Neural Networks 55 (2014) 30–41

Contents lists available at ScienceDirect

Neural Networks

journal homepage: www.elsevier.com/locate/neunet

# Discrete-time online learning control for a class of unknown nonaffine nonlinear systems using reinforcement learning☆

Xiong Yang, Derong Liu*, Ding Wang, Qinglai Wei

*The State Key Laboratory of Management and Control for Complex Systems, Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China*

ARTICLE INFO

*Article history:*
Received 14 August 2013
Received in revised form 8 February 2014
Accepted 20 March 2014
Available online 28 March 2014

*Keywords:*
Adaptive critic design
Neural network
Nonaffine nonlinear system
Online learning
Reinforcement learning

ABSTRACT

In this paper, a reinforcement-learning-based direct adaptive control is developed to deliver a desired tracking performance for a class of discrete-time (DT) nonlinear systems with unknown bounded disturbances. We investigate multi-input–multi-output unknown nonaffine nonlinear DT systems and employ two neural networks (NNs). By using Implicit Function Theorem, an action NN is used to generate the control signal and it is also designed to cancel the nonlinearity of unknown DT systems, for purpose of utilizing feedback linearization methods. On the other hand, a critic NN is applied to estimate the cost function, which satisfies the recursive equations derived from heuristic dynamic programming. The weights of both the action NN and the critic NN are directly updated online instead of offline training. By utilizing Lyapunov's direct method, the closed-loop tracking errors and the NN estimated weights are demonstrated to be uniformly ultimately bounded. Two numerical examples are provided to show the effectiveness of the present approach.



## 1. Introduction

Adaptive control theory has been an active area of research for several decades, which aims to find stable controllers for nonlinear dynamic systems (Chemachema, 2012; Chen & Khalil, 1995; Ge, Hang, & Zhang, 1999; Lewis, Yesildirek, & Liu, 1996; Liu, Venayagamoorthy, & Wunsch, 2003; Nakanishi & Schaal, 2004; Narendra & Mukhopadhyay, 1994). Nevertheless, stability is only a bare minimum requirement in a system design. The optimality based on a prescribed cost function is usually taken into consideration for control problems of nonlinear systems. In other words, control schemes should be proposed to guarantee the stability of the closed-loop system, while keeping the cost function as small as possible.

In order to derive such a controller, large amounts of significant methods have been proposed. Among these approaches, dynamic programming (DP) has been widely applied to generate optimal control for nonlinear systems by employing Bellman's principle of optimality (Bellman, 1957). The method guarantees to perform optimization backward-in-time. However, a serious shortcoming about DP is that the computation is untenable to be run with the increasing dimension of nonlinear systems, which is the well-known "curse of dimensionality". Moreover, the backward direction of search obviously prohibits the wide use of DP in real-time control. On the other hand, with considerable investigations engaged in artificial neural networks (NNs), researchers find NNs can successfully be applied to intelligent control due to their properties of nonlinearity, adaptivity, self-learning, and fault tolerance (Haykin, 2008; Yu, 2009). Consequently, NNs are extensively utilized for universal function approximation in adaptive dynamic programming (ADP) algorithms, which were proposed by Werbos (1991, 1992, 2007, 2008), as methods to solve optimal control problems forward-in-time. There are several synonyms used for ADP including "adaptive dynamic programming" (Liu, Wang, & Yang, 2013; Liu & Wei, 2013; Liu, Zhang, & Zhang, 2005; Murray, Cox, Lendaris, & Saeks, 2002; Wang, Liu, & Wei, 2012; Wang, Liu, Wei, Zhao, & Jin, 2012; Wang, Zhang, & Liu, 2009; Wei & Liu, 2012; Zhang, Wei, & Liu, 2011), "approximate dynamic programming" (Al-Tamimi, Lewis, & Abu-Khalaf, 2008), "adaptive critic designs" (ACDs) (Prokhorov & Wunsch, 1997), "neuro-dynamic programming" (NDP) (Bertsekas & Tsitsiklis, 1996), and "neural dynamic programming" (Si & Wang, 2001). Furthermore, according to Prokhorov and Wunsch

☆ This work was supported in part by the National Natural Science Foundation of China under Grants 61034002, 61233001, 61273140, 61304086, and 61374105, and in part by Beijing Natural Science Foundation under Grant 4132078.

* Corresponding author. Tel.: +86 10 82544761; fax: +86 10 82544799.
*E-mail addresses:* xiong.yang@ia.ac.cn (X. Yang), derongliu@gmail.com, derong.liu@ia.ac.cn (D. Liu), ding.wang@ia.ac.cn (D. Wang), qinglai.wei@ia.ac.cn (Q. Wei).

http://dx.doi.org/10.1016/j.neunet.2014.03.008
0893-6080/

(1997) and Werbos (1992), ADP algorithms are mainly classified as follows: heuristic dynamic programming (HDP), dual heuristic programming (DHP), globalized dual heuristic programming (GDHP). When the action is introduced as an additional input to the critic, ACDs are referred to action dependent version of the ACDs, such as action dependent HDP (ADHDP), action dependent DHP (ADDHP), and action dependent GDHP (ADGDHP).

Unfortunately, most of ADP algorithms are implemented either by an offline process via iterative schemes or need a priori knowledge of dynamics of nonlinear systems. Since the exact knowledge of nonlinear systems is often unavailable, it brings about great challenges to implement these algorithms. In order to overcome the difficulty, reinforcement learning (RL) is introduced to cope with optimal control problems. RL is a class of approaches used in machine learning to methodically revise the actions of an agent based on responses from its environment (Sutton & Barto, 1998). A distinct difference between the traditional supervised NN learning and RL is that, there is no prescribed behavior or training model proposed to RL schemes. If the cost function is viewed as the reinforcement signal, then ADP algorithms become RL approaches. Therefore, ADP algorithms are actually a class of RL methods (Lewis & Vamvoudakis, 2011; Lewis, Vrabie, & Vamvoudakis, 2012). Since RL shares considerable common features with ADP algorithms, it is often employed for adaptive optimal controller designs.

Applications of RL methods to feedback control have been widely investigated in the literature (Bhasin et al., 2013; He & Jagannathan, 2005; Lewis, Lendaris, & Liu, 2008; Lewis & Vamvoudakis, 2011; Liu, Yang, & Li, 2013; Vamvoudakis & Lewis, 2010, 2011; Yang & Jagannathan, 2012; Yang, Liu, & Huang, 2013; Yang, Si, Tsakalis, & Rodriguez, 2009). In He and Jagannathan (2005), an RL-based output feedback control was developed for multi-input–multi-output (MIMO) unknown affine nonlinear DT systems. By using Lyapunov's direct approach, the estimated state errors, the tracking errors and the NN estimated weights were all guaranteed to be uniformly ultimately bounded (UUB). After that, in Yang et al. (2009), a direct HDP was proposed to obtain online learning control for MIMO unknown affine nonlinear DT systems. With the aid of Lyapunov's direct method, the uniform ultimate boundedness of both the closed-loop tracking errors and the NN estimated weights was derived. Just as mentioned above, in this literature, the authors took the cost function as the reinforcement signal. Recently, in Vamvoudakis and Lewis (2010), an online algorithm based on RL for affine nonlinear continuous-time (CT) systems was proposed. By employing the algorithm, both the optimal cost and the optimal control were well approximated in real time, while guaranteeing the uniform ultimate boundedness of the closed-loop system. In addition, the NN estimated weights were guaranteed to be UUB by using Lyapunov's direct method. More recently, in Vamvoudakis and Lewis (2011), RL methods were also applied to multi-player differential games for nonlinear CT systems. Based on Lyapunov's direct method, the uniform ultimate boundedness of both the closed-loop system and the NN estimated weights was demonstrated.

However, all of them deal with feedback control problems of RL methods for *affine* nonlinear systems. To the best of our knowledge, there are rather few investigations on feedback control of RL approaches for *nonaffine* nonlinear systems, especially MIMO unknown nonaffine nonlinear DT systems. Though there exist some researches about nonaffine nonlinear DT systems (Deng, Li, & Wu, 2008; Noriega & Wang, 1998; Yang, Vance, & Jagannathan, 2008), most of them focus on feedback control problems of nonlinear autoregressive moving average with exogenous inputs (NARMAX) systems. This form is less convenient than the state-form of nonaffine nonlinear systems for purpose of adaptive control using NNs. On the other hand, since the output of *affine* nonlinear systems is linear with respect to the control input, it is easy to design a controller to follow prescribed trajectories by using feedback linearization methods. Nevertheless, feedback linearization approaches cannot be implemented for *nonaffine* nonlinear systems, for the output of this type of systems depends nonlinearly on the control signal. It gives rise to great difficulties for researchers to design an efficient controller of such a nonaffine nonlinear system, which aims at achieving desired trajectories. Furthermore, in real engineering, control approaches of affine nonlinear systems do not always hold and control methods for nonaffine nonlinear systems are necessary. Therefore, control problems of RL methods for unknown nonaffine nonlinear systems are very significant in both theory and applications.

The objective of this paper is to develop an online direct adaptive control based on RL methods by delivering a desired tracking performance for MIMO unknown nonaffine nonlinear DT systems with unknown bounded disturbances. Two NNs are employed in the controller design: an action NN is utilized to generate the control signal. Meanwhile, by using Implicit Function Theorem, the action NN approximation is well designed to cancel the nonlinearity of unknown nonlinear DT systems, for purpose of utilizing feedback linearization methods. A critic NN is used to estimate the prescribed cost function, which satisfies the recursive equations derived from HDP. The weights of both the action NN and the critic NN are directly updated online instead of preliminary offline training. By using Lyapunov's direct method, the closed-loop tracking errors and the NN estimated weights are verified to be UUB.

The main contributions of the paper include the following:

1. To the best of our knowledge, it is the first time that an online RL-based direct adaptive control is developed for the state-form of MIMO unknown nonaffine nonlinear DT systems with unknown bounded disturbances.
2. Compared with He and Jagannathan (2005), Yang et al. (2009), and Yang and Jagannathan (2012), we consider nonaffine nonlinear DT systems with unknown system drift dynamics. A significant difference between these literature and the present paper is that, in our case, the adaptive control is developed based on Implicit Function Theorem and RL methods since feedback linearization methods cannot be directly implemented for nonaffine nonlinear DT systems.

The rest of the paper is organized as follows. Section 2 provides the problem statement and preliminaries. Section 3 develops an online adaptive control by using RL approaches. Section 4 shows the stability analysis and the performance of the closed-loop systems. Section 5 presents two simulation results to verify the effectiveness of the established theory. Finally, Section 6 gives several concluding remarks.

For convenience, we introduce the notations, which will be used throughout the paper.

- $\mathbb{R}$ denotes the real numbers, $\mathbb{R}^m$ and $\mathbb{R}^{m\times n}$ denote the real $m$-vectors and the real $m \times n$ matrices, respectively. $\otimes$ denotes the Kronecker product. If there is no special explanation, $\mathrm{T}$ is a transposition symbol.
- $\Omega$ is a compact set of $\mathbb{R}^{mn}$, $C^m(\Omega) = \left\{f^{(m)} \in C | f: \Omega \to \mathbb{R}^m\right\}$. Let $\Omega_i \subset \Omega$ $(i = 1, 2)$, $\Omega_1 \times \Omega_2 = \left\{(x, y) | x \in \Omega_1, y \in \Omega_2\right\}$ stands for the Cartesian product of $\Omega_1$ and $\Omega_2$.
- $\|\cdot\|$ stands for any suitable norm. When $z$ is a vector, $\|z\|$ denotes the Euclidean norm of $z$. When $A$ is a matrix, $\|A\|$ denotes the 2-norm of $A$.

## 2. Problem statement and preliminaries

### 2.1. Dynamics of nonaffine nonlinear DT systems

For purpose of the present paper, we consider an $mn$th-order MIMO nonaffine nonlinear DT plant of the form

$$\begin{aligned} x_1(k+1) &= x_2(k) \\ &\vdots \\ x_{n-1}(k+1) &= x_n(k) \\ x_n(k+1) &= h\big(x(k), u(x(k))\big) + d(k) \\ y(k) &= x_1(k) \end{aligned} \tag{1}$$

with the state $x(k) = \left[x_1^{\mathrm{T}}(k), x_2^{\mathrm{T}}(k), \ldots, x_n^{\mathrm{T}}(k)\right]^{\mathrm{T}} \in \mathbb{R}^{mn}$, and each $x_i(k) \in \mathbb{R}^m$, $i = 1, 2, \ldots, n$. $u(x(k)) \in \mathbb{R}^m$ is the control input, which is a continuous function with respect to $x(k)$. For convenience, we denote $\upsilon(k) = u(x(k))$. $d(k) \in \mathbb{R}^m$ is an unknown disturbance bounded by a known constant $d_M > 0$, i.e., $\|d(k)\| \le d_M$. $h\big(x(k), \upsilon(k)\big) \in \mathbb{R}^m$ is an unknown nonaffine nonlinear function with $h(0, 0) = 0$, and $y(k) \in \mathbb{R}^m$ is the system output. In order to make the controllability of the system, we provide the assumptions as follows.

