1-1-2006

Neural Network Control of Spark Ignition Engines with High EGR Levels

A. Singh

Brian C. Kaul

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

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

et. al. For a complete list of authors, see http://scholarsmine.mst.edu/mec_aereng_facwork/3448

Recommended Citation

A. Singh et al., "Neural Network Control of Spark Ignition Engines with High EGR Levels," *The 2006 IEEE International Joint Conference on Neural Network Proceedings*, Institute of Electrical and Electronics Engineers (IEEE), Jan 2006.
The definitive version is available at https://doi.org/10.1109/IJCNN.2006.247201

2006 International Joint Conference on Neural Networks
Sheraton Vancouver Wall Centre Hotel, Vancouver, BC, Canada
July 16-21, 2006

Neural Network Control of Spark Ignition Engines with High EGR Levels

A. Singh, J. B. Vance, B. Kaul, S. Jagannathan and J. Drallmeier

Abstract— **Research has shown substantial reductions in the oxides of nitrogen (NO_x) concentrations by using 10% to 25% exhaust gas recirculation (EGR) in spark ignition (SI) engines [1]. However under high EGR levels the engine exhibits strong cyclic dispersion in heat release which may lead to instability and unsatisfactory performance. A suite of neural network (NN)-based output feedback controllers with and without reinforcement learning is developed to control the SI engine at high levels of EGR even when the engine dynamics are unknown by using fuel as the control input. A separate control loop was designed for controlling EGR levels. The neural network controllers consists of three NN: a) A NN observer to estimate the states of the engine such as total fuel and air; b) a second NN for generating virtual input; and c) a third NN for generating actual control input. For reinforcement learning, an additional NN is used as the critic. The stability analysis of the closed loop system is given and the boundedness of all signals is ensured without separation principle. Online training is used for the adaptive NN and no offline training phase is needed.**

Experimental results obtained by testing the controller on a research engine indicate an 80% drop of NO_x from stoichiometric levels using 10% EGR. Moreover, unburned hydrocarbons drop by 25% due to NN control as compared to the uncontrolled scenario.

I. Introduction

Today's automobiles utilize sophisticated microprocessor-based engine control systems to meet stringent federal regulations governing fuel economy and the emission of carbon monoxide (CO), oxides of nitrogen (NO_x) and hydrocarbons (HC). Current efforts are directed at reducing the total amount of emissions and fuel consumption. Operating a spark ignition engine lean can reduce the NO_x and will improve the fuel efficiency [1-3]. Similarly, substantial reductions in NO_x concentrations have been achieved with 10% to 25% EGR [2] along with reduction in specific fuel consumption. For example, if an engine can tolerate 20 to 25% EGR, reduction in engine-out NOx on the order of 90-95% can be realized.

However, EGR also reduces the combustion rate, which makes stable combustion [4-5] more difficult to achieve. High levels of EGR present in a spark ignition (SI) engine lead to cyclic dispersion in the heat release map of the SI engine. Under such conditions a large number of misfires develop causing problems in drivability due to cycle-to-cycle variations in output as well as large increases in unburned hydrocarbons.

Several researchers [3, 6-9] have studied lean combustion engine control technology but few results have been reported for the EGR case. Conventional control schemes [3] have been found incapable of reducing the cyclic dispersion to the levels needed to implement these concepts. Moreover, the total amount of fuel and air in a given cylinder is normally not measurable which necessitates the development of output feedback control schemes.

Several output feedback controller designs in discrete time are proposed for the signal-input-single-output (SISO) nonlinear systems [10-12]. However, no output feedback control scheme currently exists for the proposed class of nonstrict feedback nonlinear discrete-time systems. To overcome the need for complex engine dynamics and to make the controller practical, a heat release based neural network (NN)-based output feedback controller is proposed by using the NN universal approximation property [13].

In the proposed output feedback controller, the NN observer is designed to estimate the total mass of air and fuel in the cylinder by using a measured value of heat release. The estimated values are used by an adaptive NN controller. The proposed design relaxes the persistence of excitation condition, certainty equivalence principle and linear in the unknown parameter assumptions. Finally, separation principle is also not required. The NN weights are tuned on-line, with no off-line learning phase required.

II. Engine as a Nonlinear Discrete-Time System

A. Non-Strict Nonlinear System Description

Consider the following non-strict feedback nonlinear system described by

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

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

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 $|d_2(k)| < d_{2m}$. Here d_{1m} and d_{2m} are unknown positive scalars.

This work was supported in part by the National Science Foundation under Grants ECS#0296191 and ECS#0327877, Department of Education through GAANN program and Intelligent Systems Center.

A. Singh, J. B. Vance, B. Kaul, S. Jagannathan, and J. Drallmeier are with the University of Missouri-Rolla. Email: contact author: sarangap@umr.edu.

0-7803-9490-9/06/$20.00/©2006 IEEE

Equations (1) and (2) represent a discrete-time nonlinear system in non-strict feedback form [16], since $f_1(\cdot)$ and $g_1(\cdot)$ are functions of both $x_1(k)$ and $x_2(k)$, unlike in the case of strict feedback nonlinear system, where $f_1(\cdot)$ and $g_1(\cdot)$ are a function of $x_1(k)$ only [10-12]. Control of nonstrict feedback nonlinear systems is introduced in [16] since no known results are available in the literature. Engine dynamics can be expressed in this form.

