Reinforcement Learning-Based Output Feedback Control of Nonlinear Systems With Input Constraints

P. He and S. Jagannathan

***Abstract*—A novel neural network (NN)-based output feedback controller with magnitude constraints is designed to deliver a desired tracking performance for a class of multi-input and multi-output (MIMO) strict feedback nonlinear discrete-time systems. Reinforcement learning is proposed for the output feedback controller, which uses three NNs: 1) an NN observer to estimate the system states with the input-output data, 2) a critic NN to approximate certain *strategic* utility function, and 3) an action NN to minimize both the *strategic* utility function and the unknown dynamics estimation errors. Using the Lyapunov approach, the uniformly ultimate boundedness (UUB) of the state estimation errors, the tracking errors and weight estimates is shown.**

***Index Terms*—Neural networks (NNs), output feedback control, reinforcement learning.**

I. Introduction

The output feedback controller schemes are necessary when certain states of the plants become unavailable for measurement. However, the separation principle that is normally used for linear systems does not hold for nonlinear systems [1]. Consequently, the output feedback controller design is quite difficult and challenging. Several output feedback controller designs in discrete time are proposed for signal-input and single-out (SISO) nonlinear systems [2]–[4]. However, the SISO controller designs cannot be directly extended to the proposed multi-input and multi-output (MIMO) case.

In this paper, an output feedback controller design using the adaptive critic neural network (NN) architecture is considered for an unknown MIMO nonlinear discrete system. The motivation for using the reinforcement learning-based adaptive critic approach is mainly for optimal control [5]–[8]. The adaptive critic designs attempt to approximate dynamic programming in the general case [5]. The proposed adaptive critic output feedback NN controller consists of the following:

1) NN observer to estimate the system states with the input-output data;
2) action NN to drive the output to track the reference signal and to minimize both the *strategic* utility function and the unknown dynamics estimation errors;
3) adaptive critic NN to approximate the *strategic* utility function $Q(x(k))$ and to tune the weights of the action NN.

With incomplete information of the system states and dynamics, an approximate optimization is accomplished using the proposed controller. Further, the actuator constraints are incorporated as saturation nonlinearities during the controller development in contrast to other works [1]–[8] where no explicit magnitude constraints are treated. Besides optimization and the incorporation of input constraints, contributions of this paper can be summarized as follows:

1) demonstration of the UUB of the overall system is shown even in the presence of NN approximation errors and bounded unknown disturbances;
2) NN weights are tuned online instead of offline training;
3) persistent excitation (PE) condition requirement is overcome both in NN observer and controller designs.

II. Background

A. Nonlinear System Description

Consider the nonlinear system to be controlled, given in the following form:

$$\begin{aligned} x_1(k+1) &= x_2(k) \\ &\vdots \\ x_n(k+1) &= f\left(x(k)\right) + g\left(x(k)\right)u(k) + d'(k) \qquad (1) \\ y(k) &= x_1(k) \qquad (2) \end{aligned}$$

with state $x(k) = [x_1^T(k), x_2^T(k), \cdots, x_n^T(k)]^T \in R^{nm}$, and each $x_i(k) \in R^m$, $i = 1, \cdots, n$ is the state at time instant k, $f(x(k)) \in R^m$ is the unknown nonlinear function vector, $g(x(k)) \in R^{m\times m}$ is a diagonal matrix of unknown nonlinear functions, $u(k) \in R^m$ is the control input vector and $d'(k) \in R^m$ is the unknown but bounded disturbance vector, whose bound is assumed to be a known constant, $\|d'(k)\| \le d'_m$, where the Frobenius norm will be used through out this paper. It is assumed that the output, $y(k) \in R^m$, is known at the kth instant and the state vector $x_i(k) \in R^m$, $i = 2, \cdots, n$ is considered to be unavailable at the kth step.

Assumption 1: Let the diagonal matrix $g(x(k)) \in R^{m\times m}$ be a positive definite matrix for each $x(k) \in R^{nm}$, with $g_{\min} \in R^+$ and $g_{\max} \in R^+$ represent the minimum and maximum eigenvalues of the matrix $g(x(k)) \in R^{m\times m}$, respectively, such that $0 < g_{\min} < g_{\max}$.

