INTERNATIONAL JOURNAL OF ROBUST AND NONLINEAR CONTROL
Int. J. Robust Nonlinear Control 2015; **25**:1844–1861
Published online 13 April 2014 in Wiley Online Library (wileyonlinelibrary.com). DOI: 10.1002/rnc.3181

Direct adaptive control for a class of discrete-time unknown nonaffine nonlinear systems using neural networks

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

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

SUMMARY

In this paper, a direct adaptive state-feedback control approach is developed for a class of nonlinear systems in discrete-time (DT) domain. We study MIMO unknown nonaffine nonlinear DT systems and employ a two-layer NN to design the controller. By using the presented method, the NN approximation is able to cancel the nonlinearity of the unknown DT plant. Meanwhile, pretraining is not required, and the weights of NNs used in adaptive control are directly updated online. Moreover, unlike standard NN adaptive controllers yielding uniform ultimate boundedness results, the tracking error is guaranteed to be uniformly asymptotically stable by utilizing Lyapunov's direct method. Two illustrative examples are provided to demonstrate the effectiveness and the applicability of the theoretical results. Copyright © 2014 John Wiley & Sons, Ltd.

Received 29 May 2013; Revised 24 November 2013; Accepted 12 March 2014

KEY WORDS: adaptive control; discrete-time; nonaffine system; NN; feedback control; online learning; MIMO; Lyapunov method

1. INTRODUCTION

Neural networks are considered as powerful tools for modeling nonlinear functions because of their properties of nonlinearity, adaptivity, self-learning, and fault tolerance. Especially, with the introduction of the back-propagation learning algorithm by Werbos [1] (1974 PhD Thesis), multilayer NNs have become one of the most popular architectures for practical applications in many areas, including signal processing, pattern classification, and system identification. During the past several decades, research on NN-based adaptive control problems had drawn considerable attention [2–15], mainly credited to the establishment of the universal approximation theory [16], which guarantees that feedforward multilayer NNs can approximate any nonlinear function defined on a compact set to the prescribed accuracy. As a result, a huge number of methods have been developed and successfully applied to adaptive control problems.

Nevertheless, most of the previous approaches focused on nonlinear autoregressive moving average with exogenous input systems. This form is not convenient for purposes of adaptive control using NNs because of its lack of knowledge of nonlinear dynamic systems. Hence, state-space representation of nonlinear systems becomes popular. With the development of the theory of adaptive control, an increasing number of researchers pay their attention to affine nonlinear systems and little attention to nonaffine nonlinear systems. A typical feature of the affine nonlinear system is that the output of this type of systems is linear with respect to the control input. Consequently, it is easy to design an adaptive controller for such a nonlinear system by using feedback linearization methods. However, a significant difference between affine nonlinear systems and nonaffine nonlinear systems

*Correspondence to: Derong Liu, The State Key Laboratory of Management and Control for Complex Systems, Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China.
†E-mail: derong.liu@ia.ac.cn

is that the output of the nonaffine nonlinear system depends nonlinearly on the control signal. Under this circumstance, feedback linearization methods cannot be implemented. Consequently, it gives birth to a great challenge for researchers to design efficient controllers for such nonaffine nonlinear systems, which aims at deriving the prescribed output. Moreover, if the nonaffine nonlinear system is unknown, it will be more intractable to be dealt with.

Recently, NN adaptive control for nonaffine nonlinear systems has been an active research area for its practical interest. In [17] and [18], an adaptive state and output feedback neuro-control was developed for SISO nonaffine nonlinear continuous-time (CT) systems. By using a high-gain observer, semi-global uniform ultimate boundedness of the closed-loop system was obtained, and the output tracking error converging to an adjustable neighborhood of the origin was derived, respectively. In [19] and [20], NN-based adaptive feedback controls were proposed for SISO nonaffine nonlinear CT systems and SISO uncertain nonlinear CT systems, respectively. Uniform ultimate boundedness of all signals involved in the closed-loop system was derived through Lyapunov's direct method. In [21], a direct adaptive control was presented for SISO nonaffine nonlinear CT systems. Based on self-structuring NNs and Lyapunov's direct method, uniform asymptotic stability of the tracking error was obtained.

However, little literature about nonaffine nonlinear discrete-time (DT) systems exists. It is well known that DT adaptive control design is far more complex than CT adaptive controller design, due primarily to the fact that DT Lyapunov derivatives are quadratic in the state's first difference, while for CT nonlinear systems, the Lyapunov derivative is linear in the state's derivative [22]. Although there exists literature corresponding to MIMO affine nonlinear DT systems [23–26], there are rather few investigations on MIMO nonaffine nonlinear DT systems. Moreover, uniform asymptotic stability of the tracking error of DT closed-loop systems is seldom obtained. In this paper, we develop a direct adaptive state-feedback control methodology for a class of MIMO nonaffine nonlinear DT systems using the two-layer NN.

The main contributions of this paper include the following:

1. To the best of our knowledge, it is the first time that a direct neuro-based adaptive state-feedback control for unknown MIMO nonaffine nonlinear DT systems is studied.
2. Unlike standard NN-based adaptive controllers generally yielding uniform ultimate boundedness results, we derive that the adaptive NN controller can guarantee tracking errors to be uniformly asymptotically stable (UAS) based on Lyapunov's direct method.
3. Pretraining is not required, and the weights of NNs used in the adaptive control are directly updated online.
4. By using the Implicit Function Theorem, the NN approximation is guaranteed to cancel the nonlinearity of unknown MIMO nonlinear DT systems. Based on the developed approach, feedback linearization methods can be used to design a robust adaptive controller for unknown MIMO nonaffine nonlinear systems.
5. Persistent excitation (PE) assumption is not required in the adaptive NN controller design.

The rest of the paper is organized as follows. In Section 2, preliminaries of adaptive control problems are provided. In Section 3, the control structure is first developed. Then, NN implementation is discussed, and the control algorithm is proposed. Finally, the stability analysis is developed. In Section 4, two simulation examples are presented to verify the effectiveness of the established theorem in Section 3. Finally, in Section 5, several conclusions are drawn.

2. PRELIMINARIES OF ADAPTIVE CONTROL PROBLEMS

For convenience, the notations, which will be used throughout the paper, are listed as follows:

- $\mathbb{R}$ denotes the real number, and $\mathbb{R}^m$ and $\mathbb{R}^{m \times n}$ denote the real m-vectors and the real $m \times n$ matrices, respectively. T represents transposition.
- Ω and $\Omega_i (i = 1, 2, \ldots, n)$ are compact sets of $\mathbb{R}^{mn}$ and $\mathbb{R}^m$, respectively. Let $\Omega' \subset \Omega$ and $U \subset \mathbb{R}^m, \Omega' \times U = \{(x, u) | x \in \Omega', u \in U\}$ stands for the Cartesian product of Ω' and U.

DOI: 10.1002/rnc

- $\|\cdot\|$ stands for any suitable norm. When z is a vector, $\|z\|$ denotes Euclidean norm of z. When A is a matrix, $\|A\|$ denotes two-norm of A.
- I_m and I_{mn} represent the $m \times m$ identity matrix and the $mn \times mn$ identity matrix, respectively. $C^m(\Omega) = \left\{ f^{(m)} \in C \,|\, f : \Omega \to \mathbb{R}^m \right\}$.

2.1. Description of discrete-time nonaffine nonlinear systems

For purpose of the present paper, we consider an mnth-order MIMO nonaffine nonlinear DT system described by

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

with the state $x(k) = \left[x_1^{\mathrm{T}}(k), x_2^{\mathrm{T}}(k), \ldots, x_n^{\mathrm{T}}(k)\right]^{\mathrm{T}} \in \Omega \subset \mathbb{R}^{mn}$, and each $x_i(k) \in \Omega_i \subset \mathbb{R}^m$, $i = 1, 2, \ldots, n$. $u(x(k)) \in \mathbb{R}^m$ is the control input, which is the continuous function with respect to $x(k)$. For convenience, we denote that $\upsilon(k) = u(x(k))$. $y(k) \in \mathbb{R}^m$ is the output vector, and $F(x(k), \upsilon(k)) \in \mathbb{R}^m$ is an unknown nonaffine nonlinear function with $F(0, 0) = 0$. It is assumed that the state $x(k)$ is available at the kth step for the state-feedback controller. Meanwhile, we need the following assumption, which is made for the controllability of the system.

Assumption 1
The $m \times m$ matrix $\partial F(x(k), \upsilon(k))/\partial \upsilon(k)$ is a positive definite matrix for $\forall (x(k), \upsilon(k)) \in \Omega \times \mathbb{R}^m$ with the compact region $\Omega \subset \mathbb{R}^{mn}$.

Remark 1
From Assumption 1, one can easily derive that

$$\det \frac{\partial F(x(k), \upsilon(k))}{\partial \upsilon(k)} \neq 0 \tag{2}$$

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

Before continuing our discussion, we provide the following definitions that are utilized throughout this paper. These definitions are introduced in [22, 27], and readers can refer to these literature for further details.