B. Engine Dynamics

Daw, Finney, Green, Kennel and Thomas (1996) [5] and Daw et al. (1998) [4] developed a mathematical representation of the spark ignition (SI) engine to investigate nonlinear cycle dynamics under lean conditions and high EGR levels [2]. The residual air and fuel passed from one cycle to the next make the model deterministic. Stochastic effects are embodied in random fluctuations of parameters like injected air-fuel ratio or residual fraction. Actual variations in parameters due to complex processes like temperature and pressure effects, turbulence, fuel vaporization etc are not modeled but assumed to add stochastic noise to the engine output. The model for the EGR case is shown below.

$$x_1(k+1) = F(k)[x_1(k) - R.CE(k)x_2(k) + r_{O_2}(k) + r_{N_2}(k)] + x_{1new}(k) + d_1(k) \quad (3)$$

$$x_2(k+1) = F(k)(1-CE(k))x_2(k) + x_{2new}(k) + u(k) + d_2'(k) \quad (4)$$

$$x_3(k+1) = F(k)(r_{CO_2}(k) + r_{H_2O}(k) + r_{N_2}(k) + x_3(k) + EGR(k)) \quad (5)$$

$$y(k) = x_2(k)CE(k) \quad (6)$$

$$\varphi(k) = R\frac{x_2(k)}{x_1(k)}.\left[1-\gamma\frac{x_3(k)+EGR(k)}{(x_2(k)+x_1(k)+x_3(k)+EGR(k))}\right] \quad (7)$$

$$CE(k) = \frac{CE_{\max}}{1+100^{-(\varphi(k)-\varphi_m)/(\varphi_u-\varphi_l)}},\ \varphi_m = \frac{\varphi_u+\varphi_l}{2}, \quad (8)$$

$$r_{H_2O}(k) = \gamma_{H_2O}x_2(k)CE(k) \quad (9)$$

$$r_{O_2}(k) = \gamma_{O_2}x_2(k)CE(k) \quad (10)$$

$$r_{N_2}(k) = \gamma_{N_2}R.x_2(k)CE(k) \quad (11)$$

$$r_{CO_2}(k) = \gamma_{CO_2}x_2(k)CE(k) \quad (12)$$

where $x_1(k)$, $x_2(k)$ and $x_3(k)$ are the total mass of air, fuel and inert gases respectively. The heat release at the k^{th} time instant is given by $y(k)$, $CE(k)$ is the combustion efficiency and $0 < CE_{\min} < CE(k) < CE_{\max}$, $CE_{\max}$ is the maximum combustion efficiency and it is a constant, $F(k)$ is the residual gas fraction which is bounded $0 < F_{\min} < F(k) < F_{\max}$, R is the stoichiometric air-fuel ratio,~15.13 for iso-octane, $u(k)$ is the small change in fuel per cycle, $\varphi(k)$ is the equivalence ratio, $\varphi_m, \varphi_l, \varphi_u$ are equivalence ratio system parameters, $r_{H_2O}(k)$, $r_{O_2}(k)$, $r_{N_2}(k)$ and $r_{CO_2}(k)$ are the mass of water, oxygen, nitrogen and carbon dioxide respectively. It should be noted that the residual oxygen combines proportionally with the residual nitrogen to form residual air. The fraction of total nitrogen leftover after this is the residual inert nitrogen. γ is a constant and $\gamma_{H_2O}, \gamma_{O_2}, \gamma_{N_2}$ and γ_{CO_2} are constant parameters associated with water, oxygen, nitrogen and carbon dioxide respectively. The Daw model uses hydrogen and carbon proportions of the fuel along with the EGR fraction to determine the residual fractions using stoichiometry. $d_1'(k)$ and $d_2'(k)$ are unknown but bounded disturbances. It can be seen that the SI engine with EGR levels has highly nonlinear dynamics with $CE(k)$ and $F(k)$ being unknown and cannot be measured.

Remark 1: It is important to note that the output is a nonlinear function of the states unlike in many papers where the output is a linear function of states.

Remark 2: For lean engine operation, the inert gas equation (5) is not used and there are fewer parameters in (3) and (4).

C. Engine Dynamics Using Nominal Values

Substituting (6) into both (3) and (4), we get

$$x_1(k+1) = F(k)[x_1(k) - R.y(k) + r_{O_2}(k) + r_{N_2}(k)] + x_{1new}(k) + d_1(k) \quad (13)$$

$$x_2(k+1) = F(k)(x_2(k) - y(k)) + x_{2new}(k) + u(k) + d_2'(k) \quad (14)$$

In real engine operation, the fresh air x_{1new}, fresh fuel x_{2new} and residual gas fraction, $F(k)$ can all be viewed as nominal values plus some small and bounded disturbances. The inert gases include the residual exhaust gases in the cylinder and the EGR fraction. Equation (5) will not be considered for controller development as a separate control loop designed to control EGR levels makes the inert gases evolve into a stable value. Therefore, it is not included here. Consider,

$$x_{1new}(k) = x_{1new0} + \Delta x_{1new}(k) \quad (15)$$

$$x_{2new}(k) = x_{2new0} + \Delta x_{2new}(k) \quad (16)$$

$$F(k) = F_0(k) + \Delta F(k) \quad (17)$$

where x_{1new0}, x_{2new0} and F_0 are the known nominal fresh air, fuel and residual gas fraction values. Δx_{1new0}, Δx_{2new0} and ΔF_0 are unknown yet bounded disturbances on those values whose bounds are given by,

$$0 \le |\Delta x_{1new}(k)| \le \Delta x_{1newM} \quad (18)$$

$$0 \le |\Delta x_{2new}(k)| \le \Delta x_{2newM} \quad (19)$$

$$0 \le |\Delta F(k)| \le \Delta F_M \quad (20)$$

Substituting these values into the system model we can get the state equations in the following form,

$$x_1(k+1) = (F_0(k) + \Delta F(k))[x_1(k) - R.CE(k)x_2(k) + r_{O_2}(k) + r_{N_2}(k)] + x_{1new0} + \Delta x_{1new}(k) + d_1(k) \quad (21)$$

$$x_2(k+1) = (F_0(k) + \Delta F(k))(1 - CE(k))x_2(k) + x_{2new0} + \Delta x_{2new}(k) + u(k) + d_2'(k) \quad (22)$$

