SCIENCE CHINA
Information Sciences

• RESEARCH PAPER •

June 2026, Vol. 69, Iss. 6, 162202:1–162202:16
https://doi.org/10.1007/s11432-025-4737-8

Adaptive optimal control of discrete-time systems by reinforcement learning: damping coefficients-based stabilizing policy iterations

Dongdong LI[1,2,3] & Jiuxiang DONG[1,2,3*]

[1]*College of Information Science and Engineering, Northeastern University, Shenyang 110819, China*
[2]*State Key Laboratory of Synthetical Automation of Process Industries, Northeastern University, Shenyang 110819, China*
[3]*Key Laboratory of Vibration and Control of Aero-propulsion System (Northeastern University), Ministry of Education, Shenyang 110819, China*

Received 23 April 2025/Revised 24 August 2025/Accepted 17 December 2025/Published online 11 February 2026

Abstract Policy iteration is one of the classical frameworks of adaptive dynamic programming, which requires a known initial stabilizing control to start the iteration. To relax this requirement, two different stabilizing policy iteration algorithms based on variable damping coefficients are designed for unknown discrete-time linear systems. First, we design a stabilizing artificial system and then iterate it gradually to the original system by cumulating damping coefficients and thus obtaining a stabilizing control policy. Then, a data-driven version of the stabilizing policy iteration framework is designed, and the corresponding model-free scheme is proposed for determining the damping coefficients. To relax the same initial condition that exists in traditional policy iteration-based $\mathcal{Q}$-learning, another novel data-driven $\mathcal{Q}$-learning algorithm based on stabilizing policy iteration is developed. The proposed $\mathcal{Q}$-learning algorithm is equivalent to the stabilizing policy iteration framework by theoretical analysis. Ultimately, the effectiveness of the two proposed algorithms is verified by a numerical example.

Keywords reinforcement learning, stabilizing policy iteration, $\mathcal{Q}$-learning, adaptive dynamic programming, optimal control

Citation Li D D, Dong J X. Adaptive optimal control of discrete-time systems by reinforcement learning: damping coefficients-based stabilizing policy iterations. Sci China Inf Sci, 2026, 69(6): 162202, https://doi.org/10.1007/s11432-025-4737-8

1 Introduction

Reinforcement learning (RL) makes optimal decisions by learning and interaction between the agent's actions and the external environment [1,2], which is often also known as adaptive dynamic programming (ADP). Unlike dynamic programming, the "curse of dimensionality" and the physical model unknown can be overcome [3, 4]. This allows ADP widely used for optimal control or optimal decision making without accurate system model, including discrete-time (DT) systems [2, 5–11], continuous-time (CT) systems [12–15], and practical applications [4, 16]. Most ADP methods are designed by two basic iteration frameworks, i.e., policy iteration (PI) [2, 5, 7, 12, 17, 18] and value iteration (VI) [6, 14, 15, 19–21]. Initially, Kleinman algorithms for CT linear systems [17] and DT linear systems [5] are proposed, respectively, which are the basis for PI. In [12], the data-driven optimal control method based on PI for CT linear systems is proposed, and a rank condition that guarantees that the iterative equations can be solved uniquely is analyzed. It is extended in [13] to the model-free optimal output regulation (OOR) problem. The model-free approach and its analysis in [12] can be extended to DT systems. For example, the DT version of the model-free OOR is given in [7]. A model-free robust OOR algorithm is proposed in [2] based on PI and input-output data for DT linear systems. Similarly, various VI-based RL algorithms have also been developed. A model-free VI algorithm is proposed in [14] for CT linear systems, and this approach is extended in [21] to the H_∞ OOR problem for multiagent systems. The methods, including but not limited to the above, are classical model-free RL algorithms based on PI and VI. Moreover, the $\mathcal{Q}$-learning techniques are also developed based on VI and PI.

$\mathcal{Q}$-learning is the classical technique in RL, which was developed to address optimal control problems [10, 22–25]. $\mathcal{Q}$-learning algorithms are usually designed via an action-valued function that is defined according to the Bellman equation, often referred to as the $\mathcal{Q}$-function. PI-based and VI-based $\mathcal{Q}$-learning algorithms for DT linear systems were proposed in [10,22,26], respectively. In [23,24], $\mathcal{Q}$-learning algorithms of CT linear systems have been developed.

* Corresponding author (email: dongjiuxiang@ise.neu.edu.cn)

Subsequently, a VI-based output feedback $\mathcal{Q}$-learning algorithm is proposed in [25] for DT linear systems. In [27], two output-feedback $\mathcal{Q}$-learning methods are developed based on PI and VI for DT systems. Clearly, these $\mathcal{Q}$-learning methods retain the characteristics of PI or VI.

The VI method begins with an easily determined value function with an arbitrary bounded control policy. However, since the iterative process of VI is non-monotonic, it converges more slowly than PI [19]. In each iteration, the PI algorithm obtains the real-valued function of the target control policy based on the known stabilizing control policy, which in turn updates the control policy [19, 28]. Although the PI method converges faster, it must start with a known stabilizing control policy [19, 29]. In general, it is always a challenge to obtain a stabilizing control policy for unknown systems. To address the problem, some relevant results are presented. A multi-step heuristic dynamic programming is proposed in [28] to realize the trade-off between PI and VI. Subsequently, the PI and VI are combined by varying the coefficients in [30]. With further research, a bias-PI algorithm that can relax the initial stabilizing control condition by adding a bias parameter is developed in [31] for CT systems. λ-PI algorithm is developed in [32] for DT systems, which combines VI and PI by designing the Bellman operator λ. The method can be considered as PI if the Bellman operator tends to 1 and VI if the Bellman operator tends to 0. Recently, a class of hybrid iteration methods was proposed in [33–35], which obtains a stabilizing control policy through VI, then converges quickly to the optimal solution via switching to PI. This method is extended to solve the optimal control problem of unknown fast-sampling singularly perturbed systems [36]. The methods such as bias-PI, λ-PI and hybrid iteration effectively address the limitations of the initial stabilizing control policy, but all contain some characteristics of VI, especially, their convergence rate is also affected by the slow convergence of VI. In [29, 37], the homotopy-based PI methods for linear and nonlinear CT systems are proposed, which combine the homotopy method with PI to seek a stabilizing control for the unknown systems. These methods do not rely on VI to find the initial stabilizing control and converge faster than the traditional VI. However, the method cannot be directly generalized to the DT system. A method for solving the stabilizing control of a DT linear system is given in [38] by using a discount factor, but its the updating rule and the iteration step-size need to be further designed. In [39], a scaling PI algorithm is proposed for finding stabilizing control policies for DT systems, but it is not further extended to $\mathcal{Q}$-learning methods. Note that the initial policy condition also exists in traditional PI-based $\mathcal{Q}$-learning [8, 10, 22, 27]. In [40], a deadbeat control gain matrix is designed to initialize the proposed efficient off-policy $\mathcal{Q}$-learning algorithm. This initialization design is more complex and does not pre-set any desired upper bound on the spectral radius. Therefore, it is important to develop a novel framework such that the following requirements are satisfied: (1) it can obtain a stabilizing policy and can replace the pole-placement method; (2) it does not lose the fast convergence feature of PI; (3) it can be designed as various data-driven algorithms and achieve optimal control.

Inspired by the above results, we propose two different varying damping coefficients-based data-driven PI algorithms. The following are the main contributions.

(1) By designing dynamically adjusted damping coefficients, a novel stabilizing policy iteration (SPI) framework is proposed and its stability analysis is given. Unlike the traditional PI [2, 5, 7–12, 27, 41, 42], this SPI, based on varying damping coefficients, can adaptively calculate the stabilizing control policy, which relaxes the requirement for an initial stabilizing policy in the existing PI. Moreover, the SPI framework is characterized by the fast convergence of the PI and can be extended to solving various optimal control problems [2, 5, 7–11, 27, 41, 42].

(2) A data-driven version of the SPI framework is designed, and the corresponding model-free schemes are proposed for determining the initial damping coefficient and updating the explicit step-size. We only need to use the same data condition as the data-driven VI [14, 26] to obtain a stabilizing control policy and an optimal control policy, respectively.

(3) Different from [32, 38, 39], we introduce cumulative sum damping coefficients to find the stabilizing control policy and another data-driven $\mathcal{Q}$-learning algorithm based on the SPI framework is further designed to address the limitation of initial an stabilizing policy that also exists in traditional PI-based $\mathcal{Q}$-learning [8, 10, 22, 27]. The stability of this $\mathcal{Q}$-learning algorithm and its equivalence to data-driven SPI are verified by theoretical analysis and simulation.

(4) More importantly, a simple extension of the proposed framework and the corresponding data-driven algorithm can regulate any initial closed-loop spectral radius to within the range of the desired spectral radius. Thus, the method can replace the classical model-based pole-placement method.

Section 2 describes the optimal control problem for DT linear systems and introduces the classical PI. Section 3 designs an SPI framework based on varying damping coefficients that provides a scheme for solving the stabilizing control policy when the parameters of the concerned system model are given. In Section 4, an online SPI-based data-driven algorithm is proposed and the model-free method for selecting damping coefficients is designed. In Section 5, the SPI-based $\mathcal{Q}$-learning algorithm is presented and its convergence and equivalence with SPI are analyzed. In

Section 6, simulations are provided to validate the two proposed algorithms, respectively. Section 7 concludes this work.

Notations. For any matrix "$\star$", $\sigma_{\max}(\star)$ and $\sigma_{\min}(\star)$ indicate its maximal and minimal singular values, and $\rho(\star)$ is its spectral radius. For matrix $X = X^\top \in \mathbb{R}^{n\times n}$, $\mathrm{vecs}(X) = [X_{1,1}, 2X_{1,2}, \ldots, 2X_{1,n}, X_{2,2}, 2X_{2,3}, \ldots, 2X_{n-1,n}, X_{n,n}]^\top \in \mathbb{R}^{\frac{1}{2}n(n+1)}$, where $X_{i,j}$ is the ith row and jth column element of matrix X. If matrix $X \in \mathbb{R}^{n\times m}$, $\mathrm{vec}(X) = [X_1^\top, X_2^\top, \ldots, X_m^\top]^\top \in \mathbb{R}^{nm}$, where $X_i \in \mathbb{R}^n$ is ith column of matrix X for $i = 1, \ldots, m$. For vector $\chi \in \mathbb{R}^n$, $\mathrm{vecv}(\chi) = [\chi_1^2, \chi_1\chi_2, \ldots, \chi_1\chi_n, \chi_2^2, \chi_2\chi_3, \ldots, \chi_{n1}\chi_n, \chi_n^2]^\top \in \mathbb{R}^{\frac{n(n+1)}{2}}$. $\otimes$ is the Kronecker product. "$\top$" denotes the transpose.

2 Preliminaries

2.1 System and problem description

Consider a DT linear system

$$x(k+1) = Ax(k) + Bu(k), \tag{1}$$

where $x \in \mathbb{R}^{n_x}$ and $u \in \mathbb{R}^{n_u}$ are the state and input, respectively. $A \in \mathbb{R}^{n_x\times n_x}$ and $B \in \mathbb{R}^{n_x\times n_u}$ are unknown constant matrices.

Assumption 1. The pair (A, B) is stabilizable.

