01 Jul 2007

Reinforcement Learning Based Output-Feedback Control of Nonlinear Nonstrict Feedback Discrete-Time Systems with Application to Engines

Peter Shih

Jonathan B. Vance

Brian C. Kaul

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

et. al. For a complete list of authors, see https://scholarsmine.mst.edu/ele_comeng_facwork/1244

Recommended Citation

P. Shih et al., "Reinforcement Learning Based Output-Feedback Control of Nonlinear Nonstrict Feedback Discrete-Time Systems with Application to Engines," *Proceedings of the American Control Conference, 2007. ACC'07*, Institute of Electrical and Electronics Engineers (IEEE), Jul 2007.
The definitive version is available at https://doi.org/10.1109/ACC.2007.4283127

Proceedings of the 2007 American Control Conference
Marriott Marquis Hotel at Times Square
New York City, USA, July 11-13, 2007

Reinforcement Learning based Output-Feedback Control of Nonlinear Nonstrict Feedback Discrete-time Systems with Application to Engines

Peter Shih, J. Vance, B. Kaul, S. Jagannathan, and James A. Drallmeier

***Abstract*— A novel reinforcement-learning based output-adaptive neural network (NN) controller, also referred as the adaptive-critic NN controller, is developed to track a desired trajectory for a class of complex nonlinear discrete-time systems in the presence of bounded and unknown disturbances. The controller includes an observer for estimating states and the outputs, critic, and two action NNs for generating virtual, and actual control inputs. The critic approximates certain *strategic* utility function and the action NNs are used to minimize both the strategic utility function and their outputs. All NN weights adapt online towards minimization of a performance index, utilizing gradient-descent based rule. A Lyapunov function proves the uniformly ultimate boundedness (UUB) of the closed-loop tracking error, weight, and observer estimation. Separation principle and certainty equivalence principles are relaxed; persistency of excitation condition and linear in the unknown parameter assumption is not needed. The performance of this adaptive critic NN controller is evaluated through simulation with the Daw engine model in lean mode. The objective is to reduce the cyclic dispersion in heat release by using the controller.**

I. Introduction

Adaptive NN backstepping control of nonlinear discrete-time systems in strict feedback form has been addressed in the literature [1-3]. Available methods [1-3] result in a non-causal controller and optimization is not carried out. The controller designs employ either supervised training, where the user specifies a desired output, or classical online training [1-3], where a short-term system performance measure is defined by using the tracking error. By contrast, the reinforcement-learning based adaptive critic NN approach [4] has emerged as a promising tool to develop optimal NN controllers due to its potential to find approximate solutions to dynamic programming, where a strategic utility function (a long-term system performance measure) can be optimized. There are many variants of adaptive critic NN controller architectures [4-7] using state feedback even though few results [6, 7] address the controller convergence.

In this paper, a novel adaptive critic NN-based output feedback controller is developed to control a class of nonlinear non-strict feedback discrete-time system. Adaptive NN backstepping is utilized for the controller design with two action NNs being used to generate the virtual and actual control inputs, respectively. The two action NN weights are tuned by the critic NN signal to minimize the *strategic* utility function and their outputs. The critic NN approximates certain *strategic* utility function which is a variant of Bellman equation. The NN observer estimates the states and output, which are used in the controller design. The proposed controller is *model–free* since the NN weights are tuned online to approximate the unknown system dynamics.

The main contributions of this paper can be summarized as follows: 1) the adaptive NN back-stepping scheme is extended to non-strict feedback nonlinear systems. The non-causal problem is overcome by employing the universal NN approximation property; 2) optimization of a long-term performance index is undertaken in contrast with traditional adaptive NN back stepping schemes [1, 2]; 3) demonstration of the UUB of the system is shown in the presence of approximation errors and bounded unknown disturbances unlike existing adaptive critic works [7]. Stability proof is inferred by relaxing separation principle via novel weight updating rules and by selecting the Lyapunov function consisting of the system estimation errors, tracking and the NN weight estimation errors. A single critic NN is utilized to tune two action NNs; 4) a well-defined controller is presented by overcoming the problem of certain nonlinear function estimate 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 [8]; 5) the NN weights are tuned online instead of offline [5]; and finally 6) the assumption that $g_1(x_1(k), x_2(k))$ is bounded away from zero and its sign is known *a priori* is relaxed in contrast with [2].

The proposed controller is evaluated to control the spark ignition (SI) engine dynamics, a practical non-strict feedback nonlinear system, to reduce cyclic dispersion during lean operation using the Daw model [9].

II. Non-linear non-strict feedback system

Consider the nonlinear discrete-time system

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

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

$$y(k+1) = f_3(x_1(k), x_2(k)) \quad (3)$$

where $x_i(k) \in \Re; i = 1,2$ are the states, $u(k) \in \Re$ is the system input, and $d_i(k) \in \Re, i\{1,2\}$ are unknown but bounded disturbances. Bounds on these disturbances are given by $|d_i(k)| < d_{im}, i\{1,2\}$ where $d_{im}, i\{1,2\}$ are unknown positive scalars. The output is a nonlinear function of states in contrast

This work is supported in part by NSF grants ECCS#0327877 and ECCS#0621924. Peter Shih, J. Vance, and S. Jagannathan are with the Department of Electrical and Computer Engineering, B. Kaul and J. A. Drallmeier are with the Department of Mechanical Engineering at the University of Missouri-Rolla (contact author's e-mail: sarangap@umr.edu). The authors thank Qinmin Yang for his assistance in this proof.

with available literature [10, 11] where the output linear. Finally, the output is measurable whereas the states are not except when the states converge to their respective desired values the actual output converges to its desired value.

III. Observer Design

To overcome immeasurable states, an observer is used. It utilize the current heat release output, $y(k)$, to estimate the future output $\hat{y}(k+1)$ and states $\hat{x}_1(k+1)$ and $\hat{x}_2(k+1)$.