Definition 1
The equilibrium point x_e of system (1) is said to be uniformly ultimately bounded if there exists a compact set $\Omega \subset \mathbb{R}^{mn}$ such that for all $x_0 \in \Omega (x_0 = x(k_0), k_0$ is the initial time), there exists a bound $\epsilon > 0$ and a positive number $N(x_0, \epsilon)$ such that $\|x(k) - x_e\| < \epsilon$ for all $k \geqslant k_0 + N$.

Definition 2
The equilibrium point x_e of system (1) is stable in the sense of Lyapunov at $k_0 \in \mathbb{R}$ if for $\forall \epsilon > 0$, there exists a $\delta(\epsilon, k_0) > 0$ such that $\|x_0 - x_e\| < \delta(\epsilon, k_0)(x_0 = x(k_0), k_0$ is the initial time) implies that $\|x(k) - x_e\| < \epsilon$ for $k \geqslant k_0 \geqslant 0$. The equilibrium point is said to be uniformly stable if the conditions hold for all k_0. The equilibrium point x_e of system (1) is said to be UAS if it is uniformly stable and there is a positive constant c, independent of k_0, such that $\|x_0 - x_e\| < c$ implies that $\lim\limits_{k \to \infty} \|x(k) - x_e\| = 0$.

Objective of control: The main objective of this paper is to develop a robust adaptive NN controller for unknown MIMO nonaffine DT system (1) such that the tracking error between the system state $x(k)$ and the desired trajectory $x_d(k) = \left[x_{1d}^{\mathrm{T}}(k), x_{2d}^{\mathrm{T}}(k), \ldots, x_{nd}^{\mathrm{T}}(k)\right]^{\mathrm{T}} \in \Omega \subset \mathbb{R}^{mn}$ is UAS. Or equivalently, the closed-loop system output $y(k) \in \mathbb{R}^m$ can asymptotically track the desired trajectory $y_d(k) \in \mathbb{R}^m$. Meanwhile, all the signals involved in the corresponding closed-loop system are guaranteed to be bounded.

2.2. *Feedback linearizing controller design*

Because system (1) is an unknown nonaffine nonlinear system, the controller cannot be directly designed by feedback linearization methods. The purpose of this section is to develop a controller that is similar to the feedback linearizing controller of the affine nonlinear system. The presented approach for designing such a controller is based on the schemes proposed in [20] and [21], which deal with CT adaptive control problems.

From system (1), we have that

$$\begin{aligned} y(k+n) &= F(x(k), \upsilon(k)) \\ &= \alpha\upsilon(k) + [F(x(k), \upsilon(k)) - \alpha\upsilon(k)] \end{aligned} \tag{3}$$

where $\alpha > 0$ is a design constant. Denote $h(x(k), \upsilon(k)) = F(x(k), \upsilon(k)) - \alpha\upsilon(k)$. The controller for system (1) is chosen as

$$\upsilon(k) = \frac{1}{\alpha}\left[\upsilon_l(k) - \upsilon_c(k) + \upsilon_r(k)\right] \tag{4}$$

where $\upsilon_l(k)$ is a feedback controller to stabilize the linearized dynamics, $\upsilon_c(k)$ is a feedforward controller designed to cancel the unknown nonlinear term $h(x(k), \upsilon(k))$ via the design of a two-layer NN, and $\upsilon_r(k)$ is a function to be detailed subsequently, which provides robustness.

By using (4), one shall find that (3) can be rewritten as

$$y(k+n) = \upsilon_l(k) + [h(x(k), \upsilon(k)) - \upsilon_c(k)] + \upsilon_r(k) \tag{5}$$

Remark 2

As mentioned before, for convenience, we denote $\upsilon(k) = u(x(k))$. Hence, $\upsilon_l(k)$, $\upsilon_c(k)$, and $\upsilon_r(k)$ should be the functions with respect to $x(k)$. That is, in this paper, $\upsilon_l(k) = u_l(x(k))$, $\upsilon_c(k) = u_c(x(k))$, and $\upsilon_r(k) = u_r(x(k))$.

Assumption 2

Let the desired state trajectory of system (1) be $x_d(k) = \left[x_{1d}^{\mathrm{T}}(k), x_{2d}^{\mathrm{T}}(k), \ldots, x_{nd}^{\mathrm{T}}(k)\right]^{\mathrm{T}}$, $x_{id}(k)$ is selected arbitrarily and satisfies $x_{id}(k+1) = x_{(i+1)d}(k), i = 1, \ldots, n-1$. Meanwhile, the desired output trajectory $y_d(k)$ is bounded by a known smooth function over the compact set Ω.

Remark 3

From Assumption 2 and system (1), we can obtain that $x_{(i+1)d}(k) = y_d(k+i), i = 0, \ldots, n-1$. Hence, we can define the tracking error as

$$\begin{aligned} e_i(k) &= y_d(k+i) - y(k+i) \\ &= x_{(i+1)d}(k) - x_{(i+1)}(k) \end{aligned} \tag{6}$$

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

In view of the task of $\upsilon_l(k)$, we define $\upsilon_l(k)$ as

$$\upsilon_l(k) = y_d(k+n) + \lambda_1 e_{n-1}(k) + \cdots + \lambda_n e_0(k) \tag{7}$$

where $e_{n-1}(k), \ldots, e_0(k)$ are the delayed values of the tracking error $e_n(k)$, and $\lambda_1, \ldots, \lambda_n$ are constant matrices selected such that $|z^n + \lambda_1 z^{n-1} + \cdots + \lambda_n|$ is stable, that is, all the solutions of $|z^n + \lambda_1 z^{n-1} + \cdots + \lambda_n| = 0$ are located inside the unit circle centered at the origin.

Lemma 1

Assume that the tracking error $e_i(k)$ is defined as in (6) and $\upsilon_l(k)$ is proposed as in (7). Then, by using (6) and (7), we can derive the error dynamics as

$$\boldsymbol{e}(k+1) = \tilde{A}\boldsymbol{e}(k) + \tilde{B}\left([\upsilon_c(k) - h(x(k), \upsilon(k))] - \upsilon_r(k)\right) \tag{8}$$

DOI: 10.1002/rnc

where

$$\boldsymbol{e}(k) = \left[e_0^{\mathrm{T}}(k), \ldots, e_{n-1}^{\mathrm{T}}(k)\right]^{\mathrm{T}}, \quad \tilde{A} = A \otimes I_m, \quad \tilde{B} = B \otimes I_m,$$
$$A = \begin{bmatrix} 0 & 1 & \cdots & 0 \\ \vdots & \vdots & & \vdots \\ -\lambda_n & -\lambda_{n-1} & \cdots & -\lambda_1 \end{bmatrix}, \quad B = \begin{bmatrix} 0 \\ \vdots \\ 1 \end{bmatrix} \tag{9}$$

Proof
By the definition of $e_i(k)$ in (6), we have that

$$\begin{aligned} e_i(k) &= y_d(k+i) - y(k+i) \\ &= y_d[(k+1)+(i-1)] - y[(k+1)+(i-1)] \\ &= e_{i-1}(k+1) \end{aligned} \tag{10}$$

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

Meanwhile, from (5) and (7), we can obtain that

$$e_n(k) = -\boldsymbol{\lambda}^{\mathrm{T}}\boldsymbol{e}(k) + \left[\upsilon_c(k) - h(x(k), \upsilon(k))\right] - \upsilon_r(k)$$

where $\boldsymbol{\lambda} = [\lambda_n, \ldots, \lambda_1]^{\mathrm{T}}$. Observing $e_n(k) = e_{n-1}(k+1)$, we derive that

$$e_{n-1}(k+1) = -\boldsymbol{\lambda}^{\mathrm{T}}\boldsymbol{e}(k) + [\upsilon_c(k) - h(x(k), \upsilon(k))] - \upsilon_r(k) \tag{11}$$

Therefore, by utilizing (10) and (11), we can obtain that

$$\begin{cases} e_0(k+1) = e_1(k), \\ \qquad \vdots \\ e_{n-1}(k+1) = -\boldsymbol{\lambda}^{\mathrm{T}}\boldsymbol{e}(k) - \upsilon_r(k) + [\upsilon_c(k) - h(x(k), \upsilon(k))] \end{cases} \tag{12}$$

Rewriting (12) in the vector form, and noting $\boldsymbol{e}(k) \in \mathbb{R}^{mn}$, we can derive (8) and (9). □

Remark 4
If one can find a feedforward control $\upsilon_c(k)$ to cancel the nonlinear function $h(x(k), \upsilon(k))$ well, that is, $\upsilon_c(k) = h(x(k), \upsilon(k))$, and let $\upsilon_r(k) = 0$, then the closed-loop system becomes a linear system $\boldsymbol{e}(k+1) = \tilde{A}\boldsymbol{e}(k)$. Obviously, $\tilde{A}$ is able to make the linear system $\boldsymbol{e}(k+1) = \tilde{A}\boldsymbol{e}(k)$ stable (for short, $\tilde{A}$ is a stable matrix), for $\lambda_1, \ldots, \lambda_n$ are constant matrices selected such that $|z^n + \lambda_1 z^{n-1} + \cdots + \lambda_n|$ is stable. Accordingly, if there exist $\upsilon_c(k) = h(x(k), \upsilon(k))$ and the robust term $\upsilon_r(k) = 0$, then $\upsilon_l(k)$ can make the tracking error exponentially converge to zero as time increases. Therefore, the design of $\upsilon_l(k)$ in (7) makes sense.

