Fall 2007

Neural network control of nonstrict feedback and nonaffine nonlinear discrete-time systems with application to engine control

Jonathan B. Vance

Department: Electrical and Computer Engineering

NEURAL NETWORK CONTROL OF NONSTRICT FEEDBACK AND NONAFFINE NONLINEAR DISCRETE-TIME SYSTEMS WITH APPLICATION TO ENGINE CONTROL

by

JONATHAN BLAKE VANCE

A DISSERTATION

Presented to the Faculty of the Graduate School of the

UNIVERSITY OF MISSOURI-ROLLA

In Partial Fulfillment of the Requirements for the Degree

DOCTOR OF PHILOSOPHY

in

ELECTRICAL ENGINEERING

2007

J. Sarangapani, Advisor

S. Smith

K. Erickson

D. Beetner

J. Drallmeier

PUBLICATION DISSERTATION OPTION

This dissertation consists of the following five articles that have been submitted for publication as follows:

Paper 1, “Discrete-Time Neural Network Output Feedback Control of Nonlinear Discrete-time Systems in Non-Strict Form,” will appear in Automatica 2007.

Paper 2, “Output Feedback Controller for Operation of Spark Ignition Engines at Lean Conditions Using Neural Networks,” will appear in IEEE Trans. on Control Systems Technology.

Paper 3, “Neuro Emission Controller for Minimizing Cyclic Dispersion in Spark Ignition Engines with EGR Levels,” has been submitted to International Journal of General Systems.

Paper 4, “Neural Network Controller Development and Implementation for Spark Ignition Engines with High EGR Levels” has been published in IEEE Trans. on Neural Networks, Vol. 18, no. 4, July 2007.

Paper 5, “Reinforcement Learning-based State-Feedback Control of Nonaffine Nonlinear Discrete-time Systems with Application to Engine Spark Timing Control,” submitted to IEEE Trans. on Systems, Man, and Cybernetics.

ABSTRACT

In this dissertation, neural networks (NN) approximate unknown nonlinear functions in the system equations, unknown control inputs, and cost functions for two different classes of nonlinear discrete-time systems. Employing NN in closed-loop feedback systems requires that weight update algorithms be stable. This dissertation is comprised of five refereed journal-quality papers that have been published or are under review. Controllers are developed and applied to a nonlinear, discrete-time system of equations for a spark ignition engine model to reduce the cyclic dispersion of heat release. In some of the papers, the controller is also tested on a different nonlinear system using simulation.

An adaptive neural network-based output feedback controller is proposed to deliver a desired tracking performance for a class of discrete-time nonlinear systems, which are represented in non-strict feedback form. A spark ignition engine can be viewed as a nonstrict-feedback nonlinear discrete-time system. An NN controller employing output feedback is designed to reduce cyclic dispersion of heat release in a spark ignition engine that uses three NNs to estimate the unknown states, generate the virtual control input, and to generate the actual control input. Another NN controller uses state feedback to minimize cyclic dispersion caused by high levels of exhaust gas recirculation (EGR). Adding another state for EGR to the engine model, an adaptive NN controller is designed with a separate control loop for maintaining an EGR level where output feedback of heat release is used. The system becomes nonaffine with spark timing as the control input, and a novel controller based on reinforcement learning is proposed for the affine-like nonlinear error dynamic system.

ACKNOWLEDGMENTS

I would like to thank Dr. Jagannathan Sarangapani for sharing his knowledge of control systems with me so that I may use what I have learned in my career. I admire his very strong work and thank him for his patience in giving me a higher education.

Also, in addition to help they have provided to me along the way, I want to thank Dr. James Drallmeier for sharing some of his vast knowledge of internal combustion engines, Dr. Scott Smith and Dr. Daryl Beetner for the great computer engineering courses, and Dr. Kelvin Erickson for letting me teach the summer control systems course for my first exposure to teaching.

Thank you to the United States Department of Education for providing the GAANN fellowship which funded most of my graduate studies. Additionally, I would also thank National Science Foundation for providing graduate research assistant support beyond the GAANN Fellowship.

Thank you to my parents Jerry and Sylvia for their love and support, and thank you to my brother Jeremy for encouragement and genuine interest in my work.

TABLE OF CONTENTS

Page

SECTION

LIST OF ILLUSTRATIONS

Figure Page

PAPER 1

PAPER 2

PAPER 3

PAPER 4

PAPER 5

LIST OF TABLES

Table Page

PAPER 2

PAPER 3

PAPER 4

PAPER 5

SECTION

1. INTRODUCTION

The behavior of nonlinear systems cannot be described as a linear function of the dependent variables. This makes them difficult to solve, but they are very important to study as most real systems are nonlinear. A nonlinear system does not follow the principle of superposition, eliminating techniques used for linear systems. Nonlinear systems can have many isolated equilibrium points. In some scenarios the nonlinear system may have limit cycles or bifurcations. Also, these systems can exhibit chaos.

Several methods have been studied to control nonlinear systems including adaptive control where parameters can slowly vary over time or are uncertain. There is feedback linearization that applies a change of variables and control input to the nonlinear system to make it an equivalent linear system. Among others there are also Lyapunov techniques such as backstepping and sliding mode control. Lyapunov backstepping technique is used in this dissertation.

Another consideration of nonlinear system control is whether the model is given in continuous time or discrete time. Using Lyapunov techniques to show stability of a nonlinear controller requires proof that the derivative of a continuous-time Lyapunov or the first difference of a discrete-time Lyapunov is negative definite. The mathematical differences between continuous-time and discrete-time lead to very different proofs to demonstrate boundedness or stability.

The control of nonlinear systems presents many challenges to overcome when a system has either unavailable state information or unmodeled dynamics. There is difficulty obtaining a desired control input such that the system will track a desired reference signal. There may also be some difficulty in showing that a system is stable. Furthermore, a nonlinear system with a nonaffine control input presents further design problems, as conventional nonlinear controller techniques may fail.

This dissertation examines such a system in discrete-time that has a nonstrict feedback representation of system states with affine control input. The application of most of the controllers to a discrete-time spark ignition engine model demonstrates their effectiveness at controlling a nonlinear system with unmodeled dynamics and unavailable states for measurement. Neural network-based controllers employ state and output feedback and use fuel and recirculated exhaust gases for control inputs – both of which are affine to the nonstrict feedback nonlinear discrete-time system. Also, spark timing is considered as a control input, which appears as nonaffine in the nonlinear discrete-time system. Furthermore, the controller for the nonaffine system places bounds on the control input and tracks the desired target with the constraints in place.

The first paper in this dissertation, “Discrete-Time Neural Network Output Feedback Control of Nonlinear Discrete-time Systems in Non-Strict Form,” considers a class of discrete-time nonlinear systems that are represented in non-strict feedback form. A neural network-based control scheme is proposed to provide desired tracking performance. An observer neural network is used to estimate the system states and two other neural networks generate virtual and desired control inputs. Performance of the controller was demonstrated successfully by using a general nonlinear discrete-time system in nonstrict feedback form.

The second paper, “Output Feedback Controller for Operation of Spark Ignition Engines at Lean Conditions Using Neural Networks,” deals with lean operation of the spark ignition engine where the equivalence ratio of fuel to air is less than one. Heat release is used as the output feedback signal to an adaptive neural network-based output feedback controller. Since the states of total fuel and total air within the cylinder are not realistically measurable, an observer neural network is designed to estimate the heat release, and air and fuel. Two more neural networks are used to estimate the unknown functions of residual gas fraction and combustion efficiency that are affected by engine dynamics for every cycle and to create the control input in the form of a deviation from the nominal fuel for a desired operating equivalence ratio. By using two engines, significant reductions in heat release is demonstrated along with engine out emissions such as reductions in NOx from stoichiometric levels and drop in unburned hydrocarbons with control.

The third paper, “Neuro Emission Controller for Minimizing Cyclic Dispersion in Spark Ignition Engines with EGR Levels,” where EGR is exhaust gas recirculation, again deals with lean operation of the engine, but employs state feedback of the air and fuel assuming that an accurate measurement from the universal exhaust gas oxygen sensor (UEGO) can be used to determine the total air and fuel in the cylinder since total fresh fuel is known. Performance of the controller is tested using a simplified engine model.

Since the total air and fuel represented as states are not measurable, in the fourth paper, “Neural Network Controller Development and Implementation for Spark Ignition Engines with High EGR Levels,” a controller is designed with EGR control in mind. A neural network-based output feedback controller is developed to reduce cyclic variation in the heat release under 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. Use of EGR to dilute the cylinder charge is preferable since catalytic converters require a stoichiometric fuel-air ratio of one to operate properly and remain functional. The controller was implemented by using two engine systems and significant drop in engine out emissions was demonstrated.

In all the above controller developments, fuel is used as a control input. In the fifth paper, "Reinforcement Learning-based State-Feedback Control of Nonaffine Nonlinear Discrete-time Systems with Application to Engine Spark Timing Control," spark timing is used as the control input for reducing cyclic dispersion. This control input appears in the state equations as nonaffine so a control system must be designed for a nonaffine nonlinear system. The error dynamics, where output feedback is used, are transformed into an affine-like system whereupon controller development becomes less difficult. Input constraints are also designed into the controller, and simulation shows that the system is stable with limits on the control input.

A connection is made with the papers of this dissertation as it moves from one problem to the next involving a nonlinear, discrete-time systems. A controller is designed for the nonstrict feedback class of systems. Knowledge from the controller for the nonstrict feedback class is used to create an adaptive neural network-based output feedback controller for a set of spark ignition engine equations in nonstrict feedback form. Then, a modification of this controller idea leads to a different controller which used state feedback rather than output feedback. An addition to the spark ignition engine model includes a state equation for EGR, and a separate control loop is added to the controller for EGR. Finally, looking into spark timing as another control input leads to a nonaffine, nonlinear, discrete-time representation of the spark ignition engine model because the spark timing appears a nonaffine control input. For the first time, limits are placed on the control input, which force it to stay within a specified region. Focusing on the affine error dynamics of the nonaffine system, a controller is designed to maintain tracking performance while limits to the control input are imposed.

PAPER 1

Discrete-Time Neural Network Output Feedback Control of Nonlinear Discrete-time Systems in Non-Strict Form

J. Vance and S. Jagannathan

Abstract — An adaptive neural network (NN) -based output feedback controller is proposed to deliver a desired tracking performance for a class of discrete-time nonlinear systems, which are represented in non-strict feedback form. The NN backstepping approach is utilized to design the adaptive output feedback controller consisting of: 1) a NN observer to estimate the system states, and 2) two NNs to generate the virtual and actual control inputs, respectively. The non-causal problem encountered during the control design is overcome by using a dynamic NN which is constructed through a feedforward NN with a novel weight tuning law. The separation principle is relaxed, persistency of excitation condition (PE) is not needed and certainty equivalence principle is not used. The uniformly ultimate boundedness (UUB) of the closed-loop tracking error, the state estimation errors and the NN weight estimates is demonstrated. Though the proposed work is applicable for second order nonlinear discrete-time systems expressed in nonstrict feedback form, the proposed controller design can be easily extendable to an nth order nonlinear discrete-time system.

I. Introduction

The adaptive neural network (NN) backstepping control approach is a potential solution to control a large class of nonlinear systems, whose dynamics are unknown, since the NNs, which are nonlinear in the tunable parameters, can approximate the unknown dynamics. By using NNs in each stage of the backstepping procedure to estimate certain nonlinear functions, a more suitable control law can be designed without using both the linear in the parameter (LIP) assumption and regression matrix (Krstic et al. 1995).

Adaptive state feedback control of nonlinear discrete-time systems in strict feedback form has been addressed in the literature, where the nonlinear system is $x_i(k+1) = f_i(\overline{x}_i(k)) + g_i(\overline{x}_i(k))x_{i+1}(k)$, and $x_n(k+1) = f_n(\overline{x}_n(k)) + g_n(\overline{x}_n(k))u(k)$ with $x_i(k) \in R$ is the state, $u(k) \in R$ is the control input, and $\overline{x}_i(k) = [x_1(k), \cdots, x_i(k)]^T \in R^i$ and $i = 1, \ldots, (n-1)$. For the strict-feedback systems (Krstic et al. 1995), the nonlinearities $f_i(\overline{x}_i(k))$ and $g_i(\overline{x}_i(k))$ depend only upon states $x_1(k), \ldots, x_i(k)$, i.e., $\overline{x}_i(k)$. If the states become unavailable for measurement, an observer is used to estimate the states, and then the estimated values will be substituted for the unavailable states in the output feedback controller design.

Several output feedback control schemes in discrete time are developed by using the backstepping design (Yeh and Kokotovic 1995, Chen and Khalil 1995, Alolinwi and Khalil 1997, Atassi and Khalil 2003, Hovakimyan et al. 2002, Kim and Lewis 1999) either for the strict feedback or affine nonlinear systems without (Yeh and Kokotovic 1995, Alolinwi, and Khalil 1997, Atassi and Khalil 2003) and with neural networks

(Chen and Khalil 1995, Alolinwi and Khalil 1997, Atassi and Khalil 2003, Hovakimyan et al. 2002, Kim and Lewis 1999). Moreover many are developed for continuous-time systems.

However, for the non-strict feedback nonlinear discrete-time systems, the previous controller methods will result in a non-causal design (the current control input depends on the future system states) and require a total redesign. Moreover, when NNs are not used, the adaptive output feedback control schemes need an additional linear in the unknown parameter assumption. Several practical systems, for instance the spark ignition engine dynamics operating either with high exhaust gas recirculation (EGR) levels or under lean operation (Davis et al. 1999, Jagannathan 2006), can be represented only in non-strict feedback second order systems. Additionally, the nonlinearities of these practical systems cannot be expressed as linear in the unknown parameters, which necessitates new design techniques for nonstrict feedback nonlinear systems.

The causal nature encountered in Ge et al. (2003) is due to the nth order strict feedback system whereas the causal problem encountered in the proposed work is due to the nonstrict feedback issue even when a 2^{nd} order system is employed. Additionally, controller development and the NN weight updates are different for the two papers. Finally, separation principle is relaxed in the proposed work.

Therefore, an adaptive NN output feedback controller is proposed to deliver a desired tracking performance for a class of second order discrete-time nonlinear systems in non-strict feedback form. The non-causal problem encountered during the controller design is confronted by employing a dynamic NN constructed via a feedforward NN with semi recurrent structure which acts as a one step predictor. The proposed adaptive NN output feedback controller design employs three NNs: 1) a NN observer to estimate certain system states, and 2) two NNs to generate the virtual and real control input, respectively. The proposed method relaxes the linearity in the unknown parameter assumption, separation and certainty equivalence principle and persistency of excitation condition. The uniformly ultimate boundedness (UUB) of the closed-loop tracking error, the state estimation errors and the NN weight estimates is shown.

II. System Description and Observer Design

A. Nonlinear System Description

The discrete-time nonlinear system in non-strict feedback form is expressed as

$$x_1(k+1) = f_1(x_1(k), x_2(k)) + g_1(x_1(k), x_2(k))x_2(k) + d_1^{'}(k), \tag{1}$$

$$x_2(k+1) = f_2(x_1(k), x_2(k)) + g_2(x_1(k), x_2(k))u(k) + d_2^{'}(k), \tag{2}$$

$$y(k) = x_1(k), \tag{3}$$

where $x_1(k) \in R$ and $x_2(k) \in R$ are the states, $u(k) \in R$ is the control input, $y(k) \in R$ is the system output, state $x_2(k)$ is not measurable, $d_1^{'}(k) \in R$ and $d_2^{'}(k) \in R$ are bounded unknown disturbances, whose bounds are given by $\left|d_1^{'}(k)\right| < d_{1m}^{'}$ and $\left|d_2^{'}(k)\right| < d_{2m}^{'}$. Equations (1) and (2) represent a discrete-time nonlinear system in non-strict feedback form, since unknown functions $f_1(\cdot)$ and $g_1(\cdot)$ depend upon both states $x_1(k)$ and $x_2(k)$, unlike the case of strict feedback systems, where $f_1(\cdot)$ and $g_1(\cdot)$ depend upon only the state $x_1(k)$. Fortunately, for this system, we can use a one-step NN predictor.

The control objective is to drive the system state $x_1(k)$ to track the desired trajectory $x_{1d}(k)$. Since $x_2(k)$ is considered unavailable, both $x_1(k)$ and $x_2(k)$ are estimated by the NN observer. Subsequently, the estimated states are used to design the adaptive NN output feedback controller. Throughout this paper, all quantities with "^" represent estimated quantities. In addition, quantities with "~" represent the estimation errors. Subscripts "o" and "c" refer to the observer and the controller quantities respectively.

B. Observer Structure

Considering the system (1) and (2), for simplicity, let us denote $f_i(k)$ for $f_i\left(x_1(k), x_2(k)\right)$, $g_i(k)$ for $g_i(x_1(k), x_2(k)), \forall i = 1, 2$, where $f_i(k)$ and $g_i(k)$ are smooth vector fields, which are considered unknown. The system under consideration can be written as

$$x_1(k+1) = f_1(k) + g_1(k)x_2(k) + d_1^{'}(k), \tag{4}$$

$$x_2(k+1) = f_2(k) + g_2(k)u(k) + d_2^{'}(k). \tag{5}$$

Write system (4) and (5) into the vector form to get

$$x(k+1) = f(k) + d^{'}(k),$$

where

$$x(k) = \begin{bmatrix} x_1(k) \\ x_2(k) \end{bmatrix},\ f(k) = \begin{bmatrix} f_1(k) + g_1(k)x_2(k) \\ f_2(k) + g_2(k)u(k) \end{bmatrix},\ d^{'}(k) = \begin{bmatrix} d_1^{'}(k) \\ d_2^{'}(k) \end{bmatrix}, \tag{7}$$

Since $x(k)$ is unavailable for measurement, an observer can be designed to estimate the states using the past value of the control input. In other words, using (6), one can observe that estimation of $x(k)$ requires $u(k-1)$ and $f\left(k-1\right)$. Since the control input $u(k-1)$ can be made available whereas $f\left(k-1\right)$ is unknown, it is approximated next by using a NN. Alternatively, one can estimate $x(k)$ from its past values and the control input by using a dynamic mapping approximated via a NN.

The term $f\left(k-1\right)$ can be viewed as an unknown smooth function vector, and it can be approximated by a NN (Igelnik and Pao 1995) as

$$\begin{aligned} f(k-1) &= w_o^T\varphi(v_o^T z_o(k-1)) + \varepsilon_o\left(z_o\left(k-1\right)\right) \\ &= w_o^T\varphi(z_o(k-1)) + \varepsilon_o\left(z_o\left(k-1\right)\right) \end{aligned}, \tag{8}$$

where the NN input is taken as $z_o\left(k-1\right) = \left[x_1\left(k-1\right), x_2\left(k-1\right), u\left(k-1\right)\right]^T \in R^3$, the matrix $w_o \in R^{n_o\times 2}$ and $v_o \in R^{3\times n_o}$ represent the target output and hidden layer weights, the hidden layer activation function $\varphi\left(z_o\left(k-1\right)\right)$ represents $\varphi\left(v_o^T z_o\left(k-1\right)\right)$, n_o denotes the number of the nodes in the hidden layer, and $\varepsilon_o\left(z_o\left(k-1\right)\right) \in R^2$ is the functional approximation error. It is demonstrated in (Igelnik and Pao 1995) that, if the hidden layer weight, v_o, is chosen initially at random and held constant and the number of hidden layer nodes is sufficiently large, the approximation error $\varepsilon_o\left(z_o\left(k-1\right)\right)$ can be made arbitrarily small over the compact set $S \subset R^3$ since the activation function forms a basis.

The proposed NN observer for (6) is defined now as

$$\hat{x}(k) = \hat{w}_o^T(k-1)\varphi(v_o^T\hat{z}_o(k-1)) = \hat{w}_o^T(k-1)\varphi(\hat{z}_o(k-1)), \tag{9}$$

where $\hat{x}(k)=[\hat{x}_1(k),\hat{x}_2(k)]^T \in R^2$ is the estimated value of $x(k)$, and $\hat{z}_o(k-1)=[\hat{x}_1(k-1),\hat{x}_2(k-1),u(k-1)]^T \in R^3$ is the input to the NN observer, the matrix $\hat{w}_o(k-1)\in R^{n_o \times 2}$ is the actually output layer weight, the $\varphi(k)$ or $\varphi(\hat{z}_o(k-1))$ represents $\varphi(v_o^T \hat{z}_o(k-1))$ for convenience. In the rest of this paper, the input to the hidden layer weight matrix, v_o, is not updated and therefore is not explicitly indicated in the equations. However, it is not ignored. Here, it is assumed that the initial value of $u(0)$ is bounded. In the next section, it is demonstrated that all the values of $u(k)$ are bounded $\forall k \in R$.