A. Observer Design

Consider equations (1) and (2). We expand the individual nonlinear functions using Taylor series expansion into two terms: linear and higher order terms.

$$f_1(\cdot)=f_{10}+\Delta f_1(\cdot) \tag{4}$$

$$f_2(\cdot)=f_{20}+\Delta f_2(\cdot) \tag{5}$$

$$g_1(\cdot)=g_{10}+\Delta g_1(\cdot) \tag{6}$$

$$g_2(\cdot)=g_{20}+\Delta g_2(\cdot) \tag{7}$$

where the first term in (4) through (7) are known nominal values and the second term are unknown higher order terms. We use a two-layer feed-forward NN with novel weight tuning to construct a NN with semi-recurrent architecture.

$$y(k+1)=w_1^T\phi\left(v_1^T z_1(k)\right)+\varepsilon\left(z_1(k)\right), \tag{8}$$

where $z_1(k)=\left[x_1(k),x_2(k),y(k),u(k)\right]^T\in R^4$ is the network input, $y(k+1)$ and $y(k)$ are the future and current output, $w_1\in\Re^{n_1}$ and $v_1\in\Re^{2\times n_1}$ denote the ideal output and constant hidden layer weights respectively, $u(k)$ is the control input, $\phi\left(v_1^T z_1(k)\right)$ represents the hidden layer activation function, n_1 is the number of nodes in the hidden layer, and $\varepsilon\left(z_1(k)\right)\in\Re$ is the approximation error. For simplicity we define the following two equations

$$\phi_1(k)=\phi\left(v_1^T z_1(k)\right) \tag{9}$$

$$\varepsilon_1(k)=\varepsilon\left(z_1(k)\right) \tag{10}$$

Rewrite (8) using (9) and (10)

$$y(k+1)=w_1^T\phi_1(k)+\varepsilon_1(k) \tag{11}$$

The states $x_1(k)$ and $x_2(k)$ are not measurable, therefore, $z_1(k)$ is not available either. Using the estimated states and output $\hat{x}_1(k)$, $\hat{x}_2(k)$, and $\hat{y}(k)$ instead of $x_1(k)$, $x_2(k)$, and $y(k)$, the proposed observer is given as

$$\begin{aligned}\hat{y}(k+1)&=\hat{w}_1^T(k)\phi\left(v_1^T\hat{z}_1(k)\right)+l_1\tilde{y}(k)\\&=\hat{w}_1^T(k)\hat{\phi}_1(k)+l_1\tilde{y}(k)\end{aligned} \tag{12}$$

where $\hat{z}_1(k)=\left[\hat{x}_1(k),\hat{x}_2(k),\hat{y}(k),\hat{u}(k)\right]^T\in R^4$ is the input using estimated states, $\hat{y}(k+1)$ and $\hat{y}(k)$ are the estimated future and current output, $\hat{w}_1(k)$ is the actual weight matrix, $\hat{u}(k)$ is the estimate control input, $\hat{\phi}_1(k)$ is the hidden layer activation function, $l_1\in R$ is the observer gain, and $\tilde{y}(k)$ is the heat release estimation error where

$$\tilde{y}(k)=\hat{y}(k)-y(k) \tag{13}$$

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 $\varepsilon\left(z_1(k)\right)$ can be made arbitrarily small so that the bound $\left\|\varepsilon\left(z_1(k)\right)\right\|\le\varepsilon_{1m}$ holds for all $z_1(k)\in S$ since the activation function forms a basis. Now we choose, at our convenience, the observer structure as a function of output estimation errors and known quantities as

$$\hat{x}_1(k+1)=f_{10}-\hat{x}_2(k)+l_2\tilde{y}(k) \tag{14}$$

$$\hat{x}_2(k+1)=f_{20}+g_{20}u(k)+l_3\tilde{y}(k) \tag{15}$$

where $l_2\in R$ and $l_3\in R$ are design constants.

B. Observer Error Dynamics

Let us define the state estimation and output errors as

$$\tilde{x}_i(k+1)=\hat{x}_i(k+1)-x_i(k+1), i\in\{1,2\} \tag{16}$$

$$\tilde{y}(k+1)=\hat{y}(k+1)-y(k+1) \tag{17}$$

Combining (1) through (8) and, (14) through (17), we obtain the estimation error dynamics as

$$\tilde{x}_1(k+1)=f_{10}-\hat{x}_2(k)+l_2\tilde{y}(k)-f_1(\cdot)-g_1(\cdot)x_2(k)-d_1(k) \tag{18}$$

$$\tilde{x}_2(k+1)=f_{20}+g_{20}u(k)+l_3\tilde{y}(k)-f_2(\cdot)-g_2(\cdot)u(k)-d_2(k) \tag{19}$$

$$\tilde{y}(k+1)=\hat{w}_1^T(k)\hat{\phi}_1(k)+l_1\tilde{y}(k)-w_1^T\phi_1(k)-\varepsilon_1(k) \tag{20}$$

where

$$\tilde{\phi}_1(k)=\phi\left(v_1\hat{z}_1(k)\right)-\phi\left(v_1 z_1(k)\right) \tag{21}$$

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

$$\zeta_1(k)=\tilde{w}_1^T(k)\hat{\phi}_1(k) \tag{23}$$

Choose the weight tuning of the observer as

$$\hat{w}_1(k+1)=\hat{w}_1(k)-\alpha_1\hat{\phi}_1(k)\left(\hat{w}_1^T(k)\hat{\phi}_1(k)+l_4\tilde{y}(k)\right) \tag{24}$$

where $\alpha_1\in R$, and $l_4\in R$ are design constants. It will be shown that by using the above weight tuning, separation principle is relaxed and the closed-loop signals will be bounded. Next we design the adaptive critic controller.

IV. Critic Design

The purpose of the critic NN is to approximate the long-term performance index (strategic utility function) of the nonlinear system through online weight adaptation. The critic signal estimates the future performance and tunes the two action NNs. The tuning will ultimately minimize the strategic utility function itself and estimation errors so that closed-loop stability is inferred.

