Discrete-Time Neural Network Output Feedback Control of Nonlinear Systems in Non-Strict Feedback Form[1]

P. He and S. Jagannathan

Abstract—**An adaptive neural network (NN) -based output feedback controller is proposed to deliver a desired tracking performance for a class of discrete-time nonlinear systems, which is represented in non-strict feedback form. The NN backstepping approach is utilized to design the adaptive output feedback controller consisting of: 1) a NN observer to estimate the system states with the input-output data, and 2) two NNs to generate the virtual and actual control inputs, respectively. The non-causal problem in the discrete-time backstepping design is avoided by using the universal NN approximator. The persistence excitation (PE) condition is relaxed both in the NN observer and NN controller design. The uniformly ultimate boundedness (UUB) of the closed-loop tracking error, the state estimation errors and the NN weight estimates is shown.**

I. INTRODUCTION

THE adaptive neural network (NN) backstepping control approach 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 procedure to estimate certain nonlinear functions, a more suitable control law can be designed without using the LIP assumption and the need for a regression matrix of the standard backstepping approach [1].

Adaptive NN backstepping state feedback control of nonlinear discrete-time systems in strict feedback form has been addressed in the literature [2], where the nonlinear system is expressed as $x_i(k+1) = f_i(\bar{x}_i(k)) + g_i(\bar{x}_i(k))x_{i+1}(k)$, and $x_n(k+1) = f_n(\bar{x}_n(k)) + g_n(\bar{x}_n(k))u(k)$ where $x_i(k) \in R$ is the state, $u(k) \in R$ is the control input, $\bar{x}_i(k) = [x_1(k), \cdots, x_i(k)]^T \in R^i$ and $i = 1, \ldots, (n-1)$. For the strict-feedback systems [1], the nonlinearities $f_i(\bar{x}_i(k))$ and $g_i(\bar{x}_i(k))$ depend only upon states $x_1(k), \ldots, x_i(k)$, i.e., $\bar{x}_i(k)$. In state feedback control design, the control input depends on the available states. If the states are not available for measurement or if they are too expensive to measure, an observer is used to estimate the states, and then the estimated values will be substituted for the unavailable states in the output feedback controller design.

Several output feedback control schemes by using the backstepping design [3-4] in discrete time are developed for the strict feedback nonlinear systems. However, for the non-strict feedback nonlinear systems, the previous methods will result in a non-causal control problem (the current control depends on the future system states). Therefore, an adaptive NN output feedback controller is proposed to deliver a desired tracking performance for a class of discrete-time nonlinear systems in non-strict feedback form. Several practical systems, for instance the spark ignition engine dynamics operating either with high EGR levels or under lean operation [5], can be represented in non-strict feedback form. The non-causal problem is overcome by employing the NN approximator.

The proposed adaptive NN output feedback controller design employs the backstepping approach and it includes three NNs: 1) a NN observer to estimate certain system states with the input-output data, and 2) two NNs to generate the virtual and real control input, respectively. The main contributions of this paper can be summarized as follows: 1) the adaptive NN output feedback control scheme is extended to the non-strict feedback nonlinear systems. The non-causal problem is confronted by employing the universal NN approximator; 2) the requirement of the PE condition for the boundedness of NN weight estimates is relaxed for both the NN observer and controller design by using novel NN weight updating rules and selecting the overall Lyapunov function consisting of system tracking error, system state estimation errors and NN weight estimation errors; 3) a well-defined controller is presented by overcoming the problem of $\hat{g}_i(\bar{x}_i(k))$ becoming zero since a single NN is used to approximate both the nonlinear functions $f_i(\bar{x}_i(k))$ and $g_i(\bar{x}_i(k))$; 4) the assumption that the sign of $g_i(\bar{x}_i(k))$ is known *a priori* is relaxed. The uniformly ultimate boundedness (UUB) of the closed-loop tracking error, the state estimation errors and the NN weight estimates is shown.

