01 Jan 2007

Reinforcement Learning Based Output-feedback Controller for Complex Nonlinear Discrete-Time Systems

Peter Shih

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

Recommended Citation

P. Shih and J. Sarangapani, "Reinforcement Learning Based Output-feedback Controller for Complex Nonlinear Discrete-Time Systems," *Proceedings of the IEEE 22nd International Symposium on Intelligent Control (2007, Singapore)*, Institute of Electrical and Electronics Engineers (IEEE), Jan 2007.
The definitive version is available at https://doi.org/10.1109/ISIC.2007.4450920

22nd IEEE International Symposium on Intelligent Control
Part of IEEE Multi-conference on Systems and Control
Singapore, 1-3 October 2007

Reinforcement Learning based Output-Feedback Controller for Complex Nonlinear Discrete-time Systems

Peter Shih and S. Jagannathan

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 feedback nonlinear discrete-time systems in the presence of bounded and unknown disturbances. This nonlinear discrete-time system consists of a second order system in nonstrict form and an affine nonlinear discrete-time system tightly coupled together. Two adaptive critic NN controllers are designed—primary one for the nonstrict system and the secondary one for the affine system. A Lyapunov function shows the uniformly ultimate boundedness (UUB) of the closed-loop tracking error, weight estimates and observer estimates. Separation principle and certainty equivalence principles are relaxed, persistency of excitation condition is not required 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 where the objective is to reduce cyclic dispersion in heat release.**

I. Introduction

Adaptive NN backstepping control of nonlinear discrete-time systems in strict feedback form has been addressed in the literature [1-3]. The system is normally expressed as

$$x_i(k+1)=f_i(\bar{x}_i(k))+g_i(\bar{x}_i(k))x_{i+1}(k) \tag{1}$$

$$x_n(k+1)=f_n(\bar{x}_n(k))+g_n(\bar{x}_n(k))u(k) \tag{2}$$

where $x_i(k)\in\Re$ is the state, $u(k)\in\Re$ is the control input, $\bar{x}_i(k)=[x_1(k),\cdots,x_i(k)]^T\in\Re^i$ and $i=1,\dots,(n-1)$. For strict feedback nonlinear systems [1], the nonlinearities $f_i(\bar{x}_i(k))$ and $g_i(\bar{x}_i(k))$ depend only upon states $x_1(k),\dots,x_i(k)$, i.e., $\bar{x}_i(k)$. However, for non-strict feedback nonlinear system, $f_i(\bar{x}_i(k))$ and $g_i(\bar{x}_i(k))$ depend on both $\bar{x}_i(k)$ and $x_{i+1}(k)$, and there are no control design schemes currently available. Available [1-3] methods results in a non-causal controller (current control input depends on the future system states). Finally, no optimization is carried out in these control designs.

Available NN controller designs employ either supervised training 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. In supervised learning, a teacher produces a signal to guide the learning process whereas reinforcement learning, the role of the teacher is more evaluative than instructional. The critic allows for near optimal control.

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. However, NN controller results are not available for the nonlinear discrete-time systems in non-strict feedback form. Similarly, no known results are available using adaptive critic NN control-based affine nonlinear discrete-time systems. In this paper, a novel adaptive critic NN-based output feedback controller is developed to control a complex nonlinear discrete-time systems consisting of non-strict feedback form and an affine systems tightly coupled together.

For the case of nonlinear discrete-time system in nonstrict feedback form, 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 standard Bellman equation. The NN observer estimates the system states and output, which are subsequently used in the controller design. The proposed controller is *model–free* since the dynamics of the nonlinear discrete-time systems are unknown and NN weights are tuned online. For the affine nonlinear discrete-time system, a separate critic NN and an action NN are utilized and states are assumed to be available for measurement. The critic NN approximates the Bellmann equation and tunes the action NN to generate near optimal signal.

The proposed primary controller is applied to control the spark ignition (SI) engine dynamics. The controller allows the engine to operate in high EGR mode, where an inert gas displaces the stoichiometric ratio of fuel to air. The inert gas system, controlled by the secondary controller, maintains a set EGR level. Both controllers allow the engine to operate in higher EGR mode reducing heat release dispersion thereby improving engine emissions and fuel efficiency.