A. The Strategic Utility Function

The utility function $p(k)\in\Re$ is given by

$$p(k)=\begin{cases}0, & if\left(\left\|\tilde{y}(k)\right\|\right)\le c\\1, & otherwise\end{cases} \tag{25}$$

where $c\in\Re$ is a user-defined threshold. The utility function $p(k)$ represents the current performance index. The term $p(k)=0$ and $p(k)=1$ refers to good and poor tracking performance at the k[th] time step respectively. The long-term *strategic* utility function $Q(k)\in\Re$, is defined as

$$Q(k)=\beta^N p(k+1)+\beta^{N-1}p(k+2)+\cdots+\beta^{k+1}p(N)+.., \tag{26}$$

where $\beta\in\Re$ and $0<\beta<1$ is the discount factor and N is the horizon index.

B. Design of the Critic NN

We utilize the universal approximation property of NN to define the critic NN output, $\hat{Q}(k)$ as

$$\hat{Q}(k)=\hat{w}_2^T(k)\phi\left(v_2^T\hat{z}_2(k)\right)=\hat{w}_2^T(k)\hat{\phi}_2(k) \tag{27}$$

where $\hat{Q}(k)\in\Re$ is the critic signal, $\hat{w}_2(k)\in\Re^{n_2}$ is the tunable weight, $v_2\in\Re^{2\times n_2}$ represent the constant input weight matrix, $\hat{\phi}_2(k)\in\Re^{n_2}$ is the activation function vector in the hidden layer, n_2 is the number of the nodes in the hidden layer, and $\hat{z}_2(k)=[\hat{x}_1(k),\hat{x}_2(k)]^T\in R^2$ is the input.

C. Critic Weight Update Law

We define the prediction error as

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

where the subscript "c" stands for the "critic." We use a quadratic objective function to minimize, defined as

$$E_c(k)=\frac{1}{2}e_c^2(k) \tag{29}$$

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

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

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

$$\hat{w}_2(k+1)=\hat{w}_2(k)-\alpha_2\hat{\phi}_2(k)\left(\hat{Q}(k)+\beta^{N+1}p(k)-\beta\hat{Q}(k-1)\right)^T \tag{32}$$

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

V. Design of the Virtual Control Input

Next the following mild assumption is needed.

Assumption 1: The unknown function $g_2(\cdot)$ is smooth and assumed bounded away from zero for all $x_1(k)$ and $x_2(k)$ within the compact set S.

A. System Simplification

First, we simplify by rewriting the state equations with the following

$$\Phi(\cdot)=f_1\left(x_1(k),x_2(k)\right)+g_1\left(x_1(k),x_2(k)\right)x_2(k)+x_2(k) \tag{33}$$

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

$$x_1(k+1)=\Phi(\cdot)-x_2(k)+d_1(k) \tag{34}$$

$$x_2(k+1)=f_2(\cdot)+g_2(\cdot)u(k)+d_2(k) \tag{35}$$

B. Virtual Control Input Design

Our goal is to stabilize the system output $y(k)$ around a specified target point, y_d by controlling the input. Because the output is only dependent on the states $x_1(k)$ and $x_2(k)$, as they approach the desired trajectory $x_{1d}(k)$ and $x_{2d}(k)$, the target output is reached. At the same time, all signals in systems (1) and (2) must be UUB; all weights must be bounded; and a performance index must be minimized.

Define the tracking error as

$$e_1(k)=x_1(k)-x_{1d}(k) \tag{36}$$

where $x_{1d}(k)$ is the desired trajectory. Using (34), (36) can be expressed as the following

$$\begin{aligned}e_1(k+1)&=x_1(k+1)-x_{1d}(k+1)\\&=\Phi(\cdot)-x_2(k)-x_{1d}(k+1)+d_1(k)\end{aligned} \tag{37}$$

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

$$x_{2d}(k)=\Phi(\cdot)-x_{1d}(k+1)+l_5e_1(k) \tag{38}$$

where l_5 is a gain constant. Since $\Phi(\cdot)$ is an unknown function, $x_{2d}(k)$ in (38) cannot be implemented in practice. We invoke the universal approximation property of NN to estimate this unknown function.

$$\Phi(\cdot)=w_3^T\phi\left(v_3^Tz_3(k)\right)+\varepsilon\left(z_3(k)\right) \tag{39}$$

where $z_3(k)=\left[x_1(k),x_2(k)\right]^T\in\Re^2$ is the input vector, $w_3^T\in\Re^{n_2}$ and $v_3^T\in\Re^{2\times n_3}$ are the ideal and constant input weight matrix, $\phi\left(v_3^Tz_3(k)\right)\in\Re^{n_3}$ is the activation function vector in the hidden layer, n_3 is the number of the nodes in the hidden layer, and $\varepsilon\left(z_3(k)\right)$ is the estimation error.

Rewrite (38) using (39)

$$x_{2d}(k)=w_3^T\phi\left(v_3^Tz_3(k)\right)+\varepsilon\left(z_3(k)\right)-x_{1d}(k+1)+l_5e_1(k) \tag{40}$$

Replacing actual with estimated states, (40) becomes

$$\begin{aligned}\hat{x}_{2d}(k)&=\hat{w}_3^T(k)\phi\left(v_3^T\hat{z}_3(k)\right)-x_{1d}(k+1)+l_5\hat{e}_1(k)\\&=\hat{w}_3^T(k)\hat{\phi}_3(k)-x_{1d}(k+1)+l_5\hat{e}_1(k)\end{aligned} \tag{41}$$

Where $\hat{z}_3(k)=\left[\hat{x}_1(k),\hat{x}_2(k)\right]^T\in\Re^2$ is the input vector using estimated states, and $\hat{e}_1(k)=\hat{x}_1(k)-\hat{x}_{1d}(k)$. Define the error between $x_2(k)$ and $\hat{x}_{2d}(k)$ as $e_2(k)\in\Re$

$$e_2(k)=x_2(k)-\hat{x}_{2d}(k) \tag{42}$$

Equation (37) can be rewritten using (42) as

$$e_1(k+1)=\Phi(\cdot)-\hat{x}_{2d}(k)-e_2(k)+d_1(k)-x_{1d}(k+1) \tag{43}$$

Combining (41) and (43), we get

$$\begin{aligned}e_1(k+1)&=\Phi(\cdot)-\left(\hat{w}_3^T(k)\hat{\phi}_3(k)-x_{1d}(k+1)+l_5\hat{e}_1(k)\right)\\&\quad-e_2(k)-x_{1d}(k+1)+d_1(k)\\&=-\zeta_3(k)-w_3^T\tilde{\phi}_3(k)+\varepsilon_3(k)-l_5\hat{e}_1(k)-e_2(k)+d_1(k)\end{aligned} \tag{44}$$

where

$$\zeta_3(k)=\tilde{w}_3^T(k)\hat{\phi}_3(k)=\hat{w}_3^T(k)\hat{\phi}_3(k)-w_3^T\hat{\phi}_3(k) \tag{45}$$

$$\tilde{\phi}_3(k)=\phi\left(v_3\hat{z}_3(k)\right)-\phi\left(v_3z_3(k)\right) \tag{46}$$

