Proceedings of the
44th IEEE Conference on Decision and Control, and
the European Control Conference 2005
Seville, Spain, December 12-15, 2005

Neural Network-based Control of Nonlinear Discrete-Time Systems in Non-Strict Form [2]

P. He, Z. Chen, and S. Jagannathan[1]

Abstract: **A novel reinforcement learning-based adaptive neural network (NN) controller, also referred as the adaptive-critic NN controller, is developed to deliver a desired tracking performance for a class of non-strict feedback nonlinear discrete-time systems in the presence of bounded and unknown disturbances. The adaptive critic NN controller architecture includes a critic NN and two action NNs. The critic NN approximates certain *strategic* utility function whereas the action neural networks are used to minimize both the *strategic* utility function and the unknown dynamics estimation errors. The NN weights are tuned online so as to minimize certain performance index. By using gradient descent-based novel weight updating rules, the uniformly ultimate boundedness (UUB) of the closed-loop tracking error and weight estimates is shown.**

I. INTRODUCTION

Adaptive NN backstepping control of nonlinear discrete-time systems in strict feedback form has been addressed in the literature [1-3]. These nonlinear systems are 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 $x_i(k)\in\Re$ is the state, $u(k)\in\Re$ is the control input, $\bar{x}_i(k)=[x_1(k),\cdots,x_i(k)]^T\in\Re^i$ and $i=1,\ldots,(n-1)$. 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., $\bar{x}_i(k)$. However, for non-strict feedback nonlinear system, where $f_i(\bar{x}_i(k))$ and $g_i(\bar{x}_i(k))$ depend upon both $\bar{x}_i(k)$ and $x_{i+1}(k)$, available [1-3] methods will result in a non-causal controller design (current control input depends on the future system states) when the adaptive NN backstepping approach is utilized.

Available NN controller designs employ either supervised training [4], where the user specifies a desired output, or online NN training based classical adaptive control [1-3] approach, where a short-term system performance measure using the tracking error is defined. On the other hand, reinforcement learning-based adaptive critic NN approach [5] has emerged as a promising tool to develop optimal NN controllers due to its potential to find approximate solutions to dynamic programming. Instead of finding the exact minimum, the adaptive critic designs try to approximate the Bellman equation: $Q(x(k))=\min_{u(k)}\{Q(x(k+1))+U(x(k),x(k+1))\}$, where $x(k)$ is the state and $u(k)$ is the control at time step k, the *strategic* utility function $Q(x(k))$ represents the minimum cost or performance measure associated with going from k to final step N, $U(x(k),x(k+1))$ is the utility function denoting the cost incurred in going from k to $k+1$ using control $u(k)$, and $Q(k+1)$ is the minimum cost or performance measure associated in going from state $k+1$ to the final step N. The critic NN monitors the system states, approximates the *strategic* utility function and tunes the action NN whereas the action NN generates a near optimal control input.

There are many variants of adaptive critic NN controller architectures [5-10]; however controller convergence is addressed only in a few works [7-10]. In [7], *hard computing techniques* were utilized to verify the stability of a class of nonlinear systems in continuous time. Both [8] and [9] study the convergence issue based on the recursive stochastic algorithms. In [10], the critic NN is used to approximate the Hamilton-Jacobi-Bellman (HJB) equation and the error convergence for a linear time-invariant discrete system is shown. However, NN controller results are not available for nonlinear discrete-time systems.

In this paper, a novel reinforcement learning-based adaptive critic NN controller is developed to control a class of nonlinear discrete-time systems in non-strict feedback form with bounded and unknown disturbances. Two action NNs are used to generate the virtual and actual control inputs respectively and their weights are tuned by both the critic NN signal and the tracking errors to minimize the *strategic* utility function and system dynamic estimation errors. The critic NN approximates a *strategic* utility function which is similar to the Bellman equation.