**Assumption 1.** The state $x(k)$ is available from measurement at the $k$-th step for the state-feedback control.

**Assumption 2.** The $m \times m$ matrix $\partial h\big(x(k), \upsilon(k)\big)/\partial \upsilon(k)$ is positive definite. It implies

$$\det\left[\frac{\partial h\big(x(k), \upsilon(k)\big)}{\partial \upsilon(k)}\right] \neq 0 \tag{2}$$

for $\forall\, \big(x(k), \upsilon(k)\big) \in \Omega \times \mathbb{R}^m$ with a compact region $\Omega \subset \mathbb{R}^{mn}$.

**Assumption 3.** Let the desired trajectory of system (1) be $x_d(k) = \left[x_{1d}^{\mathrm{T}}(k), x_{2d}^{\mathrm{T}}(k), \ldots, x_{nd}^{\mathrm{T}}(k)\right]^{\mathrm{T}} \in \mathbb{R}^{mn}$, where $x_{id}(k)$ is arbitrarily selected and satisfies that $x_{id}(k+1) = x_{(i+1)d}(k)$, $i = 1, 2, \ldots, n-1$. The desired output trajectory $y_d(k)$ is bounded by a known smooth function over the compact set $\Omega$.

From Assumption 3 and system (1), we can obtain that $x_{(i+1)d}(k) = y_d(k+i)$, $i = 0, 1, \ldots, n-1$. Hence, the tracking error can be defined as

$$\begin{aligned} e_i(k) &= y_d(k+i) - y(k+i) \\ &= x_{(i+1)d}(k) - x_{i+1}(k) \end{aligned} \tag{3}$$

where $i = 0, 1, \ldots, n-1$.

### 2.2. A basic controller design approach

The purpose of this subsection is to develop a basic approach of the controller design for system (1). As mentioned before, due to the output of nonaffine nonlinear systems depending nonlinearly on the control input, feedback linearization methods cannot directly be used to design the controller for system (1). In order to handle this problem, we establish a novel control law for plant (1) based on Hovakimyan, Nardi, Calise, and Kim (2002) and Park, Huh, Kim, Seo, and Park (2005), which deals with CT adaptive control problems. From system (1), we have

$$\begin{aligned} y(k+n) &= h\big(x(k), \upsilon(k)\big) + d(k) \\ &= \alpha\upsilon(k) + f\big(x(k), \upsilon(k)\big) + d(k) \end{aligned} \tag{4}$$

where $f\big(x(k), \upsilon(k)\big) = h\big(x(k), \upsilon(k)\big) - \alpha\upsilon(k)$, and $\alpha > 0$ is a design constant.

Define the control input $\upsilon(k)$ as

$$\upsilon(k) = \frac{1}{\alpha}\big(\upsilon_s(k) - \upsilon_a(k)\big) \tag{5}$$

where $\upsilon_s(k)$ is a feedback controller designed to stabilize linearized error dynamics, $\upsilon_a(k)$ is an adaptive controller designed to approximate the unknown nonlinear term $f\big(x(k), \upsilon(k)\big)$ by using a single-hidden layer feedforward NN.

From (4) and (5), we obtain

$$y(k+n) = f\big(x(k), \upsilon(k)\big) - \upsilon_a(k) + \upsilon_s(k) + d(k). \tag{6}$$

In view of the objective of $\upsilon_s(k)$ and $\upsilon_a(k)$, we develop

$$\begin{cases} \upsilon_a(k) = \hat{f}\big(x(k), \upsilon(k)\big) \\ \upsilon_s(k) = y_d(k+n) + \lambda_1 e_{n-1}(k) + \cdots + \lambda_n e_0(k) \end{cases} \tag{7}$$

where $\hat{f}\big(x(k), \upsilon(k)\big)$ is an approximation of $f\big(x(k), \upsilon(k)\big)$, $e_{n-1}(k), \ldots, e_0(k)$ are the delayed values of $e_n(k)$, and $\lambda_1, \ldots, \lambda_n$ are constant matrices selected such that $\left|z^n + \lambda_1 z^{n-1} + \cdots + \lambda_n\right|$ is stable, that is, the solutions of $\left|z^n + \lambda_1 z^{n-1} + \cdots + \lambda_n\right| = 0$ are located inside the unit circle centered at the origin.

Let the approximation error of the unknown nonlinear function $f\big(x(k), \upsilon(k)\big)$ be

$$\tilde{f}\big(x(k), \upsilon(k)\big) = \hat{f}\big(x(k), \upsilon(k)\big) - f\big(x(k), \upsilon(k)\big). \tag{8}$$

Then, we can develop the following lemma.

**Lemma 1.** *Assume that the tracking error $e_i(k)$ is given by (3) and $\upsilon_s(k)$ is proposed as in (7). Then, the error dynamics can be derived as*

$$\boldsymbol{e}(k+1) = \tilde{A}\boldsymbol{e}(k) + \tilde{B}\Big[\tilde{f}\big(x(k), \upsilon(k)\big) - d(k)\Big] \tag{9}$$

*where*

$$\begin{aligned} &\boldsymbol{e}(k) = \left[e_0^{\mathrm{T}}(k), \ldots, e_{n-1}^{\mathrm{T}}(k)\right]^{\mathrm{T}} \\ &A = \begin{pmatrix} 0 & 1 & \cdots & 0 \\ \vdots & \vdots & & \vdots \\ -\lambda_n & -\lambda_{n-1} & \cdots & -\lambda_1 \end{pmatrix} \qquad B = \begin{pmatrix} 0 \\ \vdots \\ 1 \end{pmatrix} \\ &\tilde{A} = A \otimes I_m, \qquad \tilde{B} = B \otimes I_m. \end{aligned} \tag{10}$$

**Proof.** By using Assumption 3 and (3), we have that

$$\begin{aligned} e_i(k) &= y_d(k+i) - y(k+i) \\ &= x_{id}(k+1) - x_i(k+1) \\ &= e_{i-1}(k+1) \quad i = 1, 2, \ldots, n-1. \end{aligned} \tag{11}$$

Meanwhile, from (6) to (8), we can obtain

$$e_n(k) = -\boldsymbol{\lambda}^{\mathrm{T}}\boldsymbol{e}(k) + \tilde{f}\big(x(k), \upsilon(k)\big) - d(k)$$

where $\boldsymbol{\lambda} = \left[\lambda_n, \lambda_{n-1}, \ldots, \lambda_1\right]^{\mathrm{T}}$.

Noticing that $e_n(k) = e_{n-1}(k+1)$, we can derive

$$e_{n-1}(k+1) = -\boldsymbol{\lambda}^{\mathrm{T}}\boldsymbol{e}(\boldsymbol{k}) + \tilde{f}\big(x(k), \upsilon(k)\big) - d(k). \tag{12}$$

Accordingly, combining (11) and (12), we get

$$\begin{cases} e_0(k+1) = e_1(k) \\ \vdots \\ e_{n-1}(k+1) = -\boldsymbol{\lambda}^{\mathrm{T}}\boldsymbol{e}(\boldsymbol{k}) + \tilde{f}\big(x(k), \upsilon(k)\big) - d(k). \end{cases} \tag{13}$$

Rewriting (13) in the vector form, and observing the tracking error $\boldsymbol{e}(k) \in \mathbb{R}^{mn}$, we can derive (9) and (10). The proof is completed.

**Remark 1.** If there exists a control $\upsilon_a(k)$ successfully canceling the term $f\big(x(k), \upsilon(k)\big)$, i.e., $\tilde{f}\big(x(k), \upsilon(k)\big) = 0$, and ignores the disturbance term $d(k)$, i.e., $d(k) = 0$, then the closed-loop system becomes a linear system $\boldsymbol{e}(k+1) = \tilde{A}\boldsymbol{e}(k)$. Since $\lambda_1, \ldots, \lambda_n$ are constant matrices selected such that $\left|z^n + \lambda_1 z^{n-1} + \cdots + \lambda_n\right|$ is stable, it is obvious that $\tilde{A}$ can keep the linear system $\boldsymbol{e}(k+1) = \tilde{A}\boldsymbol{e}(k)$ stable (for short, $\tilde{A}$ is a stable matrix). Therefore, letting $\upsilon_a(k) = f\big(x(k), \upsilon(k)\big)$ and $d(k) = 0$, $\upsilon_s(k)$ can make the tracking error $\boldsymbol{e}(k)$ exponentially converge to zero as time increases. This shows that the definition of $\upsilon_s(k)$ in (7) makes sense.

Before continuing our discussion, we provide the Implicit Function Theorem for vector-valued functions, which plays a significant role in the subsequent proof.

**Lemma 2** (*Implicit Function Theorem (Apostol, 1974)*). *Let $\boldsymbol{f} = (f_1, \ldots, f_n)$ be a vector-valued function defined on an open set $S$ in $\mathbb{R}^{m+n}$ with values in $\mathbb{R}^n$. Suppose $\boldsymbol{f} \in C^1$ on $S$. Let $(\boldsymbol{x}_0, \boldsymbol{y}_0)$ be a point in $S$ for which $\boldsymbol{f}(\boldsymbol{x}_0, \boldsymbol{y}_0) = 0$, and for which the $n \times n$ determinant $\det\big[\partial \boldsymbol{f}(\boldsymbol{x}_0, \boldsymbol{y}_0)/\partial \boldsymbol{y}_0\big] \neq 0$. Then there exists a $n$-dimensional open set $T_0$ containing $\boldsymbol{y}_0$ and one, and only one, vector-valued function $\boldsymbol{g}$, defined on $T_0$ and having values in $\mathbb{R}^n$, such that* (i) $\boldsymbol{y}_0 = \boldsymbol{g}(\boldsymbol{x}_0)$; (ii) *for* $\forall\, (\boldsymbol{x}_0, \boldsymbol{y}_0) \in T_0, \boldsymbol{f}\big(\boldsymbol{x}_0, \boldsymbol{g}(\boldsymbol{x}_0)\big) = 0$.

By utilizing (5), we can define

$$\begin{aligned} F\big(x(k), \upsilon_a(k), \upsilon_s(k)\big) &= f\big(x(k), \upsilon(k)\big) - \upsilon_a(k) \\ &= f\left(x(k), \frac{\upsilon_s(k) - \upsilon_a(k)}{\alpha}\right) - \upsilon_a(k). \end{aligned}$$

Let $F\big(x(k), \upsilon_a(k), \upsilon_s(k)\big) = 0$. Then, we have

$$\begin{aligned} F\big(x(k), \upsilon_a(k), \upsilon_s(k)\big) &= f\left(x(k), \frac{\upsilon_s(k) - \upsilon_a(k)}{\alpha}\right) - \upsilon_a(k) \\ &= 0. \end{aligned} \tag{14}$$

From Remark 1, we know that, if ignoring the disturbance term $d(k)$, then the design of $\upsilon_s(k)$ is reasonable when there exists $\upsilon_a(k) = f\big(x(k), \upsilon(k)\big)$. However, one may doubt whether such a $\upsilon_a(k)$ exists or not. In other words, one may doubt whether there exists $\upsilon_a(k)$ guaranteeing the validity of (14). In order to deal with the problem, we develop the following theorem to show that the controller $\upsilon_a(k)$ does exist.

**Theorem 1.** *Assume that the following matrix inequality holds:*

$$\alpha\theta_1 I_m \leq \frac{\partial h\big(x(k), \upsilon(k)\big)}{\partial \upsilon(k)} \leq \alpha\theta_2 I_m \tag{15}$$

*where $0 < \theta_1 < \theta_2 \leq 2$. Then there exists a unique $\upsilon_a(k)$ satisfying (14) on a compact set $\Omega' \subseteq \Omega$.*

**Proof.** In order to utilize Lemma 2, the proof is divided into two parts. First, we show that there exists a solution of (14) (the solution is written as $\upsilon_a^*(k)$). Then, we show that $\det\big[\partial F\big(x(k), \upsilon_s(k), \upsilon_a^*(k)\big)/\partial \upsilon_a^*(k)\big] \neq 0$.