Problem. Under Assumption 1, the optimal control is to design a controller $u = -K^*x(k)$ such that system (1) is Schur stable, i.e., $\rho(A - BK^*) < 1$, where $K^* \in \mathbb{R}^{n_u\times n_x}$ is optimal control gain obtained by minimizing the performance function

$$V(x(k), u(k)) = \sum_{i=k}^{\infty} \big(x^\top(i)Qx(i) + u^\top(i)Ru(i)\big), \tag{2}$$

where $Q = Q^\top > 0$ and $R = R^\top > 0$.

2.2 Traditional discrete-time PI

According to [1], the optimal controller is given as

$$u = -K^*x(k) = -(R + B^\top P^*B)^{-1}B^\top P^*Ax(k), \tag{3}$$

where $P^* = P^{*\top} > 0$ is the unique positive definite solution to the following algebraic Riccati equation (ARE):

$$A^\top PA - P - A^\top PB(R + B^\top PB)^{-1}B^\top PA = -Q. \tag{4}$$

To avoid solving the ARE (4) directly, the classical model-based PI method is given in Lemma 1.

Lemma 1 ([5])**.** Given any initial stabilizing control gain K^0 satisfying $\rho(A - BK^0) < 1$, for $i = 0, 1, 2, \ldots$, $P^i = (P^i)^\top$ is solved by the Lyapunov equation

$$(A - BK^i)^\top P^i(A - BK^i) - P^i + Q + (K^i)^\top RK^i = 0. \tag{5}$$

Update the policy by

$$K^{i+1} = (R + B^\top P^iB)^{-1}B^\top P^iA. \tag{6}$$

Then, for $i = 0, 1, 2, \ldots, \infty$, one has that (1) $\rho(A - BK^{i+1}) < 1$; (2) $P^* \leqslant P^{i+1} \leqslant P^i$; (3) $\lim_{i\to\infty} P^i = P^*$, $\lim_{i\to\infty} K^i = K^*$.

In Lemma 1, the initial stabilizing gain K^0 and the pair (A, B) must be known. To realize data-driven optimal control, it is critical to obtain an initial stabilizing control gain without using model information. If the initial stabilizing control is obtained, then the approximate optimal solution to the ARE (4) can be obtained by the traditional PI-based ADP algorithms such as [7, 10].

3 Varying damping coefficients-based SPI framework

First, to facilitate the establishment of a varying damping coefficients-based SPI framework to obtain stabilizing control, A and B are assumed to be known. In Sections 4 and 5, we remove all dependencies of this framework on model information to achieve data-driven control.

To distinguish the process from Lemma 1, define new matrices $\tilde{P}^j$ and $\tilde{K}^j$ for $j = 0, 1, 2, \ldots$, and a positive constant $\bar{\rho}$ as

$$\bar{\rho} = \frac{1}{\rho(A - B\tilde{K}^0)}, \tag{7}$$

where $\tilde{K}^0$ is an arbitrary bounded control gain. There exists a small constant β satisfying $\bar{\rho} > \beta > 0$ such that $\rho[(\bar{\rho} - \beta)(A - B\tilde{K}^0)] < 1$. This ensures that the artificial system $(\bar{\rho} - \beta)(A - B\tilde{K}^j)$ is stabilized, but it does not ensure that the original system (1) is stable. Theorem 1 provides a way to obtain stabilizing gains, which is achieved by iterating the stabilizing artificial system to the original closed-loop system.

Theorem 1. Under Assumption 1, given arbitrary bounded policy $\tilde{K}^0$, and constants $\bar{\rho} > \beta > \alpha_0 > 0$, define $\tilde{\beta} := \bar{\rho} - \beta$ and

$$\tilde{A}^j := \left(\tilde{\beta} + \sum_{m=0}^{j} \alpha_m\right)(A - B\tilde{K}^j), \quad W^j := Q + (\tilde{K}^j)^\top R\tilde{K}^j. \tag{8}$$

For $j = 0, 1, 2, \ldots$, $\tilde{P}^j$ is solved by

$$(\tilde{A}^j)^\top \tilde{P}^j(\tilde{A}^j) - \tilde{P}^j = -W^j. \tag{9}$$

Update $\tilde{K}^{j+1}$ and choose α_{j+1} by

$$\tilde{K}^{j+1} = \left(\tilde{\beta} + \sum_{m=0}^{j} \alpha_m\right)^2 \left[R + \left(\tilde{\beta} + \sum_{m=0}^{j} \alpha_m\right)^2 B^\top \tilde{P}^j B\right]^{-1} B^\top \tilde{P}^j A, \tag{10}$$

$$\alpha_{j+1} = c\left(\tilde{\beta} + \sum_{m=0}^{j} \alpha_m\right)\left(\sqrt{\frac{\sigma_{\min}(W^{j+1})}{\sigma_{\max}(\tilde{P}^j - W^{j+1})} + 1} - 1\right) := c\left(\tilde{\beta} + \sum_{m=0}^{j} \alpha_m\right)\bar{\alpha}_{j+1}, \quad c \in (0, 1]. \tag{11}$$

Then, one has that (1) $\rho(A - B\tilde{K}^j) < 1/(\tilde{\beta} + \sum_{m=0}^{j} \alpha_m)$; (2) α_{j+1} exists and is bounded.

Proof. We use *Induction* to prove (1). Since $\bar{\rho} > \beta > \alpha_0 > 0$ and (7) for $j = 0$, there is $\rho[(\tilde{\beta} + \alpha_0)(A - B\tilde{K}^0)] < \bar{\rho}\rho(A) = 1$. Conclusion (1) holds for $j = 0$. Viewing $(\tilde{\beta} + \alpha_0)A$ and $(\tilde{\beta} + \alpha_0)B$ as A and B in Lemma 1, there is $\rho[(\tilde{\beta} + \alpha_0)(A - B\tilde{K}^1)] < 1$. Thus, there is a unique positive definite solution $\tilde{\mathcal{P}}^1$ to

$$[(\tilde{\beta} + \alpha_0)(A - B\tilde{K}^1)]^\top \tilde{\mathcal{P}}^1[(\tilde{\beta} + \alpha_0)(A - B\tilde{K}^1)] - \tilde{\mathcal{P}}^1 = -W^1. \tag{12}$$

For the pair $((\tilde{\beta} + \alpha_0)A, (\tilde{\beta} + \alpha_0)B)$, $\tilde{\mathcal{P}}^1 \leqslant \tilde{P}^0$ can be obtained by Conclusion (2) of Lemma 1, thus $W^1 \leqslant \tilde{\mathcal{P}}^1 \leqslant \tilde{P}^0$. For $j = 0$, α_1 in (11) exists and is bounded, Conclusion (2) holds. From (12), we have

$$\begin{aligned}
(\tilde{A}^1)^\top \tilde{\mathcal{P}}^1 \tilde{A}^1 - \tilde{\mathcal{P}}^1 &= \left(\frac{\tilde{\beta} + \alpha_0 + \alpha_1}{\tilde{\beta} + \alpha_0}\right)^2 (\tilde{\mathcal{P}}^1 - W^1) - \tilde{\mathcal{P}}^1 \\
&= \left(\left(1 + \frac{\alpha_1}{\tilde{\beta} + \alpha_0}\right)^2 - 1\right)(\tilde{\mathcal{P}}^1 - W^1) - W^1 \\
&\overset{(b_1)}{\leqslant} \left(\left(1 + \frac{\alpha_1}{\tilde{\beta} + \alpha_0}\right)^2 - 1\right)(\tilde{P}^0 - W^1) - W^1 \overset{(b_2)}{\leqslant} \sigma_{\min}(W^1) I_{n_x} - W^1,
\end{aligned} \tag{13}$$

where (b_1) is obtained by using $\tilde{\mathcal{P}}^1 \leqslant \tilde{P}^0$ and (b_2) is obtained by using (11). Since $W^1 < \tilde{\mathcal{P}}^1 \leqslant \tilde{P}^0$, (b_1) is monotonically increasing on $(0, +\infty)$ with respect to α_1. Thus, all α_1 chosen according to (11) satisfy $(\tilde{A}^1)^\top \tilde{\mathcal{P}}^1 \tilde{A}^1 - \tilde{\mathcal{P}}^1 < 0$. We have that $\rho[(\tilde{\beta} + \alpha_0 + \alpha_1)(A - B\tilde{K}^1)] < 1$ and $\tilde{K}^1$ is a stabilizing control gain of system $\tilde{A}^1$. Conclusion (1) holds for $j = 1$. Repeating the above process easily yields that Conclusion (2) also holds for $j = 1$. Since $\rho[(\tilde{\beta} + \alpha_0 + \alpha_1)(A - B\tilde{K}^1)] < 1$, $\tilde{P}^1$ is the unique positive-definite solution to (9) for $j = 1$.

Assume that Conclusion (1) holds for $j = a$. Then, there are $\rho[(\tilde{\beta} + \sum_{m=0}^{a} \alpha_m)(A - B\tilde{K}^a)] < 1$ and $\tilde{P}^a = (\tilde{P}^a)^\top > 0$ that is solved by $(\tilde{A}^a)^\top \tilde{P}^a (\tilde{A}^a) - \tilde{P}^a = -W^a$. By using (10), we can obtain the gain matrix as $\tilde{K}^{a+1} = (\tilde{\beta} + \sum_{m=0}^{a} \alpha_m)^2 [R + (\tilde{\beta} + \sum_{m=0}^{a} \alpha_m)^2 B^\top \tilde{P}^a B]^{-1} B^\top \tilde{P}^a A$. If $(\tilde{\beta} + \sum_{m=0}^{a} \alpha_m)A$ and $(\tilde{\beta} + \sum_{m=0}^{a} \alpha_m)B$ are taken as A and B in Lemma 1, one obtains

$$\rho\left[\left(\tilde{\beta} + \sum_{m=0}^{a} \alpha_m\right)(A - B\tilde{K}^{a+1})\right] < 1. \tag{14}$$

From (14) and Lemma 1, it follows that there is a unique $\tilde{\mathcal{P}}^{a+1} = (\tilde{\mathcal{P}}^{a+1})^\top > 0$ satisfying

$$\left[\left(\tilde{\beta} + \sum_{m=0}^{a} \alpha_m\right)(A - B\tilde{K}^{a+1})\right]^\top \tilde{\mathcal{P}}^{a+1} \left[\left(\tilde{\beta} + \sum_{m=0}^{a} \alpha_m\right)(A - B\tilde{K}^{a+1})\right] - \tilde{\mathcal{P}}^{a+1} = -W^{a+1}. \tag{15}$$

Moreover, viewing $(\tilde{\beta} + \sum_{m=0}^{a} \alpha_m)A$ and $(\tilde{\beta} + \sum_{m=0}^{a} \alpha_m)B$ as A and B in Lemma 1 with the stabilizing gain $\tilde{K}^{a+1}$, we have $\tilde{\mathcal{P}}^{a+1} \leqslant \tilde{P}^a$ from Conclusion (2) of Lemma 1. Thus, there is $W^{a+1} < \tilde{\mathcal{P}}^{a+1} \leqslant \tilde{P}^a$. Then, there must exist a positive constant α_{a+1} satisfying

$$0 < \alpha_{a+1} = c\left(\tilde{\beta} + \sum_{m=0}^{a} \alpha_m\right)\left(\sqrt{\frac{\sigma_{\min}(W^{a+1})}{\sigma_{\max}(\tilde{P}^a - W^{a+1})} + 1} - 1\right). \tag{16}$$