The main contributions of this paper can be summarized as follows: 1) optimization of certain long-term system performance index is undertaken here in contrast with traditional adaptive NN backstepping schemes [1-2]; 2) demonstration of the UUB of the overall system is shown even in the presence of NN approximation errors and bounded unknown disturbances unlike in the existing adaptive critic works [8-10]; 3) a well-defined controller is

[1] Authors are with Department of Electrical and Computer Engineering, University of Missouri-Rolla, Rolla, MO 65409. Email:sarangap@umr.edu.

[2] Research supported in part by NSF grant ECS #0327877.

0-7803-9568-9/05/$20.00 ©2005 IEEE

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 $g_i(\bar{x}_i(k))$ compared to [11]; 4) the NN weights are tuned online with no offline training phase [6]. The proposed controller is evaluated to control a spark ignition (SI) engine dynamics, a practical non-strict feedback nonlinear system.

II. BACKGROUND

A. Non-Strict Nonlinear System Description

Consider the following non-strict feedback nonlinear system described by

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

$$x_2(k+1) = f_2(x_1(k), x_2(k)) + g_2(x_1(k), x_2(k))u(k) + d_2(k), \quad (2)$$

where $x_i(k) \in \Re; i = 1,2$ are states, $u(k) \in \Re$ is the system input and $d_1(k) \in \Re$ and $d_2(k) \in \Re$ are unknown but bounded disturbances, whose bounds are given by $|d_1(k)| < d_{1m}$ and $|d_2(k)| < d_{2m}$ with d_{1m} and d_{2m} being unknown positive scalars.

III. ADAPTIVE CRITIC NN CONTROLLER DESIGN

Our objective is to design a NN controller for system (1) and (2) such that 1) all the signals in the closed-loop system remain *UUB*; 2) the state $x_1(k)$ follows a desired trajectory $x_{1d}(k)$; and 3) certain long-term system performance index is optimized.

Assumption 1: The desired trajectory $x_{1d}(k)$ is a smooth and bounded function over the compact $S \subset \Re$.

Assumption 2: The unknown smooth function, $g_2(\cdot)$, is assumed bounded away from zero within the compact set S, i.e.,

$$0 < g_{2\min} < |g_2(\cdot)| < g_{2\max},\ \forall x_1(k) \,\&\, x_2(k) \in S, \quad (3)$$

where $g_{2\min} \in \Re^+$ and $g_{2\max} \in \Re^+$. Without the loss generality, we will assume $g_2(.)$ is positive in this paper.

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

For simplicity, let us denote

$$f_1(k) = f_1(x_1(k), x_2(k)) + g_1(x_1(k), x_2(k))x_2(k) + x_2(k), \quad (4)$$

$$f_2(k) = f_2(x_1(k), x_2(k)), \quad (5)$$

and

$$g_2(k) = g_2(x_1(k), x_2(k)). \quad (6)$$

The system (1) and (2) can be rewritten as

$$x_1(k+1) = f_1(k) - x_2(k) + d_1(k), \quad (7)$$

$$x_2(k+1) = f_2(k) + g_2(k)u(k) + d_2(k). \quad (8)$$

Define the tracking error as

$$e_1(k) = x_1(k) - x_{1d}(k), \quad (9)$$

where $x_{1d}(k)$ is the desired trajectory. Using (7), (9) can be expressed as

$$\begin{aligned} e_1(k+1) &= x_1(k+1) - x_{1d}(k+1) \\ &= f_1(k) - x_2(k) - x_{1d}(k+1) + d_1(k). \end{aligned} \quad (10)$$

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

$$x_{2d}(k) = f_1(k) - x_{1d}(k+1) + l_1 e_1(k), \quad (11)$$

where $l_1 \in \Re$ is a design constant selected to stabilize the error system (10).