C. *Observer Error Dynamics*

Define the state estimation errors as

$$\tilde{x}_i(k)=\hat{x}_i(k)-x_i(k) \quad i=1, 2\,. \tag{10}$$

The estimation errors can be expressed in a vector form as

$$\tilde{x}(k)=\hat{x}(k)-x(k)\,, \tag{11}$$

where $\tilde{x}(k)\in R^2$. Combining (6), (8), (9) and (11), we obtain the estimation error dynamics as

$$\tilde{x}(k)=\hat{x}(k)-x(k)=\hat{w}_o^T(k-1)\varphi(\hat{z}_o(k-1))-w_o^T\varphi(z_o(k-1))-\varepsilon_o\left(z_o\left(k-1\right)\right)-d\,'(k-1)$$

$$=\xi_o(k-1)+d_o(k-1)\,, \tag{12}$$

where

$$\tilde{w}_o(k-1) = \hat{w}_o(k-1) - w_o\,, \tag{13}$$

$$\xi_o(k-1) = \tilde{w}_o^T(k-1)\varphi(\hat{z}_o(k-1)) = \left(\hat{w}_o(k-1) - w_o\right)^T \varphi(\hat{z}_o(k-1))\,, \tag{14}$$

$$w_o^T\varphi(\tilde{z}_o(k-1)) = w_o^T\left(\varphi(\hat{z}_o(k-1)) - \varphi(z_o(k-1))\right), \tag{15}$$

and

$$d_o(k-1) = w_o^T\varphi(\tilde{z}_o(k-1)) - d'(k-1) - \varepsilon_o\left(z_o(k-1)\right). \tag{16}$$

Remark 1: Though the first state (or the output) is available for measurement and can be used as a NN input, it is not used here since it can be demonstrated that the observer can estimate both the states with a small bounded error by using the measured output. Also, for certain practical nonlinear discrete-time systems in nonstrict feedback form, for instance spark engine dynamics operating lean (Jagannathan 2006) both the states are unavailable for measurement and the measured output is a nonlinear function of states.

III. Adaptive Neural Network Output Feedback Control

In this section, the development of the controller is discussed. First, some mild assumptions are stated.

A. Backstepping Controller Design

Assumption 1: The desired trajectory $x_{1d}(k)$ is bounded and its future values are available.

Assumption 2: The unknown smooth functions, $g_i(k)$, $\forall i = 1, 2$ are bounded away from zero within certain compact set s as $g_{1M} > |g_1(k)| > g_{1m} > 0$ and $g_{2M} > |g_2(k)| > g_{2m} > 0$, respectively.

Next the adaptive NN output feedback control design is discussed. Define the tracking error between actual and desired trajectory as

$$e_1(k) = x_1(k) - x_{1d}(k), \tag{17}$$

where $x_{1d}(k)$ is the desired trajectory. Combining with (4), (17) can be rewritten as

$$e_1(k+1) = x_1(k+1) - x_{1d}(k+1) = f_1(k) + g_1(k)x_2(k) - x_{1d}(k+1) + d_1^{'}(k) \tag{18}$$

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

$$x_{2d}(k) = \frac{1}{g_1(k)}\left(-f_1(k) + x_{1d}(k+1)\right). \tag{19}$$

Since $f_1(k)$ and $g_1(k)$ are unknown smooth vector fields, the desired feedback control $x_{2d}(k)$ cannot be implemented in practice. From (19), it can be seen that the unknown part $(1/g_1(k))(-f_1(k) + x_{1d}(k+1))$ is a smooth function of $x_1(k)$, $x_2(k)$, and $x_{1d}(k+1)$. The term $x_{2d}(k)$ can be approximated by the first NN as (Jagannathan 2006)

$$x_{2d}(k) = w_1^T \phi\left(v_1^T z_1(k)\right) + \varepsilon_1\left(z_1(k)\right) = w_1^T \phi(k) + \varepsilon_1(k), \tag{20}$$

where the NN input is $z_1\left(k\right)=[x^T\left(k\right),x_{1d}(k+1)]^T$, $w_1 \in R^{n_1}$ and $v_1 \in R^{3\times n_1}$ denote the constant ideal output and hidden layer weights, n_1 is the hidden layer nodes number, the hidden layer activation function $\phi\left(v_2^T z_1\left(k\right)\right)$ is simplified as for convenience $\phi\left(k\right)$, and $\varepsilon_1\left(k\right)$ is the approximation error. Since a single NN is used to approximate (19), this will overcome the controller singularity problem encountered when two NNs are employed (Jagannathan 2006).

Since $x_2\left(k\right)$ is unavailable, it has to be replaced with the estimated state as the NN input. Consequently, the virtual control input is taken as

$$\hat{x}_{2d}\left(k\right)=\hat{w}_1^T\left(k\right)\phi\left(v_1^T\hat{z}_1\left(k\right)\right)=\hat{w}_1^T\left(k\right)\phi\left(\hat{z}_1\left(k\right)\right), \tag{21}$$

where $\hat{w}_1\left(k\right)\in R^{n_1}$ is the actual weight matrix for the second NN with $\hat{z}_1\left(k\right)=[\hat{x}_1(k),\hat{x}_2\left(k\right),x_{1d}(k+1)]^T$. Alternatively, the measured state can be utilized as the NN input without any change in the analysis. Next define the weight estimation error by

$$\tilde{w}_1\left(k\right)=\hat{w}_1\left(k\right)-w_1,$$

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

$$e_2\left(k\right)=x_2\left(k\right)-\hat{x}_{2d}\left(k\right). \tag{23}$$

Equation (18) can be expressed using (23) for $x_2\left(k\right)$ as

$$e_1\left(k+1\right)=f_1\left(k\right)+g_1\left(k\right)\left(e_2\left(k\right)+\hat{x}_{2d}\left(k\right)\right)-x_{1d}\left(k+1\right)+d_1^{'}\left(k\right), \tag{24}$$

or equivalently

$$e_1\left(k+1\right)=g_1(k)(e_2(k)+\xi_1(k)+d_1(k)), \tag{25}$$

where

$$\xi_1(k) = \tilde{w}_1^T(k)\phi(\hat{z}_1(k)), \tag{26}$$

$$w_1^T(k)\phi(\tilde{z}_1(k)) = w_1^T(k)(\phi(\hat{z}_1(k)) - \phi(z_1(k))), \tag{27}$$

and

$$d_1(k) = \frac{d_1^{'}(k)}{g_1(k)} - \varepsilon_1(z_1(k)) + w_1^T(k)\phi(\tilde{z}_1(k)). \tag{28}$$

Rewriting the error $e_2(k)$ from (23) as

$$e_2(k+1) = x_2(k+1) - \hat{x}_{2d}(k+1) = f_2(k) + g_2(k)u(k) - \hat{x}_{2d}(k+1) + d_2^{'}(k), \tag{29}$$

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 (21), $\hat{x}_{2d}(k+1)$ is a smooth nonlinear function of the state $x(k)$, (will be replaced with its estimate), desired trajectory $x_{1d}(k+1)$ and NN weights $\hat{w}_1(k)$. However, one can replace the actual NN weights with virtual control input $\hat{x}_{2d}(k)$ in order to make it a dynamic system which in turn helps in the estimation of $\hat{x}_{2d}(k+1)$. The state error $e_1(k)$ is considered as one of the NN input instead of the desired state to allow faster convergence since preprocessed inputs rather than the raw signals (Lewis et al. 1999) always renders better performance as the NN doesn’t have to reconstruct these signals.

Using these as inputs, a single layer dynamical NN can be used to predict $\hat{x}_{2d}(k+1)$ one step ahead. Consequently, in this paper, a feed forward NN with properly chosen weight tuning law rendering a semi-recurrent or dynamic NN is utilized to predict

$\hat{x}_{2d}(k+1)$. Alternatively, the value can be obtained by employing a filter (Lewis et al. 2002). By using the above mentioned inputs, the first layer of the feedforward architecture-based NN generates $\hat{x}_{2d}(k+1)$ which in turn is used by its second layer to create a suitable control input. On the other hand, one can use a separate single layer dynamic NN to get $\hat{x}_{2d}(k+1)$ which in turn can be utilized as an input to a third control NN to generate a suitable control input. Here, these two single-layer NNs are combined into a single multilayer NN.

Next select the desired control input by using the second NN as

$$u_d(k) = \frac{1}{g_2(k)}\left(- f_2(k) + \hat{x}_{2d}(k+1)\right) + l_1 e_1(k)$$

$$= w_2^T \sigma\left(v_2^T z_2(k)\right) + \varepsilon_2\left(z_2(k)\right) \tag{30}$$

$$= w_2^T \sigma\left(z_2(k)\right) + \varepsilon_2\left(z_2(k)\right), \tag{31}$$

where $w_2 \in R^{n_2}$ and $v_2 \in R^{4\times n_2}$ denote the constant ideal output and hidden layer weights, n_2 is the number of hidden layer nodes, the hidden layer activation function $\sigma\left(v_2^T z_2(k)\right)$ is simplified as $\sigma\left(z_2(k)\right)$, $\varepsilon_1\left(z_2(k)\right)$ is the approximation error, $l_1 \in R$ is the design constant introduced to ensure stability via Lyapunov, $z_2(k) \in R^4$ is the NN input, which is defined next. Equation (30) allows us to avoid the controller singularity problem (Jagannathan 2006) since a single NN is employed.

Considering that the state $x_2(k)$ cannot be measured, $z_2(k)$ is substituted with $\hat{z}_2(k) \in R^4$, where

$$z_2\left(k\right)=\left[x_1\left(k\right),x_2\left(k\right),\hat{x}_{2d}\left(k\right),e_1\left(k\right)\right]^T\in R^4, \tag{32}$$

$$\hat{z}_2\left(k\right)=\left[\hat{x}_1\left(k\right),\hat{x}_2\left(k\right),\hat{x}_{2d}\left(k\right),\hat{e}_1\left(k\right)\right]^T\in R^4, \tag{33}$$

and

$$\hat{e}_1(k)=\hat{x}_1(k)-x_{1d}(k). \tag{34}$$

As mentioned earlier, the measured state can replace its estimate. The control input is now selected as

$$u(k)=\hat{w}_2^T(k)\sigma(\hat{z}_2(k)), \tag{35}$$

where $\hat{w}_2(k)\in R^{n_2}$ is the actual output layer NN weights. Substituting (30) and (35) into (29) yields

$$e_2(k+1)=g_2(k)(l_1e_1(k)+\xi_2(k)+d_2(k)), \tag{36}$$

where

$$\tilde{w}_2(k)=\hat{w}_2(k)-w_2, \tag{37}$$

$$\xi_2(k)=\tilde{w}_2^T(k)\sigma(\hat{z}_2(k)), \tag{38}$$

$$w_2^T(k)\sigma(\tilde{z}_2(k))=w_2^T(k)(\sigma(\hat{z}_2(k))-\sigma(z_2(k))), \tag{39}$$

and

$$d_2(k)=\frac{d_2^{'}}{g_2(k)}-\varepsilon_2(k)+w_2^T(k)\sigma(\tilde{z}_2(k)). \tag{40}$$

Equations (25) and (36) represent the closed-loop error dynamics. It is required to show that the estimation error (12), system errors (25) and (36) and the NN weight matrices $\hat{w}_o(k)$, $\hat{w}_1(k)$ and $\hat{w}_2(k)$ are bounded.

B. Weight Updates for Guaranteed Stability

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

$$\|w_o\| \leq w_{om}, \|w_1\| \leq w_{1m}\text{, and} \|w_2\| \leq w_{2m}, \tag{41}$$

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 (Lewis et al. 1999, Jagannathan 2006) is used.

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

$$\|\varphi(\cdot)\| \leq \varphi_m,\ \|\phi(\cdot)\| \leq \phi_m \text{ and } \|\sigma(\cdot)\| \leq \sigma_m, \tag{42}$$

where $\varphi_m \in R^+$, $\phi_m \in R^+$ and $\sigma_m \in R^+$ are the upper bounds.

Assumption 4 (Bounded NN Approximation Error): The approximation errors $\varepsilon_o(z_o(k-1))$, $\varepsilon_1(z_1(k)))$ and $\varepsilon_2(z_2(k))$ are bounded by ε_{om}, ε_{1m} and ε_{2m}, respectively over the compact set (Igelnik and Pao 1995).

Fact 2: The terms $d_o(k-1) \in R^2$, $d_1(k) \in R$ and $d_2(k) \in R$ are bounded over the compact set S by

$$\|d_o(k-1)\| \leq d_{om} = w_{om}\varphi_m + d_m^{'} + \varepsilon_{om}, \tag{43}$$

where $d_m^{'} \in R^{+}$ is the upper bound for $d_o^{'}(k-1)$, $\left|d_1(k)\right| \le d_{1m}$ and $\left|d_2(k)\right| \le d_{2m}$ where

$$d_{1m} = \frac{d_{1m}^{'}}{g_{1m}} + \varepsilon_{1m} + w_{1m}\phi_m , \tag{44}$$

$$d_{2m} = \frac{d_{2m}^{'}}{g_{2m}} + \varepsilon_{2m} + w_{2m}\sigma_m . \tag{45}$$

In the following theorem, it was demonstrated that the closed-loop signals are bounded.

Theorem 1: Consider the system given by (1) and (2) with the Assumptions 1 through 4 hold. Let the unknown disturbances be bounded by $\left\|d_1^{'}(k)\right\| \le d_{1m}^{'}$ and $\left\|d_2^{'}(k)\right\| \le d_{2m}^{'}$, respectively. Let the observer weight tuning be given by

$$\hat{w}_o(k+1) = \hat{w}_o(k) - \alpha_o \varphi(\hat{z}_o(k))(\hat{w}_o^T(k)\varphi(\hat{z}_o(k)) + l_1 e_1(k+1)I)^T , \tag{46}$$

where $I = [1, 1]^T$, with the virtual control NN weight tuning be provided by

$$\hat{w}_1(k+1) = \hat{w}_1(k) - \alpha_1 \phi(\hat{z}_1(k))(\hat{w}_1^T(k)\phi(\hat{z}_1(k)) + l_1 e_1(k)) , \tag{47}$$

and the control input weight be tuned by

$$\hat{w}_2(k+1) = \hat{w}_2(k) - \alpha_2 \sigma(\hat{z}_2(k))(\hat{w}_1^T(k)\sigma(\hat{z}_2(k)) + l_1 e_1(k)) , \tag{48}$$

where $\alpha_o \in R, \alpha_1 \in R, \alpha_2 \in R,$ and $l_1 \in R$ are design parameters. Let the NN observer, virtual and actual control inputs be defined as (9), (21) and (35), respectively. The estimation error (12), the tracking errors (25) and (36) and the NN weights $\hat{w}_o(k)$, $\hat{w}_1(k)$ and $\hat{w}_2(k)$ are UUB with the bounds given by (A.10) provided the design parameters are selected as

(1) $0 < \alpha_o \left\| \varphi(k) \right\|^2 < 1$, (2) $0 < \alpha_1 \left\| \phi(k) \right\|^2 < 1$,

(3) $0 < \alpha_2 \left\| \sigma(k) \right\|^2 < 1$, (4) $\left| l_1 \right| < \dfrac{1}{3\sqrt{13} g_{1m} g_{2m}}$, (49)

Proof: See Appendix.

Remark 1: Persistency of excitation condition (PE) condition and linearity in the parameter assumption are not used for the NN observer and controller to prove the boundedness of the weights. Additionally, certainty equivalence principle is not employed.

Remark 2: Generally, the separation principle used for linear systems does not hold for nonlinear systems and hence it is relaxed in this paper for the controller design since the Lyapunov function is a quadratic function of system errors and weight estimation errors of the observer and controller NNs.

Remark 3: The NN weight tuning proposed in the (46) through (48) render a semi-recurrent NN due to the proposed weight tuning law even though a feedforward NN is utilized. This semi-recurrent NN renders a dynamic NN which is capable of predicting the state one step-ahead.

IV. Simulation Results

Consider the following nonlinear system, given in non-strict feedback form as

$$x_1(k+1) = -\frac{1}{64} \frac{x_1(k)}{\left(5 + x_2^2(k)\right)} + x_1(k) + 2x_2(k), \tag{50}$$

$$x_2(k+1) = -\frac{1}{16}\frac{x_2(k)}{\left(1+x_1^2(k)\right)} + \left(-\frac{7}{\left(1+x_1^2(k)+x_2^2(k)\right)}\right)u(k), \tag{51}$$

$$y(k) = x_1(k), \tag{52}$$

where $x_i(k) \in R,\ i = 1,2$ are the states, $u(k) \in R$ is the control input, $y(k) \in R$ is the system output, the state $x_1(k)$ is known via the output $y(k)$, the state $x_2(k)$ is immeasurable. Note that $f_1(k) = -\frac{1}{64}\frac{x_1(k)}{\left(5+x_2^2(k)\right)} + x_1(k)$ is a nonlinear function of both $x_1(k)$ and $x_2(k)$.

The objective is to drive the output $y(k) = x_1(k)$ to track the reference signal, which was selected initially as $x_{1d}(k) = 2\sin(\omega kT + \xi)$, where $\omega = 0.1$, $\xi = \frac{\pi}{2}$ with a sampling interval of T = *50msec*. The total simulation time is taken as 250 seconds. The constant l_1 is taken as -0.05 which satisfies (49).

The number of hidden layer neurons in the observer NN, $\hat{w}_o^T\varphi(k)$, controller NN1 $\hat{w}_1^T\phi(k)$ and NN2 $\hat{w}_2^T\sigma(k)$ each was taken as 15. For weight updating, the learning rate is selected as $\alpha_o = 0.01$, $\alpha_1 = 0.1$ and $\alpha_2 = 0.1$. The inputs to observer NN, $\hat{w}_o^T\varphi(k)$, control NNs, $\hat{w}_1^T\phi(k)$ and $\hat{w}_2^T\sigma(k)$ are selected as $\hat{z}_o(k)$, $\hat{z}_1(k)$, and $\hat{z}_2(k)$ (32), respectively. The initial input layer weights for the three NNs are selected at random over an internal of [0, 1] and all the activation functions used are hyperbolic tangent sigmoid functions. The initial output layer weights for all the three NN are chosen to be zero. Measurement noise is assumed to be Gaussian with zero mean and variance of 0.01 was introduced.

Fig. 1 illustrates the performance of the adaptive NN output feedback controller. The system tracking performance is superior even when the state is not measured and in the presence of measured noise. The NN control input is presented in Fig. 2 where it is bounded.

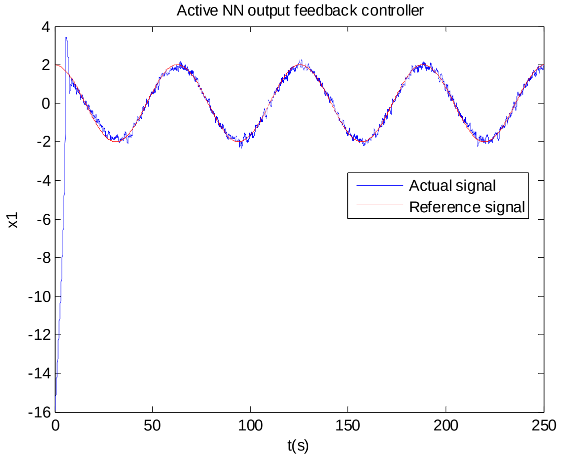

Figure 1. Performance of the NN controller in the presence of measurement noise.

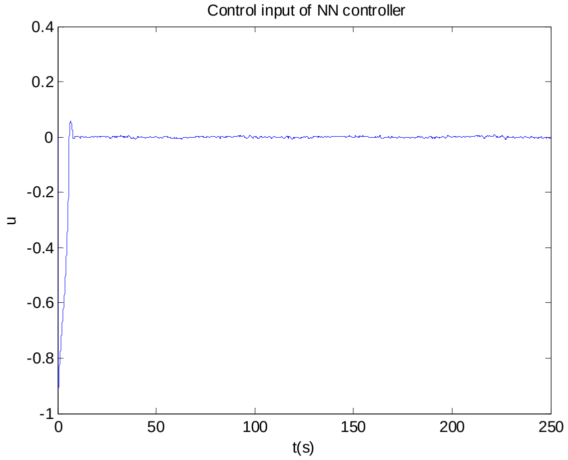

Figure 2. Adaptive NN controller input.