II. Non-linear non-strict feedback system

Consider the nonlinear discrete-time system, given in the following form

$$x_1(k+1)=f_1(\bar{x}_i(k))+g_1(\bar{x}_i(k))x_2(k)+d_1(k) \tag{3}$$

$$x_2(k+1)=f_2(\bar{x}_i(k))+g_2(\bar{x}_i(k))u(k)+d_2(k) \tag{4}$$

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 at the University of Missouri-Rolla (contact author's e-mail: sarangap@umr.edu).

1-4244-0441-X/07/$20.00 ©2007 IEEE.

$$x_3(k+1) = f_4(\bar{x}_i(k)) + g_4(\bar{x}_i(k))v(k) + d_3(k) \tag{5}$$

$$y(k+1) = f_3(\bar{x}_i(k)) \tag{6}$$

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. Bounds on the disturbances are given by $|d_1(k)| < d_{1m}$, $|d_2(k)| < d_{2m}$, and $|d_3(k)| < d_{3m}$ where d_{1m}, d_{2m}, and d_{3m} are unknown positive scalars. The output is a nonlinear function of states. Finally, the output and third state, $x_3(k)$, are measurable whereas the first two states $x_1(k)$ and $x_2(k)$ are not. For the system (3) and (4), not only the system actual output should converge to its target value but also the states should converge to their respective desired values.

The controller development is presented separately for the systems as the objectives are separate even though they are tightly coupled. The first part uses equations (3), (4), and (6) to develop the primary controller. The second part uses equation (5) to develop the secondary controller. Note that equations (3) through (6) are arbitrary unknown functions.

III. Primary Controller – Observer Design

To overcome the immeasurable states $x_1(k)$ and $x_2(k)$, 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 (3) and (4). We expand the individual nonlinear functions using Taylor series.

$$f_1(\cdot) = f_{10} + \Delta f_1(\cdot) \tag{7}$$

$$f_2(\cdot) = f_{20} + \Delta f_2(\cdot) \tag{8}$$

$$g_1(\cdot) = g_{10} + \Delta g_1(\cdot) \tag{9}$$

$$g_2(\cdot) = g_{20} + \Delta g_2(\cdot) \tag{10}$$

where the first term in (7) through (10) 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

$$y(k+1) = w_1^T \phi(v_1^T z_1(k)) + \varepsilon(z_1(k)), \tag{11}$$

where $z_1(k) = [x_1(k), x_2(k), x_3(k), y(k), u(k)]^T \in R^4$ 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, $\phi(v_1^T z_1(k))$ represents the hidden layer activation function, n_1 is the number of nodes in the hidden layer, and $\varepsilon(z_1(k)) \in \Re$ is the approximation error. For simplicity the two equations can be represented as

$$\phi_1(k) = \phi(v_1^T z_1(k)) \tag{12}$$

$$\varepsilon_1(k) = \varepsilon(z_1(k)) \tag{13}$$

Rewrite (11) using (12) and (13) to obtain

$$y(k+1) = w_1^T \phi_1(k) + \varepsilon_1(k) \tag{14}$$

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

$$\begin{aligned} \hat{y}(k+1) &= \hat{w}_1^T(k)\phi(v_1^T \hat{z}_1(k)) + l_1 \tilde{y}(k) \\ &= \hat{w}_1^T(k)\hat{\phi}_1(k) + l_1 \tilde{y}(k) \end{aligned} \tag{15}$$

where $\hat{z}_1(k) = [\hat{x}_1(k), \hat{x}_2(k), x_3(k), \hat{y}(k), u(k)]^T \in R^5$ 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 $\tilde{y}(k)$ is the heat release estimation error defined as

$$\tilde{y}(k) = \hat{y}(k) - y(k) \tag{16}$$

It is demonstrated in [9] 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 $\varepsilon(z_1(k))$ can be made arbitrarily small so that the bound $\|\varepsilon(z_1(k))\| \le \varepsilon_{1m}$ 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