Although the design of the feedback control $\upsilon_l(k)$ is reasonable, one may doubt whether the feedforward control $\upsilon_c(k)$ exists or not. In what follows, we show that the feedforward control $\upsilon_c(k)$ does exist. For convenience, the Implicit Function Theorem for vector-valued functions is first presented, which will be used in the subsequent proof.

Lemma 2 (Implicit Function Theorem [28]*)*
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 $\det[\partial \boldsymbol{f}(\boldsymbol{x}_0, \boldsymbol{y}_0)/\partial \boldsymbol{y}_0] \neq 0$. Then, there exists an 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 $\forall(\boldsymbol{x}_0, \boldsymbol{y}_0) \in T_0$, $\boldsymbol{f}(\boldsymbol{x}_0, \boldsymbol{g}(\boldsymbol{x}_0)) = 0$.

It should be emphasized that Lemma 2 is actually a local conclusion. That is, Lemma 2 is valid when $(x, y) \in T_0 \subset S$. In what follows, we present a generalized form of Implicit Function Theorem, which is a corollary of Lemma 2. We call the corollary as the Generalized Implicit Function Theorem.

DOI: 10.1002/rnc

Corollary 1 (Generalized Implicit Function Theorem)
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 $f(\boldsymbol{x}, \boldsymbol{y}) = 0$ and $\det[\partial f(\boldsymbol{x}, \boldsymbol{y})/\partial \boldsymbol{y}](\boldsymbol{x}, \boldsymbol{y}) \neq 0$ for $\forall(\boldsymbol{x}, \boldsymbol{y}) \in S$. Then, there exists a continuous vector-valued function $\boldsymbol{y} = \boldsymbol{g}(\boldsymbol{x})$ such that $\boldsymbol{f}(\boldsymbol{x}, \boldsymbol{y}) = 0$, $\forall(\boldsymbol{x}, \boldsymbol{y}) \in S$. (The proof can be found in [29].)

Denote $G(x(k), \upsilon(k)) = h(x(k), \upsilon(k)) - \upsilon_c(k)$, and let $G(x(k), \upsilon(k)) = 0$. From (3) and (4), we can find that $G(x(k), \upsilon(k))$ is actually a function with respect to $x(k)$, $\upsilon_c(k)$, $\upsilon_r(k)$ and $\upsilon_l(k)$. Therefore, $G(x(k), \upsilon(k))$ can be represented as

$$G(x(k), \upsilon_f(k), \upsilon_c(k)) = h\left(x(k), \frac{\upsilon_f(k) - \upsilon_c(k)}{\alpha}\right) - \upsilon_c(k) = 0 \tag{13}$$

where $\upsilon_f(k) = \upsilon_l(k) + \upsilon_r(k)$.

Theorem 1
Assume that the following matrix inequality holds:

$$\alpha\kappa_1 I_m \leqslant \frac{\partial F(x(k), \upsilon(k))}{\partial \upsilon(k)} \leqslant \alpha\kappa_2 I_m \tag{14}$$

where $0 < \kappa_1 < \kappa_2 \leqslant 2$. Then, there exists a compact set $\Omega' \subset \Omega$ and the unique control $\upsilon_c(x(k), \upsilon_f(k))(\upsilon_c(k)$ is a function with respect to $x(k)$ and $\upsilon_f(k))$ that satisfies (13) for $\forall(x(k), \upsilon_f(k)) \in \Omega' \times U$, where $U \subset \mathbb{R}^m$ is a compact set.

Proof
In order to utilize Lemma 2, we divide the proof into two parts. First, we show that there exists a solution of (13) (write the solution as $\upsilon_c^*(k)$). Then, we prove that $\partial G\left(x(k), \upsilon_f(k), \upsilon_c^*(k)\right) / \partial \upsilon_c^*(k)$ is a nonsingular matrix.

(a) Existence of the solution $\upsilon_c^*(k)$ of (13).
Noting the expression of (13), one shall find that if the conclusion is true, then

$$\upsilon_c^*(k) = h\left(x(k), \frac{\upsilon_f(k) - \upsilon_c^*(k)}{\alpha}\right) \tag{15}$$

Obviously, $\upsilon_c^*(k)$ is the fixed point of (15). Therefore, we just need to prove that there exists a fixed point for the operator $h(x(k), \cdot)$ on the given compact set $U \subset \mathbb{R}^m$.

Because $x(k)$ is defined on the compact set Ω, and $\upsilon(k)$ is the continuous function with respect to $x(k)$, by functional analysis [30], we obtain that $\upsilon(\Omega)$ is a compact set on $\mathbb{R}^m$. Hence, we can select $U = \upsilon(\Omega)$.

Observe that

$$\begin{aligned}\left\|\frac{\partial h(x(k), \boldsymbol{\upsilon(k)})}{\partial \upsilon_c^*(k)}\right\| &= \left\|\frac{\partial h(x(k), \upsilon(k))}{\partial \upsilon(k)} \frac{\partial \upsilon(k)}{\partial \upsilon_c^*(k)}\right\| \\ &= \left\|\left(\frac{\partial F(x(k), \upsilon(k))}{\partial \upsilon(k)} - \alpha I_m\right) \cdot \left(-\frac{I_m}{\alpha}\right)\right\| \\ &= \left\|I_m - \frac{\partial F(x(k), \upsilon(k))}{\alpha \partial \upsilon(k)}\right\|\end{aligned} \tag{16}$$

By using the matrix inequality (14), we have that

$$\begin{cases} I_m - \dfrac{\partial F(x(k), \upsilon(k))}{\alpha \partial \upsilon(k)} \geqslant (1 - \kappa_2) I_m, \\ I_m - \dfrac{\partial F(x(k), \upsilon(k))}{\alpha \partial \upsilon(k)} \leqslant (1 - \kappa_1) I_m \end{cases} \tag{17}$$

DOI: 10.1002/rnc

Therefore, by using (16) and (17), we obtain that

$$\left\| \frac{\partial h(x(k), \upsilon(k))}{\partial \upsilon_c^*(k)} \right\| \leqslant 1$$

Noting that $(x(k), \upsilon(k))$ is defined on the compact set $\Omega \times U$ and $h(x(k), \upsilon(k))$ is a continuous function, we can conclude that $h(\Omega \times U)$ is a compact set on $\mathbb{R}^{nm} \times \mathbb{R}^m$. Hence, $h(x(k), \cdot)$ is a completely continuous operator on U[31]. By using Schauder's Fixed-Point Theorem [31], we can draw the conclusion that there exists at least one fixed point of the operator $h(x(k), \cdot)$ on the compact set U. That is, the solution $\upsilon_c^*(k)$ of (13) exists.

(b) Proof of non-singularity of the $m \times m$ matrix $\partial G\left(x(k), \upsilon_f(k), \upsilon_c^*(k)\right) / \partial \upsilon_c^*(k)$.
Notice that

$$\begin{aligned} \left.\frac{\partial G(x(k), \upsilon_f(k), \upsilon_c(k))}{\partial \upsilon_c(k)}\right|_{\upsilon_c(k)=\upsilon_c^*(k)} &= \left.\frac{\partial\left(F(x(k), \upsilon(k)) - \alpha \upsilon(k)\right)}{\partial \upsilon(k)} \cdot \frac{\partial \upsilon(k)}{\partial \upsilon_c(k)}\right|_{\upsilon_c(k)=\upsilon_c^*(k)} - I_m \\ &= -\left.\frac{\partial F(x(k), \upsilon(k))}{\partial \upsilon(k)}\right|_{\upsilon_c(k)=\upsilon_c^*(k)} \end{aligned} \tag{18}$$

By utilizing (2), we can conclude that $\partial G(x(k), \upsilon_f(k), \upsilon_c(k)) / \partial \upsilon_c(k)$ in (18) is nonsingular.

Consequently, with the aid of (a) and (b) and utilizing Lemma 2, we can conclude that there exists a unique control $\upsilon_c(x(k), \upsilon_f(k))$ satisfying (13) for $\forall (x(k), \upsilon_f(k)) \in \Omega' \times U (\Omega' \subset \Omega)$. □

Remark 5
From Assumption 1, one shall notice that the matrix inequality (14) is actually a property of the positive definite matrix $\partial F(x(k), \upsilon(k))/\partial \upsilon(k)$, which is utilized in [22, 27]. Therefore, the assumption about the matrix inequality (14) makes sense. In addition, it should be mentioned that we do not need the operator $h(x(k), \cdot)$ to be strictly contractive, which is a more relaxed condition than [20, 21].