Next a bounded disturbance as in (53) is introduced where the sampling interval is taken as one-second and the desired trajectory being a step input with a 50 second period.

$$d\left(k\right)=\begin{cases}0.0, & 0\le k<100\\ 0.1, & k\ge 100\end{cases} \tag{53}$$

Figs. 3 and 4 depict the response of the proposed NN controller and the control input which is satisfactory.

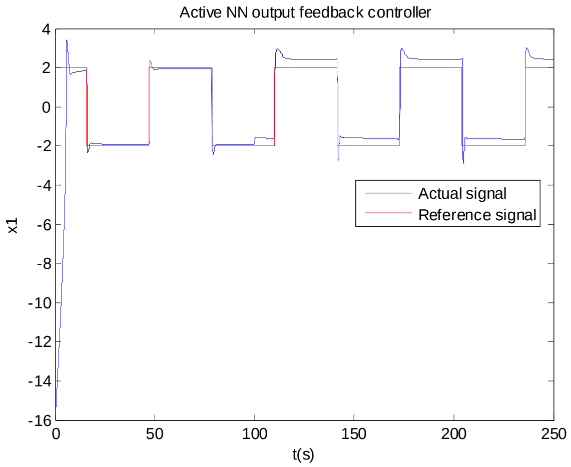

Figure 3. Performance of the NN controller in the presence of bounded disturbance.

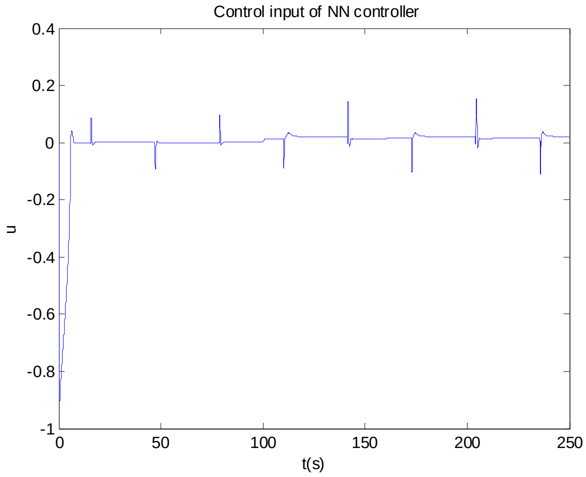

Figure 4. Control input.

V. Conclusions

An adaptive neural network (NN) -based output feedback controller is proposed which consists of three NNs: 1) a NN observer, and 2) two NNs to generate the virtual and real control inputs, respectively. The uniformly ultimate boundedness (UUB) of the closed-loop tracking error, the state estimation errors and the NN weight estimates is shown. Results show that the performance of the proposed controller schemes is highly satisfactory while meeting the closed loop stability. The proposed approach can be easily extendable to an nth order system.

Appendix

Proof of Theorem 1: Define the Lyapunov function

$$\begin{aligned} J(k) &= \frac{1}{4}\tilde{x}^T(k-1)\tilde{x}(k-1) + \frac{l_2}{6g_{1m}^2}e_1^2(k) + \frac{1}{6g_{2m}^2}e_2^2(k) \\ &+ \frac{1}{\alpha_o}tr(\tilde{w}_o^T(k-1)\tilde{w}_o(k-1)) + \frac{1}{\alpha_1}\tilde{w}_1^T(k)\tilde{w}_1(k) + \frac{1}{\alpha_2}\tilde{w}_2^T(k)\tilde{w}_2(k) \\ &+ \frac{1}{\alpha_1}\tilde{w}_1^T(k)\tilde{w}_1(k) + \frac{1}{\alpha_2}\tilde{w}_2^T(k)\tilde{w}_2(k) \end{aligned} \qquad \text{(A.1)}$$

where $l_2 \in R^+$ is a design parameter. The first difference, $\Delta J = J(k+1) - J(k)$, is given by

$$\Delta J(k) = \Delta J_1(k) + \Delta J_2(k) + \Delta J_3(k) + \Delta J_4(k) + \Delta J_5(k) + \Delta J_6(k) \qquad \text{(A.2)}$$

The first term, $\Delta J_1(k)$, is obtained using (12) and by applying the Cauchy's inequality ($\left(\sum_{l=1}^{n} a_l b_l\right)^2 \le \left(\sum_{l=1}^{n} a_l{}^2\right)\left(\sum_{l=1}^{n} b_l{}^2\right)$) as

$$\begin{aligned}\Delta J_1(k) &= \frac{1}{4}\left(\tilde{x}^T(k)\tilde{x}(k)-\tilde{x}^T(k-1)\tilde{x}(k-1)\right) \\ &= \frac{1}{4}\left(\left\|\xi_o(k-1)+d_o(k-1)\right\|^2-\left\|\tilde{x}(k-1)\right\|^2\right) \\ &\le \frac{1}{2}\left\|\xi_o(k-1)\right\|^2+\frac{1}{2}\left\|d_o(k-1)\right\|^2-\frac{1}{4}\left\|\tilde{x}(k-1)\right\|^2\end{aligned} \tag{A.3}$$

Now taking the second term in (A.1) and substituting (25) into (A.1) and using Cauchy's inequality, we get

$$\begin{aligned}&\Delta J_2(k) = \frac{l_2}{6g_{1m}^2}\left(e_1^2(k+1)-e_1^2(k)\right) \\ &= \frac{l_2}{6g_{1m}^2}\left(g_1(k)^2(e_2(k)+\xi_1(k)+d_1(k))^2-e_1^2(k)\right) \\ &\le \frac{l_2}{2}e_2^2(k)+\frac{1}{2}\xi_1^2(k)+\frac{1}{2}d_1^2(k)-\frac{l_2e_1^2(k)}{6g_{1m}^2}\end{aligned} \tag{A.4}$$

The third term in (A.2) can be calculated using (36) as

$$\begin{aligned}&\Delta J_3(k) = \frac{1}{6g_{2m}^2}\left(e_2^2(k+1)-e_2^2(k)\right) \\ &= \frac{1}{6g_{2m}^2}\left(g_2(k)^2(l_1e_1(k)+\xi_2(k)+d_2(k))^2-e_2^2(k)\right) \\ &\le \frac{l_1^2}{2}e_1^2(k)+\frac{1}{2}\xi_2^2(k)+\frac{1}{2}d_2^2(k)-\frac{1}{6g_{2m}^2}e_2^2(k)\end{aligned} \tag{A.5}$$

Taking the fourth term in (A.1) and substituting (46) to get

$$\begin{aligned}\Delta J_4(k) &= \frac{1}{\alpha_o}\left(tr(\tilde{w}_o^T(k)\tilde{w}_o(k))-tr(\tilde{w}_o^T(k-1)\tilde{w}_o(k-1))\right) \\ &= \frac{1}{\alpha_o}\left(\begin{matrix} tr((\tilde{w}_o(k-1)-\alpha_o\varphi(\hat{z}_o(k-1)) \\ (\hat{w}_o^T(k-1)\varphi(\hat{z}_o(k-1))+l_1e_1(k)I)^T)^T \end{matrix}\right.(\tilde{w}_o(k-1) \\ &\quad -\alpha_o\varphi(\hat{z}_o(k-1))(\hat{w}_o^T(k-1) \\ &\quad \varphi(\hat{z}_o(k-1))+l_1e_1(k)I)^T))-tr(\tilde{w}_o^T(k-1)\tilde{w}_o(k-1))\Big)\end{aligned}$$

$$
\begin{aligned}
&\left(\tilde{w}_o^T(k-1)\varphi(\hat{z}_o(k-1))(\hat{w}_o^T(k-1)\varphi(\hat{z}_o(k-1))+l_1e_1(k)I)^T\right)\\
&+\alpha_o tr\begin{pmatrix}(\varphi(\hat{z}_o(k-1))(\hat{w}_o^T(k-1)\varphi(\hat{z}_o(k-1))+l_1e_1(k)I)^T)^T\\ (\varphi(\hat{z}_o(k-1))(\hat{w}_o^T(k-1)\varphi(\hat{z}_o(k-1))+l_1e_1(k)I)^T)\end{pmatrix}\\
&=-2\left\|\xi_o(k-1)\right\|^2-2tr\left(\tilde{w}_o^T(k-1)\varphi(\hat{z}_o(k-1))(w_o^T\varphi(k-1)+l_1e_1(k)I)^T\right)\\
&+\alpha_o\left\|\varphi(k-1)\right\|^2 tr\begin{pmatrix}(\hat{w}_o^T(k-1)\varphi(\hat{z}_o(k-1))\\ +l_1e_1(k)I)(\hat{w}_o^T(k-1)\varphi(\hat{z}_o(k-1))+l_1e_1(k)I)^T\end{pmatrix}\\
&=-2\left\|\xi_o(k-1)\right\|^2-2\xi_o^T(k-1)\left(w_o^T\varphi(k-1)+l_1e_1(k)I\right)\\
&+\alpha_o\left\|\varphi(k-1)\right\|^2(\hat{w}_o^T(k-1)\varphi(\hat{z}_o(k-1))+l_1e_1(k)I)^T\\
&(\hat{w}_o^T(k-1)\varphi(\hat{z}_o(k-1))+l_1e_1(k)I)\\
&=-2\left\|\xi_o(k-1)\right\|^2-2\xi_o^T(k-1)\left(w_o^T\varphi(k-1)+l_1e_1(k)I\right)\\
&+\alpha_o\left\|\varphi(k-1)\right\|^2(\xi_o(k-1)+w_o^T\varphi(k-1)+l_1e_1(k)I)^T\\
&(\xi_o(k-1)+w_o^T\varphi(k-1)+l_1e_1(k)I)\\
&=-(2-\alpha_o\left\|\varphi(k-1)\right\|^2)\left\|\xi_o(k-1)\right\|^2-2(1-\alpha_o\left\|\varphi(k-1)\right\|^2)\\
&\xi_o^T(k-1)\left(w_o^T\varphi(k-1)+l_1e_1(k)I\right)+\alpha_o\left\|\varphi(k-1)\right\|^2\left\|w_o^T\varphi(k-1)+l_1e_1(k)I\right\|^2
\end{aligned}
\tag{A.6}
$$

The fifth term $\Delta J_5(k)$ is obtained using (47) as

$$
\begin{aligned}
\Delta J_5(k)&=\frac{1}{\alpha_1}\left(\tilde{w}_1^T(k+1)\tilde{w}_1(k+1)-\tilde{w}_1^T(k)\tilde{w}_1(k)\right)\\
&=-2\tilde{w}_1^T(k)\left(\phi(\hat{x}(k))(\hat{w}_1^T(k)\phi(\hat{x}(k))+l_1e_1(k))\right)+\alpha_1\left\|\phi(k)\right\|^2\\
&\ (\hat{w}_1^T(k)\phi(\hat{x}(k))+l_1e_1(k))^T(\hat{w}_1^T(k)\phi(\hat{x}(k))+l_1e_1(k))\\
&=-2\xi_1(k)(\xi_1(k)+w_1^T\phi(k)+l_1e_1(k))+\alpha_1\left\|\phi(k)\right\|^2\\
&\ (\xi_1(k)+w_1^T\phi(k)+l_1e_1(k))^T(\xi_1(k)+w_1^T\phi(k)+l_1e_1(k))\\
&=-2(1-\alpha_1\left\|\phi(k)\right\|^2)\xi_1(k)(w_1^T\phi(k)+l_1e_1(k))\\
&\ -(2-\alpha_1\left\|\phi(k)\right\|^2)\xi_1^2(k)+\alpha_1\left\|\phi(k)\right\|^2(w_1^T\phi(k)+l_1e_1(k))^2
\end{aligned}
\tag{A.7}
$$

Using (48), the last term $\Delta J_6(k)$ is expressed as

$$\begin{aligned}
\Delta J_6(k) &= \frac{1}{\alpha_2}\left(\tilde{w}_2^T(k+1)\tilde{w}_2(k+1) - \tilde{w}_2^T(k)\tilde{w}_2(k)\right) \\
&= -2\tilde{w}_2^T(k)\left(\sigma(\hat{z}(k))(\hat{w}_2^T(k)\phi(\hat{x}(k)) + l_1 e_1(k))\right) \\
&\quad +\alpha_2\left\|\sigma(k)\right\|^2(\hat{w}_2^T(k)\sigma(\hat{z}(k)) + l_1e_1(k))^T(\hat{w}_2^T(k)\sigma(\hat{z}(k)) + l_1e_1(k)) \\
&= -2\xi_2(k)(\xi_2(k) + w_2^T\sigma(k) + l_1e_1(k)) + \alpha_2\left\|\sigma(k)\right\|^2 \\
&\quad (\xi_2(k) + w_2^T\sigma(k) + l_1e_1(k))^T(\xi_2(k) + w_2^T\sigma(k) + l_1e_1(k)) \\
&= -2(1-\alpha_2\left\|\sigma(k)\right\|^2)\xi_2(k)(w_2^T\sigma(k) + l_1e_1(k)) \\
&\quad -(2-\alpha_2\left\|\sigma(k)\right\|^2)\xi_2^2(k) + \alpha_2\left\|\sigma(k)\right\|^2(w_2^T\sigma(k) + l_1e_1(k))^2
\end{aligned} \tag{A.8}$$

Combining (A.3) through (A.8) to get the first difference and simplifying to get

$$\begin{aligned}
\Delta J(k) &\le -\frac{1}{2}\left\|\xi_o(k-1)\right\|^2 - \frac{1}{2}\xi_1^2(k) - \frac{1}{2}\xi_2^2(k) + D_M^2 + \frac{1}{4}\left\|\tilde{x}(k-1)\right\|^2 \\
&\quad -(\frac{l_2}{6g_{1m}^2} - \frac{13}{2}l_1^2)e_1^2(k) - (\frac{1}{6g_{2m}^2} - \frac{l_2}{2})e_2^2(k) \\
&\quad -(1-\alpha_0\left\|\varphi(k-1)\right\|^2)\left\|\xi_o(k) + (w_o^T\varphi(k-1) + l_1e_1(k)I)\right\|^2 \\
&\quad -(1-\alpha_1\left\|\phi(k)\right\|^2)(\xi_1(k) + (w_1^T\phi(k) + l_1e_1(k)))^2 \\
&\quad -(1-\alpha_2\left\|\sigma(k)\right\|^2)(\xi_2(k) + (w_2^T\sigma(k) + l_1e_1(k)))^2
\end{aligned} \tag{A.9}$$

where $D_M^2 \le \frac{1}{2}\left(d_{om}^2 + d_{1m}^2 + d_{2m}^2\right) + 2\left(w_{om}\varphi_m + w_{1m}\phi_m + w_{2m}\sigma_m\right)$. This implies that $\Delta J\left(k\right) \le 0$ as long as (49) holds along with the following condition $0 < l_2 < \frac{1}{3g_{2m}^2}$ and

$\left\|\tilde{x}\left(k-1\right)\right\| > 2D_M$, or $\left|e_1\left(k\right)\right| > \frac{D_M}{\sqrt{\frac{l_2}{6g_{1m}^2} - \frac{13}{2}l_1^2}}$,or $\left|e_2\left(k\right)\right| > \frac{D_M}{\sqrt{\frac{1}{6g_{2m}^2} - \frac{l_2}{2}}}$, or

$\left\|\xi_o\left(k-1\right)\right\| > \sqrt{2}D_M$, or $\left|\xi_1\left(k\right)\right| > \sqrt{2}D_M$, or $\left|\xi_2\left(k\right)\right| > \sqrt{2}D_M$. (A.10)

According to the standard Lyapunov extension theorem (Jagannathan 2006), this demonstrates that $\tilde{x}(k-1)$, $e_1(k)$, $e_2(k)$ and the weight estimation errors $\|\tilde{w}_o(k)\|$, $\|\tilde{w}_1(k)\|$ and $\|\tilde{w}_2(k)\|$ or the weight estimates $\hat{w}_o$ (k), $\hat{w}_1$(k) and $\hat{w}_2$(k) are bounded.

Acknowledgments

Research supported in part by the NSF ECCS #0328777 and #0621924.

References

Alolinwi, B. and Khalil, H.K. (1997).Robust adaptive output feedback control of nonlinear systems without persistence of excitation condition, *Automatica*, 33, pp 2025-2032.

Atassi, A.N. and Khalil, H.K. (2003). A separation principle for the stabilization of a class of nonlinear systems, *IEEE Transactions on Automatic Control*, 76, pp. 334-354.

Chen, F. C. and Khalil, H. K. (1995). Adaptive control of a class of nonlinear discrete-time systems using neural networks, *IEEE Trans. Automat. Contr.*, 40(5), pp. 791-801.

Davis, Jr., Daw, C. S., Feldkamp, L. A., Hoard, J. W., Yuan, F., and Connolly, T. (1999). Method of controlling cyclic variation engine combustion, *U.S. Patent*, 5,921,221.

Ge, S. S., Li G. Y., and Lee, T. H. (2003). Adaptive NN control for a class of strict-feedback discrete-time nonlinear systems, Automatica, 39, pp. 807-819.

Hovakimyan, N., Nardi, F., Calise A., and Kim, N. (2002). Adaptive output feedback control of uncertain nonlinear systems using single-hidden-layer neural networks, *IEEE Transaction on Neural Networks*, 13, pp. 1420-1431.

Igelnik, B. and Pao, Y. H. (1995). Stochastic choice of basis functions in adaptive function approximation and the functional-link net, *IEEE Trans. Neural Networks*, 6(6), pp. 1320 – 1329.

Jagannathan, S. (2006). *Neural Network Control of Nonlinear Discrete-time Systems*, Taylor and Francis, Boca Raton, FL.

Kim, Y.H. and Lewis, F.L. (1999). Neural network output feedback control of robot manipulators, *IEEE Transactions on Robotics and Automation*, 15, pp. 301-309.

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

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

Lewis, F. L., Jagannathan, S., and Yesildirek, A. (1999). *Neural Network Control of Robot Manipulators and Nonlinear Systems*, Taylor & Francis, PA.

Yeh, P. C. and Kokotovic, P. V. (1995).Adaptive output feedback design for a class of nonlinear discrete-time systems, *IEEE Trans. Automat. Contr.*, 40(9), pp. 1663-1668.

PAPER 2

Output Feedback Controller for Operation of Spark Ignition Engines at Lean Conditions Using Neural Networks

Jonathan B. Vance[1]*, Brian C. Kaul*[2]*, S. Jagannathan*[1]*, and James A. Drallmeier*[2]

[1]Department of Electrical and Computer Engineering at the University of Missouri-Rolla

[2]Department of Mechanical Engineering at the University of Missouri-Rolla

This work supported in part by the U.S. Department of Education GAANN Fellowship and by the NSF grants ECCS#0327877 and ECCS #0621924. r

Abstract — Spark ignition (SI) engines operating at very lean conditions demonstrate significant nonlinear behavior by exhibiting cycle-to-cycle bifurcation of heat release. Past literature suggests that operating an engine under such lean conditions can significantly reduce NOx emissions by as much as 30% and improve fuel efficiency by as much as 5-10%. At lean conditions, the heat release per engine cycle is not close to constant, as it is when these engines operate under stoichiometric conditions where the equivalence ratio is 1.0. A neural network controller employing output feedback has shown ability in simulation to reduce the nonlinear cyclic dispersion observed under lean operating conditions. This neural network output controller consists of three NNs: 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. The uniform ultimate boundedness of all closed-loop signals is demonstrated by using Lyapunov analysis without using the separation principle. Persistency of excitation

condition, certainty equivalence principle and linearity in the unknown parameter assumptions are also relaxed.

The controller is implemented for a research engine as a program running on an embeddable PC that communicates with the engine through a custom hardware interface, and the results are similar to those observed in simulation. Experimental results at an equivalence ratio of 0.77 show a drop in NOx emissions by around 98% from stoichiometric levels with an improvement of fuel efficiency by 5%. A 30% drop in unburned hydrocarbons from uncontrolled case is observed at this equivalence ratio of 0.77. Similar performance was observed with the controller on a different engine.

