01 Jan 2007

Near Optimal Output-Feedback Control of Nonlinear Discrete-Time Systems in Nonstrict Feedback Form with Application to Engines

Peter Shih

Brian C. Kaul

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

J. A. Drallmeier
Missouri University of Science and Technology, drallmei@mst.edu

Recommended Citation

P. Shih et al., "Near Optimal Output-Feedback Control of Nonlinear Discrete-Time Systems in Nonstrict Feedback Form with Application to Engines," *Proceedings of the International Joint Conference on Neural Networks (2007, Orlando, FL)*, Institute of Electrical and Electronics Engineers (IEEE), Jan 2007.
The definitive version is available at https://doi.org/10.1109/IJCNN.2007.4370989

Proceedings of International Joint Conference on Neural Networks, Orlando, Florida, USA, August 12-17, 2007

Near Optimal Output-Feedback Control of Nonlinear Discrete-time Systems in Nonstrict Feedback Form with Application to Engines

Peter Shih, B. Kaul, Sarangapani Jagannathan, and J. 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 controller is evaluated on a spark ignition (SI) engine operating with high exhaust gas recirculation (EGR) levels and experimental results are demonstrated.**

I. Introduction

Adaptive NN backstepping control of nonlinear discrete-time systems in strict feedback form [1-3] result in non-causal controllers when applied for nonstrict feedback nonlinear discrete-time systems 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 non-causal problem is overcome by employing the universal NN approximation property for non-strict feedback nonlinear discrete-time systems; 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 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.

The proposed primary controller is applied to control the spark ignition (SI) engine dynamics in high EGR mode, where an inert gas displaces the stoichiometric ratio of fuel to air. The engine destabilizes in this mode and heat release (HR) dispersion increases, which the controller attempts to reduce. Consequently, the engine improves emissions and fuel efficiency.

II. Non-linear Non-strict Feedback System

Consider the following nonlinear discrete-time system.

$$x_1(k+1) = f_1(\bar{x}_i(k)) + g_1(\bar{x}_i(k))x_2(k) + d_1(k) \tag{1}$$

$$x_2(k+1) = f_2(\bar{x}_i(k)) + g_2(\bar{x}_i(k))u(k) + d_2(k) \tag{2}$$

$$x_3(k+1) = f_4(\bar{x}_i(k)) + g_4(\bar{x}_i(k))v(k) + d_3(k) \tag{3}$$

$$y(k+1) = f_3(\bar{x}_i(k)) \tag{4}$$

where $\bar{x}_i(k) = [x_1(k), x_2(k), x_3(k)]^T$ are the states; $u(k) \in \Re$ and $v(k) \in \Re$ are system inputs; and $d_1(k) \in \Re$, $d_2(k) \in \Re$ and $d_3(k) \in \Re$ are unknown but bounded disturbances. The bounds are given by $|d_i(k)| < d_{im}, i \in \{1, 2, 3\}$, where the upper bounds are unknown positive scalars. Finally, the output and third state are measurable whereas the first two states are not. For the system (3) and (4), not only the system actual

This work is supported in part by NSF grants ECCS#0327877 and ECCS#0621924. Peter Shih and Sarangapani Jagannathan are with the Department of Electrical and Computer Engineering, whereas B. Kaul and J. Drallmeier are with the Department of Mechanical and Aerospace Engineeing at the University of Missouri-Rolla (contact author's e-mail: sarangap@umr.edu).

output should converge to its target value but also the states should converge to their respective desired values.

The controller development presented use equations (1), (2), and (4). Equation (3) can be controlled by another controller such as [9] and therefore omitted. Consequently, the third state is considered bounded to its target value.

III. OBSERVER DESIGN

To overcome the immeasurable states $x_1(k)$ and $x_2(k)$, an observer is used.

