SCIENCE CHINA
Information Sciences

• RESEARCH PAPER •

January 2024, Vol. 67, Iss. 1, 112202:1–112202:14
https://doi.org/10.1007/s11432-023-3818-3

Error-based adaptive optimal tracking control of nonlinear discrete-time systems

Chun LI[1], Jinliang DING[1*], Frank L. LEWIS[2] & Tianyou CHAI[1]

[1]*State Key Laboratory of Synthetical Automation for Process Industries, Northeastern University, Shenyang* 110819*, China;*
[2]*UTA Research Institute, The University of Texas at Arlington, Arlington* 76118*, USA*

Received 5 January 2023/Revised 18 April 2023/Accepted 1 June 2023/Published online 21 December 2023

Abstract In this paper, for the output tracking problem of nonlinear discrete-time systems, a performance index is newly defined using the adaptive dynamic programming (ADP) technique to completely eliminate tracking errors in theory. In contrast to traditional definitions of performance indices in other ADP-based methods, the proposed performance index is not only designed from the perspective of output tracking errors but also introduced errors of system states and control inputs at adjacent stages, which is suitable for practical situations in many industrial applications, such as the alumina production, the flotation process, and the mineral grinding process. We proved that the obtained controller can make the system output fully track the given reference trajectory by applying the iterative criteria of the ADP technique. In addition, the proposed algorithm was implemented using a data-driven technique and neural networks to avoid analyzing and deducing the complicated dynamics of actual industrial processes. Finally, using historical data for a forced-circulation evaporation system in the alumina production process, the effect of the proposed approach was verified through a numerical simulation and compared with that of the proportional-integral controller.

Keywords adaptive dynamic programming, iterative criteria, performance index, tracking errors, tracking problem

1 Introduction

Designing an appropriate and effective controller based on the application background is of practical significance in actual productions. As an efficient approach for solving controllers, adaptive dynamic programming (ADP) has been studied by many researchers, as can be reflected in relevant reviews and books [1–4]. Combined with existing ADP approaches, appropriate performance indices are proposed to cater to the design of controllers with different requirements. Generally, the utility function of the ADP approach for solving regulator problems is defined by the quadratic terms of the system state and control input [5]. As is known to all, for control problems without any specific requirements, the performance index of the ADP approach is generally defined by the accumulation or integral of quadratic terms [6], which is conducive to simplifying mathematical analysis, especially for linear systems. Previously, Ref. [7] stated that the performance index can be formulated from the perspective of only considering the quadratic term of tracking errors to completely eliminate tracking errors. Further, for constraint inputs, an integral expression with inputs as the integral variables replaced the traditional quadratic form of the performance index [8], which forced the control inputs to fall into the specified ranges. In [9], a discounted iterative scheme was proposed for tracking problems, where a new performance index was adopted to avoid solving the feedforward control and eliminate tracking errors. Although a proper performance index is mainly designed to complete control tasks, the simplicity of mathematical processing is also an important factor that researchers pay attention to. For instance, based on the internal model principle, the performance index presented in [10] transformed the tracking problem of linear systems into a regulation one. Similarly, this transformation was applied to solve the tracking problems of nonlinear systems in other previous studies [11–13], but the internal model principle was replaced by an augmented system. Notably, tracking errors, as mentioned by Kiumarsi and Lewis [12], cannot be fully eliminated when using an augmented

* Corresponding author (email: jlding@mail.neu.edu.cn)

system. Therefore, the reasonable design of the performance index plays an important role in solving controllers using the ADP approach.

Generally, deriving and calculating the dynamics knowledge of controlled systems with complex production backgrounds are very difficult. Therefore, solving controllers based on data-driven techniques is favored by researchers. Admittedly, previous studies [14–20] noted that the theoretical development of the ADP approach in practical applications is inseparable from data-driven techniques. In [21], a model-free ADP approach was proposed for solving the optimal operational control of a class of industrial systems with multiple fast-dynamics devices and a slow-dynamics unit. For a multiagent system consisting of multiple human-driven vehicles and an autonomous vehicle, a data-driven nonmodel-based ADP approach was proposed in [22] to control the speeds of vehicles and optimize fuel usage. Meanwhile, using a data-driven technique, an off-policy reinforcement learning was developed in [23], which was used not only to calculate the controller, but also to identify the dynamics of the controlled system. Note that all the systems in these previous studies [21–23] were expressed in the form of linear systems. When using data-driven ADP approaches to solve the controllers of nonlinear systems, the neural networks technique has been broadly applied because of its powerful ability to approximate complex functions [24]. In this regard, an actor-critic neural network scheme was developed in [25] to approximate the control for nonlinear discrete-time systems with input saturation. Using a neural network as the value function approximator, a critic-only structure of the ADP approach was proposed for solving the optimal tracking of autonomous nonlinear switching systems [26]. In [27], an iterative neural dynamic programming approach was proposed for nonlinear systems, where the performance index, control policy, and system model were realized by neural networks. In [28], a neural network-based adaptive critic policy was applied to a wastewater treatment plant. Hence, the neural network-based data-driven technique is an effective way of obtaining the controllers of complex industrial systems.

Notably, nonlinear systems are ubiquitous in industrial fields, and optimal control problems are generally formulated as solving the intractable nonlinear Hamilton-Jacobi-Bellman equations [27]. Therefore, various forms of ADP methods have been used in different applications [29–35]. For example, in [36], a novel updating rule of the ADP approach was developed to eliminate the requirement of an initial stabilizing controller for a nonlinear overhead crane system, which brought unique advantages to the adaptive critic control design. As a common phenomenon in various applications, in many industrial production processes, the reference trajectory of the system output is notably usually a constant vector, which is adjusted to another one unless the production requirements are changed. For example, the cooperative adaptive cruise control problem of heterogeneous vehicle platoons studied in [37] was consistent with such a tracking objective. Meanwhile, in the process of alumina production [38], the liquid level and discharge density of the forced-circulation evaporation system were stabilized at the fixed target values. As emphasized in [39], the mixed-concentrate grade of mineral processing was only allowed to have a small fluctuation at a fixed constant value. For such control problems in industrial production processes described in [38, 39], we designed a controller based on the ADP approach in this paper. In addition, because the feedback control is one of the main methods for designing nonlinear system controllers, the system state feedback and output feedback inverse optimal control approaches were proposed for nonlinear systems [40, 41]. The effects of our proposed approach were compared with that of the feedback controller. Hence, focusing on a class of tracking problems of nonlinear systems with an industrial application background, such as the industrial process of alumina production [38], the single-cell flotation industrial process [42], and the mineral grinding industrial process [43], the main contributions of this paper are as follows:

• Aiming at the output tracking problems of nonlinear discrete-time systems, a novel data-driven ADP approach is proposed for solving the tracking controller, which is designed from the perspective of completely eliminating the output tracking errors in theory.

• In the proposed approach, a novel performance index of ADP is applied to explore the predicted trajectories of partial system states, and another is used to solve the controller on the basis of the obtained predicted trajectories.

• The controller solved by minimizing the proposed performance index can force the system output to satisfy the tracking requirements. The effect of the proposed approach is verified by simulation results on actual industrial production data, and compared with that of a proportional-integral (PI) controller.

The remainder of this paper is organized as follows. In Section 2, we formulated the output tracking problem of a class of nonlinear systems based on actual industrial production processes. In Section 3, according to the proposed nonlinear systems, two different performance indices of ADP approaches were

designed to solve the appropriate controller. In Section 4, the validity of the obtained controller was proven using ADP iterative criteria. Further, the implementation of the proposed ADP approach was introduced in combination with neural networks and data-driven techniques. Thereafter, the corresponding pseudocode was illustrated in detail. In Section 5, a numerical example based on the alumina data of an actual industrial production process verified the effectiveness of the proposed approach. Finally, the conclusion of this paper is summarized in Section 6.

Notations. Throughout this paper, we use $\mathbb{N}$, $\mathbb{N}^+$, $\mathbb{R}^n$, and $\mathbb{R}^{n\times m}$ to denote the set of natural numbers, the set of positive integers, the n-dimensional Euclidean space, and the set of all real $n\times m$ matrices, respectively. Vertical bars $\|\cdot\|$ represent the Euclidean norm for vectors, or the induced matrix norm for matrices. Let $(\cdot)^{\mathrm{T}}$ and $(\cdot)^{-1}$ denote the transpose and inverse operators, respectively.

2 Formulation of the tracking problem

Consider a class of nonlinear systems described by

$$\boldsymbol{x}_{k+1} = \boldsymbol{f}(\boldsymbol{x}_k) + \boldsymbol{G}(\boldsymbol{x}_k)\boldsymbol{u}_k, \tag{1a}$$

$$\bar{\boldsymbol{x}}_{k+1} = \bar{\boldsymbol{f}}(\bar{\boldsymbol{x}}_k) + \bar{\boldsymbol{G}}(\bar{\boldsymbol{x}}_k)\boldsymbol{x}_k, \tag{1b}$$

$$\boldsymbol{y}_k = \boldsymbol{h}(\bar{\boldsymbol{x}}_k), \tag{1c}$$

where $\boldsymbol{x}_k \in \mathbb{R}^n$ and $\bar{\boldsymbol{x}}_k \in \mathbb{R}^{\bar{n}}$ are the system states, $\boldsymbol{u}_k \in \mathbb{R}^m$ is the control input, $\boldsymbol{y}_k \in \mathbb{R}^p$ is the system output, $\boldsymbol{f} : \mathbb{R}^n \to \mathbb{R}^n$, $\bar{\boldsymbol{f}} : \mathbb{R}^{\bar{n}} \to \mathbb{R}^{\bar{n}}$, $\boldsymbol{G} : \mathbb{R}^n \to \mathbb{R}^{n\times m}$, $\bar{\boldsymbol{G}} : \mathbb{R}^{\bar{n}} \to \mathbb{R}^{\bar{n}\times n}$, and $\boldsymbol{h} : \mathbb{R}^{\bar{n}} \to \mathbb{R}^p$ denote continuous nonlinear functions, and the matrix $\boldsymbol{G}$ satisfies a full column rank.

The controller was designed to make the system output $\boldsymbol{y}_k$ completely track a given constant vector $\boldsymbol{\eta}$. That is, $\lim_{k\to\infty} \boldsymbol{y}_k = \boldsymbol{\eta}$. Let the symbol $\boldsymbol{e}_k$, $\boldsymbol{e}_k = \boldsymbol{y}_k - \boldsymbol{\eta}$ denote the tracking error. The systems (1) can be regarded as a common type of controlled systems in industrial processes, such as the industrial process of alumina production [38], the single-cell flotation industrial process [42], and the mineral grinding industrial process [43]. Generally, the following assumptions are made on systems (1).

Assumption 1. Systems (1a) and (1b) are stabilizable.

Assumption 2. Suppose that there is at least a control policy such that the system output can fully track the given reference tracking target. That is, the error between the output and the given reference tracking target is theoretically equal to the zero vector as the operating stage tends to infinity.