III. NN Observer Design

A. Observer Structure

For the system described by (1) and (2), we use the following state observer to estimate the state $x(k)$:

$$\begin{aligned} \hat{x}_1(k) &= \hat{x}_2(k-1) \\ &\vdots \\ \hat{x}_n(k) &= \hat{w}_1^T(k-1)\phi_1\left(v_1^T\hat{z}_1(k-1)\right) \\ &= \hat{w}_1^T(k-1)\phi_1\left(\hat{z}_1(k-1)\right) \qquad (3) \end{aligned}$$

where $\hat{x}_i(k) \in R^m$ is the estimated state of $x_i(k) \in R^m$ with $i = 1, \cdots, n$ and $\hat{z}_1(k-1) = [\hat{x}_1^T(k-1), \ldots, \hat{x}_n^T(k-1), u^T(k-1)]^T \in R^{(n+1)m}$ is the input vector to the NN observer at the kth instant, $\hat{w}_1(k-1) \in R^{n_1\times m}$ and $v_1 \in R^{(n+1)m\times n_1}$ denote the output and hidden layer weights, and n_1 is the number of the hidden layer nodes. For simplicity purpose, the hidden layer activation function vector $\phi_1(v_1^T\hat{z}_1(k-1)) \in R^{n_1}$ is written as $\phi_1(\hat{z}_1(k-1))$. It is demonstrated in [9] that if the hidden layer weights, v_1, are chosen initially at random and kept constant and if n_1 is sufficiently large, the NN approximation error can be made arbitrarily small since the activation function vector forms a basis.

B. Observer Error Dynamics

Define the state estimation error by

$$\tilde{x}_i(k) = \hat{x}_i(k) - x_i(k), \quad i = 1, \ldots, n \qquad (4)$$

Manuscript received November 7, 2003; revised April 10, 2004. This paper was recommended by Associate Editor W. Pedrycz.

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

Digital Object Identifier 10.1109/TSMCB.2004.840124

where $\tilde{x}_i(k) \in R^m$, $i = 1, \ldots, n$, is the state estimation error. In fact, the observer NN approximates the nonlinear function given by $f(x(k-1)) + g(x(k-1))u(k-1)$. This nonlinear function can be expressed as

$$\begin{aligned} f(x(k-1)) &+ g(x(k-1))\,u(k-1) \\ &= w_1^T \phi_1\left(v_1^T z_1(k-1)\right) + \varepsilon_1(z_1(k-1)) \\ &= w_1^T \phi_1(z_1(k-1)) + \varepsilon_1(z_1(k-1)) \end{aligned} \tag{5}$$

where $w_1 \in R^{n_1 \times m}$ is the target NN weight matrix, $\varepsilon_1(z_1(k-1))$ is the NN approximation error, and the NN input is given by $z_1(k-1) = [x_1^T(k-1), \ldots, x_n^T(k-1), u^T(k-1)]^T \in R^{(n+1)m}$. For convenience, the hidden layer activation function vector $\phi_1(v_1^T z_1(k-1)) \in R^{n_1}$ is written as $\phi_1(z_1(k-1))$.

Combining (3), (4), and (5) to get

$$\begin{aligned} \tilde{x}_n(k) &= \hat{x}_n(k) - x_n(k) \\ &= \hat{x}_n(k) - f(x(k-1)) - g(x(k-1))\,u(k-1) \\ &\quad - d'(k-1) \\ &= \hat{w}_1^T(k-1)\phi_1(\hat{z}_1(k-1)) - w_1^T \phi_1(z_1(k-1)) \\ &\quad - \varepsilon_1(z_1(k-1)) - d'(k-1) \\ &= \zeta_1(k-1) + d_1(k-1) \end{aligned} \tag{6}$$

where

$$\tilde{w}_1(k-1) = \hat{w}_1(k-1) - w_1 \tag{7}$$

$$\zeta_1(k-1) = \tilde{w}_1^T(k-1)\phi_1(\hat{z}_1(k-1)) \tag{8}$$

$$\phi_1(\tilde{z}_1(k-1)) = \phi_1(\hat{z}_1(k-1)) - \phi_1(z_1(k-1)) \tag{9}$$

$$\begin{aligned} d_1(k-1) &= w_1^T \phi_1(\tilde{z}_1(k-1)) \\ &\qquad - \left(\varepsilon_1(z_1(k-1)) + d'(k-1)\right). \end{aligned} \tag{10}$$

The dynamics of the estimation error using (4) and (6) is obtained as

$$\begin{aligned} \tilde{x}_1(k) &= \tilde{x}_2(k-1) \\ &\vdots \\ \tilde{x}_n(k) &= \zeta_1(k-1) + d_1(k-1). \end{aligned} \tag{11}$$

IV. Output Feedback Controller Design

Our objective is to design an adaptive critic NN output feedback controller with input constraints for the system (1) and (2) such that all the signals in the closed-loop system remain UUB; the state $x(k)$ follows a desired trajectory $Y_d(k) = [y_d^T(k), \cdots, y_d^T(k+n-1)]^T \in R^{nm}$, with $y_d(k) \in R^m$ and $y_d(k+i)$ referred as the future value of $y_d(k)$, $i = 1, \ldots, n-1$; and certain long-term system performance index is optimized.