Since $f_1(k)$ is an unknown function, the desired virtual control input $x_{2d}(k)$ in (11) cannot be implemented in practice. By utilizing the first action NN to approximate this unknown function $f_1(k)$, $x_{2d}(k)$ is given by

$$\begin{aligned} x_{2d}(k) &= w_1^T \phi_1(v_1^T x(k)) + \varepsilon_1(x(k)) - x_{1d}(k+1) + l_1 e_1(k) \\ &= w_1^T \phi_1(k) + \varepsilon_1(x(k)) - x_{1d}(k+1) + l_1 e_1(k), \end{aligned} \quad (12)$$

where $x(k) = [x_1(k), x_2(k)]^T \in \Re^2$ is the input vector to the first action NN, $w_1 \in \Re^{n_1}$ and $v_1 \in \Re^{2\times n_1}$ denote the constant ideal output and hidden layer weights, the hidden layer activation function $\phi_1(k) \in \Re^{n_1}$ represents $\phi_1(v_1^T x(k))$, n_1 is the number of the nodes in the hidden layer, and $\varepsilon_1(x(k)) \in \Re$ is the approximation error. It is demonstrated in [12] that, if the hidden layer weights, v_1, is chosen initially at random and kept constant and the number of hidden layer nodes is sufficiently large, the approximation error $\varepsilon_1(x(k))$ can be made arbitrarily small so that the bound $\|\varepsilon_1(x(k))\| \le \varepsilon_{1m}$ holds for all $x(k) \in S$ since the activation function forms a basis.

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

$$\begin{aligned} \hat{x}_{2d}(k) &= \hat{w}_1^T(k)\phi_1(v_1^T x(k)) - x_{1d}(k+1) + l_1 e_1(k) \\ &= \hat{w}_1^T(k)\phi_1(k) - x_{1d}(k+1) + l_1 e_1(k), \end{aligned} \quad (13)$$

where $\hat{w}_1(k) \in \Re^{n_1}$ is the actual output layer weight matrix to be tuned. The hidden layer weight, v_1, is randomly chosen initially and kept constant [12]. Define the weight estimation error $\tilde{w}_1(k) \in \Re^{n_1}$ by

$$\tilde{w}_1(k) = \hat{w}_1(k) - w_1. \quad (14)$$

Define the error between $x_2(k)$ and $\hat{x}_{2d}(k)$ as $e_2(k) \in \Re$

$$e_2(k) = x_2(k) - \hat{x}_{2d}(k). \tag{15}$$

Equation (10) can be rewritten using (15) as

$$e_1(k+1) = f_1(k) - e_2(k) - \hat{x}_{2d}(k) + d_1(k) - x_{1d}(k+1), \tag{16}$$

Combining (13) with (16), we get

$$e_1(k+1) = f_1(k) - e_2(k) - \hat{w}_1^T(k)\phi_1(k) + d_1(k) - l_1 e_1(k), \tag{17}$$

or equivalently

$$e_1(k+1) = -l_1 e_1(k) - e_2(k) - \zeta_1(k) + \varepsilon_1(x(k)) + d_1(k), \tag{18}$$

where

$$\zeta_1(k) = \widetilde{w}_1^T(k)\phi(k), \tag{19}$$

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

Writing the error $e_2(k+1)$ from (15) as

$$\begin{aligned} e_2(k+1) &= x_2(k+1) - \hat{x}_{2d}(k+1), \\ &= f_2(k) + g_2(k)u(k) + d_2(k) - \hat{x}_{2d}(k+1), \end{aligned} \tag{20}$$

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

$$u_d(k) = \frac{1}{g_2(k)}\left(-f_2(k) + \hat{x}_{2d}(k+1) + l_2 e_2(k)\right), \tag{21}$$

where $l_2 \in \Re$ is the controller gain to stabilize the system (20). Note $u_d(k)$ depends upon future states since $\hat{x}_{2d}(k+1)$ depends upon the $x(k+1)$. We solve this non-causal problem by using the universal NN approximator. It can be clear that $\hat{x}_{2d}(k+1)$ is a nonlinear function of system state $x(k)$, virtual control input $\hat{x}_{2d}(k)$, desired trajectory $x_{1d}(k+2)$ and system errors $e_1(k)$ and $e_2(k)$. Therefore, $\hat{x}_{2d}(k+1)$ can be approximated using a NN. By taking $z(k) = [x_1(k), x_2(k), e_1(k), l_2 e_2(k), \hat{x}_{2d}(k), x_{1d}(k+2)]^T \in \Re^6$ as the input to the NN, $u_d(k)$ can be approximated as