Remark 1. When the variables $\boldsymbol{x}_k$, $\boldsymbol{u}_k$, and $\bar{\boldsymbol{x}}_k$ are given, the system states $\boldsymbol{x}_{k+1}$ and $\bar{\boldsymbol{x}}_{k+1}$ must be determined; further, the system state $\bar{\boldsymbol{x}}_{k+2}$ and output $\boldsymbol{y}_{k+2}$ can be completely determined. Therefore, the control input $\boldsymbol{u}_k$ can be designed by minimizing the error between the system output $\boldsymbol{y}_{k+2}$ and the tracking trajectory $\boldsymbol{\eta}$ because it no longer has any influence on the system state $\bar{\boldsymbol{x}}_{k+1}$ and output $\boldsymbol{y}_{k+1}$. Further, we can first calculate a predicted trajectory $\boldsymbol{s}_{k+1}$ of the system state $\boldsymbol{x}_{k+1}$, which can make the system output realize the tracking tasks. Then, the control input $\boldsymbol{u}_k$ can be solved on the basis of the predicted trajectory.

3 Designing performance indices

In this section, we design a novel performance index of the ADP approach to solve the tracking problem of nonlinear discrete-time systems (1) with an aim to completely eliminate tracking errors without requiring the desired control input.

3.1 Performance index for predicting the desired state

Let $\boldsymbol{\varepsilon}_k$ and $\boldsymbol{\nu}_k$ be the errors of the system states between adjacent stages, which are given by $\boldsymbol{\varepsilon}_k = \bar{\boldsymbol{x}}_{k+1} - \bar{\boldsymbol{x}}_k$ and $\boldsymbol{\nu}_k = \boldsymbol{x}_{k+1} - \boldsymbol{x}_k$. Because the state $\boldsymbol{x}_k$ can be regarded as the input of (1b), its predicted value is worth studying. Therefore, a certain range for the error $\boldsymbol{\nu}_k$ must be specified. Suppose that the error $\boldsymbol{\nu}_k$ is within the range $\boldsymbol{\nu}_k \in (\underline{\boldsymbol{\nu}}, \overline{\boldsymbol{\nu}})$, where $\underline{\boldsymbol{\nu}} \prec \boldsymbol{0}$ and $\overline{\boldsymbol{\nu}} \succ \boldsymbol{0}$ denote the lower and upper limits, respectively. Further, let the variables $\boldsymbol{\tau}$ and $\boldsymbol{\xi}$ be $\boldsymbol{\tau} = (\overline{\boldsymbol{\nu}} + \underline{\boldsymbol{\nu}})/2$ and $\boldsymbol{\xi} = (\overline{\boldsymbol{\nu}} - \underline{\boldsymbol{\nu}})/2$, respectively.

The state $\bar{\boldsymbol{x}}_{k+1}$ can definitely be determinable if the system states $\boldsymbol{x}_k$ and $\bar{\boldsymbol{x}}_k$ are known. Therefore, the tracking error and system states (implied by the variables $\boldsymbol{\varepsilon}_k$ and $\boldsymbol{\nu}_k$) can be considered from the $(k+1)$th stage when the system states $\boldsymbol{x}_k$ and $\bar{\boldsymbol{x}}_k$ are given, as explained in Remark 1. Hence, considering

systems (1b) and (1c) only, we can define the performance index V_k, which is mainly used to predict a trajectory of the system state $\boldsymbol{x}_k$ and eliminate the tracking error $\boldsymbol{e}_k$ as follows:

$$V_k = \sum_{j=k}^{\infty} r_j = \sum_{j=k}^{\infty} \left\{ \boldsymbol{e}_{j+1}^{\mathrm{T}} \boldsymbol{Q} \boldsymbol{e}_{j+1} + \boldsymbol{\varepsilon}_j^{\mathrm{T}} \boldsymbol{T} \boldsymbol{\varepsilon}_j + \psi\left(\boldsymbol{\nu}_j\right) \right\}, \tag{2}$$

where $r_j = \boldsymbol{e}_{j+1}^{\mathrm{T}} \boldsymbol{Q} \boldsymbol{e}_{j+1} + \boldsymbol{\varepsilon}_j^{\mathrm{T}} \boldsymbol{T} \boldsymbol{\varepsilon}_j + \psi\left(\boldsymbol{\nu}_j\right)$ denotes the utility function, $\boldsymbol{Q} \in \mathbb{R}^{p \times p}$ and $\boldsymbol{T} \in \mathbb{R}^{\bar{n} \times \bar{n}}$ are symmetric positive-definite matrices, and $\psi\left(\boldsymbol{\nu}_j\right)$ is formulated by

$$\begin{aligned} \psi\left(\boldsymbol{\nu}_j\right) &= \sum_{l=1}^{n} \pi_l \xi_l \int_0^{\nu_{l,j}} \left[\sigma^{-1}\left(\frac{\mu_l - \tau_l}{\xi_l} \right) + c_l \right] \mathrm{d}\mu_l \\ &\triangleq \int_{\boldsymbol{0}}^{\boldsymbol{\nu}_j} \left[\sigma^{-1}\left(\boldsymbol{\Xi}^{-1}\left(\boldsymbol{\mu} - \boldsymbol{\tau} \right) \right) + \boldsymbol{c} \right]^{\mathrm{T}} \boldsymbol{\Xi} \boldsymbol{\Pi} \mathrm{d}\boldsymbol{\mu}, \end{aligned} \tag{3}$$

$\sigma(\cdot)$ is the hyperbolic tangent function, $\boldsymbol{\Xi} = \mathtt{diag}(\boldsymbol{\xi}) \in \mathbb{R}^{n \times n}$ and $\boldsymbol{\Pi} \in \mathbb{R}^{n \times n}$ are diagonal matrices, ξ_l is the lth entry of vector $\boldsymbol{\xi}$, τ_l is the lth entry of vector $\boldsymbol{\tau}$, $\pi_l > 0$ represents the (l,l)th entry of matrix $\boldsymbol{\Pi}$, $\nu_{l,j}$ represents the lth entry of vector $\boldsymbol{\nu}_j$, μ_l denotes the integral variable and represents the lth entry of vector $\boldsymbol{\mu}$, $\boldsymbol{c} \in \mathbb{R}^n$ is a constant vector, and the lth entry of $\boldsymbol{c}$ is given by $c_l = \sigma^{-1}(\tau_l / \xi_l)$. Note that the diagonal matrix $\boldsymbol{\Pi}$ can be regarded as a weight matrix, and its function is consistent with matrices $\boldsymbol{Q}$ and $\boldsymbol{T}$. Clearly, matrices $\boldsymbol{\Xi}$ and $\boldsymbol{\Pi}$ are positive-definite.

Remark 2. The function $\psi\left(\boldsymbol{\nu}_j\right)$ is nonnegative because each integrated term $\sigma^{-1}\left(\left(\mu_l - \tau_l\right)/\xi_l\right) + c_l$, $l = 1, 2, \ldots, n$, in (3) is not only monotonically increasing, but also equal to zero when $\mu_l = 0$. Further, because matrices $\boldsymbol{Q}$ and $\boldsymbol{T}$ are positive-definite, the performance index V_k defined by (2) must be nonnegative with appropriate control policies.

Remark 3. From Assumptions 1 and 2, the situation where the performance index V_k is equal to zero indicates that the variables $\boldsymbol{e}_j$, $\boldsymbol{\varepsilon}_j$, and $\boldsymbol{\nu}_j$, $j = k, k+1, \ldots,$ are zero vectors with appropriate dimensions. That is, there exist $\boldsymbol{x} \in \mathbb{R}^n$ and $\bar{\boldsymbol{x}} \in \mathbb{R}^{\bar{n}}$ such that $\boldsymbol{y}_j = \boldsymbol{h}(\bar{\boldsymbol{x}}) = \boldsymbol{\eta}$, $\boldsymbol{x}_j = \boldsymbol{x}$, and $\bar{\boldsymbol{x}}_j = \bar{\boldsymbol{x}} = \bar{\boldsymbol{f}}(\bar{\boldsymbol{x}}) + \bar{\boldsymbol{G}}(\bar{\boldsymbol{x}})\boldsymbol{x}$ for $j = k, k+1, \ldots,$ where both $\boldsymbol{x}$ and $\bar{\boldsymbol{x}}$ are constant vectors to be determined. Hence, the performance index V_k can be bounded without a discount factor.

Rewrite the performance index V_k as

$$V_k = r_k + \sum_{j=k+1}^{\infty} r_j = r_k + V_{k+1}. \tag{4}$$

Then, the main objective is to calculate the variable $\boldsymbol{\nu}_k$ of the following optimization problem:

$$\arg\min_{\boldsymbol{\nu}} V_k = \arg\min_{\boldsymbol{\nu}} \left\{ r_k + V_{k+1} \right\}. \tag{5}$$

From (4), the Hamiltonian function can be formulated by

$$H(k; \boldsymbol{\nu}) = r_k + V_{k+1} - V_k. \tag{6}$$

Let the symbols V_k^* and $\boldsymbol{\nu}_k^*$ denote the optimal performance index and its corresponding variable, respectively. Then, the Hamiltonian function $H^*(k; \boldsymbol{\nu}^*)$ given by $H^*(k; \boldsymbol{\nu}^*) = r_k^* + V_{k+1}^* - V_k^*$ is equal to zero. In addition, by forcing the gradient of the Hamiltonian function along the variable $\boldsymbol{\nu}_k$ to be the zero vector, we have

$$V_k^* = r_k^* + V_{k+1}^*, \tag{7}$$

$$\boldsymbol{\nu}_k^* = \boldsymbol{\tau} - \boldsymbol{\Xi} \sigma \left(\left(\boldsymbol{\Xi} \boldsymbol{\Pi} \right)^{-1} \frac{\partial V_{k+1}^*}{\partial \boldsymbol{\nu}_k^*} + \boldsymbol{c} \right), \tag{8}$$

which ensures that the variable $\boldsymbol{\nu}_k^*$ is within the specified range.

Note that when the variable $\boldsymbol{\nu}_k^*$ is solved by minimizing the performance index V_k, the predicted trajectory $\boldsymbol{s}_{k+1}$ of the system state $\boldsymbol{x}_{k+1}$ can be calculated by

$$\boldsymbol{s}_{k+1} = \boldsymbol{x}_k + \boldsymbol{\nu}_k^*, \tag{9}$$

where the system state $\boldsymbol{x}_k$ is assumed to be known. Thereafter, the remaining task to solve the tracking problem of systems (1) is to design a controller such that the system state $\boldsymbol{x}_{k+1}$ tracks the predicted trajectory $\boldsymbol{s}_{k+1}$ completely.