Assumption 2: The desired trajectory, $Y_d(k)$, is a smooth bounded function over the compact subset of R^{nm}.

A. Auxiliary Controller Design

Define the tracking error between actual and desired trajectory as

$$e_i(k+1) = x_i(k+1) - y_d(k+i), \quad i = 1, \ldots, n. \tag{12}$$

Equation (1) can be rewritten as

$$e_n(k+1) = f(x(k)) + g(x(k))\,u(k) + d'(k) - y_d(k+n). \tag{13}$$

Define the desired auxiliary control signal as

$$v_d(k) = g^{-1}(x(k))\left(-f(x(k) + y_d(k+n) + l_1 e_n(k)\right) \tag{14}$$

where $l_1 \in R^{m \times m}$ is a design matrix selected such that the tracking error, $e_n(k)$, is bounded.

Since $f(x(k))$ and $g(x(k))$ are unknown smooth functions, the desired auxiliary feedback control input $v_d(k)$ cannot be implemented. From (14) and using Assumptions 1 and 2, $v_d(k)$ can be approximated by the action NN as

$$v_d(k) = w_2^T \phi_2\left(v_2^T s(k)\right) + \varepsilon_2(s(k)) = w_2^T \phi_2(s(k)) + \varepsilon_2(s(k)) \tag{15}$$

where $s(k) = [x^T(k), e_n^T(k)]^T \in R^{(n+1)m}$ is the NN input vector, $w_2 \in R^{n_2 \times m}$ and $v_2 \in R^{(n+1)m \times n_2}$ denote the output and hidden layer target weights, $\varepsilon_2(s(k))$ is the action NN approximation error, and n_2 is the number of the nodes in the hidden layer. For simplicity purpose, the hidden layer activation function vector $\phi_2(v_2^T s(k)) \in R^{n_2}$ is written as $\phi_2(s(k))$.

Since the states $x_i(k)$, $i = 2, \ldots, n$ are not measurable at the kth time instant, replacing the actual states with their estimated values, (15) can be expressed as

$$v(k) = \hat{w}_2^T(k)\phi_2\left(v_2^T \hat{s}(k)\right) = \hat{w}_2^T(k)\phi_2(\hat{s}(k)) \tag{16}$$

where $\hat{w}_2(k) \in R^{n_2 \times m}$ is the actual weight matrix, the action NN input is given by $\hat{s}(k) = [\hat{x}^T(k), \hat{e}_n^T(k)]^T \in R^{(n+1)m}$, with $\hat{e}_n(k) \in R^m$ referred as the modified tracking error, which is defined between the estimated state vector and the desired trajectory as

$$\hat{e}_i(k+1) = \hat{x}_i(k+1) - y_d(k+i), \quad i = 1, \ldots, n \tag{17}$$

and

$$\hat{e}(k) = \begin{bmatrix} \hat{e}_1(k) \\ \vdots \\ \hat{e}_n(k) \end{bmatrix} = \begin{bmatrix} \hat{x}_1(k) - y_d(k) \\ \vdots \\ \hat{x}_n(k) - y_d(k+n-1) \end{bmatrix}. \tag{18}$$

B. Controller Design With Magnitude Constraints

By applying the magnitude constraints, the actual control input $u(k) \in R^m$ is now given by

$$u(k) = \begin{cases} v(k), & \text{if } \|v(k)\| \le u_{\max} \\ u_{\max}\operatorname{sgn}(v(k)), & \text{if } \|v(k)\| \ge u_{\max} \end{cases} \tag{19}$$

where $u_{\max}$ is the upper limit defined by the actuator.

