01 Dec 2008

Neural-Network-Based State Feedback Control of a Nonlinear Discrete-Time System in Nonstrict Feedback Form

Pingan He

Jagannathan Sarangapani
Missouri University of Science and Technology, sarangap@mst.edu

Recommended Citation

P. He and J. Sarangapani, "Neural-Network-Based State Feedback Control of a Nonlinear Discrete-Time System in Nonstrict Feedback Form," *IEEE Transactions on Neural Networks*, Institute of Electrical and Electronics Engineers (IEEE), Dec 2008.
The definitive version is available at https://doi.org/10.1109/TNN.2008.2003295

Neural-Network-Based State Feedback Control of a Nonlinear Discrete-Time System in Nonstrict Feedback Form

Sarangapani Jagannathan, *Senior Member, IEEE*, and Pingan He

***Abstract*—In this paper, a suite of adaptive neural network (NN) controllers is designed to deliver a desired tracking performance for the control of an unknown, second-order, nonlinear discrete-time system expressed in nonstrict feedback form. In the first approach, two feedforward NNs are employed in the controller with tracking error as the feedback variable whereas in the *adaptive critic* NN architecture, three feedforward NNs are used. In the adaptive critic architecture, two action NNs produce virtual and actual control inputs, respectively, whereas the third critic NN approximates certain *strategic* utility function and its output is employed for tuning action NN weights in order to attain the near-optimal control action. Both the NN control methods present a well-defined controller design and the noncausal problem in discrete-time backstepping design is avoided via NN approximation. A comparison between the controller methodologies is highlighted. The stability analysis of the closed-loop control schemes is demonstrated. The NN controller schemes do not require an offline learning phase and the NN weights can be initialized at zero or random. Results show that the performance of the proposed controller schemes is highly satisfactory while meeting the closed-loop stability.**

***Index Terms*—Adaptive critic control, near-optimal control, neural network (NN) control, nonstrict feedback system.**

I. Introduction

THE adaptive backstepping control methodology [1], [2] has been utilized to improve the performance of complex nonlinear systems. When used under some mild assumptions, many existing adaptive control techniques can be extended to a general class of nonlinear systems. A drawback with the conventional adaptive backstepping approach is that the system under consideration must be expressed as linear in the unknown parameters (LIP) and the dynamics of the nonlinear system must be known beforehand.

The backstepping methodology using NNs on the other hand is a potential solution to control a larger class of nonlinear systems since the NNs are nonlinear in the tunable parameters. By using NNs in each stage of the backstepping to estimate certain nonlinear functions, a more suitable control law can be designed without the LIP assumption and the need for the system dynamics. The application of adaptive NN control of nonlinear systems in both continuous and discrete time is dealt with in several works [3]–[10].

Adaptive NN backstepping control has been extended to strict feedback nonlinear systems in continuous time given in the following form [6]: $\dot{x}_i = f_i(\overline{x}_i) + g_i(\overline{x}_i)x_{i+1}, 1 \leq i \leq n-1$, and $\dot{x}_n = f_n(\overline{x}_n) + g_n(\overline{x}_n)u, 2 \leq n$, where $\overline{x}_i = [x_1, \ldots, x_i]^T \in R^i, i = 1, \ldots, n$, and $u \in R$ are state variables and system input, respectively. On the other hand, the backstepping-based neural network (NN) control design in discrete time is far more complex than continuous time due primarily to the fact that discrete-time Lyapunov derivatives are quadratic in the state, not linear as in the continuous case. Additionally, the design has to overcome a causal control problem. In [7], a multilayer neural networks backstepping controller is proposed for discrete-time feedback system, where $f_i(\overline{x}_i), i = 1, \ldots, n$, are considered unknown smooth functions whereas $g_i(\overline{x}_i), i = 1, \ldots, n$, are assumed as unknown constants. By contrast, in [6], both $f_i(\overline{x}_i), i = 1, \ldots, n$, and $g_i(\overline{x}_i), i = 1, \ldots, n$, are considered unknown smooth functions in discrete time. In all the above controller design methods [3]–[7], tracking error is used as the only performance measure to tune the NN weights online. Nevertheless, tracking-error-based state feedback control schemes are not available for a nonstrict feedback nonlinear discrete-time system where the system nonlinearities are functions of all the state variables.

On the other hand, adaptive critic NN control methods [8]–[10], [15], [18] often use backpropagation-based NN training offline and a utility function to meet certain complex performance criterion. The adaptive critic family of NN control [9], [10], [15]–[19] is a promising methodology to handle complex optimal control problems. In the adaptive critic NN control, the critic conveys much less information than the desired output required in supervisory learning. Nevertheless, their ability to generate correct control actions makes adaptive critics prime candidates for controlling complex nonlinear systems. However, an adaptive critic-based NN control scheme using state variable feedback is not available for a nonstrict feedback nonlinear discrete-time system.

Despite these developments in NN control, an adaptive-critic-based NN control scheme with an online reinforcement learning capability is preferred over offline training

Manuscript received April 13, 2007; revised February 14, 2008; accepted April 15, 2008. First published November 17, 2008; current version published November 28, 2008. This work was supported by the National Science Foundation under Grants ECCS #0327877 and ECCS #0621924.

S. Jagannathan is with the Department of Electrical and Computer Engineering, Missouri University of Science and Technology, Rolla, MO 65409 USA (e-mail: sarangap@mst.edu).

P. He is with GM Power Train Group, Troy, MI 48007 USA.

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TNN.2008.2003295

due to unavailability of *a priori* training data for approximating complex nonlinear functions. However, reinforcement-learning-based adaptive critic NN controller is far more complex than a traditional online-learning-based NN controller in terms of computational complexity, even though the former can optimize the controller performance. Therefore, in this paper, both tracking error and reinforcement-learning-based adaptive critic NN controller designs with an online learning feature are developed for a nonstrict feedback nonlinear discrete-time system of second order.

In the first tracking-error-based approach, two feedforward NNs with an online learning feature are used to approximate the dynamics of the nonlinear discrete-time system. In the second adaptive critic NN control architecture, two action-generating NNs with feedforward architecture approximate the dynamics of the nonlinear system and their weights are tuned using a third critic NN output. In this work, the single critic NN not only approximates a certain long term utility function but also tunes the weights of two action-generating NNs, which is in contrast with the available works in the literature where a single critic is normally used to tune the weights of an action-generating NN [9], [10]. Additionally, the closed-loop performance is demonstrated by using the Lyapunov-based analysis and novel NN weight updates in both controller methodologies. The NN weights are tuned online with no preliminary offline learning phase. Finally, a comparison between the two control methods in terms of their online learning and computational complexities are highlighted and their performance is contrasted in the simulation section.

In summary, the proposed work overcomes several deficiencies of the previous works, such as: 1) the control scheme is applicable to a nonstrict feedback nonlinear system in discrete time, 2) noncausal problem in the discrete-time backstepping design is avoided via the NN approximation, 3) the need for signs of unknown nonlinear functions $g_i(\overline{x}_i), i = 1, \ldots, n$, is relaxed in the controller design, 4) a robustifying term to overcome the persistency of excitation condition is not used in the weight updates [6], 5) a well-defined controller is developed because a single NN is utilized to compensate two nonlinearities, and 6) both online tracking error and reinforcement-based adaptive critic NN controllers are proposed.

This paper is organized as follows. Section II discusses background on neural networks and uniformly ultimately bounded (UUB) definition. The proposed tracking-error-based NN controller is presented in Section III. Section IV introduces an adaptive critic NN controller and provides a comparison with the tracking-error-based NN controller. Section V details the simulation results whereas Section VI carries the conclusions of the work.

II. Background

The following background is required for the development of the adaptive NN controller. First, the NN approximation property is introduced. Second, the definition of UUB is given. Then, the nonstrict nonlinear system description is described.

A. Stability of Systems

Consider the nonlinear system given by

$$x(k+1) = f(x(k), u(k)) \tag{1}$$
$$y(k) = h(x(k)) \tag{2}$$

where $x(k)$ is the state vector, $u(k)$ is the input vector, and $y(k)$ is the output vector. The solution is said to be *uniformly ultimately bounded* if for all $x(k_0) = x_0$ and a $\mu \geq 0$, there exists a number $N(\mu, x_0)$ such that $\|x(k)\| \leq \mu$ for all $k \geq k_0 + N$.

B. Discrete-Time Nonlinear System in Nonstrict Feedback Form

Consider the following second-order nonstrict feedback nonlinear system described by:

$$\begin{aligned} x_1(k+1) &= f_1(x_1(k), x_2(k)) \\ &\quad + g_1(x_1(k), x_2(k))\, x_2(k) + d_1'(k) \\ x_2(k+1) &= f_2(x_1(k), x_2(k)) \\ &\quad + g_2(x_1(k), x_2(k))\, u(k) + d_2'(k) \end{aligned} \tag{3}$$

where $x_i(k) \in R, i = 1, 2$, are states, $u(k) \in R$ is the system input, and $d_1'(k) \in R$ and $d_2'(k) \in R$ are unknown but bounded disturbances.

Equation (3) represents a discrete-time nonlinear system in nonstrict feedback form, since $f_1(\cdot)$ and $g_1(\cdot)$ are the functions of both $x_1(k)$ and $x_2(k)$, unlike in the case of a strict feedback nonlinear system, where $f_1(\cdot)$ and $g_1(\cdot)$ are only a function of state $x_1(k)$.

For simplicity, let us denote $f_i(k)$ for $f_i(x_1(k), x_2(k))$ and $g_i(k)$ for $g_i(x_1(k), x_2(k)), \forall i = 1, 2$, where $f_i(k)$ and $g_i(k)$ are smooth functions, which are considered unknown. The system under consideration can be written as

$$\begin{aligned} x_1(k+1) &= f_1(k) + g_1(k) x_2(k) + d_1'(k) \\ x_2(k+1) &= f_2(k) + g_2(k) u(k) + d_2'(k). \end{aligned} \tag{4}$$

Our objective is to design an NN controller using state feedback for system (4) such that: 1) all the signals in the closed-loop remain UUB, and 2) the state $x_1(k)$ follows a desired trajectory $x_{1d}(k)$.

III. Adaptive Tracking Error Controller Design

First, a tracking-error-based NN controller approach with online training of NN weights is introduced by assuming that the states of the system are available for measurement. Then, in the next section, an adaptive-critic-NN-based control design with reinforcement learning scheme is introduced. Lyapunov-based analysis is presented. To proceed, the following mild assumptions are required.

A. Controller Design

Assumption 1: The desired trajectory $x_{1d}(k)$ and its future values are known and bounded over the compact $S \subset \Re$.

Assumption 2: The unknown smooth functions $g_1(\cdot)$ and $g_2(\cdot)$ are assumed to be bounded away from zero within the

compact set S, i.e., $0 < g_{1\min} < |g_1(\cdot)| < g_{1\max}$, $0 < g_{2c\min} < |g_{2c}(\cdot)| < g_{2\max}$, and $0 < g_{2\min} < |g_2(\cdot)| < g_{2\max}$, $\forall x_1(k)$ and $x_2(k) \in S$, where $g_{1\min} \in \Re^+$, $g_{1\max} \in \Re^+$, $g_{2\min} \in \Re^+$, and $g_{2\max} \in \Re^+$. Without the loss generality, we will assume $g_1(\cdot)$, $g_2(\cdot)$, and $g_{2c}(\cdot)$ (to be defined later) are positive in this paper. Next, the adaptive backstepping NN control design is introduced.

Step 1 (Virtual Controller Design): Define the tracking error as

$$e_1(k) = x_1(k) - x_{1d}(k). \tag{5}$$

Equation (5) can be rewritten after the substitution of system dynamics from (4) as

$$\begin{aligned} e_1(k+1) &= x_1(k+1) - x_{1d}(k+1) \\ &= f_1(k) + g_1(k)x_2(k) - x_{1d}(k+1) + d_1'(k). \end{aligned} \tag{6}$$

By viewing $x_2(k)$ as a virtual control input, a desired feedback control signal can be designed as

$$x_{2d}(k) = \frac{1}{g_1(k)}\left(-f_1(k) + x_{1d}(k+1)\right) + k_1 e_1(k) \tag{7}$$

where k_1 is a design constant selected, such that the tracking error $e_1(k)$ is asymptotically stable. Assumption 1 ensures that $g_1(k)$ is bounded away from zero.

Because $f_1(k)$ and $g_1(k)$ are unknown smooth functions, the desired feedback control $x_{2d}(k)$ cannot be implemented in practice. From (7), it can be seen that the unknown part $(1/g_1(k))(-f_1(k) + x_{1d}(k+1))$ is a smooth function of $x_1(k)$, $x_2(k)$, and $x_{1d}(k+1)$. By utilizing NN to approximate this unknown part consisting of the ratio of two unknown smooth functions, $x_{2d}(k)$ can be expressed as [6]

$$\begin{aligned} x_{2d}(k) &= w_1^T(k)\phi\left(v_1^T z_1(k)\right) + \varepsilon_1(k) + k_1 e_1(k) \\ &= w_1^T(k)\phi(k) + \varepsilon_1(k) + k_1 e_1(k) \end{aligned} \tag{8}$$

where $w_1(k) \in R^{n_1 \times 1}$ denotes the constant target weights of the output layer, $v_1 \in R^{3\times n_1}$ is the weights of the hidden layer, n_1 is the nodes number of hidden layer, $\phi(k)$ is the hidden layer activation function, $\varepsilon_1(k)$ is the approximation error, $k_1 \in R$ is the design constant, and the NN input is taken as $z_1(k) = [x_1(k), x_2(k), x_{1d}(k+1)]^T$. Only the hidden-layer NN weights are updated, whereas the input-layer weights are selected initially at random and held constant so that hidden-layer activation function vector forms a basis [11].