Nomenclature

CFR	Cooperative Fuel Research
COV	coefficient of variation
IMEP	Indicated mean effective pressure, $Work/Disp.Volume$
uHC	Unburned hydrocarbons
$CE(k)$	Combustion efficiency
$d_1(k)$	Unknown disturbance in air
$d_2(k)$	Unknown disturbance in fuel
$F(k)$	Fraction of unreacted gas and fuel remaining from previous cycle
R	Stoichiometric air-fuel mass ratio
$u(k)$	Mass change fuel input
$x_1(k)$	Mass of air
$x_2(k)$	Mass of fuel
$\varphi(k)$	Equivalence ratio
φ_l, φ_u	Lower 10 and upper 90 percent locations of the combustion efficiency function
φ_m	Midpoint between φ_l and φ_u

I. Introduction

Modern automobiles utilize microprocessor-based engine control systems to meet stringent federal regulations governing fuel economy and the emissions of CO, NO_x and unburned HC. Current efforts aim to decrease emissions and minimize the fuel consumption. To address these requirements, lean combustion control technology has received increasing attention [1]. Unfortunately, significant cyclic dispersion is exhibited when operating spark ignition engines at extreme lean conditions [2-3], causing engine instability and poor performance.

Several control schemes have been proposed to stabilize engine operation at lean conditions. Inoue *et al.* [1] designed a lean combustion engine control system using a combustion pressure sensor. With the measurement of engine torsional acceleration, Davis *et al.* [4] developed a feedback control approach, which uses fuel as the control variable to reduce the cyclic dispersion. However, system stability is not guaranteed in either [1] or [4] since analysis of stability for nonlinear unknown engine dynamics during combustion is difficult. On the other hand, several control schemes [5-7] using state feedback are available to maintain air to fuel ratio near stoichiometric levels. Maintaining air to fuel ratio near a target value is different than reducing cyclic dispersion at lean engine operating conditions. Cyclic variability at lean engine operation causes instability and degraded performance levels.

Therefore, He *et al.* [8] proposed an adaptive neural network (NN) backstepping controller to maintain stable operation of the SI engine at lean conditions by altering the fuel intake as the control variable. The NN is used to model the complex unknown engine dynamics. Lyapunov analysis is applied to ensure the uniformly ultimate boundedness

(UUB) of the internal system signals. However, to implement the controller, total mass of air and fuel (system states) are required for each engine cycle. These are extremely difficult if not impossible to measure and therefore this controller cannot be implemented. In [9], another control scheme is presented using state feedback for air to fuel ratio control at stoichiometric conditions in order to maximize the benefits of the catalytic converter. As mentioned before, controlling air to fuel ratio at stoichiometric conditions is a totally different problem from reducing cyclic dispersion using heat release as the feedback parameter at lean engine operation. Additionally, cyclic variability exhibits very nonlinear, but to some level deterministic, behavior under lean conditions while being stochastic near stoichiometric operation.

Conventional control schemes [8] have been found incapable of reducing the cyclic dispersion to the levels needed to implement these concepts since the engine dynamics are not taken into consideration. Moreover, the total amount of fuel and air in a given cylinder is normally not measurable on a per-cycle basis which necessitates the development of output feedback control schemes.

Several output feedback controller designs in discrete time are proposed for the single-input-single-out (SISO) nonlinear systems [10-16]. However, no output feedback control scheme currently exists for the proposed class of nonstrict feedback nonlinear discrete-time systems. No controller design is available for nonstrict feedback nonlinear systems even with state feedback.

The separation principle [10,12] does not hold for nonlinear systems, since an exponentially decaying state estimation error can lead to instability at finite time [10].

Consequently, the output feedback control design is in general quite difficult for nonlinear discrete-time systems even though it is highly necessary.

To make the controller implementation more practical, a heat release-based neuro-output feedback controller is proposed in discrete-time to reach stable operation of a single-cylinder spark ignition (SI) engine at lean conditions. Non-catalytic SI engine designs (e.g. generator sets and other industrial applications) could make use of lean operation to reduce engine-out NO_x as well as improve fuel efficiency. The proposed output feedback controller has an observer and a 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. Consequently, the cyclic dispersion is reduced and the engine is stable even when an exact knowledge of engine dynamics is not known to the controller making the NN controller model-free.

The proposed controller is designed for a class of nonlinear discrete-time systems in non-strict feedback form. Both simulation and experimental results show satisfactory performance of the controller. It is important to note that in this work, the output is an unknown function of system states unlike in the existing literature [10-16] where the system output is a known linear function of system states.

The stability analysis of the closed-loop control system is given and the boundedness of the closed loop signals is shown since a stable open loop system can still become unstable with a controller. This stability permits higher levels of diluents to be considered for a specific engine, further enhancing NO_X reduction and fuel efficiency than would be realized on an uncontrolled engine. The NN weights are tuned on-line, with no off-line learning phase required. Moreover, separation principle, persistency of excitation condition, certainty equivalence and linearity in the unknown parameters assumptions are relaxed. Performance of the NN controller is evaluated on different engines and results show satisfactory performance of the controller.

II. Controller Design

A. Background

1) Engine Dynamics

According to the Daw model [2-3], spark ignition (SI) engine dynamics can be expressed as a class of nonlinear systems in non-strict feedback form:

$$x_1(k+1) = AF(k) + F(k)x_1(k) - R \cdot F(k)CE(k)x_2(k) + d_1(k), \tag{1}$$

$$x_2(k+1) = \left(1 - CE(k)\right)F(k)x_2(k) + \left(MF(k) + u(k)\right) + d_2(k), \tag{2}$$

$$y(k) = x_2(k)CE(k), \tag{3}$$

$$\varphi(k) = R\frac{x_2(k)}{x_1(k)}, \tag{4}$$

$$CE(k) = \frac{CE_{\max}}{1 + 100^{-(\varphi(k)-\varphi_m)/(\varphi_u-\varphi_l)}}, \tag{5}$$

and

$$\varphi_m = \frac{\varphi_u - \varphi_l}{2}, \tag{6}$$

where $x_1(k)$ and $x_2(k)$ are total mass of air and fuel, respectively, in the cylinder before k^{th} burn, $y(k)$ is the heat release at k^{th} instant, $CE(k)$ is combustion efficiency for $0 < CE_{\min} < CE(k) < CE_{\max}$, $CE_{\max}$ is the maximum combustion efficiency, $F(k)$ is residual gas fraction for $0 < F_{\min} < F(k) < F_{\max}$, $AF(k)$ is mass of fresh air per cycle, R is stoichiometric air-fuel ratio, $MF(k)$ is mass of fresh fuel per cycle, $u(k)$ is change in mass of fresh fuel per cycle, $\varphi(k)$ is input equivalence ratio, $\varphi_m, \varphi_u, \varphi_l$ are constant system parameters, and $d_1(k)$ and $d_2(k)$ are unknown but bounded disturbances. Since $y(k)$ varies each cycle, the engine is unstable. In the above engine dynamics, both $F(k)$ and $CE(k)$ are unknown nonlinear functions of $x_1(k)$ and $x_2(k)$.

Remark 1: For the system represented by (1) through (3), states of $x_1(k)$ and $x_2(k)$ are typically not measurable [17] and output $y(k)$ can be made available. The control objective is to stably operate the engine at lean conditions ($0 < \varphi(k) < 1$) with only heat release information available – to stabilize $y(k)$ around y_d, where y_d is the target heat release value.

Remark 2: We notice that in (3) the available system output $y(k)$ is an unknown nonlinear function of both immeasurable states of $x_1(k)$ and $x_2(k)$, unlike that in all past literatures [10-16], where $y(k) = x_1(k)$ or $y(k)$ is a known linear combination of system states. This issue makes the observer design more challenging.

2) Engine Dynamics in a Different Form

Substituting (3) into both (1) and (2), we get

$$x_1(k+1) = AF(k) + F(k)x_1(k) - R \cdot F(k)y(k) + d_1(k), \tag{7}$$

$$x_2(k+1) = F(k)\left(x_2(k) - y(k)\right) + \left(MF(k) + u(k)\right) + d_2(k). \tag{8}$$

For actual engine operation, fresh air, $AF(k)$, fresh fuel, $MF(k)$, and residual gas fraction, $F(k)$, can all be viewed as nominal values plus some small and bounded disturbances:

$$AF(k) = AF_0 + \Delta AF(k), \tag{9}$$

$$MF(k) = MF_0 + \Delta MF(k), \tag{10}$$

and

$$F(k) = F_0 + \Delta F(k), \tag{11}$$

where AF_0, MF_0, and F_0 are known nominal fresh air, fresh fuel and residual gas fraction values, respectively. $\Delta AF(k)$, $\Delta MF(k)$, and $\Delta F(k)$ are small, unknown but bounded disturbances for fresh air, fresh fuel, and residual gas fraction, respectively. Their bounds are given by

$$0 \le \left|\Delta AF(k)\right| \le \Delta AF_m, \tag{12}$$

$$0 \le \left|\Delta MF(k)\right| \le \Delta MF_m, \tag{13}$$

and

$$0 \le \left|\Delta F(k)\right| \le \Delta F_m , \tag{14}$$

where ΔAF_m, ΔMF_m, and ΔF_m are the respective upper bounds for $\Delta AF(k)$, $\Delta MF(k)$, and $\Delta F(k)$.

Combine (9)-(11) with (7) and (8), and rewrite (7) and (8) to get

$$\begin{aligned} x_1(k+1) &= AF_0 + F_0 x_1(k) - R \cdot F_0 \cdot y(k) + \\ &\quad \Delta AF(k) + \Delta F(k) x_1(k) - R\Delta F(k) y(k) + d_1(k) \end{aligned}, \tag{15}$$

$$\begin{aligned} x_2(k+1) &= F_0\left(x_2(k) - y(k)\right) + \left(MF_0 + u(k)\right) + \\ &\quad \Delta F(k)\left(x_2(k) - y(k)\right) + \Delta MF(k) + d_2(k) \end{aligned}. \tag{16}$$

Now, at the k^{th} step and based on (3), future heat release, $y(k+1)$ can be predicted as

$$\begin{aligned} y(k+1) &= x_2(k+1)CE(k+1) \\ &= f_3(x_1(k), x_2(k), y(k), u(k)) \end{aligned}, \tag{17}$$

where $f_3(x_1(k), x_2(k), y(k), u(k))$ is an unknown nonlinear function.

It is important to note that the closed-loop stability analysis has to be performed with the proposed NN controller even though many of the engine terms are considered bounded above since a stable open-loop system can still become unstable with a controller unless the NN weight update laws are properly selected. Moreover, a Lyapunov-based stability analysis is needed in order to show the relaxation of the separation principle for the observer and certainty equivalence principle for the controller. Next the NN observer design is introduced.

B. NN Observer Design

A two-layer NN predicts the heat release in the subsequent time interval. The heat release prediction error is utilized to design the system observer. From (17) $y(k+1)$ can be approximated by using a one layer NN as

$$y(k+1) = w_1^T \phi_1\left(v_1^T z_1(k)\right) + \varepsilon_1\left(z_1(k)\right), \tag{18}$$

where $z_1(k) = \left[x_1(k), x_2(k), y(k), u(k)\right]^T \in R^4$ is the network input, matrices $w_1 \in R^{n_1}$ and $v_1 \in R^{4\times n_1}$ represent target output and hidden layer weights, $\phi_1(\cdot)$ represents the hidden layer activation function, n_1 denotes the number of the hidden layer nodes, and $\varepsilon\left(z_1(k)\right) \in R$ is the functional approximation error. As demonstrated in [18], 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 $\varepsilon\left(z_1(k)\right)$ can be made arbitrarily small over the compact set since the activation function forms a basis.

For simplicity define

$$\phi_1\left(z_1(k)\right) = \phi_1\left(v_1^T z_1(k)\right), \tag{19}$$

and

$$\varepsilon_1(k) = \varepsilon\left(z_1(k)\right). \tag{20}$$

Given (19) and (20), (18) is re-written as

$$y(k+1) = w_1^T \phi_1\left(z_1(k)\right) + \varepsilon_1(k). \tag{21}$$

1) Observer Structure

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 is given as

$$\begin{aligned}\hat{y}(k+1) &= \hat{w}_1^T(k)\phi_1\left(v_1^T\hat{z}_1(k)\right)+l_1\tilde{y}(k) \\ &= \hat{w}_1^T(k)\phi_1\left(\hat{z}_1(k)\right)+l_1\tilde{y}(k)\end{aligned}, \tag{22}$$

where $\hat{y}(k+1)$ is the predicted heat release, $\hat{w}_1(k)\in R^{n_1}$ are output layer weights, $\hat{z}_1(k)=\left[\hat{x}_1(k),\hat{x}_2(k),\hat{y}(k),u(k)\right]^T\in R^4$ is the network input, $l_1\in R$ is the observer gain, $\tilde{y}(k)$ is the heat release estimation error, where

$$\tilde{y}(k)=\hat{y}(k)-y(k), \tag{23}$$

and $\phi_1\left(\hat{z}_1(k)\right)$ represents $\phi_1\left(v_1^T\hat{z}_1(k)\right)$, for simplicity.

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

$$\hat{x}_1(k+1)=AF_0+F_0\hat{x}_1(k)-R\cdot F_0\cdot\hat{y}(k)+l_2\tilde{y}(k), \tag{24}$$

and

$$\hat{x}_2(k+1)=F_0\left(\hat{x}_2(k)-\hat{y}(k)\right)+\left(MF_0+u(k)\right)+l_3\tilde{y}(k), \tag{25}$$

where $l_2\in R$ and $l_3\in R$ are observer gains. Here, the initial value of $u(0)$ is assumed to be bounded. Equations (22), (24), and (25) represent the proposed system observer to estimate the states of $x_1(k)$ and $x_2(k)$.

2) Observer Error Dynamics

Let us define the state estimation errors as

$$\tilde{x}_i\left(k\right)=\hat{x}_i\left(k\right)-x_i\left(k\right) \quad {}_{i=1,2}\,. \tag{26}$$

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

$$\begin{aligned}\tilde{x}_1\left(k+1\right)=F_0\tilde{x}_1\left(k\right)+\left(l_2-R\cdot F_0\right)\tilde{y}\left(k\right)-\\ \Delta AF\left(k\right)-\Delta F\left(k\right)x_1\left(k\right)+R\Delta F\left(k\right)y\left(k\right)-d_1\left(k\right)\end{aligned}, \tag{27}$$

$$\begin{aligned}\tilde{x}_2\left(k+1\right)=F_0\tilde{x}_2\left(k\right)+\left(l_3-F_0\right)\tilde{y}\left(k\right)-\\ \Delta F\left(k\right)\left(x_2\left(k\right)-y\left(k\right)\right)-\Delta MF\left(k\right)-d_2\left(k\right)\end{aligned}, \tag{28}$$

and

$$\begin{aligned}&\tilde{y}\left(k+1\right)=\hat{w}_1^T\left(k\right)\phi_1\left(\hat{z}_1\left(k\right)\right)+l_1\tilde{y}\left(k\right)-w_1^T\phi_1\left(z_1\left(k\right)\right)-\varepsilon_1\left(k\right)\\ &=\tilde{w}_1^T\left(k\right)\phi_1\left(\hat{z}_1\left(k\right)\right)+l_1\tilde{y}\left(k\right)-w_1^T\phi_1\left(z_1\left(k\right)\right)\\ &\qquad -\varepsilon_1\left(k\right)+w_1^T\phi_1\left(\hat{z}_1\left(k\right)\right)\\ &=\zeta_1\left(k\right)+l_1\tilde{y}\left(k\right)-\varepsilon_1\left(k\right)+w_1^T\phi_1\left(\tilde{z}_1\left(k\right)\right)\end{aligned} \tag{29}$$

where

$$\tilde{w}\left(k\right)=\hat{w}\left(k\right)-w_1, \tag{30}$$

$$\zeta_1\left(k\right)=\tilde{w}_1^T\left(k\right)\phi_1\left(\hat{z}_1\left(k\right)\right), \tag{31}$$

and, for simplicity, $\phi_1\left(\tilde{z}_1\left(k\right)\right)$ is $\left(\phi_1\left(\hat{z}_1\left(k\right)\right)-\phi_1\left(z_1\left(k\right)\right)\right)$. These substitutions are made to simplify the analysis and to show the boundedness of the closed-loop signals.

C. Adaptive NN Output Feedback Controller

Heat release cyclic dispersion is observed at lean conditions, and, thus, engine operation is unsatisfactory. To stabilize the engine at lean conditions, our control objective is to reduce the heat release cyclic dispersion – drive the heat release toward the target operating point of y_d. Given y_d and the engine dynamics (1) through (5), we could obtain the operating point of total mass of air and fuel in the cylinder, x_{1d} and x_{2d}, respectively. By driving states $x_1(k)$ and $x_2(k)$ to approach their respective operating points x_{1d} and x_{2d}, $y(k)$ will approach the desired value y_d. Then the control objective is realized. With the estimated states $\hat{x}_1(k)$ and $\hat{x}_2(k)$, the controller design follows the backstepping technique [19] detailed in the following sections.

1) Adaptive NN Output Feedback Controller Design

Step 1: **Virtual controller design**. Define system error as

$$e_1(k) = x_1(k) - x_{1d} . \tag{32}$$

Combining with (1), (32) can be rewritten as

$$\begin{aligned} e_1(k+1) &= x_1(k+1) - x_{1d} \\ &= AF(k) + F(k)x_1(k) - x_{1d} - R\cdot F(k)CE(k)x_2(k) + d_1(k) \end{aligned} . \tag{33}$$

For simplicity, let us denote

$$f_1(k) = AF(k) + F(k)x_1(k) - x_{1d} , \tag{34}$$

and

$$g_1(k) = R\cdot F(k)CE(k). \tag{35}$$

Then the system error equation can be expressed as

$$e_1\left(k+1\right)= f_1\left(k\right)-g_1\left(k\right)x_2\left(k\right)+d_1\left(k\right). \tag{36}$$

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

$$x_{2d}\left(k\right)=\frac{f_1\left(k\right)}{g_1\left(k\right)}. \tag{37}$$

The term $x_{2d}\left(k\right)$ can be approximated by the second NN as

$$\begin{aligned} x_{2d}\left(k\right) &= w_2^T\phi_2\left(v_2^T x\left(k\right)\right)+\varepsilon_2\left(x\left(k\right)\right) \\ &= w_2^T\phi_2\left(x\left(k\right)\right)+\varepsilon_2\left(x\left(k\right)\right) \end{aligned}, \tag{38}$$

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

Since both $x_1\left(k\right)$ and $x_2\left(k\right)$ are unavailable, the estimated state $\hat{x}\left(k\right)$ is selected as the NN input. Consequently, the virtual control input is taken as

$$\hat{x}_{2d}\left(k\right)=\hat{w}_2^T\left(k\right)\phi_2\left(v_2^T\hat{x}\left(k\right)\right)=\hat{w}_2^T\left(k\right)\phi_2\left(\hat{x}\left(k\right)\right), \tag{39}$$

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

$$\tilde{w}_2\left(k\right)=\hat{w}_2\left(k\right)-w_2. \tag{40}$$

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

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

Equation (36) can be expressed using (41) for $x_2(k)$ as

$$e_1(k+1) = f_1(k) - g_1(k)\left(e_2(k) + \hat{x}_{2d}(k)\right) + d_1(k), \tag{42}$$

or, equivalently,

$$\begin{aligned} e_1(k+1) &= f_1(k) - g_1(k)\cdot \\ &\quad \left(e_2(k) + x_{2d}(k) - x_{2d}(k) + \hat{x}_{2d}(k)\right) + d_1(k) \\ &= -g_1(k)\left(e_2(k) - x_{2d}(k) + \hat{x}_{2d}(k)\right) + d_1(k) \\ &= -g_1(k)\begin{pmatrix} e_2(k) + \hat{w}_2^T(k)\phi_2\left(\hat{x}(k)\right) - \\ w_2^T(k)\phi_2\left(x(k)\right) - \varepsilon_2\left(x(k)\right) \end{pmatrix} + d_1(k) \end{aligned}. \tag{43}$$

Similar to the calculation of (29), (43) can be further expressed as

$$\begin{aligned} e_1(k+1) &= -g_1(k)\cdot \\ &\quad \left(e_2(k) + \zeta_2(k) + w_2^T\phi_2\left(\tilde{x}(k)\right) - \varepsilon_2\left(x(k)\right)\right) + d_1(k) \end{aligned}, \tag{44}$$

where

$$\zeta_2(k) = \tilde{w}_2^T(k)\phi_2\left(\hat{x}(k)\right), \tag{45}$$

and

$$w_2^T\phi_2\left(\tilde{x}(k)\right) = w_2^T\left(\phi_2\left(\hat{x}(k)\right) - \phi_2\left(x(k)\right)\right). \tag{46}$$

Step 2: **Design of control input** $u(k)$. Rewriting the error $e_2(k)$ from (41) as

$$\begin{aligned} e_2(k+1) &= x_2(k+1) - \hat{x}_{2d}(k+1) \\ &= \left(1 - CE(k)\right)F(k)x_2(k) + \\ &\qquad \left(MF(k) + u(k)\right) - \hat{x}_{2d}(k+1) + d_2(k) \end{aligned}, \tag{47}$$

for simplicity, let us denote

$$f_2(k) = \left(1 - CE(k)\right)F(k)x_2(k) + MF(k). \tag{48}$$

Equation (47) can be written as

$$e_2(k+1) = f_2(k) + u(k) - \hat{x}_{2d}(k+1) + d_2(k). \tag{49}$$

Here, the future value $\hat{x}_{2d}(k+1)$ is not available in the current time step. However, from (37) and (39), observe that $\hat{x}_{2d}(k+1)$ is a smooth nonlinear function of the state $x(k)$ and the virtual control input $\hat{x}_{2d}(k)$. Consequently, $\hat{x}_{2d}(k+1)$ is assumed to be approximated by using another NN with semi-recurrent architecture since a first order predictor generated by this NN is sufficient to obtain this value. Alternatively, a first order filter can be used to obtain the value as given in [20].