Remark 6
From Theorem 1, one shall find that the solution of (13) is unique for the given local domain Ω'. Nevertheless, it does not mean that there are no other solutions $\upsilon_c(k)$ of (13) on the set $\Omega \setminus \Omega'$. In fact, if Assumption 1 holds, then the solution $\upsilon_c(k)$ of (13) exists on the whole Ω. Now, we prove this conclusion as follows: Noting that (2) is valid for $\forall (x(k), \upsilon(k)) \in \Omega \times \mathbb{R}^m$, we can obtain that

$$\begin{aligned} \det\left[\frac{\partial G(x(k), \upsilon_f(k), \upsilon_c(k))}{\partial \upsilon_c(k)}\right] &= \det\left[\frac{\partial(F(x(k), \upsilon(k)) - \alpha \upsilon(k))}{\partial \upsilon(k)} \frac{\partial \upsilon(k)}{\partial \upsilon_c(k)} - I_m\right] \\ &= (-1)^m \det\left[\frac{\partial F(x(k), \upsilon(k))}{\partial \upsilon(k)}\right] \neq 0 \end{aligned} \tag{19}$$

for $\forall (x(k), \upsilon(k)) \in \Omega \times U$. Combining (13) and (19) and utilizing Corollary 1, we can conclude that there exists a solution of (13) on the whole Ω. It is worth pointing out that in this sense, the solution of (13) may not be unique on the whole Ω. Theorem 1 shows the solution $\upsilon_c(k)$ of (13) is unique on the given local domain Ω' and not on the whole Ω.

3. NN CONTROLLER DESIGN

The purpose of this section is to develop an implementation of the controller $\upsilon_c(k)$. From the aforementioned analysis, we know that $\upsilon_c(k)$ is a nonlinear controller, which estimates the unknown nonlinear function $h(x(k), \upsilon(k))$. Nevertheless, the lack of a general structure makes the design of such a nonlinear controller rather intractable. In fact, the nonlinear controller design is a nonlinear

DOI: 10.1002/rnc

function approximation problem. Hence, according to the universal approximation property of NNs [32], a two-layer feedforward NN is employed to approximate $h(x(k), \upsilon(k))$. In the remainder of this section, we shall first give the preliminaries of the two-layer feedforward NN. Then, the structures of the NN controller and the error system dynamics are presented. Meanwhile, the control algorithm for system (1) is developed. Finally, the weight update laws for guaranteeing the tracking performance are established.

3.1. Two-layer feedforward NNs

A general function $g(x) \in C^m(\Omega)$ can be written as

$$g(x) = W^{\mathrm{T}}\sigma\left(V^{\mathrm{T}}x\right) + \varepsilon(x) \tag{20}$$

with $\sigma(\cdot)$ the activation function, $\varepsilon(x)$ the NN functional reconstruction error, and V and W the weights for the input layer to the hidden layer and the hidden layer to the output layer, respectively. The number of hidden layer node is denoted as N. It is shown in [33] that if the hidden layer weight vector V is selected initially at random and then keep it unchanged and if N is large enough, the NN approximation error $\varepsilon(x)$ can be arbitrarily small. That is, there exists a positive number N_0 such that $N \geqslant N_0$ implies $\|\varepsilon(x)\| \leqslant \varepsilon_N$. Typically, activation functions for $\sigma(\cdot)$ are bounded, measurable, and nondecreasing functions from the real numbers onto $[-1, 1]$, which include, for instance, hyperbolic tangent function $\sigma(x) = (e^x - e^{-x})/(e^x + e^{-x})$. Because the hidden layer weights are generally kept as constants, (20) is rewritten as

$$g(x) = W^{\mathrm{T}}\sigma(x) + \varepsilon(x) \tag{21}$$

where $\sigma(x) = \sigma(V^{\mathrm{T}}x), x \in \Omega$.

3.2. Structure of NN controller and error system dynamics

Suppose that the nonlinear function $h(x(k), \upsilon(k))$ can accurately be represented by

$$h(x(k), \upsilon(k)) = W^{\mathrm{T}}\sigma(V^{\mathrm{T}}z(k)) + \varepsilon(k) = W^{\mathrm{T}}\sigma(z(k)) + \varepsilon(k) \tag{22}$$

where $V \in \mathbb{R}^{(n+1)m\times N_1}$ and $W \in \mathbb{R}^{N_1\times m}$ are the ideal weight vectors for the input layer to the hidden layer and the hidden layer to the output layer of the NN, respectively, N_1 is the number of the nodes in the hidden layer, and $z(k) = \left[x^{\mathrm{T}}(k)\ \upsilon^{\mathrm{T}}(k)\right]^{\mathrm{T}}$. The activation function of the hidden layer $\sigma(V^{\mathrm{T}}z(k))$ is denoted as $\sigma(z(k))$ for brief; for the hidden layer, weight vector is generally kept as constants. In addition, $\varepsilon(k)$ is the NN approximation error.

Because the design of the controller $\upsilon_c(k)$ is actually to estimate the nonlinear function $h(x(k), \upsilon(k))$, we can select $\upsilon_c(k)$ as the output of NNs. That is,

$$\upsilon_c(k) = \hat{W}^{\mathrm{T}}(k)\sigma(z(k)) \tag{23}$$

with $\hat{W}(k)$ the estimates of the ideal weight W.

The error in the weight parameters during the estimation is given by

$$\tilde{W}(k) = W - \hat{W}(k) \tag{24}$$

By using (8), the closed-loop error dynamics becomes

$$\boldsymbol{e}(k+1) = \tilde{A}\boldsymbol{e}(k) + \tilde{B}\left[\tilde{W}(k)\phi(z(k)) - \upsilon_r(k) - \varepsilon(k)\right] \tag{25}$$

where $\phi(z(k)) = -\sigma(z(k))$.

Assumption 3
The NN approximation error $\varepsilon(k)$ satisfies the following inequality

$$\varepsilon^{\mathrm{T}}(k)\varepsilon(k) \leqslant \varepsilon_M(k) = \beta^*\boldsymbol{e}^{\mathrm{T}}(k)\boldsymbol{e}(k) \tag{26}$$

where β^* is a bounded constant value such that $\|\beta^*\| \leqslant \beta_M$.

Remark 7
As stated in Section 3.1, the NN approximation error $\varepsilon(k)$ can be arbitrarily small as long as the number of the hidden layer node N is large enough. Meanwhile, from (25) (and subsequent Figure 2), we know that $\varepsilon(k)$ is closely linked with $\boldsymbol{e}(k)$. Hence, we can assume that $\varepsilon(k)$ lies in a small gain-type norm-bounded conic sector [34], which satisfies the inequality (26). From a mathematical perspective, it is generally considered as a mild assumption in comparison with [22], which assumes that $\varepsilon(k)$ is bounded by a known positive constant. In addition, several literatures, such as [35–38], have all assumed that the NN approximation errors satisfy (26).

With the aid of Remark 4, we know that $\tilde{A}$ is a stable matrix. Accordingly, there exists a unique positive definite matrix $P \in \mathbb{R}^{mn \times mn}$ satisfying the Lyapunov equation

$$\tilde{A}^{\mathrm{T}} P \tilde{A} - P = -\eta I_{mn} \tag{27}$$

where $\eta > 0$ is a constant.

By using the same technique employed in [22, 27], we can present another assumption as follows:

Assumption 4
There exist two known positive constants τ and $\rho(\tau \leqslant \rho)$ such that

$$\tau I_m \leqslant \tilde{B}^{\mathrm{T}} P \tilde{B} \leqslant \rho I_m \tag{28}$$

Note that this assumption is always true given the specific form of $\tilde{B}$.

Through (25), one might find that the optimal tracking control problem is actually transformed into the problem of designing a robust controller $\upsilon_r(k)$ and adaptive laws of the weights such that the solution of (25) is uniformly convergent. Therefore, in the subsequent section, we focus on designing such a controller and adaptive laws of the weights to solve the problem.

Prior to continuing the discussion, we give another remark here for explaining the principle of designing the controller $\upsilon_c(k)$.

Remark 8
From (15), we know that the controller $\upsilon_c(k)$ is derived by solving a nonlinear equation. In general, it is difficult to make sure whether there exists a solution of such an equation. However, from Theorem 1, we know that there exists the solution $\upsilon_c(k)$ of (13) on Ω ($\upsilon_c(k)$ is unique on Ω'). Hence, from (22) and (23), we shall find that based on the two-layer NN, the problem can be solved well (Figure 1).

A general schematic diagram of the algorithm is developed in Figure 2 ($\upsilon_r(k)$ is to be detailed in the subsequent theorem).

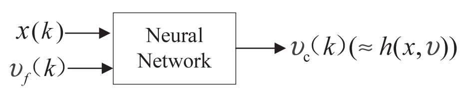

Figure 1. The principle for designing the controller $\upsilon_c(k)$.

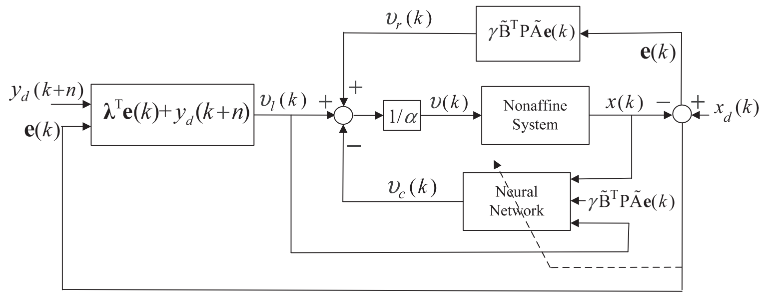

Figure 2. The schematic diagram of the control algorithm for nonaffine nonlinear systems.

DOI: 10.1002/rnc

3.3. Weight updates for guaranteeing tracking performance

In order to guarantee tracking errors to be UAS, the PE assumption [39] is often employed. Nevertheless, it is rather difficult to verify the PE condition of the hidden layer output functions. In the following theorem, a weight-tuning law is developed such that PE assumption is not required, and uniform asymptotic stability of the tracking error can be derived.