Obviously, α_{a+1} is bounded. Similar to (13), we have

$$\begin{aligned}(\tilde{A}^{a+1})^\top \tilde{\mathcal{P}}^{a+1} \tilde{A}^{a+1} - \tilde{\mathcal{P}}^{a+1} &= \left(\left(1 + \frac{\alpha_{a+1}}{\tilde{\beta} + \sum_{m=0}^{a} \alpha_m}\right)^2 - 1\right)(\tilde{\mathcal{P}}^{a+1} - W^{a+1}) - W^{a+1} \\ &\overset{(b_3)}{\leqslant} \left(\left(1 + \frac{\alpha_{a+1}}{\tilde{\beta} + \sum_{m=0}^{a} \alpha_m}\right)^2 - 1\right)(\tilde{P}^a - W^{a+1}) - W^{a+1} \\ &\overset{(b_4)}{\leqslant} \sigma_{\min}(W^{a+1}) I_{n_x} - W^{a+1},\end{aligned} \tag{17}$$

where (b_3) is obtained by using $\tilde{\mathcal{P}}^{a+1} \leqslant \tilde{P}^a$ and (b_4) is obtained by using (11). Similarly, the right-hand side of (b_3) is a monotonically increasing function on $(0, +\infty)$ with respect to α_{a+1}. Therefore, all α_{a+1} chosen according to (11) satisfy $(\tilde{A}^{a+1})^\top \tilde{\mathcal{P}}^{a+1} \tilde{A}^{a+1} - \tilde{\mathcal{P}}^{a+1} < 0$. Then, we have $\rho(\tilde{A}^{a+1}) < 1$. By *Induction*, Conclusion (1) is established. By the above process, it is easy to obtain that Eq. (16) always holds, hence, Conclusion (2) also holds.

There always exist $0 < c\bar{\alpha}_j(\tilde{\beta} + \sum_{m=0}^{j-1} \alpha_m) = \alpha_j \leqslant \bar{\alpha}_j(\tilde{\beta} + \sum_{m=0}^{j-1} \alpha_m)$ and $c\bar{\alpha}_j > 0$ for $j = 0, 1, 2, \ldots$. Let $\chi := \min\{c\bar{\alpha}_m, m = 1, \ldots, j\}$. Then, we have $\tilde{\beta} + \sum_{m=0}^{j} \alpha_m = \alpha_j + (\tilde{\beta} + \sum_{m=0}^{j-1} \alpha_m) \geqslant (1 + \chi)(\tilde{\beta} + \sum_{m=0}^{j-1} \alpha_m) \geqslant (1 + \chi)^j(\tilde{\beta} + \alpha_0)$. Since exponential functions with base greater than 1 are monotonically increasing and infinite, there exists a positive integer $\bar{j}$ such that $\tilde{\beta} + \sum_{m=0}^{\bar{j}} \alpha_m \geqslant 1$ for $j \geqslant \bar{j}$. Making $(1 + \chi)^{\bar{j}}(\tilde{\beta} + \alpha_0) = 1$ solves the bound for $\bar{j}$, i.e., at most $\ln(1/(\tilde{\beta} + \alpha_0))/\ln(1 + \chi)$.

Remark 1. Figure 1 illustrates the significance of the proposed method and Theorem 1. As the process in Theorem 1 is iterated, the open-loop artificial system $(\tilde{\beta} + \sum_{m=0}^{j} \alpha_m)A$ gradually approaches the open-loop original system A, and the closed-loop spectral radius $\rho(A - B\tilde{K}^j)$ decreases gradually. From Conclusion (1) of Theorem 1, there is $\rho(A - B\tilde{K}^j) < 1/(\tilde{\beta} + \sum_{m=0}^{j} \alpha_m)$ during the iterations. If $\tilde{\beta} + \sum_{m=0}^{j} \alpha_m \geqslant 1$ is satisfied, then $\rho(A - B\tilde{K}^j) < 1$, i.e., $\tilde{K}^j$ is a stabilizing control gain. If a larger α_{j+1} is chosen to satisfy (11), the iteration will be faster. Moreover, it can be obtained that $1/(\tilde{\beta} + \sum_{m=0}^{j} \alpha_m)$ is a designed bound on the closed-loop spectral radius $\rho(A - B\tilde{K}^j)$ from Theorem 1.

The damping coefficient in our design is inspired by the concept of viscous damping in mechanical systems (e.g., dampers). Specifically, it acts as a dynamic cumulative sum term (from small to large) with the following roles. (1) By setting smaller initial coefficients, the original system $A - B\tilde{K}^0$ is compressed into the stabilizing virtual system $(\tilde{\beta} + \alpha_0)(A - B\tilde{K}^0)$. (2) After each iteration, the system $(\tilde{\beta} + \sum_{m=0}^{j} \alpha_m)(A - B\tilde{K}^{j+1})$ converges by updating the control policy $\tilde{K}^{j+1}$, but $\tilde{\beta} + \sum_{m=0}^{j+1} \alpha_m$ amplifies the system $(\tilde{\beta} + \sum_{m=0}^{j+1} \alpha_m)(A - B\tilde{K}^{j+1})$. It is worth noting that the process and the virtual closed-loop system always remain stable. (3) As $\tilde{K}^j$ improves and the damping

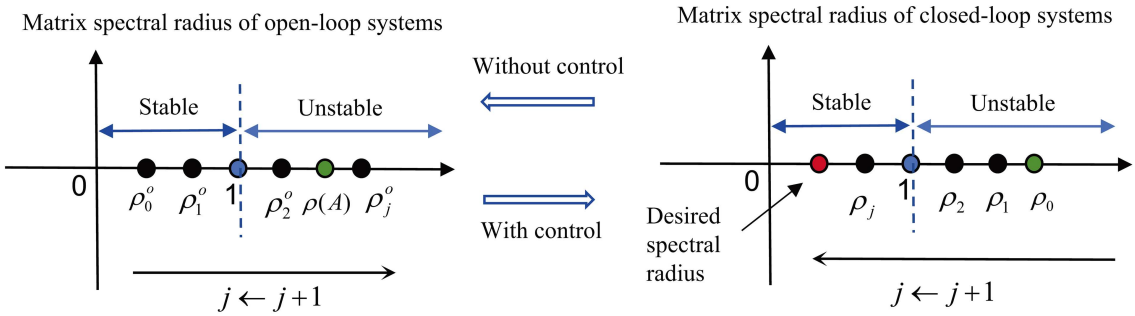

Figure 1 (Color online) Iteration process of Theorem 1, where $\rho_j^o = \rho[(\tilde{\beta} + \sum_{m=0}^{j} \alpha_m)A]$ denotes the spectral radius of the open-loop system $(\tilde{\beta} + \sum_{m=0}^{j} \alpha_m)A$, and $\rho_j = \rho(A - B\tilde{K}^j)$ denotes the spectral radius of the closed-loop system $A - B\tilde{K}^j$, respectively.

coefficients increase until $\tilde{\beta} + \sum_{m=0}^{j+1} \alpha_m \geqslant 1$, the stabilizing system $(\tilde{\beta} + \sum_{m=0}^{j} \alpha_m)(A - B\tilde{K}^j)$ is expanded into the system $A - B\tilde{K}^{j+1}$, thereby yielding the stabilizing control policy $\tilde{K}^{j+1}$.

In the above process, the damping coefficient seems to be the "damper" of the virtual system $(\tilde{\beta} + \sum_{m=0}^{j} \alpha_m)(A - B\tilde{K}^j)$, which gradually amplifies the virtual system and ensures the stability of the iteration while the control policy $\tilde{K}^j$ is updated.

4 Data-driven SPI algorithm without the initial stabilizing policy

In Theorem 1, calculating $\tilde{P}^j$ and $\tilde{K}^j$ and choosing the damping coefficients $\tilde{\beta}$ and α_{j+1} depend on the known A and B. In this section, we remove the restriction.

4.1 Solve $\tilde{P}^j$ by system data

By defining $\tilde{A}^j = \left(\tilde{\beta} + \sum_{m=0}^{j} \alpha_m\right)(A - B\tilde{K}^j)$, we can rewrite system (1) as

$$x(k+1) = \left(\tilde{\beta} + \sum_{m=0}^{j} \alpha_m\right)^{-1} \tilde{A}^j x(k) + B(\tilde{K}^j x(k) + u(k)). \tag{18}$$

By (18), we have

$$\begin{aligned} &x^\top(k+1)\tilde{P}^j x(k+1) - x^\top(k)\tilde{P}^j x(k) \\ &= \left(\tilde{\beta} + \sum_{m=0}^{j} \alpha_m\right)^{-2} x^\top(k)\tilde{A}^{j\top}\tilde{P}^j\tilde{A}^j x(k) - x^\top(k)\tilde{P}^j x(k) + 2\left(\tilde{\beta} + \sum_{m=0}^{j} \alpha_m\right)^{-1} x^\top(k)\tilde{A}^{j\top}\tilde{P}^j B(\tilde{K}^j x(k) + u(k)) \\ &\quad + (\tilde{K}^j x(k) + u(k))^\top B^\top \tilde{P}^j B(\tilde{K}^j x(k) + u(k)). \end{aligned} \tag{19}$$

Substituting (9) into (19) and using the definition of $\tilde{A}^j$ gives

$$\begin{aligned} &x^\top(k+1)\tilde{P}^j x(k+1) - x^\top(k)\tilde{P}^j x(k) \\ &= \left(\tilde{\beta} + \sum_{m=0}^{j} \alpha_m\right)^{-2} x^\top(k)(-W^j)x(k) + 2x^\top(k)\tilde{Y}_1^j(\tilde{K}^j x(k) + u(k)) - x^\top(k)\tilde{K}^{j\top}\tilde{Y}_2^j\tilde{K}^j x(k) \\ &\quad - \left[1 - \left(\tilde{\beta} + \sum_{m=0}^{j} \alpha_m\right)^{-2}\right] x^\top(k)\tilde{P}^j x(k) + u^\top(k)\tilde{Y}_2^j u(k), \end{aligned} \tag{20}$$

where

$$\tilde{Y}_1^j = A^\top\tilde{P}^j B, \quad \tilde{Y}_2^j = B^\top\tilde{P}^j B. \tag{21}$$

Collecting the data yields the following matrices:

$$\Xi_1 = [\mathrm{vecv}(x(k_0+1)) - \mathrm{vecv}(x(k_0)), \ldots, \mathrm{vecv}(x(k_s+1)) - \mathrm{vecv}(x(k_s))]^\top,$$

$$\Xi_2 = [x(k_0)\otimes x(k_0), x(k_1)\otimes x(k_1), \ldots, x(k_s)\otimes x(k_s)]^\top,$$
$$\Xi_3 = [u(k_0)\otimes x(k_0), u(k_1)\otimes x(k_1), \ldots, u(k_s)\otimes x(k_s)]^\top,$$
$$\Xi_4 = [\mathrm{vecv}(u(k_0)), \mathrm{vecv}(u(k_1)), \ldots, \mathrm{vecv}(u(k_s))]^\top,$$
$$\Xi_5 = [\mathrm{vecv}(x(k_0)), \mathrm{vecv}(x(k_1)), \ldots, \mathrm{vecv}(x(k_s))]^\top,$$
$$\Xi_6 = [\mathrm{vecv}(\tilde{K}^j x(k_0)), \ldots, \mathrm{vecv}(\tilde{K}^j x(k_s))]^\top, \tag{22}$$

where $k_0 < k_1 < \cdots < k_s$. Using (20)–(22), one has

$$\psi^j[\mathrm{vecs}(\tilde{P}^j)^\top, \mathrm{vec}(\tilde{Y}_1^j)^\top, \mathrm{vecs}(\tilde{Y}_2^j)^\top]^\top = \phi^j, \tag{23}$$

where $\psi^j = \left[\Xi_1 - ((\tilde{\beta}+\sum_{m=0}^j \alpha_m)^{-2} - 1)\Xi_5, -2\Xi_2(I_{n_x}\otimes \tilde{K}^{j\top}) - 2\Xi_3, -\Xi_4 + \Xi_6\right]$ and $\phi^j = (\tilde{\beta}+\sum_{m=0}^j \alpha_m)^{-2}\Xi_5 \cdot \mathrm{vecs}(-W^j)$.

