknown continuous-time systems via off-policy reinforcement learning. *IEEE Trans Neur Netw Learn Syst*, 26(10):2550-2562. https://doi.org/10.1109/TNNLS.2015.2441749

Prokhorov DV, Wunsch DC, 1997. Adaptive critic designs. *IEEE Trans Neur Netw*, 8(5):997-1007. https://doi.org/10.1109/72.623201

Rizvi SAA, Lin ZL, 2018. Output feedback Q-learning for discrete-time linear zero-sum games with application to the H-infinity control. *Automatica*, 95:213-221. https://doi.org/10.1016/j.automatica.2018.05.027

Sakamoto N, van der Schaft AJ, 2008. Analytical approximation methods for the stabilizing solution of the Hamilton–Jacobi equation. *IEEE Trans Autom Contr*, 53(10):2335-2350. https://doi.org/10.1109/TAC.2008.2006113

Sutton RS, Barto AG, 1998. Reinforcement Learning: an Introduction. MIT Press, Cambridge, USA.

Valadbeigi AP, Sedigh AK, Lewis FL, 2020. H_∞ static output-feedback control design for discrete-time systems using reinforcement learning. *IEEE Trans Neur Netw Learn Syst*, 31(2):396-406. https://doi.org/10.1109/TNNLS.2019.2901889

Vamvoudakis KG, Modares H, Kiumarsi B, et al., 2017. Game theory-based control system algorithms with real-time reinforcement learning: how to solve multiplayer games online. *IEEE Contr Syst Mag*, 37(1):33-52. https://doi.org/10.1109/MCS.2016.2621461

Watkins CJCH, Dayan P, 1992. *Q*-learning. *Mach Learn*, 8(3):279-292. https://doi.org/10.1007/BF00992698

Wei QL, Lewis FL, Sun QY, et al., 2017. Discrete-time deterministic *Q*-learning: a novel convergence analysis. *IEEE Trans Cybern*, 47(5):1224-1237. https://doi.org/10.1109/TCYB.2016.2542923

Wei YF, Wang ZY, Guo D, et al., 2019. Deep Q-learning based computation offloading strategy for mobile edge computing. *Comput Mater Contin*, 59(1):89-104. https://doi.org/10.32604/cmc.2019.04836

Yan HS, Zhang JJ, Sun QM, 2019. MTN optimal control of SISO nonlinear time-varying discrete-time systems for tracking by output feedback. *Intell Autom Soft Comput*, 25(3):487-507.

Zhang HG, Qin CB, Jiang B, et al., 2014. Online adaptive policy learning algorithm for H_∞ state feedback control of unknown affine nonlinear discrete-time systems. *IEEE Trans Cybern*, 44(12):2706-2718. https://doi.org/10.1109/TCYB.2014.2313915

Zhong XN, He HB, Wang D, et al., 2018. Model-free adaptive control for unknown nonlinear zero-sum differential game. *IEEE Trans Cybern*, 48(5):1633-1646. https://doi.org/10.1109/TCYB.2017.2712617

Zhu YH, Zhao DB, Li XJ, 2017. Iterative adaptive dynamic programming for solving unknown nonlinear zero-sum game based on online data. *IEEE Trans Neur Netw Learn Syst*, 28(3):714-725. https://doi.org/10.1109/TNNLS.2016.2561300