Before presenting the main result of this section, we need to provide the following assumption. This assumption is generally employed to derive the stability of closed-loop systems [22–26, 35, 36].

Assumption 5

The output layer weight W is bounded over the compact set Ω by a known positive value W_M, that is, $\|W\| \leqslant W_M$. Moreover, in light of the form of activation functions, there exists a constant $\sigma_M > 0$ such that $\|\sigma(z(k))\| \leqslant \sigma_M$.

Theorem 2

Let Assumptions 1–5 hold. Take the control input for system (1) as in (4) with (7), (23), and the robustifying term

$$\upsilon_r(k) = \gamma \tilde{B}^{\mathrm{T}} P \tilde{A} \boldsymbol{e}(k) \tag{29}$$

where the gain γ is a positive parameter. Let NN weight-tuning law be given by

$$\hat{W}(k+1) = \hat{W}(k) + \theta\phi(z(k))\left(\tilde{W}^{\mathrm{T}}(k)\phi(z(k)) - \upsilon_r(k) - \varepsilon(k)\right)^{\mathrm{T}} \tag{30}$$

with $0 < \theta < 1$. Then, $\tilde{W}$ is bounded, and the tracking error $\boldsymbol{e}(k)$ is UAS, provided that the following condition holds:

$$\frac{\beta_M}{\eta} < \gamma \leqslant \frac{1 - \theta\sigma_M^2}{\rho} \tag{31}$$

Proof

Define the Lyapunov function candidate as

$$L(k) = L_1(k) + L_2(k) \tag{32}$$

where

$$\begin{aligned} L_1(k) &= \boldsymbol{e}^{\mathrm{T}}(k) P \boldsymbol{e}(k), \\ L_2(k) &= \frac{1}{\gamma\theta} \mathrm{tr}\left(\tilde{W}^{\mathrm{T}}(k)\tilde{W}(k)\right) \end{aligned}$$

For convenience, we denote $V(k) = \tilde{W}^{\mathrm{T}}\phi(z(k))$ and $\delta(k) = \upsilon_r(k) + \varepsilon(k)$. From (24) to (29), the first difference of the Lyapunov function is derived as

$$\begin{aligned} \Delta L_1(k) &= \boldsymbol{e}^{\mathrm{T}}(k+1) P \boldsymbol{e}(k+1) - \boldsymbol{e}^{\mathrm{T}}(k) P \boldsymbol{e}(k) \\ &= \left(\tilde{A}\boldsymbol{e}(k) + \tilde{B}(V(k) - \delta(k))\right)^{\mathrm{T}} P \left(\tilde{A}\boldsymbol{e}(k) + \tilde{B}(V(k) - \delta(k))\right) - \boldsymbol{e}^{\mathrm{T}}(k) P \boldsymbol{e}(k) \\ &= (V(k) - \delta(k))^{\mathrm{T}} \left(\tilde{B}^{\mathrm{T}} P \tilde{B}\right)(V(k) - \delta(k)) + 2\boldsymbol{e}^{\mathrm{T}}(k)\tilde{A}^{\mathrm{T}} P \tilde{B}(V(k) - \delta(k)) \\ &\quad + \boldsymbol{e}^{\mathrm{T}}(k)\left(\tilde{A}^{\mathrm{T}} P \tilde{A} - P\right)\boldsymbol{e}(k) \\ &\leqslant \rho(V(k) - \delta(k))^{\mathrm{T}}(V(k) - \delta(k)) + 2\boldsymbol{e}^{\mathrm{T}}(k)\tilde{A}^{\mathrm{T}} P \tilde{B}(V(k) - \delta(k)) - \eta\boldsymbol{e}^{\mathrm{T}}(k)\boldsymbol{e}(k) \end{aligned}$$

DOI: 10.1002/rnc

and

$$\begin{aligned}\Delta L_2(k) &= \frac{1}{\gamma\theta}\mathrm{tr}\left(\tilde{W}^{\mathrm{T}}(k+1)\tilde{W}(k+1) - \tilde{W}^{\mathrm{T}}(k)\tilde{W}(k)\right)\\ &= \frac{1}{\gamma\theta}\mathrm{tr}\left\{\left(\tilde{W}(k) - \theta\phi(z(k))(V(k)-\delta(k))^{\mathrm{T}}\right)^{\mathrm{T}}\left(\tilde{W}(k) - \theta\phi(z(k))(V(k)-\delta(k))^{\mathrm{T}}\right)\right.\\ &\qquad\left. - \tilde{W}^{\mathrm{T}}(k)\tilde{W}(k)\right\}\\ &= \frac{1}{\gamma}\mathrm{tr}\left[\theta\phi^{\mathrm{T}}(z(k))\phi(z(k))(V(k)-\delta(k))(V(k)-\delta(k))^{\mathrm{T}} - 2V(k)(V(k)-\delta(k))^{\mathrm{T}}\right]\\ &= \frac{\theta}{\gamma}\phi^{\mathrm{T}}(z(k))\phi(z(k))(V(k)-\delta(k))^{\mathrm{T}}(V(k)-\delta(k)) - \frac{2}{\gamma}V^{\mathrm{T}}(k)(V(k)-\delta(k))\end{aligned}$$

Noting $\Delta L(k) = \Delta L_1(k) + \Delta L_2(k)$ and the definition of $\|\cdot\|$ and by using Assumption 5, we obtain that

$$\begin{aligned}\Delta L(k) &\leqslant -\eta \boldsymbol{e}^{\mathrm{T}}(k)\boldsymbol{e}(k) - \frac{2}{\gamma}\left(V^{\mathrm{T}}(k) - \gamma \boldsymbol{e}^{\mathrm{T}}(k)\tilde{A}^{\mathrm{T}}P\tilde{B}\right)(V(k)-\delta(k))\\ &\quad + \left(\rho + \frac{\theta}{\gamma}\phi^{\mathrm{T}}(z(k))\phi(z(k))\right)(V(k)-\delta(k))^{\mathrm{T}}(V(k)-\delta(k))\\ &\leqslant -\eta\|\boldsymbol{e}(k)\|^2 - \frac{2}{\gamma}\left(V^{\mathrm{T}}(k) - \gamma \boldsymbol{e}^{\mathrm{T}}(k)\tilde{A}^{\mathrm{T}}P\tilde{B}\right)(V(k)-\delta(k))\\ &\quad + \left(\rho + \frac{\theta}{\gamma}\sigma_M^2\right)(V(k)-\delta(k))^{\mathrm{T}}(V(k)-\delta(k))\end{aligned}$$

Observing the definition of $\delta(k)$, we derive that

$$\begin{aligned}\Delta L(k) &\leqslant -\eta\|\boldsymbol{e}(k)\|^2 - \frac{2}{\gamma}\left(V(k) - \gamma\tilde{B}^{\mathrm{T}}P\tilde{A}\boldsymbol{e}(k)\right)^{\mathrm{T}}(V(k) - \upsilon_r(k))\\ &\quad + \frac{2}{\gamma}\left(V(k) - \gamma\tilde{B}^{\mathrm{T}}P\tilde{A}\boldsymbol{e}(k)\right)^{\mathrm{T}}\varepsilon(k) + \left(\rho + \frac{\theta}{\gamma}\sigma_M^2\right)\|V(k)-\upsilon_r(k)\|^2\\ &\quad - 2\left(\rho + \frac{\theta}{\gamma}\sigma_M^2\right)(V(k)-\upsilon_r(k))^{\mathrm{T}}\varepsilon(k) + \left(\rho + \frac{\theta}{\gamma}\sigma_M^2\right)\varepsilon^{\mathrm{T}}(k)\varepsilon(k)\end{aligned}$$

From Assumptions 3, (29), and (31) and by utilizing Cauchy–Schwarz Inequality, we obtain that

$$\begin{aligned}\Delta L(k) &\leqslant -\eta\|\boldsymbol{e}(k)\|^2 - \left(\frac{2}{\gamma} - \rho - \frac{\theta}{\gamma}\sigma_M^2\right)\left\|V(k) - \gamma\tilde{B}^{\mathrm{T}}P\tilde{A}\boldsymbol{e}(k)\right\|^2\\ &\quad + 2\left(\frac{1}{\gamma} - \rho - \frac{\theta}{\gamma}\sigma_M^2\right)\varepsilon^{\mathrm{T}}(k)\left(V(k) - \gamma\tilde{B}^{\mathrm{T}}P\tilde{A}\boldsymbol{e}(k)\right)\\ &\quad + \left(\rho + \frac{\theta}{\gamma}\sigma_M^2\right)\varepsilon^{\mathrm{T}}(k)\varepsilon(k)\\ &\leqslant -\eta\|\boldsymbol{e}(k)\|^2 - \frac{1}{\gamma}\left\|V(k) - \gamma\tilde{B}^{\mathrm{T}}P\tilde{A}\boldsymbol{e}(k)\right\|^2 + \frac{1}{\gamma}\varepsilon^{\mathrm{T}}(k)\varepsilon(k)\\ &\leqslant -\left(\eta - \frac{\beta_M}{\gamma}\right)\|\boldsymbol{e}(k)\|^2 - \frac{1}{\gamma}\left\|V(k) - \gamma\tilde{B}^{\mathrm{T}}P\tilde{A}\boldsymbol{e}(k)\right\|^2\\ &\leqslant 0.\end{aligned} \tag{33}$$