III. NEURAL NETWORK BASED OBSERVER DESIGN

First a NN is used to predict the value of the heat release for the next burn cycle which will be used subsequently by the observer to predict the states of the system. The inert gases can be calculated directly if the air and fuel values are known so they are not estimated. The heat release for the next burn cycle is given by

$$y(k+1) = x(k+1)CE(k+1) \quad (23)$$

A. Observer Structure

From (23), the heat release for the next cycle $y(k+1)$ can be approximated by using a one layer neural network as

$$y(k+1) = w_1^T \phi_1(v_1^T z_1(k)) + \varepsilon_1(z_1(k)) \quad (24)$$

where the input to the NN is taken as $z_1(k) = [x_1(k), x_2(k), y(k), u(k)]^T \in R^4$, the matrix $w_1 \in R^{4 \times n_1}$ and $v_1 \in R^{4 \times n_1}$ represent the output and hidden layer weights, $\phi_1(.)$ represents the hidden layer activation function, n_1 denotes the number of the nodes in the hidden layer, and $\varepsilon_1(z_1(k)) \in R$ is the functional approximation error. It has been demonstrated that, if the hidden layer weight v_1, is chosen initially at random and held constant and the number of hidden layer nodes is sufficiently large, the approximation error $\varepsilon_1(z_1(k))$ can be made arbitrarily small over the compact set since the activation function forms a basis according to [13].

For simplicity, we define

$$\phi_1(z_1(k)) = \phi_1(v_1^T z_1(k)) \quad (25)$$

$$\varepsilon_1(k) = \varepsilon_1(z_1(k)) \quad (26)$$

Given (24) and (25), (26) is re-written as

$$y(k+1) = w_1^T \phi_1(z_1(k)) + \varepsilon_1(k) \quad (27)$$

Since states $x_1(k)$ and $x_2(k)$ are not measurable, $z_1(k)$ is not available either. Using the estimated values $\hat{x}_1(k)$, $\hat{x}_2(k)$ and $\hat{y}(k)$ instead of $x_1(k)$, $x_2(k)$ and $y(k)$ the proposed heat release observer can be given as,

$$\begin{aligned} \hat{y}(k+1) &= \hat{w}_1^T \phi_1(v_1^T \hat{z}_1(k)) + l_1 \tilde{y}(k) \\ &= \tilde{w}_1^T(k)\phi_1(\hat{z}_1(k)) + l_1 \tilde{y}(k) \end{aligned} \quad (28)$$

where $\hat{y}(k+1)$ is the predicted heat release, $\hat{w}_1(k) \in R^{n_1}$ is the actual output layer weights, the input to the NN is taken as $\hat{z}_1(k) = [\hat{x}_1(k), \hat{x}_2(k), \hat{y}(k), u(k)]^T \in R^4, l \in R$ is the observer gain, $\tilde{y}(k)$ is the heat release estimation error, which is defined as

$$\tilde{y}(k) = \hat{y}(k) - y(k) \quad (29)$$

and $\phi_1(\hat{z}_1(k))$ represents $\phi_1(v_1^T \hat{z}_1(k))$ for the purpose of simplicity.

Using the heat release estimation error, the proposed observer is given as the following form,

$$\hat{x}_1(k+1) = x_{1new0}(k) + F_o \hat{x}_1(k) - R \cdot F_o \cdot \hat{y}(k) + l_2 \tilde{y}(k) \quad (30)$$

$$\hat{x}_2(k+1) = F_o(\hat{x}_2(k) - \hat{y}(k)) + (x_{2new0}(k) + u(k)) + l_3 \tilde{y}(k) \quad (31)$$

where $l_2 \in R$ and $l_3 \in R$ are observer gains. The term $F_o(r_{O_2} + r_{N_2})$ has been pulled out from equation (30) as there are no nominal values available for the inert gases. The error introduced by this will be taken up as part of the air estimation error. Equations (26), (28) and (29) represent the dynamics of the observer to estimate the states of $x_1(k)$ and $x_2(k)$.

B. Observer Error Dynamics

Define the state estimation errors as:

$$\tilde{x}_i(k) = \hat{x}_i(k) - x_i(k), i = 1,2 \quad (32)$$

Combining (21) through (26), we obtain the estimation error dynamics as

$$\begin{aligned} \tilde{x}_1(k+1) &= F_o \tilde{x}_1(k) + (l_2 - R \cdot F_o)\tilde{y}(k) - \Delta x_{1new}(k) \\ &\quad - \Delta F(k) x_1(k) + R \Delta F(k) y(k) - F_o(r_{O_2} + r_{N_2}) \\ &\quad - \Delta F(r_{O_2} + r_{N_2}) - d_1(k) \end{aligned} \quad (33)$$

$$\begin{aligned} \tilde{x}_2(k+1) &= F_o \tilde{x}_2(k) + (l_3 - F_o)\tilde{y}(k) \\ &\quad - \Delta F(k)(x_2(k) - y(k)) - \Delta x_{2new}(k) - d_2(k) \end{aligned} \quad (34)$$

$$\begin{aligned} \tilde{y}(k+1) &= \hat{w}_1^T(k)\phi_1(\hat{z}_1(k)) + l_1 \tilde{y}(k) - w_1^T \phi_1(z_1(k)) - \varepsilon_1(k) \\ &= (\hat{w}_1(k) - w_1)^T \phi_1(\hat{z}_1(k)) + w_1^T(\phi_1(\hat{z}_1(k)) - \phi_1(z_1(k))) - \varepsilon_1(k) \\ &= \tilde{w}_1^T(k)\phi_1(\hat{z}_1(k)) + \tilde{w}_1^T(k)\phi_1(\tilde{z}_1(k)) - \varepsilon_1(k) \\ &= \zeta_1(k) + w_1^T \phi_1(\tilde{z}_1(k)) - \varepsilon_1(k) \end{aligned} \quad (35)$$

where, $\tilde{w}_1(k-1) = \hat{w}_1(k) - w_1$, (36)

$$\zeta_1(k) = \tilde{w}_1^T(k)\phi_1(z_1(k)) \quad (37)$$

and for the purpose of simplicity, $(\phi_1(\hat{z}_1(k)) - \phi_1(z_1(k)))$ is written as $(\phi_1(\tilde{z}_1(k))$.