A. Observer Design

Consider equations (1) and (2). We expand the individual nonlinear functions using Taylor series as

$$f_i(\cdot) = f_{i0} + \Delta f_i(\cdot),\ i \in \{1,2\} \tag{5}$$

$$g_i(\cdot) = g_{i0} + \Delta g_i(\cdot),\ i \in \{1,2\} \tag{6}$$

where the first term in (5) through (6) are known nominal values and the second term are unknown higher order terms. We use a two-layer feed-forward NN with semi-recurrent architecture and novel weight tuning to construct the output

$$y(k+1) = w_1^T \phi\left(v_1^T z_1(k)\right) + \varepsilon\left(z_1(k)\right), \tag{7}$$

where $z_1(k) = \left[x_1(k), x_2(k), x_3(k), y(k), u(k)\right]^T \in R^4$ is the network input, $y(k+1)$ and $y(k)$ are the future and current outputs, $w_1 \in \Re^{n_1}$ and $v_1 \in \Re^{2 \times n_1}$ denote the ideal output and constant hidden layer weight matrices, 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 the two equations can be represented as

$$\phi_1(k) = \phi\left(v_1^T z_1(k)\right) \tag{8}$$

$$\varepsilon_1(k) = \varepsilon\left(z_1(k)\right) \tag{9}$$

Rewrite (7) using (8) and (9) to obtain

$$y(k+1) = w_1^T \phi_1(k) + \varepsilon_1(k) \tag{10}$$

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 the output $\hat{x}_1(k)$, $\hat{x}_2(k)$, and $\hat{y}(k)$, respectively, instead of $x_1(k)$, $x_2(k)$, and $y(k)$, the proposed observer is given as

$$\hat{y}(k+1) = \hat{w}_1^T(k)\phi\left(v_1^T \hat{z}_1(k)\right) + l_1 \tilde{y}(k) = \hat{w}_1^T(k)\hat{\phi}_1(k) + l_1 \tilde{y}(k) \tag{11}$$

where $\hat{z}_1(k) = \left[\hat{x}_1(k), \hat{x}_2(k), x_3(k), \hat{y}(k), u(k)\right]^T \in R^5$ is the input vector 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, $u(k)$ is the estimate control input, $\hat{\phi}_1(k)$ is the hidden layer activation function, $l_1 \in R$ is the observer gain, and $\tilde{y}(k) = \hat{y}(k) - y(k)$ is the output estimation error.

It is demonstrated in [10] 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 to the nonlinear function that the NN approximates. Now we choose, at our convenience, the observer structure as a function of output estimation errors and known quantities as

$$\hat{x}_1(k+1) = f_{10} - \hat{x}_2(k) + l_2 \tilde{y}(k) \tag{12}$$

$$\hat{x}_2(k+1) = f_{20} + g_{20} u(k) + l_3 \tilde{y}(k) \tag{13}$$

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

B. Observer Error Dynamics

Define the state estimations and output error as

$$\tilde{x}_i(k+1) = \hat{x}_i(k+1) - x_i(k+1),\ i \in \{1,2\} \tag{14}$$

$$\tilde{y}(k+1) = \hat{y}(k+1) - y(k+1) \tag{15}$$

Combining (1) through (7) and, (12) through (15), to obtain the estimation and output error dynamics as

$$\tilde{x}_1(k+1) = f_{10} - \hat{x}_2(k) + l_2 \tilde{y}(k) - f_1(\cdot) - g_1(\cdot) x_2(k) - d_1(k) \tag{16}$$

$$\tilde{x}_2(k+1) = f_{20} + g_{20} u(k) + l_3 \tilde{y}(k) - f_2(\cdot) - g_2(\cdot) u(k) - d_2(k) \tag{17}$$

$$\tilde{y}(k+1) = \hat{w}_1^T(k)\hat{\phi}_1(k) + l_1 \tilde{y}(k) - w_1^T \phi_1(k) - \varepsilon_1(k) \tag{18}$$

Choose the weight tuning of the observer as

$$\hat{w}_1(k+1) = \hat{w}_1(k) - \alpha_1 \hat{\phi}_1(k)\left(\hat{w}_1^T(k)\hat{\phi}_1(k) + l_4 \tilde{y}(k)\right) \tag{19}$$

where $\alpha_1 \in R$, and $l_4 \in R$ are design constants. It will be shown later in the next section that by using the above weight tuning, separation principle is relaxed and the closed-loop signals will be bounded. In order to proceed, following assumption is required.