Using the third NN, we can now select the desired control input as

$$\begin{aligned} u_d(k) &= \left(-f_2(k) + \hat{x}_{2d}(k+1)\right) \\ &= w_3^T \phi_3\left(v_3^T z_3(k)\right) + \varepsilon_3\left(z_3(k)\right) = w_3^T \phi_3\left(z_3(k)\right) + \varepsilon_3\left(z_3(k)\right) \end{aligned}, \tag{50}$$

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 number of hidden layer nodes, the activation function $\phi_3\left(v_3^T z_3(k)\right)$ is abbreviated by $\phi_3\left(z_3(k)\right)$, $\varepsilon_3\left(z_3(k)\right)$ is the approximation error, and $z_3(k) \in R^3$ is the

NN input, which is given by (51). Considering that 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

$$z_3(k) = \left[x(k), \hat{x}_{2d}(k)\right]^T \in R^3, \tag{51}$$

and

$$\hat{z}_3(k) = \left[\hat{x}(k), \hat{x}_{2d}(k)\right]^T \in R^3. \tag{52}$$

Define

$$\hat{e}_1(k) = \hat{x}_1(k) - x_{1d}, \tag{53}$$

and

$$\hat{e}_2(k) = \hat{x}_2(k) - x_{2d}. \tag{54}$$

The actual control input is now selected as

$$\begin{aligned} u(k) &= \hat{w}_3^T(k)\phi_3\left(v_3^T\hat{z}_3(k)\right) + l_4\hat{e}_2(k) \\ &= \hat{w}_3^T(k)\phi_3\left(\hat{z}_3(k)\right) + l_4\hat{e}_2(k) \end{aligned}, \tag{55}$$

where $\hat{w}_3(k) \in R^{n_3}$ is the actual output layer weights, and $l_4 \in R$ is the controller gain selected to stabilize the system.

Similar to the derivation of (29), combine (49), (50), and (55) yielding

$$\begin{aligned} e_2(k+1) = l_4\hat{e}_2(k) + \xi_3(k) + \\ w_3^T\phi_3\left(\tilde{z}_3(k)\right) - \varepsilon_3\left(z_3(k)\right) + d_2(k) \end{aligned}, \tag{56}$$

where

$$\tilde{w}_3\left(k\right)=\hat{w}_3\left(k\right)-w_3, \tag{57}$$

$$\xi_3\left(k\right)=\tilde{w}_3^T\left(k\right)\phi_3\left(\hat{z}_3\left(k\right)\right), \tag{58}$$

and

$$w_3^T\phi_3\left(\tilde{z}\left(k\right)\right)=w_3^T\left(\phi_3\left(\hat{z}_3\left(k\right)\right)-\phi_3\left(z_3\left(k\right)\right)\right). \tag{59}$$

Equations (44) and (56) represent the closed-loop error dynamics. It is necessary to show that the estimation errors (23) and (26), the system errors (44) and (56), and the NN weight matrices $\hat{w}_1\left(k\right)$, $\hat{w}_2\left(k\right)$, and $\hat{w}_3\left(k\right)$ are bounded.

2) 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

$$\left\|w_1\right\|\le w_{1m}, \left\|w_2\right\|\le w_{2m} \text{, and } \left\|w_3\right\|\le w_{3m}, \tag{60}$$

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

$$\left\|\phi_i\left(\cdot\right)\right\|\le\phi_{im}, i=1,2,3, \tag{61}$$

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

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

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

$$\hat{w}_1\left(k+1\right)=\hat{w}_1\left(k\right)-\alpha_1\phi_1\left(\hat{z}_1\left(k\right)\right)\left(\hat{w}_1^T\left(k\right)\phi_1\left(\hat{z}_1\left(k\right)+l_5\tilde{y}\left(k\right)\right)\right), \tag{62}$$

with the virtual control NN weight tuning provided by

$$\hat{w}_2\left(k+1\right)=\hat{w}_2\left(k\right)-\alpha_2\phi_2\left(\hat{x}\left(k\right)\right)\left(\hat{w}_2^T\left(k\right)\phi_2\left(\hat{x}\left(k\right)+l_6\hat{e}_1\left(k\right)\right)\right), \tag{63}$$

and the control input weight be tuned by

$$\hat{w}_3\left(k+1\right)=\hat{w}_3\left(k\right)-\alpha_3\phi_3\left(\hat{z}_3\left(k\right)\right)\left(\hat{w}_3^T\left(k\right)\phi_3\left(\hat{z}_3\left(k\right)\right)+l_7\hat{e}_2\left(k\right)\right), \tag{64}$$

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 (22), (24), and (25), virtual and actual control inputs be defined as (39) and (55), respectively. The estimation errors (27)-(29), the tracking errors (44) and (56), and the NN weights $\hat{w}_1\left(k\right)$, $\hat{w}_2\left(k\right)$, and $\hat{w}_3\left(k\right)$ are UUB with the bounds specifically given by (A.17) through (A.24) provided the design parameters are selected as:

(a) $0<\alpha_i\left\|\phi_i\left(k\right)\right\|^2<1, \quad i=1,2,3$, (65)

(b) $$l_3^2 < 1 - \frac{\left(l_1 - R \cdot F_0\right)^2}{6R^2 \cdot \Delta F_m^2} - \frac{\left(l_2 - F_0\right)^2}{6\Delta F_m^2} - 4l_5^2, \tag{66}$$

(c) $$l_6^2 < \min\left(\frac{\left(1 - F_0^2\right)}{18R^2 \cdot \Delta F_m^2}, \frac{1}{18R^2}\right), \tag{67}$$

(d) $$l_4^2 + 6l_7^2 < \min\left(\frac{\left(1 - F_0^2\right)}{6\Delta F_m^2}, \frac{1}{3}\right). \tag{68}$$

Proof: See Appendix A.

Remark 3: Given specific values of R, F_0, and ΔF_m, we can derive the design parameters of l_i, i=1,…,7. 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 (66)-(68).

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

Remark 5: A well-defined controller is developed in this paper since a single NN is utilized to approximate two nonlinear functions thereby avoiding division by zero.

Remark 6: It is important to note that in this theorem there is no persistency of excitation (PE) condition for the NN observer and NN controller in contrast with standard work in the discrete-time adaptive control [21] since the first difference of the Lyapunov function in the Appendix does not require the PE condition on input signals to prove the boundedness of the weights. Even though the input to the hidden-layer weight matrix is not updated and only the hidden to the output-layer weight matrix alone is tuned, the NN method relaxes the linearity in the unknown parameter assumption. Additionally, certainty equivalence principle is not used in the proof.

Remark 7: Generally, the separation principle used for linear systems does not hold for nonlinear systems and hence it is relaxed in this paper for the controller design since the Lyapunov function is a quadratic function of system errors and weight estimation errors of the observer and controller NNs.

Remark 8: It is important to notice that the NN outputs are not fed as delayed inputs to the network whereas the outputs of each layer are fed as delayed inputs to the same layer. Thus the NN weight tuning proposed in the (62) through (74) renders a semi-recurrent architecture due to the proposed weight tuning law even though feed forward NNs are utilized in the observer and controller. This semi-recurrent NN architecture creates a dynamic NN which is capable of predicting the state one step-ahead overcoming the non causal controller design.

Remark 9: It is only possible to show boundedness of all the closed-loop signals by using an extension of Lyapunov stability [21-22] due to the presence of approximation errors and bounded disturbances consistent with the literature.

The block diagram representation of the controller with observer, controller and engine are shown in Fig. 1. The SI engine block represents the model during simulations and, during experimentation, the research engine itself.

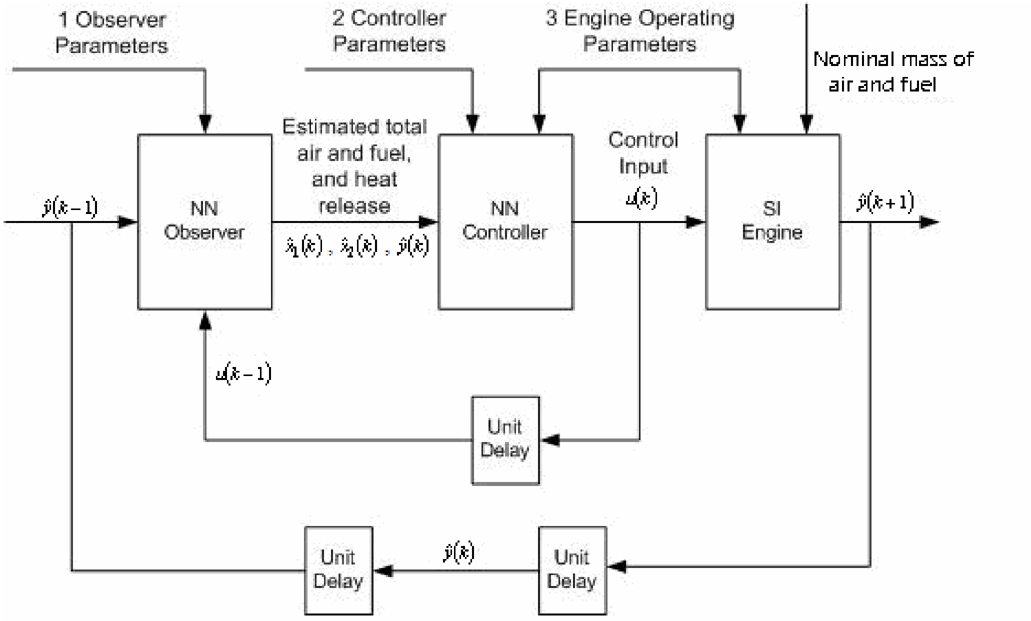

Fig. 1. Structure of system and controller shows the relationship between the observer and controller neural networks as well as the connection to the engine.

III. Simulation

System parameters are selected as: $R = 15.13$, $F = 0.09$, $\varphi_u = 0.725$, $\varphi_l = 0.695$ (by prior analysis to match the simulation output with the experimental data), $CE_{\max} = 1.0$. The controller gains are $l_1 = 1.99$, $l_2 = 0.25$, $l_3 = 0.99$, $l_4 = 0.12$, $l_5 = 0.5$, $l_6 = 0.1$, and $l_7 = 0.7$. Adaptation gains for weight updating are selected as $\alpha_1 = 0.005$, $\alpha_2 = 0.03$, and $\alpha_3 = 0.03$. All of the neural networks have 35 hidden layer nodes. The neuron activation functions are hyperbolic tangent sigmoids in order to ensure the NN approximation capability.

Parameters are chosen to correlate with the research engine used for implementation. Uncontrolled simulation of the engine model is performed for 5,000 cycles whereupon model heat release is stored for analysis. Controlled simulation for 5,000 cycles follows on the engine model using the same parameters. The entire time series of heat release values is plotted in Fig. 2. The 5,000 cycles recorded during control exhibit less instability than the first 5,000 cycles where the engine model was run without control. Observe in Fig. 2 that the average controlled heat release is slightly higher than for uncontrolled, a result of a slight increase in the operating equivalence ratio.

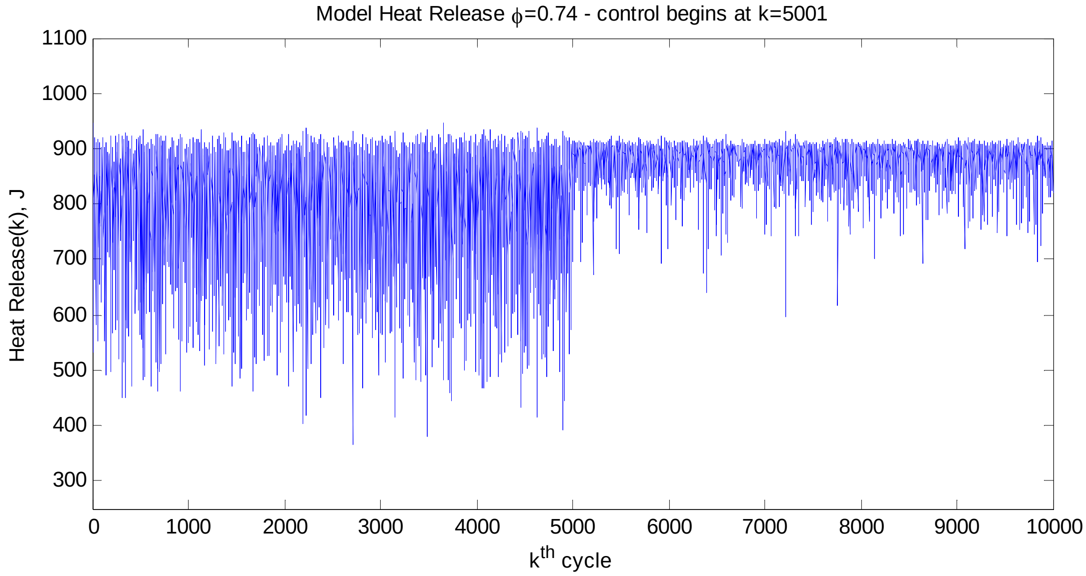

Fig. 2. Discrete time series of heat release shows control beginning at cycle 5001. Cyclic dispersion decreases since less misfires occur after control is applied. Mean heat release also increases.

Fig. 3 shows return maps for the heat release data. A return map is a plot of the heat release for the current cycle versus the next cycle heat release. Under stable engine operation, the heat release from cycle to cycle would appear to be a cluster on the 45° diagonal. The heat release recorded from the engine model without control is on the left plot and heat release during control is on the right plot. The controlled heat release return map on the right exhibits less cyclic dispersion than without control on the left. Hence, the engine model heat release output is more stable with control.

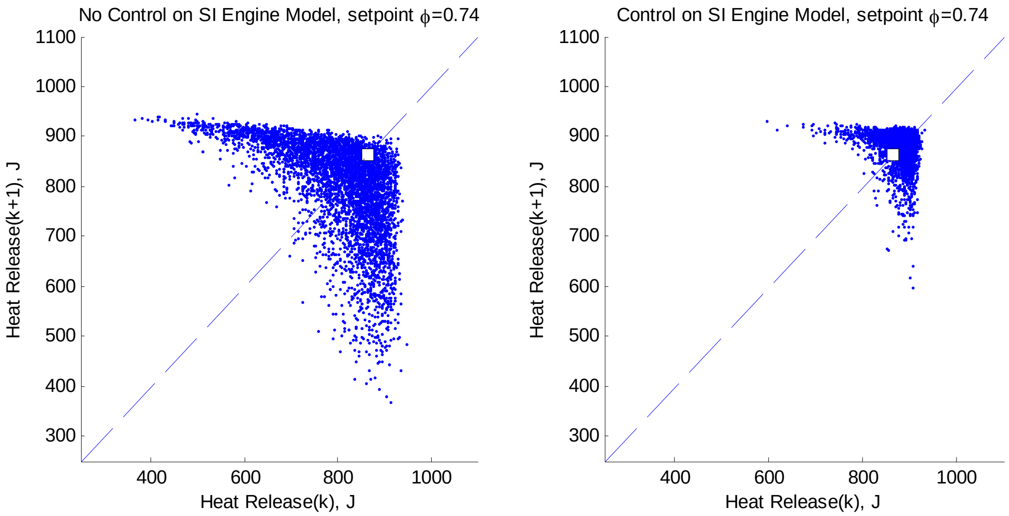

Fig. 3. Uncontrolled and controlled heat release return maps in normalized units of joules generated from the engine model. Current heat release, HR(k), is plotted against next heat release, HR(k+1), where *k* represents cycle number.

Fig. 4 highlights the response of estimated heat release and control input $u(k)$ when a weak combustion cycle is encountered. The controller modifies the fuel control input when such a misfire is detected. Increased fuel intake during control drives the equivalence ratio, ϕ , slightly higher than 0.74.

The scale of heat release seen in Fig. 3 is different from that seen in Fig. 4. The heat release values of the return maps in Fig. 3 are those from the engine model, but the heat release values plotted in Fig. 4 are the internal, controller-scaled, normalized heat release values used in calculations. Also, in Fig. 4 one can see that the observer-estimated heat release is less than the engine model heat release, but there is an observer heat release decrease that indicates engine model misfire detection.

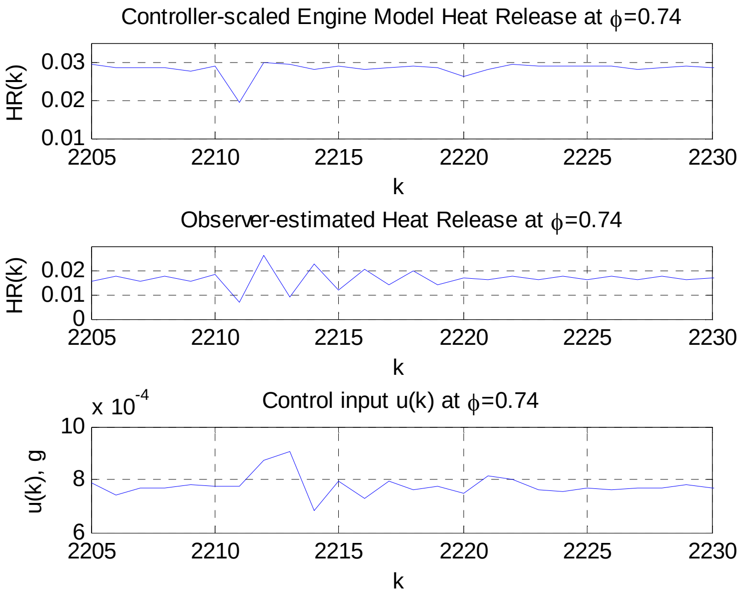

Fig. 4. Simulation heat release output in normalized units of joules from the engine model is plotted for comparison with estimated heat release. The plots are shown with zoom from cycles 2205 to 2230 for detail. When a low heat release value is detected, which is essentially a misfire, the fuel control input increases.

The existence of the observer bias is due to uncertainty of some engine parameters – such as efficiency over a range of equivalence ratios. The oscillation seen in the observer heat release – after the misfire – decays on subsequent engine cycles until another misfire is detected.

IV. Controller Hardware Design

Implementation of the controller is carried out on a cooperative fuel research (CFR) engine. Additional results are obtained on a Ricardo Hydra research engine with a Ford Zetec head. The controller itself is implemented in software, and the algorithm is processed by an embeddable PC running a Linux-based operating system. A special hardware board had to be designed in order to interface the engine and PC signals. Both engines are port fuel-injected, with the fuel injector being driven by an injector driver that receives a TTL signal from this interface board.

The research engines, seen in Fig. 5 (a) and (b), are connected to an electric dynamometer which maintains a constant engine speed of 1000 RPM. The use of a single cylinder engine eliminates the dynamics that would be introduced from interactions between multiple cylinders. A shaft encoder is mounted on the crank shaft to provide a crank angle signal and a hall effect sensor on the cam shaft provides a start of cycle signal. There are 720° of crankshaft rotation per engine cycle, so a crank angle degree is detected approximately every 167 microseconds at 1000 RPM.