II. NONLINEAR SYSTEM DESCRIPTION AND NN OBSERVER DESIGN

A. The Nonlinear System Description

The discrete-time nonlinear system in non-strict feedback form is expressed as:

$$x_1(k+1) = f_1(x_1(k), x_2(k)) + g_1(x_1(k), x_2(k))x_2(k) + d_1'(k), \quad (1)$$

$$x_2(k+1) = f_2(x_1(k), x_2(k)) + g_2(x_1(k), x_2(k))u(k) + d'_2(k), \quad (2)$$
$$y(k) = x_1(k), \quad (3)$$

where $x_1(k) \in R$ and $x_2(k) \in R$ are the states, $u(k) \in R$ is the control input, $y(k) \in R$ is the system output, state $x_2(k)$ is not measurable, $d'_1(k) \in R$ and $d'_2(k) \in R$ are bounded unknown disturbances, whose bounds are given by $|d'_1(k)| < d'_{1m}$ and $|d'_2(k)| < d'_{2m}$.

Equations (1) and (2) represent a discrete-time nonlinear system in non-strict feedback form, since unknown functions $f_1(\cdot)$ and $g_1(\cdot)$ depend upon both states $x_1(k)$ and $x_2(k)$, unlike the case of strict feedback systems, where $f_1(\cdot)$ and $g_1(\cdot)$ depend upon only the state $x_1(k)$.

The control objective is to drive the system state $x_1(k)$ to track the desired trajectory $x_{1d}(k)$. Since $x_2(k)$ is considered unavailable, it is estimated by the NN observer. Subsequently, the estimated state is used to design the adaptive NN output feedback controller.

Throughout this paper, all quantities with "^" represent estimated quantities. In addition, quantities with "~" represent the estimation errors. Subscripts "o" and "c" refer to the observer and the controller quantities, respectively.

B. Observer Structure

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

$$x_1(k+1) = f_1(k) + g_1(k)x_2(k) + d'_1(k), \quad (4)$$
$$x_2(k+1) = f_2(k) + g_2(k)u(k) + d'_2(k). \quad (5)$$

Writing system (4) and (5) into the vector form as

$$x(k+1) = f(k) + d'(k), \quad (6)$$

where

$$x(k) = \begin{bmatrix} x_1(k) \\ x_2(k) \end{bmatrix}, f(k) = \begin{bmatrix} f_1(k) + g_1(k)x_2(k) \\ f_2(k) + g_2(k)u(k) \end{bmatrix}, d'(k) = \begin{bmatrix} d'_1(k) \\ d'_2(k) \end{bmatrix}, (7)$$

The term $f(k-1)$ can be viewed as an unknown smooth function vector, and it can be estimated by a NN [6].

$$f(k-1) = w_o^T \varphi(v_o^T z_o(k-1)) + \varepsilon_o(z_o(k-1))$$
$$= w_o^T \varphi(z_o(k-1)) + \varepsilon_o(z_o(k-1)), \quad (8)$$

where the input to the NN is taken as $z_o(k-1) = [x_1(k-1), x_2(k-1), u(k-1)]^T \in R^3$, the matrix $w_o \in R^{n_o \times 2}$ and $v_o \in R^{3 \times n_o}$ represent the target output and hidden layer weights, the hidden layer activation function $\varphi(z_o(k-1))$ represents $\varphi(v_o^T z_o(k-1))$, n_o denotes the number of the nodes in the hidden layer,

and $\varepsilon_o(z_o(k-1)) \in R$ is the functional approximation error. It is demonstrated in [6] that, if the hidden layer weight, v_o, is chosen initially at random and held constant and the number of hidden layer nodes is sufficiently large, the approximation error $\varepsilon_o(z_o(k-1))$ can be made arbitrarily small over the compact set $S \subset R$ since the activation function forms a basis.

The proposed NN observer for (6) is defined as

$$\hat{x}(k) = \hat{w}_o^T(k-1)\varphi(v_o^T \hat{z}_o(k-1)) = \hat{w}_o^T(k-1)\varphi(\hat{z}_o(k-1)), \quad (9)$$

where $\hat{x}(k) = [\hat{x}_1(k), \hat{x}_2(k)]^T \in R^2$ is the estimated value of $x(k)$, and $\hat{z}_o(k-1) = [\hat{x}_1(k-1), \hat{x}_2(k-1), u(k-1)]^T \in R^3$ is the input to the NN observer, the matrix $\hat{w}_o(k-1) \in R^{n_o \times 2}$ is the actually output layer weight, the $\varphi(\hat{z}_o(k-1))$ represents $\varphi(v_o^T \hat{z}_o(k-1))$. Here, it is assumed that the initial value of $u(0)$ is bounded. In the next section, via Lyapunov analysis, it is shown that all the values of $u(k)$ are bounded $\forall k \in R$.