Assumption 1: The unknown smooth functions, $f_2(\cdot)$ and $g_2(\cdot)$, are upper bounded within the compact set S as $f_{2\max} > \left|f_2(k)\right|$, and $g_{2\max} > \left|g_2(k)\right|$.

IV. CRITIC DESIGN

The purpose of the critic NN is to approximate the long-term performance index (or 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 critic NN design is given next.

A. The Strategic Utility Function

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

$$p(k) = \begin{cases} 0, & if\left(\left|\tilde{y}(k)\right|\right) \le c \\ 1, & otherwise \end{cases} \tag{20}$$

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

$$Q(k) = \beta^N p(k+1) + \beta^{N-1} p(k+2) + \cdots + \beta p(k+N), \tag{21}$$

where $0 < \beta < 1$ is the discount factor and N is the horizon index. The term $Q(k)$ is the long system performance measure for the controller since it is the sum of future utility functions. Equation (21) can also be expressed as $Q(k) = \min_{u(k)} \left\{\alpha Q(k-1) - \alpha^{N+1} p(k)\right\}$.

B. Design of the Critic NN

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

$$\hat{Q}(k) = \hat{w}_2^T(k)\phi\left(v_2^T \hat{z}_2(k)\right) = \hat{w}_2^T(k)\hat{\phi}_2(k) \tag{22}$$

where $\hat{Q}(k) \in \Re$ is the critic signal, $\hat{w}_2(k) \in \Re^{n_2}$ and $v_2 \in \Re^{3 \times n_2}$ are the tunable weight and constant input weight matrix selected at random, $\hat{\phi}_2(k) \in \Re^{n_2}$ is the activation function vector in the hidden layer, n_2 is the number of the nodes in the hidden layer, and $\hat{z}_2(k) = [\hat{x}_1(k), \hat{x}_2(k), x_3(k)]^T \in R^3$ is the input vector.

C. Critic Weight Update Law

We define the prediction error as

$$e_c(k) = \hat{Q}(k) - \beta\left(\hat{Q}(k-1) - \beta^N p(k)\right) \tag{23}$$

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

$$E_c(k) = \frac{1}{2} e_c^2(k) \tag{24}$$

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

$$\hat{w}_2(k+1) = \hat{w}_2(k) + \alpha_2 \left[-\frac{\partial E_c(k)}{\partial \hat{w}_2(k)} \right] \tag{25}$$

$$\hat{w}_2(k+1) = \hat{w}_2(k) - \alpha_2 \hat{\phi}_2(k)\left(\hat{Q}(k) + \beta^{N+1} p(k) - \beta \hat{Q}(k-1)\right)^T \tag{26}$$

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

V. Virtual Control Input NN

In this section, the design of the virtual control input is discussed. First, the following mild assumption is needed.

Assumption 2: The unknown smooth function $g_2(\cdot)$ is bounded away from zero for all $x_1(k)$ and $x_2(k)$ within the compact set S. In other words, $0 < g_{2\min} < |g_2(\cdot)| < g_{2\max}$, , $\forall x_1(k) \,\&\, x_2(k) \in S$ where $g_{2\min} \in \Re^+$ and $g_{2\max} \in \Re^+$. Without the loss generality, we will assume $g_2(\cdot)$ is positive in this paper.