$$\hat{x}_1(k+1) = f_{10} - \hat{x}_2(k) + l_2 \tilde{y}(k) \tag{17}$$

$$\hat{x}_2(k+1) = f_{20} + g_{20} u(k) + l_3 \tilde{y}(k) \tag{18}$$

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

B. Observer Error Dynamics

Define the state estimation and output errors as

$$\tilde{x}_i(k+1) = \hat{x}_i(k+1) - x_i(k+1), i \in \{1,2\} \tag{19}$$

$$\tilde{y}(k+1) = \hat{y}(k+1) - y(k+1) \tag{20}$$

Combining (3) through (11) and, (17) through (20), to obtain the estimation and output error dynamics as

$$\tilde{x}_1(k+1) = f_{10} - \hat{x}_2(k) + l_2 \tilde{y}(k) - f_1(\cdot) - g_1(\cdot) x_2(k) - d_1(k) \tag{21}$$

$$\tilde{x}_2(k+1) = f_{20} + g_{20} u(k) + l_3 \tilde{y}(k) - f_2(\cdot) - g_2(\cdot) u(k) - d_2(k) \tag{22}$$

$$\tilde{y}(k+1) = \hat{w}_1^T(k)\hat{\phi}_1(k) + l_1 \tilde{y}(k) - w_1^T \phi_1(k) - \varepsilon_1(k) \tag{23}$$

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 was demonstrated that in the next section that by using the above weight tuning, separation principle is relaxed and the closed-loop signals will be bounded. Next we present the following theorem where it is demonstrated that the state estimation and output estimation errors along with observer NN weight estimation errors are bounded. The following mild assumptions are required.

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

Next we design the adaptive critic NN controller for the primary system where it will be demonstrated that the closed-loop system including the NN observer signals will be bounded then the control inputs will be bounded.

IV. Primary Controller – 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 tuning will ultimately minimize the strategic utility function itself and action NN outputs or control inputs 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. 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

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

where $\beta\in\Re$ and $0<\beta<1$ is the discount factor and N is the horizon. The term $Q(k)$ is viewed here as the long system performance measure for the controller since it is the sum of all future system performance indices. Equation (26) can also be expressed as $Q(k)=\min\limits_{u(k)}\left\{\alpha Q(k-1)-\alpha^{N+1}p(k)\right\}$, which is similar to the standard Bellman equation.

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

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

where $\hat{Q}(k)\in\Re$ is the critic signal, $\hat{w}_2(k)\in\Re^{n_2}$ is the tunable weight, $v_2\in\Re^{3\times n_2}$ represent the constant input weight matrix selected initially 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

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

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

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

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

$$\hat{w}_2(k+1)=\hat{w}_2(k)+\alpha_2\left[-\frac{\partial E_c(k)}{\partial\hat{w}_2(k)}\right] \tag{30}$$

$$\hat{w}_2(k+1)=\hat{w}_2(k)-\alpha_2\hat{\phi}_2(k)\left(\hat{Q}(k)+\beta^{N+1}p(k)-\beta\hat{Q}(k-1)\right)^T \tag{31}$$

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

V. Primary Controller – Virtual Control Input NN

In this section, the design of the virtual control input is discussed. Before we proceed, 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, $0<g_{2\min}<\left|g_2(\cdot)\right|<g_{2\max}$, , $\forall x_1(k)\,\&\,x_2(k)\in S$ 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

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

$$\Phi(\cdot)=f_1\left(\bar{x}_i(k)\right)+g_1\left(\bar{x}_i(k)\right)x_2(k)+x_2(k) \tag{32}$$

The system (3) and (4) can be rewritten as

$$x_1(k+1)=\Phi(\cdot)-x_2(k)+d_1(k) \tag{33}$$

$$x_2(k+1)=f_2(\cdot)+g_2(\cdot)u(k)+d_2(k) \tag{34}$$

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. The secondary objective is to make $x_1(k)$ approach the desired trajectory $x_{1d}(k)$. At the same time, all signals in systems (3) and (4) must be UUB; all weights must be bounded; and a performance index must be minimized. Define the tracking error as

$$e_1(k)=x_1(k)-x_{1d}(k) \tag{35}$$

where $x_{1d}(k)$ is the desired trajectory. Using (33), (35) can be expressed as the following

$$\begin{aligned}e_1(k+1)&=x_1(k+1)-x_{1d}(k+1)\\&=\left(\Phi(\cdot)-x_2(k)+d_1(k)\right)-x_{1d}(k+1)\end{aligned} \tag{36}$$

