

Reinforcement Learning Neural-Network-Based Controller for Nonlinear Discrete-Time Systems With Input Constraints

Pingan He and S. Jagannathan, *Senior Member, IEEE*

Abstract—A novel adaptive-critic-based neural network (NN) controller in discrete time is designed to deliver a desired tracking performance for a class of nonlinear systems in the presence of actuator constraints. The constraints of the actuator are treated in the controller design as the saturation nonlinearity. The adaptive critic NN controller architecture based on state feedback includes two NNs: the critic NN is used to approximate the “strategic” utility function, whereas the action NN is employed to minimize both the strategic utility function and the unknown nonlinear dynamic estimation errors. The critic and action NN weight updates are derived by minimizing certain quadratic performance indexes. Using the Lyapunov approach and with novel weight updates, the uniformly ultimate boundedness of the closed-loop tracking error and weight estimates is shown in the presence of NN approximation errors and bounded unknown disturbances. The proposed NN controller works in the presence of multiple nonlinearities, unlike other schemes that normally approximate one nonlinearity. Moreover, the adaptive critic NN controller does not require an explicit offline training phase, and the NN weights can be initialized at zero or random. Simulation results justify the theoretical analysis.

Index Terms—Approximate dynamic programming, neural network control, optimal control, reinforcement learning.

I. INTRODUCTION

ADAPTIVE control schemes, both in continuous and discrete time, were developed in the past several decades [1], [9], [13], [17]. Discrete-time implementation of controllers is of importance since all the controllers have to be implemented on today’s embedded hardware. Discrete-time adaptive control design is more complex than continuous-time due primarily to the fact that discrete-time Lyapunov derivatives are quadratic in state [7], not linear as in the continuous case. This has led to traditional techniques where parameter identification is decoupled from control using the so-called certainty equivalence (CE). Moreover, adaptive control schemes require that the nonlinear systems under consideration satisfy the linear-in-the-unknown-parameter (LIP) assumption.

Manuscript received January 21, 2006; revised May 9, 2006 and July 5, 2006. This work supported in part by National Science Foundation Grants ECCS 0296191, 0378777, and 0621924 and in part by the Intelligent Systems Center. This paper was recommended by Associate Editor F. Lewis.

The authors are with the Department of Electrical and Computer Engineering, University of Missouri-Rolla, Rolla, MO 65409 USA (e-mail: ph8p5@umr.edu; sarangap@umr.edu).

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/TSMCB.2006.883869

In recent years, learning-based control methodology using neural networks (NNs) has become an alternative to adaptive control since NNs are considered as general tools for modeling nonlinear systems. Work on adaptive NN control using the universal NN approximation property is now pursued by several groups of researchers, e.g., [4], [5], [10], and [14]. However, work so far on NN control is accomplished using either supervised training [14], where the user specifies a desired output, or classical adaptive control ideas [4], [5], [10], [14], where a short-term system-performance measure is defined by using the tracking error at the current time instant. Additionally, the control of nonlinear systems is attempted by most of these works in continuous time, whereas a few results [5], [14] are available for discrete time. Lyapunov stability is demonstrated for closed-loop systems in many of these works [4], [5].

Adaptive actor-critic NN-based control has emerged as a promising NN approach due to its potential to find approximate solutions to dynamic programming. In the actor-critic NN-based control, a long-term system-performance measure can be optimized, in contrast to the short-term performance measure used in classical adaptive and NN control. There are many variants of adaptive critic NN-based control schemes [2], [16], [18], [19], [21], [22], namely 1) heuristic dynamic programming (HDP), 2) dual HDP (DHP), and 3) globalized DHP (GDHP). Lyapunov stability is rarely addressed in adaptive critic designs [3], [11], [12], [15], [20]. Moreover, an offline training scheme is usually employed, except in [20]. In [3] and [20], the convergence issue based on recursive stochastic algorithms is presented, where convergence with a probability of one is achieved. In [11], the critic is used to approximate the Hamilton–Jacobi–Bellman equation, and error convergence for a linear time-invariant discrete system is only addressed. In [12], “hard computing techniques” were utilized to verify the stability for nonlinear systems in continuous time. An algorithm is presented in [15] in order to approximate the Lyapunov function by using an adaptive critic NN method.

A novel adaptive NN controller consisting of an action plus a critic NN is developed in this paper to control a class of nonlinear discrete-time systems. The proposed approach utilizes a supervised actor-critic NN methodology [23], where an additional signal is used to evaluate the action NNs (or actors). The critic NN approximates a certain “strategic” utility function that is similar to a standard Bellman equation, which is taken as the long-term performance measure of the system. The action

NN weights are tuned online by both the critic NN signal and the filtered tracking error to minimize the strategic utility function and uncertain system dynamic estimation errors so that the action NN can generate the optimal control signal. This optimal action NN control signal combined with an additional outer-loop conventional control signal is applied as the overall control input to the nonlinear discrete-time system. The outer-loop conventional signal allows the action and critic NNs to learn online while making the system stable. This conventional signal that uses the tracking error is viewed as the “supervisory” signal [23].

By selecting the appropriate objective functions for both critic and action NNs, the closed-loop stability using Lyapunov is inferred in the presence of NN approximation errors and unknown yet bounded disturbances for the overall nonlinear discrete-time system. The selection of such a Lyapunov function in this paper is a critical part of the stability proof in comparison with the existing works [3], [11], [12], [15], [20], and it is rarely straightforward.

Available adaptive critic NN controllers, for instance [21], employ a backpropagation NN learning scheme to tune the weights of the action-generating and critic NNs so that an explicit offline training phase is used, whereas with the proposed scheme, the initial weights can be selected at zero or random, and they can be tuned online. Additionally, the “actuator constraints” are considered in this paper, in contrast with the previous adaptive critic designs. The actuator constraints are treated as saturation nonlinearity and introduced as an auxiliary linear system that is similar to [8], where such constraints are considered for a linear system. By appropriately selecting the NN weight updates based on a quadratic performance index, an optimal/suboptimal control sequence can be generated.

In [24], a novel reinforcement-based “output” feedback controller design was introduced recently for multi-input multi-output nonlinear discrete-time systems by using Lyapunov stability analysis. In contrast, the proposed NN controller is developed for a different class of nonlinear discrete-time systems by using “state” feedback. The critic and action NN designs are altogether different in the proposed controller design when compared to [24]. In short, the nonlinear system under consideration, the overall NN controller design, development, and associated Lyapunov stability analysis are quite different here than in [24]. The net result is the simultaneous compensation of multiple nonlinearities such as unknown dynamics and actuator constraints, in contrast with all the existing NN controller works [2]–[5], [10]–[12], [14]–[16], [18]–[23], where a single nonlinearity is normally compensated.

This paper is organized as follows. Section II provides background on the universal approximation property of NNs, the definition of uniformly ultimately bounded (UUB), and the filtered tracking error dynamics. The proposed adaptive critic NN methodology and the associated NN weight tuning updates without saturation constraints are presented in Section III. Section IV details the novel adaptive critic-based NN controller in the presence of saturation constraints, and its stability analysis is discussed. Simulation results are illustrated in Section V, whereas Section VI presents the conclusions.

II. BACKGROUNDS AND THE FILTERED ERROR DYNAMICS

A. Approximation Property

For a suitable approximation of unknown nonlinear functions, several NN architectures are currently available. In [6], it is shown that a continuous function $f(x(k)) \in C(S)$ within a compact subset S of \mathfrak{R}^n can be approximated using a single-layer feedforward NN as

$$f(x(k)) = w^T \phi(v^T x(k)) + \varepsilon(x(k)) \quad (1)$$

where w and v are the target weights of the hidden to the output and input to the hidden layers, respectively; $\phi(v^T x(k))$ denotes the vector of activation functions (usually, they are chosen as sigmoidal functions) at instant k ; and $\varepsilon(x(k))$ is the NN functional approximation error vector. The actual NN output is defined as

$$\hat{f}(x(k)) = \hat{w}^T(k) \phi(v^T x(k)) \quad (2)$$

where $\hat{w}(k)$ is the actual weight matrix. For simplicity, $\phi(v^T x(k))$ is denoted as $\phi(k)$.

The input to the hidden-layer weights v is selected at random initially, and it will not be tuned. The output-layer weights $\hat{w}(k)$ are tunable. Then, it is demonstrated in [6] that if the number of hidden-layer nodes is sufficiently large, the approximation error $\varepsilon(x(k))$ can be made arbitrarily small on the compact set so that the bound $\|\varepsilon(x(k))\| \leq \varepsilon_m$ holds for all $x(k) \in S$ since the activation function vector will form a basis.