3.2 Performance index for calculating the control input

Suppose that the predicted trajectory $\boldsymbol{s}_{k+1}$ of the system state $\boldsymbol{x}_{k+1}$ has been calculated by (9). Because the function $\boldsymbol{G}(\cdot)$ is assumed to satisfy the full column rank, the performance index used to solve the controller $\boldsymbol{u}_k$ can be designed as [7]

$$\bar{V}_k = \sum_{j=k}^{\infty} \bar{r}(\boldsymbol{x}_j, \boldsymbol{u}_j, \boldsymbol{s}_{j+1}) = \bar{r}_k + \bar{V}_{k+1}, \tag{10}$$

$$\bar{r}_j = \frac{1}{2} \left[\boldsymbol{f}(\boldsymbol{x}_j) + \boldsymbol{G}(\boldsymbol{x}_j)\boldsymbol{u}_j - \boldsymbol{s}_{j+1}\right]^{\mathrm{T}} \boldsymbol{P} \left[\boldsymbol{f}(\boldsymbol{x}_j) + \boldsymbol{G}(\boldsymbol{x}_j)\boldsymbol{u}_j - \boldsymbol{s}_{j+1}\right], \tag{11}$$

where $\bar{r}_j = \bar{r}(\boldsymbol{x}_j, \boldsymbol{u}_j, \boldsymbol{s}_{j+1})$ denotes the utility function, and $\boldsymbol{P} \in \mathbb{R}^{n\times n}$ is a positive-definite matrix. Then, the control input that minimizes the performance index (10) can be solved by

$$\boldsymbol{u}_k = \arg\min_{\boldsymbol{u}} \left\{\bar{r}(\boldsymbol{x}_k, \boldsymbol{u}_k, \boldsymbol{s}_{k+1}) + \bar{V}_{k+1}\right\}. \tag{12}$$

From (10), the corresponding discrete-time Hamiltonian function can be defined by

$$\bar{H}_k = \bar{r}(\boldsymbol{x}_k, \boldsymbol{u}_k, \boldsymbol{s}_{k+1}) + \bar{V}_{k+1} - \bar{V}_k. \tag{13}$$

Let the symbols $\bar{V}_k^*$ and $\boldsymbol{u}_k^*$ be the optimal performance index and its corresponding control input, respectively. Then, forcing the gradient of the discrete-time Hamiltonian function $\bar{H}_k$ along the control input $\boldsymbol{u}_k$ to be the zero vector with appropriate dimensions, one has

$$\bar{V}_k^* = \bar{r}^*(\boldsymbol{x}_k, \boldsymbol{u}_k^*, \boldsymbol{s}_{k+1}) + \bar{V}_{k+1}^*, \tag{14}$$

$$\boldsymbol{u}_k^* = -\left[\boldsymbol{G}^{\mathrm{T}}(\boldsymbol{x}_k)\boldsymbol{P}\boldsymbol{G}(\boldsymbol{x}_k)\right]^{-1} \boldsymbol{G}^{\mathrm{T}}(\boldsymbol{x}_k) \left\{\frac{\partial \bar{V}_{k+1}^*}{\partial \boldsymbol{x}_{k+1}} + \boldsymbol{P}\left[\boldsymbol{f}(\boldsymbol{x}_k) - \boldsymbol{s}_{k+1}\right]\right\}. \tag{15}$$

Notably, according to Assumptions 3 and 4 proposed in Section 4, the control input $\boldsymbol{u}_k$ solved by minimizing the performance index $\bar{V}_k$ can make the system state $\boldsymbol{x}_k$ fully track the predicted trajectory $\boldsymbol{s}_k$. Then, from Remark 3, the control policy that fully eliminates the tracking errors can be obtained by minimizing the proposed performance indices, which is proven in Theorem 3 in Section 4.

4 Convergence and implementation of the proposed ADP approach

In this section, we first focus on the convergence of the proposed performance index. Then, using neural networks, the implementations of performance indices and the corresponding control policies are introduced in detail. Finally, the pseudocode of the proposed ADP algorithm is illustrated.

4.1 Convergence of the performance indices

We rewrite the optimal performance index V_k^* and its corresponding variable $\boldsymbol{\nu}_k^*$ as follows to facilitate the interpretation of expressions (4) and (5):

$$V_k^* = r_k^{\boldsymbol{\nu}^*} + V_{k+1}^{*;\boldsymbol{\nu}^*}, \tag{16}$$

$$\boldsymbol{\nu}_k^* = \arg\min_{\boldsymbol{\nu}} \left\{r_k^{\boldsymbol{\nu}} + V_{k+1}^{*;\boldsymbol{\nu}}\right\}, \tag{17}$$

where $r_k^{\boldsymbol{\nu}^*} = \boldsymbol{e}_{k+1}^{\mathrm{T}}\boldsymbol{Q}\boldsymbol{e}_{k+1} + \boldsymbol{\varepsilon}_k^{\mathrm{T}}\boldsymbol{T}\boldsymbol{\varepsilon}_k + \psi(\boldsymbol{\nu}_k^*)$, and the superscript $\boldsymbol{\nu}^*$ in the performance index $V_{k+1}^{*;\boldsymbol{\nu}^*}$ means that all the independent variables of V_{k+1}^* are transferred from states $\bar{\boldsymbol{x}}_k$ and $\boldsymbol{x}_k$ based on the variable $\boldsymbol{\nu}_k^*$, and the superscripts in $r_k^{\boldsymbol{\nu}}$ and $V_{k+1}^{*;\boldsymbol{\nu}}$ express similar meanings.

In the iterative process of the ADP approach, let the symbols $V_k^{\bar{\imath}}$ and $\boldsymbol{\nu}_k^{\bar{\imath}}$ denote the $\bar{\imath}$th iterative performance index and the corresponding variable, respectively, where the superscript $\bar{\imath} \in \mathbb{N}$ represents the iterative index. Then, the iterative criteria of $V_k^{\bar{\imath}}$ and $\boldsymbol{\nu}_k^{\bar{\imath}}$ can be designed as

$$\boldsymbol{\nu}_k^{\bar{\imath}} = \arg\min_{\boldsymbol{\nu}} \left\{r_k^{\boldsymbol{\nu}} + V_{k+1}^{\bar{\imath};\boldsymbol{\nu}}\right\}, \tag{18}$$

$$V_k^{\bar{\imath}+1} = r_k^{\boldsymbol{\nu}^{\bar{\imath}}} + V_{k+1}^{\bar{\imath};\boldsymbol{\nu}^{\bar{\imath}}}, \tag{19}$$

where $r_k^{\boldsymbol{\nu}^{\bar{\imath}}} = \boldsymbol{e}_{k+1}^{\mathrm{T}}\boldsymbol{Q}\boldsymbol{e}_{k+1}+\boldsymbol{\varepsilon}_k^{\mathrm{T}}\boldsymbol{T}\boldsymbol{\varepsilon}_k+\psi\left(\boldsymbol{\nu}_k^{\bar{\imath}}\right)$, and the superscript $\boldsymbol{\nu}^{\bar{\imath}}$ of $V_{k+1}^{\bar{\imath};\boldsymbol{\nu}^{\bar{\imath}}}$ denotes that all the independent variables of $V_{k+1}^{\bar{\imath}}$ are transferred from the states $\bar{\boldsymbol{x}}_k$ and $\boldsymbol{x}_k$ by the $\bar{\imath}$th control policy. Furthermore, the value of V_k^0 is set to zero for all the independent variables, and the value of $\boldsymbol{\nu}_k^0$ is solved using (18). As the optimal performance index V_k^* is nonnegative, the following assumption can be made on the optimal performance index V_k^*, which is generally proposed in the ADP approach [14].

Assumption 3. For the optimal performance index V_k^* and the utility function r_k, suppose that there exists a real scalar $\gamma \in (1,\ \infty)$ satisfying $\gamma r_k^{\boldsymbol{\nu}_k} \geqslant V_{k+1}^{*;\boldsymbol{\nu}_k}$, where the variable $\boldsymbol{\nu}_k$ is bounded.

Then, according to the iterative criteria (18) and (19), we can derive the following theorem.

Theorem 1. Let Assumptions 1–3 hold. Set the value of the initial iterative performance index V_k^0 to zero for all independent variables, and solve the initial value of $\boldsymbol{\nu}_k^0$ by the criterion (18). For the iterative index $\bar{\imath} \in \mathbb{N}$, using the iterative criteria (18) and (19) alternately yields the following:

(1) as $\bar{\imath}$ tends to infinity, one has $\lim_{\bar{\imath}\to\infty} V_k^{\bar{\imath}} = V_k^*$;

(2) as $\bar{\imath}$ tends to infinity, one has $\lim_{\bar{\imath}\to\infty} \boldsymbol{\nu}_k^{\bar{\imath}} = \boldsymbol{\nu}_k^*$,

where $\boldsymbol{\nu}_k^*$ and V_k^* denote the optimal variable and the optimal performance index, respectively.

Proof. (1) From $V_k^0 = 0$ and $V_k^* \geqslant 0$, one has $0 = V_k^0 \leqslant V_k^* \leqslant 2V_k^* = \{1 + [\gamma/(1+\gamma)]^0\}V_k^*$. Further, assuming that the inequality $V_k^i \leqslant \left\{1 + [\gamma/(1+\gamma)]^i\right\}V_k^*$ holds for the ith iteration, we have

$$\begin{aligned} V_k^{i+1} &= r_k^{\boldsymbol{\nu}^i} + V_{k+1}^{i;\boldsymbol{\nu}^i} \leqslant r_k^{\boldsymbol{\nu}^*} + V_{k+1}^{i;\boldsymbol{\nu}^*} \\ &\leqslant r_k^{\boldsymbol{\nu}^*} + \left[1 + \left(\frac{\gamma}{1+\gamma}\right)^i\right] V_{k+1}^{*;\boldsymbol{\nu}^*} + \frac{\gamma^i}{(1+\gamma)^{i+1}}\left(\gamma r_k^{\boldsymbol{\nu}^*} - V_{k+1}^{*;\boldsymbol{\nu}^*}\right) \\ &= \left[1 + \left(\frac{\gamma}{1+\gamma}\right)^{i+1}\right] V_k^*, \end{aligned} \tag{20}$$

where the first inequality sign is ensured by the criterion (18). Hence, it can be derived from the mathematical induction method that the inequality $V_k^{\bar{\imath}} \leqslant \{1 + [\gamma/(1+\gamma)]^{\bar{\imath}}\}V_k^*$ always holds for each performance index. Further, taking $\bar{\imath}$ as infinity yields

$$\lim_{\bar{\imath}\to\infty} V_k^{\bar{\imath}} \leqslant \lim_{\bar{\imath}\to\infty}\left[1 + \left(\frac{\gamma}{1+\gamma}\right)^{\bar{\imath}}\right] V_k^* = V_k^*. \tag{21}$$

Similarly, we can obtain that there exists a parameter $\lambda \in [0,\ 1)$ that enables $V_k^{\bar{\imath}+1} \geqslant [1 - \frac{1-\lambda}{(1+\gamma^{-1})^{\bar{\imath}+1}}]V_k^*$ to hold for each iterative index. Then, taking the iterative index $\bar{\imath}$ to infinity, one has

$$\lim_{\bar{\imath}\to\infty} V_k^{\bar{\imath}+1} \geqslant \lim_{\bar{\imath}\to\infty}\left[1 - \frac{1-\lambda}{(1+\gamma^{-1})^{\bar{\imath}+1}}\right] V_k^* = V_k^*. \tag{22}$$

Therefore, combining (21) and (22) yields that the expression $\lim_{\bar{\imath}\to\infty} V_k^{\bar{\imath}} = V_k^*$ holds.

(2) Note that the optimal performance index V_k^* is given by

$$V_k^* = r_k^{\boldsymbol{\nu}^*} + V_{k+1}^{*;\boldsymbol{\nu}^*} = \min_{\boldsymbol{\nu}}\left\{\lim_{\bar{\imath}\to\infty}\left(r_k^{\boldsymbol{\nu}} + V_{k+1}^{\bar{\imath};\boldsymbol{\nu}}\right)\right\}, \tag{23}$$

which indicates that the formulation $\lim_{\bar{\imath}\to\infty} \boldsymbol{\nu}_k^{\bar{\imath}} = \boldsymbol{\nu}_k^*$ holds.