Consequently, the virtual control is given as

$$\hat{x}_{2d}(k) = \hat{w}_1^T(k)\phi(k) + k_1 e_1(k) \tag{9}$$

where $\hat{w}_1(k) \in R^{n_1 \times 1}$ is the actual NN weight. Define the error in weights during estimation by

$$\tilde{w}_1(k) = \hat{w}_1(k) - w_1(k). \tag{10}$$

Define the error between $x_2(k)$ and $\hat{x}_{2d}(k)$ as

$$e_2(k) = x_2(k) - \hat{x}_{2d}(k). \tag{11}$$

Equation (6) can be expressed using (11) for $x_2(k)$ as

$$e_1(k+1) = f_1(k) + g_1(k)\left(e_2(k) + \hat{x}_{2d}(k)\right) - x_{1d}(k+1) + d_1'(k) \tag{12}$$

or, equivalently

$$e_1(k+1) = g_1(k)\left(k_1 e_1(k) + e_2(k) + \zeta_1(k) + d_1(k)\right) \tag{13}$$

where

$$\zeta_1(k) = \tilde{w}_1^T(k)\phi(k) \tag{14}$$

and

$$d_1(k) = \frac{d_1'(k)}{g_1(k)} - \varepsilon_1(k). \tag{15}$$

Note that $d_1(k)$ is bounded given the fact that $\varepsilon_1(k)$, $d_1'(k)$, and $g_1(k)$ are all bounded.

Step 2 (Design of the Control Input $u(k)$): Write the error $e_2(k)$ from (11) as

$$\begin{aligned} e_2(k+1) &= x_2(k+1) - \hat{x}_{2d}(k+1) \\ &= f_2(k) + g_2(k)u(k) - \hat{x}_{2d}(k+1) + d_2'(k) \end{aligned} \tag{16}$$

where $\hat{x}_{2d}(k+1)$ is the future value of $\hat{x}_{2d}(k)$. Here this problem is solved by using a semirecurrent NN because it can be used as a one-step predictor. The term $\hat{x}_{2d}(k+1)$ depends on state $x(k)$, virtual control input $\hat{x}_{2d}(k)$, and desired trajectory $x_{1d}(k+2)$. By taking the independent variables as the NN inputs, $\hat{x}_{2d}(k+1)$ can be approximated during control input selection. From (11), $\hat{x}_{2d}(k+1)$ can be obtained as a nonlinear function of $z_2(k)$, where $z_2(k) = [x_1(k), x_2(k), \hat{x}_{2d}(k), x_{1d}(k)]^T$, i.e., $\hat{x}_{2d}(k+1) = f(z_2(k))$, where $f : R^{4\times 1} \to R$ is a nonlinear mapping. Consequently, it can be approximated by the NN. Alternatively, the value of $\hat{x}_{2d}(k+1)$ can also be obtained by employing a filter [5]. In this paper, a feedforward NN with properly chosen weight tuning law rendering a semirecurrent or dynamic NN will be used to predict the future value. The first layer of the second NN using the system estimates, past value of $\hat{x}_{2d}(k)$ along with the desired value of the first state as inputs to an NN, generates $\hat{x}_{2d}(k+1)$, which in turn is used by the second layer to generate a suitable control input. On the other hand, one can use a single-layer dynamic NN to generate the future value of $\hat{x}_{2d}(k)$, which can be utilized as an input to a third control NN to generate a suitable control input. Here, these two single-layer NNs are combined into a single two-layer NN with semirecurrent architecture.

Choose the desired control input and use the second NN to approximate the unknown dynamics as

$$\begin{aligned} u_d(k) &= \frac{1}{g_2(k)}\left(-f_2(k) + \hat{x}_{2d}(k+1)\right) + k_2 e_2(k) + k_1 e_1(k) \\ &= w_2^T(k)\sigma\left(v_2^T z_2(k)\right) + \varepsilon_2(k) + k_2 e_2(k) + k_1 e_1(k) \\ &= w_2^T(k)\sigma(k) + \varepsilon_2(k) + k_2 e_2(k) + k_1 e_1(k) \end{aligned} \tag{17}$$

where $w_2(k) \in R^{n_2 \times 1}$ is the matrix of target weights of the output layer, $v_2 \in R^{4\times n_2}$ is the weights of the hidden layer, n_2 is the nodes number of hidden layer, $\sigma(k)$ is the vector of activation functions, $\varepsilon_2(k)$ is the approximation error,$k_2 \in R$ is the design constant, and the NN input is selected as $z_2(k)$. Here,

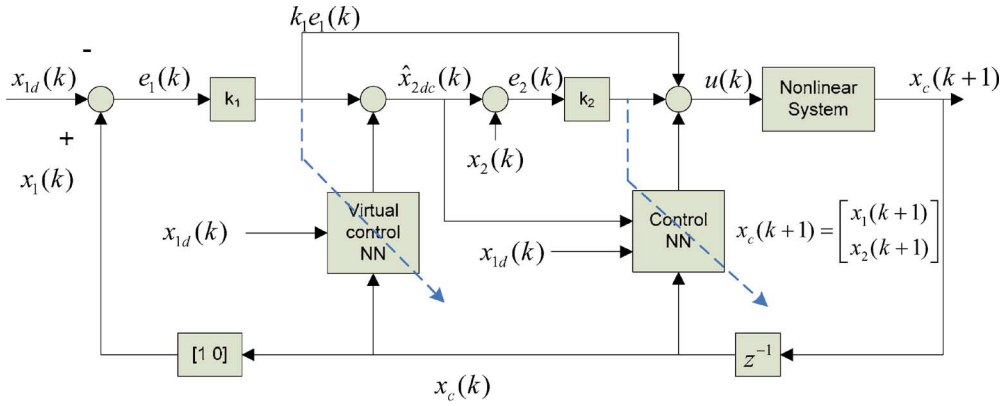

Fig. 1. Tracking error adaptive NN controller structure.

Assumption 2 ensures that the function $g_2(k)$ is bounded away from zero. The actual control input is selected as

$$u(k) = \hat{w}_2^T(k)\sigma(k) + k_2e_2(k) + k_1e_1(k) \tag{18}$$

where $\hat{w}_2(k) \in R^{n_2\times 1}$ is the actual weights for the second NN.

Substituting (17) and (18) into (16) yields

$$e_2(k+1) = g_2(k)\left(k_1e_1(k) + k_2e_2(k) + \zeta_2(k) + d_2(k)\right) \tag{19}$$

where

$$\zeta_2(k) = \tilde{w}_2^T(k)\sigma(k) \tag{20}$$

and

$$d_2(k) = \frac{d_2'(k)}{g_2(k)} - \varepsilon_2(k). \tag{21}$$

Equations (13) and (19) represent the closed-loop error dynamics. The next step is to design the NN weight tuning scheme such that the closed-loop system stability can be inferred.

B. Weight Updates

In tracking-error-based NN controller, two NNs are employed to approximate the nonlinear dynamics and their weights are tuned online using tracking errors. It is required to show the errors $e_1(k)$ and $e_2(k)$ and the NN weights $\hat{w}_1(k)$ and $\hat{w}_2(k)$ are bounded. To accomplish this, first, an assumption to define the bounds on the target weights and activation functions are presented. Second, a discrete-time online weight tuning algorithms is introduced so that closed-loop stability is inferred.

Assumption 3: Both the target weights and the activation functions for all NNs are bounded by known positive values so that

$$\begin{aligned} &\|w_1(k)\| \le w_{1\max} \quad \|w_2(k)\| \le w_{2\max} \\ &\|\phi(\cdot)\| \le \phi_{\max} \quad \text{and} \quad \|\sigma(\cdot)\| \le \sigma_{\max}. \end{aligned} \tag{22}$$

This assumption is used during the Lyapunov proof.

The proposed adaptive NN controller structure based on tracking errors as feedback variables is shown in Fig. 1. The desired and actual state of the first variable is utilized to obtain the error, which when combined with the output of the first NN generates the virtual control input. This virtual control input is combined with the second state to become the NN inputs for the second action NN. The second NN output along with the tracking error of the second state is considered as the input to the nonstrict feedback nonlinear discrete-time system. The tuning of the two NN weights is accomplished using the tracking errors.

Theorem 1: Consider the system defined in (3) along with the Assumptions 1–3 hold. Let the disturbances and NN approximation errors be bounded, whose bounds are considered known as $|d_1'(k)| \le d_{1N}$, $|d_2'(k)| \le d_{2N}$, $|\varepsilon_1(k)| \le \varepsilon_{1N}$, and $|\varepsilon_2(k)| \le \varepsilon_{2N}$. Let the first NN weight tuning be given by

$$\hat{w}_1(k+1) = \hat{w}_1(k) - \alpha_1\phi(k)\left(\hat{w}_1^T(k)\phi(k) + k_1e_1(k)\right) \tag{23}$$

with the second NN weights tuning be provided by

$$\hat{w}_2(k+1) = \hat{w}_2(k) - \frac{\alpha_2}{k_2}\sigma(k)\left(\hat{w}_2^T(k)\sigma(k) + k_2e_2(k)\right) \tag{24}$$

where $\alpha_1 \in R$, $\alpha_2 \in R$, $k_1 \in R$, and $k_2 \in R$ are design parameters. Let the virtual and actual control inputs be defined by (9) and (18), respectively. The tracking errors $e_1(k)$ from (5) and $e_2(k)$ from (11) and the NN weights estimates $\hat{w}_1(k)$ and $\hat{w}_2(k)$ are UUB, with the bounds specifically given by (A.8)–(A.11) provided the design parameters are selected as

$$(1) \quad 0 < \alpha_1\|\phi(\cdot)\|^2 < 1 \tag{25}$$

$$(2) \quad 0 < \alpha_2\|\sigma(\cdot)\|^2 < k_2 \tag{26}$$

$$(3) \quad 0 < |k_1| < \frac{1}{2g_{1M}\sqrt{5+\frac{1}{k_2}}} \tag{27}$$

$$(4) \quad 0 < k_2 < \frac{-g_{2M} + \sqrt{(5+g_{2M}^2)}}{10g_{2M}}. \tag{28}$$

Proof: See the Appendix. ■

Remark 1: A well-defined controller is developed in this paper because a single NN in (7) and (17) is utilized to approximate a ratio of two unknown smooth nonlinear functions thereby avoiding the problem of $\hat{g}_i(k), \forall i = 1, 2$, becoming zero. This is in contrast from using a single NN for each of these individual functions consistent with the previous literature [6].

Remark 2: The NN weight tuning proposed in (23) and (24) renders a semirecurrent NN due to the proposed weight tuning law even though a feedforward NN architecture is utilized. Here,

the NN outputs are not fed as delayed inputs to the network whereas the outputs of each layer are fed as delayed inputs to the same layer. This semirecurrent NN architecture renders a dynamic NN, which is capable of predicting the state one step ahead.

Remark 3: It is only possible to show boundedness of all the closed-loop signals by using an extension of Lyapunov stability [6] due to the presence of approximation errors and bounded disturbances, a result consistent with the literature [4]–[6]. The controller gains k_1 and k_2 can be selected using (27) and (28) so that the closed-loop stability can be ensured.

IV. Adaptive Critic Controller Design

Next we present the development of an adaptive critic NN controller. Our objective is to design an NN controller for systems (1) and (2) such that: 1) all the signals in the closed-loop system remain UUB; 2) the state $x_1(k)$ follows a desired trajectory $x_{1d}(k)$; and 3) certain long term system performance index is optimized. Though the adaptive critic NN controller development uses the backstepping approach, the actual control methodology is different from the one introduced in Section III as given next.

A. Design of the Virtual Control Input $\hat{x}_{2dc}(k)$

For simplicity, let us denote

$$f_{1c}(k) = f_1\left(x_1(k), x_2(k)\right) + g_1\left(x_1(k), x_2(k)\right) x_2(k) + x_2(k) \tag{29}$$

$$f_{2c}(k) = f_{2c}\left(x_1(k), x_2(k)\right) \tag{30}$$

and

$$g_{2c}(k) = g_2\left(x_1(k), x_2(k)\right). \tag{31}$$

System (3) can be rewritten as

$$x_1(k+1) = f_{1c}(k) - x_2(k) + d_1'(k) \tag{32}$$

$$x_2(k+1) = f_{2c}(k) + g_{2c}(k)u(k) + d_2'(k). \tag{33}$$

Define the tracking error as

$$e_{1c}(k) = x_1(k) - x_{1d}(k) \tag{34}$$

where $x_{1d}(k)$ is the desired trajectory and subscript "c" is introduced to minimize confusion between the error signals of the two controllers developed in this paper. Using (32), (34) can be expressed as

$$\begin{aligned} e_{1c}(k+1) &= x_1(k+1) - x_{1d}(k+1) \\ &= f_{1c}(k) - x_2(k) - x_{1d}(k+1) + d_1'(k). \end{aligned} \tag{35}$$

By viewing $x_2(k)$ as a virtual control input, a desired virtual control signal can be designed as

$$x_{2dc}(k) = f_{1c}(k) - x_{1d}(k+1) + l_{1c}e_{1c}(k) \tag{36}$$

where $l_{1c} \in \Re$ is a design constant selected to stabilize the error system (35).