(i) The proof for showing the existence of $\upsilon_a^*(k)$.

In light of the expression of (14), if the conclusion is true, we have

$$\upsilon_a^*(k) = f\left(x(k), \frac{\upsilon_s(k) - \upsilon_a^*(k)}{\alpha}\right). \tag{16}$$

That is, $\upsilon_a^*(k)$ is the fixed point of (16). Accordingly, we just need to prove that $f\big(x(k), \cdot\big)$ is the contracting operator with respect to $\upsilon_a(k)$ on a compact set $U \subset \mathbb{R}^m$. Since $x(k)$ is defined on the compact $\Omega$ and $\upsilon(k)$ is a continuous function with respect to $x(k)$, by the knowledge of Functional Analysis (Rudin, 1991), we derive that $\upsilon(\Omega)$ is a compact set on $\mathbb{R}^m$. Hence, we can select $U = \upsilon(\Omega)$.

Notice that

$$\begin{aligned} \left\| \frac{\partial f\big(x(k), \upsilon(k)\big)}{\partial \upsilon_a(k)} \right\| &= \left\| \frac{\partial f\big(x(k), \upsilon(k)\big)}{\partial \upsilon(k)} \frac{\partial \upsilon(k)}{\partial \upsilon_a(k)} \right\| \\ &= \left\| \left( \frac{\partial h\big(x(k), \upsilon(k)\big)}{\partial \upsilon(k)} - \alpha I_m \right) \left( -\frac{I_m}{\alpha} \right) \right\| \\ &= \left\| I_m - \frac{\partial h\big(x(k), \upsilon(k)\big)}{\alpha \partial \upsilon(k)} \right\|. \end{aligned} \tag{17}$$

By using the matrix inequality (15), we have

$$\begin{cases} I_m - \dfrac{\partial h\big(x(k), \upsilon(k)\big)}{\alpha \partial \upsilon(k)} \geq (1 - \theta_2) I_m \\ I_m - \dfrac{\partial h\big(x(k), \upsilon(k)\big)}{\alpha \partial \upsilon(k)} \leq (1 - \theta_1) I_m. \end{cases} \tag{18}$$

Therefore, from (17) and (18), we get

$$\left\| \frac{\partial f\big(x(k), \upsilon(k)\big)}{\partial \upsilon_a(k)} \right\| \leq 1. \tag{19}$$

Noticing that $f\big(x(k), \upsilon(k)\big)$ is a continuous function and $\big(x(k), \upsilon(k)\big)$ is defined on the compact set $\Omega \times U$, we can conclude that $f\big(\Omega \times U\big)$ is a compact set on $\mathbb{R}^{nm} \times \mathbb{R}^m$. Consequently, we can obtain that $f\big(x(k), \cdot\big)$ is a completely continuous operator (Zeidler, 1985). By using Schauder's Fixed-Point Theorem (Zeidler, 1985) and from (19), we derive that there exists at least a fixed point for the operator $f\big(x(k), \cdot\big)$ on the compact set $U$. That is, there exists $\upsilon_a^*(k) \in U$ satisfying (14). Observing the definition of $U$, therefore, there exists $\upsilon_a^*(k)$ defined on $\Omega$ satisfying (14).

(ii) The proof for $\det\big[\partial F\big(x(k), \upsilon_s(k), \upsilon_a^*(k)\big)/\partial \upsilon_a^*(k)\big] \neq 0$.

Note that

$$\begin{aligned} &\frac{\partial F\big(x(k), \upsilon_s(k), \upsilon_a(k)\big)}{\partial \upsilon_a(k)}\bigg|_{\upsilon_a(k) = \upsilon_a^*(k)} \\ &= \frac{\partial \Big(h\big(x(k), \upsilon(k)\big) - \alpha \upsilon(k)\Big)}{\partial \upsilon(k)} \frac{\partial \upsilon(k)}{\partial \upsilon_a(k)}\bigg|_{\upsilon_a(k) = \upsilon_a^*(k)} - I_m \\ &= -\frac{\partial h\big(x(k), \upsilon(k)\big)}{\partial \upsilon(k)}\bigg|_{\upsilon_a(k) = \upsilon_a^*(k)}. \end{aligned}$$

From (2), we obtain $\det\big[\partial F\big(x(k), \upsilon_s(k), \upsilon_a^*(k)\big)/\partial \upsilon_a^*(k)\big] \neq 0$. Therefore, combining (i) and (ii), and using Lemma 2, we can obtain that there exists a unique $\upsilon_a(k)$ satisfying (14) on a compact set $\Omega' \subseteq \Omega$. The proof is completed.

**Remark 2.** From Assumption 2, one shall notice that the matrix inequality (15) is actually one of the properties of the positive definite matrix $\partial h\big(x(k), \upsilon(k)\big)/\partial \upsilon(k)$. This technique was utilized in both Lewis, Jagannathan, and Yesildirek (1999) and Lewis et al. (1996). Hence, the assumption about (15) makes sense.

**Remark 3.** Though, by utilizing Schauder's Fixed-Point Theorem, we derive that there exists at least one solution $\upsilon_a^*(k)$ satisfying (14) on the whole compact set $\Omega$ in part (i), it does not impair the validity of the conclusion, for we just want to show the existence of $\upsilon_a^*(k)$. In other words, $\upsilon_a^*(k)$ might not be unique on $\Omega$. The uniqueness of $\upsilon_a^*(k)$ is guaranteed by Implicit Function Theorem (Lemma 2). Therefore, one shall find the solution of (14) to be unique for a given local domain $\Omega'$. This feature satisfies the nonlinearity of the function $F\big(x(k), \upsilon_a(k), \upsilon_s(k)\big)$. Moreover, from the above analysis, we do not need $f\big(x(k), \cdot\big)$ to be the strictly contracting operator, which is a more relaxed condition than Hovakimyan et al. (2002) and Park et al. (2005).

**Remark 4.** Since there exists the controller $\upsilon_a(k)$ satisfying (16), we can conclude that $\upsilon_a(k)$ is actually a function with respect to $x(k)$ and $\upsilon_s(k)$. Therefore, we can obtain that $f\big(x(k), \upsilon(k)\big)$ is the function with respect to $x(k)$ and $\upsilon_s(k)$. Moreover, from the definition of $e_i(k)$ in (3) and the expression of $\upsilon_s(k)$ in (7), $f\big(x(k), \upsilon(k)\big)$ can be represented by a function with respect to $x(k)$ and $x_d(k)$. Accordingly, in the remainder of this paper, we denote

$$f\big(x(k), x_d(k)\big) = f\big(x(k), \upsilon(k)\big)$$

and

$$\tilde{f}\big(x(k), x_d(k)\big) = \tilde{f}\big(x(k), \upsilon(k)\big).$$

## 3. Online learning control based on RL

The purpose of this section is to develop an online learning control by using RL methods. Two subsections are included in this section. The design of the critic NN is first introduced. Then, the design of the action NN is presented.

### 3.1. Critic NN and weight update law

In this subsection, a critic NN is used to approximate the cost function $J(k)$ defined as in (21). The utility function (Si & Wang, 2001) depending on the tracking error vector $\boldsymbol{e}(k)$ is described by

$$r(k) = \big[r_1(k), \ldots, r_m(k)\big] \in \mathbb{R}^m \tag{20}$$

with

$$r_i(k) = \begin{cases} 0, & \text{if } \left\|\tilde{e}_i(k)\right\| \le \epsilon \\ 1, & \text{if } \left\|\tilde{e}_i(k)\right\| > \epsilon \end{cases} \quad i = 1, \ldots, m,$$

where $\tilde{e}(k) = \boldsymbol{\lambda}^{\mathrm{T}}\boldsymbol{e}(k) \in \mathbb{R}^m$, $\tilde{e}_i(k)$ is the $i$th element of the vector $\tilde{e}(k)$, and $\epsilon > 0$ is a prescribed threshold. The utility function $r(k)$ is considered as the performance index: $r_i(k) = 0$ and $r_i(k) = 1$ stand for the good and poor tracking performances, respectively. The cost function $J(k) \in \mathbb{R}^m$ (He & Jagannathan, 2005) is given by

$$J(k) = \tau^N r(k+1) + \tau^{N-1} r(k+2) + \cdots + \tau^{k+1} r(N) \tag{21}$$

with $0 < \tau \le 1$ a design parameter, and $N$ the final instant of time. From (21), we can derive

$$J(k) = \tau J(k-1) - \tau^{N+1} r(k)$$

which is the Bellman equation. Hence, the prediction error for the critic NN can be described by

$$e_c(k) = \hat{J}(k) - \tau\hat{J}(k-1) + \tau^{N+1} r(k) \tag{22}$$

where $\hat{J}(k)$ is the output of the critic NN, and it is also an approximation of $J(k)$.

The critic NN is implemented by a single-hidden layer feedforward NN. The critic NN output is given by

$$\hat{J}(k) = \hat{w}_c^{\mathrm{T}}(k)\sigma\big(\vartheta_c^{\mathrm{T}} x(k)\big) = \hat{w}_c^{\mathrm{T}}(k)\sigma_c\big(x(k)\big) \tag{23}$$

where $\vartheta_c \in \mathbb{R}^{nm \times s_1}$ is the weight vector for the input layer to the hidden layer of the critic NN, $\hat{w}_c(k) \in \mathbb{R}^{s_1 \times m}$ is the estimated weight vector for the hidden layer to the output layer of the critic NN, $s_1$ is the number of nodes in the hidden layer. Since the hidden layer weights are initialized randomly and kept constant, the activation function $\sigma(\vartheta_c^{\mathrm{T}} x(k))$ is written as $\sigma_c(x(k))$ for short.

The objective function to be minimized by the critic NN is defined as

$$E_c(k) = \frac{1}{2} e_c^{\mathrm{T}}(k) e_c(k). \tag{24}$$

The weight update law for the critic NN is a gradient-based adaptation, which is given by

$$\hat{w}_c(k+1) = \hat{w}_c(k) + \Delta\hat{w}_c(k) \tag{25}$$

where

$$\begin{aligned} \Delta\hat{w}_c(k) &= l_c\left[-\frac{\partial E_c(k)}{\partial \hat{w}_c(k)}\right] \\ &= l_c\left[-\frac{\partial E_c(k)}{\partial e_c(k)}\frac{\partial e_c(k)}{\partial \hat{J}(k)}\frac{\partial \hat{J}(k)}{\partial \hat{w}_c(k)}\right] \end{aligned}$$

and $0 < l_c < 1$ is the learning rate of the critic NN.

From (22) to (25), we can derive the weight update law for the critic NN as

$$\begin{aligned} \hat{w}_c(k+1) &= \hat{w}_c(k) - l_c\sigma_c\big(x(k)\big)e_c^{\mathrm{T}}(k) \\ &= \hat{w}_c(k) - l_c\sigma_c(x(k))\Big[\hat{w}_c^{\mathrm{T}}(k)\sigma_c\big(x(k)\big) \\ &\quad + \tau^{N+1} r(k) - \tau\,\hat{w}_c^{\mathrm{T}}(k-1)\sigma_c\big(x(k-1)\big)\Big]^{\mathrm{T}}. \end{aligned} \tag{26}$$

### 3.2. Action NN and weight update law

In this subsection, an action NN is employed to generate the input signal and approximate the unknown nonlinear function $f\big(x(k), \upsilon(k)\big)$. Due to the controller design described by (7), the error for the action NN should consist of the functional approximation error $\tilde{f}\big(x(k), \upsilon(k)\big)$ and the error between the nominal prescribed cost function $J_d(k) \in \mathbb{R}^m$ and the critic NN output $\hat{J}(k) \in \mathbb{R}^m$. Noting that $\tilde{f}\big(x(k), x_d(k)\big) = \tilde{f}\big(x(k), \upsilon(k)\big)$, the prediction error for the action NN is proposed by

$$e_a(k) = \hat{J}(k) - J_d(k) + \tilde{f}\big(x(k), x_d(k)\big). \tag{27}$$

The prescribed cost function $J_d(k)$ is generally considered to be zero, i.e., $J_d(k) = 0$, which represents that the system state can track the reference signal well (Si & Wang, 2001). Therefore, the prediction error given by (27) becomes

$$e_a(k) = \hat{J}(k) + \tilde{f}\big(x(k), x_d(k)\big). \tag{28}$$

The action NN is also implemented by a single-hidden layer feedforward NN. The action NN output is given by

$$\hat{f}(k) = \hat{w}_a^{\mathrm{T}}(k)\sigma\big(\vartheta_a^{\mathrm{T}} z(k)\big) = \hat{w}_a^{\mathrm{T}}(k)\sigma_a\big(z(k)\big) \tag{29}$$

where $\hat{f}(k)$ stands for $\hat{f}\big(x(k), x_d(k)\big)$, $\vartheta_a \in \mathbb{R}^{(n+1)m \times s_2}$ is the weight vector for the input layer to the hidden layer of the action NN, $\hat{w}_a(k) \in \mathbb{R}^{s_2 \times m}$ is the estimated weight vector for the hidden layer to the output layer of the action NN, $s_2$ is the number of nodes in the hidden layer, and $z(k) = \big[x^{\mathrm{T}}(k)\ x_d^{\mathrm{T}}(k)\big]^{\mathrm{T}} \in \mathbb{R}^{(n+1)m}$. Since the hidden layer weights are initialized randomly and kept constant, for briefly, the activation function $\sigma\big(\vartheta_a^{\mathrm{T}} z(k)\big)$ is written as $\sigma_a(z(k))$.