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

$$x_{2d}(k)=\Phi(\cdot)-x_{1d}(k+1)+l_5\hat{e}_1(k) \tag{37}$$

where l_5 is a gain constant. Since $\Phi(\cdot)$ is an unknown function, $x_{2d}(k)$ in (37) cannot be implemented in practice. We invoke the universal approximation property of NN to estimate this unknown function.

$$\Phi(\cdot)=w_3^T\phi\left(v_3^Tz_3(k)\right)+\varepsilon\left(z_3(k)\right) \tag{38}$$

where $z_3(k)=\left[x_1(k),x_2(k),x_3(k)\right]^T\in\Re^3$ is the input vector, $w_3^T\in\Re^{n_2}$ and $v_3^T\in\Re^{3\times n_3}$ are the ideal and constant input weight matrices, $\phi\left(v_3^Tz_3(k)\right)\in\Re^{n_3}$ 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 functional estimation error.f

Rewrite (37) using (38), the virtual control signal can be rewritten as

$$x_{2d}\left(k\right)=w_3^T\phi\left(v_3^T z_3\left(k\right)\right)+\varepsilon\left(z_3\left(k\right)\right)-x_{1d}\left(k+1\right)+l_5\hat{e}_1\left(k\right) \tag{39}$$

Replacing actual with estimated states, (39) becomes

$$\begin{aligned}\hat{x}_{2d}\left(k\right)&=\hat{w}_3^T\left(k\right)\phi\left(v_3^T\hat{z}_3\left(k\right)\right)-x_{1d}\left(k+1\right)+l_5\hat{e}_1\left(k\right)\\&=\hat{w}_3^T\left(k\right)\hat{\phi}_3\left(k\right)-x_{1d}\left(k+1\right)+l_5\hat{e}_1\left(k\right)\end{aligned} \tag{40}$$

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

$$e_2\left(k\right)=x_2\left(k\right)-\hat{x}_{2d}\left(k\right) \tag{41}$$

Equation (36) can be rewritten using (41) as

$$\begin{aligned}e_1\left(k+1\right)&=\left(\Phi\left(\cdot\right)-x_2\left(k\right)+d_1\left(k\right)\right)-x_{1d}\left(k+1\right)\\&=\Phi\left(\cdot\right)-\hat{x}_{2d}\left(k\right)-e_2\left(k\right)-x_{1d}\left(k+1\right)+d_1\left(k\right)\end{aligned} \tag{42}$$

Combine (40) , (42), then (38)

$$e_1\left(k+1\right)=-\zeta_3\left(k\right)-w_3^T\tilde{\phi}_3\left(k\right)+\varepsilon_3\left(k\right)-l_5\hat{e}_1\left(k\right)-e_2\left(k\right)+d_1\left(k\right) \tag{43}$$

where

$$\zeta_3\left(k\right)=\tilde{w}_3^T\left(k\right)\hat{\phi}_3\left(k\right)=\hat{w}_3^T\left(k\right)\hat{\phi}_3\left(k\right)-w_3^T\hat{\phi}_3\left(k\right) \tag{44}$$

$$\tilde{\phi}_3\left(k\right)=\phi\left(v_3\hat{z}_3\left(k\right)\right)-\phi\left(v_3 z_3\left(k\right)\right) \tag{45}$$

C. Virtual Control Weight Update

Let us define

$$e_{a1}\left(k\right)=\hat{w}_3^T\left(k\right)\hat{\phi}_3\left(k\right)+\left(\hat{Q}\left(k\right)-Q_d\left(k\right)\right) \tag{46}$$

where $\hat{Q}(k)$ is defined in (27), 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, (46) becomes