Because $f_{1c}(k)$ is an unknown function, the desired virtual control input $x_{2dc}(k)$ in (37) cannot be implemented in practice. By utilizing the first action NN to approximate this unknown function $f_{1c}(k)$, $x_{2dc}(k)$ is given by

$$\begin{aligned} x_{2dc}(k) &= w_{1c}^T \phi_{1c}\left(v_{1c}^T x(k)\right) + \varepsilon_{1c}\left(x_c(k)\right) \\ &\quad - x_{1d}(k+1) + l_{1c}e_{1c}(k) \\ &= w_{1c}^T \phi_{1c}(k) + \varepsilon_{1c}\left(x_c(k)\right) - x_{1d}(k+1) + l_{1c}e_{1c}(k) \end{aligned} \tag{37}$$

where $x_c(k) = [x_1(k), x_2(k)]^T \in \Re^2$ is the input vector to the first action NN, $w_{1c} \in \Re^{n_1}$ and $v_{1c} \in \Re^{2\times n_1}$ denote the constant ideal output- and hidden-layer weights, the hidden-layer activation function $\phi_{1c}(k) \in \Re^{n_1}$ represents $\phi_{1c}(v_{1c}^T x_c(k))$, n_1 is the number of the nodes in the hidden layer, and $\varepsilon_{1c}(x_c(k)) \in \Re$ is the approximation error. It is demonstrated in [11] that, if the hidden-layer weight v_{1c} is chosen initially at random and kept constant and the number of hidden-layer nodes is sufficiently large, the approximation error $\varepsilon_{1c}(x_c(k))$ can be made arbitrarily small so that the bound $\|\varepsilon_{1c}(x_c(k))\| \le \varepsilon_{1mc}$ holds for all $x_c(k) \in S$ because the activation function forms a basis.

Consequently, the virtual control $\hat{x}_{2dc}(k)$ is taken as

$$\begin{aligned} \hat{x}_{2dc}(k) &= \hat{w}_{1c}^T(k)\phi_{1c}\left(v_{1c}^T x_c(k)\right) - x_{1d}(k+1) + l_{1c}e_{1c}(k) \\ &= \hat{w}_{1c}^T(k)\phi_{1c}(k) - x_{1d}(k+1) + l_{1c}e_{1c}(k) \end{aligned} \tag{38}$$

where $\hat{w}_{1c}(k) \in \Re^{n_1}$ is the actual output-layer weight matrix to be tuned. The hidden-layer weight v_{1c} is randomly chosen initially and kept constant. Define the weight estimation error $\tilde{w}_{1c}(k) \in \Re^{n_1}$ by

$$\tilde{w}_{1c}(k) = \hat{w}_{1c}(k) - w_{1c}. \tag{39}$$

Define the error between $x_{2dc}(k)$ and $\hat{x}_{2dc}(k)$ as $e_{2c}(k) \in \Re$

$$e_{2c}(k) = x_{2dc}(k) - \hat{x}_{2dc}(k). \tag{40}$$

Equation (37) can be rewritten using (39) as

$$e_{1c}(k+1) = f_{1c}(k) - e_{2c}(k) - \hat{x}_{2dc}(k) + d_1(k) - x_{1d}(k+1). \tag{41}$$

Combining (40) with (41), we get

$$e_{1c}(k+1) = f_{1c}(k) - e_{2c}(k) - \hat{w}_{1c}^T(k)\phi_{1c}(k) + d_1'(k) - l_{1c}e_{1c}(k) \tag{42}$$

or, equivalently

$$e_{1c}(k+1) = -l_{1c}e_{1c}(k) - e_{2c}(k) - \zeta_{1c}(k) + \varepsilon_{1c}\left(x_c(k)\right) + d_1'(k) \tag{43}$$

where

$$\zeta_{1c}(k) = \tilde{w}_{1c}^T(k)\phi_c(k). \tag{44}$$

B. Design of the Control Input $u(k)$

Write the error $e_{2c}(k+1)$ from (40) as

$$\begin{aligned} e_{2c}(k+1) &= x_2(k+1) - \hat{x}_{2dc}(k+1) \\ &= f_{2c}(k) + g_{2c}(k)u(k) + d_2'(k) - \hat{x}_{2dc}(k+1) \end{aligned} \tag{45}$$

where $\hat{x}_{2dc}(k+1)$ is the future value of $\hat{x}_{2dc}(k)$. To stabilize the above system, the desired control input is chosen as

$$u_{dc}(k) = \frac{1}{g_{2c}(k)}\left(-f_{2c}(k) + \hat{x}_{2dc}(k+1) + l_{2c}e_{2c}(k)\right) \quad (46)$$

where $l_{2c} \in \Re$ is the controller gain to stabilize system (45). Note that $u_{dc}(k)$ depends upon future states because $\hat{x}_{2dc}(k+1)$ depends upon the $x_c(k+1)$. We solve this noncausal problem by using the universal NN approximator. It can be clear that $\hat{x}_{2dc}(k+1)$ is a nonlinear function of system state $x_c(k)$, virtual control input $\hat{x}_{2dc}(k)$, desired trajectory $x_{1d}(k+2)$, and system error $e_{1c}(k)$. Therefore, $\hat{x}_{2dc}(k+1)$ can be approximated using an NN. By taking $z_c(k) = [x_1(k), x_2(k), x_{1d}(k), \hat{x}_{2dc}(k)]^T \in \Re^4$ as the input to the NN, $u_{dc}(k)$ can be approximated as

$$\begin{aligned} u_{dc}(k) &= w_{2c}^T\phi_{2c}\left(v_{2c}^T z_c(k)\right) + \varepsilon_{2c}\left(z_c(k)\right) \\ &= w_{2c}^T\phi_{2c}(k) + \varepsilon_{2c}\left(z_c(k)\right) \end{aligned} \quad (47)$$

where $w_{1c} \in \Re^{n_2}$ and $v_{1c} \in \Re^{4\times n_2}$ denote the constant ideal output- and hidden-layer weights, the hidden-layer activation function $\phi_{2c}(k) \in \Re^{n_2}$ represents $\phi_{2c}(v_{2c}^T z(k))$, n_2 is the number of the nodes in the hidden layer, and $\varepsilon_{2c}(z(k)) \in \Re$ is the approximation error.

The actual control input is selected as the output of the second action NN

$$u(k) = \hat{w}_{2c}^T(k)\phi_{2c}\left(v_{2c}^T z_c(k)\right) = \hat{w}_{2c}^T(k)\phi_{2c}(k) \quad (48)$$

where $\hat{w}_{2c}(k) \in \Re^{n_2}$ is the actual output-layer weight.

Substituting (46)–(48) into (45), we get

$$\begin{aligned} e_{2c}(k+1) &= f_{2c}(k) + g_{2c}(k)\left(\hat{w}_{2c}^T(k)\phi_{2c}(k)\right) \\ &\quad + d_2'(k) - \hat{x}_{2dc}(k+1) \\ &= f_{2c}(k) + g_{2c}(k)\left(w_{2c}^T(k)\phi_{2c}(k)\right) \\ &\quad + g_{2c}(k)\zeta_2(k) + d_2'(k) - \hat{x}_{2d}(k+1) \\ &= f_{2c}(k) + g_{2c}(k)\left(u_d(k) - \varepsilon_{2c}\left(z_c(k)\right)\right) \\ &\quad + g_{2c}(k)\zeta_{2c}(k) + d_2'(k) - \hat{x}_{2dc}(k+1) \\ &= l_{2c}e_{2c}(k) + g_{2c}(k)\zeta_{2c}(k) - g_{2c}(k)\varepsilon_{2c}\left(z_c(k)\right) \\ &\quad + d_2'(k) \end{aligned} \quad (49)$$

where

$$\zeta_{2c}(k) = \hat{w}_{2c}^T(k)\phi_{2c}(k) - w_{2c}^T\phi_{2c}(k). \quad (50)$$

Equations (43) and (49) represent the closed-loop error dynamics. The next step is to design the adaptive critic NN controller weight updating rules. The critic NN is trained online to approximate the *strategic* utility function (long term system performance index). The critic signal, with a potential for estimating the future system performance, is employed to tune the two action NNs to minimize the strategic utility function and the unknown system estimation errors so that closed-loop stability is inferred.

C. The Strategic Utility Function

The utility function $p(k) \in \Re$ is defined based on the current system errors and it is given by

$$p(k) = \begin{cases} 0, & \text{if } (|e_{1c}(k)| + |e_{2c}(k)|) \le c \\ 1, & \text{otherwise} \end{cases} \quad (51)$$

where $c \in \Re$ is a predefined threshold. The utility function $p(k)$ is viewed as the current system performance index; $p(k) = 0$ and $p(k) = 1$ refer to the good and poor tracking performance, respectively.

The long term system performance measure or the *strategic* utility function $Q(k) \in \Re$ is defined as

$$Q(k) = \alpha^N p(k+1) + \alpha^{N-1}p(k+2) + \cdots + \alpha^{k+1}p(N) \quad (52)$$

where $\alpha \in \Re$ and $0 < \alpha < 1$, and N is the horizon. The term $Q(k)$ is viewed here as the long system performance measure because it is the sum of all future system performance indices. Equation (52) can also be expressed as $Q(k) = \min_{u(k)}\{\alpha Q(k-1) - \alpha^{N+1}p(k)\}$, which is similar to the standard Bellman equation.

D. Design of the Critic NN

The critic NN is used to approximate the *strategic* utility function $Q(k)$. We define the prediction error as

$$e_c(k) = \hat{Q}(k) - \alpha\left(\hat{Q}(k-1) - \alpha^N p(k)\right) \quad (53)$$

where the subscript "c" stands for the "critic,"

$$\hat{Q}(k) = \hat{w}_{3c}^T(k)\phi_{3c}\left(v_{3c}^T x(k)\right) = \hat{w}_{3c}^T(k)\phi_{3c}(k) \quad (54)$$

$\hat{Q}(k) \in \Re$ is the critic signal, $\hat{w}_{3c}(k) \in \Re^{n_3}$ and $v_{3c} \in \Re^{2\times n_3}$ represent the matrix of weight estimates, $\phi_{3c}(k) \in \Re^{n_3}$ is the activation function vector in the hidden layer, n_3 is the number of the nodes in the hidden layer, and the critic NN input is given by $x_c(k) \in \Re^2$. The objective function to be minimized by the critic NN is defined as

$$E_c(k) = \frac{1}{2}e_c^2(k). \quad (55)$$

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

$$\hat{w}_{3c}(k+1) = \hat{w}_{3c}(k) + \Delta\hat{w}_{3c}(k) \quad (56)$$

where

$$\Delta\hat{w}_{3c}(k) = \alpha_{3c}\left[-\frac{\partial E_c(k)}{\partial \hat{w}_{3c}(k)}\right] \quad (57)$$

or

$$\begin{aligned} \hat{w}_{3c}(k+1) &= \hat{w}_{3c}(k) - \alpha_{3c}\phi_{3c}(k) \\ &\quad \times \left(\hat{Q}(k) + \alpha^{N+1}p(k) - \alpha\hat{Q}(k-1)\right)^T \end{aligned} \quad (58)$$

where $\alpha_{3c} \in \Re$ is the NN adaptation gain.

E. Weight Updating Rule for the First Action NN

The first action NN $\hat{w}_{1c}^T(k)\phi_{1c}(k)$ weight is tuned by minimizing the functional estimation error $\zeta_{1c}(k)$ and the error between the desired *strategic* utility function $Q_d(k) \in \Re$ and the critic signal $\hat{Q}(k)$. Define

$$e_{a1}(k) = \zeta_{1c}(k) + \left(\hat{Q}(k) - Q_d(k)\right) \quad (59)$$

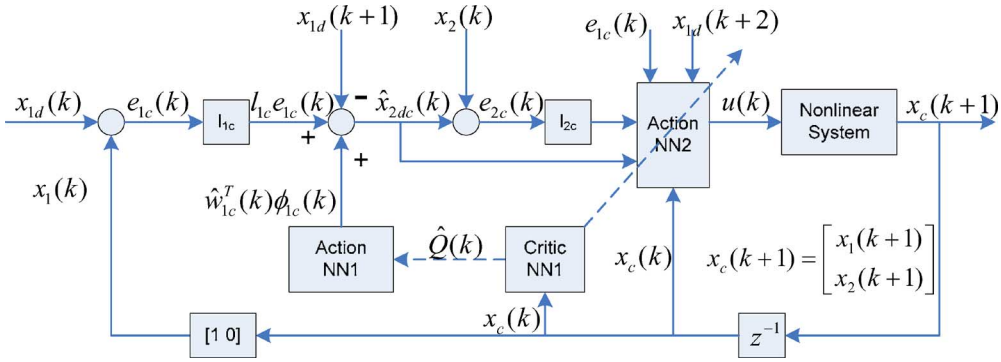

Fig. 2. Adaptive-critic-NN-based controller structure.

where $\zeta_{1c}(k)$ is defined in (44), $e_{a1}(k) \in \Re$, and the subscript "$a1$" stands for the "first action NN." The value for the desired *strategic* utility function $Q_d(k)$ is taken as "0" [18], i.e., to indicate that at every step, the nonlinear system can track the reference signal well. Thus, (59) becomes

$$e_{a1}(k) = \zeta_{1c}(k) + \hat{Q}(k). \tag{60}$$

The objective function to be minimized by the first action NN is given by

$$E_{a1}(k) = \frac{1}{2}e_{a1}^2(k). \tag{61}$$

The weight update rule for the action NN is also a gradient-based adaptation, which is defined as

$$\hat{w}_{1c}(k+1) = \hat{w}_{1c}(k) + \Delta\hat{w}_{1c}(k) \tag{62}$$

where

$$\Delta\hat{w}_{1c}(k) = \alpha_{1c}\left[-\frac{\partial E_{a1}(k)}{\partial \hat{w}_{1c}(k)}\right] \tag{63}$$

or

$$\hat{w}_{1c}(k+1) = \hat{w}_{1c}(k) - \alpha_{1c}\phi_{1c}(k)\left(\hat{Q}(k) + \zeta_{1c}(k)\right) \tag{64}$$

where $\alpha_{2c} \in \Re$ is the NN adaptation gain.