$$u_d(k) = w_2^T \phi_2\left(v_2^T z(k)\right) + \varepsilon_2(z(k)) = w_2^T \phi_2(k) + \varepsilon_2(z(k)), \tag{22}$$

where $w_1 \in \Re^{n_2}$ and $v_1 \in \Re^{6\times n_2}$ denote the constant ideal output and hidden layer weights, the hidden layer activation function $\phi_2(k) \in \Re^{n_2}$ represents $\phi_2\left(v_2^T z(k)\right)$, n_2 is the number of the nodes in the hidden layer, and $\varepsilon_2(z(k)) \in \Re$ is the approximation error.

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

$$u(k) = \hat{w}_2^T(k)\phi_2\left(v_2^T z(k)\right) = \hat{w}_2^T(k)\phi_2(k), \tag{23}$$

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

Substituting (21) through (23) into (20) , we get

$$\begin{aligned} e_2(k+1) &= f_2(k) + g_2(k)\left(\hat{w}_2^T(k)\phi_2(k)\right) + d_2(k) - \hat{x}_{2d}(k+1) \\ &= f_2(k) + g_2(k)\left(w_2^T(k)\phi_2(k)\right) + g_2(k)\zeta_2(k) + d_2(k) - \hat{x}_{2d}(k+1) \\ &= f_2(k) + g_2(k)\left(u_d(k) - \varepsilon_2(z(k))\right) + g_2(k)\zeta_2(k) + d_2(k) - \hat{x}_{2d}(k+1) \\ &= l_2 e_2(k) + g_2(k)\zeta_2(k) - g_2(k)\varepsilon_2(z(k)) + d_2(k), \end{aligned} \tag{24}$$

where

$$\zeta_2(k) = \hat{w}_2^T(k)\phi_2(k) - w_2^T\phi_2(k), \tag{25}$$

Equations (18) and (24) represent the closed-loop error dynamics. The next step is to design the adaptive critic NN controller weight updating rules.

IV. NN Weight Updating Rules

The critic NN is trained online to approximate the *strategic* utility function (long-term system performance index). Then the critic signal, with a potential for estimating the future system performance, is employed to tune the two action NNs to minimize the strategic utility function and the unknown system estimation errors so that closed-loop stability is inferred.

A. The Strategic Utility Function

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

$$p(k) = \begin{cases} 0, & if\left(\left|e_1(k)\right| + \left|e_2(k)\right|\right) \le c \\ 1, & otherwise \end{cases}, \tag{26}$$

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

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

$$Q(k) = \alpha^N p(k+1) + \alpha^{N-1} p(k+2) + \cdots + \alpha^{k+1} p(N), \tag{27}$$

where $\alpha \in \Re$ and $0 < \alpha < 1$, and N is the final time instant. The term $Q(k)$ is viewed here as the future system performance measure. This measure is similar to the Bellman equation.

B. Design of the Critic NN

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

$$e_c(k) = \hat{Q}(k) - \alpha\left(\hat{Q}(k-1) - \alpha^N p(k)\right), \tag{28}$$

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

$$\hat{Q}(k) = \hat{w}_3^T(k)\phi_3\left(v_3^T x(k)\right) = \hat{w}_3^T(k)\phi_3(k), \tag{29}$$

and $\hat{Q}(k) \in \Re$ is the critic signal, $\hat{w}_3(k) \in \Re^{n_3}$ and $v_3 \in \Re^{2\times n_3}$ represent the matrix of weight estimates,

$\phi_3(k) \in \Re^{n_3}$ is the activation function vector in the hidden layer, n_3 is the number of the nodes in the hidden layer, and the critic NN input is given by $x(k) \in \Re^2$. The objective function to be minimized by the critic NN is defined as

$$E_c(k) = \frac{1}{2} e_c^2(k). \tag{30}$$

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

$$\hat{w}_3(k+1) = \hat{w}_3(k) + \Delta \hat{w}_3(k), \tag{31}$$

where

$$\Delta \hat{w}_3(k) = \alpha_3 \left[-\frac{\partial E_c(k)}{\partial \hat{w}_3(k)} \right], \tag{32}$$

or

$$\hat{w}_3(k+1) = \hat{w}_3(k) - \alpha_3 \phi_3(k) \left(\hat{Q}(k) + \alpha^{N+1} p(k) - \alpha \hat{Q}(k-1) \right)^T, \tag{33}$$

where $\alpha_3 \in \Re$ is the NN adaptation gain.