C. Observer Error Dynamics

Define the state estimation errors as

$$\tilde{x}_i(k) = \hat{x}_i(k) - x_i(k) \quad i = 1, 2. \quad (10)$$

The estimation errors can be expressed in a vector form as

$$\tilde{x}(k) = \hat{x}(k) - x(k), \quad (11)$$

where $\tilde{x}(k) \in R^2$. Combining (6), (8), (9) and (11), we obtain the estimation error dynamics as

$$\tilde{x}(k) = \hat{x}(k) - x(k) = \hat{w}_o^T(k-1)\varphi(\hat{z}_o(k-1))$$
$$- w_o^T \varphi(z_o(k-1)) - \varepsilon_o(z_o(k-1)) - d'(k-1)$$
$$= \xi_o(k-1) + d_o(k-1), \quad (12)$$

where

$$\tilde{w}_o(k-1) = \hat{w}_o(k-1) - w_o, \quad (13)$$

$$\xi_o(k-1) = \tilde{w}_o^T(k-1)\varphi(\hat{z}_o(k-1))$$
$$= (\hat{w}_o(k-1) - w_o)^T \varphi(\hat{z}_o(k-1)), \quad (14)$$

$$w_o^T \varphi(\tilde{z}_o(k-1)) = w_o^T(\varphi(\hat{z}_o(k-1)) - \varphi(z_o(k-1))), \quad (15)$$

and

$$d_o(k-1) = w_o^T \varphi(\tilde{z}_o(k-1)) - d'(k-1) - \varepsilon_o(z_o(k-1)). \quad (16)$$

III. ADAPTIVE NN OUTPUT FEEDBACK CONTROLLER DESIGN

A. Adaptive NN Output Feedback Controller Design

Assumption 1: The desired trajectory $x_{1d}(k)$ is a smooth function, and hence it is bounded over the compact set S.

Assumption 2: The unknown smooth functions, $g_i(k)$, $\forall i = 1, 2$ are bounded away from zero within certain

compact set S as $g_{1M} > |g_1(k)| > g_{1m} > 0$ and $g_{2M} > |g_2(k)| > g_{2m} > 0$, respectively.

Next the adaptive NN output feedback control design is discussed.

Step 1: Virtual controller design.

Define the tracking error between actual and desired trajectory as

$$e_1(k) = x_1(k) - x_{1d}(k), \quad (17)$$

where $x_{1d}(k)$ is the desired trajectory. Combining with (4), (17) can be rewritten as

$$e_1(k+1) = x_1(k+1) - x_{1d}(k+1)$$
$$= f_1(k) + g_1(k)x_2(k) - x_{1d}(k+1) + d_1'(k). \quad (18)$$

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

$$x_{2d}(k) = \frac{1}{g_1(k)}(-f_1(k) + x_{1d}(k+1)). \quad (19)$$

The term $x_{2d}(k)$ can be approximated by the first action NN as

$$x_{2d}(k) = w_1^T \phi(v_1^T x(k)) + \varepsilon_1(x(k)) = w_1^T \phi(x(k)) + \varepsilon_1(x(k)), \quad (20)$$

where the input is the state $x(k)$, $w_1 \in R^{n_1}$ and $v_1 \in R^{2 \times n_1}$ denote the constant ideal output and hidden layer weights, n_1 is the hidden layer nodes number, the hidden layer activation function $\phi(v_2^T x(k))$ is simplified as $\phi(x(k))$, and $\varepsilon_1(x(k))$ is the approximation error.

Since $x_2(k)$ is unavailable, the estimated state $\hat{x}(k)$ is selected as the NN input. Consequently, the virtual control input is taken as

$$\hat{x}_{2d}(k) = \hat{w}_1^T(k)\phi(v_1^T \hat{x}(k)) = \hat{w}_1^T(k)\phi(\hat{x}(k)), \quad (21)$$

where $\hat{w}_1(k) \in R^{n_1}$ is the actual weight matrix for the first action NN. Define the weight estimation error by

$$\tilde{w}_1(k) = \hat{w}_1(k) - w_1(k), \quad (22)$$

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

$$e_2(k) = x_2(k) - \hat{x}_{2d}(k). \quad (23)$$

Equation (18) can be expressed using (23) for $x_2(k)$ as

$$e_1(k+1) = f_1(k) + g_1(k)(e_2(k) + \hat{x}_{2d}(k)) - x_{1d}(k+1) + d_1'(k), \quad (24)$$

or equivalently

$$e_1(k+1) = g_1(k)(e_2(k) + \xi_1(k) + d_1(k)), \quad (25)$$

where

$$\xi_1(k) = \tilde{w}_1^T(k)\phi(\hat{x}(k)), \quad (26)$$

$$w_1^T(k)\phi(\tilde{x}(k)) = w_1^T(k)(\phi(\hat{x}(k)) - \phi(x(k))), \quad (27)$$

and

$$d_1(k) = \frac{d_1'(k)}{g_1(k)} - \varepsilon_1(x(k)) + w_1^T(k)\phi(\tilde{x}(k)). \quad (28)$$