The NN weight updating rule in (64) cannot be implemented in practice because the target weight w_{1c} is unknown. However, using (43), the functional estimation error $\zeta_{1c}(k)$ is given by

$$\zeta_{1c}(k) = -e_{1c}(k+1) - l_{1c}e_{1c}(k) - e_{2c}(k) + \varepsilon_{1c}\left(x_c(k)\right) + d_1'(k). \tag{65}$$

Substituting (65) into (64), we get

$$\begin{aligned}\hat{w}_{1c}(k+1) = \hat{w}_{1c}(k) &- \alpha_{1c}\phi_{1c}(k)\hat{Q}(k) - \alpha_{1c}\phi_{1c}(k)\\ &\times\left(-e_{1c}(k+1) - l_{1c}e_{1c}(k) - e_{2c}(k)\right.\\ &\left.+\varepsilon_{1c}\left(x_c(k)\right) + d_1'(k)\right).\end{aligned} \tag{66}$$

Assume that bounded disturbance $d_1'(k)$ and the NN approximation error $\varepsilon_{1c}(x_c(k))$ are zeros for weight tuning implementation, then (66) is rewritten as

$$\begin{aligned}\hat{w}_{1c}(k+1) = \hat{w}_{1c}(k) &- \alpha_{1c}\phi_{1c}(k)\hat{Q}(k)\\ &+\alpha_{1c}\phi_{1c}(k)\left(e_{1c}(k+1) + l_{1c}e_{1c}(k) + e_{2c}(k)\right).\end{aligned} \tag{67}$$

Equation (67) is the adaptive-critic-based weight updating rule for the first action NN $\hat{w}_{1c}^T(k)\phi_{1c}(k)$. Next, we present the weight updating rule for the second action NN $\hat{w}_{2c}^T(k)\phi_{2c}(k)$.

F. Weight Updating Rule for the Second Action NN

Define

$$e_{a2}(k) = \sqrt{g_{2c}(k)}\zeta_{2c}(k) + \frac{\hat{Q}(k)}{\sqrt{g_{2c}(k)}} \tag{68}$$

where $\zeta_{2c}(k)$ is defined in (50), $g_{2c}(k) \in \Re^+$, and $e_{a2}(k) \in \Re$, where the subscript "$a2$" stands for the "second action NN." Following the similar design procedure and taking the bounded unknown disturbance $d_2(k)$ and the NN approximation error $\varepsilon_{2c}(z(k))$ to be zeros, the second action NN $\hat{w}_{2c}^T(k)\phi_{2c}(k)$ weight updating rule is given by

$$\begin{aligned}\hat{w}_{2c}(k+1) = \hat{w}_{2c}(k) &- \alpha_{2c}\phi_{2c}(k)\\ &\times\left(\hat{Q}(k) + e_{2c}(k+1) - l_{2c}e_{2c}(k)\right).\end{aligned} \tag{69}$$

The proposed adaptive critic architecture based on weight tuning (67) and (69) with $e_{1c}(k)$ and $e_{2c}(k)$ is similar to supervised actor–critic architecture [10] wherein the supervisory signals $e_{1c}(k)$ and $e_{2c}(k)$ supply an additional source of evaluative feedback or reward that essentially simplifies the task faced by the learning system. As the actor gains proficiency, the supervisory signals are then gradually withdrawn to shape the learned policy towards optimality.

To implement (69), the value of $e_{2c}(k+1)$ at the k instant has to be known, which can be obtained by the following steps.

1) Calculate $u(k)$, $\hat{x}_{2d}(k)$, $\hat{Q}(k)$, $e_{1c}(k)$, $e_{2c}(k)$, $z_c(k)$, $\hat{w}_{1c}(k)$, and $\hat{w}_{2c}(k)$ at the k instant.
2) Apply the control input $u(k)$ to system (3) to obtain the states $x_{1c}(k+1)$ and $x_{2c}(k+1)$.
3) Use the tracking error definition to get $e_{1c}(k+1)$ as

$$e_{1c}(k+1) = x_1(k+1) - x_{1dc}(k+1). \tag{70}$$

4) Use the first action NN weight updating rule (66) to get $\hat{w}_{1c}(k+1)$.
5) Once we have $\hat{w}_{1c}(k+1)$ and $x_c(k+1)$, the value of $\hat{x}_{2dc}(k+1)$ can be determined by using

$$\hat{x}_{2dc}(k+1) = \hat{w}_{1c}^T(k+1)\phi_{1c}(k+1) - x_{1d}(k+2) + l_{1c}e_{1c}(k+1) \tag{71}$$

and

$$e_{2c}(k+1) = x_{2c}(k+1) - \hat{x}_{2dc}(k+1). \tag{72}$$

The proposed adaptive critic NN controller is depicted in Fig. 2 based on the development given in this section. The desired and actual state of the first variable is utilized to obtain

the error, which when combined with the output of the first action NN generates the virtual control input. This virtual control input is combined with the second state and fed as inputs for the second action NN. The output of the second action NN becomes the input to the nonstrict feedback nonlinear discrete-time system. The critic NN by using the states of the nonlinear system approximates the strategic utility function, which is subsequently employed to tune the NN weights of both action NNs.

G. Stability Analysis

Assumption 3 (Bounded Ideal Weights): Let w_{1c}, w_{2c}, and w_{3c} be the unknown output-layer target weights for the two action NNs and the critic NN and assume that they are bounded above so that

$$\|w_{1c}\| \leq w_{1\,\mathrm{cm}} \quad \|w_{2c}\| \leq w_{2\,\mathrm{cm}} \quad \text{and} \quad \|w_{3c}\| \leq w_{3\,\mathrm{cm}} \tag{73}$$

where $w_{1\,\mathrm{cm}} \in \Re$, $w_{2\,\mathrm{cm}} \in \Re$, and $w_{3\,\mathrm{cm}} \in \Re$ represent the bounds on the unknown target weights where the Frobenius norm [11] is used.

Fact 1: The activation functions are bounded by known positive values so that

$$\|\phi_{ic}(k)\| \leq \phi_{i\mathrm{cm}}, \qquad i = 1, 2, 3 \tag{74}$$

where $\phi_{i\mathrm{cm}} \in \Re, i = 1, 2, 3$, is the upper bound for $\phi_{ic}(k), i = 1, 2, 3$.

Assumption 5 (Bounded NN Approximation Error): The NN reconstruction errors $\varepsilon_{1c}(x_c(k))$ and $\varepsilon_{2c}(z_c(k))$ are bounded over the compact set $S \subset \Re$ by $\varepsilon_{1\,\mathrm{cm}}$ and $\varepsilon_{2\,\mathrm{cm}}$, respectively [6].

Fact 1 and Assumption 5 are required during the Lyapunov proof.

Theorem 2: Consider the system given by (3). Let the Assumptions 1–4 hold and the disturbance bounds $d_{1\,\mathrm{m}}$ and $d_{2\,\mathrm{m}}$ be known constants. Let the critic NN $\hat{w}_{3c}^T(k)\phi_{3c}(k)$ weight tuning be given by (58), the first action NN $\hat{w}_{1c}^T(k)\phi_{1c}(k)$ weight tuning provided by (67), and the second action NN $\hat{w}_{2c}^T(k)\phi_{2c}(k)$ weight tuning provided by (69). Given the virtual control input $\hat{x}_{2dc}(k)$ (38) and the control input $u(k)$ (48), the tracking errors $e_{1c}(k)$ and $e_{2c}(k)$ and the NN weight estimates $\hat{w}_{1c}(k)$, $\hat{w}_{2c}(k)$, and $\hat{w}_{3c}(k)$ are UUB, with the bounds specifically given by (A.26)–(A.28) provided the controller design parameters are selected as

$$\text{(a)} \quad 0 < \alpha_{1c}\|\phi_{1c}(k)\|^2 < 1 \tag{75}$$

$$\text{(b)} \quad 0 < \alpha_{2c}\|\phi_{2c}(k)\|^2 < \frac{1}{g_{2c\max}} \tag{76}$$

$$\text{(c)} \quad 0 < \alpha_{2c}\|\phi_{3c}(k)\|^2 < 1 \tag{77}$$

$$\text{(d)} \quad |l_{1c}| < \frac{1}{2} \tag{78}$$

$$\text{(e)} \quad |l_{2c}| < \frac{\sqrt{3}}{3} \tag{79}$$

$$\text{(f)} \quad 0 < \alpha < \frac{\sqrt{2}}{2} \tag{80}$$

where α_{1c}, α_{2c}, and α_{3c} are NN adaptation gains, l_{1c} and l_{2c} are controller gains, and α is employed to define the *strategic* utility function.

Proof: See the Appendix. ■

Remark 4: The weight updates indicate that the critic NN and the functional approximation NNs have a semirecurrent architecture where the output from each node both in the input and output layers is fed back to the inputs.

Remark 5: The mutual dependence between the two NNs (action-generating and critic NNs) in the adaptive critic NN architecture results in coupled tuning law equations because the critic output $\hat{Q}(k)$ is utilized to tune the action NNs whereas the action NN outputs are utilized as inputs by the system to generate new states, which in turn are used by the critic NN. Moreover, additional complexities arise due to the addition of the second action NN because this addition causes further interaction among the three NNs. However, the Lyapunov stability analysis presented in the Appendix guarantees that the closed-loop system with all the three NNs is stable while ensuring the boundedness of all the signals.

Remark 6: The weights of the action and critic NNs can be initialized at zero or random. This means that there is no explicit offline learning phase needed in the proposed controller in contrast with the existing works where a preliminary offline training phase is normally used.

Remark 7: The proposed scheme results in a well-defined controller by avoiding the problem of $\hat{g}_i(k), \forall i = 1, 2$, becoming zero because a single NN is employed to approximate the ratio of two nonlinear smooth functions. Moreover, the controller gains l_{1c} and l_{2c} are selected using (78) and (79) to ensure closed-loop stability.

Remark 8: Condition (78) can be verified easily. For instance, the hidden layer of the critic NN consists of n_1 nodes with the hyperbolic tangent sigmoid function as its activation function, then $\|\phi_{1c}(\cdot)\|^2 \leq n_1$. The NN adaptation gain α_1 can be selected as $0 < \alpha_{1c} < (1/n_1)$ to satisfy (78). Similar analysis can be performed to obtain the NN adaptation gains α_{2c} and α_{3c}.

Remark 9: The number of hidden-layer neurons required for suitable approximation can be addressed by using the stability of the closed-loop system and the error bounds of the NNs. From (75)–(80) and Remark 8, to stabilize the closed-loop system, the numbers of the hidden-layer nodes can be selected as $n_1 < (1/\alpha_{1c})$, $n_2 < (1/\alpha_{2c}g_{2c\min})$, and $n_3 < (1/\alpha_{3c})$ once the NN adaptation gains α_{1c}, α_{2c}, and α_{3c} are selected. However, to get a better approximation performance and according to [11], the hidden-layer nodes have to be selected large enough to make the approximation error $\varepsilon(k)$ approach zero. To balance stability and good approximation requirement, we start with a small number of nodes, and increase it until the controller achieves the satisfactory performance.

Corollary 3: Given the hypothesis, the proposed adaptive critic NN controller, and the weight updating rules in Theorem 2, the state $x_{2c}(k)$ approaches the desired virtual control input $x_{2dc}(k)$.