C. Weight Updating Rule for the First Action NN

The first action NN $\hat{w}_1^T(k)\phi_1(k)$ weight is tuned by using the functional estimation error, $\zeta_1(k)$, and the error between the desired *strategic* utility function $Q_d(k) \in \Re$ and the critic signal $\hat{Q}(k)$. Define

$$e_{a1}(k) = \zeta_1(k) + (\hat{Q}(k) - Q_d(k)), \tag{34}$$

where $\zeta_1(k)$ is defined in (19), $e_{a1}(k) \in \Re$, and the subscript "a1" stands for the "first action NN".

The value for the desired *strategic* utility function $Q_d(k)$ is taken as "0" [9], i.e., to indicate that at every step, the nonlinear system can track the reference signal well. Thus, (34) becomes

$$e_{a1}(k) = \zeta_1(k) + \hat{Q}(k), \tag{35}$$

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

$$E_{a1}(k) = \frac{1}{2} e_{a1}^2(k), \tag{36}$$

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

$$\hat{w}_1(k+1) = \hat{w}_1(k) + \Delta \hat{w}_1(k), \tag{37}$$

where

$$\Delta \hat{w}_1(k) = \alpha_1 \left[-\frac{\partial E_{a1}(k)}{\partial \hat{w}_1(k)} \right], \tag{38}$$

or

$$\hat{w}_1(k+1) = \hat{w}_1(k) - \alpha_1 \phi_1(k) \left(\hat{Q}(k) + \zeta_1(k) \right), \tag{39}$$

where $\alpha_2 \in \Re$ is the NN adaptation gain.

The NN weight updating rule in (39) cannot be implemented in practice since the target weight w_1 is unknown. However, using (18), the functional estimation error $\zeta_1(k)$ is given by

$$\zeta_1(k) = -e_1(k+1) - l_1 e_1(k) - e_2(k) + \varepsilon_1(x(k)) + d_1(k). \tag{40}$$

Substituting (40) into (39), we get

$$\hat{w}_1(k+1) = \hat{w}_1(k) - \alpha_1 \phi_1(k) \hat{Q}(k)$$
$$- \alpha_1 \phi_1(k) \left(-e_1(k+1) - l_1 e_1(k) - e_2(k) + \varepsilon_1(x(k)) + d_1(k) \right). \tag{41}$$

Assume that bounded disturbance $d_1(k)$ and the NN approximation error $\varepsilon_1(x(k))$ are zeros for weight tuning implementation, then (41) is rewritten as

$$\hat{w}_1(k+1) = \hat{w}_1(k) - \alpha_1 \phi_1(k) \hat{Q}(k) + \alpha_1 \phi_1(k) \left(e_1(k+1) + l_1 e_1(k) + e_2(k) \right). \tag{42}$$

Equation (42) is the adaptive critic based weight updating rule for the first action NN, $\hat{w}_1^T(k)\phi_1(k)$. Similarly, we could derive the weight updating rule for the second action NN $\hat{w}_2^T(k)\phi_2(k)$ as given next.