C. Virtual Control Weight Update

Let us define

$$e_{a1}(k)=\hat{w}_3^T(k)\hat{\phi}_3(k)+\left(\hat{Q}(k)-Q_d(k)\right) \tag{47}$$

where $\zeta_3(k)$ and $\hat{Q}(k)$ are defined in (45) and (27). $e_{a1}(k)\in\Re$ is the error for the first action NN, indicated by the a1 subscript. The desired *strategic* utility function $Q_d(k)$ is "0" to indicate that the nonlinear system can track the reference signal well at all steps. Thus, (47) becomes

$$e_{a1}(k)=\hat{w}_3^T(k)\hat{\phi}_3(k)+\hat{Q}(k) \tag{48}$$

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

$$E_{a1}(k)=\frac{1}{2}e_{a1}^2(k) \tag{49}$$

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

$$\hat{w}_3(k+1)=\hat{w}_3(k)+\Delta\hat{w}_3(k) \tag{50}$$

$$\Delta\hat{w}_3(k)=\alpha_3\left[-\frac{\partial E_{a1}(k)}{\partial\hat{w}_3(k)}\right] \tag{51}$$

$$\hat{w}_3(k+1) = \hat{w}_3(k) - \alpha_3 \hat{\phi}_3(k)\left(\hat{Q}(k) + \hat{w}_3^T(k)\hat{\phi}_3(k)\right) \quad (52)$$

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

VI. Control Input Design

Choose the following desired control input

$$u_d(k) = \frac{1}{g_2(k)}\left(-f_2(k) + \hat{x}_{2d}(k+1) + l_6 e_2(k)\right), \quad (53)$$

Note that $u_d(k)$ is non-causal since it depends upon future value of $\hat{x}_{2d}(k+1)$. We solve this problem by using a semi-recurrent NN. The term $\hat{x}_{2d}(k+1)$ depends on state $x(k)$, virtual control input $\hat{x}_{2d}(k)$, desired trajectory $x_{1d}(k+2)$ and system errors $e_1(k)$ and $e_2(k)$. By taking the independent variables as the input to a NN, $\hat{x}_{2d}(k+1)$ can be approximated during control input selection. Assume the NN input as $z_4(k) = \left[x_1(k), x_2(k), e_1(k), l_6 e_2(k), \hat{x}_{2d}(k), x_{1d}(k+2)\right]^T \in \Re^6$, then $u_d(k)$ can be approximated as

$$u_d(k) = w_4^T \phi\left(v_4^T z_4(k)\right) + \varepsilon\left(z_4(k)\right) = w_4^T \phi_4(k) + \varepsilon_4(k), \quad (54)$$

where $w_4 \in \Re^{n_4}$ and $v_4 \in \Re^{6\times n_4}$ denote the constant ideal output and hidden layer weights, $\phi_4(k) \in \Re^{n_4}$ is the activation function vector, n_4 is the number of hidden layer nodes , and $\varepsilon\left(z_4(k)\right)$ is the estimation error. Again, we hold the input weights constant and adapt the output weights only. We also replace actual with estimated states.

$$\hat{u}(k) = \hat{w}_4^T(k)\phi\left(v_4^T \hat{z}_4(k)\right) = \hat{w}_4^T(k)\hat{\phi}_4(k), \quad (55)$$

where $\hat{z}_4(k) = \left[\hat{x}_1(k), \hat{x}_2(k), \hat{e}_1(k), l_6 \hat{e}_2(k), \hat{x}_{2d}(k), x_{1d}(k+2)\right]^T \in \Re^6$ is the input vector. Rewriting (42) and substituting (53) through (55), we get

$$\begin{aligned}
e_2(k+1) &= x_2(k+1) - \hat{x}_{2d}(k+1) \\
&= \left(f_2(\cdot) + g_2(\cdot)\hat{w}_4^T(k)\hat{\phi}_4(k) + d_2(k)\right) - \hat{x}_{2d}(k+1) \\
&= f_2(\cdot) + g_2(\cdot)\left(w_4^T(k)\phi_4(k)\right) + g_2(\cdot)\left(\zeta_4(k) + w_4^T\tilde{\phi}_4(k)\right) \\
&\quad + d_2(k) - \hat{x}_{2d}(k+1) \\
&= f_2(\cdot) + g_2(\cdot)\left(u_d(k) - \varepsilon_4(k)\right) + g_2(\cdot)\left(\zeta_4(k) + w_4^T\tilde{\phi}_4(k)\right) \\
&\quad + d_2(k) - \hat{x}_{2d}(k+1) \\
&= l_6 e_2(k) - g_2(\cdot)\varepsilon_4(k) + g_2(\cdot)\zeta_4(k) + g_2(\cdot)w_4^T\tilde{\phi}_4(k) + d_2(k)
\end{aligned} \quad (56)$$

where

$$\zeta_4(k) = \tilde{w}_4^T(k)\hat{\phi}_4(k) = \hat{w}_4^T(k)\hat{\phi}_4(k) - w_4^T\hat{\phi}_4(k), \quad (57)$$

$$\tilde{\phi}_4(k) = \hat{\phi}_4(k) - \phi_4(k) \quad (58)$$

Equations (44) and (56) represent the closed-loop error dynamics. Next we derive the weight update law. Define

$$e_{a2}(k) = \hat{w}_4^T(k)\hat{\phi}_4(k) + \hat{Q}(k), \quad (59)$$

where $e_{a2}(k) \in \Re$ is the error where the subscript a2 stands for the second action NN. Following the similar design procedure and taking the bounded unknown disturbance $d_2(k)$ and the NN approximation error $\varepsilon\left(z_4(k)\right)$ to be zeros, the second action NN weight updating rule is given by

$$\hat{w}_4(k+1) = \hat{w}_4(k) - \alpha_4\hat{\phi}_4(k)\left(\hat{w}_4^T(k)\hat{\phi}_4(k) + \hat{Q}(k)\right), \quad (60)$$

The proposed controller structure is shown in Figure 1.