Proof: Combining (37) and (38), the difference between $\hat{x}_{2dc}(k)$ and $x_{2dc}(k)$ is given by

$$\begin{aligned}\hat{x}_{2dc}(k) - x_{2dc}(k) &= \tilde{w}_{1c}(k)\phi_{1c}(k) - \varepsilon_{1c}\left(x_c(k)\right) \\ &= \zeta_{1c}(k) - \varepsilon_{1c}\left(x_c(k)\right)\end{aligned} \tag{81}$$

where $\tilde{w}_{1c}(k) \in \Re^{n_1}$ defined in (39) is the first action NN weight estimation error and $\zeta_{1c}(k) \in \Re$ is defined in (44). Be-

cause both $\zeta_{1c}(k) \in \Re$ and $\varepsilon_{1c}(x_c(k))$ are bounded, $\hat{x}_{2dc}(k)$ is bounded to $x_{2dc}(k)$. In Theorem 2, we show that $e_{2c}(k)$ is bounded, i.e., the state $x_{2c}(k)$ is bounded to the virtual control signal $\hat{x}_{2dc}(k)$. Thus, the state $x_{2c}(k)$ is bounded to the desired virtual control signal $x_{2dc}(k)$. ■

Remark 10: Feedforward NNs are used as building blocks both in tracking error and adaptive-critic-based NN controllers. In the case of the first control methodology, tracking error is used as a feedback signal to tune the NN weights online. The only objective there is to reduce the tracking error, and therefore, no performance criterion is set. To the contrary, adaptive critic NN architectures use reinforcement learning signal generated by a critic NN. The critic signal can be generalized using complex optimization criteria including the variant of the standard Bellman equation. As a consequence, an adaptive critic NN architecture results in a considerable computational overhead due to the addition of a second NN for generating the critic signal. In the proposed work, a single NN is used to generate a critic signal for tuning the two action-generating NN weights. As a result, computational complexity is slightly reduced but still requires three NN when compared to two NN in the case of tracking-error-based NN controller. Moreover, all the NNs are tuned online compared to standard work in the adaptive critic NN literature [11]–[13]. Lyapunov-based analysis is demonstrated for stability whereas available adaptive critic papers use purely numerical simulation results without any analytical proofs. Simulation results are presented in Section V to justify the theoretical conclusions.

V. Simulations

The purpose of the simulation is to verify the performance of the adaptive critic NN controller. Two cases are considered. The first is to apply the proposed adaptive critic NN controller to a nonlinear system. Then, a practical nonlinear system [e.g., emission control in spark ignition (SI) engine] is considered and the proposed approach is employed.

Example 1 (Adaptive Critic Controller for Nonstrict Feedback Nonlinear System): The control objective is to make the state $x_1(k)$ follow the desired trajectory $x_{1d}(k)$. The proposed adaptive critic NN controller is used on the following nonlinear system, given in nonstrict feedback form:

$$x_1(k+1) = -\frac{3}{16}\left(\frac{x_1(k)}{1+x_2^2(k)}\right) + 0.5x_1(k) + (3+1.5\sin(x_2(k)))\,x_2(k) \quad (82)$$

$$x_2(k+1) = -\frac{1}{16}\left(\frac{x_2(k)}{1+x_1^2(k)}\right) + x_2(k) + (0.2+0.15*\cos(x_1(k)))\,u(k) \quad (83)$$

where $x_i(k) \in R, i = 1,2$, are the states and $u(k) \in R$ is the control input. Note that both $f_1(k)$ and $g_1(k)$ include state $x_2(k)$.

The reference signal was selected as $x_{1d}(k) = 2\sin(\omega kT + \xi)$, where $\omega = 0.5$ and $xi = \pi/2$ with a sampling interval of $T = 50$ msec. The total simulation time is taken as 250 s. The gains of the standard proportional controller are selected *a priori* $l_{1c} = -0.2$ and $l_{2c} = 0.3$ using (78) and (79).

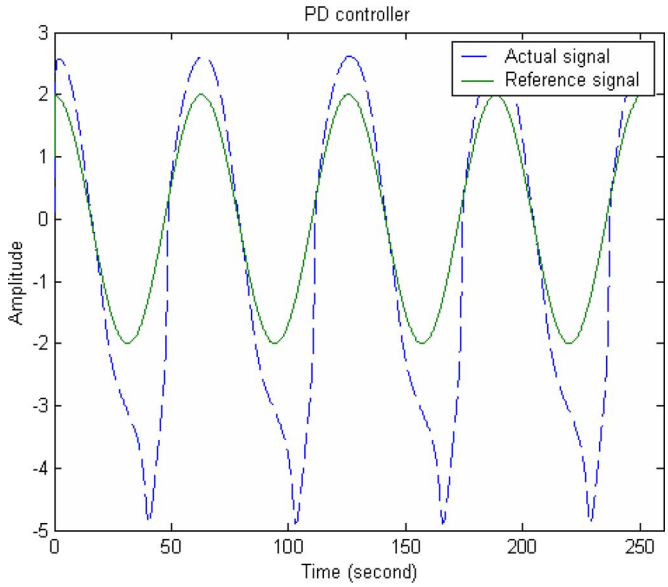

Fig. 3. Performance of a standard controller without NN.

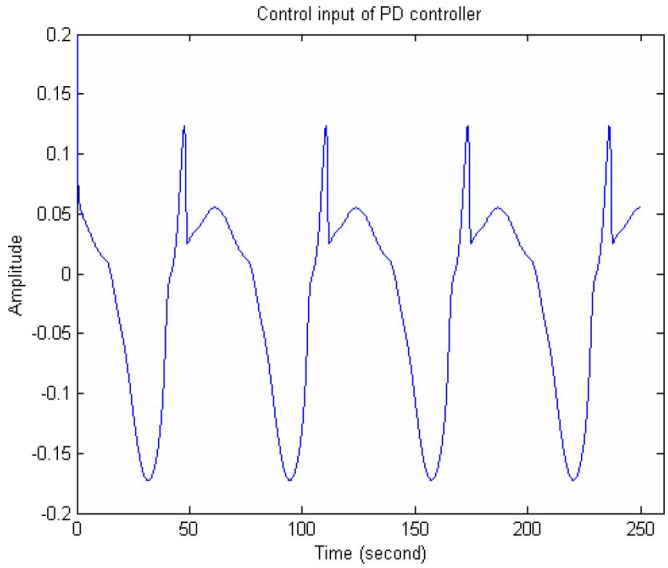

Fig. 4. Control input.

NN1 $\hat{w}_1^T\phi(k)$, NN2 $\hat{w}_2^T\sigma(k)$, and critic NN3 $\hat{w}_3^T\varphi(k)$ each consists of 15 nodes in the hidden layer. For weight updating, the learning rates are selected as $\alpha_{1c} = 0.1$, $\alpha_{2c} = 0.03$, and $\alpha_{3c} = 0.1$. All the initial weights are selected at random over an internal of $[0,1]$ and all the activation functions used are hyperbolic tangent sigmoid functions.

Figs. 3 and 4 present the performance of the standard proportional controller alone from the adaptive critic controller and associated control input, respectively, without the NNs included. From Fig. 3, it is clear that the tracking performance has deteriorated in comparison with Fig. 5 when the NNs were included in the controller. Fig. 5 illustrates the superior performance of the adaptive critic NN controller. The gains were not altered in both simulations. Fig. 6 depicts the NN control input that appears to be sufficiently smooth such that it can be implemented in today's embedded system hardware. Because the control input

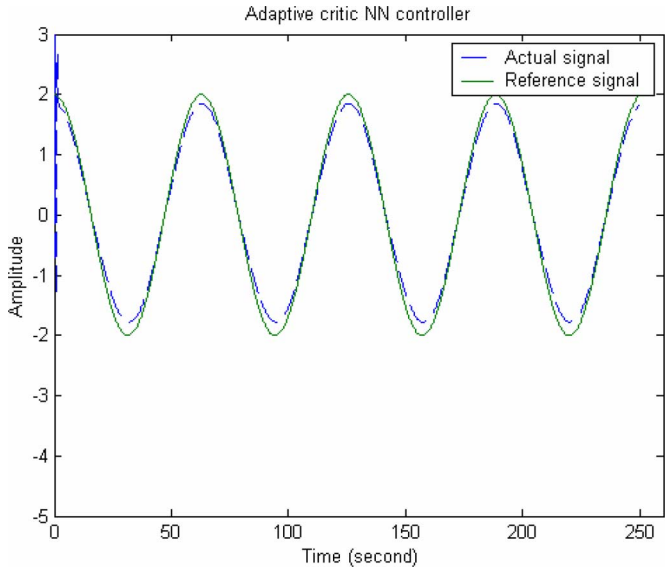

Fig. 5. Performance of the adaptive critic NN controller.

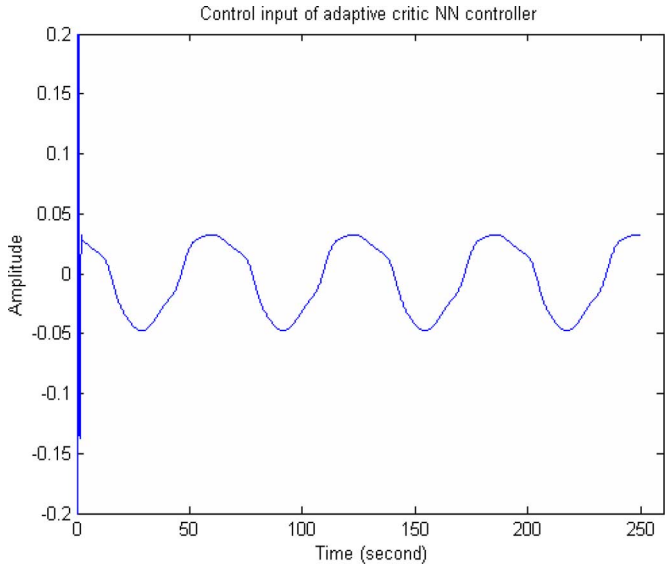

Fig. 6. Adaptive critic NN control input.

is bounded and according to (38) and (48), the NN weights are indeed bounded. The NNs are not trained offline and the output layer weights are initialized at zero. Next, we present another simulation example where a practical nonlinear system is considered and the proposed controller is applied.

Example 2 (Adaptive NN Controller for SI Engines: A Practical Example): Lean operation of SI engine allows low emissions and improved fuel efficiency. However, at lean operation, the engine becomes unstable due to the cyclic dispersion in heat release. Literature shows that by controlling engines at lean operating conditions can reduce emissions as much as 60% [18] and it improves fuel efficiency by about 5% to 10%. Unfortunately, the engine exhibits strong cyclic dispersion, which causes instability. The simulation is designed to verify the performance of the proposed adaptive NN controller for the practical application, where the objective is to reduce cyclic dispersion. The adaptive NN controller is designed to stabilize the SI engine operating at lean conditions. The engine dynamics can be expressed as a nonstrict feedback nonlinear discrete-time system of second order of the form [12]–[14] given by

$$x_1(k+1)=(1-F(k))AF+F(k)x_1(k) \\ -R\cdot F(k)CE(k)x_2(k) \tag{84}$$

$$x_2(k+1)=(1-CE(k))F(k)x_2(k) \\ +(1-F(k))(MF(k)+u(k)) \tag{85}$$

$$CE(k)=\frac{CE_{\max}}{1+100^{-(\varphi(k)-\varphi_m)/(\varphi_u-\varphi_l)}} \quad \varphi(k)=R\frac{x_2(k)}{x_1(k)} \tag{86}$$

$$\varphi_m=\frac{\varphi_u+\varphi_l}{2} \quad H(k)=x_2(k)CE(k) \tag{87}$$

where $x_1(k)$ and $x_2(k)$ are the mass of air and fuel before kth burn, respectively, $F(k)$ is the unknown residual gas fraction, AF is the mass of fresh air fed per cycle, R is the stoichiometric air–fuel ratio, $\sim$ 14.6, $CE(k)$ is the combustion efficiency, $MF(k)$ is the mass of fresh fuel per cycle, $u(k)$ is the small changes in mass of fresh fuel per cycle, $CE_{\max}$ is the maximum combustion efficiency, which is a constant, $\varphi(k)$ is the equivalence ratio, $\varphi_m, \varphi_u, \varphi_l$ are constant system parameters, and $H(k)$ is the heat release in the kth cycle. Because $H(k)$ varies cycle by cycle, the engine is considered unstable without any control. In (56) and (57), $F(k)$ and $CE(k)$ are unknown nonlinear functions of both $x_1(k)$ and $x_2(k)$, so the system is a nonlinear discrete-time system of second order in nonstrict feedback form.

Given Theorem 2 and Corollary 3 and using the proof in [6], we could show that, with the proposed controller, both states can be bounded to their respective target values x_{1d} and x_{2d}. Then, the equivalence ratio $\varphi(k)$ (86) combustion efficiency $CE(k)$ (86), heat release $H(k)$ (87), and the engine dynamics are stabilized.

The cyclic dispersion is reduced when the variation in equivalence ratio $(x_2(k)/x_1(k))$ is reduced, and this goal can be achieved by driving both states close enough to their desired target. The system parameters are selected as the following: $\varphi = 0.71, F = 0.14, AF = 222.2, MF = 10.81, \varphi_u = 0.685, \varphi_l = 0.665, CE_{\max} = 0.9, x_{1d} = 0.9058$, and $x_{2d} = 9.3895$. We add the unknown white noise with the deviation of 0.1081 and 0.007 to the F and MF.

The controller gains are selected as $k_1 = k_2 = 0.1$ using (27) and (28). For weight updating, the adaptation gains are selected as $\alpha_1 = 0.01$ and $\alpha_2 = 0.001$. The two NNs have 15 hidden-layer nodes each. All the hidden-layer weights are selected uniformly within an interval of $[0, 1]$ and all the activation functions are selected as hyperbolic tangent sigmoid functions.