The iterative operation in Theorem 1 is generally named the value-iteration method. Policy-iteration method, as another common iterative operation, can also calculate $\boldsymbol{\nu}_k^*$ and V_k^*. For the policy-iteration method, supposing that the value of $\boldsymbol{\nu}^0$ is given, the iterative criteria can be indicated by

$$V_k^{\bar{\imath}} = r_k^{\boldsymbol{\nu}^{\bar{\imath}}} + V_{k+1}^{\bar{\imath};\boldsymbol{\nu}^{\bar{\imath}}}, \tag{24}$$

$$\boldsymbol{\nu}_k^{\bar{\imath}+1} = \arg\min_{\boldsymbol{\nu}}\left\{r_k^{\boldsymbol{\nu}} + V_{k+1}^{\bar{\imath};\boldsymbol{\nu}}\right\}, \tag{25}$$

where the superscript $\bar{\imath} \in \mathbb{N}$ represents the iterative index. For calculating the $\bar{\imath}$th iterative performance index that satisfies the iterative criterion (24), define an auxiliary iterative function $J_k^{\bar{\jmath}}$, $\bar{\jmath} \in \mathbb{N}$ as

$$J_k^{\bar{\jmath}+1} = r_k^{\boldsymbol{\nu}^{\bar{\imath}}} + J_{k+1}^{\bar{\jmath};\boldsymbol{\nu}^{\bar{\imath}}}, \tag{26}$$

where J_k^0 is equal to $V_k^{\bar{\imath}-1}$ when the index $\bar{\imath}$ is a positive integer, J_k^0 is zero when $\bar{\imath}=0$, and the symbol $J_{k+1}^{\bar{\jmath};\boldsymbol{\nu}^{\bar{\imath}}}$ denotes the value of $J_{k+1}^{\bar{\jmath}}$, wherein all the variables of $J_{k+1}^{\bar{\jmath}}$ are transferred from the states $\bar{\boldsymbol{x}}_k$ and $\boldsymbol{x}_k$ by the $\bar{\imath}$th iterative control policy. Then, the $\bar{\imath}$th iterative performance index $V_k^{\bar{\imath}}$ satisfying criterion (24) can be calculated using the following lemma.

Lemma 1. For any given variable $\boldsymbol{\nu}^{\bar{\imath}}$ in the policy-iteration method, the $\bar{\imath}$th iterative performance index that satisfies (24) can be obtained by $V_k^{\bar{\imath}}=\lim_{\bar{\jmath}\to\infty}J_k^{\bar{\jmath}}$, where $J_k^{\bar{\jmath}}$ is updated by criterion (26).

Proof. By repeatedly utilizing criterion (26), one has

$$\begin{aligned}\lim_{\bar{\jmath}\to\infty}J_k^{\bar{\jmath}}&=\lim_{\bar{\jmath}\to\infty}\left\{r_k^{\boldsymbol{\nu}^{\bar{\imath}}}+J_{k+1}^{\bar{\jmath}-1;\boldsymbol{\nu}^{\bar{\imath}}}\right\}\\&=\lim_{\bar{\jmath}\to\infty}\left\{r_k^{\boldsymbol{\nu}^{\bar{\imath}}}+r_{k+1}^{\boldsymbol{\nu}^{\bar{\imath}}}+J_{k+2}^{\bar{\jmath}-2;\boldsymbol{\nu}^{\bar{\imath}}}\right\}\\&=\lim_{\bar{\jmath}\to\infty}\sum_{j=k}^{k+\bar{\jmath}}r_j^{\boldsymbol{\nu}^{\bar{\imath}}}+\lim_{\bar{\jmath}\to\infty}J_{k+\bar{\jmath}}^{0;\boldsymbol{\nu}^{\bar{\imath}}}\\&=\sum_{j=k}^{\infty}r_j^{\boldsymbol{\nu}^{\bar{\imath}}}+J_{\infty}^{0;\boldsymbol{\nu}^{\bar{\imath}}}=V_k^{\bar{\imath}},\end{aligned}\tag{27}$$

where $\sum_{j=k}^{\infty}r_j^{\boldsymbol{\nu}^{\bar{\imath}}}$ is equal to $V_k^{\bar{\imath}}$ according to definition (2), and $J_{\infty}^{0;\boldsymbol{\nu}^{\bar{\imath}}}=0$ is ensured by the precondition that $\boldsymbol{\nu}_k^0$ is admissible. Hence, the performance index $V_k^{\bar{\imath}}$ satisfying (24) can be solved.

Based on Lemma 1, the results of the policy-iteration method are shown in the following theorem.

Theorem 2. Let Assumptions 1–3 hold. Suppose that there is an initial variable $\boldsymbol{\nu}_k^0$ so that the state $\bar{\boldsymbol{x}}_k$ can track the given reference trajectory. Then, using criteria (24) and (25), the iterative performance index $V_k^{\bar{\imath}}$ and its corresponding variable $\boldsymbol{\nu}_k^{\bar{\imath}}$ satisfy $V_k^{\bar{\imath}+1}\leqslant V_k^{\bar{\imath}}$, $\lim_{\bar{\imath}\to\infty}V_k^{\bar{\imath}}=V_k^*$, and $\lim_{\bar{\imath}\to\infty}\boldsymbol{\nu}_k^{\bar{\imath}}=\boldsymbol{\nu}_k^*$, where $\boldsymbol{\nu}_k^*$ and V_k^* are the optimal variable and the optimal performance index, respectively.

Proof. Define an auxiliary iterative function $\mathscr{V}_k^{\bar{\imath}+1}$ as

$$\mathscr{V}_k^{\bar{\imath}+1}=r_k^{\boldsymbol{\nu}^{\bar{\imath}+1}}+V_{k+1}^{\bar{\imath};\boldsymbol{\nu}^{\bar{\imath}+1}},\tag{28}$$

which implies that the inequality $\mathscr{V}_k^{\bar{\imath}+1}\leqslant r_k^{\boldsymbol{\nu}^{\bar{\imath}}}+V_{k+1}^{\bar{\imath};\boldsymbol{\nu}^{\bar{\imath}}}=V_k^{\bar{\imath}}$ must hold. Therefore, the auxiliary iterative function $\mathscr{V}_k^{\bar{\imath}+1}$ must conform to

$$\begin{aligned}\mathscr{V}_k^{\bar{\imath}+1}&=r_k^{\boldsymbol{\nu}^{\bar{\imath}+1}}+V_{k+1}^{\bar{\imath};\boldsymbol{\nu}^{\bar{\imath}+1}}\geqslant r_k^{\boldsymbol{\nu}^{\bar{\imath}+1}}+\mathscr{V}_{k+1}^{\bar{\imath}+1;\boldsymbol{\nu}^{\bar{\imath}+1}}\\&=r_k^{\boldsymbol{\nu}^{\bar{\imath}+1}}+r_{k+1}^{\boldsymbol{\nu}^{\bar{\imath}+1}}+V_{k+2}^{\bar{\imath};\boldsymbol{\nu}^{\bar{\imath}+1}}\geqslant\sum_{j=k}^{k+1}r_j^{\boldsymbol{\nu}^{\bar{\imath}+1}}+\mathscr{V}_{k+2}^{\bar{\imath}+1;\boldsymbol{\nu}^{\bar{\imath}+1}}\\&=\cdots\geqslant\sum_{j=k}^{\infty}r_j^{\boldsymbol{\nu}^{\bar{\imath}+1}}+\mathscr{V}_{\infty}^{\bar{\imath}+1;\boldsymbol{\nu}^{\bar{\imath}+1}}=V_k^{\bar{\imath}+1},\end{aligned}\tag{29}$$

which implies that the inequality $V_k^{\bar{\imath}+1}\leqslant\mathscr{V}_k^{\bar{\imath}+1}\leqslant V_k^{\bar{\imath}}$ holds. Furthermore, for any admissible variable $\boldsymbol{\varsigma}$, define the criterion as $\bar{\mathscr{V}}_k^{\bar{\imath}+1}=r_k^{\boldsymbol{\varsigma}}+\bar{\mathscr{V}}_{k+1}^{\bar{\imath};\boldsymbol{\varsigma}}$ with $\bar{\mathscr{V}}_k^0=V_k^{\infty}$. Then, according to (18), one has $V_k^{\infty}\leqslant\bar{\mathscr{V}}_k^1$. Thereafter, we can conclude from mathematical induction that $V_k^{\infty}\leqslant\bar{\mathscr{V}}_k^{\bar{\imath}}$ holds. Thereby, $V_k^{\infty}\leqslant\lim_{\bar{\imath}\to\infty}\bar{\mathscr{V}}_k^{\bar{\imath}}=V_k^*$ clearly holds when the variable $\boldsymbol{\varsigma}$ is replaced with the variable $\boldsymbol{\nu}^*$. In addition, considering that the optimal performance index V_k^* must satisfy the inequality $V_k^*\leqslant V_k^{\infty}$, it can be obtained from the squeeze theorem that $\lim_{\bar{\imath}\to\infty}V_k^{\bar{\imath}}=V_k^*$. Therefore, as explained in (23), the conclusion $\lim_{\bar{\imath}\to\infty}\boldsymbol{\nu}_k^{\bar{\imath}}=\boldsymbol{\nu}_k^*$ holds, which shows that the proof is complete.

Consider the performance index $\bar{V}_k$ given in (10), where the controlled system is described as (1a) and the desired tracking trajectory $\boldsymbol{s}_{k+1}$ is calculated using (9). Let Assumptions 1 and 2 hold. Rewrite the optimal performance index $\bar{V}_k^*$ to satisfy

$$\bar{V}_k^*=\bar{r}_k^{\boldsymbol{u}^*}+\bar{V}_{k+1}^{*;\boldsymbol{u}^*},\tag{30}$$

where $\bar{r}_k^{\boldsymbol{u}^*}=\bar{r}(\boldsymbol{x}_k,\boldsymbol{u}_k^*,\boldsymbol{s}_{k+1})$. The value $\bar{V}_{k+1}^{*;\boldsymbol{u}^*}$ is equal to the performance index $\bar{V}_{k+1}^*$, wherein all the independent variables of $\bar{V}_{k+1}^*$ are transferred from the system state $\boldsymbol{x}_k$ according to the optimal control

input $\boldsymbol{u}_k^*$, and $\boldsymbol{u}_k^*$ is solved by

$$\boldsymbol{u}_k^* = \arg\min_{\boldsymbol{u}} \left\{ \bar{r}_k^{\boldsymbol{u}} + \bar{V}_{k+1}^{*;\boldsymbol{u}} \right\}, \tag{31}$$

where $\bar{r}_k^{\boldsymbol{u}} = \bar{r}(\boldsymbol{x}_k, \boldsymbol{u}_k, \boldsymbol{s}_{k+1})$, and the superscript $\boldsymbol{u}$ in $\bar{V}_{k+1}^{*;\boldsymbol{u}}$ represents a connotation similar to that in $\bar{V}_{k+1}^{*;\boldsymbol{u}^*}$. Then, the following assumption can be made on the optimal performance index $\bar{V}_k^*$ and its corresponding utility function.