Case 1: $\|v(k)\| \le u_{\max}$: In this case, the control input $u(k) = v(k)$. Substituting, (14), (15) and (16) into (13) yields

$$\begin{aligned} e_n(k+1) &= f(x(k)) + g(x(k))\,v(k) + d'(k) - y_d(k+n) \\ &= f(x(k)) + g(x(k))\,(v_d(k) + v(k) - v_d(k)) \\ &\quad + d'(k) - y_d(k+n) \\ &= l_1 e_n(k) + g(x(k))\,\zeta_2(k) + g(x(k)) \\ &\quad \times \left(w_2^T \phi_2(\tilde{s}(k)) + \varepsilon_2(s(k))\right) + d'(k) \\ &= l_1 e_n(k) + g(x(k))\,\zeta_2(k) + d_2(k) \end{aligned} \tag{20}$$

where

$$\tilde{w}_2(k) = \hat{w}_2(k) - w_2 \tag{21}$$

$$\xi_2(k) = \tilde{w}_2^T(k)\phi_2(\hat{s}(k)) \tag{22}$$

$$\phi_2(\tilde{s}(k)) = \phi_2(\hat{s}(k)) - \phi_2(s(k)) \tag{23}$$

$$d_2(k) = g(x(k))\Big(w_2^T \phi_2(\tilde{s}(k)) - \varepsilon_2(s(k))\Big) + d'(k). \tag{24}$$

Thus, the tracking error dynamics is given by

$$\begin{aligned} e_1(k+1) &= e_2(k) \\ &\vdots \\ e_n(k+1) &= l_1 e_n(k) + g(x(k))\,\zeta_2(k) + d_2(k) \end{aligned} \tag{25}$$

Case 2: $\|v(k)\| \geq u_{\max}$*:* In this case, the control input $u(k) = u_{\max}\mathrm{sgn}(v(k))$. Combining with (13), (14), (15), and (16) to get

$$\begin{aligned} e_n(k+1) &= f\left(x(k)\right) + g\left(x(k)\right)u(k) + d'(k) - y_d(k+n) \\ &= l_1 e_n(k) + g(x(k))\Big(u_{\max}\mathrm{sgn}(v(k)) - w_2^T\phi_2(s(k)) \\ &\qquad - \varepsilon_2\left(s(k)\right)) + d'(k) \\ &= l_1 e_n(k) + d_2'(k) \end{aligned} \tag{26}$$

where

$$\begin{aligned} d_2'(k) = g\left(x(k)\right)\Big(u_{\max}\mathrm{sgn}\left(v(k)\right) - w_2^T\phi_2\left(s(k)\right) \\ - \varepsilon_2\left(s(k)\right)) + d'(k). \end{aligned} \tag{27}$$

Therefore, for the *Case 2*, the tracking error dynamics is written as

$$\begin{aligned} e_1(k+1) &= e_2(k) \\ &\vdots \\ e_n(k+1) &= l_1 e_n(k) + d_2'(k). \end{aligned} \tag{28}$$

V. Weight Updates for Guaranteed Performance

The next step is to design the observer, action and critic NN's weight updating rules using Lyapunov analysis.

A. Weights Updating Rule for the Observer NN

The observer NN weight update is driven by the state estimation error $\widetilde{x}_1(k)$, i.e.,

$$\hat{w}_1(k+1) = \hat{w}_1(k) - \alpha_1\phi_1(\hat{z}_1(k))\left(\hat{w}_1^T(k)\phi_1(\hat{z}_1(k)) + l_2\widetilde{x}_1(k)\right)^T \tag{29}$$

where $l_2 \in R^{m\times m}$ is a design matrix, and $\alpha_1 \in R^+$ is the adaptation gain of the NN observer.

B. Strategic Utility Function

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

$$p_i(k) = \begin{cases} 0, & \text{if abs}\left(e_n^i(k)\right) \leq c \\ 1, & \text{otherwise} \end{cases}, \quad i = 1, 2, \ldots, m \tag{30}$$

where $e_n^i(k) \in R$ is the ith element of vector $e_n(k)$, $abs(e_n^i(k))$ is the absolute value of $e_n^i(k)$, $c \in R^+$ is a pre-defined threshold. The utility function $p(k)$ is viewed as the current system performance index: $p_i(k) = 0$ and $p_i(k) = 1$ refer to the good and unacceptable tracking performance, respectively.

The *strategic* utility function $Q(k) \in R^m$ is defined as

$$Q(k) = \alpha^N p(k+1) + \alpha^{N-1}p(k+2) + \cdots + \alpha^{k+1}p(N) \tag{31}$$

where $\alpha \in R$ is a design parameter, $0 < \alpha < 1$, and N is the final time instant. The term $Q(k)$ is viewed here as the long system performance measure since it is the sum of all future system performance indices. Equation (31) can be also be expressed as $Q(k) = \min\limits_{u(k)}\{\alpha Q(k-1) - \alpha^{N+1}p(k)\}$, which is similar to the standard Bellman equation.