The cyclic dispersion observed at a lean equivalence ratio of 0.71 is presented in Fig. 7 when no control scheme is employed for 10 000 cycles. It shows that, without any control, the engine performance is unsatisfactory. Fig. 8 illustrates the performance of the NN controller. The dispersion is small and bounded and can be tolerable. Fig. 9 depicts the error between actual equivalence ratio and its desired value, which is bounded. Fig. 10 displays the norm of the weights $\|\hat{w}_1(k)\|$, $\|\hat{w}_2(k)\|$, $\|\hat{w}_3(k)\|$, and $e_1(k)$. It is clear that the NN weights and the error $e_1(k)$ converge and are bounded. The performance of a tuned conventional proportional and derivative controller is illustrated in [13]

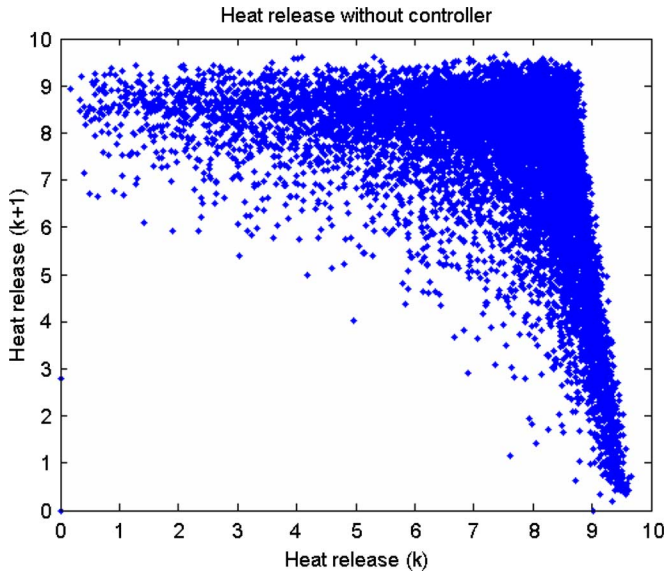

Fig. 7. Cyclic dispersion without control.

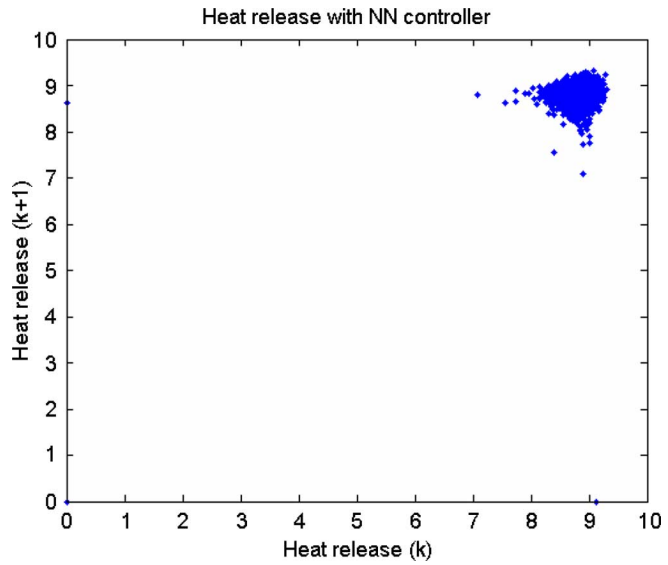

Fig. 8. Heat release with NN controller.

and from the results in [13], the NN controller outperforms the conventional controllers. Next the proposed adaptive critic NN controller is applied.

Adaptive Critic NN Controller: The simulation parameters are selected as follows: 1000 cycles are considered at equivalence ratio of 0.71 with $R = 14.6$, $F(k) = 0.14$, mass of new air $= 1.0$, the standard deviation of mass of new fuel is 0.007, $\phi_u = 0.685$, $\phi_l = 0.665$, the desired mass of air is taken as $X_{1d} = 0.9058$, and the desired mass of fuel is calculated as $X_{2d} = R \times 0.71 \times X_{1d} = 9.3895$. A 5% unknown noise is added to the residual gas fraction as a way to include stochastic perturbation of system parameters. The gains of controllers are selected as $l_{1c} = l_{2c} = 0.1$, respectively, using (78) and (79).

NN1 $\hat{w}_1^T \phi(k)$, NN2 $\hat{w}_2^T \sigma(k)$, and critic NN3 $\hat{w}_3^T \varphi(k)$ each consists of 15 nodes in the hidden layer. For weight updating, the learning rates are selected as $\alpha_{1c} = 0.01$, $\alpha_{2c} = 0.001$, and $\alpha_{3c} = 0.1$. The initial weights are selected randomly over an internal of $[0, 1]$ and all the activation functions are hyperbolic tangent sigmoid functions.

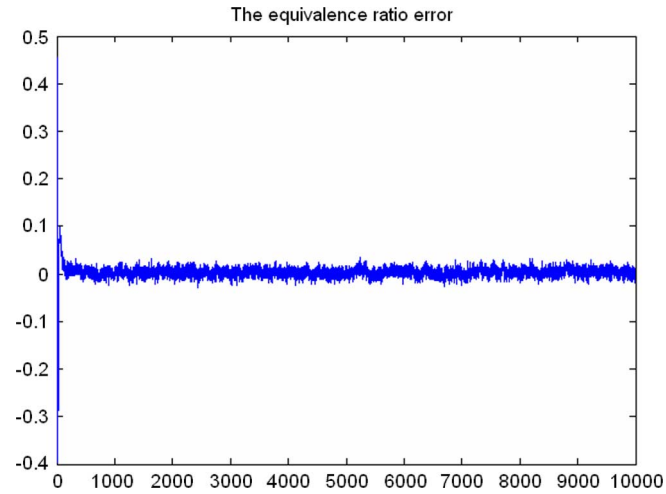

Fig. 9. Error in equivalence ratio.

Fig. 10 shows the cyclic dispersion without control now for 1000 cycles at an equivalence ratio of 0.71. Here, the combustion process dynamics consisting of residual gas fraction and combustion efficiency are taken unknown. The cyclic dispersion is presented in Fig. 11, which indicates that, without any control, the engine performance is unsatisfactory. By contrast, Fig. 12 displays that the engine works satisfactorily at lean conditions, but the heat release appears to exhibit minimal dispersion. The overall controller performance appears to be satisfactory and fuel efficiency is 8% better than the standard adaptive NN controller due to optimal design.

A comparison of the two controller approaches—a tracking error NN based and adaptive critic NN based—is shown in Example 2 highlighting the differences, whereas in Example 1, an adaptive critic NN controller is compared with that of a standard controller. These clearly show that an adaptive critic NN controller is far superior even though it is computationally intensive than a tracking-error-based NN controller. On the other hand, Example 1 illustrates that a tracking-error-based NN controller performs better than a conventional controller. These clearly demonstrate that an adaptive critic NN controller renders a near-optimal performance for a nonlinear discrete-time system in nonstrict feedback form.

VI. Conclusion

This paper proposes a novel adaptive critic NN controller to deliver a desired tracking performance for a class of discrete-time nonstrict feedback nonlinear systems under the assumption that the states are available for measurement. A well-defined controller is developed since a single NN is utilized to approximate the ratio of two smooth nonlinear functions. Two NNs are employed for generating suitable control inputs in the case of tracking-error-based controller whereas three NNs are employed for the adaptive critic NN controller. The stability analysis of the closed-loop control system is introduced and the *boundedness* of the tracking error is demonstrated for both designs. The performance of the controller is demonstrated on a

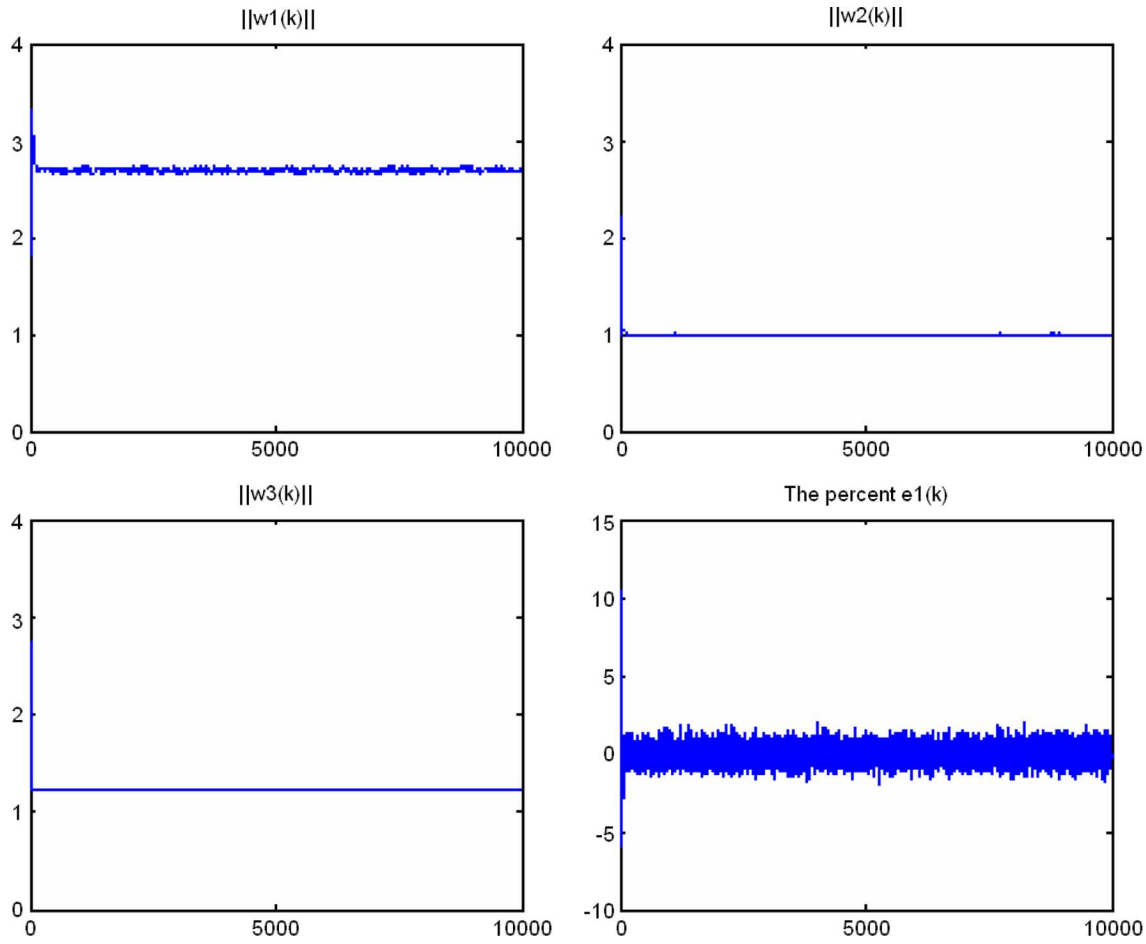

Fig. 10. NN weight norms and error $e_1(k)$.

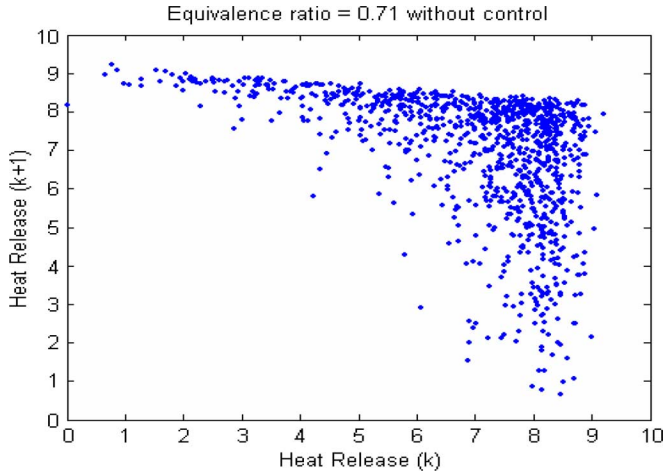

Fig. 11. Cyclic dispersion without control.

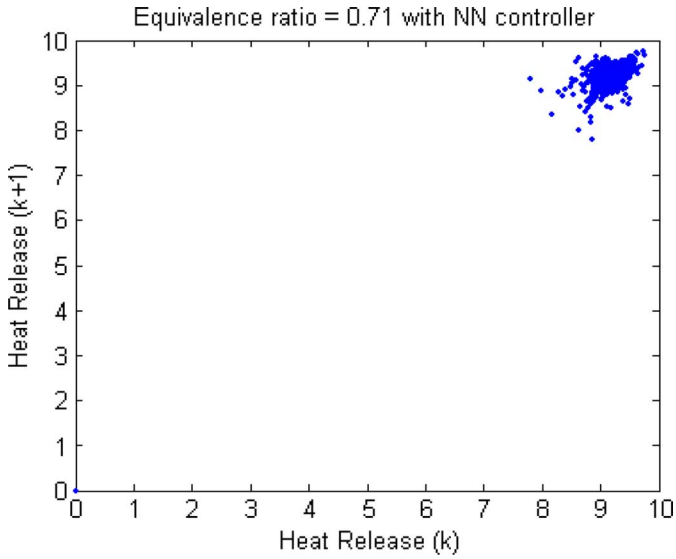

Fig. 12. Cyclic dispersion with control.

practical nonlinear system. This paper proposes a novel adaptive NN controller to deliver a desired tracking performance for the control of a second-order unknown nonlinear discrete-time system in nonstrict feedback form. The NN controllers do not require an offline learning phase and the weights can be initialized at zero or random. Results show that the performance of the proposed controllers is highly satisfactory while meeting the closed-loop stability.

APPENDIX

Proof of Theorem 1: Define the Lyapunov function candidate

$$J(k)=\frac{e_1^2(k)}{8g_{1M}^2}+\frac{e_2^2(k)}{8k_2g_{2M}^2}+\frac{1}{\alpha_1}\widetilde{w}_1^T(k)\widetilde{w}_1(k)+\frac{1}{\alpha_2}\widetilde{w}_2^T(k)\widetilde{w}_2(k) \tag{A.1}$$

where g_{1M} and g_{2M} are the upper bounds of function $g_1(k)$ and $g_2(k)$, respectively, given a compact set (see Assumption 2), and k_2, α_1, and α_2 are design parameters (see Theorem 1).