D. Weight Updating Rule for the Second Action NN

Define

$$e_{a2}(k) = \sqrt{g_2(k)} \zeta_2(k) + \frac{\hat{Q}(k)}{\sqrt{g_2(k)}}, \tag{43}$$

where $\zeta_2(k)$ is defined in (25), $g_2(k) \in \Re^+$ and $e_{a2}(k) \in \Re$, the subscript "a2" stands for the "second action NN". Following the similar design procedure and taking the bounded unknown disturbance $d_2(k)$ and the NN approximation error $\varepsilon_2(z(k))$ to be zeros, the second action NN $\hat{w}_2^T(k)\phi_2(k)$ weight updating rule is given by

$$\hat{w}_2(k+1) = \hat{w}_2(k) - \alpha_2 \phi_2(k) \left(\hat{Q}(k) + e_2(k+1) - l_2 e_2(k) \right), \tag{44}$$

To implement (44), the value of $e_2(k+1)$ at the k instant must be known and it is given by the following steps:

1) Calculate $u(k)$, $\hat{x}_{2d}(k)$, $\hat{Q}(k)$, $e_1(k)$, $e_2(k)$, $z(k)$, $\hat{w}_1(k)$ and $\hat{w}_1(k)$ at the k instant;

2) Apply the control input $u(k)$ to the system (1) and (2) to obtain the states $x_1(k+1)$ and $x_2(k+1)$;

3) Using the tracking error definition to get $e_1(k+1)$ as

$$e_1(k+1) = x_1(k+1) - x_{1d}(k+1), \tag{45}$$

4) Using first action NN weight updating rule (42) to get $\hat{w}_1(k+1)$;

5) Once we have $\hat{w}_1(k+1)$ and $x(k+1)$, the value of $\hat{x}_{2d}(k+1)$ can be determined by using

$$\hat{x}_{2d}(k+1) = \hat{w}_1^T(k+1)\phi_1(k+1) - x_{1d}(k+2) + l_1 e_1(k+1), \tag{46}$$

and

$$e_2(k+1) = x_2(k+1) - \hat{x}_{2d}(k+1). \tag{47}$$

The NN controller structure is shown in Fig. 1.

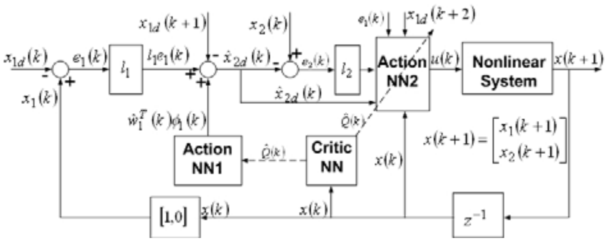

Fig. 1. Adaptive critic NN-based controller.

V. MAIN RESULT

Assumption 3 (Bounded Ideal Weights): Let w_1, w_2 and w_3 be the unknown output layer target weights for the two action NNs and the critic NN and assume that they are bounded above so that

$$\|w_1\| \le w_{1m}, \|w_2\| \le w_{2m}, \text{ and } \|w_3\| \le w_{3m}, \quad (48)$$

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

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

$$\|\phi_i(k)\| \le \phi_{im}, \quad i = 1,2,3, \quad (49)$$

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

Assumption 4 (Bounded NN Approximation Error): The NN reconstruction errors $\varepsilon_1(x(k))$ and $\varepsilon_2(z(k))$ are bounded over the compact set $S \subset \Re$ by ε_{1m} and ε_{2m}, respectively [12].

Theorem 1: Consider the system given by (1) and (2), let the Assumptions 1 through 4 hold, and the disturbance bounds d_{1m} and d_{2m} be known constants. Let the critic NN $\hat{w}_3^T(k)\phi_3(k)$ weight tuning be given by (33), the first action NN $\hat{w}_1^T(k)\phi_1(k)$ weight tuning provided by (42) and the second action NN $\hat{w}_2^T(k)\phi_2(k)$ weight tuning provided by (44). Given the virtual control input $\hat{x}_{2d}(k)$ (13) and the control input $u(k)$ (23), the tracking errors, $e_1(k)$ and $e_2(k)$, 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.15) through (A.17) provided the controller design parameters are:

$$\text{(a) } 0 < \alpha_1 \|\phi_1(k)\|^2 < 1, \quad (50)$$

$$\text{(b) } 0 < \alpha_2 \|\phi_2(k)\|^2 < \frac{1}{g_{2\max}}, \quad (51)$$

$$\text{(c) } 0 < \alpha_2 \|\phi_3(k)\|^2 < 1, \quad (52)$$

$$\text{(d) } |l_1| < \frac{1}{2}, \quad (53)$$

$$\text{(e) } |l_2| < \frac{\sqrt{3}}{3}, \quad (54)$$

$$\text{(f) } 0 < \alpha < \frac{\sqrt{2}}{2}, \quad (55)$$

where α_1, α_2 and α_3 are NN adaptation gains, l_1 and l_2 are controller gains, α is employed to define the *strategic* utility function. ■