C. Design of the Critic NN

The critic NN is employed to approximate the *strategic* utility function $Q(k)$, since $Q(k)$ is unavailable at the kth time instant. The critic signal is then used to tune the action NN to minimize $Q(k)$. The prediction error is defined as

$$e_c(k) = \hat{Q}(k) - \alpha\left(\hat{Q}(k-1) - \alpha^N p(k)\right) \tag{32}$$

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

$$\hat{Q}(k) = \hat{w}_3^T(k)\phi_3\left(v_3^T\hat{x}(k)\right) = \hat{w}_3^T(k)\phi_3\left(\hat{x}(k)\right) \tag{33}$$

and $\hat{Q}(k) \in R^m$ is the critic signal, $\hat{w}_3(k) \in R^{n_3\times m}$ and $v_3 \in R^{nm\times n_3}$ represent the matrix of weight estimates, n_3 is the number of the nodes in the hidden layer, and the critic NN input is selected as the state estimate $\hat{x}(k) = [\hat{x}_1^T(k), \ldots, \hat{x}_n^T(k)]^T \in R^{nm}$. The activation function vector of the hidden layer $\phi_3(v_3^T\hat{x}(k)) \in R^{n_3}$ is written as $\phi_3(\hat{x}(k))$. The objective function to be minimized by the critic NN is defined as

$$E_c(k) = \frac{1}{2}e_c^T(k)e_c(k). \tag{34}$$

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

$$\hat{w}_3(k+1) = \hat{w}_3(k) + \Delta\hat{w}_3(k) \tag{35}$$

where

$$\Delta\hat{w}_3(k) = \alpha_3\left[-\frac{\partial E_c(k)}{\partial\hat{w}_3(k)}\right]. \tag{36}$$

where $\alpha_3 \in R$ is the adaptation gain. Before we proceed further, the following Lemma is required.

Lemma 1: Given the matrices $A \in R^{m\times m}$, $X \in R^{n\times m}$ and vectors $b \in R^n$ and $q \in R^m$, the derivative of the following quadratic term with respect to the matrix X is given by

$$\frac{d\left((AX^Tb+q)^T(AX^Tb+q)\right)}{dX} = 2b\left(A^T(AX^Tb+q)\right)^T. \tag{37}$$

where the matrix A, vectors b and q are independent of the matrix X.

Combining (32), (33), (34) with (36), we derive the critic NN weight updating rule as shown in (38) at the bottom of the page. Using *Lemma 1* (note: in this case, A is an identity matrix), (38) can be simplified as

$$\begin{aligned} \Delta\hat{w}_3(k) =& -\alpha_3\phi_3\left(\hat{x}(k)\right)\Big(\hat{w}_3^T(k)\phi_3\left(\hat{x}(k)\right) \\ &\qquad - \alpha\left(\hat{Q}(k-1) - \alpha^N p(k)\right)\Big)^T \\ =& -\alpha_3\phi_3\left(\hat{x}(k)\right)\Big(\hat{Q}(k) + \alpha^{N+1}p(k) \\ &\qquad - \alpha\hat{Q}(k-1)\Big)^T. \end{aligned} \tag{39}$$

Thus the critic NN weight updating rule is obtained as

$$\begin{aligned} \hat{w}_3(k+1) = \hat{w}_3(k) - \alpha_3\phi_3\left(\hat{x}(k)\right) \\ \times\left(\hat{Q}(k) + \alpha^{N+1}p(k) - \alpha\hat{Q}(k-1)\right)^T. \end{aligned} \tag{40}$$

$$\Delta\hat{w}_3(k) = -\frac{1}{2}\alpha_3\left[\frac{\partial e_c^T(k)e_c(k)}{\partial\hat{w}_3(k)}\right] = -\frac{1}{2}\alpha_3\left[\frac{\partial\left[\left(\hat{Q}(k) - \alpha\left(\hat{Q}(k-1) - \alpha^N p(k)\right)\right)^T\left(\hat{Q}(k) - \alpha\left(\hat{Q}(k-1) - \alpha^N p(k)\right)\right)\right]}{\partial\hat{w}_3(k)}\right]. \tag{38}$$

D. Weight Updating Rule for the Action NN

The action NN weight $\hat{w}_2^T(k)$ are tuned by using the functional estimation error, $\zeta_2(k)$, and the error between the desired *strategic* utility function $Q_d(k) \in R^m$ and the critic signal $\hat{Q}(k)$. Define

$$e_a(k) = \sqrt{g(x(k))}\zeta_2(k) + \left(\sqrt{g(x(k))}\right)^{-1}\left(\hat{Q}(k) - Q_d(k)\right) \quad (41)$$

where $\zeta_2(k)$ is defined in (22), $\sqrt{g(x(k))} \in R^{m\times m}$ is the principle square root of the diagonal positive definite matrix $g(x(k))$, i.e., $(\sqrt{g(x(k))})^2 = g(x(k))$, and $(\sqrt{g(x(k))})^T = (\sqrt{g(x(k))})$, $e_a(k) \in R^m$, and the subscript "a" stands for the "action NN".