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

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

$$E_{a1}(k)=\frac{1}{2}e_{a1}^2(k) \tag{48}$$

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

$$\hat{w}_3\left(k+1\right)=\hat{w}_3\left(k\right)+\alpha_3\left[-\frac{\partial E_{a1}\left(k\right)}{\partial\hat{w}_3\left(k\right)}\right] \tag{49}$$

$$\hat{w}_3\left(k+1\right)=\hat{w}_3\left(k\right)-\alpha_3\hat{\phi}_3\left(k\right)\left(\hat{Q}\left(k\right)+\hat{w}_3^T\left(k\right)\hat{\phi}_3\left(k\right)\right) \tag{50}$$

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

VI. Primary Controller – Control Input Design

Choose the following desired control input

$$u_d\left(k\right)=\frac{1}{g_2\left(k\right)}\left(-f_2\left(k\right)+\hat{x}_{2d}\left(k+1\right)+l_6 e_2\left(k\right)\right)^, \tag{51}$$

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 since it can be a one step predictor. 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. Alternatively, the value can be obtained by employing a filter [10]. 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 a suitable control input. Define the input as

$$z_4\left(k\right)=\left[x_1\left(k\right),x_2\left(k\right),x_3\left(k\right),e_1\left(k\right),l_6 e_2\left(k\right),\hat{x}_{2d}\left(k\right),x_{1d}\left(k+2\right)\right]^T\in\Re^7,$$

then $u_d(k)$ can be approximated as

$$u_d\left(k\right)=w_4^T\phi\left(v_4^T z_4\left(k\right)\right)+\varepsilon\left(z_4\left(k\right)\right)=w_4^T\phi_4\left(k\right)+\varepsilon_4\left(k\right), \tag{52}$$

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

$$\hat{u}\left(k\right)=\hat{w}_4^T\left(k\right)\phi\left(v_4^T\hat{z}_4\left(k\right)\right)=\hat{w}_4^T\left(k\right)\hat{\phi}_4\left(k\right), \tag{53}$$

where

$\hat{z}_4\left(k\right)=\left[\hat{x}_1\left(k\right),\hat{x}_2\left(k\right),x_3\left(k\right),\hat{e}_1\left(k\right),l_6\hat{e}_2\left(k\right),\hat{x}_{2d}\left(k\right),x_{1d}\left(k+2\right)\right]^T\in\Re^7$ is the input vector. Rewriting (41) and substituting (51) through (53), to get

$$\begin{aligned}e_2\left(k+1\right)&=x_2\left(k+1\right)-\hat{x}_{2d}\left(k+1\right)\\&=l_6 e_2\left(k\right)-g_2\left(\cdot\right)\varepsilon_4\left(k\right)+g_2\left(\cdot\right)\zeta_4\left(k\right)+g_2\left(\cdot\right)w_4^T\tilde{\phi}_4\left(k\right)+d_2\left(k\right)\end{aligned} \tag{54}$$

where

$$\zeta_4\left(k\right)=\tilde{w}_4^T\left(k\right)\hat{\phi}_4\left(k\right)=\hat{w}_4^T\left(k\right)\hat{\phi}_4\left(k\right)-w_4^T\hat{\phi}_4\left(k\right), \tag{55}$$

$$\tilde{\phi}_4\left(k\right)=\hat{\phi}_4\left(k\right)-\phi_4\left(k\right) \tag{56}$$

Equations (43) and (54) represent the closed-loop error dynamics. Next we derive the weight update law. Define

$$e_{a2}\left(k\right)=\hat{w}_4^T\left(k\right)\hat{\phi}_4\left(k\right)+\hat{Q}\left(k\right), \tag{57}$$

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

$$E_{a2}\left(k\right)=\frac{1}{2}e_{a2}^2\left(k\right) \tag{58}$$

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

$$\hat{w}_4\left(k+1\right)=\hat{w}_4\left(k\right)+\alpha_4\left[-\frac{\partial E_{a2}\left(k\right)}{\partial\hat{w}_4\left(k\right)}\right] \tag{59}$$

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

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 observer, critic and two action NNs and assume that they are bounded above so that

$$\left\|w_1\right\|\le w_{1m},\ \left\|w_2\right\|\le w_{2m},\left\|w_3\right\|\le w_{3m},\text{ and }\left\|w_4\right\|\le w_{4m} \tag{61}$$

where $w_{om}\in R^+$, $w_{1m}\in R^+$ and $w_{2m}\in R^+$ represent the bounds on the unknown target weights where the Frobenius norm [10] is used.