B. Stability of Systems

To formulate the discrete-time controller, the following stability notion is needed. Consider the nonlinear system given by

$$\begin{aligned} x(k+1) &= f(x(k), u(k)) \\ y(k) &= h(x(k)) \end{aligned} \quad (3)$$

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

C. Nonlinear System Description

Consider the following nonlinear system to be controlled:

$$\begin{aligned} x_1(k+1) &= x_2(k) \\ &\vdots \\ x_n(k+1) &= f(x(k)) + u(k) + d(k) \end{aligned} \quad (4)$$

where $x(k) = [x_1^T(k), x_2^T(k), \dots, x_n^T(k)]^T \in \mathfrak{R}^{nm}$ with each $x_i(k) \in \mathfrak{R}^m$, $i = 1, \dots, n$ is the state at time instant k ; $f(x(k)) \in \mathfrak{R}^m$ is the unknown nonlinear dynamics of the system; $u(k) \in \mathfrak{R}^m$ is the input; and $d(k) \in \mathfrak{R}^m$ is the unknown but bounded disturbance, whose bound is assumed to be a known constant $\|d(k)\| \leq d_m$. Several NN learning schemes are proposed recently in the literature to control the class of

nonlinear systems described in (4) and [10], but the main contribution of this paper is the adaptive critic NN-based controller in the presence of magnitude constraints and the associated stability analysis.

Given a desired trajectory $x_{\text{nd}}(k) \in \mathfrak{R}^m$ and its past values, define tracking error $e_i(k) \in \mathfrak{R}^m$ as

$$e_i(k) = x_i(k) - x_{\text{nd}}(k + i - n) \quad (5)$$

and the filtered tracking error $r(k) \in \mathfrak{R}^m$ as

$$r(k) = [\Lambda \ I]e(k) \quad (6)$$

with $e(k) = [e_1^T(k), e_2^T(k), \dots, e_n^T(k)]^T$; $e_1(k+1) = e_2(k)$, where $e_1(k+1)$ is the next future value for error $e_1(k)$; $e_{n-1}(k), \dots, e_1(k)$ are past values of error $e_n(k)$; $I \in \mathfrak{R}^{m \times m}$ is an identity matrix; $\Lambda = [\lambda_{n-1}, \lambda_{n-2}, \dots, \lambda_1] \in \mathfrak{R}^{m \times (n-1)m}$; and $\lambda_i \in \mathfrak{R}^{m \times m}, i = 1, \dots, (n-1)$ is the constant diagonal positive definite matrix selected such that the eigenvalues are within a unit disc. Consequently, if the filtered tracking error $r(k)$ tends to zero, then all the tracking errors go to zero. Equation (6) can be expressed as

$$r(k+1) = f(x(k)) - x_{\text{nd}}(k+1) + \lambda_1 e_n(k) + \dots + \lambda_{n-1} e_2(k) + u(k) + d(k). \quad (7)$$

The control objective is to make all the tracking errors bounded close to zero and all the internal signals UUB.

D. Basic Controller Design Using Filtered Tracking Error

Define the control input $u(k) \in \mathfrak{R}^m$ as

$$u(k) = x_{\text{nd}}(k+1) - \hat{f}(x(k)) + l_v r(k) - \lambda_1 e_n(k) - \dots - \lambda_{n-1} e_2(k) \quad (8)$$

where $\hat{f}(x(k)) \in \mathfrak{R}^m$ is an estimate of the unknown function $f(x(k))$ and $l_v \in \mathfrak{R}^{m \times m}$ is a diagonal gain matrix. Then, the closed-loop system becomes

$$r(k+1) = l_v r(k) - \tilde{f}(x(k)) + d(k) \quad (9)$$

where the functional estimation error is given by $\tilde{f}(x(k)) = \hat{f}(x(k)) - f(x(k))$.

Equation (9) relates the filtered tracking error with the functional estimation error. In general, the filtered tracking error system (9) can also be expressed as

$$r(k+1) = l_v r(k) + \delta_0(k) \quad (10)$$

where $\delta_0(k) = -\tilde{f}(x(k)) + d(k)$. If the functional estimation error $\tilde{f}(x(k))$ is bounded above such that $\|\tilde{f}(x(k))\| \leq f_M$ for some known value $f_M \in \mathfrak{R}$, then the next stability results hold.

Theorem 2.1: Consider the system given by (4). Let the control action be provided by (8). Assume that the functional estimation error and the unknown disturbance are bounded. The filtered tracking error system (7) is stable provided that

$$0 < l_{v \max} < 1 \quad (11)$$

where $l_{v \max} \in \mathfrak{R}$ is the maximum eigenvalue of matrix l_v .

Proof: Let us consider the following Lyapunov function candidate:

$$J(k) = r(k)^T r(k). \quad (12)$$

The first difference is

$$\Delta J(k) = r(k+1)^T r(k+1) - r(k)^T r(k). \quad (13)$$

Substituting the filtered tracking error dynamics (9) in (13) results in

$$\begin{aligned} \Delta J(k) &= \left(l_v r(k) - \tilde{f}(x(k)) + d(k) \right)^T \\ &\quad \times \left(l_v r(k) - \tilde{f}(x(k)) + d(k) \right) - r(k)^T r(k). \end{aligned} \quad (14)$$

It implies that $\Delta J(k) \leq 0$ provided that $\|(l_v r(k) - \tilde{f}(x(k)) + d(k))\| \leq l_{v \max} \|r(k)\| + f_M + d_M < \|r(k)\|$. This further implies that

$$\|r(k)\| < \frac{f_M + d_M}{(1 - l_{v \max})}. \quad (15)$$

The closed-loop system is UUB. ■

III. ADAPTIVE NN CONTROLLER DESIGN

To minimize the computational overhead, a single-layer NN is considered here for both critic and action NNs, and the NN controller is designed first without considering the input constraints. A novel strategic utility function is defined, and it is taken as the long-term performance measure for the system. The NN critic signal approximates the utility function. The action NN signal is constructed to minimize this strategic utility function. By using a quadratic optimization function, the critic NN and action NN weight tuning laws are derived. Stability analysis using the Lyapunov direct method is carried out for the closed-loop system (7) with novel weight tuning updates. In the next section, the controller design with input constraints is dealt.

A. Strategic Utility Function

The utility function $p(k) = [p_i(k)]_{i=1}^m \in \mathfrak{R}^m$ is defined based on the filtered tracking error $r(k)$, and it is given by

$$p_i(k) = \begin{cases} 0, & \text{if } r_i^2(k) \leq c \\ 1, & \text{if } r_i^2(k) > c \end{cases}, \quad i = 1, 2, \dots, m \quad (16)$$

where $p_i(k) \in \mathfrak{R}, i = 1, \dots, m$ and $c \in \mathfrak{R}$ is a predefined threshold. The binary utility function $p(k)$ is viewed as the current system-performance index: $p_i(k) = 0$ stands for good tracking performance, and $p_i(k) = 1$ stands for poor tracking performance. The long-term system-performance measure or the strategic utility function $Q'(k) \in \mathfrak{R}^m$ is defined using the binary utility function as [24]

$$\begin{aligned} Q'(k) &= \alpha^N p(k+1) + \alpha^{N-1} p(k+2) + \dots \\ &\quad + \alpha^{k+1} p(N) + \dots \end{aligned} \quad (17)$$

where $\alpha \in \mathfrak{R}$ and $0 < \alpha < 1$, and N is the horizon. Equation (17) can also be expressed as $Q(k) = \min_{u(k)} \{\alpha Q(k-1) - \alpha^{N+1} p(k)\}$. This measure is similar to the standard Bellman equation [16], [20].