The desired *strategic* utility function $Q_d(k)$ is considered to be zero ("0") [7], to indicate that at every step, the nonlinear system can track the reference signal well. Thus, (41) becomes

$$e_a(k) = \sqrt{g(x(k))}\zeta_2(k) + \left(\sqrt{g(x(k))}\right)^{-1}\hat{Q}(k). \quad (42)$$

The objective function to be minimized is given by

$$E_a(k) = \frac{1}{2}e_a^T(k)e_a(k). \quad (43)$$

Combining (22), (42), (43) with Lemma 1, we get $\Delta\hat{w}_2(k)$ as

$$\Delta\hat{w}_2(k)^T = -\alpha_2\phi_2(\hat{s}(k))\left(g(x(k))\zeta_2(k) + \hat{Q}(k)\right). \quad (44)$$

Using (25), (44) can be further expressed as

$$\Delta\hat{w}_2(k) = -\alpha_2\phi_2(\hat{s}(k))\left(e_n(k+1) - l_1e_n(k) - d_2(k) + \hat{Q}(k)\right)^T \quad (45)$$

where $\alpha_2 \in R^+$ is the adaptation gain of the action NN. Since $e_n(k+1)$ and $e_n(k)$ are unavailable, the modified tracking errors $\hat{e}_n(k+1)$ and $\hat{e}_n(k)$ respectively are used instead. In the ideal case, we take the disturbance $d_2(k)$ as zero to obtain the action NN weight updating rule as

$$\hat{w}_2(k+1) = \hat{w}_2(k) - \alpha_2\phi_2(\hat{s}(k))\left(\hat{e}_n(k+1) - l_1\hat{e}_n(k) + \hat{Q}(k)\right)^T. \quad (46)$$

VI. Main Result

Assumption 3: Let w_1, w_2, and w_3 be the unknown output layer target weights for the observer, action and critic NNs, and assume that they are bounded above so that

$$\|w_1\| \le w_{1m}, \quad \|w_2\| \le w_{2m}, \text{ and } \|w_3\| \le w_{3m} \quad (47)$$

where $w_{1m} \in R^+$, $w_{2m} \in R^+$ and $w_{3m} \in R^+$ represent the bounds on the unknown target weights.

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

$$\|\phi_i(k)\| \le \phi_{im}, \quad i = 1, 2, 3 \quad (48)$$

where $\phi_{im} \in R^+, i = 1, 2, 3$ is the upper bound for $\phi_i(k), i = 1, 2, 3$.

Assumption 4: The NN approximation errors $\varepsilon_1(z_1(k))$ and $\varepsilon_2(s(k))$ are bounded over the compact set $S \subset R^m$ by ε_{1m} an ε_{2m}, respectively, [9].

Fact 2: With the Assumptions (1), (3), (4), and Fact 1, the terms $d_1(k)$ ((10)), $d_2(k)$ ((24)) and $d_2'(k)$ ((27)) are bounded over the compact set $S \subset R^m$ by d_{1m}, d_{2m} and d_{2m}', respectively.

Theorem 1: Consider the system given by (1) and (2). Let the Assumptions 1 through 4 hold with the disturbance bound d_m' a known constant. Let the state estimate vector and control input be provided by the observer (3) and (19) respectively. Let the NN observer, action NN, and the critic NN weight tuning be given by (29), (46) and (40), respectively. Then the state estimation error $\tilde{x}_i(k)$, the tracking error $e_i(k)$, and the NN weight estimates, $\hat{w}_1(k)$, $\hat{w}_2(k)$ and $\hat{w}_3(k)$ are UUB, with the bounds specifically given by (A.5)through (A.9) provided the controller design parameters are selected as:

$$\text{(a) } 0 < \alpha_1\|\phi_1(\hat{z}_1(k))\|^2 < 1 \quad (49)$$

$$\text{(b) } 0 < \alpha_2\|\phi_2(k)\|^2 < \min\left(\frac{g_{\min}}{g_{\max}^2}, \frac{1}{g_{\min}}\right) \quad (50)$$

$$\text{(c) } 0 < \alpha_3\|\phi_3(\hat{x}(k))\|^2 < 1 \quad (51)$$

$$\text{(d) } 0 < \alpha < \frac{\sqrt{2}}{2} \quad (52)$$

where α_1, α_2, and α_3 are NN adaptation gains, and α is employed to define the *strategic* utility function.