Let the parameter $\ell \in \mathbb{N}$ denote the iterative index. Further, set the initial iterative performance index $\bar{V}_k^0$ to zero for all the independent variables, where the superscript represents the iterative index. Then, the iterative criteria of $\bar{V}_k^\ell$ and $\boldsymbol{u}_k^\ell$ can be illustrated by

$$\boldsymbol{u}_k^\ell = \arg\min_{\boldsymbol{u}} \left\{ \bar{r}_k^{\boldsymbol{u}} + \bar{V}_{k+1}^{\ell;\boldsymbol{u}} \right\}, \tag{32}$$

$$\bar{V}_k^{\ell+1} = \bar{r}_k^{\boldsymbol{u}^\ell} + \bar{V}_{k+1}^{\ell;\boldsymbol{u}^\ell}, \tag{33}$$

where $\bar{V}_{k+1}^{\ell;\boldsymbol{u}^\ell}$ is equal to the value of $\bar{V}_{k+1}^\ell$, wherein the superscript $\boldsymbol{u}^\ell$ means all the independent variables of $\bar{V}_{k+1}^\ell$ are transferred from the system state $\boldsymbol{x}_k$ based on the ℓth control policy. We provide the following assumption similar to the explanation in Assumption 3.

Assumption 4. For the optimal performance index $\bar{V}_k^*$ and the utility function $\bar{r}_k$, suppose that there exists a scalar $\bar{\gamma}$ large enough, $\bar{\gamma} \in (1,\ \infty)$, satisfying $\bar{\gamma}\bar{r}_k^{\boldsymbol{u}_k} \geqslant \bar{V}_{k+1}^{*;\boldsymbol{u}_k}$, where the input $\boldsymbol{u}_k$ is bounded.

Then, according to criteria (32) and (33), we can derive the following theorem.

Lemma 2 ([7])**.** Let Assumptions 1, 2, and 4 hold. Set the initial iterative performance index $\bar{V}_k^0$ to zero for all the independent variables, where the superscript represents the iterative index. Then, using criteria (32) and (33) alternately yields $\lim_{\ell\to\infty} \bar{V}_k^\ell = \bar{V}_k^*$, and $\lim_{\ell\to\infty} \boldsymbol{u}_k^\ell = \boldsymbol{u}_k^*$, where $\bar{V}_k^*$ and $\boldsymbol{u}_k^*$ are the optimal performance index and the optimal control input, respectively.

Note that when Assumptions 1–4 hold, the performance indices (2) and (10) are bounded for any admissible variable $\boldsymbol{\nu}_k$ and control input $\boldsymbol{u}_k$, implying that the tracking errors can be completely eliminated in theory. In addition, based on Theorem 1 and Lemma 2, the variable $\boldsymbol{\nu}_k^*$ and the control input $\boldsymbol{u}_k^*$ that minimize the performance indices (2) and (10) can be solved using the ADP approach. Therefore, the following theorem can be proposed.

Theorem 3. Let Assumptions 1–4 hold. Design the control input $\boldsymbol{u}_k$ of nonlinear system (1) as (15), where $\boldsymbol{s}_{k+1}$ in (15) is calculated using (8) and (9). Then, the nonlinear discrete-time system with the state $[\boldsymbol{\nu}_k^{\mathrm{T}}, \boldsymbol{\varepsilon}_k^{\mathrm{T}}, \boldsymbol{e}_{k+1}^{\mathrm{T}}]^{\mathrm{T}}$ is uniformly ultimately bounded, and the control policy obtained by (15) can eliminate the tracking errors of system (1) in theory, where $\boldsymbol{\nu}_k$, $\boldsymbol{\varepsilon}_k$, and $\boldsymbol{e}_{k+1}^{\mathrm{T}}$ are defined as $\boldsymbol{\nu}_k = \boldsymbol{x}_{k+1} - \boldsymbol{x}_k$, $\boldsymbol{\varepsilon}_k = \bar{\boldsymbol{x}}_{k+1} - \bar{\boldsymbol{x}}_k$, and $\boldsymbol{e}_{k+1} = \boldsymbol{y}_{k+1} - \boldsymbol{\eta}$, respectively, and $\bar{\boldsymbol{x}}_k$, $\boldsymbol{x}_k$, and $\boldsymbol{y}_{k+1}$ are described in systems (1).

Proof. For the nonlinear system with the state $[\boldsymbol{\nu}_k^{\mathrm{T}}, \boldsymbol{\varepsilon}_k^{\mathrm{T}}, \boldsymbol{e}_{k+1}^{\mathrm{T}}]^{\mathrm{T}}$, define the Lyapunov function as

$$\begin{aligned} L(\boldsymbol{\nu}_k, \boldsymbol{\varepsilon}_k, \boldsymbol{e}_{k+1}) = \sum_{j=k}^{\infty} \Big\{ & \psi(\boldsymbol{\nu}_j) + \boldsymbol{\varepsilon}_j^{\mathrm{T}} \boldsymbol{T} \boldsymbol{\varepsilon}_j + \boldsymbol{e}_{j+1}^{\mathrm{T}} \boldsymbol{Q} \boldsymbol{e}_{j+1} \\ & + \frac{1}{2} \left[\boldsymbol{f}(\boldsymbol{x}_j) + \boldsymbol{G}(\boldsymbol{x}_j)\boldsymbol{u}_j - \boldsymbol{s}_{j+1} \right]^{\mathrm{T}} \boldsymbol{P} \left[\boldsymbol{f}(\boldsymbol{x}_j) + \boldsymbol{G}(\boldsymbol{x}_j)\boldsymbol{u}_j - \boldsymbol{s}_{j+1} \right] \Big\}, \end{aligned} \tag{34}$$

where $\boldsymbol{s}_j$ and $\boldsymbol{u}_j$ are determined by (8) and (15), respectively. From Remark 2, the Lyapunov function $L(\boldsymbol{\nu}_k, \boldsymbol{\varepsilon}_k, \boldsymbol{e}_{k+1})$ must satisfy $L(\boldsymbol{\nu}_k, \boldsymbol{\varepsilon}_k, \boldsymbol{e}_{k+1}) \geqslant 0$. Further, according to Theorem 1 and Lemma 2, the control input $\boldsymbol{u}_k$ formulated using (15) is the optimal one of systems (1). Then, the Lyapunov function $L(\boldsymbol{\nu}_k, \boldsymbol{\varepsilon}_k, \boldsymbol{e}_{k+1})$ can be rewritten as

$$L(\boldsymbol{\nu}_k, \boldsymbol{\varepsilon}_k, \boldsymbol{e}_{k+1}) = V_k^* + \bar{V}_k^*, \tag{35}$$

where V_k^* and $\bar{V}_k^*$ are the optimal performance indices defined in (7) and (14), respectively. Thereby, one has

$$\begin{aligned} \Delta L_k &= L(\boldsymbol{\nu}_{k+1}, \boldsymbol{\varepsilon}_{k+1}, \boldsymbol{e}_{k+2}) - L(\boldsymbol{\nu}_k, \boldsymbol{\varepsilon}_k, \boldsymbol{e}_{k+1}) \\ &= V_{k+1}^* + \bar{V}_{k+1}^* - \left(V_k^* + \bar{V}_k^* \right) \end{aligned}$$

$$= -\Big\{\psi(\boldsymbol{\nu}_k) + \boldsymbol{\varepsilon}_k^{\mathrm{T}}\boldsymbol{T}\boldsymbol{\varepsilon}_k + \boldsymbol{e}_{k+1}^{\mathrm{T}}\boldsymbol{Q}\boldsymbol{e}_{k+1} + \frac{1}{2}\left[\boldsymbol{f}(\boldsymbol{x}_k) + \boldsymbol{G}(\boldsymbol{x}_k)\boldsymbol{u}_k - \boldsymbol{s}_{k+1}\right]^{\mathrm{T}}\boldsymbol{P}\left[\boldsymbol{f}(\boldsymbol{x}_k) + \boldsymbol{G}(\boldsymbol{x}_k)\boldsymbol{u}_k - \boldsymbol{s}_{k+1}\right]\Big\} \leqslant 0, \tag{36}$$

which deduces that the nonlinear discrete-time system with the state $[\boldsymbol{\nu}_k^{\mathrm{T}}, \boldsymbol{\varepsilon}_k^{\mathrm{T}}, \boldsymbol{e}_{k+1}^{\mathrm{T}}]^{\mathrm{T}}$ is uniformly ultimately bounded. Further, because ΔL_k is equal to zero when and only when $\boldsymbol{\nu}_k$, $\boldsymbol{\varepsilon}_k$, and $\boldsymbol{e}_{k+1}$ are zero vectors with the appropriate dimensions, the obtained control policy can eliminate the tracking errors in theory. Hence, the proof is complete.

Remark 4. From Theorem 3, for any given constant vector $\boldsymbol{\eta}$ satisfying Assumption 2, the input $\boldsymbol{u}_k$ shown as (15) can effectively make the system output $\boldsymbol{y}_k$ of nonlinear discrete-time system (1) completely track the given reference trajectory $\boldsymbol{\eta}$ in theory.

4.2 Implementation based on neural networks

Because neural networks are, as mentioned by Arbib [24], regarded as effective tools for approximating complex functions, the proposed ADP approach was implemented using neural networks. Using a three-layer neural network with n_v and n_ν neurons, approximations of the performance index V_k and the variable $\boldsymbol{\nu}_k$ can be formulated by

$$\hat{V}_k = \boldsymbol{W}_{v2}\sigma\left(\boldsymbol{W}_{v1}\boldsymbol{z}_k + \boldsymbol{b}_{v1}\right) + \boldsymbol{b}_{v2}, \tag{37}$$

$$\hat{\boldsymbol{\nu}}_k = \boldsymbol{\Xi}\sigma\left(\boldsymbol{W}_{\nu2}\sigma\left(\boldsymbol{W}_{\nu1}\boldsymbol{z}_k + \boldsymbol{b}_{\nu1}\right) + \boldsymbol{b}_{\nu2}\right) + \boldsymbol{\tau}, \tag{38}$$

where $\boldsymbol{W}_{v\bar{\ell}}$ and $\boldsymbol{W}_{\nu\bar{\ell}}$, $\bar{\ell} \in \{1, 2\}$ are the weight matrices, $\boldsymbol{b}_{v\bar{\ell}}$ and $\boldsymbol{b}_{\nu\bar{\ell}}$ are the bias vectors, and $\boldsymbol{z}_k = [\boldsymbol{x}_k^{\mathrm{T}}, \bar{\boldsymbol{x}}_k^{\mathrm{T}}, \boldsymbol{\eta}^{\mathrm{T}}]^{\mathrm{T}}$ is the input vector of the neural networks. For determining the weight matrices and bias vectors, the loss functions can be defined by $E_v = \frac{1}{2}\left\|\hat{V}_k - V_k\right\|^2$ and $E_\nu = \frac{1}{2}\|\hat{\boldsymbol{\nu}}_k - \boldsymbol{\nu}_k\|^2$, respectively. Then, the weight matrices and bias vectors can be solved using the gradient descent method.