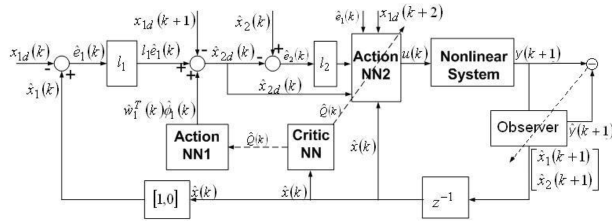

Figure 1 Adaptive-critic NN-based controller diagram.

Theorem 1: Consider the system given by (1) and (2), and the disturbance bounds d_{1m} and d_{2m} be known constants. Let the observer, critic, virtual control, and control input NN weight tuning be given by (24), (32), (52), and (60), respectively. Let the virtual control input and control input be given by (41), and (55), the tracking errors $e_1(k)$ and $e_2(k)$ and weight estimates $\hat{w}_1(k)$, $\hat{w}_2(k)$, $\hat{w}_3(k)$, and $\hat{w}_4(k)$ are UUB, with the bounds specifically given by (A.15) with the controller design parameters selected as

$$0 < \alpha_i \left\|\phi_i(k)\right\|^2 < 1, i \in \{1,2,3,4\} \quad (61)$$

$$|l_1| < \frac{1}{2}; |l_2| < \frac{\sqrt{3}}{3}; |l_3| < \frac{\sqrt{3}}{3}; |l_4| < \frac{\sqrt{3}}{3}; |l_5| < \frac{1}{\sqrt{5}}; |l_6| < \frac{\sqrt{3}}{3} \quad (62)$$

$$0 < \beta < \frac{\sqrt{2}}{2} \quad (63)$$

where $\alpha_i \in \{1,2,3,4\}$ are NN adaptation gains, $l_i \in \{1,2,...,6\}$ are controller gains, β is the *strategic* utility function constant.

Proof: See Appendix. ■

Corollary 1: The proposed adaptive critic NN controller and the weight updating rules with the parameter selection based on (61) through (63), the state $x_2(k)$ approaches the desired virtual control input $x_{2d}(k)$.

Proof: Combining (40) and (41), the difference between $\hat{x}_{2d}(k)$ and $x_{2d}(k)$ is given by

$$\hat{x}_{2d}(k) - x_{2d}(k) = \tilde{w}_3(k)\phi_3(k) - \varepsilon\left(z_3(k)\right) = \zeta_3(k) - \varepsilon_3(k) \quad (64)$$

where $\tilde{w}_3(k) \in \Re^{n_3}$ is the first action NN weight estimation error and $\zeta_3(k) \in \Re$ is defined in (45). Since both $\zeta_3(k) \in \Re$ and $\varepsilon_3(k)$ 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 $x_{2d}(k)$.

VII. Experimental Results

Lean operation of SI engine allows lower emissions and improved fuel efficiency. However, the engine becomes unstable due to the cyclic dispersion of heat release. The adaptive critic NN controller is designed to stabilize the SI engine operating lean.

A. Daw Engine Model

Spark ignition (SI) engine dynamics can be expressed according to the Daw model [9] as follows.

$$x_1(k+1) = AF(k) + F(k)x_1(k) - R\cdot F(k)CE(k)x_2(k) + d_1(k), \quad (65)$$

$$x_2(k+1) = \left(1 - CE(k)\right)F(k)x_2(k) + \left(MF(k) + u(k)\right) + d_2(k), \quad (66)$$

$$y(k) = x_2(k) CE(k), \tag{67}$$

$$\varphi(k) = R\frac{x_2(k)}{x_1(k)}, \tag{68}$$

$$CE(k) = \frac{CE_{\max}}{1 + 100^{-(\varphi(k)-\varphi_m)/(\varphi_u-\varphi_l)}}, \tag{69}$$

$$\varphi_m = \frac{\varphi_u - \varphi_l}{2}, \tag{70}$$

where $x_1(k)$ and $x_2(k)$ are mass of air and fuel, $y_1(k)$ is the heat release, $CE(k)$ is bounded by $0 < CE_{\min} < CE(k) < CE_{\max}$, $F(k)$ is bounded by $0 < F_{\min} < F(k) < F_{\max}$, $d_i(k), i \in \{1,2\}$ are unknown but bounded disturbances bounded by $|d_i(k)| < d_{im}, i \in \{1,2\}$ with $d_{im}, i \in \{1,2\}$ being unknown positive scalars. To implement the observer, replace the following from the Daw model into the general case

$$\begin{aligned}
&f_1(\cdot) = AF(k) + F(k)x_1(k)\\
&g_1(\cdot) = -R \cdot F(k)CE(k)\\
&f_2(\cdot) = (1 - CE(k))F(k)x_2(k) + MF(k)\\
&g_2(\cdot) = 1\\
&f_{10} = AF_0 + F_0\hat{x}_1(k)\\
&g_{10} = -R \cdot F_0 CE_0\\
&f_{10} = (1 - CE_0)F_0\hat{x}_2(k) + MF_0\\
&g_{10} = 1
\end{aligned} \tag{71}$$

To implement the controller, replace the following

$$\Phi(\cdot) = AF(k) + F(k)x_1(k) - R \cdot F(k)CE(k)x_2(k) + x_2(k) \tag{72}$$

To calculate the nominal values for equations (4) through (7), we run the engine at the desired equivalence ratio. That will give us the nominal fuel, air, and equivalence ratio - MF_0, AF_0 and φ_0. From those, combustion efficiency CE_0 is calculated.