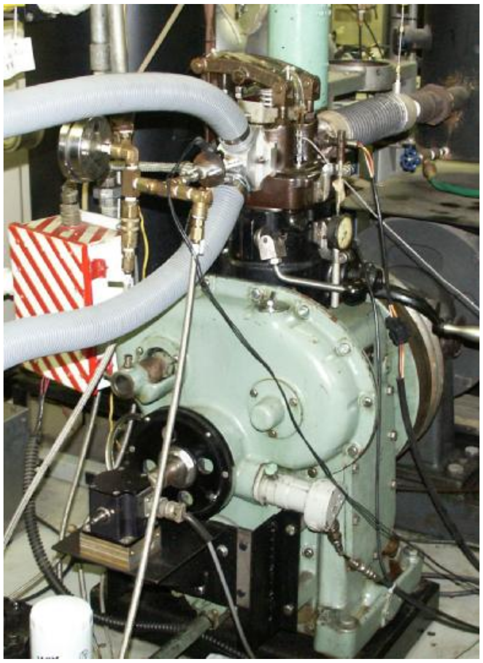

Fig.5 (a). Cooperative Fuel Research (CFR) Engine.

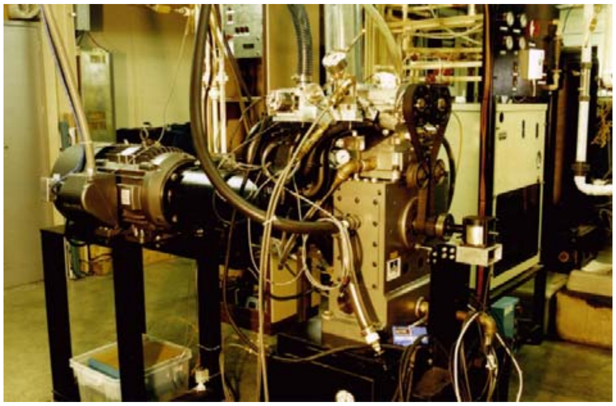

Fig. 5 (b): Ricardo Engine.

In-cylinder pressure measurements are obtained using a Kistler model 6061B water-cooled pressure transducer, coupled to a charge amplifier, which converts the pC charge from the transducer to a 0-10 V signal. The laboratory-grade pressure transducers used in collecting experimental data are too expensive and fragile for production use. However, low-cost, in-cylinder pressure measurement devices are being developed including lower-cost piezo-resistive sensors [23], spark plug boss mounted sensors [24], and fiber-optic sensors [25], so that in-cylinder pressure measurements will be feasible in production automotive engines in the near future. Production quality in-cylinder pressure sensors are currently under development by various companies including Siemens, Kistler, and Delphi.

Heat release for a given engine cycle is calculated by integrating in-cylinder pressure and volume over time. In-cylinder pressure is measured from the engine every half crank angle degree during combustion, over a cycle window from 345° to 490° for the CFR engine (Fig. 5(a)), and every crank angle from 330° to 490° for the Ricardo engine (Fig. 5(b)), for a total of 290, and 130 pressure measurements respectively. At

1000 RPM pressure measurements must be made approximately every 83.3 microseconds.

Fig. 6 shows the timing events in terms of degrees and again in seconds. Start of cycle is labeled SOC, and top dead center is labeled TDC. The pressure window is shown in milliseconds on the second plot as well as the calculation window and the fuel injection window.

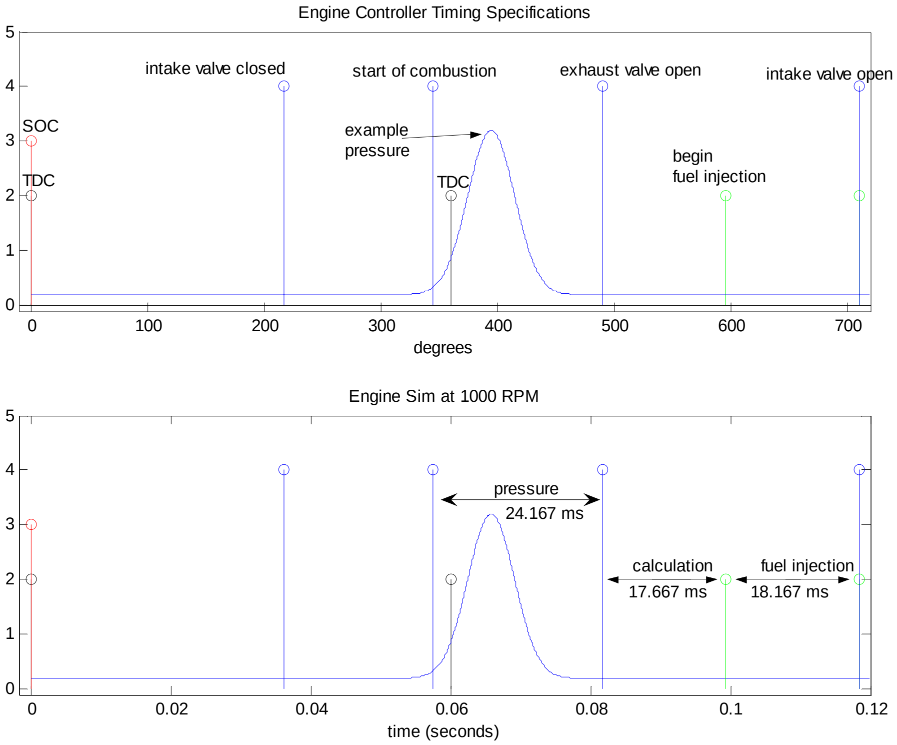

Fig. 6. Timing specifications per cycle for the CFR engine at 1,000 RPM are shown in terms of crank angle degree and again in seconds after the start of cycle.

Notice the timing constraints that are present when an engine is running at 1,000 RPM. The pressure measurement window from 345° to 490° corresponds to 24.167 milliseconds. Also, observe the fuel for the next cycle is injected at the end of the current cycle. The measurement of pressure data and the injection of fuel leave about 17.67 ms for the PC to collect the pressure measurements, calculate heat release, run the controller algorithm, and return the new fuel pulse width to the fuel injector.

The control input is an adjustment to the nominal fuel required at a given equivalence ratio. Fuel injection is controlled by a TTL signal to a fuel injector driver circuit developed for the engine. Pressure measurements come from a charge amplifier which receives a signal from a water-cooled piezoelectric pressure transducer inside the cylinder.

An engine-to-PC interface board was designed to manage the shaft encoder signals, pressure measurements, and fuel injector signal since timing is crucial to correct engine operation. The board uses a microcontroller to buffer the engine hardware signals. A high speed 8-bit A/D converts the pressure measurements. Pressure measurements are sent to the PC where heat release is calculated and then passed to the controller algorithm. A change to the fuel control input, $u(k)$, is returned by the controller algorithm and used to calculate the fuel pulse width for the next engine cycle. This pulse width is a function of mass of fuel to be injected.

The controller algorithm and neural network data structures are implemented in C and compiled to run on an x86 PC. The controller was compiled using the same structure and parameters as for simulation. Configuration files allow the controller parameters to be modified without recompiling. In Fig. 7, a plot of the controller runtime to calculate heat release and the new fuel control input is shown for varying neural network hidden layer size.

Since the number of nodes required in a multilayer NN for a given approximation error is not clear in the literature, the plot in Fig. 7 illustrates that even with large number of hidden-layer NNs the proposed controller can be implemented on the embedded hardware. However, it was found from offline analysis that the improvement in approximation accuracy is not significant beyond 35 hidden-layer nodes and therefore the hidden-layer NN nodes in the observer and controller are limited to 35. From this figure one can see that the time to compute the controller calculations is less than 100 microseconds.

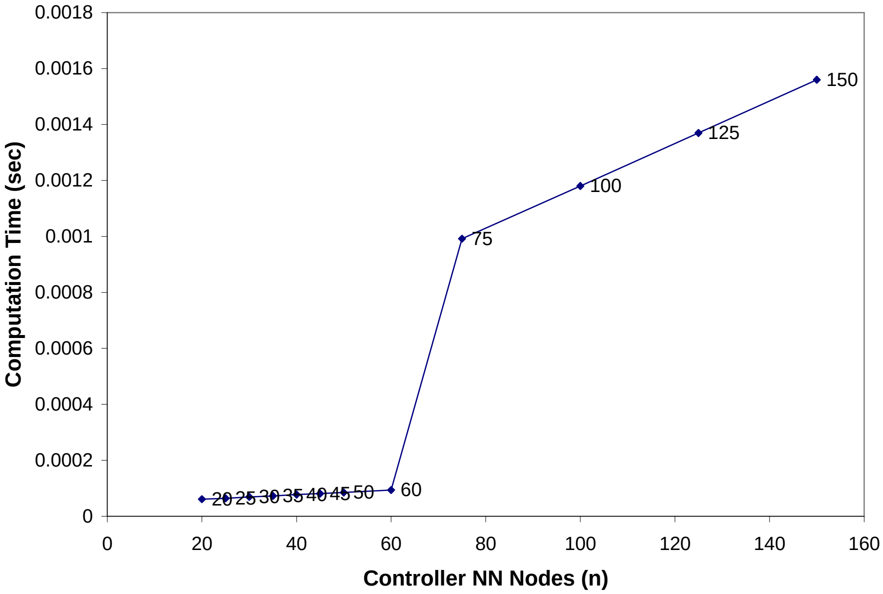

Fig. 7. Controller algorithm runtimes for varying neural network hidden layer size.

V. Experimental Results

During experimentation, the controller was tested at a variety of steady state operating conditions (determined by a combination of engine speed and load) on the engines. The speed was maintained at a constant 1000 RPM for all tests, and the pressure in the intake manifold (manifold absolute pressure, or MAP) was maintained at around 80 kPa for the CFR engine which is roughly a mid-load operating condition, and at around 90 kPa for the Ricardo engine. MAP at full load would be nearly atmospheric pressure and at low load is typically around 40 kPa.

Since the work output from the engine varies with equivalence ratio because reduction in fuel will reduce the engine output, each operating condition is a unique speed/load case. The operation on two different engines also yields more varied test conditions for the controller.

Before activating the controller, air flow is measured and nominal fuel is calculated for the desired equivalence ratio by

$$\varphi = R\left({}^{MF}\!\big/_{AF} \right), \tag{69}$$

where *MF* is nominal mass of fuel and *AF* is nominal mass of air. The nominal fuel and air are loaded into the controller configuration. During data acquisition, ambient pressure is measured when the exhaust valve is fully open at 600° and used to calibrate the combustion pressure measurements. This is necessary to remove any bias generated by charge accumulation on the pressure transducer from which pressure measurements are obtained.

Uncontrolled and controlled heat release data were collected at lean equivalence ratios from 0.79 down to 0.72. NO_x and unburned hydrocarbons (uHC) emissions data were also collected for both uncontrolled and controlled engine operations.

NOx data were measured using a Rosemount Analytical Model 951A NOx analyzer, and uHC data were measured using a Rosemount Analytical Model 400A flame ionization detector. All emissions data are dry gas measurements, averaged over 2 minutes through a data acquisition system.

Uncontrolled engine data means the controller algorithm was not used to modify the fuel injected for each cycle, but the amount of fuel to be injected was set to a nominal value. Controlled engine data comes from the controller modifying the fuel injector pulse width for every cycle. The engine ran for 3,000 cycles uncontrolled, and then 5,000 cycles with the control. Before collecting data the engine was allowed to reach a steady state for each set point according to stable exhaust temperature.

Heat release data is shown in time series and return maps. Time series show the heat release data for the last 500 cycles without control and for the first 500 cycles with control. This illustrates the change in heat release when control is activated. Return maps of heat release are the current cycle of heat release plotted against the next cycle of heat release. This shows the heat release on a per-cycle-basis as well as the general cyclic dispersion. For fair comparison of cyclic dispersion, 3,000 cycles are used to create the uncontrolled return map and 3,000 cycles for the controlled return map.

On each return map of controlled data, there is a percentage that the equivalence ratio increased during control. This percentage increase of the set-point is due to the mean value of fuel during control increasing from the nominal value injected for the cycles without controller operation.

Fig. 8 shows the time series of heat release for an equivalence ratio of 0.79. At index $k=0$ the controller is activated, and mean heat release increases. Note that heat release increases when control is activated, and there are fewer misfires. In Fig. 9 return maps of the uncontrolled and controlled heat release are plotted next to each other. Both the return maps exhibit cyclic dispersion, however, with control the dispersion has decreased. This fact is emphasized by the lower coefficient of variation (COV) of heat release per cycle calculated for each return map.

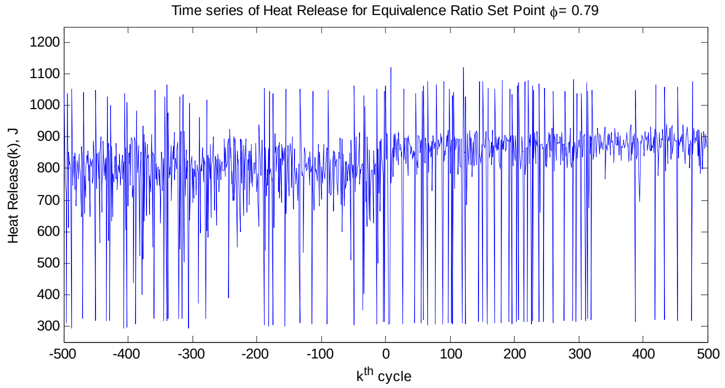

Fig. 8. CFR engine - Time series of heat release at equivalence ratio 0.79.

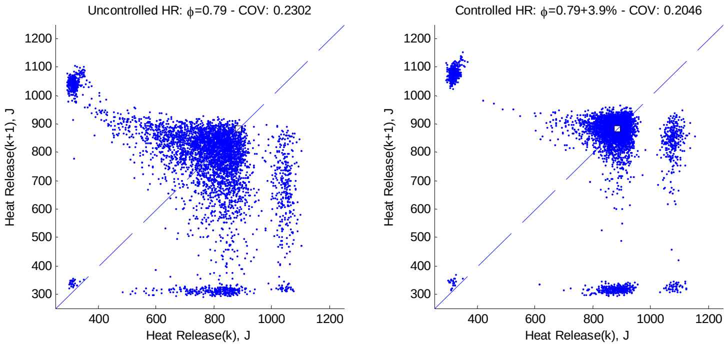

Fig. 9. CFR engine - Return maps of heat release at equivalence ratio 0.79.

The coefficient of variation metric – hereafter referred to as COV – is used to quantify cyclic dispersion in heat release, and is often used as a measure of variability in engine output. It is calculated as the standard deviation of a set of heat release data divided by the mean heat release for that set. A larger COV indicates that heat release values were more dispersed on the return map. With regard to COV, a goal for this controller implementation is to observe a reduction in COV when the control loop is closed on the engine.

Note that heat release appears to be much higher than average after a misfire or partial burn. This stronger-than-average burn can be explained by residual fuel left over in the cylinder from the previous cycle that experienced the weak burn. This results in more fuel to burn for the next cycle causing a higher heat release since the engine is operating lean.

Next, in Fig. 10 the time series of heat release for equivalence ratio 0.77 is plotted. Without control, there is more instability seen at this leaner equivalence ratio than at 0.79. From the plot one can see abundant misfires for the uncontrolled portion of the time series where control begins at index *k=0*. With control applied, the instabilities in the heat release time series reduce substantially. Coefficient of variation decreases from 38.7% to 13.6% when control has been applied.

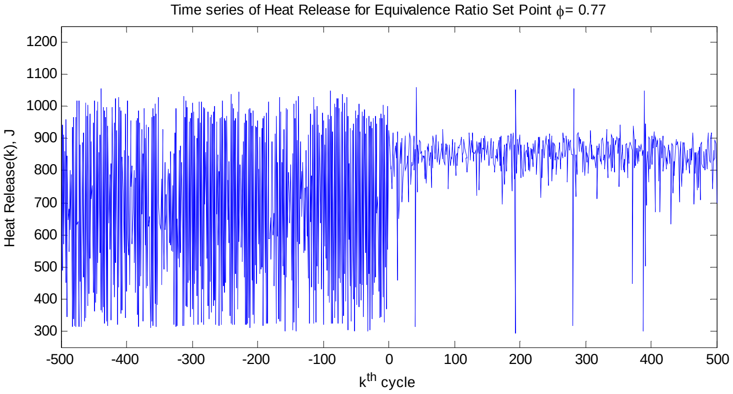

Fig. 10. CFR engine - Time series of heat release at equivalence ratio 0.77.

Looking at Fig. 11, one can see the return maps for the data collected at equivalence ratio 0.77. A decrease in cyclic dispersion is shown by the drop in COV from the uncontrolled return map to the controlled return map.

In Figs. 12 and 13 the time series and return maps of heat release for equivalence ratio 0.75 are plotted. Again, with control applied, instabilities in the heat release time series are reduced substantially. Comparison of the uncontrolled and controlled return maps at equivalence ratio 0.75 in Fig. 13 shows significant decrease in cyclic dispersion. Coefficient of variation decreases from 46.3% to 20.7% when control has been applied.

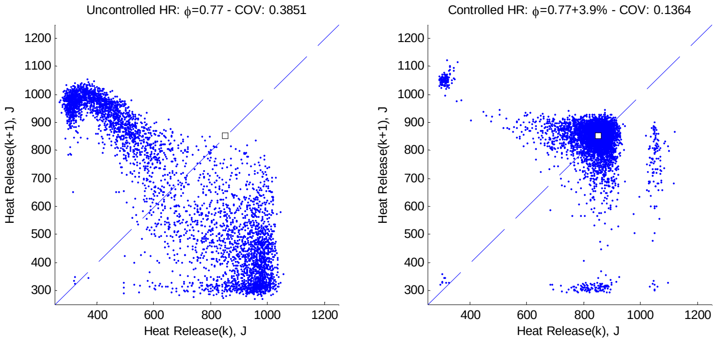

Fig. 11. CFR engine - Return maps of heat release at equivalence ratio 0.77.

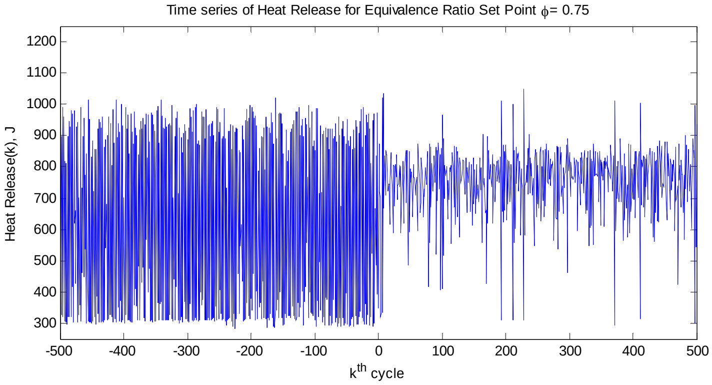

Fig. 12. CFR engine - Time series of heat release at equivalence ratio 0.75.

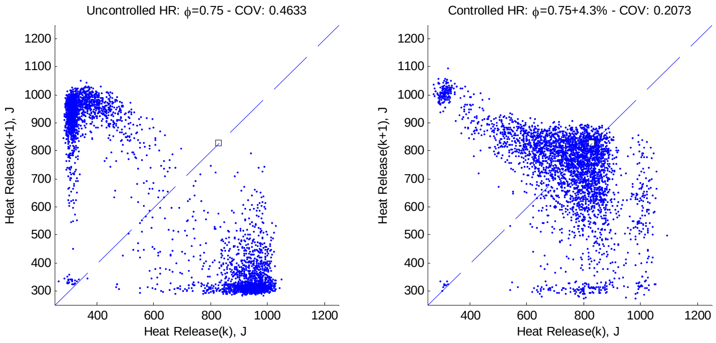

Fig. 13. CFR engine - Return maps of heat release at equivalence ratio 0.75.

The coefficient of variation (COV) for all of the uncontrolled and controlled heat release return maps is shown in Table I. For each equivalence ratio, the uncontrolled COV is greater than the controlled COV, since cyclic dispersion reduced when control was applied. The most significant decrease in cyclic dispersion was observed at equivalence ratio 0.77, where COV fell from 38.6% to 13.6%. This reduction in dispersion translated into a drop of 30% in measured unburned hydrocarbons compared to the uncontrolled case at an equivalence ratio of 0.77. Measured NO_x values decreased by around 98% from levels at stoichiometric conditions.