Similarly, to approximate the performance index $\bar{V}_k$ and the control policy $\boldsymbol{u}_k$, the three-layer neural networks with $n_{\bar{v}}$ and n_u neurons are used as

$$\hat{\bar{V}}_k = \boldsymbol{W}_{\bar{v}2}\sigma\left(\boldsymbol{W}_{\bar{v}1}\bar{\boldsymbol{z}}_k + \boldsymbol{b}_{\bar{v}1}\right) + \boldsymbol{b}_{\bar{v}2}, \tag{39}$$

$$\hat{\boldsymbol{u}}_k = \boldsymbol{W}_{u2}\sigma\left(\boldsymbol{W}_{u1}\bar{\boldsymbol{z}}_k + \boldsymbol{b}_{u1}\right) + \boldsymbol{b}_{u2}, \tag{40}$$

where $\boldsymbol{W}_{\bar{v}\bar{\ell}}$ and $\boldsymbol{W}_{u\bar{\ell}}$, $\bar{\ell} \in \{1, 2\}$ are the weight matrices, $\boldsymbol{b}_{\bar{v}\bar{\ell}}$ and $\boldsymbol{b}_{u\bar{\ell}}$ are the bias vectors, and $\bar{\boldsymbol{z}}_k = [\bar{\boldsymbol{x}}_k^{\mathrm{T}}, \boldsymbol{s}_{k+1}^{\mathrm{T}}]^{\mathrm{T}}$ represents the input vector of the neural networks. Furthermore, defining the loss functions as $E_{\bar{v}} = \frac{1}{2}\|\hat{\bar{V}}_k - \bar{V}_k\|^2$ and $E_u = \frac{1}{2}\|\hat{\boldsymbol{u}}_k - \boldsymbol{u}_k\|^2$, we can obtain the weight matrices and bias vectors using the gradient descent method.

In the proposed ADP algorithm, considering that calculating the variable $\boldsymbol{\nu}_k$ and the control input $\boldsymbol{u}_k$ requires predicting the system states and the control output, the models of the controlled system can be designed based on the neural networks as follows:

$$\hat{\boldsymbol{x}}_{k+1} = \boldsymbol{W}_{x2}\sigma\left(\boldsymbol{W}_{x1}\left[\boldsymbol{x}_k^{\mathrm{T}}, \boldsymbol{u}_k^{\mathrm{T}}\right]^{\mathrm{T}} + \boldsymbol{b}_{x1}\right) + \boldsymbol{b}_{x2}, \tag{41}$$

$$\hat{\bar{\boldsymbol{x}}}_{k+1} = \boldsymbol{W}_{\bar{x}2}\sigma\left(\boldsymbol{W}_{\bar{x}1}\left[\bar{\boldsymbol{x}}_k^{\mathrm{T}}, \boldsymbol{x}_k^{\mathrm{T}}\right]^{\mathrm{T}} + \boldsymbol{b}_{\bar{x}1}\right) + \boldsymbol{b}_{\bar{x}2}, \tag{42}$$

$$\hat{\boldsymbol{y}}_k = \boldsymbol{W}_{y2}\sigma\left(\boldsymbol{W}_{y1}\bar{\boldsymbol{x}}_k + \boldsymbol{b}_{y1}\right) + \boldsymbol{b}_{y2}, \tag{43}$$

where $\boldsymbol{W}_{x\bar{\ell}}$, $\boldsymbol{W}_{\bar{x}\bar{\ell}}$, and $\boldsymbol{W}_{y\bar{\ell}}$, $\bar{\ell} \in \{1, 2\}$ represent the weight matrices, and $\boldsymbol{b}_{x\bar{\ell}}$, $\boldsymbol{b}_{\bar{x}\bar{\ell}}$, and $\boldsymbol{b}_{y\bar{\ell}}$ denote the bias vectors with appropriate dimensions when the neurons of hidden-layers are determined. As the methods of calculating these weight matrices and bias vectors are similar to those above, the detailed procedures are not repeated.

Remark 5. Note that using neural networks to represent complex functions is one of the common techniques in ADP approaches, as can be seen from previous studies [29, 30, 32]. The specific details of the relevant training process and convergence proof can be found in [27]. Therefore, no further detailed description or explanation is elaborated here.

Then, applying the above implementations based on neural networks, the pseudocode of the proposed ADP approach for solving the tracking problem of nonlinear system (1) is illustrated in Algorithm 1.

Algorithm 1 Error-based ADP for solving the tracking problem of nonlinear discrete-time systems

Initialization: Set the numbers of hidden-layer neurons of the neural networks and randomly initialize the weight matrices and the bias vectors; set the initial performance indices V_k^0 and $\bar{V}_k^0$, the variable $\boldsymbol{\nu}_k^0$, the control input $\boldsymbol{u}_k^0$, the initial iterative indices $\bar{\imath}=0$ and $\ell=0$, the maximum iterative indices $\bar{\imath}_v$ and $\ell_{\bar{v}}$, and the accuracies ξ_v and $\xi_{\bar{v}}$.
Output: The control input $\boldsymbol{u}_k^*$ and predicted trajectory $\boldsymbol{s}_{k+1}$.
Train the neural networks of the system models based on historical data until the maximum iteration indices are reached or the accuracy requirements are met;
while true **do**
  Calculate the iterative variable $\boldsymbol{\nu}_k^{\bar{\imath}}$ using (18) and then train the neural network $\hat{\boldsymbol{\nu}}_k$ based on $\boldsymbol{\nu}_k^{\bar{\imath}}$;
  Calculate the iterative index $V_k^{\bar{\imath}+1}$ using (19) and then train the neural network $\hat{V}_k$ based on $V_k^{\bar{\imath}+1}$;
  if $\|\boldsymbol{W}_{v1}^{\bar{\imath}+1}-\boldsymbol{W}_{v1}^{\bar{\imath}}\|+\|\boldsymbol{W}_{v2}^{\bar{\imath}+1}-\boldsymbol{W}_{v2}^{\bar{\imath}}\|+\|\boldsymbol{b}_{v1}^{\bar{\imath}+1}-\boldsymbol{b}_{v1}^{\bar{\imath}}\|+\|\boldsymbol{b}_{v2}^{\bar{\imath}+1}-\boldsymbol{b}_{v2}^{\bar{\imath}}\|\leqslant\xi_v$ or $\bar{\imath}_v\leqslant\bar{\imath}$ **then**
    Set $\boldsymbol{W}_{v1}^{\bar{\imath}+1}$, $\boldsymbol{W}_{v2}^{\bar{\imath}+1}$, $\boldsymbol{b}_{v1}^{\bar{\imath}+1}$, and $\boldsymbol{b}_{v2}^{\bar{\imath}+1}$ to $\boldsymbol{W}_{v1}^*$, $\boldsymbol{W}_{v2}^*$, $\boldsymbol{b}_{v1}^*$, and $\boldsymbol{b}_{v2}^*$, respectively;
    Set $\boldsymbol{W}_{\nu1}^{\bar{\imath}}$, $\boldsymbol{W}_{\nu2}^{\bar{\imath}}$, $\boldsymbol{b}_{\nu1}^{\bar{\imath}}$, and $\boldsymbol{b}_{\nu2}^{\bar{\imath}}$ to $\boldsymbol{W}_{\nu1}^*$, $\boldsymbol{W}_{\nu2}^*$, $\boldsymbol{b}_{\nu1}^*$, and $\boldsymbol{b}_{\nu2}^*$, respectively;
    break;
  end if
end while
Calculate the predicted trajectory $\boldsymbol{s}_{k+1}=\boldsymbol{x}_k+\boldsymbol{\nu}_k$ based on $\boldsymbol{W}_{\nu1}^*$, $\boldsymbol{W}_{\nu2}^*$, $\boldsymbol{b}_{\nu1}^*$, and $\boldsymbol{b}_{\nu2}^*$;
while true **do**
  Calculate the iterative variable $\boldsymbol{u}_k^{\ell}$ using (32) and then train the neural network $\hat{\boldsymbol{u}}_k$ based on $\boldsymbol{u}_k^{\ell}$;
  Calculate the iterative index $\bar{V}_k^{\ell+1}$ using (33) and then train the neural network $\hat{\bar{V}}_k$ based on $\bar{V}_k^{\ell+1}$;
  if $\|\boldsymbol{W}_{\bar{v}1}^{\ell+1}-\boldsymbol{W}_{\bar{v}1}^{\ell}\|+\|\boldsymbol{W}_{\bar{v}2}^{\ell+1}-\boldsymbol{W}_{\bar{v}2}^{\ell}\|+\|\boldsymbol{b}_{\bar{v}1}^{\ell+1}-\boldsymbol{b}_{\bar{v}1}^{\ell}\|+\|\boldsymbol{b}_{\bar{v}2}^{\ell+1}-\boldsymbol{b}_{\bar{v}2}^{\ell}\|\leqslant\xi_{\bar{v}}$ or $\ell_{\bar{v}}\leqslant\ell$ **then**
    Set $\boldsymbol{W}_{\bar{v}1}^{\ell+1}$, $\boldsymbol{W}_{\bar{v}2}^{\ell+1}$, $\boldsymbol{b}_{\bar{v}1}^{\ell+1}$, and $\boldsymbol{b}_{\bar{v}2}^{\ell+1}$ to $\boldsymbol{W}_{\bar{v}1}^*$, $\boldsymbol{W}_{\bar{v}2}^*$, $\boldsymbol{b}_{\bar{v}1}^*$, and $\boldsymbol{b}_{\bar{v}2}^*$, respectively;
    Set $\boldsymbol{W}_{u1}^{\ell}$, $\boldsymbol{W}_{u2}^{\ell}$, $\boldsymbol{b}_{u1}^{\ell}$, and $\boldsymbol{b}_{u2}^{\ell}$ to $\boldsymbol{W}_{u1}^*$, $\boldsymbol{W}_{u2}^*$, $\boldsymbol{b}_{u1}^*$, and $\boldsymbol{b}_{u2}^*$, respectively;
    break;
  end if
end while
Calculate the control input $\boldsymbol{u}_k^*$ based on $\boldsymbol{W}_{u1}^*$, $\boldsymbol{W}_{u2}^*$, $\boldsymbol{b}_{u1}^*$, and $\boldsymbol{b}_{u2}^*$.

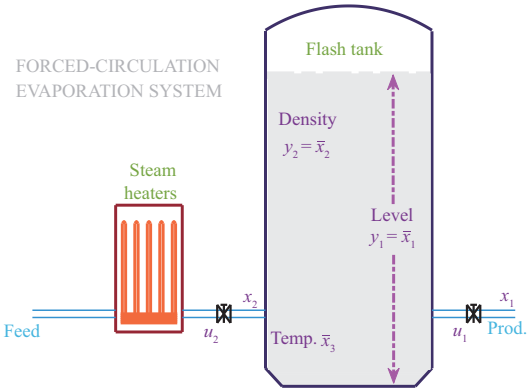

Figure 1 (Color online) Forced-circulation evaporation system in an alumina production process.

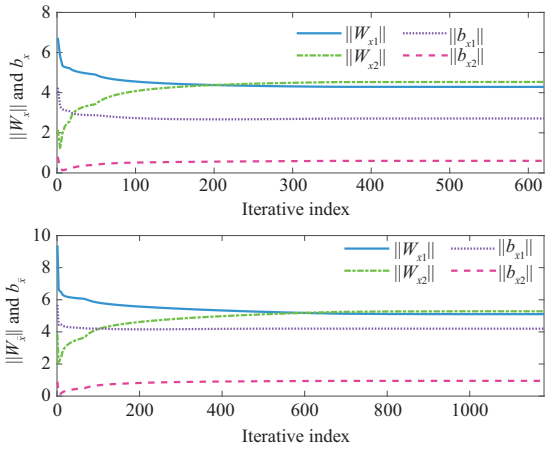

Figure 2 (Color online) Curves of normalized weights and normalized bias of neural networks $\hat{\boldsymbol{x}}_{k+1}$ and $\hat{\bar{\boldsymbol{x}}}_{k+1}$.