Step 2: Design of the control input $u(k)$.

Rewriting the error $e_2(k)$ from (23) as

$$e_2(k+1) = x_2(k+1) - \hat{x}_{2d}(k+1)$$
$$= f_2(k) + g_2(k)u(k) - \hat{x}_{2d}(k+1) + d_2'(k), \quad (29)$$

where $\hat{x}_{2d}(k+1)$ is the future value of $\hat{x}_{2d}(k)$. Here, $\hat{x}_{2d}(k+1)$ is not available in the current time step. However, from (19) and (21), it can be clear that $\hat{x}_{2d}(k+1)$ is a smooth nonlinear function of the state $x(k)$, virtual control input $\hat{x}_{2d}(k)$ and the system errors $e_1(k)$ and $e_2(k)$. Consequently, $\hat{x}_{2d}(k+1)$ is assumed to be approximated by using a NN.

Select the desired control input by using the second NN in the controller design as

$$u_d(k) = \frac{1}{g_2(k)}(-f_2(k) + \hat{x}_{2d}(k+1)) + l_1 e_1(k)$$
$$= w_2^T \sigma(v_2^T z(k)) + \varepsilon_2(z(k)) = w_2^T \sigma(z(k)) + \varepsilon_2(z(k)), \quad (30)$$

where $w_1 \in R^{n_2}$ and $v_1 \in R^{5 \times n_2}$ denote the constant ideal output and hidden layer weights, n_2 is the hidden layer nodes number, the hidden layer activation function $\sigma(v_2^T z(k))$ is simplified as $\sigma(z(k))$, $\varepsilon_1(z(k))$ is the approximation error, $l_1 \in R$ is the design constant, $z(k) \in R^5$ is the NN input, which is given by (31). Considering the fact state $x_2(k)$ cannot be measured, $z(k)$ is substituted with $\hat{z}(k) \in R^5$, where

$$z(k) = [x_1(k), x_2(k), \hat{x}_{2d}(k), l_1 e_1(k), e_2(k)]^T \in R^5, \quad (31)$$

and

$$\hat{z}(k) = [\hat{x}_1(k), \hat{x}_2(k), \hat{x}_{2d}(k), l_1 \hat{e}_1(k), \hat{e}_2(k)]^T \in R^5, \quad (32)$$

where

$$\hat{e}_1(k) = \hat{x}_1(k) - x_{1d}(k), \quad (33)$$

and

$$\hat{e}_2(k) = \hat{x}_2(k) - \hat{x}_{2d}(k). \quad (34)$$

The actual control input is now selected as

$$u(k) = \hat{w}_2^T(k)\sigma(\hat{z}(k)), \quad (35)$$

where $\hat{w}_2(k) \in R^{n_2}$ is the actual output layer weights.

Substituting (30) and (35) into (29) yields

$$e_2(k+1) = g_2(k)(l_1 e_1(k) + \xi_2(k) + d_2(k)), \quad (36)$$

where

$$\tilde{w}_2(k) = \hat{w}_2(k) - w_2(k), \quad (37)$$

$$\xi_2(k) = \tilde{w}_2^T(k)\sigma(\hat{z}(k)), \quad (38)$$

$$w_2^T(k)\sigma(\tilde{z}(k)) = w_2^T(k)(\sigma(\hat{z}(k)) - \sigma(z(k))), (39)$$

and

$$d_2(k) = \frac{d_2'}{g_2(k)} - \varepsilon_2(k) + w_2^T(k)\sigma(\tilde{z}(k)). (40)$$

Equations (25) and (36) represent the closed-loop error dynamics. It is required to show that the estimation error (12), the system errors (25) and (36) and the NN weight matrices $\hat{w}_o(k)$, $\hat{w}_1(k)$ and $\hat{w}_2(k)$ are bounded.