Theorem 1: Consider the system given by (3) and (4), 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), (31), (50), and (60), respectively. Let the virtual control input and control input be given by (40), and (53), 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 controller design parameters selected as

$$0<\alpha_i\left\|\phi_i(k)\right\|^2<1, i\in\{1,2,3,4\} \tag{62}$$

$$|l_1|<\frac{1}{2};|l_2|<\frac{\sqrt{3}}{3};|l_3|<\frac{\sqrt{3}}{3};|l_4|<\frac{\sqrt{3}}{3};|l_5|<\frac{1}{\sqrt{5}};|l_6|<\frac{\sqrt{3}}{3} \tag{63}$$

$$0<\beta<\frac{\sqrt{2}}{2} \tag{64}$$

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. ■

VII. Secondary Controller – Critic Design

For maintaining dilution to a desired level, the third equation will be employed with EGR(k) as the control input and inert gas as an additional state. To simplify the controller development and since the residual gas fraction is upper bounded, this third equation can be simplified as

$$x_3(k+1)=f_4\left(x(k)\right)+g_4\left(x(k)\right)v(k)+d_3(k) \tag{65}$$

where $x(k)=[x_1(k),x_2(k),x_3(k)]^T$. Define

$$x(k)=[x_1(k),x_2(k),x_3(k)]^T \tag{66}$$

A. Design of the Critic

Let the long-term cost function be defined as

$$J(k)=\sum_{i=t_0}^{\infty}\gamma^i r(k+i) \tag{67}$$

where

$$r(k)=\left(x(k)-x_d(k)\right)^T Q\left(x(k)-x_d(k)\right)+v^T(k)Rv(k) \tag{68}$$

where R and Q are positive definite matrices and γ is the discount factor within the range of $0\le\gamma\le1$. Invoke the universal approximation property of NN to estimate (67) as

$$J(k)=w_c^T\phi_c\left(v_c^T z_c(k)\right)+\varepsilon\left(z_c(k)\right) \tag{69}$$

where $\varepsilon\left(z_c(k)\right)$ is the estimation error. Replace the states with estimated states.

$$\hat{J}(k)=\hat{w}_c^T(k)\phi_c\left(v_c^T\hat{z}_c(k)\right)=\hat{w}_c^T(k)\phi_c(k) \tag{70}$$

where $\hat{w}_c\in\Re^{n_c}$ and $v_c\in\Re^{2\times n_c}$ denote the ideal output and constant hidden layer weights, $\phi_c(k)\in\Re^{n_c}$ is the activation function vector, n_c is the number of hidden layer nodes. Again, we hold the input weights constant and adapt the output weights. $\hat{z}_c(k)=\left[\hat{x}_1(k),\hat{x}_2(k),x_3(k)\right]^T\in\Re^3$ is the input.

B. Critic Weight Update Law

Define the prediction error as

$$\begin{aligned} e_c(k)&=\gamma\hat{J}(k)-[\hat{J}(k-1)-r(k)] \\ &=\gamma\zeta_c(k)+\gamma J(k)-\zeta_c(k-1)-J(k-1)+r(k)-\varepsilon_c(k)+\varepsilon_c(k-1) \end{aligned} \tag{71}$$

where

$$\zeta_c(k)=\tilde{w}_c^T(k)\hat{\phi}_c(k)=\hat{w}_c^T(k)\hat{\phi}_c(k)-w_c^T\hat{\phi}_c(k) \tag{72}$$

Use a quadratic minimizing function and gradient-based adaptation method, the weight update is given by

$$\begin{aligned} \hat{w}_c(k+1)&=\hat{w}_c(k)+\alpha_c\left[-\frac{\partial E_c(k)}{\partial\hat{w}_c(k)}\right] \\ &=\hat{w}_c(k)-\alpha_c\gamma\phi_c(k)\left(\gamma\hat{J}(k)+r(k)-\hat{J}(k-1)\right) \end{aligned} \tag{73}$$