IV. ADAPTIVE NN OUTPUT FEEDBACK CONTROLLER DESIGN

The control objective of maintaining the heat release constant is achieved by holding the fuel and combustion efficiency within a close bound, i.e., the heat release is driven to a target heat release y_d. Given y_d and the engine dynamics (3) – (5), we could obtain the nominal values for the total mass of air and fuel in the cylinder, x_{1d} and x_{2d}, respectively. By driving the states $x_1(k)$ and $x_2(k)$ to approach to their respective nominal values x_{1d} and x_{2d}, $y(k)$ will approach y_d. By developing a controller to maintain the EGR at a constant level separately, we can see that the inert gases evolve into a stable value since equation (5) can be viewed as a feedback linearizable nonlinear discrete-time system with $F(k)$ being less than one and the weights of the gases kept constant with minor variations. The controller for the EGR system (5) is developed separately and not presented here. With the estimated

states $\hat{x}_1(k)$ and $\hat{x}_2(k)$, the controller design follows the backstepping technique [14-15]. The details are given in the following sections.

A. Adaptive NN Output Feedback Controller Design

The controller design is now given.

Step 1: Virtual controller design.

Define the error between actual and desired air as

$$e_1(k) = x_1(k) - x_{1d} \tag{38}$$

which can be rewritten as

$$\begin{aligned} e_1(k+1) &= x_1(k+1) - x_{1d} \\ &= F(k)[x_1(k) - R.CE(k)x_2(k) + r_{O_2}(k) \\ &\quad + r_{N_2}(k)] - x_{1d} \quad + x_{1new}(k) + d_1(k) \end{aligned} \tag{39}$$

For simplicity let us denote

$$f_1(k) = F(k)[x_1(k) + r_{O_2}(k) + r_{N_2}(k)] + x_{1new}(k) - x_{1d} \tag{40}$$

$$g_1(k) = R \cdot F(k)CE(k) \tag{41}$$

Then the system error equation can be expressed as

$$e_1(k+1) = f_1(k) - g_1(k)x_2(k) + d_1(k) \tag{42}$$

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

$$x_{2d}(k) = \frac{f_1(k)}{g_1(k)} \tag{43}$$

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

$$x_{2d}(k) = w_2^T \phi_2(v_2^T x(k)) + \varepsilon_2(x(k)) = w_2^T \phi_2(x(k)) + \varepsilon_2(x(k)) \tag{44}$$

where the input in the state $x(k) = [x_1(k), x_2(k)]^T$, $w_2 \in R^{n_2}$ and $v_2 \in R^{2\times n_1}$ denote the constant ideal output and hidden layer weights, n_2 is the number of nodes in the hidden layer, the hidden layer activation function $\phi_2(v_2^T x(k))$ is simplified as $\phi_2(x(k))$ and $\varepsilon_2(x(k))$ is the approximation error. Since both $x_1(k)$ and $x_2(k)$ are unavailable, the estimated state $\hat{x}(k)$ is selected as the NN input.

Consequently, the virtual control input is taken as

$$\hat{x}_{2d}(k) = \hat{w}_2^T(k)\phi_2(v_2^T\hat{x}(k)) = \hat{w}_2^T(k)\phi_2(\hat{x}(k)) \tag{45}$$

where $\hat{w}_2(k) \in R^{n_2}$ is the actual weight matrix for the first action NN . Define the weight estimation error by

$$\tilde{w}_2(k) = \hat{w}_2(k) - w_2 \tag{46}$$

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

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

Equation (35) can be expressed using (47) for $x_2(k)$ as

$$e_1(k+1) = f_1(k) - g_1(k)(e_2(k) + \hat{x}_{2d}(k)) + d_1(k), \tag{48}$$

or equivalently

$$\begin{aligned} e_1(k+1) &= f_1(k) - g_1(k)(e_2(k) + x_{2d}(k) - x_{2d}(k) \\ &\quad + x_{2d}(k)) + d_1(k) \\ &= f_1(k) - g_1(k)(e_2(k) + x_{2d}(k) - x_{2d}(k) \\ &\quad + \hat{x}_{2d}(k)) + d_1(k) \\ &= -g_1(k)(e_2(k) - x_{2d}(k) + \hat{x}_{2d}(k)) + d_1(k) \\ &= -g_1(k)(e_2(k) + \hat{w}_2^T(k)\phi_2(\hat{x}(k)) - w_2^T\phi_2(x(k)) \\ &\quad - \varepsilon_2(x(k))) + d_1(k) \end{aligned} \tag{49}$$

Similar to (35), (49) can be further expressed as

$$e_1(k+1) = -g_1(k)(e_2(k) - \zeta_2(k) + \tilde{w}_2^T\phi_2(\tilde{x}(k)) - \varepsilon_2(x(k))) + d_1(k) \tag{50}$$

where $\zeta_2(k) = \tilde{w}_2^T(k)\phi_2(\hat{x}(k))$ (51)

$$w_2^T\phi_2(\tilde{x}(k)) = w_2^T(\phi_2(\hat{x}(k)) - \phi_2(x(k))) \tag{52}$$

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

Rewriting the error $e_2(k)$ from (47) as

$$\begin{aligned} e_2(k+1) &= x_2(k+1) - \hat{x}_{2d}(k+1) \\ &= (1 - CE(k))F(k)x_2(k) + (MF(k) + u(k)) - \hat{x}_{2d}(k+1) + d_2(k) \end{aligned} \tag{53}$$

For simplicity, let us denote,

$$x_2(k+1) = F(k)(1 - CE(k))x_2(k) + x_{2new}(k) \tag{54}$$

Equation (53) can be written as

$$e_2(k+1) = f_2(k) + u(k) - \hat{x}_{2d}(k+1) + d_2(k) \tag{55}$$

where $\hat{x}_{2d}(k+1)$ is the future value of $\hat{x}_{2d}(k)$. Here, $\hat{x}_{2d}(k+1)$ is not available in the current time step. However, from (43) and (45), it can be clear that $\hat{x}_{2d}(k+1)$ is a smooth nonlinear function of the state $x(k)$ and virtual control input $\hat{x}_{2d}(k)$. Another NN can be used to approximate the value of $\hat{x}_{2d}(k+1)$ since an NN is a first order predictor since the proposed NNs use a semi-recurrent architecture. Other methods via filtering approach [18] do exist in the literature in order to obtain this value.