A. System Simplification

Simplify by rewriting the state equations with

$$\Phi(\cdot) = f_1(\bar{x}_i(k)) + g_1(\bar{x}_i(k)) x_2(k) + x_2(k) \tag{27}$$

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

$$x_1(k+1) = \Phi(\cdot) - x_2(k) + d_1(k) \tag{28}$$

$$x_2(k+1) = f_2(\cdot) + g_2(\cdot) u(k) + d_2(k) \tag{29}$$

B. Virtual Control Input Design

Our goal is to stabilize the system output $y(k)$ around a specified target point, y_d. The secondary objective is to make $x_1(k)$ approach the desired trajectory $x_{1d}(k)$. At the same time, all signals in systems (1) and (2) must be UUB while a performance index must be minimized. Define the tracking error as

$$e_1(k) = x_1(k) - x_{1d}(k) \tag{30}$$

where $x_{1d}(k)$ is the desired trajectory. Using (28), (30) can be expressed as the following

$$\begin{aligned} e_1(k+1) &= x_1(k+1) - x_{1d}(k+1) \\ &= \left(\Phi(\cdot) - x_2(k) + d_1(k)\right) - x_{1d}(k+1) \end{aligned} \tag{31}$$

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

$$x_{2d}(k) = \Phi(\cdot) - x_{1d}(k+1) + l_5 \hat{e}_1(k) \tag{32}$$

where l_5 is a gain constant. Since $\Phi(\cdot)$ is an unknown function, $x_{2d}(k)$ in (32) cannot be implemented in practice. We invoke the universal approximation property of NN to estimate this unknown function.

$$\Phi(\cdot) = w_3^T \phi\left(v_3^T z_3(k)\right) + \varepsilon\left(z_3(k)\right) \tag{33}$$

where $z_3(k) = \left[x_1(k), x_2(k), x_3(k)\right]^T \in \Re^3$ is the input vector, $w_3^T \in \Re^{n_3}$ and $v_3^T \in \Re^{3 \times n_3}$ are the ideal and constant input weight matrices, $\phi\left(v_3^T z_3(k)\right) \in \Re^{n_3}$ is the activation function vector in the hidden layer, n_3 is the number of the nodes in the hidden layer, and $\varepsilon(z_3(k))$ is the functional estimation error. Rewrite (32) using (33), the virtual control signal can be rewritten as

$$x_{2d}(k) = w_3^T \phi\left(v_3^T z_3(k)\right) + \varepsilon\left(z_3(k)\right) - x_{1d}(k+1) + l_5 \hat{e}_1(k) \tag{34}$$

Replacing actual with estimated states, (34) becomes

$$\begin{aligned} \hat{x}_{2d}(k) &= \hat{w}_3^T(k) \phi\left(v_3^T \hat{z}_3(k)\right) - x_{1d}(k+1) + l_5 \hat{e}_1(k) \\ &= \hat{w}_3^T(k) \hat{\phi}_3(k) - x_{1d}(k+1) + l_5 \hat{e}_1(k) \end{aligned} \tag{35}$$

where $\hat{z}_3(k) = \left[\hat{x}_1(k), \hat{x}_2(k), x_3(k)\right]^T \in \Re^3$ is the input vector using estimated states, and $\hat{e}_1(k) = \hat{x}_1(k) - x_{1d}(k)$. Define

$$e_2(k) = x_2(k) - \hat{x}_{2d}(k) \tag{36}$$

Equation (31) can be rewritten using (36) as

$$\begin{aligned} e_1(k+1) &= \left(\Phi(\cdot) - x_2(k) + d_1(k)\right) - x_{1d}(k+1) \\ &= \Phi(\cdot) - \hat{x}_{2d}(k) - e_2(k) - x_{1d}(k+1) + d_1(k) \end{aligned} \tag{37}$$

Combine (35) , (37), then (33)

$$e_1(k+1) = -\zeta_3(k) - w_3^T \tilde{\phi}_3(k) + \varepsilon_3(k) - l_5 \hat{e}_1(k) - e_2(k) + d_1(k) \tag{38}$$

where

$$\zeta_3(k) = \tilde{w}_3^T(k) \hat{\phi}_3(k) = \hat{w}_3^T(k) \hat{\phi}_3(k) - w_3^T \hat{\phi}_3(k) \tag{39}$$

and

$$\tilde{\phi}_3(k) = \phi\left(v_3 \hat{z}_3(k)\right) - \phi\left(v_3 z_3(k)\right) \tag{40}$$