VIII. Secondary Controller – Control Input Design

A. Design of the Control input

The tracking error is defined as

$$\begin{aligned} e_4(k)&=x_3(k)-x_{3d}(k) \\ e_4(k+1)&=f_4(\cdot)+g_4(\cdot)v(k)+d_3(k)-x_d(k+1) \end{aligned} \tag{74}$$

where $x_{3d}(k)$ is the target bounded trajectory. Define the desired control signal as

$$v_d(k)=g_4^{-1}(\cdot)\left(-f_4(\cdot)+x_{3d}(k+1)+l_7e_4(k)\right) \tag{75}$$

Using the universal approximation property of NN and the approximate states

$$\hat{v}_d(k)=\hat{w}_a^T(k)\phi_a\left(v_a^T\hat{z}_a(k)\right)=\hat{w}_a^T(k)\hat{\phi}_a(k) \tag{76}$$

where $\hat{w}_a\in\Re^{n_a}$ and $v_a\in\Re^{2\times n_a}$ denote the ideal output and constant hidden layer weight matrices, $\phi_a(k)\in\Re^{n_a}$ is the activation function vector, n_a is the number of hidden layer nodes and $\hat{z}_a(k)=\left[\hat{x}_1(k),\hat{x}_2(k),x_3(k)\right]^T\in\Re^3$ is the input vector. Again, we hold the input weights constant and adapt the output weights only. Rewrite (74) as

$$\begin{aligned} e_4(k+1)&=l_7e_4(k)+g(\cdot)\left(v(k)-v_d(k)\right)+d_3(k) \\ &=l_7e_4(k)+g(\cdot)\zeta_a(k)+d_a(k) \end{aligned} \tag{77}$$

where

$$d_a(k)=-g(\cdot)\varepsilon_a(k)+d_a(k) \tag{78}$$

$$\zeta_a(k)=\tilde{w}_a^T(k)\hat{\phi}_a(k) \tag{79}$$

B. Control Input Weight Update Law

Define the control input cost function

$$\begin{aligned} e_a(k)&=\sqrt{g_4(\cdot)}\zeta_a(k)+\left(\sqrt{g_4(\cdot)}\right)^{-1}\left(J(k)-J_d(k)\right) \\ &=\sqrt{g_4(\cdot)}\zeta_a(k)+\left(\sqrt{g_4(\cdot)}\right)^{-1}J(k) \end{aligned} \tag{80}$$

where $J_d(k)$ is the desired long-term cost function and is equal to zero. Define a quadratic error to minimize and utilize a gradient decent minimization strategy

$$\begin{aligned} \hat{w}_a(k+1)&=\hat{w}_a(k)+\alpha_a\left[-\tfrac{\partial E_a(k)}{\partial\hat{w}_a(k)}\right] \\ &=\hat{w}_a(k)-\alpha_a\gamma\phi_a(k)\left(e_4(k+1)-l_7e_4(k)-d_a(k)+J(k)\right)^T \end{aligned} \tag{81}$$

The proposed controller structure is shown in Figure 1.

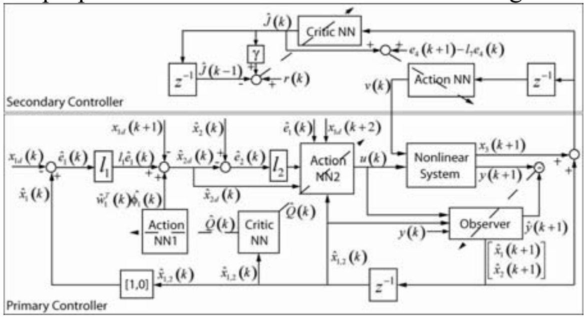

Figure 1 Combined primary and secondary controller structure.

Theorem 2: Consider the system given by (5), and the disturbance bound d_{3m} be known constants. Let the observer and control input NN weight tuning be given by (73) and (81), respectively. Let the control input given by (76), the tracking error $e_4(k)$ and weight estimates $\hat{w}_a(k)$ and $\hat{w}_c(k)$ are UUB, with the controller design parameters selected as:

$$0<\alpha_i\left\|\phi_i(k)\right\|^2<1, i\in\{\alpha,c\} \tag{82}$$

$$\left|l_7\right|<\frac{1}{2} \tag{83}$$

where α_a and α_c are NN adaptation gains, and l_7 is the controller gain. ■

IX. Results and Analysis

Spark ignition (SI) engine dynamics can be expressed according to the Daw model as a class of nonlinear systems in nonstrict feedback form [11]. At high EGR levels, the engine can be expressed as complex discrete-time system [12].