B. Critic NN

The critic NN is used to approximate the strategic utility function $Q'(k)$. We define the prediction error [20] as

$$e_c(k) = \hat{Q}(k) - \alpha \left(\hat{Q}(k-1) - \alpha^N p(k) \right) \quad (18)$$

where

$$\hat{Q}(k) = \hat{w}_1^T(k) \phi_1(v_1^T x(k)) = \hat{w}_1^T(k) \phi_1(k) \quad (19)$$

and $e_c(k) \in \mathfrak{R}^m$, subscript “c” stands for critic NN, $\hat{Q}(k) \in \mathfrak{R}^m$ is the critic signal, $\hat{w}_1(k) \in \mathfrak{R}^{n_1 \times m}$ and $v_1 \in \mathfrak{R}^{nm \times n_1}$ represent the matrix of weight estimates, $\phi_1(k) \in \mathfrak{R}^{n_1}$ is the activation function vector in the hidden layer, n_1 is the number of the nodes in the hidden layer, and the critic NN input is given by $x(k) \in \mathfrak{R}^{nm}$. The objective function to be minimized by the critic NN is defined as

$$E_c(k) = \frac{1}{2} e_c^T(k) e_c(k). \quad (20)$$

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

$$\hat{w}_1(k+1) = \hat{w}_1(k) + \Delta \hat{w}_1(k) \quad (21)$$

where

$$\Delta \hat{w}_1(k) = \alpha_1 \left[-\frac{\partial E_c(k)}{\partial \hat{w}_1(k)} \right] \quad (22)$$

or

$$\begin{aligned} \hat{w}_1(k+1) &= \hat{w}_1(k) - \alpha_1 \phi_1(k) \\ &\times \left(\hat{w}_1^T(k) \phi_1(k) + \alpha^{N+1} p(k) - \alpha \hat{w}_1^T(k-1) \phi_1(k-1) \right)^T \end{aligned} \quad (23)$$

where $\alpha_1 \in \mathfrak{R}$ is the NN adaptation gain. The critic NN weights are tuned by the reinforcement learning signal and discounted values of critic NN past outputs.

C. Action NN

The output of the action NN is to approximate the unknown nonlinear function $f(x(k))$ and to provide an optimal control signal to be part of the overall input $u(k)$ as

$$\hat{f}(k) = \hat{w}_2^T(k) \phi_2(v_2^T x(k)) = \hat{w}_2^T(k) \phi_2(k) \quad (24)$$

where $\hat{w}_2(k) \in \mathfrak{R}^{n_2 \times m}$ and $v_2 \in \mathfrak{R}^{nm \times n_2}$ represent the matrix of weight estimate, $\phi_2(k) \in \mathfrak{R}^{n_2}$ is the activation function in the hidden layer, n_2 is the number of nodes in the hidden layer, and $x(k) \in \mathfrak{R}^{nm}$ is the input to the critic NN.

Suppose that the unknown target output-layer weight for the action NN is w_2 ; then we have

$$f(k) = w_2^T \phi_2(v_2^T x(k)) + \varepsilon_2(x(k)) = w_2^T(k) \phi_2(k) + \varepsilon_2(x(k)) \quad (25)$$

where $\varepsilon_2(x(k)) \in \mathfrak{R}^m$ is the NN approximation error. Combining (24) and (25), we get

$$\tilde{f}(k) = \hat{f}(k) - f(k) = (\hat{w}_2(k) - w_2)^T \phi_2(k) - \varepsilon_2(x(k)) \quad (26)$$

where $\tilde{f}(k) \in \mathfrak{R}^m$ is the functional estimation error. The action NN weights are tuned by using the functional estimation error $\tilde{f}(k)$ and the error between the desired strategic utility function $Q_d(k) \in \mathfrak{R}^m$ and the critic signal $\hat{Q}(k)$. Define

$$e_a(k) = \tilde{f}(k) + (\hat{Q}(k) - Q_d(k)) \quad (27)$$

where $e_a(k) \in \mathfrak{R}^m$, with subscript “a” standing for the action NN.

Our desired value for the utility function $Q_d(k)$ is “0” [20], i.e., at every step, then the nonlinear system can track the reference signal well. Thus, (27) becomes

$$e_a(k) = \tilde{f}(k) + \hat{Q}(k). \quad (28)$$

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

$$E_a(k) = \frac{1}{2} e_a^T(k) e_a(k). \quad (29)$$

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

$$\hat{w}_2(k+1) = \hat{w}_2(k) + \Delta \hat{w}_2(k) \quad (30)$$

where

$$\Delta \hat{w}_2(k) = \alpha_2 \left[-\frac{\partial E_a(k)}{\partial \hat{w}_2(k)} \right] \quad (31)$$

or

$$\hat{w}_2(k+1) = \hat{w}_2(k) - \alpha_2 \phi_2(k) \left(\hat{Q}(k) + \tilde{f}(k) \right)^T \quad (32)$$

where $\alpha_2 \in \mathfrak{R}$ is the NN adaptation gain.

The NN weight updating rule in (32) cannot be implemented in practice since the nonlinear function $f(x(k))$ is unknown. However, using (9), the functional estimation error is given by

$$\tilde{f}(x(k)) = l_\nu r(k) - r(k+1) + d(k). \quad (33)$$

Substituting (33) into (32), we get

$$\begin{aligned} \hat{w}_2(k+1) &= \hat{w}_2(k) - \alpha_2 \phi_2(k) \\ &\times \left(\hat{Q}(k) + l_\nu r(k) - r(k+1) + d(k) \right)^T. \end{aligned} \quad (34)$$

the control parameters and adaptation gains. Their selection is given by the direct Lyapunov method.

Theorem 3.1: Let the desired trajectory $x_{nd}(k)$ and its past values be bounded. In addition, let Assumptions 1 and 2 hold, and the disturbance bound d_m is a known constant. Let the critic NN weight tuning be given by (23) and the action NN weight tuning be provided by (35). Then, the filtered tracking error $r(k)$ and the NN weight estimates $\hat{w}_1(k)$ and $\hat{w}_2(k)$ are UUB, with the bounds specifically given by (A15)–(A17) provided that the controller design parameters are selected as

$$(a) \quad \alpha_1 \|\phi_1(k)\|^2 < 1 \quad (40)$$

$$(b) \quad \alpha_2 \|\phi_2(k)\|^2 < 1 \quad (41)$$

$$(c) \quad 0 < \alpha < \frac{\sqrt{2}}{2} \quad (42)$$

$$(d) \quad 0 < l_{v \max} < \frac{\sqrt{3}}{3} \quad (43)$$

where $l_{v \max} \in \mathfrak{R}$ is the maximum singular value of the gain matrix, l_v .

Proof: See the Appendix. ■

Remark 2: It is important to note that in this theorem, there is no CE and LIP assumptions for the NN controller, in contrast to standard work in discrete-time adaptive control [1]. In the latter one, a parameter identifier is first designed, and the parameter estimation errors are shown to converge to small values by using a Lyapunov function. Then, in the tracking proof, it is assumed that the parameter estimates are exact by invoking a CE assumption, and another Lyapunov function that weighs only the tracking error terms is selected to demonstrate the closed-loop stability and tracking performance. In contrast to our proof, the Lyapunov function shown in the Appendix is of the form of (A1), which weighs the filtered tracking errors and the NN weight estimation errors $\tilde{w}_1(k)$ and $\tilde{w}_2(k)$. The proof is exceedingly complex due to the presence of several different variables. However, it obviates the need for the CE assumption, and it allows weight-tuning algorithms to be derived during the proof by minimizing certain quadratic objective functions, which were not selected *a priori* in an *ad hoc* manner.

Remark 3: The NN weight updating rules (23) and (35) are much simpler than that in [10] since they do not include an extra discrete-time ε -modification term [10], which is normally used to provide robustness due to the coupling in the proof between the tracking errors and NN weight estimation error terms. The Lyapunov proof demonstrates that the persistence-of-excitation condition is relaxed without the additional term in the weight tuning.

Remark 4: Both NN weight tuning rules (23) and (35) are updated online, in contrast to the offline training in previous works.

Remark 5: Condition (40) can be verified easily. For instance, if the hidden layer of the critic NN consists of n_1 nodes, with the hyperbolic tangent sigmoid function as its activation function, then $\|\phi_1(\cdot)\|^2 \leq n_1$. The NN adaptation gain α_1 can be selected as $0 < \alpha_1 < 1/n_1$ to satisfy (40). Similar analysis can be performed to obtain the NN adaptation gain α_2 .

Remark 6: Controller parameter $l_{v \max}$ and parameter α have to be selected using (42) and (43) in order for the closed-loop

system to be stable. This outer-loop signal is viewed as the supervisor's evaluation feedback to the actor and the critic.

Remark 7: The weights of the action-generating and critic NNs can be initialized at zero, and stability will be maintained by the outer-loop conventional controller until the NNs learn. This means that there is no explicit offline learning phase needed.

Remark 8: To the best of our knowledge, there is no information currently available to decide the number of hidden-layer neurons for the NN structure. However, 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 (40) and (41) and Remark 5, to make the closed-loop system stable, the numbers of hidden-layer nodes can be selected as $n_1 < (1/\alpha_1)$ and $n_2 < (1/\alpha_2)$ once the NN adaptation gains α_1 and α_2 are selected. However, in order to get better approximation performance and according to [6], the hidden-layer nodes have to be selected to be large enough to make the approximation error $\varepsilon(k)$ approach zero. To balance stability and good approximation requirements, we start with a small number of nodes and increase it until the controller achieves satisfactory performance.