Lemma 2. If data are collected to satisfy

$$\mathrm{rank}([\Xi_2, \Xi_3, \Xi_4]) = \frac{1}{2}(n_x+n_u)(n_x+n_u+1), \tag{24}$$

then ψ^j is full column-rank.

Proof. The proof is omitted here, and a similar process can be found in [2,12,13].

For $j = 0, 1, 2, \ldots$, if Eq. (24) is satisfied, we have

$$[\mathrm{vecs}(\tilde{P}^j)^\top, \mathrm{vec}(\tilde{Y}_1^j)^\top, \mathrm{vecs}(\tilde{Y}_2^j)^\top]^\top = (\psi^{j\top}\psi^j)^{-1}\psi^{j\top}\phi^j. \tag{25}$$

4.2 Choose $\tilde{\beta}$ and α_{j+1} and calculate $\tilde{K}^{j+1}$ by the model-free method

Recalling Section 3, the designed $\tilde{\beta}$ and α_0 should satisfy $\tilde{\beta}+\alpha_0 < \bar{\rho} = 1/\rho(A-B\tilde{K}^0)$. We should design a model-free condition to replace it. If the solution $\tilde{P}^j$ to the Lyapunov function (9) of the artificial system $(\tilde{\beta}+\sum_{m=0}^j \alpha_m)(A-B\tilde{K}^j)$ is positive definite, it follows that the artificial system $(\tilde{\beta}+\sum_{m=0}^j \alpha_m)(A-B\tilde{K}^j)$ is Schur, i.e., $\rho[(\tilde{\beta}+\sum_{m=0}^j \alpha_m)(A-B\tilde{K}^j)] < 1$. Therefore, when $\tilde{P}^0$ is positive from (25), it can be deduced that $\rho[(\tilde{\beta}+\alpha_0)(A-B\tilde{K}^0)] < 1$.

Determine $\tilde{\beta}$ by the model-free method. Set $\tilde{\beta} \leftarrow \tilde{\beta}_z$ and any sufficiently small $\alpha_0 > 0$, where $\tilde{\beta}_z$ is a decreasing constant and satisfies

$$0 < \tilde{\beta}_{z+1} < \tilde{\beta}_z, \quad \lim_{z\to+\infty} \tilde{\beta}_z = 0, \tag{26}$$

for $z = 0, 1, 2, \ldots$. For step z, if $\tilde{P}^0 > 0$ from (25), then $\tilde{\beta}_z$ is used as $\tilde{\beta}$ in (25), otherwise repeat (26) for $z \leftarrow z+1$ until $\tilde{P}^0 > 0$. Output: $\tilde{\beta} \leftarrow \tilde{\beta}_z$.

Remark 2. Through the above process, $\tilde{\beta}$ and α_0 can be determined satisfying $\tilde{\beta}+\alpha_0 < \bar{\rho} = 1/\rho(A-B\tilde{K}^0)$. $\tilde{\beta}$ is a damping coefficient that is usually set as $\tilde{\beta} \in (0,1)$ for open-loop unstable systems. When the $\tilde{\beta}$ is set to satisfy $\tilde{\beta} \in (0, \bar{\rho}-\alpha_0)$, then a larger $\tilde{\beta}$ accelerates the convergence of $\rho(A-B\tilde{K}^j)$ (from Theorem 1). When the $\tilde{\beta}$ is set to satisfy $\tilde{\beta} \in (\bar{\rho}-\alpha_0, 1)$, then the above model-free process quickly adjusts $\tilde{\beta}$ to the interval $(0, \bar{\rho}-\alpha_0)$.

Update gain $\tilde{K}^{j+1}$ by the model-free method. By (25), $\tilde{Y}_1^j$ and $\tilde{Y}_2^j$ are obtained. Then, from (10), $\tilde{K}^{j+1}$ is updated by

$$\tilde{K}^{j+1} = \left(\tilde{\beta}+\sum_{m=0}^j \alpha_m\right)^2 \left(R + \left(\tilde{\beta}+\sum_{m=0}^j \alpha_m\right)^2 \tilde{Y}_2^j\right)^{-1} \tilde{Y}_1^{j\top}. \tag{27}$$

Select α_{j+1} by the model-free scheme. After $\tilde{P}^j$ and $\tilde{K}^{j+1}$ have been updated, α_{j+1} can be selected by (11).

4.3 The overall data-driven SPI algorithm design

Once the stabilizing gain $\tilde{K}^{j+1}$ is obtained, set $K^0 = \tilde{K}^{j+1}$ and write system (1) as

$$x(k+1) = A^i x(k) + B(K^i x(k) + u(k)), \tag{28}$$

where $A^i = (A-BK^i)$ and K^0 is the initial control gain for $i = 0, 1, 2, \ldots$. By using (28), we have

$$x^\top(k+1)P^i x(k+1) - x^\top(k)P^i x(k) = x^\top(k)(A^{i\top}P^iA^i - P^i)x(k) + 2x^\top(k)A^\top PB(K^i x(k) + u(k))$$

$$+ (K^i x(k) + u(k))^\top B^\top P^i B(K^i x(k) + u(k)). \tag{29}$$

Substituting (5) and (28) into (29) gives

$$\begin{aligned} &x^\top(k+1)P^i x(k+1) - x^\top(k)P^i x(k) \\ &= x^\top(k)(-Q - (K^i)^\top R K^i)x(k) + 2x^\top(k)Y_1^i(K^i x(k) + u(k)) - (K^i x(k))^\top Y_2^i x(k) + u^\top(k)Y_2^i u(k), \end{aligned} \tag{30}$$

where $Y_1^i = A^\top P^i B$ and $Y_2^i = B^\top P^i B$. Using the data matrices Ξ_1, Ξ_2, Ξ_3, Ξ_4 and Ξ_5 collected in (22) and $\Xi_6 = [\mathrm{vecv}(K^i x(k_0)), \ldots, \mathrm{vecv}(K^i x(k_s))]^\top$, the matrices $\Psi^i = \left[\Xi_1, -2\Xi_2(I_{n_x} \otimes K^{i\top}) - 2\Xi_3, -\Xi_4 + \Xi_6\right]$ and $\Phi^i = \Xi_5 \mathrm{vecs}(-Q - (K^i)^\top R K^i)$ are obtained. Then, Eq. (30) is rewritten as

$$\Psi^i[\mathrm{vecs}(P^i)^\top, \mathrm{vec}(Y_1^i)^\top, \mathrm{vecs}(Y_2^i)^\top]^\top = \Phi^i. \tag{31}$$

Similar to Lemma 2, if Eq. (24) is satisfied, Ψ^i has full column-rank. Then Eq. (31) has a unique solution as

$$[\mathrm{vecs}(P^i)^\top, \mathrm{vec}(Y_1^i)^\top, \mathrm{vecs}(Y_2^i)^\top]^\top = (\Psi^{i\top}\Psi^i)^{-1}\Psi^{i\top}\Phi^i. \tag{32}$$

According to (6), we have

$$K^{i+1} = (R + Y_2^i)^{-1} Y_1^{i\top}. \tag{33}$$

Finally, the overall Algorithm 1 and its convergence analysis are given.

Algorithm 1 Lyapunov-based data-driven SPI algorithm.

Initialize: Set arbitrary bounded control gain $\tilde{K}^0$ and two sufficiently small positive constants ε_1 and α_0. Collect data until (24) is satisfied. $j \leftarrow 0$, $z \leftarrow 0$.
1: **repeat**
2: Calculate $\tilde{P}^0$ from (25) and update $\tilde{\beta} \leftarrow \tilde{\beta}_z$ by (26); $z \leftarrow z + 1$;
3: **until** $\hat{P}_i^0 > 0$; (**Phase 1 end**)
4: **repeat**
5: Solve $\tilde{P}^j$, $\tilde{Y}_1^j$ and $\tilde{Y}_2^j$ by (25);
6: Calculate $\tilde{K}^{j+1}$ by (27);
7: Choose α_{j+1} from (11); $j \leftarrow j + 1$;
8: **until** $\tilde{\beta} + \sum_{m=0}^{j} \alpha_m \geqslant 1$; (**Phase 2 end**)
9: Set $K^i \leftarrow \tilde{K}^{j+1}$, $i \leftarrow 0$;
10: **repeat**
11: Solve P^i, Y_1^i and Y_2^i from (32);
12: Calculate K^{i+1} by (33); $i \leftarrow i + 1$;
13:**until** $\|P^i - P^{i-1}\| < \varepsilon_1$; (**Phase 3 end**)
14:**return** K^{i+1} and P^i as solutions to ARE (4).

Theorem 2. Under Assumption 1, $\tilde{P}^j$ and $\tilde{K}^{j+1}$ are updated by Algorithm 1 until $\tilde{\beta}+\sum_{m=0}^{j+1}\alpha_m \geqslant 1$, the obtained $\tilde{K}^{j+1}$ is a stabilizing control gain of system (1) and the closed-loop system satisfies $\rho(A-B\tilde{K}^{j+1}) < 1/(\tilde{\beta}+\sum_{m=0}^{j+1}\alpha_m)$ for $j = 0, 1, 2, \ldots$. Setting $K^0 = \tilde{K}^{j+1}$ and updating P^i and K^{i+1} by Algorithm 1 for $i = 0, 1, \ldots$, one has $\lim_{i\to\infty} P^i = P^*$ and $\lim_{i\to\infty} K^i = K^*$. Then, the optimal control problem is solved by $u = -K^{i+1}x$.

Proof. If Eq. (24) is satisfied, Eq. (23) is equivalent to (20). The unique $\tilde{P}^j$, $\tilde{L}_1^j$ and $\tilde{L}_2^j$ can be solved from (23) and the unique $\tilde{K}^{j+1}$ can be obtained according to (27). From (27), it follows that $\tilde{K}^{j+1}$ satisfies (10). Then, it is equivalent to solving $\tilde{P}^j$ and $\tilde{K}^{j+1}$ from Theorem 1. Thus, the closed-loop system satisfies Conclusion (1) in Theorem 1. If $\tilde{\beta} + \sum_{m=0}^{j+1}\alpha_m \geqslant 1$, the obtained $\tilde{K}^{j+1}$ is a stabilizing control gain of system (1).