Remark 1: The weights of the action and critic NNs can be initialized at zero or random. This means that there is no explicit off-line learning phase needed.

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

Remark 3: Condition (50) can be verified easily. For instance, the hidden layer of the critic NN consists of n_1 nodes with the hyperbolic tangent sigmoid function as its activation function, then $\|\phi_1(\cdot)\|^2 \le n_1$. The NN adaptation gain α_1 can be selected as $0 < \alpha_1 < \frac{1}{n_1}$ to satisfy (50).

Corollary 1: Given the hypothesis in *Theorem 1* with the parameter selection based on (50) through (55), the state $x_2(k)$ approaches the desired virtual control input $x_{2d}(k)$.

Proof: Combining (12) and (13), the difference between $\hat{x}_{2d}(k)$ and $x_{2d}(k)$ is given by

$$\hat{x}_{2d}(k) - x_{2d}(k) = \tilde{w}_1(k)\phi_1(k) - \varepsilon_1(x(k)) = \zeta_1(k) - \varepsilon_1(x(k)), \quad (56)$$

where $\tilde{w}_1(k) \in \Re^{n_1}$ defined in (14) is the first action NN weight estimation error and $\zeta_1(k) \in \Re$ is defined in (19). Since both $\zeta_1(k) \in \Re$ and $\varepsilon_1(x(k))$ are bounded, $\hat{x}_{2d}(k)$ is bounded to $x_{2d}(k)$. In *Theorem 1*, we show that $e_2(k)$ is bounded, i.e., the state $x_2(k)$ is bounded to the virtual control signal $\hat{x}_{2d}(k)$. Thus the state $x_2(k)$ is bounded to the desired virtual control signal $x_{2d}(k)$. ■

VI. SIMULATION

Lean operation of SI engine allows low emissions and improved fuel efficiency. However, at lean operation, the engine exhibits cyclic dispersion of heat release which degrades performance. The adaptive critic NN controller is designed to stabilize the SI engine operating at lean conditions. The engine dynamics can be expressed as non-strict feedback nonlinear system of the form [13]:

$$x_1(k+1) = (1 - F(k))AF + F(k)x_1(k) - R \cdot F(k)CE(k)x_2(k) \quad (57)$$

$$x_2(k+1) = (1 - CE(k))F(k)x_2(k) + (MF(k) + u(k)), \quad (58)$$

$$CE(k) = \frac{CE_{\max}}{1 + 100^{-(\varphi(k) - \varphi_m)/(\varphi_u - \varphi_l)}}, \quad \varphi(k) = R\frac{x_2(k)}{x_1(k)}, \quad (59)$$

$$\varphi_m = \frac{\varphi_u + \varphi_l}{2}, \quad H(k) = x_2(k) CE(k), \qquad (60)$$

where $x_1(k)$ and $x_2(k)$ are the mass of air and fuel before *kth* burn respectively, *F(k)* is the residual gas fraction*, AF* is the mass of fresh air fed per cycle, *R* is the stoichiometric air-fuel ratio, ~14.6, *CE(k)* is the combustion efficiency, *MF(k)* is the mass of fresh fuel per cycle, *u(k)* is the small changes in mass of fresh fuel per cycle. $CE_{\max}$ is the maximum combustion efficiency and it is a constant, $\varphi(k)$ is the equivalence ratio, $\varphi_m, \varphi_u, \varphi_l$ are constant system parameters, and $H(k)$ is the heat release in the *kth* cycle. Since $H(k)$ varies cycle by cycle, the engine exhibits misfire and unsatisfactory behavior. In (56) and (57), *F(k)* and *CE(k)* are unknown nonlinear functions of both $x_1(k)$ and $x_2(k)$.