Table I. Coefficient of Variation for Lean Set-Points of the CFR Engine

$\varphi_{set\text{-}point}$	Uncontrolled COV	Controlled COV
0.79	0.230	0.205
0.77	0.385	0.136
0.75	0.463	0.207

Emissions data are given in Table II. The (u) and (c) prefixes in the column headings stand for uncontrolled and controlled, respectively. The exhaust gas analyzers were used to measure parts-per-million of nitrogen oxides (NO_x) and parts-per-million C_3 unburned hydrocarbons (uHC). Looking at the uncontrolled and controlled data independently, uHC increases as equivalence ratio decreases due to more abundant partial fuel burns. To reduce uHC at lower equivalence ratios, cyclic dispersion must be decreased. The controller is able to reduce the cyclic dispersion which in turn minimizes the uHC. NO_x is decreased at lower equivalence ratios because of lower combustion temperatures.

Additional results from the Ricardo research engine also show the controller's effectiveness at reducing cyclic dispersion. The Ricardo engine was operated at 1,000 rpm like the CFR. The same emissions analyzers were used, and the in-cylinder pressure measurement is similar. In Fig. 14 and 15, time series and return maps are shown for lean equivalence ratio 0.72.

Table II. Emissions Data for Lean Set-Points of the CFR Engine

$\varphi_{set\text{-}point}$	(c) NO_x (PPM)	(u) uHC (PPM)	(c) uHC (PPM)
0.79	351.7	81.4	77.7
0.77	48.2	387.3	283.7
0.75	54.5	913.3	386.1

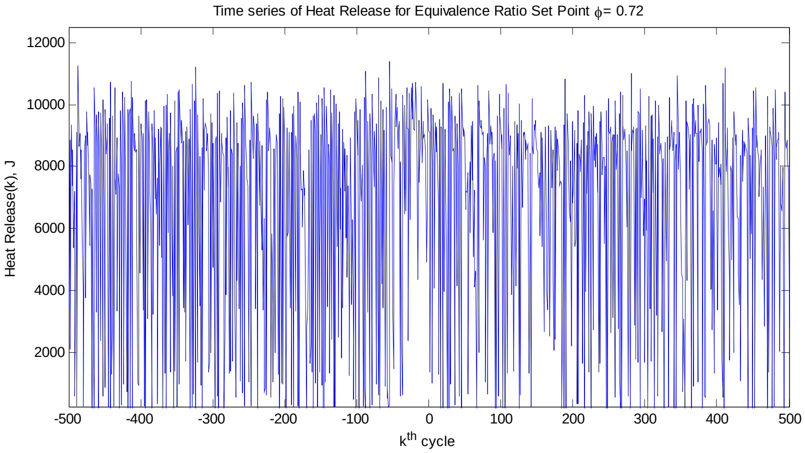

Fig. 14 Ricardo engine - Time series of heat release at equivalence ratio 0.72.

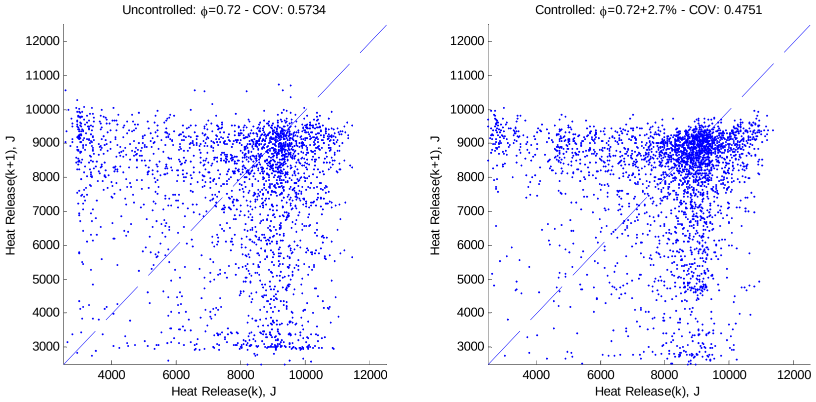

Fig. 15. Ricardo engine - Return maps of heat release at equivalence ratio 0.72

Figs. 16 and 17 contain the heat release information recorded at equivalence ratio 0.75. The coefficient of variation (COV) for the uncontrolled and controlled heat release return maps of the Ricardo engine is shown in Table III. For each equivalence ratio, the uncontrolled COV is greater than the controlled COV. This is an expected result, since the controller should be reducing the cyclic dispersion.

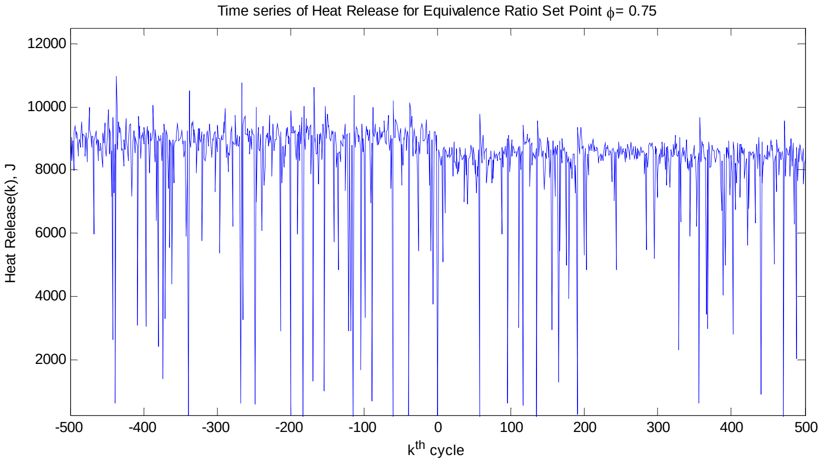

Fig. 16 Ricardo engine - Time series of heat release at equivalence ratio 0.75

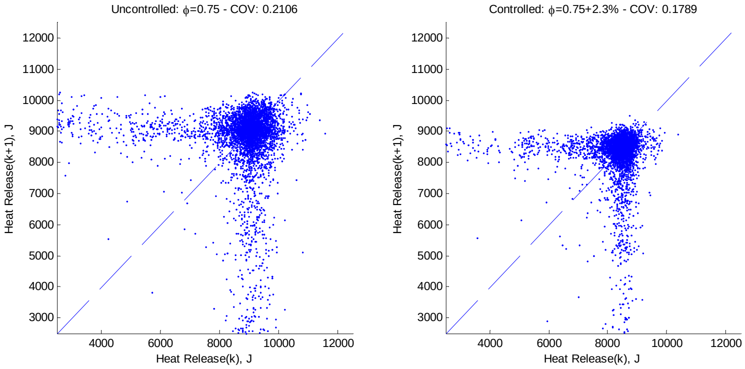

Fig. 17. Ricardo engine - Return maps of heat release at equivalence ratio 0.75.

Table III. Coefficient of Variation for Lean Set-Points of the Ricardo Engine

$\varphi_{\text{set-point}}$	Uncontrolled COV	Controlled COV
0.72	0.573	0.475
0.75	0.211	0.179

The indicated fuel conversion efficiency, η_f, is a measure of the efficiency of the engine in converting the chemical potential energy present in the fuel to actual work. This metric was also calculated for both the uncontrolled and controlled cases. To determine η_f, the net indicated mean effective pressure (IMEP) is calculated by integrating the pressure measured in the cylinder with respect to the cylinder volume, then normalizing by the displacement volume of the engine. The net IMEP, which is a measure of the work output of the engine, is combined with the engine speed to determine an indicated power. Dividing the fuel consumed by the power produced will yield a specific fuel consumption rate, which is then used along with the lower heating value of the fuel, which quantifies its chemical potential energy content, to determine the indicated fuel conversion efficiency.

Due to reduced cyclic dispersion and fewer misfires and low energy cycles, a gain of approximately 5% in indicated fuel conversion efficiency was observed for controlled engine operation.

In Table IV one can see that NOx levels are lower at reduced equivalence ratios. Since cyclic dispersion has been reduced and the engine can operate in a more stable fashion, the amount of partial burns and misfires are reduced. This leads to a reduction of unburned hydrocarbons in the exhaust.

Table IV. Emissions Data for Lean Set-Points of the Ricardo Engine

$\varphi_{set-point}$	(c) NO_x (PPM)	(u) uHC (PPM)	(c) uHC (PPM)
0.72	89	8918	8378
0.75	320	4146	3915

Results from the controller implementation on two different engines exemplify the controller's flexibility. Only engine parameters such as fuel injector information and cylinder geometry had to be changed to extend the controller from the CFR engine to the Ricardo engine. No offline NN training is required and the controller is model-free. Finally, the task of identifying stabilizing initial weights for the observer and controller NNs, a well known problem in the literature [21-22], is overcome by initializing the NN weights to zero.

VI. Conclusions

The spark ignition engine controller aims to decrease emissions by reducing cyclic dispersion encountered during lean operation. Both in model simulation and engine experimentation the controller minimizes estimated heat release error given by (23) returning a noticeable decrease in cyclic dispersion. Although model heat release output does not exhibit all the nonlinearities of actual engine heat release, the controller is still able to reduce heat release error. Correlating the reduction in cyclic dispersion to the measured values of NO_x and unburned hydrocarbons, it is clear that a modest drop in emission products is observed between controlled and uncontrolled scenarios and a significant drop in NO_x from stoichiometric levels while the fuel conversion efficiency shows a 5% improvement. Persistency of excitation condition is not needed, separation principle and certainty equivalence principle are relaxed and linearity in the unknown parameter assumption is not used.

While transient conditions are also encountered in actual engine operations, it is necessary to first develop the ability to control the engine dynamics under steady state conditions. Also, the avoidance of speed and load transients eliminates the need for additional controllers in the system to control equivalence ratio, spark timing, and other parameters, leaving the controller being tested as the only controller in the system so that there are no conflicts or impacts due to other control systems. Once control of lean engine dynamics under steady state speed and load conditions is perfected, transient control will be a logical next step.

Experimental results indicate that the controller can improve engine stability and reduce unburned hydrocarbons at lean engine operation where significant reductions in

NO_x can be realized. Furthermore, the controller is flexible enough to be implemented on two spark ignition research engines.

Appendix

Proof of Theorem 1: Define the Lyapunov function

$$\begin{aligned} J(k) &= \sum_{i=1}^{3} \frac{\gamma_i}{\alpha_i} \tilde{w}_i^T(k)\tilde{w}_i(k) + \frac{\gamma_4}{\alpha_4}\tilde{x}_1^2(k) + \frac{\gamma_5}{5}\tilde{x}_2^2(k) \\ &+ \frac{\gamma_6}{3}\tilde{y}^2(k) + \frac{\gamma_7}{3}e_1^2(k) + \frac{\gamma_8}{4}e_2^2(k) \end{aligned}, \tag{A.1}$$

where $0 < \gamma_i, i = 1,5,8$ are auxiliary constants; the NN weights estimation errors $\tilde{w}_1$, $\tilde{w}_2$, and $\tilde{w}_3$ are defined in (30), (40), and (57), respectively; the observation errors $\tilde{x}_1(k)$, $\tilde{x}_2(k)$, and $\tilde{y}(k)$ are defined in (26) and (23), respectively; the system errors $e_1(k)$ and $e_2(k)$ are defined in (32) and (41), respectively; and $\alpha_i, i = 1,2,3$ are NN adaptation gains. The Lyapunov function (A.1) consisting of the system errors, observation errors, and the weights estimation errors obviates the need for CE condition.

The first difference of the Lyapunov function is given by

$$\Delta J(k) = \sum_{i=1}^{8} \Delta J_i(k). \tag{A.2}$$

The first item of $\Delta J_1(k)$ is obtained using (62) as

$$\begin{aligned}&\Delta J_1(k)=\frac{\gamma_1}{\alpha_1}\left[\tilde{w}_1^T(k+1)\tilde{w}_1(k+1)-\tilde{w}_1^T(k)\tilde{w}_1(k)\right]\\&\le-\gamma_1\left(1-\alpha_1\left\|\phi_1(\cdot)\right\|^2\right)\left(\hat{w}_1^T(k)\phi_1(\cdot)+l_5\tilde{y}(k)\right)^2 \qquad ,\\&-\gamma_1\zeta_1^2(k)+2\gamma_1 l_5^2\tilde{y}^2(k)+2\gamma_1\left(w_1^T\phi_1(\cdot)\right)^2\end{aligned} \tag{A.3}$$

where $\zeta_1(k)$ is defined in (31).

Now taking the second term in the first difference (A.1) and substituting (63) into (A.2), obtain

$$\begin{aligned}&\Delta J_2(k)=\frac{\gamma_2}{\alpha_2}\left[\tilde{w}_2^T(k+1)\tilde{w}_2(k+1)-\tilde{w}_2^T(k)\tilde{w}_2(k)\right]\\&\le-\gamma_2\left(1-\alpha_2\left\|\phi_2(\cdot)\right\|^2\right)\left(\hat{w}_2^T(k)\phi_2(\cdot)+l_6\tilde{x}_1(k)+l_6e_1(k)\right)^2-,\\&\quad\gamma_2\zeta_2^2(k)+3\gamma_2 l_6^2\tilde{x}_1^2(k)+3\gamma_2 l_6^2 e_1(k)+3\gamma_2\left(w_2^T\phi_2(\cdot)\right)^2\end{aligned} \tag{A.4}$$

Taking the third term in the first difference (A.1) and substituting (64) into (A.2), then

$$\begin{aligned}&\Delta J_3(k)=\frac{\gamma_3}{\alpha_3}\left[\tilde{w}_3^T(k+1)\tilde{w}_3(k+1)-\tilde{w}_3^T(k)\tilde{w}_3(k)\right]\\&\le-\gamma_3\left(1-\alpha_3\left\|\phi_3(\cdot)\right\|^2\right)\left(\hat{w}_3^T(k)\phi_3(\cdot)+l_7\tilde{x}_2(k)+l_7e_2(k)\right)^2-\gamma_3\zeta_3^2(k) \quad ,\\&+3\gamma_3 l_7^2\tilde{x}_2^2(k)+3\gamma_3 l_7^2 e_2^2(k)+3\gamma_3\left(w_3^T\phi_3(\cdot)\right)^2\end{aligned} \tag{A.5}$$

Similarly,

$$\begin{aligned}&\Delta J_4(k)=\gamma_4\left[F_0^2\tilde{x}_1^2(k)+\left(l_1-R\cdot F_0\right)^2\tilde{y}^2(k)+\Delta F^2(k)e_1^2(k)\right]\\&+\gamma_4\left[\left(l_1^{'}(k)\right)^2e_2^2(k)+\left(l_1^{'}(k)\right)^2\zeta_2^2(k)+d_{11}^2(k)-\tilde{x}_1^2(k)\right] \qquad ,\end{aligned} \tag{A.6}$$

where

$$l_1^{'}(k)=R\cdot\Delta F(k)\cdot CE(k), \tag{A.7}$$

$$d_{11}\left(k\right)=R\cdot\Delta F\left(k\right)\cdot CE\left(k\right)\cdot w_2^T\phi_2\left(\cdot\right)-\Delta AF\left(k\right)-d_1\left(k\right), \tag{A.8}$$

and $\zeta_2\left(k\right)$ is defined in (45).

$$\begin{aligned}\Delta J_5\left(k\right)&=\gamma_5\left[F_0^2\tilde{x}_2^2\left(k\right)+\left(l_2-F_0\right)^2\tilde{y}^2\left(k\right)+d_{21}^2\left(k\right)-\tilde{x}_2^2\left(k\right)\right]\\&+\gamma_5\left[\left(\left(1-CE\left(k\right)\right)\Delta F\left(k\right)\right)^2\left(e_2^2\left(k\right)+\zeta_2^2\left(k\right)\right)\right]\end{aligned}, \tag{A.9}$$

where

$$d_{21}=-d_2\left(k\right)-\Delta F\left(k\right)\left(1-CE\left(k\right)\right)\cdot w_2^T\phi_2\left(k\right), \tag{A.10}$$

$$\Delta J_6\left(k\right)\le\gamma_6\left(\zeta_1^2\left(k\right)+l_3^2\tilde{y}^2\left(k\right)+d_3^2\left(k\right)-\tilde{y}^2\left(k\right)\right), \tag{A.11}$$

$$\Delta J_7\left(k\right)\le\gamma_7\left(g_1^2\left(k\right)e_2^2\left(k\right)+g_1^2\left(k\right)\zeta_2^2\left(k\right)+d_{12}^2\left(k\right)-e_1^2\left(k\right)\right), \tag{A.12}$$

$$\Delta J_8\left(k\right)\le\gamma_8\left(l_4^2e_2^2\left(k\right)+l_4^2\tilde{x}_2^2\left(k\right)+\zeta_3^2\left(k\right)+d_{22}^2\left(k\right)-e_2^2\left(k\right)\right), \tag{A.13}$$

Combining (A.3) through (A.13) to get the first difference of the Lyapunov function and simplifying it, get

$$\begin{aligned}\Delta J\left(k\right)&\le-\gamma_1\left(1-\alpha_1\left\|\phi_1\left(\cdot\right)\right\|^2\right)\left(\hat{w}_1^T\left(k\right)\phi_1\left(\cdot\right)+l_5\tilde{y}\left(k\right)\right)^2\\&-\gamma_2\left(1-\alpha_2\left\|\phi_2\left(\cdot\right)\right\|^2\right)\left(\hat{w}_2^T\left(k\right)\phi_2\left(\cdot\right)+l_6\tilde{x}_1\left(k\right)+l_6e_1\left(k\right)\right)^2\\&-\gamma_3\left(1-\alpha_3\left\|\phi_3\left(\cdot\right)\right\|^2\right)\left(\hat{w}_3^T\left(k\right)\phi_3\left(\cdot\right)+l_7\tilde{x}_2\left(k\right)+l_7e_2\left(k\right)\right)^2\\&-\left(\gamma_1-\gamma_6\right)\zeta_1^2\left(k\right)\\&-\left(\gamma_2-\gamma_4\left(l_1'\left(k\right)\right)^2-\gamma_5\left(\left(1-CE\left(k\right)\right)\Delta F\left(k\right)\right)^2-\gamma_7g_1^2\left(k\right)\right)\zeta_2^2\left(k\right)\end{aligned}$$

$$\begin{aligned}&-\left(\gamma_3-\gamma_8\right)\zeta_3^2\left(k\right)\\&-\left(\left(1-F_0^2\right)\gamma_4-3\gamma_2 l_6^2\right)\tilde{x}_1^2\left(k\right)\\&-\left(\left(1-F_0^2\right)\gamma_5-3\gamma_3 l_7^2-\gamma_8 l_4^2\right)\tilde{x}_2^2\left(k\right)\\&-\left(\left(1-l_3^2\right)\gamma_6-\left(l_1-R\cdot F_0\right)^2\gamma_4-\left(l_2-F_0\right)^2\gamma_5-2\gamma_1 l_5^2\right)\tilde{y}^2\left(k\right),\\&-\left(\gamma_7-3\gamma_2 l_6^2-\gamma_4\Delta F^2\left(k\right)\right)e_1^2\left(k\right)+D_M^2\\&-\begin{pmatrix}\left(1-l_4^2\right)\gamma_8-3\gamma_3 l_7^2-\gamma_4\left(l_1'\left(k\right)\right)^2-\\\gamma_5\left(\left(1-CE\left(k\right)\right)\Delta F\left(k\right)\right)^2-\gamma_7 g_1^2\left(k\right)\end{pmatrix}e_2^2\left(k\right)\end{aligned}\tag{A.14}$$

where

$$\begin{aligned}D_M^2&=2\gamma_1 w_{1m}^2\phi_{1m}^2+3\gamma_2 w_{2m}^2\phi_{2m}^2+3\gamma_3 w_{3m}^2\phi_{3m}^2+\gamma_4 d_{11m}^2\\&+\gamma_5 d_{21m}^2+\gamma_6 d_{3m}^2+\gamma_7 d_{12m}^2+\gamma_8 d_{22m}^2\end{aligned}.\tag{A.15}$$