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

$$\begin{aligned} u_d(k) &= (-f_2(k) + \hat{x}_{2d}(k+1)) \\ &= w_3^T\phi_3(v_3^T z_3(k)) + \varepsilon_3(z_3(k)) \\ &= w_3^T\phi_3(z_3(k)) + \varepsilon_3(z_3(k)) \end{aligned} \tag{56}$$

where $w_3 \in R^{n_3}$ and $v_3 \in R^{3\times n_3}$ denote the constant ideal output and hidden layer weights, n_3 is the hidden layer nodes number, the hidden layer activation function $\phi_3(v_3^T z_3(k))$ is simplified as $\phi_3(z_3(k))$, $\varepsilon_3(z_3(k))$ is the approximation error, $z_3(k) \in R^3$ is the NN input, which is given by (55). Considering the fact both $x_1(k)$ and $x_2(k)$ cannot be measured, $z_3(k)$ is substituted with $\hat{z}_3(k) \in R^3$ where

$$z_3(k) = [x(k), \hat{x}_{2d}(k)]^T \in R^3 \tag{57}$$

and

$$\hat{z}_3(k) = [\hat{x}(k), \hat{x}_{2d}(k)]^T \in R^3 \tag{58}$$

Now define

$$\hat{e}_1(k) = \hat{x}_1(k) - x_{1d}, \tag{59}$$

and

$$\hat{e}_2(k) = \hat{x}_2(k) - x_{2d} \tag{60}$$

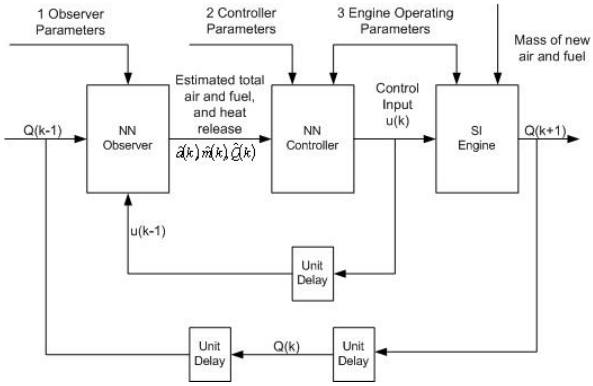

Fig.1. Neuro-controller structure.

The actual control input is now selected as

$$\begin{aligned} u(k) &= \hat{w}_3^T(k)\phi_3(v_c^T\hat{z}_3(k)) + l_4\hat{e}_2(k) \\ &= \hat{w}_3^T(k)\phi_3(\hat{z}_3(k)) + l_4\hat{e}_2(k) \end{aligned} \tag{61}$$

where $\hat{w}_3^T \in R^{n_3}$ is the actual output layer weights, $l_4 \in R$ is the controller gain selected to stabilize the system. Similar to the derivation of (39), combining (55), (56) with (61) yields

$$e_2(k+1) = l_4\hat{e}_2(k) + \xi_3(k) + w_3^T\phi_3(\tilde{z}(k)) - \varepsilon_3(z_3(k)) + d_2(k) \tag{62}$$

where

$$\tilde{w}_3(k) = \hat{w}_3(k) - w_3 \tag{63}$$

$$\xi_3(k) = \tilde{w}_3^T(k)\phi_3(\hat{z}_3(k)) \tag{64}$$

and

$$w_3^T\phi_3(\tilde{z}(k)) = w_3^T(\phi_3(\hat{z}_3(k)) - \phi_3(z_3(k))) \tag{65}$$

Equations (50) and (62) represent the closed-loop error dynamics. It is required to show that the estimation error (29) and (32), the system errors (50) and (62) and the NN weight matrices $\hat{w}_1(k)$, $\hat{w}_2(k)$ and $\hat{w}_3(k)$ are bounded. Fig. 1 shows the block diagram of the final structure of the designed neuro-controller where *a(k) and m(k)* denote the mass of air and fuel $x_1(k)$ and $x_2(k)$ respectively and *Q(k)* denotes heat release, y(k).

B. Weight Updates for Guaranteed Performance

Assumption 1 (Bounded Ideal Weights): Let w_1, w_2 and w_3 be the unknown output layer target weights for the observer and two action NNs and assume that they are bounded above so that

$$\|w_1\| \le \|w_{1m}\|, \|w_2\| \le \|w_{2m}\|, \text{ and } \|w_3\| \le \|w_{3m}\|, \tag{66}$$

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

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

$$\|\phi_i(\cdot)\| \le \phi_{im}, i = 1,2,3 \tag{67}$$

where $\phi_{im}, i = 1,2,3$ are the upper bounds.

Assumption 2 (Bounded NN Approximation Error): The NN approximation errors $\varepsilon_1(z_1(k))$, $\varepsilon_2(x(k))$ and $\varepsilon_3(z_3(k))$ are bounded over the compact set by ε_{1m}, ε_{2m} and ε_{3m}, respectively.