Equations (32) and (33) guarantee that both $\boldsymbol{e}(k)$ and $\tilde{W}(k)$ are bounded, for $L(k)$ is nonincreasing. Summing both sides of (33) to infinity, we have that

$$\sum_{k=0}^{\infty}\left(\left(\eta - \frac{\beta_M}{\gamma}\right)\|\boldsymbol{e}(k)\|^2 + \frac{1}{\gamma}\left\|V(k) - \gamma\tilde{B}^{\mathrm{T}}P\tilde{A}\boldsymbol{e}(k)\right\|^2\right) \leqslant L(0) - L(\infty)$$

Therefore, we obtain that

$$\sum_{k=0}^{\infty}\left(\eta-\frac{\beta_M}{\gamma}\right)\|\boldsymbol{e}(k)\|^2 \leqslant L(0)-L(\infty) \tag{34}$$

In view of the boundedness of $L(0)-L(\infty)$, we can derive that the series, which is the left side of (34), is convergent. By using the necessity of convergent series [28], we can obtain that $\lim\limits_{k\to\infty}\left(\eta-\frac{\beta_M}{\gamma}\right)\|\boldsymbol{e}(k)\|^2=0$. That is, $\lim\limits_{k\to\infty}\|\boldsymbol{e}(k)\|=0$. Therefore, $\boldsymbol{e}(k)$ is UAS. □

Remark 9

From (25) and (30), we shall find that the weight-tuning rule can be derived as

$$\hat{W}(k+1)=\hat{W}(k)+\theta\phi(z(k))\left(\boldsymbol{e}(k+1)-\tilde{A}\boldsymbol{e}(k)\right)^{\mathrm{T}}\tilde{B} \tag{35}$$

Because the NN approximation error $\varepsilon(k)$ in (30) is typically unknown, the weight updating law (35) is often used to implement the developed algorithm.

4. NUMERICAL EXAMPLES

The purpose of this section is to verify the theoretical results. Two numerical examples are presented here.

4.1. Example 1

Consider the nonaffine nonlinear DT system [40] described by

$$\begin{aligned} x_1(k+1)&=x_2(k)\\ x_2(k+1)&=\frac{x_1(k)x_2(k)(x_1(k)+2.5)}{1+x_1^2(k)+x_2^2(k)}+u(k)+0.1u^3(k)+d(k)\\ y(k)&=x_1(k)\end{aligned} \tag{36}$$

where $d(k)$ is the bounded external disturbance and has the following form:

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

From (36), we can derive that $\partial F(x(k),u(k))/\partial u(k)=1+0.3u^2(k)$. It is obvious that $\partial F(x(k),u(k))/\partial u(k)\neq 0$. Choose $\Omega=[-1,1]\times[-1,1]$. Because $u(x(k))$ is a continuous function with respect to $x(k)$, we can conclude that $u(x(k))$ is bounded on Ω. Consequently, $\partial F(x(k),u(k))/\partial u(k)$ is also bounded on Ω.

The design parameters are listed as follows. Let $\lambda_1=1$ and $\lambda_2=0.25$ (i.e., $z^2+\lambda_1 z+\lambda_2$ is stable). Meanwhile, we assume that $\alpha=1.25, \gamma=0.1429, \theta=0.5, \eta=1, \beta_M=0.125$, and $\rho=3.15$. Let the tracking objective be $y_d(k)=0.8\sin(0.05k\pi)$ and the initial system state be $x(0)=[0.5\,0.5]^{\mathrm{T}}$. Without loss of generality, the initial weight for the input layer to the hidden layer is selected randomly within an interval of [0, 1] and held constant. The initial weight for the output layer is selected randomly within an interval of [−0.2, 0.2]. Meanwhile, the hidden layer of the two-layer NN has eight nodes, that is, the structure of the two-layer NN is 3–8–1. It is worth emphasizing that the number of neurons required for any particular application is still an open problem. Selecting the proper neurons for NNs is more of an art than a science [41]. In this example, the number of neurons is obtained by computer simulations. We find that selecting eight neurons for the hidden layer can lead to satisfactory simulation results.

In our illustration, we denote $\upsilon(k)=u(x(k))$. The computer simulation results are shown in Figures 3–6. Figure 3 shows the trajectories of $y(k)$ and $y_d(k)$. Figure 4 indicates the tracking errors $e_0(k)$ and $e_1(k)$, which consist of the tracking error vector $\boldsymbol{e}(k)$. Figure 5 is used to illustrate the control input $\upsilon(k)$, and Figure 6 is employed to present the nonlinear function $h(x(k),\upsilon(k))$ and the NN output $\upsilon_c(k)$. By using Figures 3–6, we shall find that the proposed controller can make the system output $y(k)$ track the desired reference $y_d(k)$ rather well. The convergence of the

DOI: 10.1002/rnc

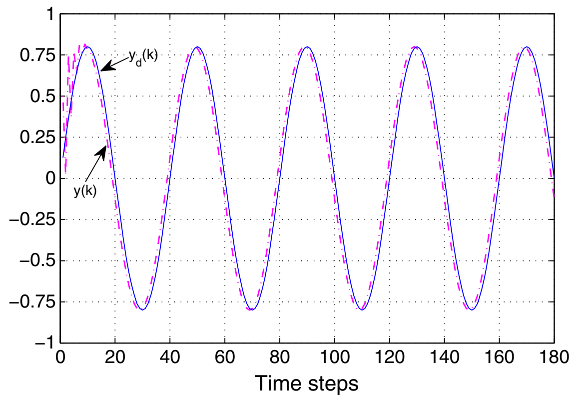

Figure 3. Trajectories of $y(k)$ and $y_d(k)$ in Example 1.

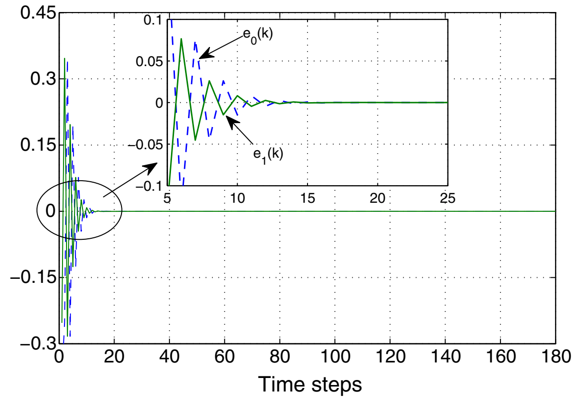

Figure 4. Tracking errors $e_0(k)$ and $e_1(k)$ in Example 1.

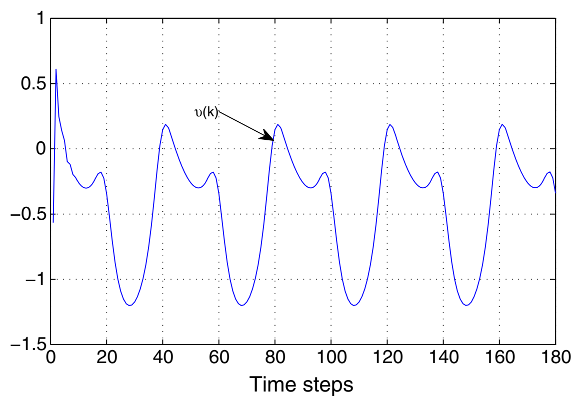

Figure 5. Control input $\upsilon(k)$ in Example 1.

tracking errors is fast, and tracking errors are UAS. Meanwhile, it is also observed that the NN output $\upsilon_c(k)$ can approximate the nonlinear function $h(x(k), \upsilon(k))$ very well. Furthermore, it is significant to note that the number of iterative steps in our example is much less than the method proposed in [40].

DOI: 10.1002/rnc

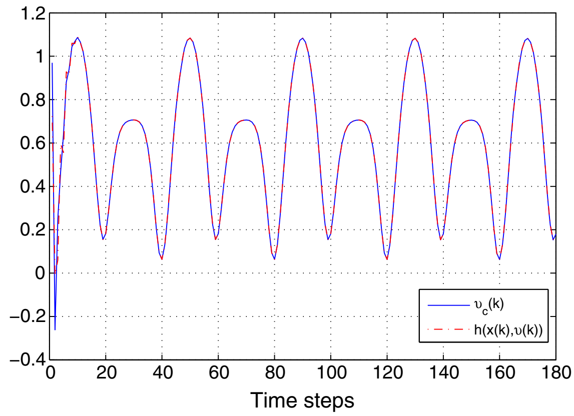

Figure 6. $h(x(k), \upsilon(k))$ and NN output $\upsilon_c(k)$ in Example 1.