Proof: See Appendix. ■

VII. Conclusion

A novel adaptive critic NN based output feedback controller with magnitude constraints is designed to deliver a desired tracking performance for a class of MIMO strict feedback nonlinear discrete-time systems. The adaptive critic NN structure optimizes certain *strategic* utility function, which is very similar to the standard Bellman equation. Magnitude constraints on the control input allow the designer to meet the physical limits of the actuator while meeting the closed-loop stability and tracking performance. The UUB of the closed-loop tracking and estimation errors, and NN weight estimates was demonstrated.

Appendix

Proof of Theorem 1

Case 1: $\|v(k)\| \le u_{\max}$*:* Define the Lyapunov function as

$$\begin{aligned} J(k) = {} & \frac{\gamma_1}{2}\sum_{i=1}^{n}\|\tilde{x}_i(k-1)\|^2 + \frac{\gamma_2}{2}\sum_{i=1}^{n}\|\tilde{x}_i(k)\|^2 + \frac{\gamma_3}{3}\sum_{i=1}^{n}\|e_i(k)\|^2 \\ & + \frac{\gamma_4}{3}\sum_{i=1}^{n}\|e_n(k)\|^2 + \frac{\gamma_5}{\alpha_1}tr\left(\tilde{w}_1^T(k-1)\tilde{w}_1(k-1)\right) \\ & + \frac{\gamma_6}{\alpha_1}tr\left(\tilde{w}_1^T(k)\tilde{w}_1(k)\right) + \frac{\gamma_7}{\alpha_2}tr\left(\tilde{w}_2^T(k)\tilde{w}_2(k)\right) \\ & + \frac{\gamma_8}{\alpha_3}tr\left(\tilde{w}_3^T(k)\tilde{w}_3(k)\right) + \gamma_9\|\zeta_3(k)\|^2 \end{aligned} \quad \text{(A.1)}$$

where $\gamma_i \in R^+, i = 1, \ldots, 9$ are design parameters. The first difference of Lyapunov function is given by

$$\Delta J(k) = \sum_{i=1}^{9}\Delta J_i(k). \quad \text{(A.2)}$$

Using (11), (17), (25), (29), and (46) to obtain the $\Delta J(k)$ as

$$\begin{aligned} \Delta J(k) = {} & -\frac{1}{2}\left(\gamma_1 - 4\gamma_5 l_{2\max}^2\right)\|\tilde{x}_1(k-1)\|^2 \\ & - \frac{1}{2}\left(\gamma_2 - 4\gamma_6 l_{2\max}^2\right)\|\tilde{x}_1(k)\|^2 - \frac{\gamma_3}{3}\|e_1(k)\|^2 \\ & - \frac{1}{3}\left(\gamma_4 - 3(\gamma_3 + \gamma_4)l_{1\max}^2\right)\|e_n(k)\|^2 \\ & - \gamma_6\left(1 - \alpha_1\|\phi_1(k)\|^2\right)\left\|\zeta_1(k) + l_2\tilde{x}_1(k) + w_1^T\phi_1(k)\right\|^2 \\ & - \left(\gamma_6 - \gamma_2 - 2\gamma_7'\right)\|\zeta_1(k)\|^2 - \gamma_5\left(1 - \alpha_1\|\phi_1(k-1)\|^2\right) \\ & \times\left\|\zeta_1(k-1) + l_2\tilde{x}_1(k-1) + w_1^T\phi_1(k-1)\right\|^2 \\ & - \left(\gamma_5 - \gamma_1 - 2\gamma_7' l_{2\max}^2\right)\|\zeta_1(k-1)\|^2 \\ & - \gamma_7\left(g_{\min} - \alpha_2\|\phi_2(k)\|^2 g_{\max}^2\right) \end{aligned}$$