The adaptive critic NN controller does not include the saturation constraint for the control input. To embed the input constraints as saturation nonlinearity in the controller structure, an auxiliary linear system [8] is introduced, and the stability of the closed system is demonstrated as given next.

IV. ADAPTIVE NN DESIGN WITH SATURATION NONLINEARITY

A. Design of the Auxiliary Linear System

Define the auxiliary control input $v(k)$ as

$$v(k) = x_{nd}(k+1) - \hat{f}(x(k)) + l_v r(k) - \lambda_1 e_n(k) - \dots - \lambda_{n-1} e_2(k) \quad (44)$$

where $\hat{f}(x(k))$ is an estimate of the unknown function $f(x(k))$ and $l_v \in \mathfrak{R}^{m \times m}$ is a diagonal gain matrix. The actual control input after the incorporation of saturation constraints is selected as

$$u(k) = \begin{cases} v(k), & \text{if } \|v(k)\| \leq u_{\max} \\ u_{\max} \text{sgn}(v(k)), & \text{if } \|v(k)\| > u_{\max} \end{cases} \quad (45)$$

where $u_{\max} \in \mathfrak{R}$ is the upper bound for the control input $u(k)$. Then, the closed-loop system becomes

$$r(k+1) = l_v r(k) - \tilde{f}(x(k)) + d(k) + \Delta u(k) \quad (46)$$

where the functional estimation error is given by $\tilde{f}(x(k)) = \hat{f}(x(k)) - f(x(k))$ and $\Delta u(k) = u(k) - v(k)$. To remove the effect of $\Delta u(k) \in \mathfrak{R}^m$, which can be considered as a disturbance, we generate a signal $e_{\Delta}(k) \in \mathfrak{R}^m$ as the output of a difference equation

$$e_{\Delta}(k+1) = l_v e_{\Delta}(k) + \Delta u(k) \quad (47)$$

with $e_{\Delta}(k_0) = 0$, where k_0 is the starting time instant.

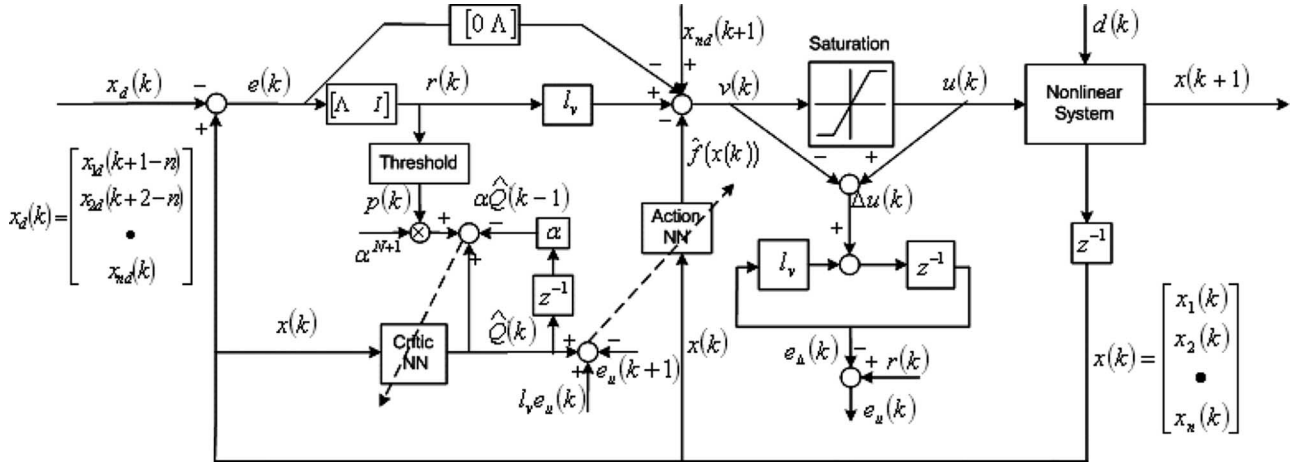


Fig. 2. Adaptive critic NN controller structure with input constraints.

Define now

$$e_u(k) = r(k) - e_{\Delta}(k) \quad (48)$$

we get

$$e_u(k+1) = l_v e_u(k) - \tilde{f}(x(k)) + d(k). \quad (49)$$

The auxiliary linear error system given by (47) is aimed at proving the stability of the filtered tracking error $r(k)$ and taking care of the effect of Δu . In the remainder of this paper, (49) is used to focus on selecting NN tuning algorithms that guarantee the stability of the auxiliary error $e_u(k)$. Once $e_u(k)$ is proven stable, it is required to show that the filtered tracking error system $r(k)$ is stable.

B. Adaptive Critic NN Controller Structure With Saturation

The critic NN $\hat{w}_1^T(k)\phi_1(k)$ design is the same as that of Section III. The action NN $\hat{w}_3^T(k)\phi_3(k)$ is also similar to that in Section III, except that an auxiliary error signal $e_u(k)$ is now used, instead of the filtered tracking error $r(k)$, to accommodate the input constraints. The procedure of obtaining the action NN weight update is very similar to that in Section III, and it is given by

$$\hat{w}_3(k+1) = \hat{w}_3(k) - \alpha_3 \phi_3(k) \times \left(\hat{Q}(k) + l_v e_u(k) - e_u(k+1) \right)^T \quad (50)$$

where $\hat{w}_3(k) \in \mathbb{R}^{n_3 \times m}$ represents the matrix of the weight estimate, $\phi_3(k) \in \mathbb{R}^{n_3}$ is the activation function in the hidden layer, n_3 is the number of the nodes in the hidden layer, and $x(k) \in \mathbb{R}^{nm}$ is the input to the critic NN. Here, the auxiliary signal is utilized, instead of the tracking error.

C. Closed-Loop System Stability Analysis

Assumption 3 (Bounded Ideal Weight): Let w_3 be the unknown output-layer target NN weight for the action NN, and assume that it is bounded above so that

$$\|w_3\| \leq w_{3m} \quad (51)$$

where $w_{3m} \in \mathbb{R}^+$ is the maximum bound on the unknown weight. Then, the error in weight during estimation is given by

$$\tilde{w}_3(k) = \hat{w}_3(k) - w_3. \quad (52)$$

Fact 2: The activation function is bounded by known positive value so that

$$\|\phi_3(k)\| \leq \phi_{3m} \quad (53)$$

where $\phi_{3m} \in \mathbb{R}$ is the upper bound for $\phi_3(k)$.

Let the auxiliary control input $v(k)$ be selected by (44) and actual control input be chosen as (45); the auxiliary error system is given as

$$e_u(k+1) = l_v e_u(k) - \zeta_3(k) + \varepsilon_3(x(k)) + d(k) \quad (54)$$

where $\zeta_3(k)$ is defined by

$$\zeta_3(k) = \tilde{w}_3^T(k)\phi_3(k) \quad (55)$$

and $\varepsilon_3(x(k)) \in \mathbb{R}^m$ is the NN approximation error.

Assumption 4 (Bounded NN Approximation Error): The NN approximation error $\varepsilon_3(x(k))$ is bounded over the compact set S by ε_{3m} .

The structure of the proposed adaptive critic NN controller with magnitude constraints is shown in Fig. 2, in contrast with Fig. 1. The next theorem presents how to select the controller parameters and the adaptation gains to ensure that the performance of the closed-loop system is guaranteed and that all the internal signals are UUB.