C. Virtual Control Weight Update

Let us define

$$e_{a1}(k) = \hat{w}_3^T(k) \hat{\phi}_3(k) + \left(\hat{Q}(k) - Q_d(k)\right) \tag{41}$$

where $\hat{Q}(k)$ is defined in (22), and the a1 subscript represents the error for the first action NN, $e_{a1}(k) \in \Re$. The desired *strategic* utility function $Q_d(k)$ is "0" to indicate perfect tracking at all steps. Thus, (41) becomes

$$e_{a1}(k) = \hat{w}_3^T(k) \hat{\phi}_3(k) + \hat{Q}(k) \tag{42}$$

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

$$E_{a1}(k) = \frac{1}{2} e_{a1}^2(k) \tag{43}$$

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

$$\hat{w}_3(k+1) = \hat{w}_3(k) + \alpha_3 \left[-\frac{\partial E_{a1}(k)}{\partial \hat{w}_3(k)} \right] \tag{44}$$

$$\hat{w}_3(k+1) = \hat{w}_3(k) - \alpha_3 \hat{\phi}_3(k)\left(\hat{Q}(k) + \hat{w}_3^T(k) \hat{\phi}_3(k)\right) \tag{45}$$

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

VI. Control Input Design

Choose the following desired control input

$$u_d(k) = \frac{1}{g_2(k)}\left(-f_2(k) + \hat{x}_{2d}(k+1) + l_6 e_2(k)\right), \tag{46}$$

Note that $u_d(k)$ is non-causal since it depends upon future value $\hat{x}_{2d}(k+1)$. We solve this problem by using a semi-recurrent one step predictor 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. The first layer of the second NN using the system errors, state estimates and past value $\hat{x}_{2d}(k)$ as inputs generates $\hat{x}_{2d}(k+1)$ which in turn is used by the second layer to generate an output, which is used as the control input. Define the NN input as

$$z_4(k)=\left[x_1(k),x_2(k),x_3(k),e_1(k),l_6e_2(k),\hat{x}_{2d}(k),x_{1d}(k+2)\right]^T\in\Re^7,$$

then $u_d(k)$ can be approximated as

$$u_d(k)=w_4^T\phi\left(v_4^Tz_4(k)\right)+\varepsilon\left(z_4(k)\right)=w_4^T\phi_4(k)+\varepsilon_4(k), \tag{47}$$

where $w_4\in\Re^{n_4}$ and $v_4\in\Re^{7\times n_4}$ denote the constant ideal output and hidden layer weight matrices, $\phi_4(k)\in\Re^{n_4}$ is the activation function vector, n_4 is the number of hidden layer nodes , and $\varepsilon(z_4(k))$ is the estimation error. Again, we hold the input weights constant and adapt the output weights only. We also replace actual with estimated states

$$\hat{u}(k)=\hat{w}_4^T(k)\phi\left(v_4^T\hat{z}_4(k)\right)=\hat{w}_4^T(k)\hat{\phi}_4(k), \tag{48}$$

where

$\hat{z}_4(k)=\left[\hat{x}_1(k),\hat{x}_2(k),x_3(k),\hat{e}_1(k),l_6\hat{e}_2(k),\hat{x}_{2d}(k),x_{1d}(k+2)\right]^T\in\Re^7$ is the input vector. Rewriting (36) and substituting (46) through (48), to get

$$\begin{aligned}e_2(k+1)&=x_2(k+1)-\hat{x}_{2d}(k+1)\\&=l_6e_2(k)-g_2(\cdot)\varepsilon_4(k)+g_2(\cdot)\zeta_4(k)+g_2(\cdot)w_4^T\tilde{\phi}_4(k)+d_2(k)\end{aligned} \tag{49}$$

where

$$\zeta_4(k)=\tilde{w}_4^T(k)\hat{\phi}_4(k)=\hat{w}_4^T(k)\hat{\phi}_4(k)-w_4^T\hat{\phi}_4(k), \tag{50}$$