B. Simulation Data

The controller is easily simulated in C in conjunction with the Daw model. The learning rates are 0.01, 0.001, 0.04, and 0.005 for for the observer, critic, virtual control input, and control input, respectively. The gains l_1, l_2, l_3, l_4, l_5, and l_6 are selected as 0.99, 1.99, 0.13, 0.49, 0.1 and 0.08. The system constants CEmax, φ_l, and φ_u are chosen as 1, 0.73, and 0.66. The critic constants β and N are 0.4 and 4 for all equivalence ratios. All NNs have 20 hidden neurons and hyperbolic tangent sigmoid activation functions. Simulations are run for 5000 cycles for both uncontrolled and controlled system. The maximum mole of molecules a single cylinder holds is set as 0.021. Using this constant along with the following equations,

$$\varphi = R\left(MF/AF\right) \tag{73}$$

$$tm = \frac{MF}{mw_{fuel}} + \frac{AF}{mw_{air}} \tag{74}$$

where mw_{fuel} and mw_{air} are molecular weights of fuel and air, respectively. tm is the maximum mole of molecules each cylinder is capable of holding. For each equivalence ratio set point, φ, MF and AF can be calculated. Figures 2 and 3 shows one simulation result at 0.79.

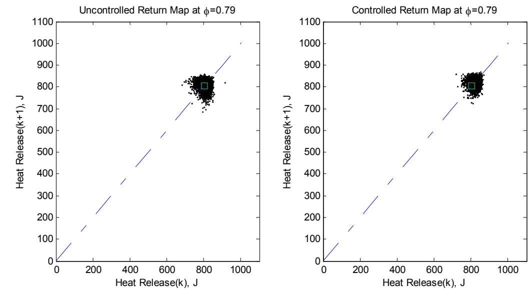

Figure 2 Uncontrolled and controlled heat release return map at φ=0.75.

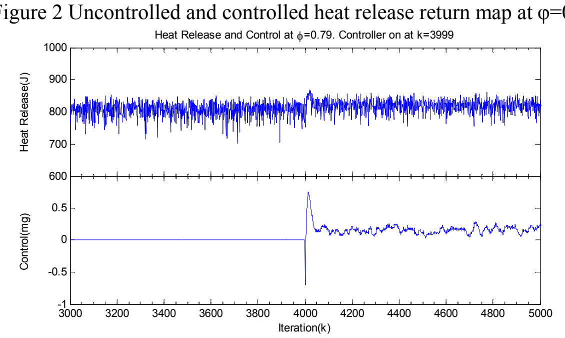

Figure 3 Heat release vs iteration number. Controller turns on at k=3999.

In order to quantify the performance of the controller, we compare the coefficient of variation (COV). It is the ratio of the deviation over the mean of the heat release. As the COV decreases, the performance of the controller increases. The return map consequently approaches a target value. Table 1 tabulates all of the data from simulation. The COV of each set point decreased drastically as the controller operates. The performance outperformed the slight increase in the mean fuel input.

Table 1 Covariance and fuel data. The fuel change is nominal.

PHI	Covariance (COV) Uncontrolled	Covariance (COV) Controlled	% COV change	% Fuel Change
0.90	0.0080	0.0078	-4.1	0.29
0.80	0.0267	0.0221	-17.0	0.66
0.77	0.0475	0.0435	-8.3	0.48
0.75	0.1217	0.1071	-12.0	0.56

VIII. Conclusions

The controller presented successfully controlled a SI engine to reduce cyclic dispersion under lean condition. The system is modeled under a nonstrict feedback nonlinear descrete-time system. It converged upon a near *optimal* solution through the use of a long-term strategic utility function even though the exact dynamics are not known beforehand. Simulation shows the stability of the controller under a variety of set points. The output is stable, as predicted by the Lyapunov proof.

Appendix

Proof of Theorem 1: Define the Lyapunov function

$$J(k) = \sum_{i=1}^{10} J_i(k) = \frac{\gamma_1}{5}e_1^2(k) + \frac{\gamma_2}{3}e_2^2(k) + \sum_{j=3}^{6}\frac{\gamma_j}{\alpha_{j-2}}\tilde{w}_j^T(k)\tilde{w}_j(k) + \gamma_7\zeta_2^2(k-1) + \frac{\gamma_8}{3}\tilde{x}_1^2(k) + \frac{\gamma_9}{3}\tilde{x}_2^2(k) + \frac{\gamma_{10}}{3}\tilde{y}^2 \tag{A.1}$$

where $0 < \gamma_i, i \in \{1,\ldots,6\}$ are auxiliary constants; the NN weights estimation errors $\tilde{w}_1^T(k+1)$, $\tilde{w}_2^T(k+1)$, $\tilde{w}_3^T(k+1)$, and $\tilde{w}_4^T(k+1)$ are defined in (24), (32), (52), and (60), by subtracting their respective ideal weights $w_i, i \in \{1,2,3,4\}$ on both sides; the observation errors $\tilde{x}_1(k+1)$, $\tilde{x}_2(k+1)$, are defined in (18) and (19), respectively; the system errors $e_1(k+1)$ and $e_2(k+1)$ are defined in (44) and (56), respectively; and $\alpha_i, i \in \{1,2,3,4\}$ are NN adaptation gains. The Lyapunov function (A.1) obviates the need for the separation principle. Take the first term and the first difference using (44) to get

$$\Delta J_1(k) \le \gamma_1 l_5^2 \tilde{x}_1^2(k) + \gamma_1 l_5^2 e_1^2(k) + \gamma_1 e_2^2(k) + \gamma_1 \zeta_3^2(k) + \gamma_1\left(\varepsilon_{3m} + w_{3m}\tilde{\phi}_{3m} + d_{1m}\right)^2 - \tfrac{\gamma_1}{5} e_1^2(k) \tag{A.2}$$

Take the second term, substitute (56), and simplify

$$\Delta J_2(k) \le 3 l_6^2 e_2^2(k) + 3 g_{2\max}^2 \zeta_4^2(k) + \gamma_2\left(d_{2m} + g_{2\max}\varepsilon_{4m} + g_{2\max} w_{4m}\tilde{\phi}_{4m}\right)^2 - e_2^2(k) \tag{A.3}$$