Theorem 4.1: Consider the system given in (4) and the control input given by (45). Let the hypotheses presented in Theorem 3.1 and Assumptions 3 and 4 hold. Let the critic NN $\hat{w}_1^T(k)\phi_1(k)$ weight tuning be (23) and the action-generating NN $\hat{w}_3^T(k)\phi_3(k)$ weight tuning be provided by (50). Then, the auxiliary error $e_u(k)$ and the NN weight estimates $\hat{w}_1(k)$ and $\hat{w}_3(k)$ are UUB, with the bounds specifically given by (A19)–(A21) provided that the design parameters are selected as (40), (42), (43), and

$$\alpha_3 \|\phi_3(k)\|^2 < 1. \quad (56)$$

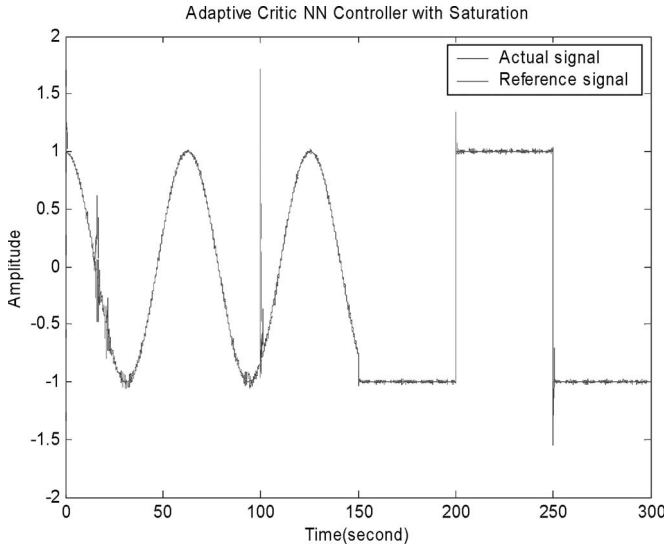


Fig. 3. Performance of the NN controller.

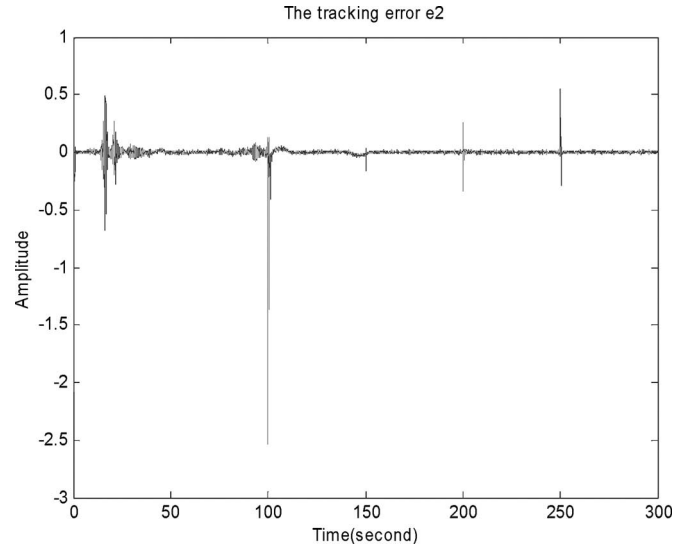


Fig. 4. Tracking error.

Proof: See the Appendix. ■

Remark 9: From Theorem 4.1, the UUB of the filtered tracking error $r(k)$ is derived in the Appendix.

Remark 10: The critic NN weight tuning is performed now using the auxiliary error signal, which is obtained from the filtered tracking error and the output of the linear system that is driven by $\Delta u(k)$.

V. SIMULATION

The nonlinear system is described by

$$\begin{aligned} x_1(k+1) &= x_2(k) \\ x_2(k+1) &= f(x(k)) + u(k) + d(k) \end{aligned} \quad (57)$$

where

$$f(x(k)) = -\frac{5}{8} \left[\frac{x_1(k)}{1+x_2^2(k)} \right] + 0.3x_2(k).$$

The objective is to make the state $x_2(k)$ track a reference signal using the proposed adaptive critic NN controller with input saturation. The reference signal used was selected as

$$x_{2d} = \begin{cases} \sin(\omega kT + \xi), & 0 \leq k \leq 3000 \\ -1, & 3000 < k \leq 4000 \\ & \text{or } 5000 < k \leq 6000 \\ 1, & 4000 < k \leq 5000 \end{cases} \quad (58)$$

where the desired signal is a sine wave and a unit step signal. The two different reference signals are used to evaluate the learning ability of the adaptive critic NN controller.

The sampling interval T is taken as 50 ms, and the white Gaussian noise with a standard deviation of 0.005 is added to the system. The time duration is taken to be 300 s. The unknown disturbance is taken as

$$d(k) = \begin{cases} 0, & k < 2000 \\ 1.5, & 2000 \leq k \leq 6000. \end{cases} \quad (59)$$

The gain of the PD controller is selected as $l_v = 0.1$, with $\lambda = 0.2$. The actuator limit for the control signal is set at 3.0 and $c = 0.0025$. Both critic NN $\hat{w}_1^T \phi_1(k)$ and action NN $\hat{w}_3^T \phi_3(k)$ contain ten nodes in the hidden layer. For weight updating, the gains are selected as $\alpha_1 = \alpha_3 = 0.1$ and $\alpha = 0.5$. The initial weights are selected at random from $[0, 1]$, and hyperbolic tangent sigmoid functions are employed. The initial states are set at zero.

Fig. 3 illustrates the good tracking performance of the proposed adaptive critic NN controller with input saturation. The transient observed during the initial phase of the simulation is the result of no offline learning, and the NN is trying to learn the unknown dynamics online. However, within a very short time, the NN learns, as demonstrated in the simulation. When we introduce the unknown but bounded disturbance in the system, a large spike, which is indicative of the disturbance, is observed. The tracking error quickly converges close to zero after the application of the disturbance, which indicates that the controller has good disturbance rejection. This phenomenon also demonstrates the good learning ability of the proposed adaptive critic NN controller. The subtle chattering observed in the tracking error shown in Fig. 4 is due to the presence of an unknown white noise. In fact, the tracking error is close to zero, except at some points where the reference signal is discontinuous. Fig. 5 presents the norm of the output-layer weights, where the weights are bounded. Fig. 6 shows the associated NN control input, where it is bounded by a magnitude of three.

To show the contribution of the NNs in the controller, the NN inner loop (Fig. 2) is removed, and the outer loop is kept, which leads to a proportional and derivative (PD) type conventional controller. The controller parameters were not altered in both cases. From Figs. 7 and 8, it is clear that the tracking performance has deteriorated even though the tracking error is bounded. This clearly demonstrates that the NNs are able to compensate the unknown dynamics by providing an additional signal. Moreover, the outer-loop PD controller provides a stable system initially when the NN begins to learn. The PD control input is depicted in Fig. 9, where it is bounded as expected.

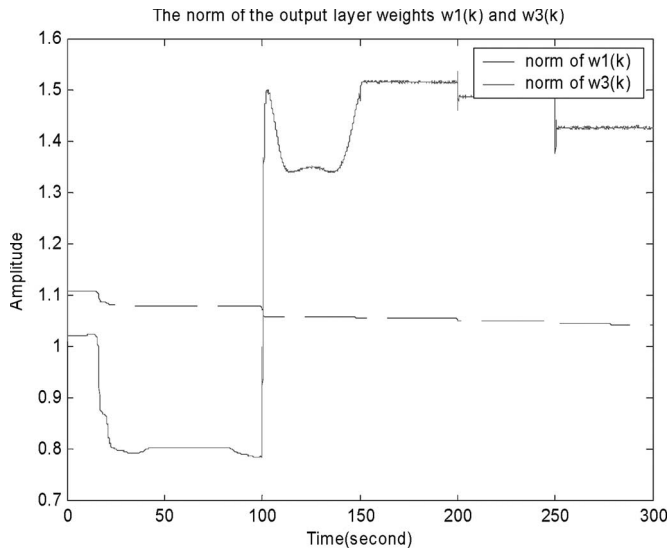


Fig. 5. Norm of the output-layer weights of both NNs.

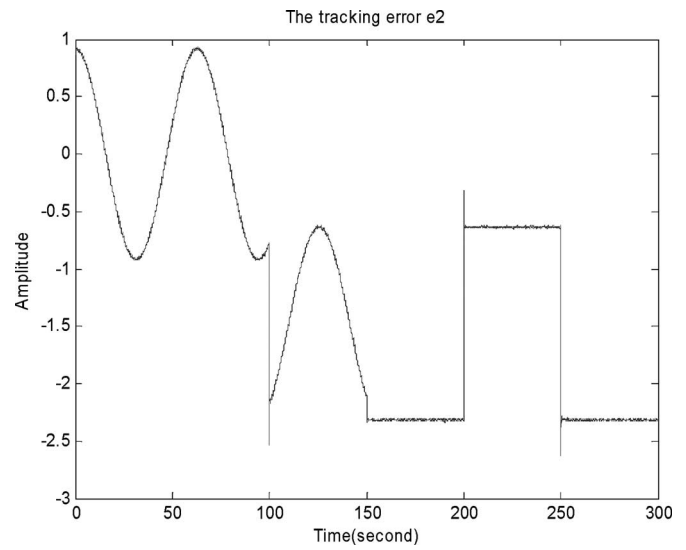


Fig. 8. Tracking error.