**Remark 5.** From (7), we shall find that the output of the action NN is actually the controller $\upsilon_a(k)$. Meanwhile, we have that $z(k) = \big[x^{\mathrm{T}}(k)\ x_{nd}^{\mathrm{T}}(k) + \boldsymbol{e}^{\mathrm{T}}(k)\boldsymbol{\lambda}\big]^{\mathrm{T}} \in \mathbb{R}^{(n+1)m}$.

According to the universal approximation property of NNs (Igelnik & Pao, 1995), $f(k)$ can accurately be represented as

$$f(k) = w_a^{\mathrm{T}}\sigma_a\big(z(k)\big) + \varepsilon(k) \tag{30}$$

where $f(k)$ denotes $f\big(x(k), x_d(k)\big)$, $w_a \in \mathbb{R}^{s_2 \times m}$ is the ideal weight vector for the hidden layer to the output layer of the action NN, and $\varepsilon(k)$ is the action NN approximation error.

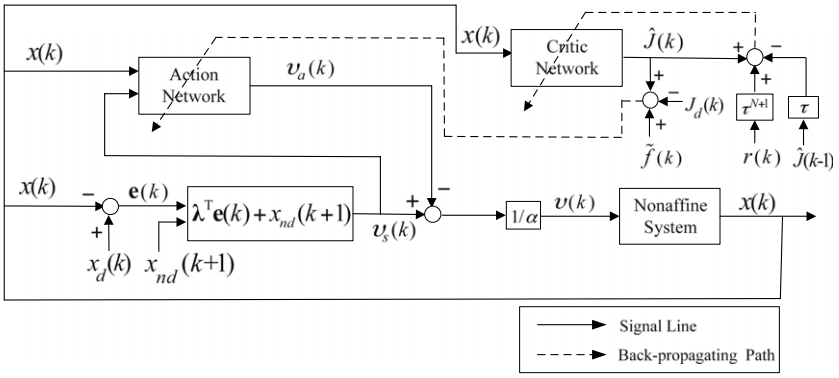


**Fig. 1.** The developed control algorithm for nonaffine nonlinear DT systems.

Define the weight approximation error of the action NN as

$$\tilde{w}_a(k) = \hat{w}_a(k) - w_a. \tag{31}$$

It is significant to note that, in this paper, the relationship between $\tilde{w}_c(k)$ and $\hat{w}_c(k)$ is defined as the same as in (31). From (8) and (29)–(31), the function approximation error $\tilde{f}(k)$ can be represented as

$$\tilde{f}(k) = \tilde{w}_a^{\mathrm{T}}(k)\sigma_a\big(z(k)\big) - \varepsilon(k). \tag{32}$$

The objective function to be minimized by the action NN is defined as a quadratic function of error as

$$E_a(k) = \frac{1}{2}e_a^{\mathrm{T}}(k)e_a(k). \tag{33}$$

The weight update law for the action NN is given by

$$\hat{w}_a(k+1) = \hat{w}_a(k) + \Delta\hat{w}_a(k) \tag{34}$$

where

$$\begin{aligned}\Delta\hat{w}_a(k) &= l_a\left[-\frac{\partial E_a(k)}{\partial \hat{w}_a(k)}\right] \\ &= l_a\left[-\frac{\partial E_a(k)}{\partial e_a(k)}\frac{\partial e_a(k)}{\partial \tilde{f}(k)}\frac{\partial \tilde{f}(k)}{\partial \hat{w}_a(k)}\right]\end{aligned}$$

and $0 < l_a < 1$ is the learning rate for the action NN.

From (28)–(34), we get the weight update law for the action NN as

$$\begin{aligned}\hat{w}_a(k+1) &= \hat{w}_a(k) - l_a\sigma_a\big(z(k)\big)e_a^{\mathrm{T}}(k) \\ &= \hat{w}_a(k) - l_a\sigma_a\big(z(k)\big)\Big(\hat{w}_c^{\mathrm{T}}(k)\sigma_c\big(x(k)\big) \\ &\quad + \tilde{w}_a^{\mathrm{T}}(k)\sigma_a\big(z(k)\big) - \varepsilon(k)\Big)^{\mathrm{T}}.\end{aligned} \tag{35}$$

By using (9) and Remark 4, we have

$$\tilde{f}(k) = \tilde{B}^{\mathrm{T}}\big(\boldsymbol{e}(k+1) - \tilde{A}\boldsymbol{e}(k)\big) + d(k). \tag{36}$$

By utilizing (32) and (36), (35) can be written as

$$\begin{aligned}\hat{w}_a(k+1) &= \hat{w}_a(k) - l_a\sigma_a\big(z(k)\big)\Big(\hat{w}_c^{\mathrm{T}}(k)\sigma_c\big(x(k)\big) \\ &\quad + \tilde{B}^{\mathrm{T}}\big(\boldsymbol{e}(k+1) - \tilde{A}\boldsymbol{e}(k)\big) + d(k)\Big)^{\mathrm{T}}.\end{aligned} \tag{37}$$

However, $d(k)$ is typically unavailable. Consequently, the ideal case of $d(k)$ is generally considered, i.e., $d(k) = 0$ (He & Jagannathan, 2005; Yang & Jagannathan, 2012; Yang et al., 2009). Then, the weight update rule (37) is represented as

$$\begin{aligned}\hat{w}_a(k+1) &= \hat{w}_a(k) - l_a\sigma_a(z(k))\Big(\hat{w}_c^{\mathrm{T}}(k)\sigma_c(x(k)) \\ &\quad + \tilde{B}^{\mathrm{T}}\big(\boldsymbol{e}(k+1) - \tilde{A}\boldsymbol{e}(k)\big)\Big)^{\mathrm{T}}.\end{aligned} \tag{38}$$

A general schematic programming of the proposed control algorithm is presented in Fig. 1.

## 4. Stability analysis and performance of the closed-loop system

The purpose of this section is to establish our main results by using Lyapunov's direct method. Before engaging in demonstrating the main theorem, we need to develop some mild assumptions and facts as follows:

**Assumption 4.** Let ideal output layer weights $w_a$ and $w_c$ be bounded over the compact set $\Omega$ by known positive constants $w_{aM}$ and $w_{cM}$, respectively. That is,

$$\|w_a\| \le w_{aM}, \qquad \|w_c\| \le w_{cM}.$$

**Assumption 5.** Let the NN approximation error $\varepsilon(k)$ be bounded by a known positive constant $\varepsilon_M$ over the compact set $\Omega$. That is, $\|\varepsilon(k)\| \le \varepsilon_M$.

**Fact 1.** The activation functions for the action NN and the critic NN are bounded by known positive constants over the compact set $\Omega$. That is, there exist $\sigma_{aM} > 0$ and $\sigma_{cM} > 0$ such that

$$\|\sigma_a\big(z(k)\big)\| \le \sigma_{aM}, \qquad \|\sigma_c\big(x(k)\big)\| \le \sigma_{cM}. \tag{39}$$

**Fact 2.** From Remark 1, we know that $\tilde{A}$ is a stable matrix. Hence, there exists a positive definite matrix $P \in \mathbb{R}^{mn\times mn}$ satisfying the Lyapunov equation

$$\tilde{A}^{\mathrm{T}}P\tilde{A} - P = -\beta I_{mn} \tag{40}$$

where $\beta$ is a positive constant.

**Fact 3.** By the same technique used in Lewis et al. (1999), there exist two known positive constants $\varrho$ and $\rho$ ($\varrho < \rho$) such that

$$\varrho I_m \le \tilde{B}^{\mathrm{T}}P\tilde{B} \le \rho I_m. \tag{41}$$

**Remark 6.** From the definition of $\tilde{B}$ in (10), we derive that $\tilde{B} \in \mathbb{R}^{mn\times m}$ and $\mathrm{rank}(\tilde{B}) = m$. Noting that $P \in \mathbb{R}^{mn\times mn}$ is positive definite, by using the corollary developed in Matrix Analysis (Horn & Johnson, 1985, p. 399), we obtain that $\tilde{B}^{\mathrm{T}}P\tilde{B} \in \mathbb{R}^{m\times m}$ is positive definite. Then, by utilizing the property of the positive definite matrix which has been used in Lewis et al. (1999), we can get Fact 3.

Next, several lemmas are developed to lay bases for deriving our main theorem.

**Lemma 3.** *Let Assumptions 1–5 hold. Taking the control input as (5), and combining (9), (32), (40), (41), we can derive that the first difference of*

$$L_1(k) = \gamma_1 \boldsymbol{e}^{\mathrm{T}}(k) P \boldsymbol{e}(k)$$

*as*

$$\begin{aligned}\Delta L_1(k) \leq\ & 2\gamma_1(\rho+\eta)\|\varepsilon(k)+d(k)\|^2 \\ & + 2\gamma_1(\rho+\eta)\|\xi_a(k)\|^2 - \gamma_1(\beta-1)\|\boldsymbol{e}(k)\|^2 \end{aligned} \tag{42}$$

*where* $\eta = \|\tilde{A}^{\mathrm{T}} P \tilde{B}\|^2$, $\xi_a(k) = \tilde{w}_a^{\mathrm{T}}(k)\sigma_a\big(z(k)\big)$, *and* $\gamma_1 > 0$ *is a design parameter.*

**Proof.** The first difference of $L_1(k)$ is

$$\Delta L_1(k) = \gamma_1\big[\boldsymbol{e}^{\mathrm{T}}(k+1)P\boldsymbol{e}(k+1) - \boldsymbol{e}^{\mathrm{T}}(k)P\boldsymbol{e}(k)\big].$$

Define $Q(k) = \xi_a(k) - \big(\varepsilon(k)+d(k)\big)$. Combining (9), (32), (40) and (41), we obtain

$$\begin{aligned}\Delta L_1(k) &= \gamma_1\big[\tilde{A}\boldsymbol{e}(k) + \tilde{B}Q(k)\big]^{\mathrm{T}} P\big[\tilde{A}\boldsymbol{e}(k)+\tilde{B}Q(k)\big] - \gamma_1\boldsymbol{e}^{\mathrm{T}}(k)P\boldsymbol{e}(k) \\ &= \gamma_1\boldsymbol{e}^{\mathrm{T}}(k)\big(\tilde{A}^{\mathrm{T}}P\tilde{A} - P\big)\boldsymbol{e}(k) + 2\gamma_1\boldsymbol{e}^{\mathrm{T}}(k) \\ &\quad \times \tilde{A}^{\mathrm{T}}P\tilde{B}Q(k) + \gamma_1 Q^{\mathrm{T}}(k)\big(\tilde{B}^{\mathrm{T}}P\tilde{B}\big)Q(k) \\ &\leq -\gamma_1\beta\|\boldsymbol{e}(k)\|^2 + 2\gamma_1\boldsymbol{e}^{\mathrm{T}}(k)\tilde{A}^{\mathrm{T}}P\tilde{B}Q(k) + \gamma_1\rho\|Q(k)\|^2 \\ &\leq \gamma_1\big(\rho + \|\tilde{A}^{\mathrm{T}}P\tilde{B}\|^2\big)\|Q(k)\|^2 - \gamma_1(\beta-1)\|\boldsymbol{e}(k)\|^2. \end{aligned} \tag{43}$$

Applying the Cauchy–Schwarz inequality $\|x+y\|^2 \leq 2\|x\|^2 + 2\|y\|^2$ to $\|Q(k)\|^2$ and using (43), we can obtain (42). The proof is completed.