and

$$\tilde{\phi}_4(k)=\hat{\phi}_4(k)-\phi_4(k) \tag{51}$$

Equations (38) and (49) represent the closed-loop error dynamics. Next we derive the weight update law. Define

$$e_{a2}(k)=\hat{w}_4^T(k)\hat{\phi}_4(k)+\hat{Q}(k), \tag{52}$$

where $e_{a2}(k)\in\Re$ is the error where the subscript a2 stands for the second action NN. Following the similar design, choose a quadratic objective function to minimize

$$E_{a2}(k)=\frac{1}{2}e_{a2}^2(k) \tag{53}$$

Define a gradient-based adaptation where the general form is given by

$$\hat{w}_4(k+1)=\hat{w}_4(k)+\alpha_4\left[-\frac{\partial E_{a2}(k)}{\partial\hat{w}_4(k)}\right] \tag{54}$$

$$\hat{w}_4(k+1)=\hat{w}_4(k)-\alpha_4\hat{\phi}_4(k)\left(\hat{w}_4^T(k)\hat{\phi}_4(k)+\hat{Q}(k)\right), \tag{55}$$

Before we proceed, the following assumptions are needed.

Assumption 3 (Bounded Ideal Weights): Let w_1, w_2, w_3 and w_4 be the unknown output layer target weights for the four NNs and assume that they are bounded above so that

$$\|w_1\|\le w_{1m},\ \|w_2\|\le w_{2m},\|w_3\|\le w_{3m},\text{ and }\|w_4\|\le w_{4m} \tag{56}$$

where $w_{om}\in R^+$, $w_{1m}\in R^+$ and $w_{2m}\in R^+$ represent the bounds where the Frobenius norm [11] is used.

Theorem 2: 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 (19), (26), (45), and (55), respectively. Let the virtual control input and control input be given by (35), and (48), the estimation errors and 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

$$0<\alpha_i\left\|\phi_i(k)\right\|^2<1, i\in\{1,2,3,4\} \tag{57}$$

$$|l_1|<\frac{1}{2};|l_2|<\frac{\sqrt{3}}{3};|l_3|<\frac{\sqrt{3}}{3};|l_4|<\frac{\sqrt{3}}{3};|l_5|<\frac{1}{\sqrt{5}};|l_6|<\frac{\sqrt{3}}{3} \tag{58}$$

$$0<\beta<\frac{\sqrt{2}}{2} \tag{59}$$

where α_1, α_2, α_3 and α_4 are NN adaptation gains, l_1, l_2, l_3, l_4, l_5, and l_6 are controller gains, β is employed to define the *strategic* utility function.

Proof: See Appendix. ■

Corollary 1: *Consider* the proposed adaptive critic NN controller and the weight updating rules with the parameter selection based on (57) through (59), the state $x_2(k)$ approaches the desired virtual control input $x_{2d}(k)$.

Proof: Combining (34) and (35), the difference between $\hat{x}_{2d}(k)$ and $x_{2d}(k)$ is given by

$$\hat{x}_{2d}(k)-x_{2d}(k)=\tilde{w}_3(k)\phi_3(k)-\varepsilon\left(z_3(k)\right)=\zeta_3(k)-\varepsilon_3(k) \tag{60}$$

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 (39). Since both $\zeta_3(k)\in\Re$ and $\varepsilon_3(k)$ are bounded, $\hat{x}_{2d}(k)$ is bounded near $x_{2d}(k)$. 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 $x_{2d}(k)$. ■