Set $K^0 = \tilde{K}^{j+1}$ as the initial stabilizing gain. This satisfies the initial condition of Lemma 1. If condition (24) is satisfied, Eq. (31) is uniquely solved by (32). Since Eq. (31) is equivalent to (5), solving P^i from (32) is equivalent to solving P^i from (5). It is follows that K^i in (33) satisfies (6). This is equivalent to finding P^i and K^i by Lemma 1. Therefore, there exist $\lim_{i\to\infty} P^i = P^*$ and $\lim_{i\to\infty} K^i = K^*$. Then, the optimal control problem is solved by $u = -K^{i+1}x$.

Remark 3. Since the policy evaluation process of Algorithm 1 is based on the Lyapunov equation (9), it can be referred to as a Lyapunov-based SPI algorithm in order to easily distinguish it from the $\mathcal{Q}$-learning SPI in the later section. In Algorithm 1, the data need only be collected once until the rank condition (24) is satisfied. After Phases 1 and 2, the stabilizing gain $\tilde{K}^{j+1}$ is obtained. The solutions P^i and K^{i+1} to ARE (4) are obtained by Phase 3. Unlike methods [28, 30–33, 36] that combine PI and VI, the proposed data-driven PI Algorithm 1 does not contain any properties of VI. Algorithm 1 only needs to use the same data conditions as methods [14, 26, 28, 30–33, 36] to obtain the stabilizing control policy and optimal control policy, respectively.

5 SPI-based $\mathcal{Q}$-learning algorithm without the initial stabilizing policy

PI-based $\mathcal{Q}$-learning is another important algorithm in ADP that is also limited by the initial stabilizing control policy. To address the problem, an SPI-based $\mathcal{Q}$-learning is developed.

5.1 Choose $\tilde{\beta}$ and α_{j+1} and calculate stabilizing gain $\tilde{K}^{j+1}$ by $\mathcal{Q}$-learning

We design the following artificial system as:

$$\bar{x}(k+1) = \left(\tilde{\beta} + \sum_{m=0}^{j} \alpha_m\right)(Ax(k) + Bu(k)). \tag{34}$$

The Bellman equation for system (34) is obtained as

$$\tilde{V}(x(k)) = x^\top(k)Qx(k) + u^\top(k)Ru(k) + \tilde{V}(\bar{x}(k+1)), \tag{35}$$

where $\tilde{V}(x(k)) = x^\top(k)\tilde{P}x(k)$ and $\tilde{V}(\bar{x}(k+1)) = \bar{x}^\top(k+1)\tilde{P}\bar{x}(k+1)$ with $\tilde{P} = \tilde{P}^\top > 0$. Define $\mathcal{Q}$-function as $\tilde{\mathcal{Q}}(x(k), u(k)) = \tilde{V}(x(k))$. From (35), one has

$$\tilde{\mathcal{Q}}(x(k), u(k)) = x^\top(k)Qx(k) + u^\top(k)Ru(k) + \tilde{V}(\bar{x}(k+1)). \tag{36}$$

For convenience, let $\gamma_j = \tilde{\beta} + \sum_{m=0}^{j} \alpha_m$. Substituting (34) and (35) into (36), one has

$$\begin{aligned}\tilde{\mathcal{Q}}(x(k), u(k)) =& x^\top(k)Qx(k) + u^\top(k)Ru(k) + \gamma_j^2(Ax(k) + Bu(k))^\top\tilde{P}(Ax(k) + Bu(k)) \\ =& X^\top(k)\underbrace{\begin{bmatrix}\tilde{H}_{xx} & \tilde{H}_{xu} \\ \tilde{H}_{ux} & \tilde{H}_{uu}\end{bmatrix}}_{=:\tilde{H}} X(k),\end{aligned} \tag{37}$$

where $X(k) = [x^\top(k), u^\top(k)]^\top$, $\tilde{H}_{xx} = \gamma_j^2 A^\top\tilde{P}A + Q$, $\tilde{H}_{uu} = \gamma_j^2 B^\top\tilde{P}B + R$, $\tilde{H}_{xu} = \gamma_j^2 A^\top\tilde{P}B$ and $\tilde{H}_{ux} = \tilde{H}_{xu}^\top$. Using (37) and $\partial\tilde{\mathcal{Q}}(x(k), u(k))/\partial u(k) = 0$, we have

$$u(k) = -(\tilde{H}_{uu})^{-1}\tilde{H}_{ux}x(k) = -\tilde{K}x(k), \tag{38}$$

where $\tilde{K}$ is the control gain. Using (36) and defining $\bar{Q} = \mathrm{diag}\{Q, R\}$, one obtains

$$X^\top(k)\tilde{H}X(k) = X^\top(k)\bar{Q}X(k) + \gamma_j^2\begin{bmatrix}x(k+1) \\ -\tilde{K}x(k+1)\end{bmatrix}^\top\tilde{H}\begin{bmatrix}x(k+1) \\ -\tilde{K}x(k+1)\end{bmatrix}. \tag{39}$$

Letting $\mathcal{M} = \gamma_j[I_{n_x}, -\tilde{K}^\top]^\top[A \;\; B]$, it can be obtained that Eq. (39) is equivalent to

$$\tilde{H} = \bar{Q} + \mathcal{M}^\top\tilde{H}\mathcal{M}. \tag{40}$$

Lemma 3. Lyapunov equation (40) has a unique positive definite solution $\tilde{H}$ if and only if $\gamma_j(A - B\tilde{K})$ is stable.

Proof. Since $\mathcal{M} = \gamma_j[I_{n_x}, -\tilde{K}^\top]^\top[A \;\; B]$, there exists an invertible matrix Λ such that

$$\Lambda^{-1}\mathcal{M}\Lambda = \gamma_j\begin{bmatrix}A - B\tilde{K} & B \\ 0 & 0\end{bmatrix}, \tag{41}$$

where $\Lambda = \gamma_j\left[\begin{smallmatrix} I_{n_x} & I_{n_u} \\ -\tilde{K} & 0\end{smallmatrix}\right]$. From (41), $\mathcal{M}$ is stable if and only if $\gamma_j(A - B\tilde{K})$ is stable. Clearly, $\mathcal{M}$ is stable if and only if (40) has a unique positive definite solution $\tilde{H}$.

First, we design the following PI-based $\mathcal{Q}$-learning without the dynamic damping coefficients.

Lemma 4 (PI-based $\mathcal{Q}$-learning)**.** Given a stabilizing policy K^0 such that $\rho(A-BK^i)<1$, for $i=0,1,\ldots$, solve the matrix H^i by

$$X^\top(k)H^iX(k)=X^\top(k)\bar{Q}X(k)+\begin{bmatrix}x(k+1)\\-K^ix(k+1)\end{bmatrix}^\top\underbrace{\begin{bmatrix}H^i_{xx}&H^i_{xu}\\H^i_{ux}&H^i_{uu}\end{bmatrix}}_{=:H^i}\begin{bmatrix}x(k+1)\\-K^ix(k+1)\end{bmatrix},\tag{42}$$

where $H^i_{xx}=A^\top P^iA+Q$, $H^i_{xu}=A^\top P^iB$, $H^i_{uu}=B^\top P^iB+R$ and $H^i_{ux}=H^{i\top}_{xu}$. Update the policy by

$$K^{i+1}=(H^i_{uu})^{-1}H^i_{ux}.\tag{43}$$

Then, one has that (1) $\rho(A-BK^{i+1})<1$; (2) $H^*\leqslant H^{i+1}\leqslant H^i$, $\lim_{i\to\infty}H^i=H^*$ and $\lim_{i\to\infty}K^i=K^*$; (3) $P^i=[I_{n_x},(-K^i)^\top]H^i[I_{n_x},(-K^i)^\top]^\top$.

Proof. The proofs of Conclusions (1) and (2) are similar to [40, Theorems 3 and 4]. Substituting the policy $u(k)=-K^ix(k)$ into the Q-function $\mathcal{Q}(x(k),u(k))=x^\top(k)Qx(k)+u^\top(k)Ru(k)+(Ax(k)+Bu(k))^\top P^i(Ax(k)+Bu(k))$ associated with system (1) yields $\mathcal{Q}(x(k),-K^ix(k))=x^\top(k)(Q+K^{i\top}RK^i)x(k)+x^\top(k)(A-BK^i)^\top P^i(A-BK^i)x(k)=[x^\top(k),(-K^ix(k))^\top]H^i[x^\top(k),(-K^ix(k))^\top]^\top$. Since $\mathcal{Q}(x(k),-K^ix(k))=x^\top(k)P^ix(k)$, one has $P^i=[I_{n_x},(-K^i)^\top]H^i[I_{n_x},(-K^i)^\top]^\top$.

Then, based on (38) and (39), we design the following SPI-based $\mathcal{Q}$-learning.

Lemma 5 (SPI-based $\mathcal{Q}$-learning)**.** Given the same $\tilde{\beta}$ and α_0 as in Theorem 1, for $j=0,1,2,\ldots$, solve matrix $\tilde{H}^j$ by

$$X^\top(k)\tilde{H}^jX(k)=X^\top(k)\bar{Q}X(k)+\gamma_j^2\begin{bmatrix}x(k+1)\\-\tilde{K}^jx(k+1)\end{bmatrix}^\top\underbrace{\begin{bmatrix}\tilde{H}^j_{xx}&\tilde{H}^j_{xu}\\\tilde{H}^j_{ux}&\tilde{H}^j_{uu}\end{bmatrix}}_{=:\tilde{H}^j}\begin{bmatrix}x(k+1)\\-\tilde{K}^jx(k+1)\end{bmatrix}.\tag{44}$$

The control policy can be improved by

$$\tilde{K}^{j+1}=(\tilde{H}^j_{uu})^{-1}\tilde{H}^j_{ux}.\tag{45}$$

Choose α_{j+1} by (11), where $\tilde{P}^j$ is given by

$$\tilde{P}^j=[I_{n_x},-\tilde{K}^{j\top}]\tilde{H}^j[I_{n_x},-\tilde{K}^{j\top}]^\top.\tag{46}$$

Then, there are (1) $\rho(A-B\tilde{K}^{j+1})<1/(\tilde{\beta}+\sum_{m=0}^{j+1}\alpha_m)$; (2) α_{j+1} exists and is bounded; (3) this $\mathcal{Q}$-learning is equivalent to Theorem 1.

Proof. Define $\mathcal{M}_{j,\tilde{K}^j}=\gamma_j[I_{n_x},-\tilde{K}^{j\top}]^\top[A\ \ B]$. Solving (44) is equivalent to solving $\tilde{H}^j=\bar{Q}+\mathcal{M}^\top_{j,\tilde{K}^j}\tilde{H}^j\mathcal{M}_{j,\tilde{K}^j}$. Conclusions (1) and (2) of Lemma 5 can be proved by Conclusions (1) and (2) of Lemma 4 with the stability Lemma 3. The detailed proof is similar to *Induction* in Theorem 1.

By using (46) and $[I_{n_x},-\tilde{K}^{j\top}]\tilde{H}^j[I_{n_x},-\tilde{K}^{j\top}]^\top=[I_{n_x},-\tilde{K}^{j\top}](\bar{Q}+\mathcal{M}^\top_{j,\tilde{K}^j}\tilde{H}^j\mathcal{M}_{j,\tilde{K}^j})[I_{n_x},-\tilde{K}^{j\top}]^\top$, we can directly obtain Lyapunov equation (9). Since (44) is equivalent to (11), Lemma 5 is equivalent to Theorem 1. By the equivalence, Conclusions (1) and (2) can also be proved.