**Lemma 4.** *Given that Assumptions 1–5 hold. Take the control input as (5), and the utility function as defined in (20). Combining (23) and (26), we can obtain the first difference of*

$$L_2(k) = \frac{\gamma_2}{l_c}\mathrm{tr}\big(\tilde{w}_c^{\mathrm{T}}(k)\tilde{w}_c(k)\big)$$

*as*

$$\begin{aligned}\Delta L_2(k) \leq\ & -\gamma_2\|\xi_c(k)\|^2 - \gamma_2\Big(1 - l_c\big\|\sigma_c\big(x(k)\big)\big\|^2\Big) \\ & \times \big\|\xi_c(k) + w_c^{\mathrm{T}}\sigma_c\big(x(k)\big) + \tau^{N+1}r(k) \\ & - \tau\hat{w}_c^{\mathrm{T}}(k-1)\sigma_c\big(x(k-1)\big)\big\|^2 \\ & + 2\tau^2\gamma_2\|\xi_c(k-1)\|^2 + 2\gamma_2\big\|w_c^{\mathrm{T}}\sigma_c\big(x(k)\big) \\ & + \tau^{N+1}r(k) - \tau w_c^{\mathrm{T}}\sigma_c\big(x(k-1)\big)\big\|^2 \end{aligned} \tag{44}$$

*where* $\xi_c(k) = \tilde{w}_c^{\mathrm{T}}(k)\sigma_c\big(x(k)\big)$ *and* $\gamma_2 > 0$ *is a design parameter.*

**Proof.** The first difference of $L_2(k)$ is

$$\Delta L_2(k) = \frac{\gamma_2}{l_c}\mathrm{tr}\big[\tilde{w}_c^{\mathrm{T}}(k+1)\tilde{w}_c(k+1) - \tilde{w}_c^{\mathrm{T}}(k)\tilde{w}_c(k)\big]. \tag{45}$$

Observing that $\tilde{w}_c(k) = \hat{w}_c(k) - w_c$ and using (26), we have

$$\begin{aligned}\tilde{w}_c(k+1) =\ & \Big(I_{s_1} - l_c\sigma_c\big(x(k)\big)\sigma_c^{\mathrm{T}}\big(x(k)\big)\Big)\tilde{w}_c(k) \\ & - l_c\sigma_c\big(x(k)\big)\Big(w_c^{\mathrm{T}}\sigma_c\big(x(k)\big) + \tau^{N+1}r(k) \\ & - \tau\hat{w}_c^{\mathrm{T}}(k-1)\sigma_c\big(x(k-1)\big)\Big)^{\mathrm{T}}. \end{aligned} \tag{46}$$

Combining (45) and (46), we obtain

$$\Delta L_2(k) = \frac{\gamma_2}{l_c}\mathrm{tr}\big(\mathfrak{R}_1(k) + \mathfrak{R}_2(k) + \mathfrak{R}_3(k)\big) \tag{47}$$

where

$$\begin{aligned}\mathfrak{R}_1(k) &= \Big[\Big(I_{s_1} - l_c\sigma_c\big(x(k)\big)\sigma_c^{\mathrm{T}}\big(x(k)\big)\Big)\tilde{w}_c(k)\Big]^{\mathrm{T}} \\ &\quad \times \Big[\Big(I_{s_1} - l_c\sigma_c\big(x(k)\big)\sigma_c^{\mathrm{T}}\big(x(k)\big)\Big)\tilde{w}_c(k)\Big] - \tilde{w}_c^{\mathrm{T}}(k)\tilde{w}_c(k) \\ &= -l_c\Big(1 - l_c\big\|\sigma_c\big(x(k)\big)\big\|^2\Big)\xi_c(k)\xi_c^{\mathrm{T}}(k) - l_c\xi_c(k)\xi_c^{\mathrm{T}}(k) \\ \mathfrak{R}_2(k) &= -2\tilde{w}_c^{\mathrm{T}}(k)\Big(I_{s_1} - l_c\sigma_c\big(x(k)\big)\sigma_c^{\mathrm{T}}\big(x(k)\big)\Big) \\ &\quad \times l_c\sigma_c\big(x(k)\big)\Big(w_c^{\mathrm{T}}\sigma_c\big(x(k)\big) + \tau^{N+1}r(k) \\ &\quad - \tau\hat{w}_c^{\mathrm{T}}(k-1)\sigma_c\big(x(k-1)\big)\Big)^{\mathrm{T}} \\ &= -2l_c\xi_c(k)\Big(1 - l_c\big\|\sigma_c\big(x(k)\big)\big\|^2\Big) \\ &\quad \times \Big(w_c^{\mathrm{T}}\sigma_c\big(x(k)\big) + \tau^{N+1}r(k) \\ &\quad - \tau\hat{w}_c^{\mathrm{T}}(k-1)\sigma_c\big(x(k-1)\big)\Big)^{\mathrm{T}} \end{aligned}$$

and

$$\begin{aligned}\mathfrak{R}_3(k) =\ & l_c^2\Big(w_c^{\mathrm{T}}\sigma_c\big(x(k)\big) + \tau^{N+1}r(k) \\ & - \tau\hat{w}_c^{\mathrm{T}}(k-1)\sigma_c\big(x(k-1)\big)\Big)\sigma_c^{\mathrm{T}}\big(x(k)\big) \\ & \times \sigma_c\big(x(k)\big)\Big(w_c^{\mathrm{T}}\sigma_c\big(x(k)\big) + \tau^{N+1}r(k) \\ & - \tau\hat{w}_c^{\mathrm{T}}(k-1)\sigma_c\big(x(k-1)\big)\Big)^{\mathrm{T}} \\ =\ & l_c^2\big\|\sigma_c\big(x(k)\big)\big\|^2\Big(w_c^{\mathrm{T}}\sigma_c\big(x(k)\big) + \tau^{N+1}r(k) \\ & - \tau\hat{w}_c^{\mathrm{T}}(k-1)\sigma_c\big(x(k-1)\big)\Big)\Big(w_c^{\mathrm{T}}\sigma_c\big(x(k)\big) \\ & + \tau^{N+1}r(k) - \tau\hat{w}_c^{\mathrm{T}}(k-1)\sigma_c\big(x(k-1)\big)\Big)^{\mathrm{T}}. \end{aligned}$$

Define

$$M(k) = w_c^{\mathrm{T}}\sigma_c\big(x(k)\big) + \tau^{N+1}r(k) - \tau\hat{w}_c^{\mathrm{T}}(k-1)\sigma_c\big(x(k-1)\big). \tag{48}$$

From (47), we derive

$$\begin{aligned}\Delta L_2(k) =\ & \gamma_2\mathrm{tr}\Big[-\Big(1 - l_c\big\|\sigma_c\big(x(k)\big)\big\|^2\Big)\xi_c(k)\xi_c^{\mathrm{T}}(k) \\ & - 2\xi_c(k)\Big(1 - l_c\big\|\sigma_c\big(x(k)\big)\big\|^2\Big)M^{\mathrm{T}}(k) \\ & + l_c\big\|\sigma_c\big(x(k)\big)\big\|^2 M(k)M^{\mathrm{T}}(k) - \xi_c(k)\xi_c^{\mathrm{T}}(k)\Big] \\ =\ & -\gamma_2\xi_c^{\mathrm{T}}(k)\xi_c(k) - \Big(1 - l_c\big\|\sigma_c\big(x(k)\big)\big\|^2\Big) \\ & \times \gamma_2\Big(\xi_c^{\mathrm{T}}(k)\xi_c(k) + 2\xi_c^{\mathrm{T}}(k)M(k)\Big) \\ & + \gamma_2 l_c\big\|\sigma_c\big(x(k)\big)\big\|^2 M^{\mathrm{T}}(k)M(k) \\ =\ & -\gamma_2\|\xi_c(k)\|^2 - \Big(1 - l_c\big\|\sigma_c\big(x(k)\big)\big\|^2\Big) \\ & \times \gamma_2\|\xi_c(k) + M(k)\|^2 + \gamma_2\|M(k)\|^2. \end{aligned} \tag{49}$$

Observe that (48) can be rewritten as

$$\begin{aligned}M(k) =\ & w_c^{\mathrm{T}}\sigma_c\big(x(k)\big) + \tau^{N+1}r(k) \\ & - \tau w_c^{\mathrm{T}}\sigma_c\big(x(k-1)\big) - \tau\xi_c\big(x(k-1)\big). \end{aligned} \tag{50}$$

Applying the Cauchy–Schwarz inequality $\|x+y\|^2 \leq 2\|x\|^2 + 2\|y\|^2$ to $\|M(k)\|^2$, and combining (49) and (50), we can derive (44). The proof is completed.

**Lemma 5.** *Suppose that Assumptions 1–5 hold. Take the control input as (5), and the utility function as defined in (20). Combining (23) and (38), we can derive the first difference of*

$$L_3(k) = \frac{\gamma_3}{l_a}\mathrm{tr}\big(\tilde{w}_a^{\mathsf{T}}(k)\tilde{w}_a(k)\big)$$

*as*

$$\begin{aligned}\Delta L_3(k) \le\ & 2\gamma_3\big\|w_c^{\mathsf{T}}\sigma_c\big(x(k)\big) - \varepsilon(k) - d(k)\big\|^2 \\ & + 2\gamma_3\|\xi_c(k)\|^2 - \gamma_3\|\xi_a(k)\|^2 \\ & - \gamma_3\Big(1 - l_a\big\|\sigma_a(z(k))\big\|^2\Big)\big\|\hat{w}_c^{\mathsf{T}}(k)\sigma_c\big(x(k)\big) \\ & + \xi_a(k) - \varepsilon(k) - d(k)\big\|^2 \end{aligned} \tag{51}$$

*where $\xi_a(k) = \tilde{w}_a^{\mathsf{T}}(k)\sigma_a\big(z(k)\big)$ and $\gamma_3 > 0$ is a design parameter.*

**Proof.** The first difference of $L_3(k)$ is

$$\Delta L_3(k) = \frac{\gamma_3}{l_a}\mathrm{tr}\big[\tilde{w}_a^{\mathsf{T}}(k+1)\tilde{w}_a(k+1) - \tilde{w}_a^{\mathsf{T}}(k)\tilde{w}_a(k)\big]. \tag{52}$$

Combining (9), (31) and (38), we obtain

$$\begin{aligned}\tilde{w}_a(k+1) =\ & \tilde{w}_a(k) - l_a\sigma_a\big(z(k)\big)\Big(\hat{w}_c^{\mathsf{T}}(k)\sigma_c\big(x(k)\big) \\ & + \xi_a(k) - \varepsilon(k) - d(k)\Big)^{\mathsf{T}}. \end{aligned} \tag{53}$$

Define

$$N(k) = \hat{w}_c^{\mathsf{T}}(k)\sigma_c\big(x(k)\big) + \xi_a(k) - \varepsilon(k) - d(k).$$

From (52) and (53), we have

$$\begin{aligned}\Delta L_3(k) =\ & \frac{\gamma_3}{l_a}\mathrm{tr}\Big[-l_a\tilde{w}_a^{\mathsf{T}}(k)\sigma_a\big(z(k)\big)N^{\mathsf{T}}(k) - l_aN(k) \\ & \times\sigma_a^{\mathsf{T}}\big(z(k)\big)\Big(\tilde{w}_a(k) - l_a\sigma_a\big(z(k)\big)N^{\mathsf{T}}(k)\Big)\Big] \\ =\ & -\gamma_3\mathrm{tr}\Big[\xi_a(k)N^{\mathsf{T}}(k) + N(k) \\ & \times\Big(\xi_a^{\mathsf{T}}(k) - l_a\big\|\sigma_a\big(z(k)\big)\big\|^2N^{\mathsf{T}}(k)\Big)\Big] \\ =\ & l_a\gamma_3\big\|\sigma_a\big(z(k)\big)\big\|^2N^{\mathsf{T}}(k)N(k) - 2\gamma_3\xi_a^{\mathsf{T}}(k)N(k) \\ =\ & l_a\gamma_3\big\|\sigma_a\big(z(k)\big)\big\|^2\|N(k)\|^2 \\ & + \gamma_3\|\xi_a(k) - N(k)\|^2 - \gamma_3\|N(k)\|^2 - \gamma_3\|\xi_a(k)\|^2. \end{aligned} \tag{54}$$

Notice that $N(k)$ can be rewritten as

$$N(k) = w_c^{\mathsf{T}}\sigma_c\big(x(k)\big) + \xi_a(k) - \varepsilon(k) - d(k) + \xi_c(k). \tag{55}$$

Applying the Cauchy–Schwarz inequality $\|x+y\|^2 \le 2\|x\|^2+2\|y\|^2$ to $\|\xi_a(k) - N(k)\|^2$ and using (54) and (55), we can get (51). The proof is completed.