VII. Results and Analysis

A. Daw Engine Model

Spark ignition (SI) engine dynamics can be expressed according to the Daw model as a class of nonlinear systems in non-strict feedback form [8]. At high EGR levels, the engine can be expressed as the following [11]

$$\begin{aligned}x_1(k+1)=&AF(k)+F(k)x_1(k)-R\cdot F(k)CE(k)x_2(k)+\\&F(k)\left(r_{O_2}(k)+r_{N_2}(k)\right)+d_1(k)\end{aligned} \tag{61}$$

$$x_2(k+1)=\left(1-CE(k)\right)F(k)x_2(k)+\left(MF(k)+u(k)\right)+d_2(k) \tag{62}$$

$$x_3(k+1)=F(k)\left(r_{CO_2}(k)+r_{H_2O}(k)+r_{N_2}(k)+x_3(k)+EGR(k)\right) \tag{63}$$

$$y(k)=x_2(k)CE(k) \tag{64}$$

$$\varphi(k)=R\frac{x_2(k)}{x_1(k)}\left[1-\gamma\frac{x_3(k)+EGR(k)}{\left(x_3(k)+x_1(k)+x_3(k)+EGR(k)\right)}\right] \tag{65}$$

$$CE(k)=\frac{CE_{\max}}{1+100^{-(\varphi(k)-\varphi_m)/(\varphi_u-\varphi_l)}} \tag{66}$$

$$\varphi_m=\frac{\varphi_u-\varphi_l}{2} \tag{67}$$

$$r_i(k)=\gamma_ix_2(k)CE(k),\ i\in\{H_2O,\ O_2,\ N_2,\ CO_2\} \tag{68}$$

where $x_1(k)$, $x_2(k)$, and $x_3(k)$ are total mass of air, fuel, and inert gas, respectively. $y_1(k)$ is the HR. The value of $CE(k)$ is within the range of $0<CE_{\min}<CE(k)<CE_{\max}$. $F(k)$ is bounded by $0<F_{\min}<F(k)<F_{\max}$. $d_i(k),\ i\in\{1,2,3\}$ are unknown but bounded disturbances bounded by $\left|d_i(k)\right|<d_{im},\ i\in\{1,2,3\}$

with bounds being unknown positive scalars. $\varphi_m, \varphi_l, \varphi_u$ are equivalence ratio parameters. $r_i(k), i \in \{H_2O, O_2, N_2, CO_2\}$ are the mass of water, oxygen, nitrogen, and carbon dioxide, respectively whereas γ, γ_{H_2O}, γ_{O_2}, γ_{N_2}, and γ_{CO_2} are design constants, and constants associated with their respective chemicals. Equation (63) can be viewed as affine nonlinear discrete-time systems and standard methods [11] without any optimization can be applied separately. Therefore, it is omitted here.

B. Ricardo Engine

The experimental results are collected from a Ricardo Hydra engine with a four valve Ford Zetek head. It contains a single cylinder running at 1000 rpm. A piezoelectric pressure transducer records cylinder pressure every crank angle degree. The cylinder pressure is integrated along with volume between 345° to 490° during the 17.7 ms calculation window. The output of our controller controls the fuel input. This is controlled by a TTL signal to a fuel injector driver circuit. All signals communicate through a custom interface board using a microcontroller. The board interfaces with the PC through a parallel port and with the engine hardware through an analog signal.

C. Experimental Data

The learning rates are chosen as 0.01 for all NNs. The gains l_1, l_2, l_3, l_4, l_5, and l_6 are selected as 0.05, 0.05, 0.04, 0.05, 0.2 and 0.1. The system constants CE_{max}, φ_l, and φ_u are chosen as 1, 0.54, and 0.58. The critic constants β and N are 0.4 and 4. All NNs use 20 hidden neurons with hyperbolic tangent sigmoid activation functions. Uncontrolled and controlled data were collected at EGR percentages of 18, 20, and 23. The uncontrolled engine ran for 5,000 cycles and then the controller is turned on for another 5,000 cycles. Steady state was ensured prior to data collection by measuring heat release (HR).

Figure 1 shows two HR return maps, one controlled and the other uncontrolled for the 18% EGR set point. HR at k+1 instance is plotted against HR at k instance. The target HR is at 870J. At this set point, cyclic dispersion can clearly be seen, indicated by deviation of the points away from the main cluster on the 45 degree line. The controller decreases dispersion, indicated by tighter grouping of the HR.