B. Weight Updates for Guaranteed Performance

Assumption 3 (Bounded Ideal Weights): Let w_o, w_1 and w_2 be the unknown output layer target weights for the observer and two action NNs and assume that they are bounded above so that

$$\|w_o\| \le w_{om}, \|w_1\| \le w_{1m}, \text{ and } \|w_2\| \le w_{2m}, \quad (41)$$

where $w_{om} \in R^+$, $w_{1m} \in R^+$ and $w_{2m} \in R^+$ represent the bounds on the unknown target weights where the Frobenius norm [7] is used.

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

$$\|\varphi(\cdot)\| \le \varphi_m, \|\phi(\cdot)\| \le \phi_m \text{ and } \|\sigma(\cdot)\| \le \sigma_m, \quad (42)$$

where $\varphi_m \in R^+$, $\phi_m \in R^+$ and $\sigma_m \in R^+$ are the upper bounds.

Assumption 4 (Bounded NN Approximation Error): The NN approximation errors $\varepsilon_o(z_o(k-1))$, $\varepsilon_1(x(k))$ and $\varepsilon_2(z(k))$ are bounded over the compact set $S \subset R$ by ε_{om}, ε_{1m} and ε_{2m}, respectively [6].

Fact 2: The terms $d_o(k-1) \in R^2$, $d_1(k) \in R$ and $d_2(k) \in R$ are bounded over the compact set S by

$$\|d_o(k-1)\| \le d_{om} = w_{om}\varphi_m + d_m' + \varepsilon_{om}, \quad (43)$$

where $d_m' \in R^+$ is the upper bound for $d_o'(k-1)$,

$$|d_1(k)| \le d_{1m} = \frac{d_{1m}'}{g_{1m}} + \varepsilon_{1m} + w_{1m}\phi_m, \quad (44)$$

and

$$|d_1(k)| \le d_{2m} = \frac{d_{1m}'}{g_{1m}} + \varepsilon_{1m} + w_{1m}\phi_m, \quad (45)$$

Theorem 1: Consider the system given in (1) and (2) and let the Assumptions 1 through 4 hold. Let the unknown disturbances be bounded by $\|d_1'(k)\| \le d_{1m}'$ and $\|d_2'(k)\| \le d_{2m}'$, respectively. Let the observer NN weight tuning be given by

$$\hat{w}_o(k+1) = \hat{w}_o(k) - \alpha_o\varphi(\hat{z}_o(k))(\hat{w}_o^T(k)\varphi(\hat{z}_o(k)) + l_1 e_1(k+1)I)^T, (46)$$

where $I = [1, 1]^T$, with the virtual control NN weight tuning be provided by

$$\hat{w}_1(k+1) = \hat{w}_1(k) - \alpha_1\phi(\hat{x}(k))(\hat{w}_1^T(k)\phi(\hat{x}(k)) + l_1 e_1(k)), (47)$$

and the control input weight be tuned by

$$\hat{w}_2(k+1) = \hat{w}_2(k) - \alpha_2\sigma(\hat{z}(k))(\hat{w}_1^T(k)\sigma(\hat{z}(k)) + l_1 e_1(k)), (48)$$

where $\alpha_o \in R$, $\alpha_1 \in R$, $\alpha_2 \in R$, and $l_1 \in R$ are design parameters. Let the NN observer, virtual and actual control inputs be defined as (9), (21) and (35), respectively. The estimation error (12), the tracking errors (25) and (36) and the NN weights $\hat{w}_o(k)$, $\hat{w}_1(k)$ and $\hat{w}_2(k)$ are UUB with the bounds specifically given by (A.12) through (A.17) provided the design parameters are selected as:

$$(1)\ 0 < \alpha_o\|\varphi(k)\|^2 < 1, \qquad (49)$$

$$(2)\ 0 < \alpha_1\|\phi(k)\|^2 < 1, \qquad (50)$$

$$(3)\ 0 < \alpha_2\|\sigma(k)\|^2 < 1, \qquad (51)$$

$$(4)\ |l_1| < \frac{1}{3\sqrt{13}g_{1m}g_{2m}}, \qquad (52)$$

Proof: See Appendix. ∎

Remark 1: A well-defined controller is developed in this paper by avoiding the problem of $\hat{g}_i(k), \forall i = 1, 2$ becoming zero.