With the aid of Assumptions 1–5 and Facts 1–3, our main theorem is established.

**Theorem 2.** *Consider the nonaffine nonlinear system described by (1). Let Assumptions 1–5 hold. Take the control input for system (1) as (5) with (7) and the critic NN (23), as well as the action NN (29). Moreover, let the weight update law for the critic NN and the action NN be (26) and (38), respectively. Then, the tracking error vector $\boldsymbol{e}(k)$, the weights of estimation error for the action NN $\tilde{w}_a(k)$, and the weights of estimation error for the critic NN $\tilde{w}_c(k)$ are UUB by positive constants $\mathfrak{D}_1$, $\mathfrak{D}_2$ and $\mathfrak{D}_3$, respectively, which are given by*

$$\mathfrak{D}_1 = \sqrt{\frac{\mathfrak{B}_M^2}{\gamma_1(\beta-1)}}$$

$$\mathfrak{D}_2 = \frac{1}{\sigma_{aM}}\sqrt{\frac{\mathfrak{B}_M^2}{\gamma_3 - 2\gamma_1(\rho+\eta)}}$$

$$\mathfrak{D}_3 = \frac{1}{\sigma_{cM}}\sqrt{\frac{\mathfrak{B}_M^2}{(1-2\tau^2)\gamma_2 - 2\gamma_3}}$$

*provided that the following conditions hold:*

$$\begin{aligned}&\text{(a) } \beta > 1 \\ &\text{(b) } 0 < l_c\big\|\sigma_c\big(x(k)\big)\big\|^2 < 1 \\ &\text{(c) } 0 < l_a\big\|\sigma_a\big(z(k)\big)\big\|^2 < 1 \\ &\text{(d) } 0 < \tau < \frac{\sqrt{2}}{2}. \end{aligned} \tag{56}$$

**Proof.** Consider the Lyapunov function candidate

$$L(k) = L_1(k) + L_2(k) + L_3(k) + L_4(k)$$

where

$$\begin{aligned}L_1(k) &= \gamma_1\boldsymbol{e}^{\mathsf{T}}(k)P\boldsymbol{e}(k) \\ L_2(k) &= \frac{\gamma_2}{l_c}\mathrm{tr}\big(\tilde{w}_c^{\mathsf{T}}(k)\tilde{w}_c(k)\big) \\ L_3(k) &= \frac{\gamma_3}{l_a}\mathrm{tr}\big(\tilde{w}_a^{\mathsf{T}}(k)\tilde{w}_a(k)\big) \\ L_4(k) &= \gamma_4\|\xi_c(k-1)\|^2. \end{aligned}$$

The first difference of the Lyapunov function candidate is

$$\Delta L(k) = \Delta L_1(k) + \Delta L_2(k) + \Delta L_3(k) + \Delta L_4(k). \tag{57}$$

For convenience, we denote

$$P(k) = w_c^{\mathsf{T}}\sigma_c\big(z(k)\big) + \tau^{N+1}r(k) - \tau w_c^{\mathsf{T}}\sigma_c\big(x(k-1)\big).$$

By employing Lemmas 3–5 and utilizing (57), we derive

$$\begin{aligned}\Delta L(k) \le\ & -\gamma_1(\beta-1)\|\boldsymbol{e}(k)\|^2 + \mathfrak{B}^2(k) \\ & - \big(\gamma_2 - 2\gamma_3 - \gamma_4\big)\|\xi_c(k)\|^2 \\ & - \big(\gamma_4 - 2\tau^2\gamma_2\big)\|\xi_c(k-1)\|^2 \\ & - \big(\gamma_3 - 2\gamma_1(\rho+\eta)\big)\|\xi_a(k)\|^2 \\ & - \gamma_2\Big(1 - l_c\big\|\sigma_c\big(x(k)\big)\big\|^2\Big)\|\xi_c(k) + M(k)\|^2 \\ & - \gamma_3\Big(1 - l_a\|\sigma_a\big(z(k)\big)\|^2\Big)\|N(k)\|^2 \end{aligned} \tag{58}$$

where

$$\begin{aligned}\mathfrak{B}^2(k) =\ & 2\gamma_2\|P(k)\|^2 + 2\gamma_3\big\|w_c^{\mathsf{T}}\sigma_c(k) - \varepsilon(k) - d(k)\big\|^2 \\ & + 2\gamma_1(\rho+\eta)\|\varepsilon(k) + d(k)\|^2. \end{aligned}$$

By using Assumptions 4–5 and the Cauchy–Schwarz inequality, we obtain

$$\begin{aligned}\mathfrak{B}^2(k) \le\ & (12\gamma_2 + 6\gamma_3)w_{cM}^2\sigma_{cM}^2 + 6\gamma_2r_M^2 \\ & + \big(6\gamma_3 + 4\gamma_1(\rho+\eta)\big)\big(\varepsilon_M^2 + d_M^2\big) \\ \triangleq\ & \mathfrak{B}_M^2 \end{aligned} \tag{59}$$

where $r_M$ is an upper bound of $\|r(k)\|$, i.e., $\|r(k)\| \le r_M$.

The parameters $r_i$ $(i = 1, 2, 3)$ are selected to satisfy that

$$\gamma_1 < \frac{\gamma_3}{2(\rho+\eta)}, \qquad \gamma_2 = \frac{\gamma_4}{2\tau^2}, \quad \text{and} \quad \gamma_2 > \frac{2\gamma_3}{1-2\tau^2}. \tag{60}$$

Therefore, by using (56) and (60), we can conclude that (58) and (59) yield $\Delta L(k) < 0$ as long as one of the following conditions holds:

$$\|\boldsymbol{e}(k)\| > \sqrt{\frac{\mathfrak{B}_M^2}{\gamma_1(\beta-1)}}$$

or

$$\|\xi_a(k)\| > \sqrt{\frac{\mathfrak{B}_M^2}{\gamma_3 - 2\gamma_1(\rho+\eta)}} \tag{61}$$

or

$$\|\xi_c(k)\| > \sqrt{\frac{\mathfrak{B}_M^2}{(1-2\tau^2)\gamma_2 - 2\gamma_3}} \tag{62}$$

where $r_i > 0$ $(i = 1, 2, 3)$ are design parameters, $\beta$, $\rho$, and $\eta$ are defined as in (40)–(42), respectively.

Note $\|\xi_a(k)\| \le \sigma_{aM}\|\tilde{w}_a(k)\|$, $\|\xi_c(k)\| \le \sigma_{cM}\|\tilde{w}_c(k)\|$. Then, by using (61) and (62), we can derive

$$\|\tilde{w}_a(k)\| > \frac{1}{\sigma_{aM}}\sqrt{\frac{\mathfrak{B}_M^2}{\gamma_3 - 2\gamma_1(\rho+\eta)}}$$

$$\|\tilde{w}_c(k)\| > \frac{1}{\sigma_{cM}}\sqrt{\frac{\mathfrak{B}_M^2}{(1-2\tau^2)\gamma_2 - 2\gamma_3}},$$

where $\sigma_{aM}$ and $\sigma_{cM}$ defined as in (39). By using the standard Lyapunov extension theorem (Lewis et al., 1999), we can draw the conclusion that the tracking error vector $\boldsymbol{e}(k)$, the weights of estimation error for the action NN $\tilde{w}_a(k)$, and the weights of estimation error for the critic NN $\tilde{w}_c(k)$ are all UUB. The proof is completed.

## 5. Numerical examples

In order to verify our theoretical results, two examples are provided for numerical experiments.

### 5.1. Example 1

Our first example is selected from Zhang, Ge, and Lee (2002). We consider nonaffine nonlinear DT systems described by

$$\begin{aligned} x_1(k+1) &= x_2(k) \\ x_2(k+1) &= \frac{x_1(k)x_2(k)\big(x_1(k)+2.5\big)}{1+x_1^2(k)+x_2^2(k)} + u(k) + 0.1u^3(k) + d(k) \\ y(k) &= x_1(k) \end{aligned} \tag{63}$$

where $d(k)$ is a bounded external disturbance, and has the form

$$d(k) = 0.1\cos(0.001k).$$

The control objective is to control the system output $y(k)$ to track the prescribed trajectory $y_d(k) = 0.6\sin(\pi k/265)$. From (63), we can obtain

$$\partial h\big(x(k), u(k)\big)/\partial u(k) = 1 + 0.3u^2(k).$$

Obviously, $\det\big[\partial h\big(x(k), u(k)\big)/\partial u(k)\big] \neq 0$. Select $\Omega = [-1, 1] \times [-1, 1]$. Since $u(x(k))$ is a continuous function with respect to $x(k)$, we can conclude that $u(x(k))$ is bounded on $\Omega$. Hence, $\partial h(x(k), u(k))/\partial u(k)$ is bounded on $\Omega$.

The initial state is chosen to be $x_0 = [0.5, -0.5]^{\mathrm{T}}$. Let $\lambda_1 = 1$, $\lambda_2 = 0.25$ (i.e., $z^2 + \lambda_1 z + \lambda_2$ is stable). Meanwhile, we select $\alpha = 2$, $\beta = 2$, $\tau = 0.7$, and the prescribed threshold $\epsilon = 8\times10^{-3}$. The learning rates of the action NN and the critic NN are selected as $l_a = 0.01$ and $l_c = 0.001$, respectively. Define

$$\Delta(k) = \upsilon_a(k) - f\big(x(k), \upsilon(k)\big) \tag{64}$$

where $f(x(k), \upsilon(k))$ is defined as in (4). In fact, $\Delta(k)$ is the NN approximation error, which is utilized to show the performance of the approximation of the action NN canceling the nonlinearity of system (63).

Both the action NN and the critic NN are implemented by a single-hidden layer feedforward NN. Without loss of generality, the initial weights for the input layer to the hidden layer are selected randomly within an interval of $[0, 1]$ and held as constants. Meanwhile, the initial weights for the output layer are selected randomly within an interval of $[-0.5, 0.5]$. There are 8 nodes in the hidden-layer of the action NN, i.e., the structure of the action NN is 3–8–1. Meanwhile, the structure of the critic NN is chosen to be 2–8–1. It is worth pointing out that selecting the structure of an NN is more of an art than science (Padhi, Unnikrishnan, Wang, & Balakrishnan, 2006). In this example, the number of neurons is derived by computer simulations, and we find that choosing 8 neurons for the hidden layer can lead to satisfactory simulation results.

The computer simulation results are shown by Figs. 2–5. Fig. 2 shows the trajectories of $y(k)$ and $y_d(k)$. Fig. 3 presents the tracking error $e(k)$. Fig. 4 illustrates the control input $u(k)$. Fig. 5 indicates the NN approximation error $\Delta(k)$. From Figs. 2–5, it is observed that the developed controller can make the system output $y(k)$ track the desired trajectory $y_d(k)$ very well. Meanwhile, the tracking error can converge to a small neighborhood of zero. It is also observed that the approximation of the action NN can cancel the nonlinearity of system (63) well except for several peaks, and all signals involved in the closed-loop system are bounded. Due to the property of NNs, one shall find that the peaks of the NN approximation error have a close connection with the performance of system dynamics. If $h\big(x(k), u(k)\big)$ is smooth enough, the peaks of the NN approximation error can be alleviated. In addition, from Fig. 2, we find that it costs about 10 time steps for the system output $y(k)$ to track the desired trajectory $y_d(k)$, which is faster than the method proposed in Zhang et al. (2002). Meanwhile, from Fig. 3, we find that the peak of the tracking error becomes smaller as time increases. Compared with Zhang et al. (2002), in our case, the tracking performance is much better.