5 Numeral simulation

In this section, a numeral example is provided to verify the effectiveness of the proposed approach, which is based on alumina data in an actual industrial production process. In this simulation, we conducted comparative simulations and simulated the tracking effect in the presence of disturbances.

As described in system (1), the system state $\boldsymbol{x}_k$ comprises the discharge liquor flow $x_{1,k}$ and the heat steam flow $x_{2,k}$, which are controlled by the system input $\boldsymbol{u}_k$ consisting of the discharge valve opening $u_{1,k}$ and the heat steam valve opening $u_{2,k}$. The system state $\bar{\boldsymbol{x}}_k$ contains three variables directly affected by the system state $\boldsymbol{x}_k$: the flash tank liquid level $\bar{x}_{1,k}$, the liquid discharge density $\bar{x}_{2,k}$, and the liquid discharge temperature $\bar{x}_{3,k}$. As the evaluation indicators are closely related to production, the flash tank liquid level $\bar{x}_{1,k}$ and liquid discharge density $\bar{x}_{2,k}$ constitute the system output; that is, $\boldsymbol{y}_k=[\bar{x}_{1,k};\ \bar{x}_{2,k}]$. The simple structure of the forced-circulation evaporation system in alumina production processes is presented in Figure 1, where the system state, control input, and output are illustrated. Then, according to the above explanations and using historical data, the neural network models of a controlled plant can be trained using expressions (41)–(43). Figure 2 illustrates the iterative curves of the 2-norm of the weight matrices. Considering that the system output vector is directly composed of $\bar{x}_{1,k}$ and $\bar{x}_{2,k}$ in

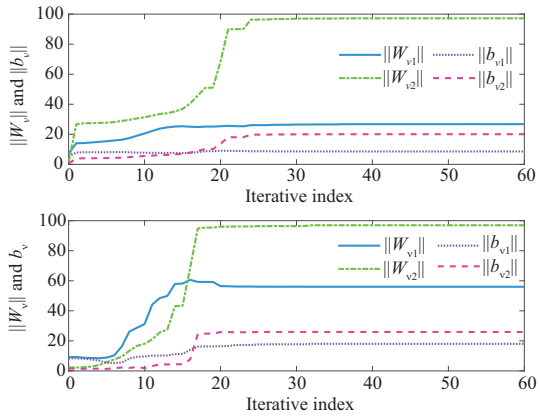

Figure 3 (Color online) Curves of normalized weights and normalized bias of neural networks $\hat{\boldsymbol{V}}$ and $\hat{\boldsymbol{\nu}}$.

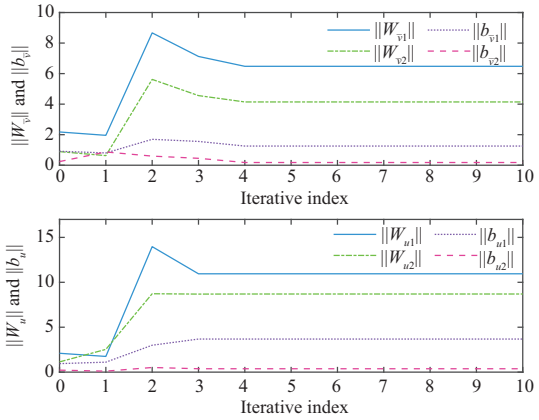

Figure 4 (Color online) Curves of normalized weights and normalized bias of neural networks $\hat{\boldsymbol{V}}$ and $\hat{\boldsymbol{u}}$.

this simulation, a neural network model needs not to be separately provided with respect to the system output equation.

Notably, in actual production processes, the change amplitude between the states $\boldsymbol{x}_k$ and $\boldsymbol{x}_{k+1}$ is affected by the adjustment range of the two valve openings in the adjacent time (the interval between adjacent states is 0.03 s). Hence, as described in Section 3, there exists a restricted requirement $\boldsymbol{\nu}_k \in (\underline{\boldsymbol{\nu}}, \overline{\boldsymbol{\nu}})$, indicating that the definition of performance index (2) is reasonable in practical applications. Drawing on historical experience and data, let the parameters $\underline{\boldsymbol{\nu}}$ and $\overline{\boldsymbol{\nu}}$ be set to $\underline{\boldsymbol{\nu}} = [0.7000; 0.1900]$ and $\overline{\boldsymbol{\nu}} = [-0.4800; -0.1900]$, respectively. Thereby, the neural networks of the performance index V_k and variable $\boldsymbol{\nu}_k$ can be trained according to lines 4–12 of Algorithm 1, where the variable $\boldsymbol{\nu}_k$ is used to predict the reference trajectory of the system state $\boldsymbol{x}_{k+1}$. In Figure 3, the curves represent the iterative values of the 2-norm of all the weight matrices in these two neural networks. Thereafter, the optimal weight matrices of the neural networks that realize the performance index $\bar{V}_k$ and control policy $\boldsymbol{u}_k$ can be obtained using lines 14–22 of Algorithm 1. In the iterative process, the updated values of the 2-norm of the weight matrices in these neural networks are illustrated in Figure 4.

In actual production processes, the reference trajectory $\boldsymbol{\eta}$ in a certain period is a fixed vector. When and only when the production requirements change, the reference trajectory is adjusted to another fixed vector. Thus, in the simulation, the reference trajectories are designated as $[2.0; 1420]$, $[1.9; 1440]$, $[2.0; 1430]$, and $[2.1; 1420]$ in turn. Moreover, the initial system states are assumed to be $\boldsymbol{x}_0 = [68; 11]$ and $\bar{\boldsymbol{x}}_0 = [1.85; 1415; 110]$, and the control input is $\boldsymbol{u}_0 = [45; 20]$. Notably, the change amounts between $\boldsymbol{u}_k$ and $\boldsymbol{u}_{k+1}$ must be limited to a certain range. According to experience and data, let the maximum amplitude of $\|u_{\bar{\ell},k} - u_{\bar{\ell},k-1}\|$, $\bar{\ell} \in \{1, 2\}$ be set to 4. Therefore, when the control input $u_{\bar{\ell},k}$ calculated by the obtained control policy satisfies $\|u_{\bar{\ell},k} - u_{\bar{\ell},k-1}\| > 4$, one has $u_{\bar{\ell},k} = u_{\bar{\ell},k-1} + 4\mathtt{sgn}(u_{\bar{\ell},k} - u_{\bar{\ell},k-1})$, where $\mathtt{sgn}(\cdot)$ represents the sign function. Further, note that both the discharge valve opening and the heat steam valve opening range from 0% to 100%. Hence, whenever a value in the control input solved by the obtained control policy exceeds the upper limit, the value is set to 100%. Similarly, if a value in the calculated control input is below the lower limit, the value is set to 0%. Then, using the above criteria of the control input and applying the obtained control policies $\boldsymbol{u}_k$ and $\boldsymbol{\nu}_k$ into the plant, the curves of entries in the output vector are shown in the two upper subgraphs of Figure 5, where the dotted lines represent the reference trajectories. The two lower subgraphs of Figure 5 illustrate the system output and the reference trajectories when the state of the controlled plant is disturbed, where the disturbance is assumed to be Gaussian noise with zero mean, to further demonstrate the control effectiveness of the proposed ADP approach. Correspondingly, Figures 6 and 7 respectively display the curves of the control input and system states in the absence and presence of the Gaussian disturbance. The two upper subgraphs of Figure 5 show that the system output can track the given reference trajectory, which means that the proposed algorithm can enable the system output to stably track the reference trajectories. Meanwhile, the two lower subgraphs of Figure 5 clearly demonstrate that the system output can be maintained near the given reference trajectories, indicating that the proposed algorithm can effectively deal with such tracking problems under disturbances.

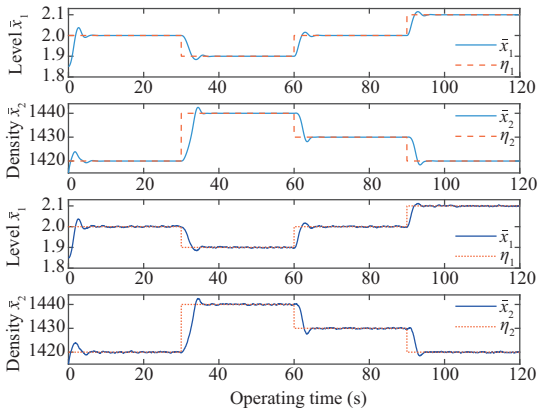

Figure 5 (Color online) Curves of system outputs and reference trajectories.

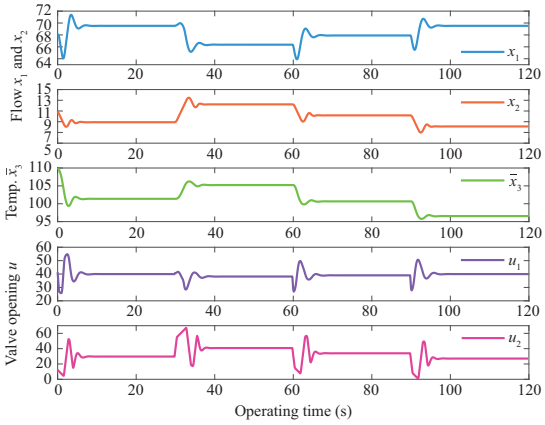

Figure 6 (Color online) Curves of system states x_1, x_2, and $\bar{x}_3$ and the control inputs u_1 and u_2 without disturbance.

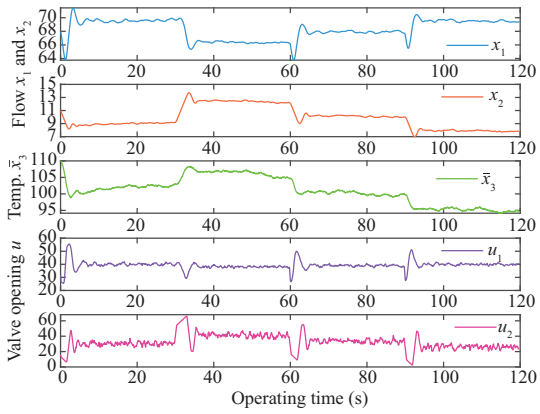

Figure 7 (Color online) Curves of system states x_1, x_2, and $\bar{x}_3$ and the control inputs u_1 and u_2 under disturbance.

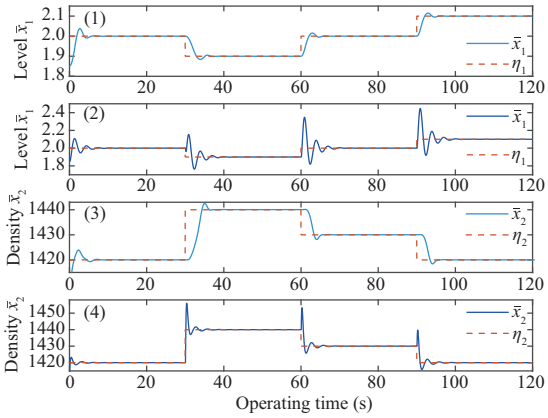

Figure 8 (Color online) Curves of system outputs and reference trajectories in the comparative simulation.