Theorem 1: Consider the system given in (3) – (5) and let the Assumptions 1 and 2 hold. Let the unknown disturbances be bounded by $|d_1(k)| \le d_{1m}$ and $|d_2(k)| \le d_{2m}$, respectively. Let the observer NN weight tuning be given by

$$\hat{w}_1(k+1) = \hat{w}_1(k) - \alpha_1\phi_1(\hat{z}_1(k))(\hat{w}_1^T(k)\phi_1(\hat{z}_1(k) + l_5\tilde{y}(k)) \tag{68}$$

with the virtual control NN weight tuning be provided by

$$\hat{w}_2(k+1) = \hat{w}_2(k) - \alpha_2\phi_2(\hat{x}(k))(\hat{w}_2^T(k)\phi_2(\hat{x}(k) + l_6\hat{e}_1(k)) \tag{69}$$

and the control NN weight tuning be provided by

$$\hat{w}_3(k+1) = \hat{w}_3(k) - \alpha_3\phi_3(\hat{z}_3(k))(\hat{w}_3^T(k)\phi_3(\hat{z}_3(k) + l_7\hat{e}_2(k)) \tag{70}$$

where $\alpha_1 \in R, \alpha_2 \in R, \alpha_3 \in R,$ and $l_5 \in R, l_6 \in R,$ and $l_7 \in R$ are design parameters. Let the system observer be given by (28), (29) and (30), virtual and actual control inputs be defined as (45) and (61), respectively. The estimation errors (33) through (35), the tracking errors (50) and (62), and the NN weight estimates $\hat{w}_1(k)$, $\hat{w}_2(k)$ and $\hat{w}_3(k)$ are uniformly ultimately bounded provided the design parameters are selected as

(a) $0 < \alpha_i \|\phi_i(k)\|^2 < 1, \quad i = 1,2,3$ (71)

(b) $l_3^2 < 1 - \dfrac{(l_1 - R\cdot F_0)^2}{6R^2\cdot\Delta F_m^{\,2}} - \dfrac{(l_2 - F_0)^2}{6\Delta F_m^{\,2}} - 4l_5^2$ (72)

(c) $l_6^2 < \min\left(\dfrac{(1-F_0^{\,2})}{18R^2\cdot\Delta F_m^{\,2}}, \dfrac{1}{18R^2}\right),$ (73)

(d) $l_4^2 + 6l_7^2 < \min\left(\dfrac{(1-F_0^{\,2})}{6\Delta F_m^{\,2}}, \dfrac{1}{3}\right),$ (74)

Remark: For instance, given R = 14.6, F_0 = 0.14, and ΔF_m = 0.02, we can select l_1 = 1.99, l_2 = 0.13, l_3 = 0.4, l_4 = 0.14, l_5 = 0.25, l_6 = 0.016, and l_7 = 0.1667 to satisfy (71) – (74).

Remark 4: Given the hypotheses, this proposed neuro-output NN control scheme and the weight updating rules in *Theorem 1* with the parameter selection based on (65) through (68), the state $x_2(k)$ approaches the operating point x_{2d}.

Remark 5: It is important to note that in this theorem there is no persistence of excitation condition, certainty equivalence, linearity in the unknown parameter assumptions for the NN observer and NN controller. In our proof, the Lyapunov function consists of the observer estimation errors, system errors, and the NN estimation

errors and therefore separation principle is not used.

V. Reinforcement Weight Updating

In this section, we develop an alternate weight updating rules based on reinforcement learning where actor-critic architecture is utilized. 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

$$p(k) = \begin{cases} 0, & if \left(|\hat{e}_1(k)| + |\hat{e}_2(k)|\right) \le c, \\ 1, & otherwise \end{cases} \quad (75)$$

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

$$Q(k) = \alpha^N p(k+1) + \alpha^{N-1} p(k+2) + \cdots + \alpha^{k+1} p(N), \quad (76)$$

where $\alpha \in \Re$ and $0 < \alpha < 1$, and N is the depth or horizon. The term $Q(k)$ is viewed here as the future system performance measure.

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

$$e_c(k) = \hat{Q}(k) - \alpha\left(\hat{Q}(k-1) - \alpha^N p(k)\right), \quad (77)$$

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

$$\hat{Q}(k) = \hat{w}_3^T(k)\phi_3\left(v_3^T x(k)\right) = \hat{w}_3^T(k)\phi_3(k), \quad (78)$$

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 $\hat{x}(k) \in \Re^2$. The objective function to be minimized by the critic NN is defined as

$$E_c(k) = \frac{1}{2} e_c^2(k). \quad (79)$$

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

$$\hat{w}_3(k+1) = \hat{w}_3(k) + \Delta\hat{w}_3(k), \quad (80)$$

where

$$\Delta\hat{w}_3(k) = \alpha_3\left[-\frac{\partial E_c(k)}{\partial \hat{w}_3(k)}\right], \quad (81)$$

or

$$\hat{w}_3(k+1) = \hat{w}_3(k) - \alpha_3\phi_3(k)\left(\hat{Q}(k) + \alpha^{N+1}p(k) - \alpha\hat{Q}(k-1)\right)^T, \quad (82)$$

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

$$e_{a1}(k) = \zeta_1(k) + (\hat{Q}(k) - Q_d(k)), \quad (83)$$

where $\zeta_1(k)$ is defined in (31), $e_{a1}(k) \in \Re$, 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", i.e., to indicate that at every step, the nonlinear system can track the reference signal well. Thus, (34) becomes

$$e_{a1}(k) = \zeta_1(k) + \hat{Q}(k), \quad (84)$$

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

$$E_{a1}(k) = \frac{1}{2} e_{a1}^2(k), \quad (85)$$

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

$$\hat{w}_1(k+1) = \hat{w}_1(k) + \Delta\hat{w}_1(k), \quad (86)$$

where

$$\Delta\hat{w}_1(k) = \alpha_1\left[-\frac{\partial E_{a1}(k)}{\partial \hat{w}_1(k)}\right], \quad (87)$$

or

$$\hat{w}_1(k+1) = \hat{w}_1(k) - \alpha_1\phi_1(k)\left(\hat{Q}(k) + \zeta_1(k)\right), \quad (88)$$

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

The NN weight updating rule in (88) 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

$$\zeta_1(k) = -\hat{e}_1(k+1) - l_1\hat{e}_1(k) - \hat{e}_2(k) + \varepsilon_1\left(x(k)\right) + d_1(k). \quad (89)$$