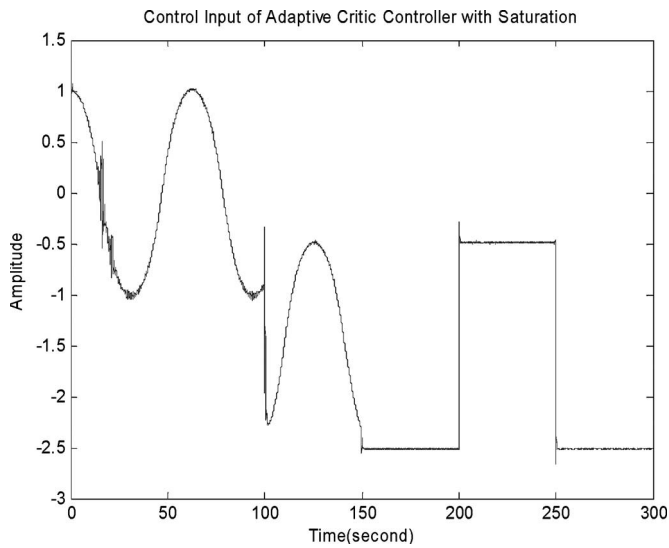


Fig. 6. Control input.

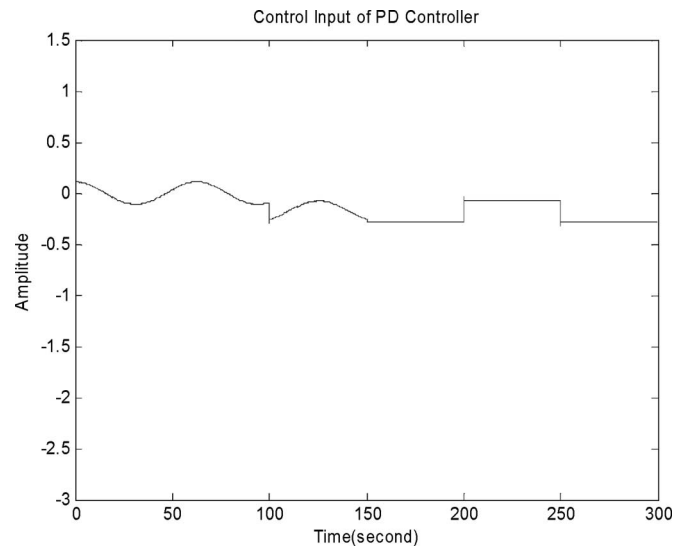


Fig. 9. PD control input.

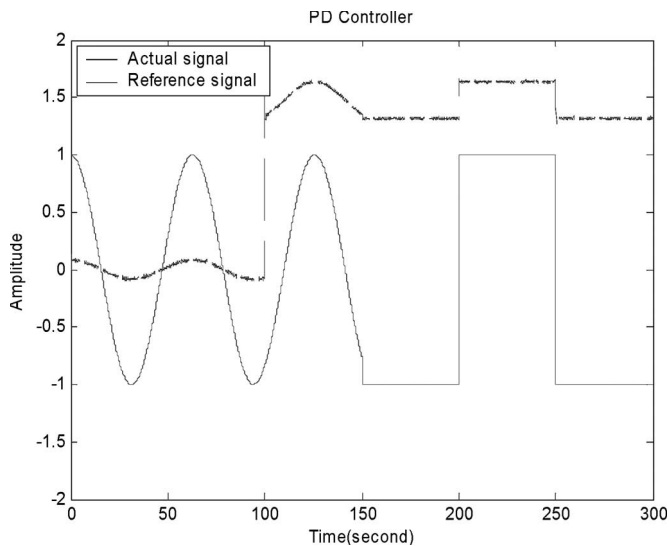


Fig. 7. Performance of the PD controller.

VI. CONCLUSION

This paper proposes an adaptive NN-based controller using reinforcement learning for a class of nonlinear systems in the presence of magnitude constraints represented as saturation nonlinearity. This adaptive NN-based approach does not require information about the system dynamics. The adaptive critic NN controller includes an action NN for compensating the unknown dynamics, a critic signal for approximating the strategic utility function, and an outer PD control loop. The tuning of the action-generating NN is performed online without an explicit offline learning phase. The outer-loop conventional signal can be viewed as the supervisor’s evaluation feedback signal, which will allow the tuning of NN weights online rather than offline training, which is usually deployed in the adaptive actor-critic NN architecture.

The input magnitude constraint is modeled as saturation nonlinearity, and it was treated by converting the nonlinearity into an input disturbance, which was suitably accommodated by the

adaptive weight tuning law. The proposed adaptive critic NN controller was applied to a nonlinear system with and without saturation, and the controller performance was demonstrated. Results demonstrate that the Lyapunov-based adaptive critic design renders satisfactory performance while ensuring closed-loop stability.

APPENDIX

Proof of Theorem 3.1: Define the Lyapunov function candidate

$$J(k) = \frac{1}{\gamma_1} r^T(k)r(k) + \frac{1}{\alpha_1} \text{tr}(\tilde{w}_1^T(k)\tilde{w}_1(k)) + \frac{1}{\gamma_2} \|\zeta_1(k-1)\|^2 + \frac{1}{\gamma_3 \alpha_2} \text{tr}(\tilde{w}_2^T(k)\tilde{w}_2(k)) \quad (\text{A1})$$

where $\zeta_1(k-1) = (\hat{w}_1(k-1) - w_1)^T \phi_1(k-1) = \tilde{w}_1^T(k-1)\phi_1(k-1)$ and $0 < \gamma_i, i = 1, 2, 3$. The first difference of the Lyapunov function is calculated as

$$\Delta J(k) = \Delta J_1(k) + \Delta J_2(k) + \Delta J_3(k) + \Delta J_4(k). \quad (\text{A2})$$

$\Delta J_1(k)$ is obtained using the filtered tracking error dynamics (39) as

$$\begin{aligned} \Delta J_1(k) &= \frac{1}{\gamma_1} (r^T(k+1)r(k+1) - r^T(k)r(k)) \\ &= \frac{1}{\gamma_1} \left((l_v r(k) - \zeta_2(k) + \varepsilon_2(x(k)) + d(k))^T \right. \\ &\quad \times (l_v r(k) - \zeta_2(k) + \varepsilon_2(x(k)) + d(k)) \\ &\quad \left. - r^T(k)r(k) \right) \\ &\leq \frac{3}{\gamma_1} \left(\left(l_{v \max}^2 - \frac{1}{3} \right) \|r(k)\|^2 \right. \\ &\quad \left. + \|\zeta_2(k)\|^2 + \|\varepsilon_2(k) + d(k)\|^2 \right) \end{aligned} \quad (\text{A3})$$

where $l_{v \max} \in R$ is the maximum eigenvalue of matrix $l_v \in R^{m \times m}$. Now, take the second term in the first difference of (A2) and rewrite it as

$$\Delta J_2(k) = \frac{1}{\alpha_1} \text{tr}[\tilde{w}_1^T(k+1)\tilde{w}_1(k+1) - \tilde{w}_1^T(k)\tilde{w}_1(k)]. \quad (\text{A4})$$

Substituting the NN weight updates from (23) yields

$$\begin{aligned} \tilde{w}_1(k+1) &= (I - \alpha_1 \phi_1(k)\phi_1^T(k)) \tilde{w}_1(k) - \alpha_1 \phi_1(k) \\ &\quad \times (w_1^T(k)\phi_1(k) + \alpha^{N+1}p(k) \\ &\quad - \alpha \hat{w}_1^T(k-1)\phi_1(k-1))^T. \end{aligned} \quad (\text{A5})$$

Now, substituting (A5) into (A4) and combining them, we get

$$\begin{aligned} \Delta J_2(k) &\leq - (1 - \alpha_1 \phi_1^T(k)\phi_1(k)) \\ &\quad \times \|\zeta_1(k) + w_1^T(k)\phi_1(k) + \alpha^{N+1}p(k) - \alpha \hat{w}_1^T(k-1)\phi_1(k-1)\|^2 \\ &\quad - \|\zeta_1(k)\|^2 \\ &\quad + 2 \|w_1^T(k)\phi_1(k) + \alpha^{N+1}p(k) - \alpha \hat{w}_1^T(k-1)\phi_1(k-1)\|^2 \\ &\quad + 2\alpha^2 \|\zeta_1(k-1)\|^2. \end{aligned} \quad (\text{A6})$$

Now, taking the third term in (A2), we get

$$\Delta J_3(k) = \frac{1}{\gamma_2} \left(\|\zeta_1(k)\|^2 - \|\zeta_1(k-1)\|^2 \right). \quad (\text{A7})$$