Generally, the utility function of the ADP approach for solving tracking problems is defined by the addition of the tracking error quadratic form and control input quadratic form; then, the boundedness of the performance index is ensured by a discount factor. Notably, as described by Kiumarsi and Lewis [12], the control policy obtained using such a performance index cannot theoretically eliminate the tracking errors. Therefore, our proposed approach cannot be compared with the ADP approach based on such a utility function. For comparison purposes, the PI controller is considered in the simulation. As presented by Jiang et al. [44], two PI controllers were designed in the device layer and operational layer, which could be intuitively understood from the structure in Figure 2 in this previous study [44]. In the simulation, the PI parameters in the device layer controller were $K_{P1} = \texttt{diag}\ \{0.9976, 4.9131\}$ and $K_{I1} = \texttt{diag}\ \{0.1903, 1.1789\}$, and the parameters in the operational layer controller were $K_{P2} = \texttt{diag}\ \{-38.7623, 1.2655\}$ and $K_{I2} = \texttt{diag}\ \{-5.8193, 0.1092\}$, where $\texttt{diag}\{a, b\}$ denotes a diagonal matrix, with diagonal elements being a and b. The simulation results obtained by our proposed approach are illustrated in Figures 8(1) and (3), and the results simulated by the comparative method are plotted in Figures 8(2) and (4).

The adjustment time and overshoot of the flash tank liquid level in Figure 8(1) are smaller than those in Figure 8(2). Similarly, for the liquid discharge density, Figures 8(3) and (4) show that the proposed algorithm has superiority over the comparative method as regards the adjustment time and overshoot. The main reason for this superiority is that the control policy obtained by our proposed approach has a relatively global optimization perspective, whereas that of the comparative method was formulated by PI controllers determined using only a few parameters.

6 Conclusion

In this paper, a novel ADP approach was proposed for solving the tracking problem of nonlinear discrete-time systems. As the controlled plant could be divided into two subsystems as described in system (1), a performance index was defined to predict the reference trajectory of state $\boldsymbol{x}_k$, which was designed from the perspective of completely eliminating errors in theory. Hence, we proved the effectiveness of minimizing the proposed performance index to obtain the reference trajectory of state $\boldsymbol{x}_k$ and further deduced that the minimization process can be realized using a value-iteration-based ADP approach. In addition, solving the control input could be regarded as dealing with the tracking problem that makes the system state $\boldsymbol{x}_k$ track the predicted reference trajectory, which could also be solved using the ADP approach. Finally, using a data-driven technique, we implemented the proposed ADP approach using neural networks. As the tracking problem described by system (1) is common in actual industrial productions, the proposed ADP approach is of great significance.

Acknowledgements This work was supported by National Key R&D Plan Project (Grant No. 2022YFB3304700), National Natural Science Foundation of China (Grant Nos. 61988101, 62161160338), 111 Project 2.0 (Grant No. B08015), and Liaoning Province Central Leading Local Science and Technology Development Special Project (Grant No. 2022JH6/100100055).

References

1 Kiumarsi B, Vamvoudakis K G, Modares H, et al. Optimal and autonomous control using reinforcement learning: a survey. IEEE Trans Neural Netw Learn Syst, 2018, 29: 2042–2062
2 Lewis F L, Liu D. Reinforcement Learning and Approximate Dynamic Programming for Feedback Control. Hoboken: John Wiley & Sons, 2013
3 Jiang Y, Jiang Z P. Robust Adaptive Dynamic Programming. Hoboken: John Wiley & Sons, 2017
4 Liu D, Xue S, Zhao B, et al. Adaptive dynamic programming for control: a survey and recent advances. IEEE Trans Syst Man Cybern Syst, 2021, 51: 142–160
5 Wang D, Mu C X. Developing nonlinear adaptive optimal regulators through an improved neural learning mechanism. Sci China Inf Sci, 2017, 60: 058201
6 Lewis F L, Vrabie D. Reinforcement learning and adaptive dynamic programming for feedback control. IEEE Circuits Syst Mag, 2009, 9: 32–50
7 Li C, Ding J, Lewis F L, et al. A novel adaptive dynamic programming based on tracking error for nonlinear discrete-time systems. Automatica, 2021, 129: 109687
8 Modares H, Lewis F L, Naghibi-Sistani M B. Adaptive optimal control of unknown constrained-input systems using policy iteration and neural networks. IEEE Trans Neural Netw Learn Syst, 2013, 24: 1513–1525
9 Ha M, Wang D, Liu D. Discounted iterative adaptive critic designs with novel stability analysis for tracking control. IEEE CAA J Autom Sin, 2022, 9: 1262–1272
10 Gao W, Jiang Z P, Lewis F L, et al. Leader-to-formation stability of multiagent systems: an adaptive optimal control approach. IEEE Trans Automat Contr, 2018, 63: 3581–3587
11 Vamvoudakis K G, Mojoodi A, Ferraz H. Event-triggered optimal tracking control of nonlinear systems. Int J Robust Nonlinear Control, 2017, 27: 598–619
12 Kiumarsi B, Lewis F L. Actor-critic-based optimal tracking for partially unknown nonlinear discrete-time systems. IEEE Trans Neural Netw Learn Syst, 2015, 26: 140–151
13 Wang D, Mu C. Adaptive-critic-based robust trajectory tracking of uncertain dynamics and its application to a spring-mass-damper system. IEEE Trans Ind Electron, 2018, 65: 654–663
14 Liu D, Wei Q, Wang D, et al. Adaptive Dynamic Programming With Application in Optimal Control. Berlin: Springer, 2017
15 Li X X, Peng Z H, Liang L, et al. Policy iteration based Q-learning for linear nonzero-sum quadratic differential games. Sci China Inf Sci, 2019, 62: 052204
16 Li X X, Peng Z H, Jiao L, et al. Online adaptive Q-learning method for fully cooperative linear quadratic dynamic games. Sci China Inf Sci, 2019, 62: 222201
17 Zhang H, Liu C, Su H, et al. Echo state network-based decentralized control of continuous-time nonlinear large-scale interconnected systems. IEEE Trans Syst Man Cybern Syst, 2021, 51: 6293–6303
18 Yang Y, Gao W, Modares H, et al. Robust actor-critic learning for continuous-time nonlinear systems with unmodeled dynamics. IEEE Trans Fuzzy Syst, 2022, 30: 2101–2112
19 Wei Q, Liao Z, Shi G. Generalized actor-critic learning optimal control in smart home energy management. IEEE Trans Ind Inf, 2021, 17: 6614–6623
20 Gao W, Gao J, Ozbay K, et al. Reinforcement-learning-based cooperative adaptive cruise control of buses in the Lincoln tunnel corridor with time-varying topology. IEEE Trans Intell Transp Syst, 2019, 20: 3796–3805
21 Zhao J, Yang C, Dai W, et al. Reinforcement learning-based composite optimal operational control of industrial systems with multiple unit devices. IEEE Trans Ind Inf, 2022, 18: 1091–1101
22 Gao W, Jiang Z P, Ozbay K. Data-driven adaptive optimal control of connected vehicles. IEEE Trans Intell Transp Syst, 2017, 18: 1122–1133
23 Chen C, Modares H, Xie K, et al. Reinforcement learning-based adaptive optimal exponential tracking control of linear systems with unknown dynamics. IEEE Trans Automat Contr, 2019, 64: 4423–4438
24 Arbib M A. The Handbook of Brain Theory and Neural Networks. Cambridge: MIT Press, 2003
25 Wang X, Ding D, Dong H, et al. Neural-network-based control for discrete-time nonlinear systems with input saturation under stochastic communication protocol. IEEE CAA J Autom Sin, 2021, 8: 766–778
26 Li X, Dong L, Sun C. Data-based optimal tracking of autonomous nonlinear switching systems. IEEE CAA J Autom Sin, 2021, 8: 227–238
27 Mu C, Wang D, He H. Novel iterative neural dynamic programming for data-based approximate optimal control design. Automatica, 2017, 81: 240–252

28 Wang D, Ha M, Qiao J. Data-driven iterative adaptive critic control toward an urban wastewater treatment plant. IEEE Trans Ind Electron, 2021, 68: 7362–7369
29 Jiang Y, Fan J, Chai T, et al. Dual-rate operational optimal control for flotation industrial process with unknown operational model. IEEE Trans Ind Electron, 2019, 66: 4587–4599
30 Yang Y, Ding Z, Wang R, et al. Data-driven human-robot interaction without velocity measurement using off-policy reinforcement learning. IEEE CAA J Autom Sin, 2022, 9: 47–63
31 Dong L, Yuan X, Sun C Y. Event-triggered receding horizon control via actor-critic design. Sci China Inf Sci, 2020, 63: 150210
32 Wang D, He H, Zhong X, et al. Event-driven nonlinear discounted optimal regulation involving a power system application. IEEE Trans Ind Electron, 2017, 64: 8177–8186
33 Wei Q, Li T, Liu D. Learning control for air conditioning systems via human expressions. IEEE Trans Ind Electron, 2021, 68: 7662–7671
34 Jiang Y, Shi D, Fan J, et al. Set-valued feedback control and its application to event-triggered sampled-data systems. IEEE Trans Automat Contr, 2020, 65: 4965–4972
35 Zhang H, Li Y, Gao D W, et al. Distributed optimal energy management for energy internet. IEEE Trans Ind Inf, 2017, 13: 3081–3097
36 Wang D, He H, Liu D. Intelligent optimal control with critic learning for a nonlinear overhead crane system. IEEE Trans Ind Inf, 2018, 14: 2932–2940
37 Song X L, Ding F, Xiao F, et al. Data-driven optimal cooperative adaptive cruise control of heterogeneous vehicle platoons with unknown dynamics. Sci China Inf Sci, 2020, 63: 190204
38 Wang Y, Chai T, Fu J, et al. Adaptive decoupling switching control of the forced-circulation evaporation system using neural networks. IEEE Trans Contr Syst Technol, 2013, 21: 964–974
39 Ding J, Modares H, Chai T, et al. Data-based multiobjective plant-wide performance optimization of industrial processes under dynamic environments. IEEE Trans Ind Inf, 2016, 12: 454–465
40 Li Y, Min X, Tong S. Adaptive fuzzy inverse optimal control for uncertain strict-feedback nonlinear systems. IEEE Trans Fuzzy Syst, 2020, 28: 2363–2374
41 Li Y, Min X, Tong S. Observer-based fuzzy adaptive inverse optimal output feedback control for uncertain nonlinear systems. IEEE Trans Fuzzy Syst, 2021, 29: 1484–1495
42 Jiang Y, Fan J, Chai T, et al. Data-driven flotation industrial process operational optimal control based on reinforcement learning. IEEE Trans Ind Inf, 2018, 14: 1974–1989
43 Lu X, Krstić M, Chai T, et al. Hardware-in-the-loop multiobjective extremum-seeking control of mineral grinding. IEEE Trans Contr Syst Technol, 2021, 29: 961–971
44 Jiang Y, Jia Y, Fan J, et al. Compensation-signal-based dual-rate operational feedback decoupling control for flotation processes. IEEE Trans Ind Electron, 2022, 69: 8306–8316