By collecting data, we can obtain the following matrices as:

$$\begin{aligned}&\Gamma_X=[\mathrm{vecv}(X(k_0)),\mathrm{vecv}(X(k_1)),\ldots,\mathrm{vecv}(X(k_s))]^\top,\Gamma_x=[\mathrm{vecv}(z(k_1+1)),\ldots,\mathrm{vecv}(z(k_s+1))]^\top,\\&\tilde{\Theta}^j=\Gamma_X-\left(\tilde{\beta}+\sum_{m=0}^{j}\alpha_m\right)^2\Gamma_x,\quad\xi^j=\Gamma_X\mathrm{vecs}(\bar{Q}),\quad z(k+1)=[x^\top(k+1),(\tilde{K}^jx(k+1))^\top]^\top,\end{aligned}\tag{47}$$

where $k_0<k_1<\cdots<k_s$. From (44), one has

$$\tilde{\Theta}^j\mathrm{vecs}(\tilde{H}^j)=\xi^j.\tag{48}$$

Similar to Lemma 2, it can be deduced from (47) and (48) that if the collected system data satisfy

$$\mathrm{rank}(\Gamma_X)=\frac{1}{2}(n_x+n_u)(n_x+n_u+1),\tag{49}$$

matrix $\tilde{\Theta}^j$ has full column-rank. For $j = 0, 1, 2, \ldots$, the unique solution to (48) is derived as

$$\text{vecs}(\tilde{H}^j) = (\tilde{\Theta}^{j\top}\tilde{\Theta}^j)^{-1}\tilde{\Theta}^{j\top}\xi^j. \tag{50}$$

Determine $\tilde{\beta}$ by model-free method. From Lemma 3, if the derived $\tilde{H}^0$ is positive definite, then $(\tilde{\beta} + \alpha_0)(A - B\tilde{K}^0)$ is stable. We still update $\tilde{\beta}$ based on (26). If $\tilde{H}^0$ from (50) is positive definite, then $\tilde{\beta}$ is determined, otherwise, update $\tilde{\beta}$ by (26) until $\tilde{H}^0 > 0$.

Choose α_{j+1} by model-free method. After $\tilde{\beta}$ is determined and $\tilde{K}^{j+1}$ is updated by (45), α_{j+1} is chosen by (46) and (11).

5.2 The overall SPI-based $\mathcal{Q}$-learning algorithm design

If the stabilizing control gain $\tilde{K}^{j+1}$ is obtained, we use Lemma 4 and the data matrices Γ_X and Γ_x collected in (47), where $\tilde{K}^j$ is replaced by K^i in Γ_x. According to (42), there is

$$\Theta^i \text{vecs}(H^i) = \xi^i, \tag{51}$$

where $\xi^i = \Gamma_X \text{vecs}(\bar{Q})$ and $\Theta^i = \Gamma_X - \Gamma_x$. If condition (49) is satisfied, Eq. (51) can be uniquely solved as

$$\text{vecs}(H^i) = (\Theta^{i\top}\Theta^i)^{-1}\Theta^{i\top}\xi^i. \tag{52}$$

Further, update K^{i+1} by (43). Finally, we can design Algorithm 2.

Algorithm 2 SPI-based data-driven $\mathcal{Q}$-learning algorithm.

Initialize: Set arbitrary bounded control gain $\tilde{K}^0$ and two sufficiently small positive constants ε_2 and α_0. Collect data until (49) is satisfied. $j \leftarrow 0$, $z \leftarrow 0$.
1: **repeat**
2: Solve $\tilde{H}^0$ by (50); select $\tilde{\beta} = \tilde{\beta}_z$ by (26); $z \leftarrow z + 1$;
3: **until** $\tilde{H}^0 > 0$; (**Phase 1 end**)
4: **repeat**
5: Solve $\tilde{H}^j$ by (50);
6: Calculate $\tilde{P}^j$ from (46) and $\tilde{K}^{j+1}$ from (45);
7: Choose α_{j+1} from (11); $j \leftarrow j + 1$;
8: **until** $\tilde{\beta} + \sum_{m=0}^{j} \alpha_m \geqslant 1$; (**Phase 2 end**)
9: Set $K^i \leftarrow \tilde{K}^{j+1}$, $i \leftarrow 0$;
10: **repeat**
11: Solve P^i, L_1^i and L_2^i from (32); $i \leftarrow i + 1$;
12:**until** $\|H^i - H^{i-1}\| < \varepsilon_2$; (**Phase 3 end**)
13:**return** K^{i+1} and H^i as the optimal solutions.

Theorem 3. Under Assumption 1, $\tilde{H}^j$ and $\tilde{K}^{j+1}$ are updated by Algorithm 2 until $\tilde{\beta} + \sum_{m=0}^{j+1} \alpha_m \geqslant 1$ for $j = 0, 1, \ldots$. Then, the obtained $\tilde{K}^{j+1}$ is a stabilizing control gain of system (1) and the closed-loop system satisfies $\rho(A - B\tilde{K}^{j+1}) < 1/(\tilde{\beta} + \sum_{m=0}^{j+1} \alpha_m)$ for $j = 0, 1, 2, \ldots$. By setting $K^0 = \tilde{K}^{j+1}$ and updating H^i and K^{i+1} by Algorithm 2 for $i = 0, 1, 2, \ldots$, one has $\lim_{i\to\infty} H^i = H^*$ and $\lim_{i\to\infty} K^i = K^*$, where H^* is solution to $\mathcal{Q}$-function of system (1). Then, the optimal control problem is solved by $u = -K^{i+1}x$.

Proof. If condition (49) is satisfied, then solving (50) is equivalent to solving (44). According to Lemma 5, the SPI-based $\mathcal{Q}$-learning (Lemma 5) is equivalent to Theorem 1. According to Theorem 1, the obtained $\tilde{K}^{j+1}$ is a stabilizing control gain of system (1) and the closed-loop spectral radius satisfies $\rho(A - B\tilde{K}^{j+1}) < 1/(\tilde{\beta} + \sum_{m=0}^{j+1} \alpha_m)$ for $j = 0, 1, 2, \ldots$.

Set $K^0 = \tilde{K}^{j+1}$. If condition (49) is satisfied, then solving (52) is equivalent to solving (42). From Lemma 4, if updating H^i and K^{i+1} by (42) and (43), then one has that $\lim_{i\to\infty} H^i = H^*$ and $\lim_{i\to\infty} K^i = K^*$. The optimal control problem is solved by $u = -K^{i+1}x$.

Remark 4. By Lemmas 4 and 5, we show that the SPI-based $\mathcal{Q}$-learning is equivalent to Theorem 1, and the PI-based $\mathcal{Q}$-learning is equivalent to Lemma 1. Different from [8, 10, 22, 27], the proposed SPI-based $\mathcal{Q}$-learning Algorithm 2 is not limited by the initial stabilizing control policy and can adaptively obtain stabilizing and optimal solutions.

Remark 5. Unlike [39], we propose the SPI framework based on [43, 44] by introducing the dynamic cumulative sum damping coefficients. It can be extended to Lyapunov-based SPI algorithm and $\mathcal{Q}$-learning-based SPI algorithm, thus giving model-free computation methods for stabilizing control policy under different PI frameworks. The two proposed SPI algorithms require the same data conditions as the existing [8, 10, 22, 27, 41, 42] and VI [14, 26]

algorithms, but can compute stabilizing control policy and implement functions similar to pole-placement, for example, $\tilde{\beta}+\sum_{m=0}^{j+1}\alpha_m \geqslant l$ is set as the termination threshold for the Phase 2 of both algorithms, where $l>1$ is a constant. In this case, there is $\rho(A-B\tilde{K}^{j+1})<l^{-1}$ (the function of pole-placement is achieved). According to Lyapunov theory, it is known that the rate of convergence of the system must be greater than l^{-1}.

6 Simulation

We consider a power system [45], which is detailed as follows [39, 45]:

$$x(k+1)=\begin{bmatrix}0.8825 & 0.0014 & 0.0470\\ 0.0894 & 0.9049 & 0.0023\\ 0.0028 & 0.0571 & 0.9995\end{bmatrix}x(k)+\begin{bmatrix}0.0001\\ 0.1190\\ 0.0036\end{bmatrix}u(k), \tag{53}$$

where $x(k)=[x_1(k),x_2(k),x_3(k)]^\top$. The initial state vector is set as $x(0)=[1,1,1]^\top$. Its open-loop poles are $0.8847\pm 0.0405\mathrm{i}$ and 1.0176. Set $Q=I_3$ and $R=1$. The optimal solutions are

$$P^*=\begin{bmatrix}6.4588 & 3.2440 & 6.3335\\ 3.2440 & 7.6482 & 10.1251\\ 6.3335 & 10.1251 & 33.4770\end{bmatrix},\ K^*=\begin{bmatrix}0.4025 & 0.8350 & 1.2052\end{bmatrix}, \tag{54}$$

and the solution of the $\mathcal{Q}$-function is

$$H^*=\begin{bmatrix}30.1049 & 6.7720 & 15.7633 & 0.8316\\ 6.7720 & 26.0829 & 27.7090 & 2.5334\\ 15.7633 & 27.7090 & 142.5754 & 3.1115\\ 0.8316 & 2.5334 & 3.1115 & 1.3211\end{bmatrix}. \tag{55}$$

6.1 Validation of data-driven SPI Algorithm 1

We set $\tilde{\beta}=0.7$, $\alpha_0=0.0001$ and choose the initial control gain as $\tilde{K}^0=[0,0,0]$. To satisfy condition (24), we use the probing noise superimposed by high-frequency sine and cosine functions. When enough data have been collected, the stabilizing control gain $\tilde{K}^j$ is obtained by Phases 1 and 2 of the Algorithm 1, where $\alpha_{j+1}=0.8\times\bar{\alpha}$. The iterative update process is shown in Figure 2(b). The iteration stops at step 4, i.e., $\tilde{\beta}+\sum_{m=0}^{j}\alpha_m\geqslant 1$ is reached. At this time, $\tilde{K}^j$ is iterated as $\tilde{K}^4=[0.3428,0.7320,0.9670]$ and $\rho(A-B\tilde{K}^4)$ is iterated to 0.9652 at $j=4$. As shown in Figure 2(b), the closed-loop spectral radius $\rho(A-B\tilde{K}^j)$ always satisfies the inequality $\rho(A-B\tilde{K}^j)<1/(\tilde{\beta}+\sum_{m=0}^{j}\alpha_m)$. To further test the effectiveness of the scheme (11) and its impact on the convergence speed, we set $c=0.2;0.4;0.6;0.8$ in (11) and run Phases 1 and 2 of Algorithm 1, respectively, to obtain Figure 3(b). This indicates that scheme (11) is effective and the larger the α_{j+1}, the faster the convergence.