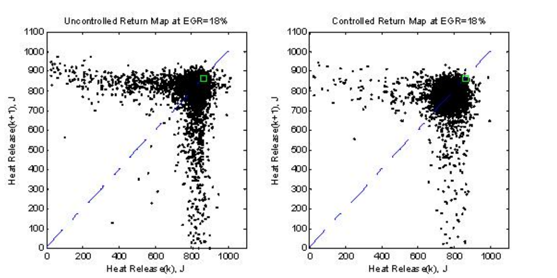

Figure 1. Uncontrolled and controlled HR return map at EGR=18%.

Figure 2 shows the time series of the HR and control input for the same set point. Note the immediate learning of the controller after the controller is turned on.

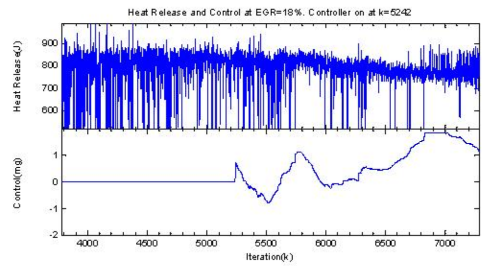

Figure 2. Uncontrolled and controlled HR return map at EGR=18%.

In order to quantify the performance of the controller, we compare the coefficient of variation (COV), which is the standard deviation normalized by dividing by the mean of the HR. Table 1 shows the improved COV when the controller is in operation compared to an uncontrolled engine and also the corresponding change in nominal fuel.

Table 1 Coefficient of variation (COV) and fuel data.

	COV		%COV	%Fuel
EGR	**Uncontrolled**	**Controlled**	**Change**	**Change**
0.18	0.2112	0.1511	-28.4	1.36
0.20	0.2139	0.1400	-34.6	0.77
0.23	0.5777	0.5066	-12.3	2.11

On average, the COV decreased significantly by 25% compared to the controlled case. The COV and fuel change data indicates an improved performance compared to the previous controller [11]. The average drop in COV was 17% between uncontrolled and controlled, compared to 25% for the current controller. The previous controller increased the average fuel to 2.4% which is well beyond the detection error compared to less than 1% for the proposed optimal controller.

VIII. Conclusions

The controller presented successfully controlled a SI engine to reduce cyclic dispersion under high EGR conditions. The system is modeled under a non-strict feedback nonlinear discrete-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. Experimental results show the stability of the closed-loop system under a variety of set points.

Appendix

Proof of Theorem 1: Define the Lyapunov function

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

where $0<\gamma_i, i\in\{1,\dots,6\}$ 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 (19), (26), (45), and (55), 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 (16) and (17), respectively; the system errors $e_1(k+1)$ and $e_2(k+1)$ are defined in (38) and (49), respectively; and

$\alpha_i, i \in \{1,2,3,4\}$ are NN adaptation gains. The Lyapunov function (A.1) obviates the need for CE condition. Taking the first term and the first difference using (38) to get

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

Take the second term, substitute (49), and simplify

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

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

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

Take the fourth term, substitute (26), and simplify

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

Take the fifth term, substitute (45), and simplify

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

Take the sixth term, substitute (55), and simplify

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

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 (16), and simplify

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

Take the ninth term, substitute (17), and simplify

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

Take the tenth and final term, substitute (18), and simplify

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

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

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

where

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

Select

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

This implies $\Delta J(k) < 0$ as long as (57) through (59) hold *and* any *one* of the following hold [8]

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

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 discrete-time systems using 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] Q. Yang and S. Jagannathan, "Online Reinforcement Learning-Based Neural Network Controller Design for Affine Nonlinear Discrete-Time Systems," in *American Control Conference*, to appear 2007.

[10] 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.

[11] J. B. Vance, A. Singh, B. Kaul, S. Jagannathan, and J. Drallmeier, "Neural Network Controller Development and Implementation for Spark Ignition Engines with High EGR Levels," *IEEE Tans. Neural Networks, to appear in* 2007.