Substituting (89) into (88), we get

$$\hat{w}_1(k+1) = \hat{w}_1(k) - \alpha_1\phi_1(k)\hat{Q}(k) - \alpha_1\phi_1(k)\left(-\hat{e}_1(k+1) - l_1\hat{e}_1(k) - \hat{e}_2(k) + \varepsilon_1\left(x(k)\right) + d_1(k)\right). \quad (90)$$

Assume that bounded disturbance $d_1(k)$ and the NN approximation error $\varepsilon_1(x(k))$ are zeros for weight tuning implementation, then (84) is rewritten as

$$\hat{w}_1(k+1) = \hat{w}_1(k) - \alpha_1\phi_1(k)\hat{Q}(k) + \alpha_1\phi_1(k)\left(\hat{e}_1(k+1) + l_1\hat{e}_1(k) + \hat{e}_2(k)\right). \quad (91)$$

Equation (91) is the adaptive critic based weight updating rule for the first action NN. Similarly, the weight updating rule for the second action NN $\hat{w}_2^T(k)\phi_2(k)$ is given next.

D. Weight Updating Rule for the Second Action NN

Define

$$e_{a2}(k) = \sqrt{g_2(k)}\zeta_2(k) + \frac{\hat{Q}(k)}{\sqrt{g_2(k)}}, \quad (92)$$

where $\zeta_2(k)$ is defined in (45), $g_2(k) \in \Re^+$ 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

$$\hat{w}_2(k+1) = \hat{w}_2(k) - \alpha_2 \phi_2(k)\left(\hat{Q}(k) + \hat{e}_2(k+1) - l_2 \hat{e}_2(k)\right), \quad (93)$$

One can use these weight tuning schemes and prove the closed-loop stability.

VI. Controller Hardware Design

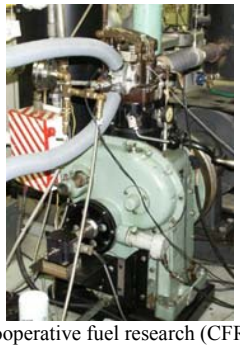

Fig. 2. Cooperative fuel research (CFR) engine.

The experimental setup involves a Cooperative Fuel Research (CFR) engine, shown in Fig.2, on which the controller operates. The CFR is operated at 1000 RPM. Being a single cylinder engine, dynamics introduced by multiple cylinders are avoided. Shaft encoders are mounted on the cam and crank shafts that return start-of-cycle and crank angle signals, respectively. There are 720° of crank angle per engine cycle, so a crank angle degree is detected every 167 microseconds. For the exhaust-gas-recirculation (EGR) portion of gaseous intake, nitrogen is used. EGR is comprised mainly of inert gases from the previous combustion cycle, so nitrogen, an inert gas in the combustion process is used in place of the residual inert gases. This allows for an accurate fraction of EGR to be introduced to the cylinder.

Heat release for a given engine cycle is calculated by integrating in-cylinder pressure and volume over time. In-cylinder pressure is measured every half crank angle degree during combustion, which is considered from 345° to 490°, for a total of 290 pressure measurements. At 1000 RPM pressure measurements must be made every 83.3 microseconds. The calculation window is 106° wide or 17.667 milliseconds. In this time all engine-to-PC-to-engine communication are completed. The algorithm designed uses 15 neurons to approximate the output.

VII. Experimental Results

An equivalence ratio of one was maintained with variation of 1%, R = 15.13 for iso-octane, residual gas fraction F = 0.09, mass of nominal new air = 0.52485, mass of nominal new fuel = 0.02428, the standard deviation of mass of new fuel is 0.007, cylinder volume in moles = 0.021, molecular weight of fuel = 114, molecular weight of air = 28.84, $\phi_u = 0.665$, $\phi_l = 0.645$, maximum combustion efficiency = 1, and the gains (l_1, l_2) of backstepping controller are selected at 0.1 and placed diagonally. EGR was assumed to be a mixture of Nitrogen and Oxygen gases with a molecular weight of 30.4 and the hydrogen carbon ratio was 1.87. All initial values of air, fuel and inert gases were chosen to be 0. The neural networks were designed to with learning rates of 0.01 each.

The activation functions used were the hyperbolic tangent sigmoid functions. The results for engine operation at a near-stoichiometric equivalence ratio and addition of a percentage of EGR to the contents of the cylinder are discussed. The uncontrolled engine equivalence ratio was 0.97. The controller pushed the equivalence ratio to 1.0, due to the behavior of the control input $u(k)$ additional mass of fuel injected. The EGR used for this experiment was nitrogen, rather than actual exhaust gas. The nominal mass of EGR is set such that its mass is a desired percentage of the total mass of cylinder contents. The following equation shows that a mass of nitrogen, m_{EGR}, can be chosen to give a desired percentage of EGR.

$$\%EGR = 100 \times \left(\frac{m_{EGR}}{m_f + m_a + m_{EGR}} \right) \quad (94)$$

Heat release time series and return maps were generated for both controlled and uncontrolled cases for each of two EGR set points: 5%, and 10%. Before engine tests, air flow is measured and nominal fuel is calculated for the desired equivalence ratio. The nominal fuel and air are loaded into the controller configuration. During data acquisition, ambient pressure is referenced in the acquired cylinder pressure each engine cycle based on the in-cylinder pressure when the exhaust valve is fully open at 600°.

The NN weight values are all initialized at zero. Figs. 3 and 4 display a decrease in cyclic dispersion for 10% EGR respectively During the absence of control there is much cyclic dispersion and occasional misfires, illustrated as spikes, and during control the misfires and dispersion are reduced. The return maps at 10% EGR show distinct cyclic dispersion during no control and a significant decrease in those dispersed data points during control.

When using fuel as the control input, the controller must change the fuel to affect the engine, and therefore changes the equivalence ratio. Fuel intake increases slightly during control causing the actual operating equivalence ratio to be slightly higher than the set point, here, at 1.0. It is thought that this is partly due to a higher value specified for target heat release compared to uncontrolled case. Moreover, this slight offset remains due to slow learning of the NNs which eventually becomes zero with time. A tradeoff exists between speed of learning and performance. Higher learning rate for NNs slightly degrades performance in terms of dispersion and vice versa.