The controller is simulated in conjunction with the Daw model. The learning rates for the observer, critic, virtual control input, and control input networks are 0.01, 0.01, 0.01, and 0.01, respectively. 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 CEmax, 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 for all EGR levels. All NNs use 20 hidden neurons with hyperbolic tangent sigmoid activation functions in the hidden layer. The computation power used is minimal because the input weight matrices are constant; only the output layer matrices are turned using gradient descent method.

Figure 2 shows two heat release return maps, one controlled and the other uncontrolled, for the set point at 13% EGR. Each subfigure shows the next time step versus the current time step heat release. Note the clustering of the points around the target heat release of 850J denoted by a square. There are no complete misfires, but the heat release variation can be clearly seen. Figure 3 shows the time series of the heat release and control input. The controller converges fast and to a stable operation point after several cycles. The spikes in control output indicate a decline in heat release such as misfire, translating into additional fuel control to counteract. The heat release dispersion during control is improved compared to the uncontrolled case as shown in Table 1 using the coefficient of variation (COV) metric (negative sign shows a drop). The performance outperformed the slight increase in the mean fuel input.

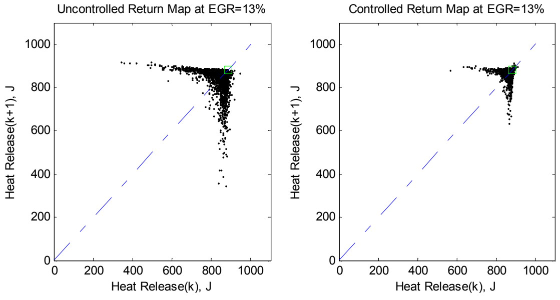

Figure 2 Uncontrolled and controlled heat release return map at 13% EGR.

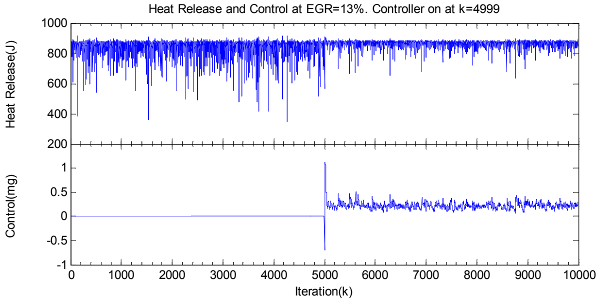

Figure 3 Heat release vs iteration number. Controller turns on at k=5000.

Table 1 Coefficient of variation (COV) and fuel data

	Covariance (COV)		%COV	%Fuel
EGR	Uncontrolled	Controlled	Change	Change
0.00	0.0058	0.0057	-0.75	0.00
0.13	0.0548	0.0384	-29.94	0.40
0.15	0.1387	0.0773	-44.30	0.71
0.19	0.3421	0.2383	-30.34	0.42

X. Conclusions

The controller presented successfully controlled a SI engine to reduce cyclic dispersion under high EGR condition. The system is modeled a complex 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. Simulation shows the stability of the controller under a variety of set points. The output is stable, as predicted by the Lyapunov proof.

References

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

[2] S. S. Ge, T. H. Lee, G. Y. Li, and J. Zhang, "Adaptive NN control for a class of discrete-time nonlinear systems," *Int. J. Contr.,* vol. 76, pp. 334-354, 2003.

[3] F. C. Chen and H. K. Khalil, "Adaptive control of a class of nonlinear discrete-time systems using neural networks," *IEEE Trans. Automat. Contr,* vol. 40, 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] F. L. Lewis, S. Jagannathan, and A. Yesilderek, *Neural Network control of robot manipulators and nonlinear systems*. UK: Taylor and Francis, 1999.

[9] 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.

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

[11] 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.

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