The fourth term in (A2) is expanded as

$$\Delta J_4(k) = \frac{1}{\gamma_3 \alpha_2} \text{tr}[\tilde{w}_2^T(k+1)\tilde{w}_2(k+1) - \tilde{w}_2^T(k)\tilde{w}_2(k)]. \quad (\text{A8})$$

Substituting the weight updates for the NN (35) and simplifying it, we get

$$\begin{aligned} \Delta J_4(k) &\leq \frac{1}{\gamma_3} \left\{ - (1 - \alpha_2 \phi_2^T(k)\phi_2(k)) \right. \\ &\quad \times \|\zeta_2(k) + \hat{w}_1^T(k)\phi_1(k) \\ &\quad \left. - (\varepsilon_2(x(k)) + d(k))\|^2 - \|\zeta_2(k)\|^2 \right\} \\ &\quad + \frac{2}{\gamma_3} \left\{ \|w_1^T(k)\phi_1(k) - (\varepsilon_2(x(k)) + d(k))\|^2 \right. \\ &\quad \left. + \|\zeta_1(k)\|^2 \right\}. \end{aligned} \quad (\text{A9})$$

Combining (A3), (A6), (A7), and (A9) to get the first difference of the Lyapunov (A2), we get

$$\begin{aligned} \Delta J(k) &\leq \frac{-1}{\gamma_1} (1 - 3l_{v \max}^2) \|r(k)\|^2 - \left(1 - \frac{1}{\gamma_2} - \frac{2}{\gamma_3}\right) \|\zeta_1(k)\|^2 \\ &\quad - \left(\frac{1}{\gamma_3} - \frac{3}{\gamma_1}\right) \|\zeta_2(k)\|^2 - \left(\frac{1}{\gamma_2} - 2\alpha^2\right) \|\zeta_1(k-1)\|^2 \\ &\quad - (1 - \alpha_1 \phi_1^T(k)\phi_1(k)) \\ &\quad \times \|\zeta_1(k) + w_1^T(k)\phi_1(k) + \alpha^{N+1}p(k) \\ &\quad - \alpha \hat{w}_1^T(k-1)\phi_1(k-1)\|^2 \\ &\quad - \frac{1}{\gamma_3} \left\{ (1 - \alpha_2 \phi_2^T(k)\phi_2(k)) \right. \\ &\quad \left. \times \|\zeta_2(k) + \hat{w}_1^T(k)\phi_1(k) - (\varepsilon_2(x(k)) + d(k))\|^2 \right\} \\ &\quad + 2 \|w_1^T(k)\phi_1(k) + \alpha^{N+1}p(k) - \alpha w_1^T(k)\phi_1(k-1)\|^2 \\ &\quad + \frac{2}{\gamma_3} \left\{ \|w_1^T(k)\phi_1(k) - (\varepsilon_2(x(k)) + d(k))\|^2 \right\} \\ &\quad + \frac{3}{\gamma_1} \|\varepsilon_2(k) + d(k)\|^2. \end{aligned} \quad (\text{A10})$$

Choose

$$\begin{cases} \gamma_1 > 3\gamma_3 \\ \gamma_2 = \frac{\sqrt{2}}{2}\alpha \\ \gamma_3 > \frac{2}{1-2\alpha^2} \end{cases} \quad (\text{A11})$$

and define

$$\begin{aligned} D^2 = & 2 \left\| w_1^T(k) \phi_1(k) + \alpha^{N+1} p(k) - \alpha w_1^T(k-1) \phi_1(k-1) \right\|^2 \\ & + \frac{2}{\gamma_3} \left\{ \left\| w_1^T(k) \phi_1(k) - (\varepsilon_2(x(k)) + d(k)) \right\|^2 \right\} \\ & + \frac{3}{\gamma_1} \left\| \varepsilon_2(k) + d(k) \right\|^2. \end{aligned} \quad (\text{A12})$$

The upper bound D_m for D is

$$\begin{aligned} D^2 \leq D_m^2 = & 6 \left(1 + \alpha^2 + \frac{1}{\gamma_3} \right) w_{1m}^2 \phi_{1m}^2 \\ & + 6 \left(\frac{1}{\gamma_1} + \frac{1}{\gamma_3} \right) (\varepsilon_{2m}^2 + d_m^2). \end{aligned} \quad (\text{A13})$$

Using (A11) and (A12) to rewrite (A10), we get

$$\begin{aligned} \Delta J(k) \leq & \frac{-1}{\gamma_1} (1 - 3l_{v\max}^2) \|r(k)\|^2 - \left(1 - \frac{1}{\gamma_2} - \frac{2}{\gamma_3} \right) \|\zeta_1(k)\|^2 \\ & - \left(\frac{1}{\gamma_3} - \frac{3}{\gamma_1} \right) \|\zeta_2(k)\|^2 - (1 - \alpha_1 \phi_1^T(k) \phi_1(k)) \\ & \times \left\| \zeta_1(k) + w_1^T(k) \phi_1(k) + \alpha^{N+1} p(k) \right. \\ & \quad \left. - \alpha \hat{w}_1^T(k-1) \phi_1(k-1) \right\|^2 \\ & - \frac{1}{\gamma_3} \left\{ (1 - \alpha_2 \phi_2^T(k) \phi_2(k)) \right. \\ & \quad \times \left\| \zeta_2(k) + \hat{w}_1^T(k) \phi_1(k) \right. \\ & \quad \left. \left. - (\varepsilon_2(x(k)) + d(k)) \right\|^2 \right\} + D^2. \end{aligned} \quad (\text{A14})$$

This further implies that the first difference $\Delta J(k) \leq 0$ as long as (40)–(43) hold and

$$\|r(k)\| > \sqrt{\frac{\gamma_1}{1 - 3l_{v\max}^2}} D_m \quad (\text{A15})$$

or

$$\|\zeta_1(k)\| > \frac{D_m}{\sqrt{1 - \frac{1}{\gamma_2} - \frac{2}{\gamma_3}}} \quad (\text{A16})$$

or

$$\|\zeta_2(k)\| > \frac{D_m}{\sqrt{\frac{1}{\gamma_3} - \frac{3}{\gamma_1}}}. \quad (\text{A17})$$

According to a standard Lyapunov extension theorem [25], this demonstrates that the filtered tracking error and the error in weight estimates are UUB. The boundedness of $\|\zeta_1(k)\|$ and $\|\zeta_2(k)\|$ implies that $\|\hat{w}_1(k)\|$ and $\|\hat{w}_2(k)\|$ are bounded, and this further implies that the weight estimates $\hat{w}_1(k)$ and $\hat{w}_2(k)$ are bounded.

Note: Condition (A11) is easy to check. For instance, we could choose $\alpha = 1/2$, $\gamma_1 = 16$, $\gamma_2 = \sqrt{2}/4$, and $\gamma_3 = 5$ to satisfy (A11). ■

Proof of Theorem 4.1: Define the Lyapunov function candidate as

$$\begin{aligned} J(k) = & \frac{1}{\gamma_1} e_u^T(k) e_u(k) + \frac{1}{\alpha_1} \text{tr}(\tilde{w}_1^T(k) \tilde{w}_1(k)) \\ & + \frac{1}{\gamma_2} \|\zeta_1(k)\|^2 + \frac{1}{\gamma_3 \alpha_3} \text{tr}(\tilde{w}_3^T(k) \tilde{w}_3(k)). \end{aligned} \quad (\text{A18})$$

The proof is similar to that of Theorem 3.1, so it is omitted. The first difference $\Delta J(k) \leq 0$ as long as (40), (42), (43), (56), and (A11) are satisfied and

$$\|e_u(k)\| > \sqrt{\frac{\gamma_1}{1 - 3l_{v\max}^2}} D_m \quad (\text{A19})$$

or

$$\|\zeta_1(k)\| > \frac{D_m}{\sqrt{1 - \frac{1}{\gamma_2} - \frac{2}{\gamma_3}}} \quad (\text{A20})$$

or

$$\|\zeta_3(k)\| > \frac{D_m}{\sqrt{\frac{1}{\gamma_3} - \frac{3}{\gamma_1}}} \quad (\text{A21})$$

where

$$\zeta_3(k) = (\hat{w}_3(k) - w_3)^T \phi_3(k) = \tilde{w}_3^T(k) \phi_3(k). \quad (\text{A22})$$

According to a standard Lyapunov extension theorem [25], this demonstrates that the auxiliary error and the error in weight estimates are UUB. The boundedness of $\|\zeta_1(k)\|$ and $\|\zeta_3(k)\|$ implies that $\|\tilde{w}_1(k)\|$ and $\|\tilde{w}_3(k)\|$ are bounded, and this further implies that the weight estimates $\hat{w}_1(k)$ and $\hat{w}_3(k)$ are bounded.