4.2. *Example 2*

Consider the MIMO nonaffine nonlinear DT system described by

$$\begin{aligned}
x_{11}(k+1) &= x_{21}(k) \\
x_{12}(k+1) &= x_{22}(k) \\
x_{21}(k+1) &= 0.4x_{22}(k) - 0.1\cos(x_{21}(k)) - 0.15\sin(x_{12}(k)) + 0.2u_1(k) - 0.1\tanh(u_2(k)) \\
x_{22}(k+1) &= 0.1x_{11}(k) - 0.3\sin^2(x_{22}(k))u_1(k) + 0.2u_2(k) \\
y_1(k) &= x_{11}(k) \\
y_2(k) &= x_{12}(k)
\end{aligned} \tag{37}$$

where $x_1(k) = [x_{11}(k)\, x_{12}(k)]^{\mathrm{T}}, x_2(k) = [x_{21}(k)\, x_{22}(k)]^{\mathrm{T}}, u(k) = [u_1(k)\, u_2(k)]^{\mathrm{T}}$, and $y(k) = [y_1(k)\, y_2(k)]^{\mathrm{T}}$.

The control objective is to control the system output $y(k)$ to track the prescribed trajectory

$$y_d(k) = [2.6\sin(k\pi/200)\; 3\cos(k\pi/180)]^{\mathrm{T}}$$

Choose $\Omega = [-3, 3] \times [-3, 3]$. By using (37), we know that $\partial F(x(k), u(k))/\partial u(k)$ is a positive definite matrix. Meanwhile, it is easy to derive that

$$0.01I_2 \leqslant \partial F(x(k), u(k))/\partial u(k) \leqslant 0.04I_2.$$

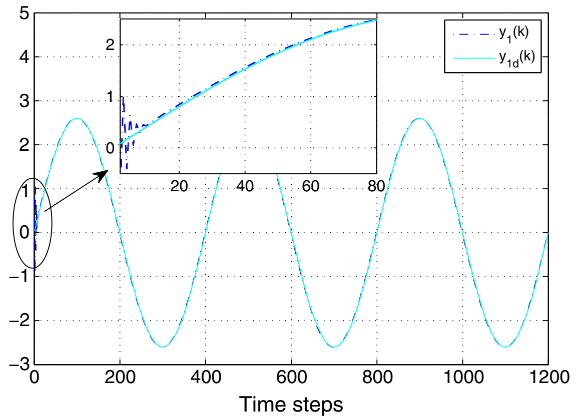

Figure 7. Trajectories of $y_1(k)$ and $y_{1d}(k)$ in Example 2.

DOI: 10.1002/rnc

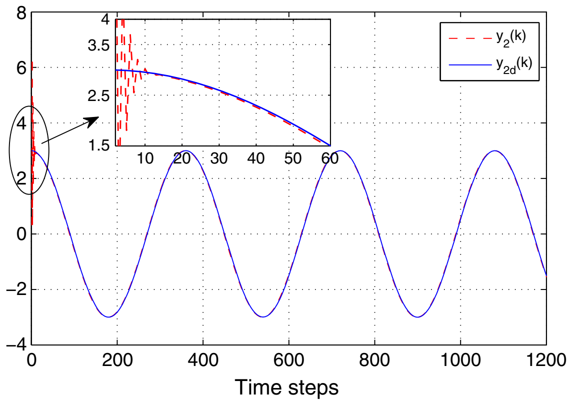

Figure 8. Trajectories of $y_2(k)$ and $y_{2d}(k)$ in Example 2.

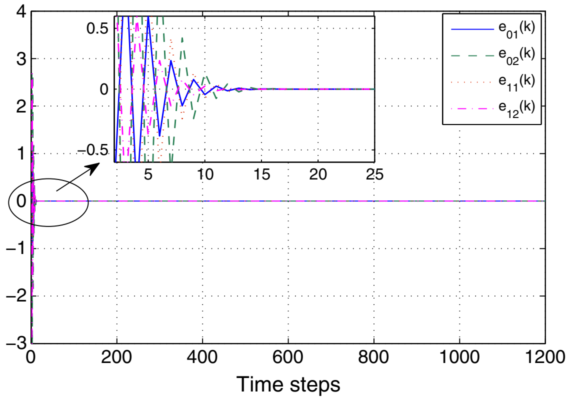

Figure 9. Tracking errors $e_{ij}(k)(i = 0, 1; j = 1, 2)$ in Example 2.

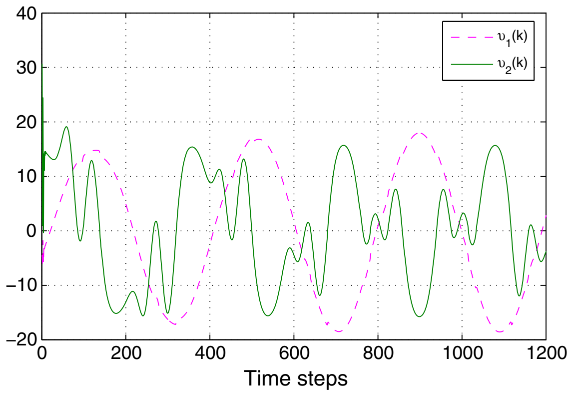

Figure 10. Control input $\upsilon(k)$ in Example 2.

The design parameters are selected as follows. Let $\lambda_1 = 1$ and $\lambda_2 = 0.25$ (i.e., $z^2 + \lambda_1 z + \lambda_2$ is stable). Meanwhile, we choose that $\alpha = 2, \gamma = 0.1429, \theta = 0.5, \eta = 1, \beta_M = 0.125$, and $\rho = 3.15$. The initial system state is selected to be $x_0 = [0.95\,0.31\,0.95\,0.31]^{\mathrm{T}}$. Without loss of generality, the method of selecting initial weights is the same as Example 1. The hidden layer of the NN has 28 nodes, that is, the structure of the two-layer NN is 6–28–2. It should be mentioned that the number of neurons is derived by computer simulations. In this example, we find that selecting 28 neurons for the hidden layer can lead to satisfactory simulation results.

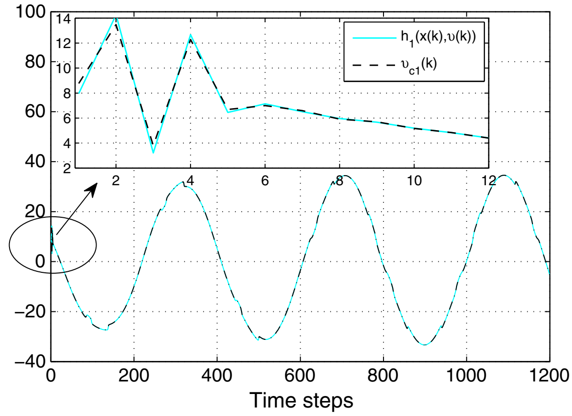

Figure 11. $h_1(x(k), \upsilon(k))$ and NN output $\upsilon_{c1}(k)$ in Example 2.

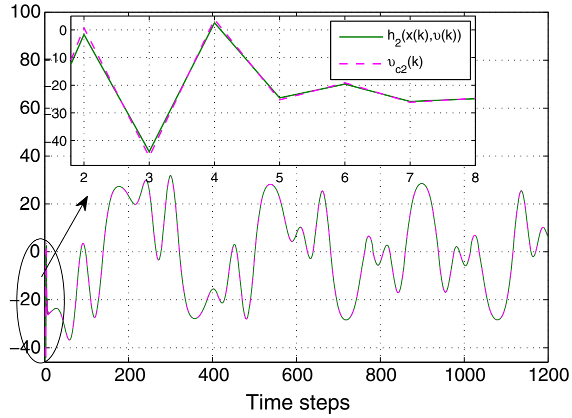

Figure 12. $h_2(x(k), \upsilon(k))$ and NN output $\upsilon_{c2}(k)$ in Example 2.

In our illustration, we denote $\upsilon(k) = u(x(k))$ and $\upsilon(k) = [\upsilon_1(k)\, \upsilon_2(k)]^{\mathrm{T}}$. The computer simulation results are shown in Figures 7–12. Figures 7 and 8 indicate the trajectories of $y_1(k)$, $y_{1d}(k)$ and $y_2(k)$, $y_{2d}(k)$, respectively. Figure 9 shows the tracking errors $e_{ij}(k)(i = 0, 1; j = 1, 2)$, which consist of the tracking error vector $\boldsymbol{e}(k)$. As a matter of fact, the tracking error vector $\boldsymbol{e}(k)$ is composed of $\boldsymbol{e}_i(k)(i = 0, 1)$, and each $\boldsymbol{e}_i(k)$ contains subelements $e_{i1}(k)$ and $e_{i2}(k)$. Figure 10 is employed to present the control input $\upsilon_i(k)(i = 1, 2)$, which consists of the control vector $\upsilon(k)$. Let $h_i(x(k), \upsilon(k))(i = 1, 2)$ be the subelements of the vector-valued nonlinear function $h(x(k), \upsilon(k))$, that is, $h(x(k), \upsilon(k)) = [h_1(x(k), \upsilon(k))\, h_2(x(k), \upsilon(k))]^{\mathrm{T}}$, and $\upsilon_{ci}(k)(i = 1, 2)$ be the subelements of the NN output $\upsilon_c(k)$, that is, $\upsilon_c(k) = [\upsilon_{c1}(k)\, \upsilon_{c2}(k)]^{\mathrm{T}}$. Figures 11 and 12 are used to show $h_1(x(k), \upsilon(k))$, $\upsilon_{c1}(k)$, and $h_2(x(k), \upsilon(k))$, $\upsilon_{c2}(k)$, respectively. From the aforementioned simulation results (Figures 7–12), it is observed that the system output $y(k)$ can track the desired trajectory $y_d(k)$ very well. Meanwhile, the tracking errors are UAS, and the convergence of the tracking error is very fast. Moreover, it is also observed that the NN output $\upsilon_c(k)$ can approximate the nonlinear function $h(x(k), \upsilon(k))$ rather well.