Remark 2: It is important to note that in this theorem there is no PE condition for the NN observer and NN controller as well as the linearity in the parameters assumption, in contrast with standard work in the discrete-time adaptive control.

Remark 3: Generally, a nonlinear separation principle is not valid and hence it is relaxed in this paper for the controller design.

IV. SIMULATION

To verify the performance of the adaptive NN output feedback controller, consider the following nonlinear system, given in non-strict feedback form, as

$$x_1(k+1) = -\frac{1}{64}\frac{x_1(k)}{(5 + x_2^2(k))} + x_1(k) + 2x_2(k), \quad (53)$$

$$x_2(k+1) = -\frac{1}{16}\frac{x_2(k)}{(1 + x_1^2(k))} + \left(-\frac{7}{(1 + x_1^2(k) + x_2^2(k))}\right)u(k), (54)$$

$$y(k) = x_1(k), \qquad (55)$$

where $x_i(k) \in R$, $i = 1, 2$ are the states, $u(k) \in R$ is the control input, $y(k) \in R$ is the system output, the state $x_1(k)$ is known via the output $y(k)$, the state $x_2(k)$ is immeasurable. Note that $f_1(k) = -\frac{1}{64}\frac{x_1(k)}{(5 + x_2^2(k))} + x_1(k)$ is a nonlinear function of both states $x_1(k)$ and $x_2(k)$.

The objective is to drive the state $x_1(k)$ to track the reference signal, which was selected as $x_{1d}(k) = 2\sin(\omega kT + \xi)$, where $\omega = 0.1$, $\xi = \dfrac{\pi}{2}$ with a sampling interval of $T = 50msec$. The total simulation time is taken as 250 seconds. l_1 is taken as -0.05.

The number of hidden layer neurons in the observer NN, $\hat{w}_o^T \varphi(k)$, controller NN1 $\hat{w}_1^T \phi(k)$ and NN2 $\hat{w}_2^T \sigma(k)$ each was taken as 15. For weight updating, the learning rate is selected as $\alpha_o = 0.01$, $\alpha_1 = 0.1$ and $\alpha_2 = 0.1$. The inputs to observer NN, $\hat{w}_o^T \varphi(k)$, control NNs, $\hat{w}_1^T \phi(k)$ and $\hat{w}_2^T \sigma(k)$ are selected as $\hat{z}_o(k)$, $\hat{x}(k)$, and $\hat{z}(k)$ (32), respectively. The initial inputs to the hidden layer weights for the three NNs are selected at random over an internal of [0, 1] and all the activation functions used are hyperbolic tangent sigmoid functions. The initial output layer weights for all the three NN are chosen to be zero.

Two cases are considered: first, the adaptive output feedback NN controller is considered to the system, and then a proportional controller is applied. Fig. 1 illustrates the performance of the adaptive NN output feedback controller. From the figure, it is obvious that the system tracking performance is superior even when the state is not measured. The NN control input is presented in Fig. 2 where it clearly shows that the input is bounded. On the other hand, Figs 3 and 4 present the performance of the conventional proportional controller and the control input. The gain of the controller is also taken as -0.05. From Fig. 3, it is clear that the tracking performance has deteriorated in comparison with Fig. 1.
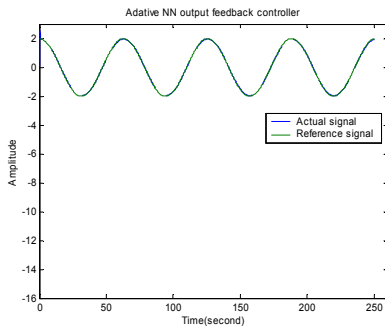

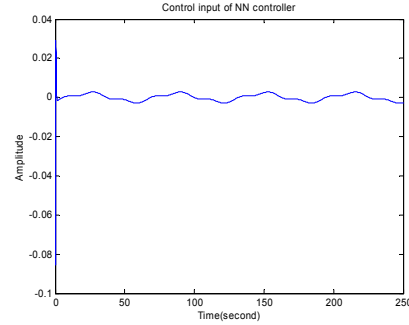**Fig. 1.** Performance of the adaptive NN output controller.

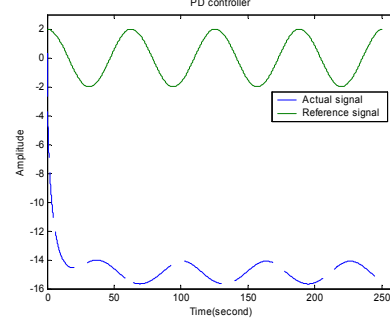**Fig. 2.** Adaptive NN controller input.