The next step is to show the filtered tracking error $r(k)$ is bounded. Here, two cases are being discussed. The first is when $\|v(k)\| \leq u_{\max}$, and the second is when $\|v(k)\| > u_{\max}$.

Case 1: $\|v(k)\| \leq u_{\max}$

If $\|v(k)\| \leq u_{\max}$, then $u(k) = v(k)$. The closed-loop error system (31) becomes

$$r(k+1) = l_v r(k) - \zeta_2(k) + \varepsilon_2(x(k)) + d(k). \quad (\text{A23})$$

This is a linear system driven by function estimation error and disturbances. Since the disturbances are bounded and the weight estimation error is shown to be bounded above, the filtered tracking error system is driven by bounded inputs. Therefore, the filtered tracking error is bounded; hence, all the tracking errors are bounded.

Case 2: $\|v(k)\| > u_{\max}$

If $\|v(k)\| > u_{\max}$, then $u(k) = u_{\max} \text{sgn}(v(k))$. For the non-linear system (4), the tracking error should be in the form of

$$\begin{aligned} e_n(k+1) &= x_n(k+1) - x_{\text{nd}}(k+1) \\ &= f(x(k)) + u_{\max} \text{sgn}(v(k)) + d(k) - x_{\text{nd}}(k+1). \end{aligned} \quad (\text{A24})$$

Over a compact set, the smooth function is bounded by F_{\max} , and the desired trajectory is bounded by $x_{d\max}$. Then, we obtain the upper bound of $e_n(k)$, i.e.,

$$\|e_n(k)\| \leq F_{\max} + u_{\max} + d_M + x_{d\max}. \quad (\text{A25})$$

Based on the definition of the filtered tracking error of (6) and $e_n(k)$ having an upper bound, in this case, the filtered tracking error is UUB. Considering cases 1 and 2, the proof of the UUB of the filtered tracking error is complete. ■

REFERENCES

- [1] K. J. Åström and B. Wittenmark, *Adaptive Control*. Reading, MA: Addison-Wesley, 1989.
- [2] A. G. Barto, "Reinforcement learning and adaptive critic methods," in *Handbook of Intelligent Control*, D. A. White and D. A. Sofge, Eds. New York: Van Nostrand, 1992, pp. 65–90.
- [3] D. P. Bertsekas and J. N. Tsitsiklis, *Neuro-Dynamic Programming*. Belmont, MA: Athena Scientific, 1996.
- [4] A. J. Calise, "Neural networks in nonlinear aircraft flight control," *IEEE Aerosp. Electron. Syst. Mag.*, vol. 11, no. 7, pp. 5–10, Jul. 1996.
- [5] F. C. Chen and H. K. Khalil, "Adaptive control of a class of nonlinear discrete-time systems using neural networks," *IEEE Trans. Autom. Control*, vol. 40, no. 5, pp. 791–801, May 1995.
- [6] B. Igel'nik 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.
- [7] I. Kanellakopoulos, "A discrete-time adaptive nonlinear system," *IEEE Trans. Autom. Control*, vol. 39, no. 11, pp. 2362–2365, Nov. 1994.
- [8] S. P. Karason and A. M. Annaswamy, "Adaptive control in the presence of input constraints," *IEEE Trans. Autom. Control*, vol. 39, no. 11, pp. 2325–2330, Nov. 1994.
- [9] M. Krstic, I. Kanellakopoulos, and P. Kokotovic, *Nonlinear and Adaptive Control Design*. Hoboken, NJ: Wiley, 1995.
- [10] F. L. Lewis, S. Jagannathan, and A. Yesilderek, *Neural Network Control of Robot Manipulators and Nonlinear Systems*. London, U.K.: Taylor & Francis, 1999.
- [11] X. Lin and S. N. Balakrishnan, "Convergence analysis of adaptive critic based optimal control," in *Proc. Amer. Control Conf.*, 2000, pp. 1929–1933.
- [12] J. J. Murray, C. Cox, G. G. Lendaris, and R. Saeks, "Adaptive dynamic programming," *IEEE Trans. Syst., Man, Cybern.*, vol. 32, no. 2, pp. 140–153, May 2002.
- [13] K. S. Narendra and A. M. Annaswamy, *Stable Adaptive Systems*. Englewood Cliffs, NJ: Prentice-Hall, 1989.
- [14] K. S. Narendra and K. S. Parthasarathy, "Identification and control of dynamical systems using neural networks," *IEEE Trans. Neural Netw.*, vol. 1, no. 1, pp. 4–27, Mar. 1990.
- [15] D. V. Prokhorov and L. A. Feldkamp, "Analyzing for Lyapunov stability with adaptive critics," in *Proc. IEEE Conf. Syst. Man and Cybern.*, San Diego, CA, 1998, pp. 1658–1661.
- [16] D. V. Prokhorov and D. C. Wunsch, "Adaptive critic designs," *IEEE Trans. Neural Netw.*, vol. 8, no. 5, pp. 997–1007, Sep. 1997.
- [17] S. S. Sastry and A. Isidori, "Adaptive control of linearizable systems," *IEEE Trans. Autom. Control*, vol. 34, no. 11, pp. 1123–1131, Nov. 1989.
- [18] S. Shervais, T. T. Shannon, and G. G. Lendaris, "Intelligent supply chain management using adaptive critic learning," *IEEE Trans. Syst., Man, Cybern.*, vol. 33, no. 2, pp. 235–244, Mar. 2003.
- [19] J. Si, in *Proc. NSF Workshop Learn. and Approx. Dyn. Program.*, Mexico, 2002. [Online]. Available: <http://ebrains.la.asu.edu/~nsfadp/>
- [20] 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.
- [21] P. J. Werbos, "Neurocontrol and supervised learning: An overview and evaluation," in *Handbook of Intelligent Control*, D. A. White and D. A. Sofge, Eds. New York: Van Nostrand, 1992, pp. 65–90.
- [22] ———, "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.
- [23] M. T. Rosenstein and A. G. Barto, "Supervised actor-critic reinforcement learning," in *Handbook of Learning and Approximate Dynamic Programming*, J. Si et al. Eds. Piscataway, NJ: IEEE Press, 2004, pp. 359–380.
- [24] P. He and S. Jagannathan, "Reinforcement-based neuro-output feedback control of discrete-time systems with input constraints," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 35, no. 1, pp. 150–154, Feb. 2005.
- [25] S. Jagannathan, *Neural Network Control of Nonlinear Discrete-Time Systems*. Boca Raton, FL: Taylor & Francis, 2006.



Pingan He received the Ph.D. degree in electrical engineering from the University of Missouri-Rolla in 2004.

He is currently a Controls Engineer with MotoTron Corporation, Oshkosh, WI, where he has developed an engine management system and diagnostics. His research interests include power-train control systems.

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



S. Jagannathan (M'95–SM'99) was born in Madurai, India. He received the B.S. degree from Guindy at Anna University, Chennai, India, in 1987, the M.S. degree from the University of Saskatchewan, Saskatoon, SK, Canada, in 1989, and the Ph.D. degree from the University of Texas, Arlington, in 1994, all in electrical engineering.

From 1986 to 1987, he was a Junior Engineer with Engineers India Ltd., New Delhi, India. From 1990 to 1991, he was a Research Associate and an Instructor with the University of Manitoba, Winnipeg, MB, Canada. From 1994 to 1998, he was a Program Director with the Systems and Controls Research Division, Caterpillar Inc., Peoria, IL. From 1998 to 2001, he was with the University of Texas at San Antonio. Since September 2001, he has been with the University of Missouri-Rolla, where he is currently a Professor. He is also the Site Director for the National Science Foundation Industry (NSF)/University Cooperative Research Center. He has coauthored more than 160 refereed conference and journal articles, several book chapters, and three text books entitled *Neural Network Control of Robot Manipulators and Nonlinear Systems* (Taylor & Francis, 1999), *Neural Network Control of Nonlinear Discrete-Time Systems* (CRC, 2006), and *Quality of Service Control in High-Speed and Wireless Networks* (Taylor & Francis, 2007). He currently holds 17 patents, with several others still in process. His research interests include adaptive and neural network control, microelectromechanical systems/nanotechnology, computer/communication/sensor networks, prognostics, and autonomous systems/robotics.

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