### 5.2. Example 2

The purpose of this section is to further examine our method. Consider an MIMO nonaffine nonlinear DT system described by

$$\begin{aligned} x_{11}(k+1) &= x_{21}(k) \\ x_{12}(k+1) &= x_{22}(k) \\ x_{21}(k+1) &= 0.4x_{22}(k) - 0.3\cos(x_{21}(k)) \\ &\quad + 0.2u_1(k) - 0.1\tanh(u_2(k)) + d_1(k) \\ x_{22}(k+1) &= 0.1x_{11}(k) + 0.2u_2(k) \\ &\quad - 0.3\sin^2\big(x_{22}(k)\big)u_1(k) + d_2(k) \\ y_1(k) &= x_{11}(k) \\ y_2(k) &= x_{12}(k) \end{aligned} \tag{65}$$

where $x_1(k) = [x_{11}(k), x_{12}(k)]^{\mathrm{T}}$, $x_2(k) = [x_{21}(k), x_{22}(k)]^{\mathrm{T}}$, $u(k) = [u_1(k), u_2(k)]^{\mathrm{T}}$, $y(k) = [y_1(k), y_2(k)]^{\mathrm{T}}$ and $d(k) = [d_1(k), d_2(k)]^{\mathrm{T}}$, $d_i(k) = 0.01\sin(k)$, $i = 1, 2$.

The control objective is to control the system output $y(k)$ to track the desired trajectory $y_d(k)$ which is given by $y_d(k) = [2.6\sin(k\pi/200), 3\cos(k\pi/180)]^{\mathrm{T}}$.

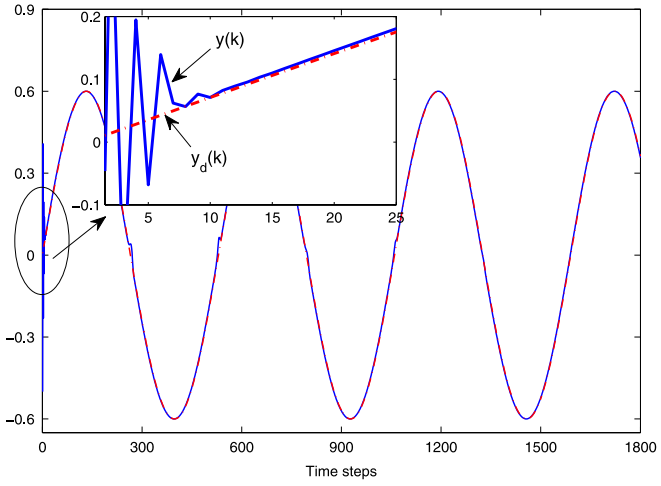


**Fig. 2.** Trajectories of $y(k)$ and $y_d(k)$ in Example 1.

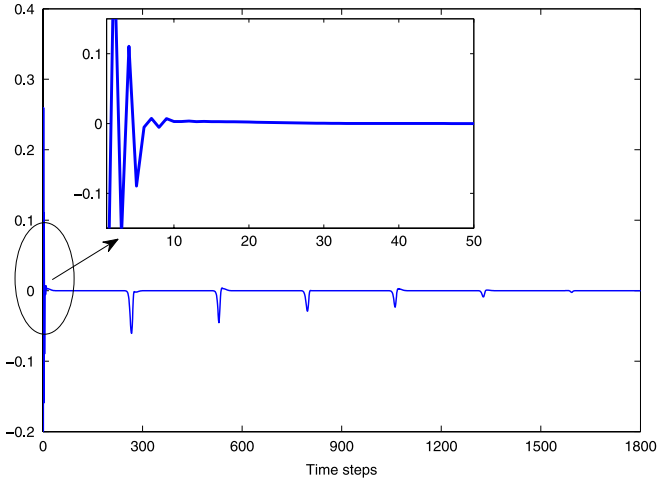


**Fig. 3.** Tracking error $e(k)$ in Example 1.

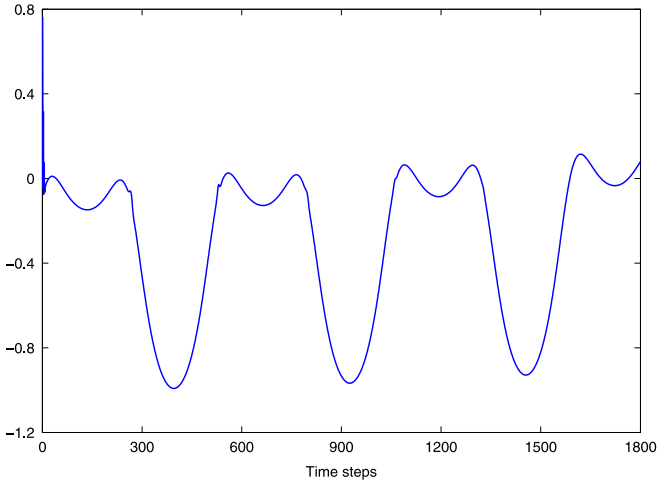


**Fig. 4.** Control input $u(k)$ in Example 1.

From (65), we can derive

$$\partial h\big(x(k), u(k)\big)/\partial u(k) = \begin{pmatrix} 0.2 & -0.4/(e^{x}+e^{-x})^2 \\ -0.3\sin^2\big(x_{22}(k)\big) & 0.2 \end{pmatrix}.$$

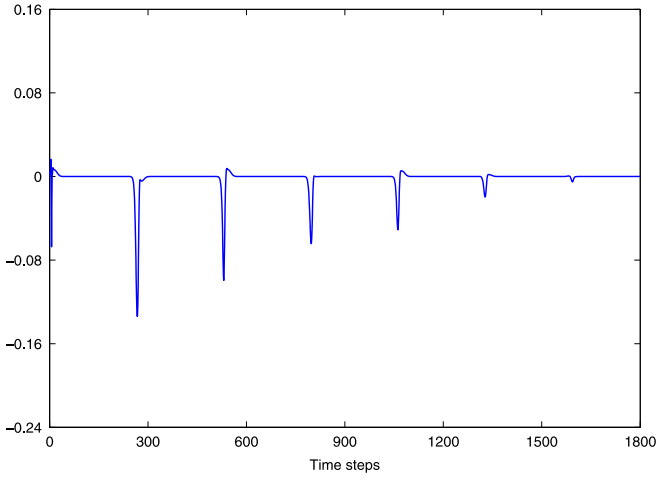


**Fig. 5.** NN approximation error $\Delta(k)$ in Example 1.

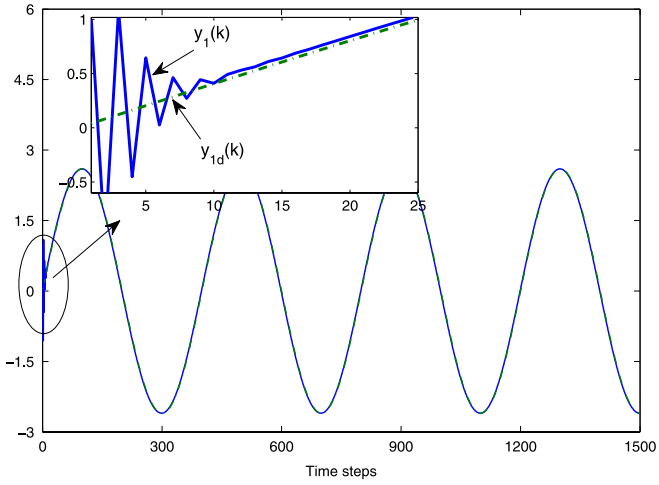


**Fig. 6.** Trajectories of $y_1(k)$ and $y_{1d}(k)$ in Example 2.

Then, one can easily find that $\partial h\big(x(k), u(k)\big)/\partial u(k)$ is a positive definite matrix. Meanwhile, one can also get

$$0.01I_2 \leq \partial h\big(x(k), u(k)\big)/\partial u(k) \leq 0.04I_2.$$

Choose $\Omega = [-3, 3] \times [-3, 3]$, and the initial state is selected to be $x_0 = [1, 0.31, 1, 0.31]^{\mathrm{T}}$. The design parameters are chosen to be the same as in Example 1. A single-hidden layer NN is applied to both the action NN and the critic NN, and the initial weights are selected in the same way as in Example 1. The structures of the action NN and the critic NN are designed as 6-30-2 and 4-24-2, respectively. In this example, the number of neurons in the hidden layers for both the action NN and the critic NN is obtained by computer simulations, and we find that it can lead to satisfactory simulation results.

The computer simulation results are presented in Figs. 6–10. Figs. 6 and 7 show the trajectories of $y_1(k)$ and $y_{1d}(k)$, and $y_2(k)$ and $y_{2d}(k)$, respectively. Fig. 8 describes tracking errors $e_0(k)$ and $e_1(k)$, which are the components of the tracking error vector $\boldsymbol{e}(k)$. Fig. 9 indicates the control input $u_i(k)$ $(i = 1, 2)$, which consist of the control vector $u(k)$. Fig. 10 describes the NN approximation error $\Delta_i(k)$ $(i = 1, 2)$, which consist of the NN approximation error vector $\Delta(k)$ defined as in (64). From the simulation results, it is observed that the system output $y(k)$ tracks the desired trajectory $y_d(k)$ very well, and the tracking errors converge to a small neighborhood of zero. It is also observed that the approximation of action NN can cancel the nonlinearity of system (65) rather well.

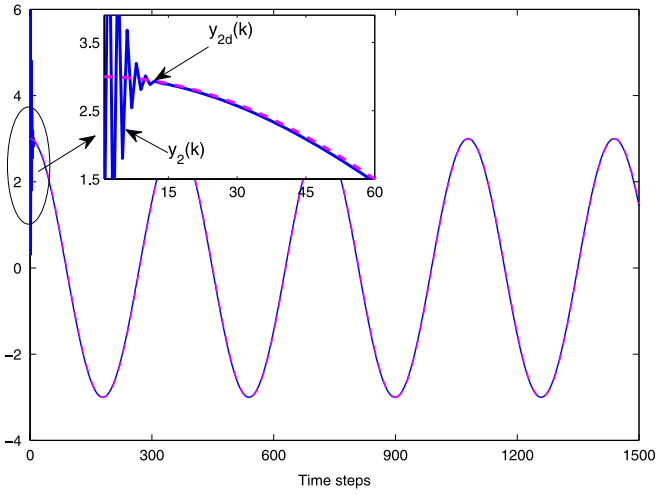


**Fig. 7.** Trajectories of $y_2(k)$ and $y_{2d}(k)$ in Example 2.

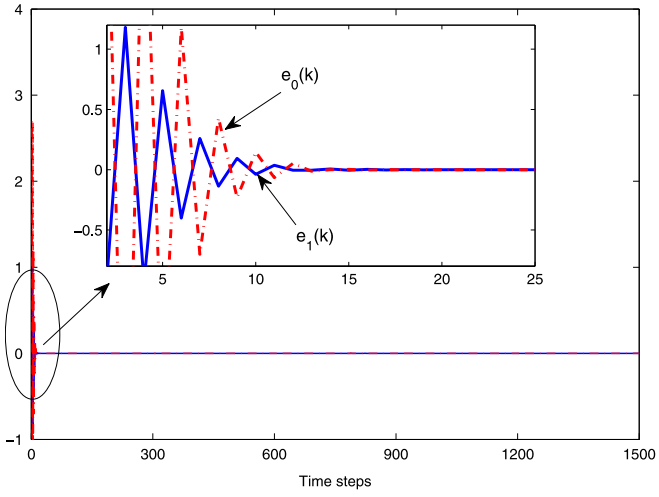


**Fig. 8.** Tracking errors $e_0(k)$ and $e_1(k)$ in Example 2.

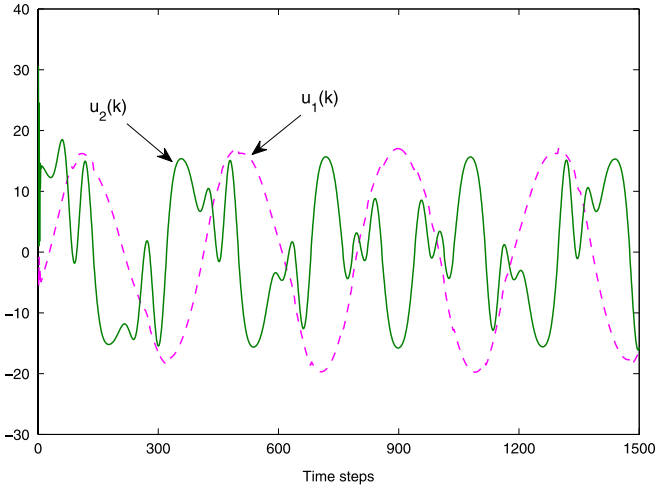


**Fig. 9.** Control input $u(k)$ in Example 2.

Moreover, we can find that all signals involved in the closed-loop system are bounded.