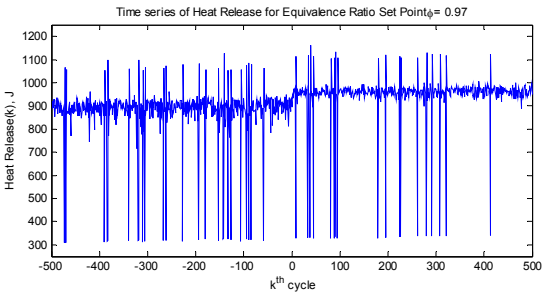

Fig. 3. Heat release time series at 10% EGR

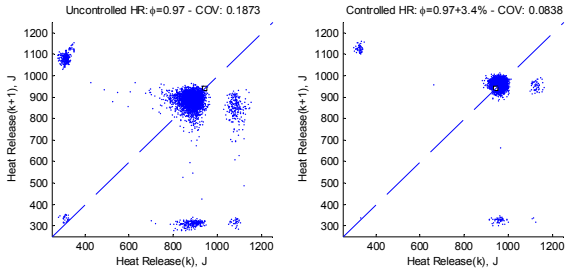

Fig. 4. Uncontrolled and controlled heat release return maps plotting current cycle y(k) against next cycle y(k+1) at 10% EGR.

The coefficient of variation, COV, for all of the EGR return maps is listed in Table 1. As the EGR percentage of cylinder contents is increased from 0% to 10%, the coefficient of variation increases for both uncontrolled and controlled engine operation. The increased coefficient of variation indicates increased cyclic dispersion as seen in the EGR return maps. The coefficient of variation decreases when control is applied.

Table 1. Coefficient of Variation of the Return Maps

EGR	Uncontrolled COV	Controlled COV
5%	0.0873	0.0347
10%	0.1873	0.0838

Experimental results indicate that 80% drop of NO_x at stoichiometric levels (2153 PPM C_3H_8 at 0% EGR to 436 PPM C_3H_8) using 10% EGR. The percentage of NO_x reduction should roughly remain the same between the controlled and uncontrolled cases as it is a strong function of the percentage of EGR. The percentage of unburned hydrocarbons at 10% also shows a drop of 28% due to control (58 PPM C_3H_8) as compared to the uncontrolled scenario (81 PPM C_3H_8).

VIII. Conclusions

A novel NN controller scheme is presented to reduce the cyclic dispersion in heat release at high EGR levels. The stability analysis of the closed-loop control system was conducted and the boundedness of the closed loop signals was shown. Experimental results show that the performance of the proposed controller is highly satisfactory while meeting the closed loop stability even though the dynamics are not known beforehand. The presented work can be extended by introducing a separate control loop for the EGR while varying the air-fuel ratio to include lean operation.

References

[1] K. P. Dudek and M. K. Sain, "A control-oriented model for cylinder pressure in internal combustion engines," *IEEE Trans. on automatic control*, vol. 34(4), 1989, pp. 386-397.

[2] R. W. Sutton and J. A. Drallmeier, 2000, "Development of nonlinear cyclic dispersion in spark ignition engines under the influence of high levels of EGR", in *Proc. of the Central States Section of the Combustion Institute*, Indianapolis, Indiana, April 16-18, 2000, pp. 175-180.

[3] P. He and S. Jagannathan, "Neuroemission controller for reducing cyclic dispersion in lean combustion spark ignition engines," in *Automatica*, vol. 41, April 2005, pp. 1133-1142.

[4] C. S. Daw, C. E. A. Finney, M. B. Kennel and F. T. Connolly, "Observing and modeling nonlinear dynamics in an internal combustion engine," in *Physical Review. E*, vol. 57(3), pp. 2811 – 2819, 1998.

[5] C. S. Daw, C. E. A. Finney, J. B. Green, M. B. Kennel and J. F. Thomas, "A simple model for cyclic variations in a spark-ignition engine," *SAE,* 962086, May 1996.

[6] J.B. Heywood, *Internal combustion engine fundamentals*, MGraw-Hill, New York, 1998.

[7] T. Inoue, S. Matsushita, K. Nakanishi, and H. Okano, Toyota lean combustion system-The third generation system. *SAE Technical Paper series*, 930873, 1993.

[8] R. M. Wagner, "Identification and characterization of complex dynamic structure in spark ignition engines," Ph.D. dissertation, Univ. Missouri – Rolla, Dept. Mech. Eng. Rolla, MO, 1998.

[9] R. M. Wagner, J. A. Drallmeier, & C. S. Daw, "Nonlinear cycle dynamics in lean spark ignition combustion," presented at the 27th Symposium (International) of Combustion, 1999.

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

[11] 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, May 1995, pp. 791–801.

[12] 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, 2003, pp. 334–354.

[13] 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, Nov.1995, pp. 1320-1329.

[14] S. Jagannathan, "Robust backstepping control of robotic systems using neural networks," in *Proc. 37th IEEE Conf. on Decision and Control*, 1998.

[15] H. K. Khalil, *Nonlinear Systems*, 3rd ed., Prentice Hall, 2002.

[16] P. He, Z. Chen and S. Jagannathan, "Reinforcement learning based neural network control of nonstrict feedback nonlinear systems", *Proc. of IEEE Conference on Decision and Control*, Dec 2005.

[17] F. L. Lewis, S. Jagannathan, and A. Yesilderek, *Neural Network Control of Robot Manipulator and Nonlinear Systems*, Taylor & Francis Inc., UK, 1999.

[18] F. L. Lewis, J. Campos, and R. Selmic, *Neuro-Fuzzy Control of Industrial Systems with Actuator Nonlinearities,* Society for Industrial and Applied Mathematics, Philadelphia, 2002.