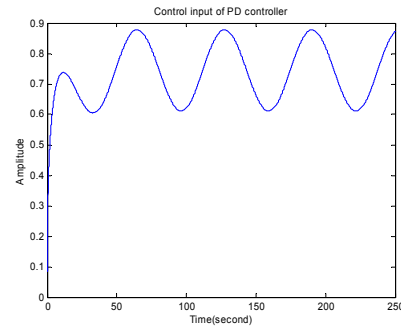**Fig. 3.** Performance of a conventional controller without NNs.

Fig. 4. Control input.

V. CONCLUSION

An adaptive neural network (NN) -based output feedback controller is proposed to deliver a desired tracking performance for a class of discrete-time nonlinear systems, which is expressed in non-strict feedback form. The adaptive NN output feedback controller consists of three NNs: 1) a NN observer to estimate the system states with the input-output data, and 2) two NNs to generate the virtual and real control inputs, respectively. The uniformly ultimate boundedness (UUB) of the closed-loop tracking error, the state estimation errors and the NN weight estimates is shown. Results show that the performance of the proposed controller schemes is highly satisfactory while meeting the closed loop stability. The controller scheme does not require an offline learning phase and the NN weights can be initialized randomly or to zeros.

APPENDIX

Proof of Theorem 1: Define the Lyapunov function

$$J(k) = \frac{1}{4}\widetilde{x}^T(k-1)\widetilde{x}(k-1) + \frac{l_2}{6g_{1m}^2}e_1^2(k) + \frac{1}{6g_{2m}^2}e_2^2(k)$$

$$+ \frac{1}{\alpha_o}tr(\widetilde{w}_o^T(k-1)\widetilde{w}_o(k-1)) + \frac{1}{\alpha_1}\widetilde{w}_1^T(k)\widetilde{w}_1(k) + \frac{1}{\alpha_2}\widetilde{w}_2^T(k)\widetilde{w}_2(k) \text{,(A.1)}$$

where $l_2 \in R^+$ is a design parameter. The first difference of Lyapunov function is given by

$$\Delta J(k) = \Delta J_1(k) + \Delta J_2(k) + \Delta J_3(k) + \Delta J_4(k) + \Delta J_5(k) + \Delta J_6(k), \text{ (A.2)}$$

The first term, $\Delta J_1(k)$, is obtained using (12) as

$$\Delta J_1(k) \le \frac{1}{2}\|\xi_o(k-1)\|^2 + \frac{1}{2}\|d_o(k-1)\|^2 + \frac{1}{4}\|\widetilde{x}(k-1)\|^2, \text{ (A.3)}$$

Now taking the second term in the first difference (A.1) and substituting (25) into (A.1), we get

$$\Delta J_2(k) \le \frac{k_2}{2}e_2^2(k) + \frac{1}{2}\xi_1^2(k) + \frac{1}{2}d_1^2(k) - \frac{l_2 e_1^2(k)}{6g_{1m}^2}, \text{ (A.4)}$$

Taking the third term in (A.1) and substituting the (36) into it and simplifying, we get

$$\Delta J_3(k) \le \frac{l_1^2}{2}e_1^2(k) + \frac{1}{2}\xi_2^2(k) + \frac{1}{2}d_2^2(k) - \frac{1}{6g_{2m}^2}e_2^2(k), \text{ (A.5)}$$

Taking the fourth term in (A.1) and substituting the (46) and simplifying, we get

$$\Delta J_4(k) = -(2 - \alpha_o\|\varphi(k-1)\|^2)\|\xi_o(k-1)\|^2$$

$$-(2 - \alpha_o\|\varphi(k-1)\|^2)\xi_o^T(k-1)(w_o^T\varphi(k-1) + l_1 e_1(k)I)$$

$$+ \alpha_o\|\varphi(k-1)\|^2\|w_o^T\varphi(k-1) + l_1 e_1(k)I\|^2, \text{ (A.6)}$$

The fifth term $\Delta J_5(k)$ is obtained using weight updating rule (47) as

$$\Delta J_5(k) = -2(1 - \alpha_1\|\phi(k)\|^2)\xi_1(k)(w_1^T\phi(k) + l_1 e_1(k))$$

$$-(2 - \alpha_1\|\phi(k)\|^2)\xi_1^2(k) + \alpha_1\|\phi(k)\|^2(w_1^T\phi(k) + l_1 e_1(k))^2, \text{ (A.7)}$$