The first difference of Lyapunov function is given by

$$\Delta J(k)=\Delta J_1(k)+\Delta J_2(k)+\Delta J_3(k)+\Delta J_4(k). \tag{A.2}$$

The first difference $\Delta J_1(k)$ is obtained using (13) as

$$\begin{aligned}\Delta J_1(k)&=\frac{1}{8g_{1M}^2}\left(e_1^2(k+1)-e_1^2(k)\right)\\&=\frac{1}{8g_{1M}^2}\left(\left(g_1(k)\left(k_1e_1(k)+e_2(k)\right.\right.\right.\\&\qquad\left.\left.\left.+\zeta_1(k)+d_1(k)\right)\right)^2-e_1^2(k)\Big)\\&\le\frac{1}{2}\left(\left(k_1^2-\frac{1}{4g_{1M}^2}\right)e_1^2(k)+e_2^2(k)\right.\\&\qquad\left.+\zeta_1^2(k)+d_1^2(k)\right).\end{aligned} \tag{A.3}$$

Now take the second term in the first difference (A.1) and substitute (19) into (A.1) to get

$$\begin{aligned}\Delta J_2(k)&=\frac{1}{8k_2g_{2M}^2}\left(e_2^2(k+1)-e_2^2(k)\right)\\&=\frac{1}{8k_2g_{2M}^2}\left(\left(g_2(k)\left(k_1e_1(k)+k_2e_2(k)\right.\right.\right.\\&\qquad\left.\left.\left.+\zeta_2(k)+d_2(k)\right)\right)^2-e_2^2(k)\Big)\\&\le\frac{1}{2}\left((1/k_2)\left(k_2^2-\frac{1}{3g_{2M}^2}\right)e_2^2(k)+\frac{\zeta_2^2(k)}{k_2}\right.\\&\qquad\left.+\frac{d_2^2(k)}{k_2}+\frac{k_1^2e_1^2(k)}{k_2}\right)\end{aligned} \tag{A.4}$$

Take the third term in (A.1) and substitute the weights updates from (23) and simplify to get

$$\begin{aligned}\Delta J_3(k)&=\frac{1}{\alpha_1}\widetilde{w}_1^T(k+1)\widetilde{w}_1(k+1)-\frac{1}{\alpha_1}\widetilde{w}_1^T(k)\widetilde{w}_1(k)\\&=\frac{1}{\alpha_1}\left[\left(I-\alpha_1\phi(k)\phi^T(k)\right)\widetilde{w}_1(k)\right.\\&\qquad\left.-\alpha_1\phi(k)\left(w_1^T(k)\phi(k)+k_1e_1(k)\right)\right]^T\\&\quad\times\left[\left(I-\alpha_1\phi(k)\phi^T(k)\right)\widetilde{w}_1(k)\right.\\&\qquad\left.-\alpha_1\phi(k)\left(w_1^T(k)\phi(k)+k_1e_1(k)\right)\right]\\&\quad-\frac{1}{\alpha_1}\widetilde{w}_1^T(k)\widetilde{w}_1(k)\\&=-\left(2-\alpha_1\phi^T(k)\phi(k)\right)\zeta_1^2(k)\\&\quad-2\left(1-\alpha_1\phi^T(k)\phi(k)\right)\left(w_1^T\phi(k)+k_1e_1(k)\right)\zeta_1(k)\\&\quad+\alpha_1\phi^T(k)\phi(k)\left(w_1^T\phi(k)+k_1e_1(k)\right)^2.\end{aligned} \tag{A.5}$$

Take the fourth term in (A.1) and substitute the weights updates from (24) and simplify to get

$$\begin{aligned}\Delta J_4(k)&=\frac{1}{\alpha_2}\widetilde{w}_2^T(k+1)\widetilde{w}_2(k+1)-\frac{1}{\alpha_2}\widetilde{w}_2^T(k)\widetilde{w}_2(k)\\&=\frac{1}{\alpha_2}\left[\left(I-\frac{\alpha_2}{k_2}\sigma(k)\sigma^T(k)\right)\widetilde{w}_2(k)\right.\\&\qquad\left.-\frac{\alpha_2}{k_2}\sigma(k)\left(w_2^T\sigma(k)+k_2e_2(k)\right)\right]^T\\&\quad\times\left[\left(I-\frac{\alpha_2}{k_2}\sigma(k)\sigma^T(k)\right)\widetilde{w}_2(k)\right.\\&\qquad\left.-\frac{\alpha_2}{k_2}\sigma(k)\left(w_2^T\sigma(k)+k_2e_2(k)\right)\right]\\&\quad-\frac{1}{\alpha_2}\widetilde{w}_2^T(k)\widetilde{w}_2(k)\\&=-\left(\frac{1}{k_2}\right)\left(2-\frac{\alpha_2}{k_2}\sigma^T(k)\sigma(k)\right)\zeta_2^2(k)\\&\quad-2\left(\frac{1}{k_2}\right)\left(1-\frac{\alpha_2}{k_2}\sigma^T(k)\sigma(k)\right)\\&\quad\times\left(w_2^T\sigma(k)+k_2e_2(k)\right)\zeta_2(k)\\&\quad+\frac{\alpha_2}{k_2^2}\sigma^T(k)\sigma(k)\left(w_2^T\sigma(k)+k_2e_2(k)\right)^2.\end{aligned} \tag{A.6}$$

Combine (A.3), (A.4), (A.5), and (A.6) to get the first difference and simplify to get

$$\begin{aligned}\Delta J(k)&=\Delta J_1(k)+\Delta J_2(k)+\Delta J_3(k)+\Delta J_4(k)\\&\le\frac{1}{2}\left(k_1^2+\frac{k_1^2}{k_2}-\frac{1}{4g_{1M}^2}\right)e_1^2(k)\\&\quad+\frac{1}{2k_2}\left(k_2^2+k_2-\frac{1}{4g_{2M}^2}\right)e_2^2(k)-\frac{1}{2}\xi_1^2(k)\\&\quad-\frac{1}{2k_2}\xi_2^2(k)-\left(1-\alpha_1\phi^T(k)\phi(k)\right)\\&\quad\times\left(\zeta_1^2(k)+2\left(w_1^T(k)\phi(k)+k_1e_1(k)\right)\zeta_1(k)\right)\\&\quad-\frac{1}{k_2}\left(1-\frac{\alpha_2}{k_2}\sigma^T(k)\sigma(k)\right)\\&\quad\times\left(\zeta_2^2(k)+2\left(w_2^T\sigma(k)+k_2e_2(k)\right)\zeta_2(k)\right)\\&\quad+\alpha_1\phi^T(k)\phi(k)\left(w_1^T\phi(k)+k_1e_1(k)\right)^2\\&\quad+\frac{\alpha_2}{k_2^2}\left(w_2^T\sigma(k)+k_2e_2(k)\right)^2\sigma^T(k)\sigma(k)\\&\quad+d_1^2(k)+\frac{d_2^2(k)}{k_2}\\\Delta J(k)&\le\frac{1}{2}\left(5k_1^2+\frac{k_1^2}{k_2}-\frac{1}{4g_{1M}^2}\right)e_1^2(k)\\&\quad+\frac{1}{2k_2}\left(3k_2^2+k_2-\frac{1}{4g_{2M}^2}\right)e_2^2(k)-\frac{1}{2}\xi_1^2(k)\\&\quad-\frac{1}{2k_2}\xi_2^2(k)-\left(1-\alpha_1\phi^T(k)\phi(k)\right)\\&\quad\times\left(\zeta_1(k)+w_1^T\phi(k)+k_1e_1(k)\right)^2\\&\quad-(1/k_2)\left(1-\frac{\alpha_2}{k_2}\sigma^T(k)\sigma(k)\right)\\&\quad\times\left(\zeta_2(k)+w_2^T\sigma(k)+k_2e_2(k)\right)^2+d_1^2(k)+d_2^2(k)/k_2\\&\quad+2w_{1\max}^2\phi_{\max}^2+w_{2\max}^2\sigma_{\max}^2/k_2.\end{aligned} \tag{A.7}$$

This implies that $\Delta J\le 0$ as long as (25) through (28) hold and

$$|e_1(k)|>\frac{2\sqrt{2}g_{1M}D_M}{\sqrt{1-20k_1^2g_{1M}^2-\frac{4g_{1M}^2k_1^2}{k_2}}} \tag{A.8}$$

or

$$|e_2(k)| > \frac{2\sqrt{2k_2}g_{2M}D_M}{\sqrt{(1-4k_2g_{2M}^2-20k_2^2g_{2M}^2)}} \quad (A.9)$$

or

$$|\zeta_1(k)| > \sqrt{2}D_M \quad (A.10)$$

or

$$|\zeta_2(k)| > \sqrt{2k_2}D_M \quad (A.11)$$

where

$$D_M^2 = d_1^2(k)+\frac{d_2^2(k)}{k_2}+2w_{1\max}^2\phi_{\max}^2+\frac{w_{2\max}^2\sigma_{\max}^2}{k_2}. \quad (A.12)$$

According to a standard Lyapunov extension theorem [6], this demonstrates that the system tracking errors and the weight estimation errors are UUB. The boundedness of $|\zeta_1(k)|$ and $|\zeta_2(k)|$ implies that $\|\tilde{w}_1(k)\|$ and $\|\tilde{w}_2(k)\|$ are bounded, and this further implies that the weight estimates $\hat{w}_1(k)$ and $\hat{w}_2(k)$ are bounded. Therefore, all the signals in the closed-loop system are bounded.

Proof of Theorem 2: Define the Lyapunov function

$$J(k)=\frac{\gamma_1}{4}e_{1c}^2(k)+\frac{\gamma_2}{3}e_{2c}^2(k)+\frac{\gamma_3}{\alpha_{1c}}\tilde{w}_{1c}^T(k)\tilde{w}_{1c}(k)+\frac{\gamma_4}{\alpha_2'(k)} \times\tilde{w}_{2c}^T(k)\tilde{w}_{2c}(k)+\frac{\gamma_5}{\alpha_{3c}}\tilde{w}_{3c}^T(k)\tilde{w}_{3c}(k)+\gamma_6\zeta_{3c}^2(k-1) \quad (A.13)$$

where $0<\gamma_i, i=1,\ldots,6$, are constants, $\zeta_{3c}(k-1)$ is defined as

$$\zeta_{3c}(k-1)=\left(\hat{w}_{3c}^T(k-1)-w_{3c}^T\right)\phi_{3c}(k-1) \quad (A.14)$$

and $\alpha_2'(k)$ is defined as

$$\alpha_2'(k)=\alpha_{2c}g_{2c}(k)=\alpha_{2c}g_2(k) \quad (A.15)$$

and α_{1c}, α_{2c}, and α_{3c} are the NN adaptation gains. The Lyapunov function consisting of the tracking errors and the weights estimation errors obviates the need for certainty equivalence assumption.

The first difference of Lyapunov function is given by

$$\Delta J(k)=\Delta J_1(k)+\Delta J_2(k)+\Delta J_3(k)+\Delta J_4(k) +\Delta J_5(k)+\Delta J_6(k). \quad (A.16)$$

The first difference $\Delta J_1(k)$ is obtained using (43) as

$$\Delta J_1(k)\le\gamma_1 l_{1c}^2e_{1c}^2(k)+\gamma_1e_{2c}^2(k)+\gamma_1\zeta_{1c}^2(k) +\left(\varepsilon_{1c}(k)+d_1'(k)\right)^2-\frac{\gamma_1}{4}e_{1c}^2(k). \quad (A.17)$$

Now taking the second term in the first difference (A.13) and substituting (49) into (A.16), we get

$$\Delta J_2(k)\le\gamma_2 l_{2c}^2e_{2c}^2(k)+\gamma_2g_{2\max}^2\zeta_{2c}^2(k) +\gamma_2\left(g_{2c}(k)\zeta_{2c}(k)+d_2'(k)\right)^2-\frac{\gamma_2}{3}e_{2c}^2(k). \quad (A.18a)$$

Taking the third term in (A.16) and substituting the weights updates from (67) and simplifying, we get

$$\Delta J_3(k)\le-\gamma_3\left(1-\alpha_{1c}\phi_{1c}^T(k)\phi_{1c}(k)\right) \times\left(\zeta_{1c}(k)+\hat{Q}(k)-\varepsilon_{1c}(k)-d_1'(k)\right)^2 -\gamma_3\zeta_{1c}^2(k)+2\gamma_3\zeta_{3c}^2(k) +2\gamma_3\left(w_{3c}^T\phi_{3c}(k)+\varepsilon_{1c}(k)+d_1'(k)\right)^2. \quad (A.18b)$$

Taking the fourth term in (A.16) and substituting the weights updates from (69) and simplifying $\Delta J_4(k)$, we get

$$\Delta J_4(k)\le-\gamma_4\left(1-\alpha_{2c}'(k)\phi_{2c}^T(k)\phi_{2c}(k)\right) \times\left(\zeta_{2c}(k)+\frac{\hat{Q}(k)}{g_{2c}(k)}-\varepsilon_{2c}(k)+\frac{d_2(k)}{g_{2c}(k)}\right)^2 -\gamma_4\zeta_{2c}^2(k)+\frac{2\gamma_4}{g_{2c\min}^2}\zeta_{3c}^2(k) +2\gamma_4\left(\varepsilon_{2c}(k)+\frac{w_{3c}^T\phi_{3c}(k)-d_2'(k)}{g_{2c}(k)}\right)^2. \quad (A.19)$$