Take the third term, substitute (24), and simplify

$$\Delta J_3(k) \le -\gamma_3\left(1-\alpha_1\left\|\hat{\phi}_1(k)\right\|^2\right)\left(\hat{w}_1(k)\hat{\phi}_1(k) + l_4\tilde{y}(k)\right)^2 + 2\gamma_3\left(w_{1m}\hat{\phi}_{1m}\right)^2 + 2\gamma_3 l_4^2 \tilde{y}^2(k) - \gamma_3\zeta_1^2(k) \tag{A.4}$$

Take the fourth term, substitute (32), and simplify

$$\Delta J_4(k) \le -\gamma_4\left(1-\alpha_2\left\|\hat{\phi}_2(k)\right\|^2\right)\left(\hat{Q}(k) + \beta^{N+1}p(k) - \beta\hat{Q}(k-1)\right)^2 - \gamma_4\zeta_2^2(k) + 2\gamma_4\beta^2\zeta_2^2(k-1) + 2\gamma_4\left(w_{2m}\hat{\phi}_{2m}(1+\beta) + \beta^{N+1}\right)^2 \tag{A.5}$$

Take the fifth term, substitute (52), and simplify

$$\Delta J_5(k) \le -\gamma_5\left(1-\alpha_3\left\|\hat{\phi}_3(k)\right\|^2\right)\left(\hat{Q}(k) + \hat{w}_3^T(k)\hat{\phi}_3(k)\right)^2 + 2\gamma_5\zeta_2^2(k) + 2\gamma_5\left(w_{2m}\hat{\phi}_{2m} + w_{3m}\hat{\phi}_{3m}\right)^2 - \gamma_5\zeta_3^2(k) \tag{A.6}$$

Take the sixth term, substitute (60), and simplify

$$\Delta J_6(k) = -\gamma_6\left(1-\alpha_4\left\|\hat{\phi}_4(k)\right\|^2\right)\left(\hat{w}_4^T(k)\hat{\phi}_4(k) + \hat{Q}(k)\right)^2 + 2\gamma_6\left(w_{4m}\hat{\phi}_m + w_{2m}\hat{\phi}_{2m}\right)^2 + 2\gamma_6\zeta_2^2(k) - \gamma_6\zeta_4^2(k) \tag{A.7}$$

Take the seventh term, set $\gamma_7 = 2\gamma_4\beta^2$

$$\Delta J_7(k) = 2\gamma_4\beta^2\zeta_2^2(k) - 2\gamma_4\beta^2\zeta_2^2(k-1) \tag{A.8}$$

Take the eighth term, substitute (18), and simplify

$$\Delta J_8(k) \le \gamma_8 l_2^2 \tilde{y}^2(k) + \gamma_8\tilde{x}_2^2(k) + \gamma_8\left(w_{3m}\phi_{3m} + f_{10} + \varepsilon_{3m} + d_{1m}\right)^2 - \tfrac{\gamma_8}{3}\tilde{x}_1^2(k) \tag{A.9}$$

Take the ninth term, substitute (19), and simplify

$$\Delta J_9(k) \le \gamma_9\left(f_{20} + \left(g_{20} + g_{2\max}\right)w_{4m}\hat{\phi}_{4m} + f_{2\max} + d_{2m}\right)^2 + \gamma_9\left(g_{20} + g_{2\max}\right)\zeta_4(k) + \gamma_9 l_3^2\tilde{y}^2(k) - \tfrac{\gamma_9}{3}\tilde{x}_2^2(k) \tag{A.10}$$

Take the final term, substitute (20), and simplify

$$\Delta J_{10}(k) \le \gamma_{10}\zeta_1^2(k) + \gamma_{10} l_1^2\tilde{y}(k) + \gamma_{10}\left(w_{1m}\tilde{\phi}_{1m} + \varepsilon_{1m}\right) - \tfrac{\gamma_{10}}{3}\tilde{y}^2(k) \tag{A.11}$$

Combining (A.2) through (A.11) and simplify to get the first difference of the Lyapunov function

$$\begin{aligned}
\Delta J \le & -\left(\gamma_4 - 2\gamma_5 - 2\gamma_6 - 2\gamma_4\beta^2\right)\zeta_2^2(k) - \left(\gamma_6 - \gamma_2 g_{2\max}^2 - \gamma_9\left(g_{20} + g_{2\max}\right)\right)\zeta_4^2(k) \\
& -\left(\tfrac{\gamma_8}{3} - \gamma_1 l_5^2\right)\tilde{x}_1^2(k) - \left(\tfrac{\gamma_{10}}{3} - 2\gamma_3 l_4^2 - \gamma_8 l_2^2 - \gamma_9 l_3^2 - \gamma_{10} l_1^2\right)\tilde{y}^2(k) - \left(\gamma_3 - \gamma_{10}\right)\zeta_1^2(k) \\
& -\gamma_3\left(1-\alpha_1\left\|\hat{\phi}_1(k)\right\|^2\right)\left(\hat{w}_1(k)\hat{\phi}_1(k) + l_4\tilde{y}(k)\right)^2 - \left(\tfrac{\gamma_9}{3} - \gamma_8\right)\tilde{x}_2^2(k) + D_M^2 \cdots \\
& -\gamma_4\left(1-\alpha_2\left\|\hat{\phi}_2(k)\right\|^2\right)\left(\hat{Q}(k) + \beta^{N+1}p(k) - \beta\hat{Q}(k-1)\right)^2 - \left(\gamma_5 - \gamma_1\right)\zeta_3^2(k) \\
& -\gamma_5\left(1-\alpha_3\left\|\hat{\phi}_3(k)\right\|^2\right)\left(\hat{Q}(k) + \hat{w}_3^T(k)\hat{\phi}_3(k)\right)^2 - \left(\tfrac{\gamma_1}{5} - \gamma_1 l_5^2\right)e_1^2(k) \\
& -\gamma_6\left(1-\alpha_4\left\|\hat{\phi}_4(k)\right\|^2\right)\left(\hat{w}_4^T(k)\hat{\phi}_4(k) + \hat{Q}(k)\right)^2 - \left(\tfrac{\gamma_2}{3} - \gamma_1 - \gamma_2 l_6^2\right)e_2^2(k)
\end{aligned} \tag{A.12}$$