$$
\begin{aligned}
&\times \left\| \zeta_2(k) + \frac{\left(I - \alpha_2 \left\|\phi_2(k)\right\|^2 g\left(x(k)\right)\beta(k)\right)}{g_{\min} - \alpha_2 \left\|\phi_2(k)\right\|^2 g_{\max}^2} \right\|^2 \\
&- \left(\gamma_7 g_{\min} - \gamma_3 g_{\max}^2 - \gamma_4 g_{\max}^2\right) \left\|\zeta_2(k)\right\|^2 \\
&- \gamma_8 \left(1 - \alpha_3 \left\|\phi_3(k)\right\|^2\right) \\
&\times \left\| \zeta_3(k) + w_3^T \phi_3(k) + \alpha^{N+1} p(k) - \alpha \hat{Q}(k-1) \right\|^2 \\
&- \left(\gamma_8 - 2\gamma_8 \alpha^2 - \gamma_7'\right) \left\|\zeta_3(k)\right\|^2 + D_M^2 \qquad \text{(A.3)}
\end{aligned}
$$

where

$$
\begin{aligned}
D_M^2 = &\left(\gamma_1 + \gamma_2 + 2\left(1 + l_{2\max}^2\right)\gamma_7'\right) d_{1m}^2 + \left(\gamma_3 + \gamma_4 + 2\gamma_7'\right) d_{2m}^2 \\
&+ 2\gamma_5 w_{1m}^2 \phi_{1m}^2 + 2\gamma_6 w_{1m}^2 \phi_{1m}^2 + 6\gamma_8 \\
&+ 2\left(\gamma_7' + 3\gamma_8(1 + \alpha^2)\right) w_{3m}^2 \phi_{3m}^2 \qquad \text{(A.4)}
\end{aligned}
$$

$l_{1\max} \in R$ and $l_{2\max} \in R$ are the maximum eigenvalues of matrix l_1 and l_2, respectively.

This implies that $\Delta J(k) \leq 0$ as long as (50) through (52) is satisfied and the following conditions hold:

$$
\left\|\tilde{x}_1(k)\right\| \geq \frac{\sqrt{2} D_M}{\sqrt{\gamma_2 - 4\gamma_6 l_{2\max}^2}} \qquad \text{(A.5)}
$$

or

$$
\left\|e_n(k)\right\| \geq \frac{\sqrt{3} D_M}{\sqrt{\gamma_4 - 3(\gamma_3 + \gamma_4) l_{1\max}^2}} \qquad \text{(A.6)}
$$

or

$$
\left\|\zeta_1(k)\right\| \geq \frac{D_M}{\sqrt{\gamma_6 - \gamma_2 - 2\gamma_7'}} \qquad \text{(A.7)}
$$

or

$$
\left\|\zeta_2(k)\right\| \geq \frac{D_M}{\sqrt{\gamma_7 g_{\min} - \gamma_3 g_{\max}^2 - \gamma_4 g_{\max}^2}} \qquad \text{(A.8)}
$$

or

$$
\left\|\zeta_3(k)\right\| \geq \frac{D_M}{\sqrt{\gamma_8 - 2\gamma_8 \alpha^2 - \gamma_7'}}. \qquad \text{(A.9)}
$$

Case 2: $\|v(k)\| > u_{\max}$: The proof is similar to that in *Case 1* and it is omitted.

For both Case 1 and Case 2, $\Delta J(k) \leq 0$ for all k is greater than zero. According to the standard Lyapunov extension theorem, this demonstrates that $\tilde{x}_1(k)$, $e_n(k)$ and the weight estimation errors are UUB. The boundedness of $\|\zeta_1(k)\|$, $\|\zeta_2(k)\|$ and $\|\zeta_3(k)\|$ implies that $\|\tilde{w}_1(k)\|$, $\|\tilde{w}_2(k)\|$ and $\|\tilde{w}_3(k)\|$ and weight estimates $\hat{w}_1(k)$, $\hat{w}_2(k)$ and $\hat{w}_3(k)$ are bounded. Since $\tilde{x}_1(k)$ is bounded, using (11), the estimation errors are bounded. Similarly, bounded $e_n(k)$ implies that all the tracking errors are bounded from (25) and (28). Therefore, all the signals in the observer-controller system are bounded.

References

[1] M. Krstic, I. Kanellakopoulos, and P. Kokotovic, *Nonlinear and Adaptive Control Design*. New York: Wiley, 1995.

[2] P. C. Yeh and P. V. Kokotovic, "Adaptive output feedback design for a class of nonlinear discrete-time systems," *IEEE Trans. Automat. Contr.*, vol. 40, no. 9, pp. 1663–1668, Sep. 1995.

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

[4] S. S. Ge, T. H. Lee, G. Y. Li, and J. Zhang, "Adaptive NN control for a class of discrete-time nonlinear systems," *Int. J. Contr.*, vol. 76, no. 4, pp. 334–354, 2003.

[5] 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 Reinhold, 1992, pp. 65–90.

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

[7] 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.

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

[9] 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.