5. CONCLUSION

In this paper, an NN-based direct adaptive control has been investigated for a class of unknown MIMO nonaffine nonlinear DT systems. In order to utilize feedback linearization methods, the controller is divided into three parts: the first part is to stabilize linearized dynamics, the second part

DOI: 10.1002/rnc

is to cancel the nonlinearity of unknown nonlinear DT plants through the design of NNs, and the third part is the robustness term. By using Implicit Function Theorem, the output of two-layer NN is guaranteed to approximate the unknown nonlinear function very well. Pretraining is not required here, and the weights of the NNs used in adaptive control are directly updated online. By designing such a controller and without the PE assumption, the tracking error of the output of the closed-loop system is guaranteed to be UAS based on Lyapunov's method. In our future work, we shall focus on how to design an optimal controller for nonaffine nonlinear DT systems.

ACKNOWLEDGEMENTS

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.

REFERENCES

1. Werbos PJ. *The Roots of Backpropagation: From Ordered Derivatives to Neural Networks and Political Forecasting*. Wiley: New York, 1994.
2. Chen FC, Khalil HK. Adaptive control of a class of nonlinear discrete-time systems using neural networks. *IEEE Transactions on Automatic Control* 1995; **40**(5):791–801.
3. Wang D, Lan W. Adaptive neural control for a class of uncertain nonlinear systems with unknown time delay. *International Journal of Robust and Nonlinear Control* 2009; **19**:807–821.
4. Wang H, Chen B, Lin C. Adaptive neural tracking control for a class of stochastic nonlinear systems. *International Journal of Robust and Nonlinear Control* 2014; **24**(7):1262–1280.
5. Yang Q, Vance JB, Jagannathan S. Control of nonaffine nonlinear discrete-time systems using reinforcement-learning-based linearly parameterized neural networks. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics* 2008; **38**(4):994–1001.
6. Deng H, Li HX, Wu YH. Feedback-linearization-based neural adaptive control for unknown nonaffine nonlinear discrete-time systems. *IEEE Transactions on Neural Networks* 2008; **19**(9):1615–1625.
7. Yu Z, Li S, Du H. Razumikhin Nussbaum-lemma-based adaptive neural control for uncertain stochastic pure-feedback nonlinear systems with time-varying delays. *International Journal of Robust and Nonlinear Control* 2013; **23**(11):1214–1239.
8. Sun T, Pei H, Pan Y, Zhang C. Robust adaptive neural network control for environmental boundary tracking by mobile robots. *International Journal of Robust and Nonlinear Control* 2013; **23**:123–136.
9. Liu D, Zhang Y, Zhang H. A self-learning call admission control scheme for CDMA cellular networks. *IEEE Transactions on Neural Networks* 2005; **16**(5):1219–1228.
10. Liu D, Javaherian H, Kovalenko O, Huang T. Adaptive critic learning techniques for engine torque and air-fuel ratio control. *IEEE Transactions on Systems, Man, and Cybernetics Part B: Cybernetics* 2008; **38**(4):988–993.
11. Wang FY, Zhang H, Liu D. Adaptive dynamic programming: an introduction. *IEEE Computational Intelligence Magazine* 2009; **4**(2):39–47.
12. Liu D, Wang D, Zhao D, Wei Q, Jing N. Neural-network-based optimal control for a class of unknown discrete-time nonlinear systems using globalized dual heuristic programming. *IEEE Transactions on Automation Science and Engineering* 2012; **9**(3):628–634.
13. Liu D, Wei Q. Finite-approximation-error-based optimal control approach for discrete-time nonlinear systems. *IEEE Transactions on Cybernetics* 2013; **43**(2):779–789.
14. Jiang Y, Jiang ZP. Robust adaptive dynamic programming for large-scale systems with an application to multimachine power systems. *IEEE Transactions on Circuits and Systems II: Express Briefs* 2012; **59**(10):693–697.
15. Jiang Y, Jiang ZP. Computational adaptive optimal control for continuous-time linear systems with completely unknown dynamics. *Automatica* 2012; **48**(10):2699–2704.
16. Hornic K. Stinchombe M. Multilayer feedforward neural networks are universal approximators. *Neural Networks* 1989; **2**(5):359–366.
17. Ge SS, Hang CC, Zhang T. Adaptive neural network control of nonlinear systems by state and output feedback. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics* 1999; **29**(6):818–828.
18. Ge SS, Zhang J. Neural-network control of nonaffine nonlinear systems with zero dynamics by state and output feedback. *IEEE Transactions on Neural Networks* 2003; **14**(4):900–918.
19. Yang B, Calise AJ. Adaptive control of a class of nonaffine systems using neural networks. *IEEE Transactions on Neural Networks* 2007; **18**(4):1149–1159.
20. Hovakimyan N, Nardi F, Calise A, Kim N. Adaptive output feedback control of uncertain nonlinear systems using single-hidden-layer neural networks. *IEEE Transactions on Neural Networks* 2002; **13**(6):1420–1431.

DOI: 10.1002/rnc

21. Park JH, Huh SH, Kim SH, Seo SJ, Park GT. Direct adaptive controller for nonaffine nonlinear systems using self-structuring neural networks. *IEEE Transactions on Neural Networks* 2005; **16**(2):414–422.
22. Lewis FL, Jagannathan S, Yesildirek A. *Neural Network Control of Robot Manipulators and Nonlinear Systems*. Taylor & Francis: London, 1999.
23. Jagannathan S, Lewis FL. Discrete-time neural net controller for a class of nonlinear dynamical systems. *IEEE Transactions on Automatic Control* 1996; **41**(11):1693–1699.
24. He P, Jagannathan S. Reinforcement learning-based output feedback control of nonlinear systems with input constraints. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics* 2005; **35**(1):150–154.
25. Yang L, Si J, Tsakalis KS, Rodriguez AA. Direct heuristic dynamic programming for nonlinear tracking control with filtered tracking error. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics* 2009; **19**(6): 1617–1622.
26. Yang Q, Jagannathan S. Reinforcement learning controller design for affine nonlinear discrete-time systems using approximators. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics* 2012; **42**(2):377–390.
27. Khalil H. *Nonlinear Systems* (3rd edn). Prentice-Hall: NJ, 2002.
28. Apostol TM. *Mathematical Analysis* (2nd edn). Addison-Wesley: MA, 1974.
29. Ge SS, Hang CC, Lee TH, Zhang T. *Stable Adaptive Neural Network Control*. Kluwer: Boston, MA, 2002.
30. Rudin W. *Functional Analysis* (3rd edn). McGraw-Hill: Singapore, 1991.
31. Zeidler E. *Nonlinear Functional Analysis and its Applications: Part 1: Fixed-Point Theorems*. Springer-Verlag: New York, 1985.
32. Haykin S. *Neural Networks and Learning Machines*. Prentice Hall: New Jersey, 2008.
33. Igelnik B, Pao YH. Stochastic choice of basis functions in adaptive function approximation and the function-link net. *IEEE Transactions on Neural Networks* 1995; **6**(6):1320–1329.
34. Hayakawa T, Haddad WM, Hovakimyan N. Neural network adaptive control for nonlinear uncertain dynamical systems with asymptotic stability guarantees. *IEEE Transactions on Neural Networks* 2008; **19**(1):80–89.
35. Dierks T, Thumati BT, Jagannathan S. Optimal control of unknown affine nonlinear discrete-time systems using offline-trained neural networks with proof of convergence. *Neural Networks* 2009; **22**:851–860.
36. Zhang H, Cui L, Zhang X, Luo Y. Data-driven robust approximate optimal tracking control for unknown general nonlinear systems using adaptive dynamic programming method. *IEEE Transactions on Neural Networks* 2011; **22**(12):2226–2236.
37. Zhang X, Zhang H, Sun Q, Luo Y. Adaptive dynamic programming-based optimal control of unknown nonaffine nonlinear discrete-time systems with proof of convergence. *Neurocomputing* 2012; **91**:48–55.
38. Liu D, Wang D, Yang X. An iterative adaptive dynamic programming algorithm for optimal control of unknown discrete-time nonlinear systems with constrained inputs. *Information Sciences* 2013; **220**:331–342.
39. Sadegh N. A perceptron network for functional identification and control of nonlinear systems. *IEEE Transactions on Neural Networks* 1993; **4**(6):982–988.
40. Zhang J, Ge SS, Lee TH. Direct RBF neural network control of a class of discrete-time nonaffine nonlinear systems. *Proceedings of the American Control Conference*, Anchorage, AK, May 2002; 424–429.
41. Padhi R, Unnikrishnan N, Wang X, Balakrishnan SN. A single network adaptive critic (SNAC) architecture for optimal control synthesis for a class of nonlinear systems. *Neural Networks* 2006; **19**:1648–1660.

DOI: 10.1002/rnc