Let the control gain be $K^0=\tilde{K}^4$ and run Algorithm 1 until $\|P^i-P^{i-1}\|<\varepsilon_1$, where $\varepsilon_1=10^{-5}$. After simulation, the results are obtained as in Figures 2(a) and 3(a). Figure 2(a) shows the results of the system with the controller updated and uncontrolled. This verifies that Algorithm 1 is effective for open-loop unstable systems. From Figure 3(a), it can be shown that P^i and K^i can be converged to P^* and K^* quickly by PI. The optimal error is obtained as $\|P^i-P^*\|=2.1842\times 10^{-8}$. The final optimal solutions are obtained as

$$P^2=\begin{bmatrix}6.4588 & 3.2440 & 6.3335\\ 3.2440 & 7.6482 & 10.1251\\ 6.3335 & 10.1251 & 33.4770\end{bmatrix},\ K^3=\begin{bmatrix}0.4025 & 0.8350 & 1.2052\end{bmatrix}. \tag{56}$$

Comparison of the above results with (54) shows that optimal control is achieved through Algorithm 1.

6.2 Validation of SPI-based $\mathcal{Q}$-learning Algorithm 2

Set $\tilde{\beta}=0.7$, $\alpha_0=0.0001$ and $\tilde{K}^0=[0,0,0]$. Choose α_{j+1} according to (11) with $c=0.9$. A probing noise identical to Algorithm 1 is used to collect data to satisfy (49), and subsequently we iterate Phases 1 and 2 of Algorithm 2.

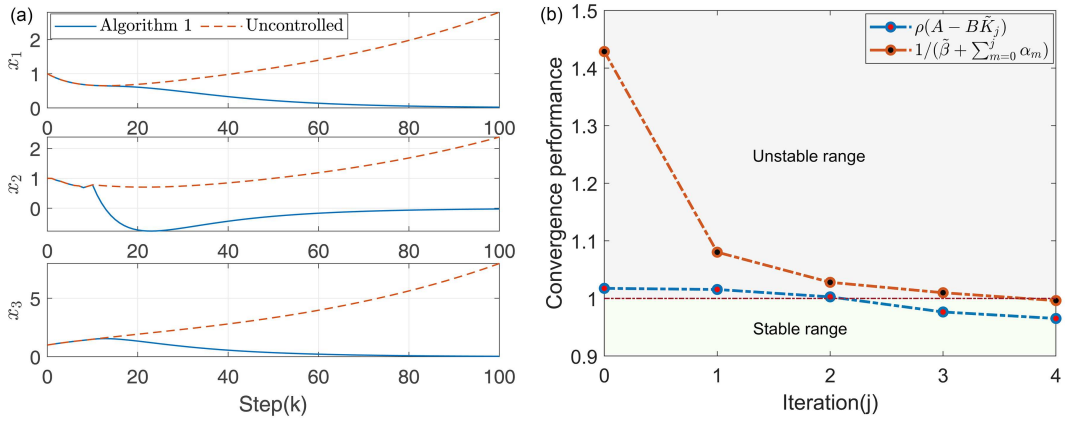

Figure 2 (Color online) (a) Comparison of system states of using Algorithm 1 and without control; (b) the closed-loop spectral radius $\rho(A-B\tilde{K}^j)$ with its upper bound $1/(\tilde{\beta}+\sum_{m=0}^{j}\alpha_m)$ obtained by Algorithm 1 ($c=0.8$).

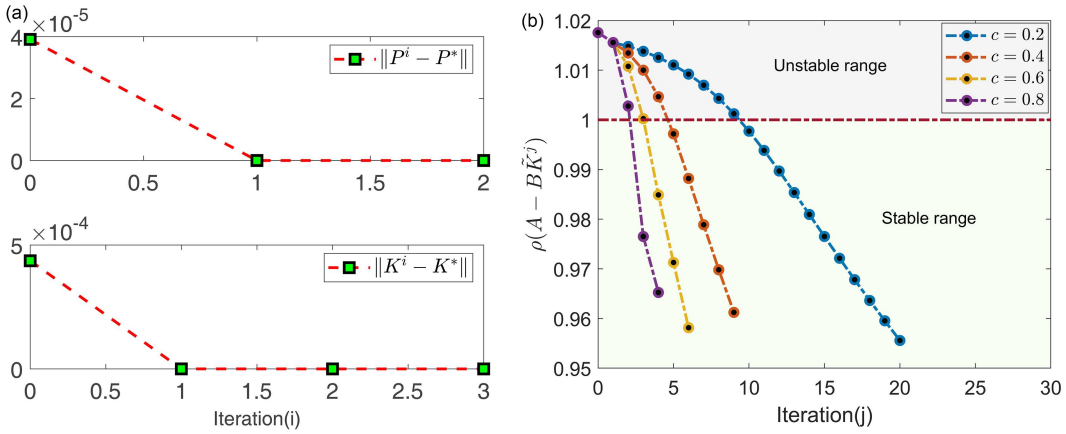

Figure 3 (Color online) (a) Convergence of P^i and K^i; (b) $\rho(A-B\tilde{K}^j)$ obtained by Algorithm 1.

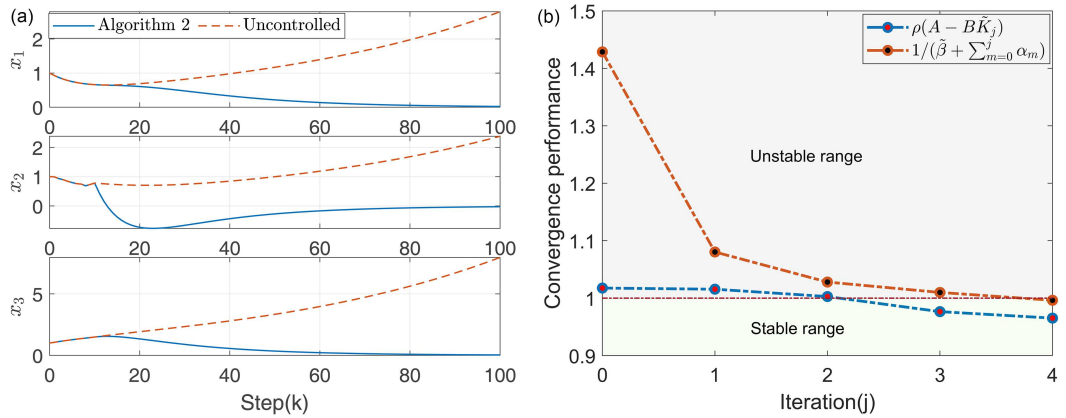

Figure 4 (Color online) (a) Comparison of system states of using Algorithm 2 and without control; (b) the closed-loop spectral radius $\rho(A-B\tilde{K}^j)$ with its upper bound $1/(\tilde{\beta}+\sum_{m=0}^{j}\alpha_m)$ obtained by Algorithm 2 ($c=0.8$).

The iteration results are shown in Figure 4, and after 4 steps iterations, a stabilizing control gain is obtained as $\tilde{K}^4=[0.3428,0.7320,0.9670]$ and $\rho(A-B\tilde{K}^j)$ is iterated to 0.9652 at $j=4$.

As shown in Figure 4(b), $\rho(A-B\tilde{K}^j)$ always converges with an upper bound of $1/(\tilde{\beta}+\sum_{m=0}^{j}\alpha_m)$, i.e., it satisfies $\rho(A-B\tilde{K}^j)<1/(\tilde{\beta}+\sum_{m=0}^{j}\alpha_m)$. To further test the effectiveness of the scheme (11) for Algorithm 2 and its impact on the convergence speed, we set $c=0.2$, $c=0.4$, $c=0.6$ and $c=0.8$ in (11) and run Phases 1 and 2 of Algorithm 2, respectively, to obtain Figure 5(b). This indicates that scheme (11) is effective and the larger the α_{j+1}, the faster the convergence.

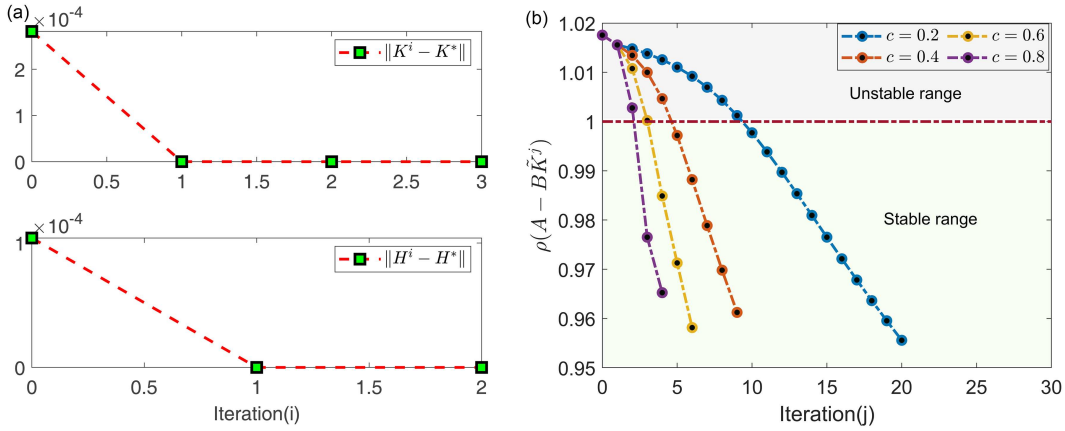

Figure 5 (Color online) (a) Convergence of H^i and K^i; (b) $\rho(A - B\tilde{K}^j)$ obtained by Algorithm 2.

Table 1 Number of iterations for Phases 1 and 2 in Algorithms 1 and 2 in different parameter c conditions.

	$c = 0.1$	$c = 0.2$	$c = 0.4$	$c = 0.6$	$c = 0.8$	$c = 0.9$
Algorithm 1	41	20	9	6	4	4
Algorithm 2	41	20	9	6	4	4

Table 2 CPU time (s) of Algorithms 1 and 2 in different parameter c conditions.

	$c = 0.1$	$c = 0.2$	$c = 0.4$	$c = 0.6$	$c = 0.8$	$c = 0.9$
Algorithm 1	0.044139	0.040314	0.039820	0.037187	0.035762	0.034201
Algorithm 2	0.043026	0.038625	0.037745	0.036081	0.034763	0.033811

Let the control gain be $K^0 = \tilde{K}^4$ and run Algorithm 2 until $\|H^i - H^{i-1}\| < \varepsilon_2$, where $\varepsilon_2 = 10^{-5}$. The simulation results are shown in Figures 4(a) and 5(a). After the stabilizing control policy is obtained, the solution of the $\mathcal{Q}$-function can be quickly converged by Algorithm 2. The optimal error of the algorithm is finally $\|H^i - H^*\| = 1.4512 \times 10^{-9}$. Ultimately, the obtained near-optimal solution to the $\mathcal{Q}$-function and near-optimal gain are

$$H^2 = \begin{bmatrix} 30.1049 & 6.7720 & 15.7633 & 0.8316 \\ 6.7720 & 26.0829 & 27.7090 & 2.5334 \\ 15.7633 & 27.7090 & 142.5754 & 3.1115 \\ 0.8316 & 2.5334 & 3.1115 & 1.3211 \end{bmatrix}, \quad K^3 = [0.4025 \quad 0.8350 \quad 1.2052]. \tag{57}$$

To further analyze the performance of the two algorithms, we simulated and counted the CPU time and the number of iterations with different parameters c. From (11), it can be seen that a larger c setting indicates a larger damping coefficient chosen. The simulation was performed on a computer with CPU parameters "AMD Ryzen 7 5800H with Radeon Graphics (3.20 GHz)".