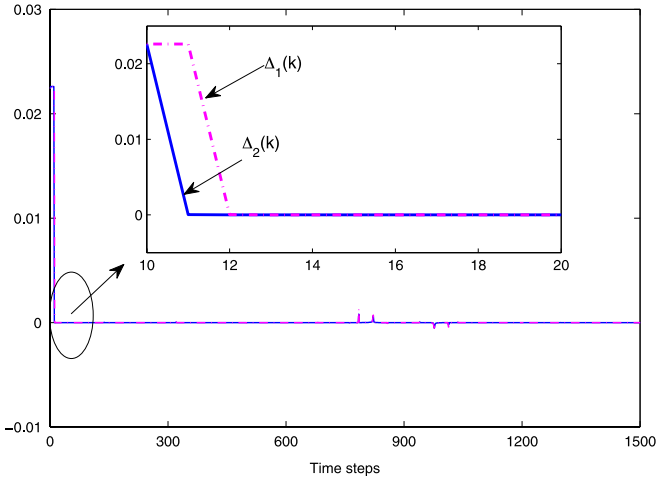


**Fig. 10.** NN approximation errors $\Delta_1(k)$ and $\Delta_2(k)$ in Example 2.

## 6. Conclusion

In this paper, we have developed an RL-based direct adaptive control law, which delivers a desired tracking performance for a class of unknown nonaffine nonlinear DT systems with unknown bounded disturbances. In order to utilize feedback linearization methods, the controller is divided into two parts: the first part of the controller is the feedback controller designed to stabilize linearized dynamics; the second part of the controller is the feedforward controller designed to cancel the nonlinearity of nonaffine nonlinear DT systems. The actor–critic architecture is employed in the controller design. Based on the presented architecture, the action NN and the critic NN are tuned online. By using Lyapunov's direct method, the uniform ultimate boundedness of both the closed-loop tracking errors and the NN weight estimates is demonstrated. The computer simulation results indicate that the developed online controller can perform control successfully and attain the desired performance. A limitation of the practical applicability of the presented method is that the full states of nonaffine nonlinear DT systems are required to be available. In addition, it should be mentioned that, in this paper, the system is unknown implying that the knowledge of the nonaffine nonlinear function in the system is unavailable. In our future work, we shall focus on relaxing this condition.

## References

Al-Tamimi, A., Lewis, F. L., & Abu-Khalaf, M. (2008). Discrete-time nonlinear HJB solution using approximate dynamic programming: convergence proof. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, *38*(4), 943–949.

Apostol, T. M. (1974). *Mathematical analysis* (2nd ed.). Cambridge, MA: Addison-Wesley.

Bellman, R. E. (1957). *Dynamic programming*. Princeton, New Jersey: Princeton University Press.

Bertsekas, D. P., & Tsitsiklis, J. N. (1996). *Neuro-dynamic programming*. Cambridge, MA: Athena Scientific.

Bhasin, S., Kamalapurkar, R., Johnson, M., Vamvoudakis, K. G., Lewis, F. L., & Dixon, W. E. (2013). A novel actor-critic-identifier architecture for approximate optimal control of uncertain nonlinear systems. *Automatica*, *49*(1), 82–92.

Chemachema, M. (2012). Output feedback direct adaptive neural network control for uncertain SISO nonlinear systems using a fuzzy estimator of the control error. *Neural Networks*, *36*, 25–34.

Chen, F. C., & Khalil, H. K. (1995). Adaptive control of a class of nonlinear discrete-time systems using neural networks. *IEEE Transactions on Automatic Control*, *40*(5), 791–801.

Deng, H., Li, H. X., & Wu, Y. H. (2008). Feedback-linearization-based neural adaptive control for unknown nonaffine nonlinear discrete-time systems. *IEEE Transactions on Neural Networks*, *19*(9), 1615–1625.

Ge, S. S., Hang, C. C., & Zhang, T. (1999). Adaptive neural network control of nonlinear systems by state and output feedback. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, *29*(6), 818–828.

Haykin, S. (2008). *Neural networks and learning machines* (3rd ed.). New Jersey: Prentice Hall.
He, P., & Jagannathan, S. (2005). Reinforcement learning-based output feedback control of nonlinear systems with input constraints. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, *35*(1), 150–154.
Horn, R. A., & Johnson, C. R. (1985). *Matrix analysis*. New York: Cambridge University Press.
Hovakimyan, N., Nardi, F., Calise, A., & Kim, N. (2002). Adaptive output feedback control of uncertain nonlinear systems using single-hidden-layer neural networks. *IEEE Transactions on Neural Networks*, *13*(6), 1420–1431.
Igelnik, B., & Pao, Y. H. (1995). Stochastic choice of basis functions in adaptive function approximation and the function-link net. *IEEE Transactions on Neural Networks*, *6*(6), 1320–1329.
Lewis, F. L., Jagannathan, S., & Yesildirek, A. (1999). *Neural network control of robot manipulators and nonlinear systems*. London: Taylor & Francis.
Lewis, F. L., Lendaris, G., & Liu, D. (2008). Special issue on approximate dynamic programming and reinforcement learning for feedback control. *IEEE Transactions on Systems, Man, Cybernetics, Part B, Cybernetics*, *38*(4), 896–897.
Lewis, F. L., & Vamvoudakis, K. G. (2011). Reinforcement learning for partially observable dynamic processes: adaptive dynamic programming using measured output data. *IEEE Transactions on Systems, Man, Cybernetics, Part B, Cybernetics*, *41*(1), 14–25.
Lewis, F. L., Vrabie, D., & Vamvoudakis, K. G. (2012). Reinforcement learning and feedback control: using natural decision methods to design optimal adaptive controllers. *IEEE Control Systems*, *32*(6), 76–105.
Lewis, F. L., Yesildirek, A., & Liu, K. (1996). Multilayer neural-net robot controller with guaranteed tracking performance. *IEEE Transactions on Neural Networks*, *7*(2), 388–399.
Liu, W., Venayagamoorthy, G. K., & Wunsch, D. C., II (2003). Design of an adaptive neural network based power system stabilizer. *Neural Networks*, *16*(5–6), 891–898.
Liu, D., Wang, D., & Yang, X. (2013). An iterative adaptive dynamic programming algorithm for optimal control of unknown discrete-time nonlinear systems with constrained inputs. *Information Sciences*, *220*(20), 331–342.
Liu, D., & Wei, Q. (2013). Finite-approximation-error-based optimal control approach for discrete-time nonlinear systems. *IEEE Transactions on Cybernetics*, *43*(2), 779–789.
Liu, D., Yang, X., & Li, H. (2013). Adaptive optimal control for a class of continuous-time affine nonlinear systems with unknown internal dynamics. *Neural Computing and Applications*, *23*(7–8), 1843–1850.
Liu, D., Zhang, Y., & Zhang, H. (2005). A self-learning call admission control scheme for CDMA cellular networks. *IEEE Transactions on Neural Networks*, *16*(5), 1219–1228.
Murray, J. J., Cox, C. J., Lendaris, G. G., & Saeks, R. (2002). Adaptive dynamic programming. *IEEE Transactions on Systems, Man and Cybernetics, Part C: Applications and Reviews*, *32*(2), 140–153.
Nakanishi, J., & Schaal, S. (2004). Feedback error learning and nonlinear adaptive control. *Neural Networks*, *17*(10), 1453–1465.
Narendra, K. S., & Mukhopadhyay, S. (1994). Adaptive control of nonlinear multivariable systems using neural networks. *Neural Networks*, *7*(5), 737–752.
Noriega, J. R., & Wang, H. (1998). A direct adaptive neural-network control for unknown nonlinear systems and its applications. *IEEE Transactions on Neural Networks*, *9*(1), 27–34.
Padhi, R., Unnikrishnan, N., Wang, X., & Balakrishnan, S. N. (2006). A single network adaptive critic (SNAC) architecture for optimal control synthesis for a class of nonlinear systems. *Neural Networks*, *19*(10), 1648–1660.
Park, J. H., Huh, S. H., Kim, S. H., Seo, S. J., & Park, G. T. (2005). Direct adaptive controller for nonaffine nonlinear systems using self-structuring neural networks. *IEEE Transactions on Neural Networks*, *16*(2), 414–422.
Prokhorov, D. V., & Wunsch, D. C. (1997). Adaptive critic designs. *IEEE Transactions on Neural Networks*, *8*(5), 997–1007.
Rudin, W. (1991). *Functional analysis* (2nd ed.). Singapore: McGraw-Hill, Inc.
Si, J., & Wang, Y. T. (2001). On-line learning control by association and reinforcement. *IEEE Transactions on Neural Networks*, *12*(2), 264–276.
Sutton, R. S., & Barto, A. G. (1998). *Reinforcement learning—an introduction*. Cambridge, MA: MIT Press.
Vamvoudakis, K. G., & Lewis, F. L. (2010). Online actor-critic algorithm to solve the continuous-time infinite horizon optimal control problem. *Automatica*, *46*(5), 878–888.
Vamvoudakis, K. G., & Lewis, F. L. (2011). Multi-player non-zero-sum games: online adaptive learning solution of coupled Hamilton–Jacobi equations. *Automatica*, *47*(8), 1556–1569.
Wang, D., Liu, D., & Wei, Q. (2012). Finite-horizon neuro-optimal tracking control for a class of discrete-time nonlinear systems using adaptive dynamic programming approach. *Neurocomputing*, *78*(1), 14–22.
Wang, D., Liu, D., Wei, Q., Zhao, D., & Jin, N. (2012). Optimal control of unknown nonaffine nonlinear discrete-time systems based on adaptive dynamic programming. *Automatica*, *48*(8), 1825–1832.
Wang, F. Y., Zhang, H., & Liu, D. (2009). Adaptive dynamic programming: an introduction. *IEEE Computational Intelligence Magazine*, *4*(2), 39–47.
Wei, Q., & Liu, D. (2012). An iterative $\varepsilon$-optimal control scheme for a class of discrete-time nonlinear systems with unfixed initial state. *Neural Networks*, *32*, 236–244.
Werbos, P. J. (1991). A menu of designs for reinforcement learning over time. In W. T. Miller, R. S. Sutton, & P. J. Werbos (Eds.), *Neural networks for control* (pp. 67–95). Cambridge, MA: MIT Press.
Werbos, P. J. (1992). Approximate dynamic programming for real-time control and neural modeling. In D. A. White, & D. A. Sofge (Eds.), *Handbook of intelligent control* (pp. 493–525). New York: Van Nostrand Reinhold.
Werbos, P.J. (2007). Using ADP to understand and replicate brain intelligence: the next level design. In *Proceedings of the IEEE symposium on approximate dynamic programming and reinforcement learning* (pp. 209–216). Honolulu, HI, April.
Werbos, P. J. (2008). ADP: the key direction for future research in intelligent control and understanding brain intelligence. *IEEE Transactions on Systems, Man, Cybernetics, Part B, Cybernetics*, *38*(4), 898–900.
Yang, Q., & Jagannathan, S. (2012). Reinforcement learning controller design for affine nonlinear discrete-time systems using approximators. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, *42*(2), 377–390.
Yang, X., Liu, D., & Huang, Y. (2013). Neural-network-based online optimal control for uncertain non-linear continuous-time systems with control constraints. *IET Control Theory & Applications*, *7*(17), 2037–2047.
Yang, L., Si, J., Tsakalis, K. S., & Rodriguez, A. A. (2009). Direct heuristic dynamic programming for nonlinear tracking control with filtered tracking error. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, *19*(6), 1617–1622.
Yang, Q., Vance, J. B., & Jagannathan, S. (2008). Control of nonaffine nonlinear discrete-time systems using reiforcement-learning-based linearly parameterized neural networks. *IEEE Transactions on Systems, Man, Cybernetics, Part B, Cybernetics*, *38*(4), 994–1001.
Yu, W. (2009). *Recent advances in intelligent control systems*. London: Springer-Verlag.
Zeidler, E. (1985). *Nonlinear functional analysis and its applications: part 1: fixed-point theorems*. New York: Springer-Verlag.
Zhang, J., Ge, S.S., & Lee, T.H. (2002). Direct RBF neural network control of a class of discrete-time nonaffine nonlinear systems. In *Proceeding of the American control conference* (pp. 424–429). Anchorage, AK, May.
Zhang, H., Wei, Q., & Liu, D. (2011). An iterative adaptive dynamic programming method for solving a class of nonlinear zero-sum differential games. *Automatica*, *47*(1), 207–214.