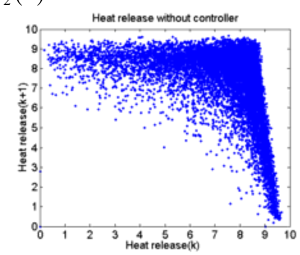

Fig. 2. Cyclic dispersion without control.

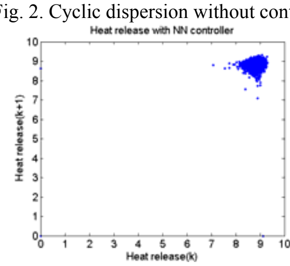

Fig. 3. Heat release with NN controller.

Given *Theorem 1*, *Corollary 1* and using the proof in [14], we could show that, with the proposed controller, both states can be bounded to their respective target values x_{1d} and x_{2d}. Then the equivalence ratio $\varphi(k)$ (59) combustion efficiency $CE(k)$ (59), heat release $H(k)$ (60) and the engine dynamics are stabilized.

The system parameters are selected as the following: $\varphi = 0.71$, $F = 0.14$, $AF = 2222$, $MF = 10.81$, $\varphi_u = 0.685$, $\varphi_l = 0.665$, $CE_{\max} = 0.9$, $x_{1d} = 201.2$, $x_{2d} = 9.882$. We add the unknown white noise with the deviation of 0.1081 and 0.007 to the *F* and *MF*.

The controller gains are selected as $l_1 = l_2 = 0.1$. For weight updating, the adaptation gains are selected as $\alpha_1 = 0.01$, $\alpha_2 = 0.001$, and $\alpha_3 = 0.01$. All the three NNs have 15 hidden layer nodes each. All the hidden layer weights are selected uniformly within an interval of [0, 1] and all the activation functions are selected as hyperbolic tangent sigmoid functions.

The cyclic dispersion observed at a lean equivalence ratio of 0.71 is presented in Fig. 2 when no control scheme is employed. It shows that the engine exhibits misfires which is a problem. Fig. 3 illustrates the NN controller performance where the dispersion is tolerable.

VII. Conclusions

This paper proposed a novel adaptive critic NN controller to deliver a desired tracking performance for a class of discrete-time non-strict feedback nonlinear systems. A well-defined controller was developed by using two action NNs for generating suitable control input, and a NN critic signal for approximating the *strategic* utility function. By using gradient-based online weight tuning, the stability of the closed-loop system was demonstrated. The controller performance was shown on a practical nonlinear system.

References

[1] M. Krstic, I. Kanellakopoulos, and P. Kokotovic, *Nonlinear and Adaptive Control Design*, John Wiley & Sons, Inc., 1995.

[2] 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.

[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] K. S. Narendra and K. S. Parthasarathy, "Identification and control of dynamical systems using neural networks," *IEEE Trans. Neural Networks*, vol. 1, no. 1, pp. 4-27, 1990.

[5] J. Si, *NSF Workshop on Learning and Approximate Dynamic Programming*, Playacar, Mexico, 2002. Available: http://ebrains.la.asu.edu/~nsfadp/.

[6] P. J. Werbos, "Neurocontrol and supervised learning: An overview and evaluation", *Handbook of Intelligent Control*, edited by D. A. White and D. A. Sofge, Van Nostrand Reinhold, New York, pp. 65-90, 1992.

[7] 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, 2002.

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

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

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

[11] F. L. Lewis, S. Jagannathan and A. Yesilderek, *Neural Network control of robot manipulators and nonlinear systems*, Taylor and Francis, UK, 1999.

[12] 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.

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

[14] P. He and S. Jagannanthan, "Neuro emission controller for minimizing cyclic dispersion in spark ignition engines", *Proc. Int. Joint Conf. Neural Networks*, vol. 2, 2003, pp. 1535 – 1540, .