From Table 1, it can be seen that choosing larger dynamic damping coefficients enhances the convergence speed of Algorithms 1 and 2, allowing them to obtain stabilizing control policies more quickly. Furthermore, the data in Table 1 correspond exactly to Figures 3(b) and 5(b), which both illustrate the equivalence between Algorithms 1 and 2. From Table 2, we can see that (1) choosing larger dynamic damping coefficients reduces the running time of Algorithms 1 and 2, and (2) Algorithm 2 is more efficient than Algorithm 1. Note that the equivalence between Algorithms 1 and 2 is the iteration results, and does not imply that both are equally efficient. This is verified by the results in Tables 1 and 2.

Remark 6. Comparing the results of the above figures, it is easy to see that Algorithm 1 is fully equivalent to Algorithm 2. Note that Algorithms 1 and 2 lead to identical stabilizing and optimal control policies, but the respective solutions P^i and H^i are not identical. They have a relationship as described in (46). In contrast, $\mathcal{Q}$-learning updates the policy based on the $\mathcal{Q}$-function rather than solving for the Lyapunov function in each model-free iteration step (e.g., the difference between (44) and (20)). $\mathcal{Q}$-learning is more straightforward, as can be seen from the policy updating, data collection and solution process of Algorithms 1 and 2. Compared with Algorithm 2, Lyapunov-based SPI Algorithm 1 directly uses the Lyapunov equation as the policy evaluation, which is easier to extend to complex optimization problems. This is due to the fact that the establishment of Lyapunov

equations under complex problems is much simpler than the establishment of Q-functions.

7 Conclusion

Two model-free SPI algorithms based on varying damping coefficients are proposed for optimal control of DT systems. A stable artificial system is constructed through the damping coefficients, and a stabilizing control policy is obtained by varying the damping coefficients and gradually iterating this system to the original closed-loop system. The model-free SPI algorithm and SPI-based $\mathcal{Q}$-learning algorithm are designed and their convergence is analyzed. Through simulations, it is verified that the proposed algorithms can obtain a stabilizing control policy and an optimal control policy. The desired control objective is realized by the proposed algorithms. In future, we will try to extend the method to safe optimal fault-tolerant control [46] and zero-sum games [47].

Acknowledgements This work was supported in part by National Natural Science Foundation of China (Grant Nos. 625B2040, 62273079) and Research Fund of State Key Laboratory of Synthetical Automation for Process Industries in China (Grant No. 2013ZCX01).

References

1 Lewis F L, Vrabie D, Syrmos V L. Optimal Control. Hoboken: John Wiley & Sons, 2012
2 Chen C, Xie L, Jiang Y, et al. Robust output regulation and reinforcement learning-based output tracking design for unknown linear discrete-time systems. IEEE Trans Automat Contr, 2022, 68: 2391–2398
3 Bertsekas D. Dynamic Programming and Optimal Control: Volume I. 4th ed. Worcester: Athena Scientific, 2012
4 Liu M, Liu Q H, Zhang L X, et al. Adaptive dynamic programming-based fault-tolerant attitude control for flexible spacecraft with limited wireless resources. Sci China Inf Sci, 2023, 66: 202201
5 Hewer G. An iterative technique for the computation of the steady state gains for the discrete optimal regulator. IEEE Trans Automat Contr, 1971, 16: 382–384
6 Xie K D, Jiang Y, Yu X, et al. Data-driven cooperative optimal output regulation for linear discrete-time multi-agent systems by online distributed adaptive internal model approach. Sci China Inf Sci, 2023, 66: 170202
7 Jiang Y, Kiumarsi B, Fan J, et al. Optimal Output Regulation of Linear Discrete-Time Systems With Unknown Dynamics Using Reinforcement Learning. IEEE Trans Cybern, 2019, 50: 3147–3156
8 Kiumarsi B, Lewis F L, Jiang Z P. H_∞ control of linear discrete-time systems: Off-policy reinforcement learning. Automatica, 2017, 78: 144–152
9 Fan J, Wu Q, Jiang Y, et al. Model-free optimal output regulation for linear discrete-time lossy networked control systems. IEEE Trans Syst Man Cybern Syst, 2019, 50: 4033–4042
10 Kiumarsi B, Lewis F L, Modares H, et al. Reinforcement Q-learning for optimal tracking control of linear discrete-time systems with unknown dynamics. Automatica, 2014, 50: 1167–1175
11 Lewis F L, Vamvoudakis K G. Reinforcement learning for partially observable dynamic processes: Adaptive dynamic programming using measured output data. IEEE Trans Syst Man Cybern B Cybern, 2010, 41: 14–25
12 Jiang Y, Jiang Z P. Computational adaptive optimal control for continuous-time linear systems with completely unknown dynamics. Automatica, 2012, 48: 2699–2704
13 Gao W, Jiang Z P. Adaptive dynamic programming and adaptive optimal output regulation of linear systems. IEEE Trans Automat Contr, 2016, 61: 4164–4169
14 Bian T, Jiang Z P. Value iteration and adaptive dynamic programming for data-driven adaptive optimal control design. Automatica, 2016, 71: 348–360
15 Wang Z Y, Wang Y Q, Liang L. Optimal dynamic output-feedback controller design for linear output regulation problems with its applications. Sci China Inf Sci, 2025, 68: 182203
16 Ping Z W, Jia Y J, Xiong B G, et al. Optimal output regulation for PMSM speed servo system using approximate dynamic programming. Sci China Inf Sci, 2023, 66: 170206
17 Kleinman D. On an iterative technique for Riccati equation computations. IEEE Trans Autom Control, 1968, 13: 114–115
18 Xie K, Yu X, Lan W. Optimal output regulation for unknown continuous-time linear systems by internal model and adaptive dynamic programming. Automatica, 2022, 146: 110564
19 Luo B, Yang Y, Liu D. Policy iteration Q-learning for data-based two-player zero-sum game of linear discrete-time systems. IEEE Trans Cybern, 2020, 51: 3630–3640
20 Wang G C, Zhang H. Value iteration algorithm for continuous-time linear quadratic stochastic optimal control problems. Sci China Inf Sci, 2024, 67: 122204
21 Jiang Y, Gao W, Wu J, et al. Reinforcement learning and cooperative H_∞ output regulation of linear continuous-time multi-agent systems. Automatica, 2023, 148: 110768
22 Al-Tamimi A, Lewis F L, Abu-Khalaf M. Model-free Q-learning designs for linear discrete-time zero-sum games with application to H-infinity control. Automatica, 2007, 43: 473–481
23 Lee J Y, Park J B, Choi Y H. Integral Q-learning and explorized policy iteration for adaptive optimal control of continuous-time linear systems. Automatica, 2012, 48: 2850–2859
24 Vamvoudakis K G. Q-learning for continuous-time linear systems: A model-free infinite horizon optimal control approach. Syst Control Lett, 2017, 100: 14–20
25 Rizvi S A A, Lin Z. Output feedback Q-learning for discrete-time linear zero-sum games with application to the H-infinity control. Automatica, 2018, 95: 213–221
26 Li C, Ding J, Lewis F L, et al. Model-free Q-learning for the tracking problem of linear discrete-time systems. IEEE Trans Neural Netw Learn Syst, 2024, 35: 3191–3201
27 Rizvi S A A, Lin Z. Output feedback Q-learning control for the discrete-time linear quadratic regulator problem. IEEE Trans Neural Netw Learn Syst, 2019, 30: 1523–1536
28 Luo B, Liu D, Huang T, et al. Output tracking control based on adaptive dynamic programming with multistep policy evaluation. IEEE Trans Syst Man Cybern Syst, 2019, 49: 2155–2165
29 Chen C, Lewis F L, Xie K, et al. Adaptive optimal control of unknown nonlinear systems via homotopy-based policy iteration. IEEE Trans Automat Contr, 2024, 69: 3396–3403
30 Luo B, Yang Y, Wu H N, et al. Balancing value iteration and policy iteration for discrete-time control. IEEE Trans Syst Man Cybern Syst, 2019, 50: 3948–3958
31 Jiang H, Zhou B. Bias-policy iteration based adaptive dynamic programming for unknown continuous-time linear systems. Automatica, 2022, 136: 110058

32 Yang Y, Kiumarsi B, Modares H, et al. Model-free λ-policy iteration for discrete-time linear quadratic regulation. IEEE Trans Neural Netw Learn Syst, 2021, 34: 635–649
33 Gao W, Deng C, Jiang Y, et al. Resilient reinforcement learning and robust output regulation under denial-of-service attacks. Automatica, 2022, 142: 110366
34 Liu S, Gao W, Jiang Z P. Adaptive optimal output-feedback control of discrete-time systems based on hybrid iteration. In: Proceedings of the 63rd Conference on Decision and Control (CDC), 2024. 4512–4517
35 Qasem O, Gao W, Vamvoudakis K G. Adaptive optimal control of continuous-time nonlinear affine systems via hybrid iteration. Automatica, 2023, 157: 111261
36 Shen H, Peng C, Yan H, et al. Data-driven near optimization for fast sampling singularly perturbed systems. IEEE Trans Automat Contr, 2024, 69: 4689–4694
37 Chen C, Lewis F L, Li B. Homotopic policy iteration-based learning design for unknown linear continuous-time systems. Automatica, 2022, 138: 110153
38 Lamperski A. Computing stabilizing linear controllers via policy iteration. In: Proceedings of the 59th Conference on Decision and Control (CDC), 2020. 1902–1907
39 Pang Z, Tang S, Cheng J, et al. Scaling policy iteration based reinforcement learning for unknown discrete-time linear systems. Automatica, 2025, 176: 112227
40 Lopez V G, Alsalti M, Müller M A. Efficient off-policy Q-learning for data-based discrete-time LQR problems. IEEE Trans Automat Contr, 2023, 68: 2922–2933
41 Jiang Y, Fan J, Chai T, et al. Tracking control for linear discrete-time networked control systems with unknown dynamics and dropout. IEEE Trans Neural Netw Learn Syst, 2017, 29: 4607–4620
42 Huang C, Chen C, Xie K, et al. Specified convergence rate guaranteed output tracking of discrete-time systems via reinforcement learning. Automatica, 2024, 161: 111490
43 Li D, Dong J. Data-based efficient off-policy stabilizing optimal control algorithms for discrete-time linear systems via damping coefficients. ArXiv:2412.20845
44 Li D, Dong J. Cooperative optimal output tracking for discrete-time multiagent systems: Stabilizing policy iteration frameworks. IEEE Trans Automat Contr, 2025, doi: 10.1109/TAC.2025.3631555
45 Vamvoudakis K G, Miranda M F, Hespanha J P. Asymptotically stable adaptive-optimal control algorithm with saturating actuators and relaxed persistence of excitation. IEEE Trans Neural Netw Learn Syst, 2016, 27: 2386–2398
46 Zhang D, Wang Y, Meng L, et al. Adaptive critic design for safety-optimal FTC of unknown nonlinear systems with asymmetric constrained-input. ISA Trans, 2024, 155: 309–318
47 Qin C, Qiao X, Wang J, et al. Barrier-critic adaptive robust control of nonzero-sum differential games for uncertain nonlinear systems with state constraints. IEEE Trans Syst Man Cybern Syst, 2023, 54: 50–63