Using (48), the last term $\Delta J_6(k)$ is expressed as

$$\Delta J_6(k) = -2(1 - \alpha_2\|\sigma(k)\|^2)\xi_2(k)(w_2^T\sigma(k) + l_1 e_1(k))$$

$$-(2 - \alpha_2\|\sigma(k)\|^2)\xi_2^2(k) + \alpha_2\|\sigma(k)\|^2(w_2^T\sigma(k) + l_1 e_1(k))^2, \text{ (A.8)}$$

Combining (A.3) through (A.8) to get the first difference and simplifying to get

$$\Delta J(k) = -\frac{1}{2}\|\xi_o(k-1)\|^2 - \frac{1}{2}\xi_1^2(k) - \frac{1}{2}\xi_2^2(k) + D_M^2$$

$$+ \frac{1}{4}\|\widetilde{x}(k-1)\|^2 - (\frac{l_2}{6g_{1m}^2} - \frac{13}{2}l_1^2)e_1^2(k) - (\frac{1}{6g_{2m}^2} - \frac{l_2}{2})e_2^2(k)$$

$$-(1 - \alpha_0\|\varphi(k-1)\|^2)\|\xi_o(k) - (w_o^T\varphi(k-1) + l_1 e_1(k)I)\|^2$$

$$-(1 - \alpha_1\|\phi(k)\|^2)(\xi_1(k) - (w_1^T\phi(k) + l_1 e_1(k)))^2$$

$$-(1 - \alpha_2\|\sigma(k)\|^2)(\xi_2(k) - (w_2^T\sigma(k) + l_1 e_1(k)))^2, \text{ (A.9)}$$

where

$$D_M^2 \le \frac{1}{2}(d_{om}^2 + d_{1m}^2 + d_{2m}^2) + 2(w_{om}\varphi_m + w_{1m}\phi_m + w_{2m}\sigma_m), \text{ (A. 10)}$$

This implies that $\Delta J(k) \le 0$ as long as (49) through (52) hold along with the following condition

$$0 < l_2 < \frac{1}{3g_{2m}^2}, \text{ (A.11)}$$

and

$$\|\widetilde{x}(k-1)\| > 2D_M, \text{ (A.12)}$$

or

$$|e_1(k)| > \frac{D_M}{\sqrt{\frac{l_2}{6g_{1m}^2} - \frac{13}{2}l_1^2}}, \text{ (A.13)}$$

or

$$|e_2(k)| > \frac{D_M}{\sqrt{\frac{1}{6g_{2m}^2} - \frac{l_2}{2}}}, \text{ (A.14)}$$

or

$$\|\xi_o(k-1)\| > \sqrt{2}D_M, \text{ (A.15)}$$

or

$$|\xi_1(k)| > \sqrt{2}D_M, \text{ (A.16)}$$

or

$$|\xi_2(k)| > \sqrt{2}D_M, \text{ (A.17)}$$

According to the standard Lyapunov extension theorem [7], this demonstrates that $\widetilde{x}(k-1), e_1(k), e_1(k)$ and the weight estimation errors are *UUB*. The boundedness of $\xi_o(k-1), \xi_1(k)$ and $\xi_2(k)$ implies that $\|\widetilde{w}_o(k)\|, \|\widetilde{w}_1(k)\|$ and $\|\widetilde{w}_2(k)\|$ are bounded, and this further implies that the weight estimates $\hat{w}_o$(k), $\hat{w}_1$(k) and $\hat{w}_2$(k) are bounded. Therefore all the closed-loop signals in the observer-controller system are bounded.

REFERENCE

[1] M. Krstic, I. Kanellakopoulos, and P. Kokotovic, *Nonlinear and Adaptive Control Design*, John Wiley & Sons, Inc., 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, 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, 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] Jr. Davis, C. S. Daw, L. A. Feldkamp, J. W. Hoard, F. Yuan and T. Connolly, "Method of controlling cyclic variation engine combustion", *U.S. Patent*, 5,921,221, 1999.

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

[7] F. L. Lewis, S. Jagannathan, and A. Yesildirek, *Neural Network Control of Robot Manipulators and Nonlinear Systems*, Taylor & Francis, PA, 1999.