Using the critic NN weights updating rule (58) to calculate the fifth and sixth item in (A.16), we obtain

$$\Delta J_5(k)\le-\gamma_5\left(1-\alpha_{3c}\phi_{3c}^T(k)\phi_{3c}(k)\right) \times\left(\zeta_{2c}(k)+w_{3c}^T\phi_{3c}(k)+\alpha^{N+1}p(k)-\alpha p(k-1)\right)^2 -\gamma_5\zeta_{3c}^2(k)+2\gamma_5\alpha^2\zeta_{3c}^2(k-1) +2\gamma_5\left(w_{3c}^T\left(\phi_{3c}(k)-\alpha\phi_{3c}(k-1)\right)+\alpha^{N+1}p(k)\right)^2 \quad (A.20)$$

$$\Delta J_6(k)=\gamma_6\zeta_{3c}^2(k)-\gamma_6\zeta_{3c}^2(k-1). \quad (A.21)$$

Taking $\gamma_6=2\alpha^2\gamma_5$, and combining (A.17)–(A.21) to get the first difference of the Lyapunov function and simplifying it, we get

$$\Delta J(k)\le\gamma_1\left(l_{1c}^2-\frac{1}{4}\right)e_{1c}^2(k)+\left(\gamma_1+\gamma_2l_{2c}^2-\frac{\gamma_2}{3}\right)e_{2c}^2(k) +D_m^2+(\gamma_1-\gamma_2)\zeta_{1c}^2(k)+\left(\gamma_2g_{2c\max}^2-\gamma_4\right)\zeta_{2c}^2(k) -\gamma_3\left(1-\alpha_{1c}\phi_{1c}^T(k)\phi_{1c}(k)\right) \times\left(\zeta_{1c}(k)+\hat{Q}(k)-\varepsilon_{1c}(k)-d_1'(k)\right)^2 -\gamma_4\left(1-\alpha_{2c}'(k)\phi_{2c}^T(k)\phi_{2c}(k)\right) \times\left(\zeta_{2c}(k)+\frac{\hat{Q}(k)}{g_{2c}(k)}-\varepsilon_2(k)+\frac{d_2'(k)}{g_{2c}(k)}\right)^2 -\gamma_5\left(1-\alpha_{3c}\phi_{3c}^T(k)\phi_{3c}(k)\right) \times\left(\zeta_{2c}(k)+w_{3c}^T\phi_{3c}(k)+\alpha^{N+1}p(k)-\alpha(k-1)\right)^2 +\left(2\gamma_3+\frac{2\gamma_4}{g_{2c\max}^2}-\gamma_5+2\gamma_5\alpha^2\right)\zeta_{3c}^2(k) \quad (A.22)$$

where

$$D_M^2=2(1+3\gamma_3)\varepsilon_{1cm}^2+2\left(\gamma_2g_{2m}^2+3\gamma_4\right)\varepsilon_{2cm}^2 +2(1+3\gamma_3)d_{1m}^2+2\gamma_5\alpha^{2N+2}+2\left(\gamma_2+\frac{3\gamma_4}{g_{2c\min}^2}\right)d_{2m}^2 +6\left(\gamma_3+\frac{\gamma_4}{g_{2c\min}^2}+\gamma_5(1+\alpha^2)\right)w_{3cm}^2\phi_{3cm}^2. \quad (A.23)$$

Select

$$\gamma_3>\gamma_1 \quad \gamma_4>\gamma_2g_{2\max}^2 \quad (A.24)$$

$$\gamma_5 > \frac{2\gamma_3 + \frac{2\gamma_4}{g_{2c\min}^2}}{1-2\alpha^2} \quad \text{and} \quad \gamma_2 > \frac{\gamma_1}{\frac{1}{3}-l_{2c}^2}. \tag{A.25}$$

This implies $\Delta J(k) < 0$ as long as (75)–(80) hold and

$$|e_{1c}(k)| > \frac{D_M}{\sqrt{\gamma_1\left(\frac{1}{4}-l_{1c}^2\right)}}$$

or

$$|e_{2c}(k)| > \frac{D_M}{\sqrt{\frac{\gamma_2}{3}-\gamma_2 l_{2c}^2-\gamma_1}} \tag{A.26}$$

or

$$|\zeta_{1c}(k)| > \frac{D_M}{\sqrt{\gamma_3-\gamma_1}}$$

or

$$|\zeta_{2c}(k)| > \frac{D_M}{\sqrt{\gamma_4-\gamma_2 g_{2c\max}^2}} \tag{A.27}$$

or

$$|\zeta_{3c}(k)| > \frac{D_M}{\sqrt{\gamma_5-2\gamma_5\alpha^2-2\gamma_3-\frac{2\gamma_4}{g_{2c\min}^2}}}. \tag{A.28}$$

According to a standard Lyapunov extension theorem [6], this demonstrates that the system tracking error and the weight estimation errors are UUB. The boundedness of $\|\zeta_{1c}(k)\|$, $\|\zeta_{2c}(k)\|$, and $\|\zeta_{3c}(k)\|$ implies that $\|\tilde{w}_{1c}(k)\|$, $\|\tilde{w}_{2c}(k)\|$, and $\|\tilde{w}_{3c}(k)\|$ are bounded, and this further implies that the weight estimates $\hat{w}_{1c}(k)$, $\hat{w}_{2c}(k)$, and $\hat{w}_{3c}(k)$ are bounded. Therefore, all the signals in the closed-loop system are bounded.

REFERENCES

[1] I. Kanellakopoulos, P. V. Kokotovic, and A. S. Morse, "Systematic design of adaptive controllers for feedback linearizable systems," *IEEE Trans. Autom. Control*, vol. 36, no. 11, pp. 1241–1253, Nov. 1991.

[2] P. V. Kokotovic, "The joy of feedback: Nonlinear and adaptive," *IEEE Control Syst. Mag.*, vol. 12, no. 3, pp. 7–17, Jun. 1992.

[3] F. C. Chen and H. K. Khalil, "Adaptive control of nonlinear discrete-time systems using neural networks," *IEEE Trans. Autom. Control*, vol. 40, no. 5, pp. 791–801, May 1995.

[4] F. L. Lewis, S. Jagannathan, and A. Yesilderek, *Neural Network Control of Robot Manipulators and Nonlinear Systems*. New York: Taylor and Francis, 1999.

[5] F. L. Lewis, J. Campos, and R. Selmic, *Neuro-Fuzzy Control of Industrial Systems With Actuator Nonlinearities*. Philadelphia, PA: SIAM, 2002.

[6] S. Jagannathan, *Neural Network Control of Nonlinear Discrete-Time Systems*. Orlando, FL: CRC Press, 2006.

[7] S. Jagannathan, "Control of a class of nonlinear discrete-time systems using multi layer neural networks," *IEEE Trans. Neural Netw.*, vol. 12, no. 5, pp. 1113–1120, Sep. 2001.

[8] P. J. Werbos, "ADP: Goals, opportunities and principles," in *Handbook of Learning and Approximate Dynamic Programming*, ser. Computational Intelligence, J. Si, A. G. Barto, W. B. Powell, and D. Wunsch, II, Eds. Piscataway, NJ: IEEE Press, 2004, pp. 3–44.

[9] P. J. Werbos, "A menu of designs for reinforcement learning over time," in *Neural Networks for Control*, W. T. Miller, R. S. Sutton, and P. J. Werbos, Eds. Cambridge, MA: MIT Press, 1991, pp. 67–95.

[10] M. T. Rosenstein and A. G. Barto, "Supervised actor-critic reinforcement learning," in *Handbook of Learning and Approximate Dynamic Programming*, ser. Computational Intelligence, J. Si, A. G. Barto, W. B. Powell, and D. Wunsch, II, Eds. Piscataway, NJ: IEEE Press, 2004, pp. 337–358.

[11] B. Igelnik and Y. H. Pao, "Stochastic choice of basis functions in adaptive function approximation and the functional-link net," *IEEE Trans. Neural Netw.*, vol. 6, no. 6, pp. 1320–1329, Nov. 1995.

[12] C. S. Daw, C. E. A. Finney, M. B. Kennel, and F. T. Connolly, "Observing and modeling nonlinear dynamics in an internal combustion engine," *Phys. Rev. E, Stat. Phys. Plasmas Fluids Relat. Interdiscip. Top.*, vol. 57, no. 3, pp. 2811–2819, 1997.

[13] P. He and S. Jagannathan, "Neuroemission controller for reducing cyclic dispersion in lean combustion spark ignition engines," *Automatica*, vol. 41, pp. 1133–1142, 2005.

[14] T. Inoue, S. Matsushita, K. Nakanishi, and H. Okano, *Toyota Lean Combustion System—The Third Generation System*, ser. SAE Technical Papers. New York: SAE, 1993, Pub. 930873.

[15] G. G. Lendaris and J. C. Neidhoefer, "Guidance in the use of adaptive critics for control," in *Handbook of Learning and Approximate Dynamic Programming*, ser. Computational Intelligence, J. Si, A. G. Barto, W. B. Powell, and D. Wunsch, II, Eds. Piscataway, NJ: IEEE Press, 2004, pp. 97–124.

[16] J. J. Murray, C. Cox, G. G. Lendaris, and R. Saeks, "Adaptive dynamic programming," *IEEE Trans. Syst. Man Cybern. C, Appl. Rev.*, vol. 32, no. 2, pp. 140–153, May 2002.

[17] D. P. Bertsekas and J. N. Tsitsiklis, *Neuro-Dynamic Programming*. Belmont, MA: Athena Scientific, 1996.

[18] J. Si and Y. T. Wang, "On-line learning control by association and reinforcement," *IEEE Trans. Neural Netw.*, vol. 12, no. 2, pp. 264–276, Mar. 2001.

[19] X. Lin and S. N. Balakrishnan, "Convergence analysis of adaptive critic based optimal control," in *Proc. Amer. Control Conf.*, 2000, pp. 1929–1933.

Sarangapani Jagannathan (S'89–M'89–SM'99) received the B.S. degree in electrical engineering from College of Engineering, Guindy, Anna University, Madras, India, in 1987, the M.S. degree in electrical engineering from the University of Saskatchewan, Saskatchewan, Saskatoon, SK, Canada, in 1989, and the Ph.D. degree in electrical engineering from the University of Texas, San Antonio, in 1994.

During 1986–1987, he was a Junior Engineer at Engineers India Limited, New Delhi, India, as a Research Associate and Instructor from 1990 to 1991, at the University of Manitoba, Winnipeg, MB, Canada, and worked at Systems and Controls Research Division, Caterpillar Inc., Peoria, IL, as a Consultant during 1994–1998. During 1998–2001, he was at the University of Texas at San Antonio, and since September 2001, he has been at the University of Missouri, Rolla, where he is currently Rutledge-Emerson Distinguished Professor in the Department of Electrical and Computer Engineering and Site Director for the NSF Industry/University Cooperative Research Center on Intelligent Maintenance Systems. He has coauthored more than 190 refereed conference and juried journal articles and several book chapters and three books entitled *Neural Network Control of Robot Manipulators and Nonlinear Systems* (London, U.K.: Taylor & Francis, 1999), *Discrete-Time Neural Network Control of Nonlinear Discrete-Time Systems* (Boca Raton, FL: CRC Press, 2006), and *Wireless Ad Hoc and Sensor Networks: Performance, Protocols and Control* (Boca Raton, FL: CRC Press, 2007). He currently holds 17 patents and several are in process. His research interests include adaptive and neural network control, computer/communication/sensor networks, prognostics, and autonomous systems/robotics.

Dr. Jagannathan received several gold medals and scholarships during his undergraduate program. He was the recipient of Region 5 IEEE Outstanding Branch Counselor Award in 2006, Faculty Excellence Award in 2006, St. Louis Outstanding Branch Counselor Award in 2005, Teaching Excellence Award in 2005, Caterpillar Research Excellence Award in 2001, Presidential Award for Research Excellence at UTSA in 2001, National Science Foundation (NSF) CAREER award in 2000, Faculty Research Award in 2000, Patent Award in 1996, and Sigma Xi "Doctoral Research Award" in 1994. He has served and currently serving on the program committees of several IEEE conferences. He is currently serving as the Associate Editor for the IEEE TRANSACTIONS ON CONTROL SYSTEMS TECHNOLOGY, the IEEE TRANSACTIONS ON NEURAL NETWORKS, the IEEE TRANSACTIONS ON SYSTEMS ENGINEERING, and on several program committees. He is a member of Tau Beta Pi, Eta Kappa Nu, and Sigma Xi and IEEE Committee on Intelligent Control. He served as the Program Chair for the 2007 IEEE International Symposium on Intelligent Control, and Publicity Chair for the 2007 International Symposium on Adaptive Dynamic Programming.

Pingan He received the M.S. degree in electrical engineering from the University of Missouri, Rolla, in 2004.

Currently, he is a Controls Engineer at GM Power Train Group, Troy MI. His research interests include power train control systems.

Mr. He is a member of Society of Automotive Engineers.