where

$$\begin{aligned}
D_m^2 = & \gamma_1\left(\varepsilon_{3m} + w_{3m}\tilde{\phi}_{3m} + d_{1m}\right)^2 + 2\gamma_5\left(w_{2m}\hat{\phi}_{2m} + w_{3m}\hat{\phi}_{3m}\right)^2 + \\
& 2\gamma_3\left(w_{1m}\hat{\phi}_{1m}\right)^3 + 2\gamma_4\left(w_{2m}\hat{\phi}_{2m}(1+\beta) + \beta^{N+1}\right)^2 + \\
& 2\gamma_6\left(w_{4m}\hat{\phi}_m + w_{2m}\hat{\phi}_{2m}\right)^2 + \gamma_8\left(w_{3m}\phi_{3m} + f_{10} + \varepsilon_{3m} + d_{1m}\right)^2 + \\
& \gamma_9\left(f_{20} + \left(g_{20} + g_{2\max}\right)w_{4m}\hat{\phi}_{4m} + f_{2\max} + d_{2m}\right)^2 + \\
& \gamma_2\left(d_{2m} + g_{2\max}\varepsilon_{4m} + g_{2\max}w_{4m}\tilde{\phi}_{4m}\right)^2 + \gamma_{10}\left(w_{1m}\tilde{\phi}_{1m} + \varepsilon_{1m}\right)^2
\end{aligned} \tag{A.13}$$

Select

$$\begin{aligned}
&\gamma_1 > 5\gamma_1 l_5^2;\ \gamma_2 > 3\gamma_1 + 3\gamma_2 l_6^2;\ \gamma_3 > \gamma_{10};\ \gamma_4 > 2\gamma_5 + 2\gamma_6 + 2\gamma_4\beta^2; \\
&\gamma_6 > \gamma_2 g_{2\max}^2 + \gamma_9\left(g_{20} + g_{2\max}\right);\ \gamma_7 = 2\gamma_4\beta^2;\ \gamma_8 > 3\gamma_1 l_5^2; \\
&\gamma_{10} > 6\gamma_3 l_4^2 + 3\gamma_8 l_2^2 + 3\gamma_9 l_3^2 + 3\gamma_{10} l_1^2;\ \gamma_5 > \gamma_1;\ \gamma_9 > 3\gamma_8;
\end{aligned} \tag{A.14}$$

This implies $\Delta J(k) < 0$ as long as (61) through (63) hold *and* any *one* of the following hold

$$\begin{aligned}
&\left|e_1(k)\right| > \frac{D_M}{\sqrt{\frac{\gamma_1}{5} - \gamma_1 l_5^2}};\ \left|e_2(k)\right| > \frac{D_M}{\sqrt{\frac{\gamma_2}{3} - \gamma_1 - \gamma_2 l_6^2}};\ \left|\zeta_1(k)\right| > \frac{D_M}{\sqrt{\gamma_3 - \gamma_{10}}}; \\
&\left|\zeta_3(k)\right| > \frac{D_M}{\sqrt{\gamma_5 - \gamma_1}};\ \left|\zeta_2(k)\right| > \frac{D_M}{\sqrt{\gamma_4 - 2\gamma_5 - 2\gamma_6 - 2\gamma_4\beta^2}}; \\
&\left|\zeta_4(k)\right| > \frac{D_M}{\sqrt{\gamma_6 - \gamma_2 g_{2\max}^2 - \gamma_9\left(g_{20} + g_{2\max}\right)}};\ \left|\tilde{x}_1(k)\right| > \frac{D_M}{\sqrt{\frac{\gamma_8}{3} - \gamma_1 l_5^2}}; \\
&\left|\tilde{x}_2(k)\right| > \frac{D_M}{\sqrt{\frac{\gamma_9}{3} - \gamma_8}};\ \left|\tilde{y}(k)\right| > \frac{D_M}{\sqrt{\frac{\gamma_{10}}{3} - 2\gamma_3 l_4^2 - \gamma_8 l_2^2 - \gamma_9 l_3^2 - \gamma_{10} l_1^2}};
\end{aligned} \tag{A.15}$$

REFERENCES

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

[2] S. Jagannathan, "Control of a class of nonlinear systems using multilayered neural networks", *IEEE Transactions on Neural Networks*, vol.12, No. 5, September 2001.

[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, pp. 791-801, 1995.

[4] J. Si, in *NSF Workshop on Learning and Approximate Dynamic Programming*, Playacar, Mexico, 2002.

[5] P. J. Werbos, *Neurocontrol and supervised learning: An overview and evaluation*. New York: Van Nostrand Reinhold, 1992.

[6] J. J. Murray, C. Cox, G. G. Lendaris, and R. Saeks, "Adaptive dynamic programming," *IEEE Trans. Syst., Man, Cybern.,* vol. 32, pp. 140-153, 2002.

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

[8] S. Jagannathan, *Neural Network Control of Nonlinear Discrete-time Systems*. London, UK: Taylor and Francis, 2006.

[9] C. S. Daw, C. E. A. Finney, M. B. Kennel, F. T. Connolly, "Observing and Modeling Nonlinear Dynamics in an Internal Combustion Engine," *Phys. Rev. E,* vol. 57, pp. 2811-2819, 1998.

[10] N. Hovakimyan, F. Nardi, A. Calise, and N. Kim, "Adaptive output feedback control of uncertain nonlinear systems using single-hidden-layer neural networks," *IEEE Trans. on Neural Networks,* vol. 13, pp. 1420-1431, 2002.

[11] A. N. Atassi and H. K. Khalil, "A separation principle for the stabilization of a class of nonlinear systems," *IEEE Trans. Automat. Contr,* vol. 76, pp. 334-354, 2003.

[12] B. Igelruk and Y. H. Pao, "Stochastic choice of basis functions in adaptive function approximation and the functional-link net," *IEEE Trans. Neural Networks,* vol. 6, pp. 1320-1329, 1995.