Choose $\gamma_1=2$, $\gamma_2=1$, $\gamma_3=2$, $\gamma_4=\frac{1}{6R^2\Delta F_m^2}$, $\gamma_5=\frac{1}{6\Delta F_m^2}$, $\gamma_6=1$, $\gamma_7=\frac{1}{3R^2}$, and $\gamma_8=1$,

then, (A.14) is simplified as

$$\begin{aligned}&\Delta J\left(k\right)\le-2\left(1-\alpha_1\left\|\phi_1\left(\cdot\right)\right\|^2\right)\left(\hat{w}_1^T\left(k\right)\phi_1\left(\cdot\right)+l_5\tilde{y}\left(k\right)\right)^2\\&-\left(1-\alpha_2\left\|\phi_2\left(\cdot\right)\right\|^2\right)\left(\hat{w}_2^T\left(k\right)\phi_2\left(\cdot\right)+l_6\tilde{x}_1\left(k\right)+l_6 e_1\left(k\right)\right)^2\\&-2\left(1-\alpha_3\left\|\phi_3\left(\cdot\right)\right\|^2\right)\left(\hat{w}_3^T\left(k\right)\phi_3\left(\cdot\right)+l_7\tilde{x}_2\left(k\right)+l_7 e_2\left(k\right)\right)^2\\&-\zeta_1^2\left(k\right)-\frac{1}{3}\zeta_2^2\left(k\right)-\zeta_3^2\left(k\right)-\left(\frac{\left(1-F_0^2\right)}{6R^2\Delta F_m^2}-3l_6^2\right)\tilde{x}_1^2\left(k\right)\\&-\left(\frac{\left(1-F_0^2\right)}{6\Delta F_m^2}-6l_7^2-l_4^2\right)\tilde{x}_2^2\left(k\right)\\&-\left(\left(1-l_3^2\right)-\frac{\left(l_1-R\cdot F_0\right)^2}{6R^2\Delta F_m^2}-\frac{\left(l_2-F_0\right)^2}{6\Delta F_m^2}-4l_5^2\right)\tilde{y}^2\left(k\right)\\&-\left(\frac{1}{6R^2}-3l_6^2\right)e_1^2\left(k\right)+D_M^2-\left(\left(1-l_4^2\right)-6l_7^2-\frac{2}{3}\right)e_2^2\left(k\right),\end{aligned}\tag{A.16}$$

This implies $\Delta J\left(k\right) < 0$ as long as (66)-(68) hold and

$$\left|\zeta_1\left(k\right)\right| > D_M\ , \tag{A.17}$$

or

$$\left|\zeta_2\left(k\right)\right| > \sqrt{3}D_M\ , \tag{A.18}$$

or

$$\left|\zeta_3\left(k\right)\right| > D_M\ , \tag{A.19}$$

or

$$\left|\tilde{x}_1\left(k\right)\right| > \frac{D_M}{\sqrt{\frac{\left(1-F_0^2\right)}{6R^2\Delta F_m^2} - 3l_6^2}}\ , \tag{A.20}$$

or

$$\left|\tilde{x}_2\left(k\right)\right| > \frac{D_M}{\sqrt{\frac{\left(1-F_0^2\right)}{6\Delta F_m^2} - 6l_7^2 - l_4^2}}\ , \tag{A.21}$$

or

$$\left|\tilde{y}\left(k\right)\right| > \frac{D_M}{\sqrt{\left(1-l_3^2\right) - \frac{\left(l_1 - R\cdot F_0\right)^2}{6R^2\Delta F_m^2} - \frac{\left(l_2 - F_0\right)^2}{6\Delta F_m^2} - 4l_5^2}}\ , \tag{A.22}$$

or

$$\left|e_1(k)\right| > \frac{D_M}{\sqrt{\frac{1}{6R^2} - 3l_6^2}}, \tag{A.23}$$

or

$$\left|e_2(k)\right| > \frac{D_M}{\sqrt{\left(\frac{1}{3} - l_4^2\right) - 6l_7^2}}. \tag{A.24}$$

According to a standard Lyapunov extension theorem [22], this demonstrates that the system tracking error and the weight estimation errors are *UUB*. The boundedness of $\left\|\zeta_1(k)\right\|$, $\left\|\zeta_2(k)\right\|$, and $\left\|\zeta_3(k)\right\|$ implies that $\left\|\tilde{w}_1(k)\right\|$, $\left\|\tilde{w}_2(k)\right\|$, and $\left\|\tilde{w}_3(k)\right\|$ are bounded, and, further, that the weight estimates $\hat{w}_1(k)$, $\hat{w}_2(k)$, and $\hat{w}_3(k)$ are bounded. Therefore, signals in the closed-loop system are bounded.

References

[1] T. Inoue, S. Matsushita, K. Nakanishi, and H. Okano, “Toyota lean combustion system-The third generation system,” Society of Automotive Engineers, 930873, 1993.

[2] 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.

[3] C. S. Daw, C. E. A. Finney, M. B. Kennel and F. T. Connolly, “Observing and modeling nonlinear dynamics in an internal combustion engine,” *Phys. Rev. E,* vol. 57, no 3, pp.2811 – 2819, 1998.

[4] Jr. Davis, C. S. Daw, L. A. Feldkamp, J. W. Hoard, F. Yuan and T. Connolly, "Method of controlling cyclic variation engine combustion," *U.S. Patent*, 5,921,221, 1999.

[5] C. Alippi, C. de Russis, and V. C. Piuri, "A neural-network based control solution to air fuel ratio control for automotive fuel-injection systems," *IEEE Transactions on Systems, Man and Cybernetics*, Part C, Volume 33, Issue 2, pp. 259– 268, May 2003,.

[6] Z. Weige, J. Jiuchun, X. Yuan, and Z. Xide, "CNG engine air-fuel ratio control using fuzzy neural networks," The 2nd International Workshop on Autonomous Decentralized System, 2002. 6-7, pp.156 – 161, Nov. 2002.

[7] D. R. Hamburg, M. A. Shulman, "A Closed-Loop A/F Control Model for Internal Combustion Engines," SAE Technical Papers: Document Number: 800826.

[8] 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.

[9] H. Itoyama, H. Iwano, K. Osamura and K. Oota, "Air-fuel ratio control system for internal combustion engine," U.S. Patent 0,100,454 A1, 2002.

[10] H. K. Khalil, *Nonlinear Systems*, 2rd Edition, Prentice Hall, Chapter 11, p. 482, 2002.

[11] B. Alolinwi and H.K. Khalil, "Robust adaptive output feedback control of nonlinear systems without persistence of excitation condition," *Automatica*, vol. 33, pp 2025-2032, 1997.

[12] A.N. Atassi and H.K. Khalil, “A separation principle for the stabilization of a class of nonlinear systems,” *IEEE Transactions on Automatic Control*, vol. 76, pp. 334-354, 2003.

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

[14] Y.H. Kim and F.L. Lewis, “Neural network output feedback control of robot manipulators,” *IEEE Transactions on Robotics and Automation*, vol. 15, pp. 301-309, 1999.

[15] 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.

[16] 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.

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

[18] Igelnik, B. and Pao, Y. H., “Stochastic choice of basis functions in adaptive function approximation and the functional-link net,” *IEEE Trans. Neural Networks*, 6(6), 1320 – 1329, 1995.

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

[20] J. Campos, F. L. Lewis, and R. Selmic, “Backlash compensation with filtered prediction in discrete time nonlinear systems by dynamic inversion using neural network,” *Proc. IEEE Conference on Decision and Control*, Sydney, Dec. 2000.

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

[22] S. Jagannathan, *Neural Network Control of Nonlinear Discrete-time Systems*, CRC Press, Boca Raton, FL 2006.

[23] Mark C. Sellnau, Frederic A. Matekunas, Paul A. Battiston, Chen-Fang Chang, and David R. Lancaster, “Cylinder-pressure-based engine control using pressure-ratio-management and low-cost non-intrusive cylinder pressure sensors,” SAE Paper 2000-01-0932, 2000.

[24] Michael Fitzpatrick, Ralf Pechstedt, and Yicheng Lu, “A new design of optical in-cylinder pressure sensor for automotive applications,” SAE Paper 2000-01-0539, 2000.

[25] Rainer Müller, Martin Hart, Gerhard Krötz, Martin Eickhoff, Anthony Truscott, Andrew Noble, Claudio Cavalloni, and Marco Gnielka, “Combustion pressure based engine management system,” SAE Paper 2000-01-0928, 2000.

PAPER 3

Neuro Emission Controller for Minimizing Cyclic Dispersion in Spark Ignition Engines with EGR Levels

J. Vance, S. Jagannathan, and J. Drallmeier*
Dept. of Electrical and Computer Engineering *Dept. of Mechanical and Aerospace Engineering
University of Missouri-Rolla
1870 Miner Circle, Rolla, MO 65409.

Abstract — Past literature has indicated that a significant amount of NO_x can be reduced by operating a spark ignition (SI) engine at the stoichiometric condition with high exhaust gas recirculation (EGR) levels. However, the problem has been the engine instability due to cyclic dispersion in heat release. Since the unknown engine dynamics with high EGR levels are expressed as a nonlinear system in nonstrict feedback form, a suite of novel neural network (NN) control schemes is developed to reduce the cyclic dispersion in heat release by using fuel as the control input. The NN approximation property is utilized to approximate the unknown dynamics. A separate control loop is designed for controlling EGR levels. The first NN control scheme uses the total fuel and air as the state feedback variables whereas the second scheme is a heat release-based output feedback scheme. The stability analysis of the closed loop system is given and the boundedness of all signals is ensured. Online training is used for the adaptive NN and no offline training phase is needed. Simulation and experimental results demonstrate that the cyclic dispersion is reduced approximately by 30%, NO_x dropping by 80% from stiochiometric levels and unburned hydrocarbons by 28% using the proposed controller

from the uncontrolled scenario. The NN controller can also be used to minimize engine emissions at extreme lean condition where similar complex cyclic dynamics are observed.

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). Global warming and its impact on the environment have shifted the focus of the automotive industry. Current efforts are directed at reducing the total amount of emissions and fuel consumption. The engine control system can be classified into three categories [Dudek and Sain 1989]: the spark advance (SA) control, the air-fuel ratio (A/F) control and the exhaust gas recirculation (EGR) control. Partial recirculation of exhaust gases, a technique introduced in the early 70's, has been consistently used for attaining lower emission levels [Sutton and Drallmeier 2000]. Operating a spark ignition engine lean can reduce the NO_x and will improve the fuel efficiency [Inoue et al. 1993, Wagner 1999, He and Jagannathan 2005]. Similarly, substantial reductions in NO_x concentrations have been achieved with 10% to 25% EGR 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% [Sutton and Drallmeier 2000] can be realized. This is the primary motivation of this work.

However, EGR also reduces the combustion rate, which makes stable combustion [Daw et al. 1996, 1998] more difficult to achieve. High levels of EGR present in a spark ignition (SI) engine, can result in further reduction in NO_x but lead to cyclic dispersion in the heat release. 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 authors (Inoue et al. 1993, Wagner 1999, He and Jagannathan 2005) have studied the lean combustion control technology. However, few EGR controller-based engine-out emission results have been reported due to unknown complex engine dynamics and cyclic dispersion in heat release with high EGR dilution, which results in significant performance deterioration of engine performance. Investigation of the onset of complex dynamic behavior in a SI engine with high levels of simulated EGR (added nitrogen) as compared to the lean equivalence ratio case has demonstrated a bifurcation phenomenon (Sutton and Drallmeier, 2000) similar to when the engine was operating under lean conditions. The dynamics of the engine under these conditions are not accurately known before hand due to fuel and air residuals in a cylinder per cycle and combustion efficiency.

Conventional schemes such as proportional, derivative and integral controllers [He and Jagannathan 2005] have been found incapable of reducing the cyclic dispersion to the levels needed to implement these concepts. Moreover, an engine with high EGR levels can only be modeled as a nonlinear system in nonstrict feedback form [He et al. 2005]. At present, no control scheme is reported in the literature for the proposed class of nonstrict feedback nonlinear discrete-time systems [Khalil 2002]. Therefore, to overcome the need to know the complex engine dynamics, a suite of neural network (NN) controller schemes is utilized to minimize the cyclic dispersion because of high levels of EGR dilution since the NN universal approximation property [Igelnik and Pao 1995] can guarantee that the NN can learn to approximate the unknown dynamics.

In this paper, a suite of direct adaptive neural network (NN) controllers is proposed with and without using output feedback for stable operation of the SI engine with high EGR levels at stoichiometric condition. The designed controller learns the unknown engine dynamics by using two NNs and reduces the cyclic dispersion by minimizing variations in equivalence ratio. In the first NN controller using state feedback, equivalence ratio variations are reduced by: 1) keeping the mass of both air and fuel close to their respective target values by an adaptive backstepping approach [Jagannathan 1998]; and 2) maintaining the variations of injected EGR as small as possible around a target level by a separate EGR control loop. In the backstepping approach, the total fuel in the cylinder per cycle is then treated as the virtual control signal to the air intake system so that both the air and fuel states are bounded tightly to their respective targets. Consequently, the cyclic dispersion is reduced and the engine performance becomes satisfactory. A separate control loop ensures the boundedness of actual EGR close to its target for maintaining EGR levels.

In the second output feedback control scheme, heat release is used as the feedback variable and backstepping is utilized to develop a different NN controller. Heat release variations are minimized here by using fuel as the control input. For both the control schemes, the stability analysis of the closed-loop control system is given and the boundedness of the closed loop signals is shown. The NN weights are tuned on-line, with no off-line learning phase required. Simulation and experimental results are included to demonstrate the performance of the proposed controller schemes.

II. Engine as a Nonlinear Discrete-time System

First we introduce the class of nonlinear discrete-time system in nonstrict feedback form and then we show that the engine model falls into this category before we present the development of controller schemes.

A. Non-Strict Nonlinear System Description

Consider the following non-strict feedback nonlinear system described by,

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

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

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

Equations (1) and (2) represent a discrete-time nonlinear system in non-strict feedback form [He et al. 2005], since $f_1(\cdot)$ and $g_1(\cdot)$ are a function 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 [He and Jagannathan 2005]. Control of nonstrict feedback nonlinear systems is introduced in [He at al. 2005] since no known results are available in the literature and controller results from strict feedback nonlinear systems cannot be extended to nonstrict feedback nonlinear systems. The SI engine behavior with high EGR levels can be expressed in this form as discussed next.

B. Engine Dynamics

Daw et al. [Daw et al. 1996, 1998] developed a mathematical representation of the spark ignition (SI) engine behavior to investigate nonlinear cycle dynamics both under lean conditions and high EGR levels. 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. This experimentally validated model for the EGR case [Sutton and Drallmeier 2000] is discussed next.

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

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

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

$$y(k) = x_2(k)CE(k) \tag{6}$$

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

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

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

$$r_{O_2}(k) = \gamma_{O_2} x_2(k) CE(k) \tag{9b}$$

$$r_{N_2}(k) = \gamma_{N_2} R.x_2(k) CE(k) \tag{10a}$$

$$r_{CO_2}(k) = \gamma_{CO_2} x_2(k) CE(k) \tag{10b}$$

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 kth 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. The terms $d_1^{'}(k)$ 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: The control objective then is to operate the engine with high EGR levels by assuming that the states $x_1(k)$ and $x_2(k)$ are available for measurement first and without knowing precisely the engine dynamics.

III. State-feedback NN Controller Design

The control objective is to reduce the cyclic dispersion in the heat release and to improve the performance of a SI engine operating with high EGR levels without needing to know its dynamics, residual gas fraction and combustion efficiency. The heat release, $Q(k) = m(k) \times CE(k)$, is proportional to the mass of the fuel burned and combustion efficiency. In order to reduce the cyclic dispersion in heat release, the variations in both equivalence ratio and the mass of the fuel injected *m(k)* must be minimized. From (8), the combustion efficiency *CE(k)* is a function of equivalence ratio ($\varphi(k) = \frac{1}{R}\frac{m(k)}{a(k)}$) alone. Consequently, the control objective is equivalent to attaining a constant steady state value of *m(k)* and reduction in the variations of$\varphi(k)$. From Eq. (5), this can be realized with a separately controlled EGR system and by keeping both mass of air and fuel tightly bounded to their respective targets. By doing so, inert gas in Eq. (5) will automatically evolve to a stable value since Eq. (5) is a stable linear system with $F(k)$ being always less than one and $r_{H_2O}(k)$, $r_{N_2}(k)$, and $r_{CO_2}(k)$ are finite with small variations. Once the amount of air, fuel, EGR and inert gas is controlled precisely, the variations in equivalence ratio from Eq. (7) are minimized. This implies that the combustion efficiency variations in Eq. (8) are minimized. So the variations in heat release (6) are reduced. Thus, the objective of minimizing the cyclic dispersion is achieved by either driving the mass of fuel and air in the cylinder to approach their respective targets by using mass of the injected fuel as the control variable.

Also, one of the difficulties in designing controllers for the system described by (3) through (5), is that the combustion efficiency (*CE(k)*) and the residual gas fraction, *F(k)*, are unknown. Therefore, the NN function approximation property makes it an ideal candidate to model the unknown nonlinear functions. In the following discussion, a unified and general approach to backstepping type control of nonlinear systems using neural networks is presented. Two NNs are employed to estimate the nonlinear engine dynamics in the backstepping procedure. A virtual control is designed so that air intake is indirectly controlled and the actual control input is designed to control the fuel intake such that both the mass of the fuel and air fed into the cylinder will attain values close to their targets respectively. When the total fuel and air are tightly controlled, the equivalence ratio and the heat release variations are reduced, and the engine can operate smoothly with high EGR levels.

A. Controller Design

Step 1: (State space model) Let us denote

$$x_1\left(k\right)=a\left(k\right),\;\; x_2\left(k\right)=m\left(k\right),\;\; x_3(k)=i(k)\,. \tag{11}$$

Let us choose small changes of the injected fresh fuel as the control variable

$$\delta\, MF\left(k\right)=u\left(k\right). \tag{12}$$

Substituting Eq. (11) and (12) into Eq. (3) through (5) results in

$$
\begin{aligned}
x_1(k+1) = F(k) \cdot \left[x_1(k) - R \cdot CE(k) \cdot x_2(k) + r_{O_2}(k) + r_{N_2}(k) \right] \\
+ \left(1 - F(k)\right) \cdot AF(k) + d_1^{'}(k)
\end{aligned}
\tag{13}
$$

$$
\begin{aligned}
x_2(k+1) = x_2(k) \cdot \left(1 - CE(k)\right) \cdot F(k) \\
+ \left(1 - F(k)\right) \cdot MF(k) + \left(1 - F(k)\right) \cdot u(k) + d_2^{'}(k)
\end{aligned},
\tag{14}
$$

$$
\begin{aligned}
x_3(k+1) = F(k) \cdot x_3(k) \\
+ F(k) \cdot \left(EGR(k) + \left(r_{CO_2}(k) + r_{H_2O}(k) + r_{N_2}(k) \right) \right)
\end{aligned}.
\tag{15}
$$

Once states $x_1(k)$ and $x_2(k)$ are controlled tightly to their respective target values, Eq. (15) can be viewed as a stable linear system (due to $F(k)$ being less than one) with variations in $F(k) \cdot \left(EGR(k) + \left(r_{CO_2}(k) + r_{H_2O}(k) + r_{N_2}(k) \right) \right)$ as the bounded input. Standard linear system theory [Khalil 2002] shows that the state in (15) will be bounded. So in the NN controller design, we do not consider Eq. (15) except treating EGR fed into the system as a finite quantity. EGR will influence the total mass of air and fuel present in the cylinder at each cycle via residuals and inert gases. Now assigning

$$
f_1(k) = F(k) \cdot (x_1(k) + r_{O_2}(k) + r_{N_2}(k)) + \left(1 - F(k)\right) \cdot AF(k), \tag{16}
$$

$$
g_1(k) = -F(k) \cdot R \cdot CE(k), \tag{17}
$$

$$
f_2(k) = x_2(k) \cdot \left(1 - CE(k)\right) \cdot F(k) + \left(1 - F(k)\right) \cdot MF(k), \tag{18}
$$

$$
g_2(k) = 1 - F(k). \tag{19}
$$

Then system Eq. (13) and (14) becomes

$$x_1(k+1) = f_1(k) + g_1(k)x_2(k) + d_1^{'}(k), \tag{20}$$

$$x_2(k+1) = f_2(k) + g_2(k)u(k) + d_2^{'}(k). \tag{21}$$

Since the residual gas fraction $F(k)$ and combustion efficiency *CE(k)* are typically unknown beforehand, $f_1(k)$, $g_1(k)$, $f_2(k)$, and $g_2(k)$ are unknown. Here, NNs are employed to approximate these nonlinear functions.

Step 2: (Virtual controller design).

Assumption 1: Since $g_i(k), i = 1, 2$ are smooth functions, they are bounded within the compact subset S, whose bounds are $g_{1M} > |g_1(k)| > 0$ and $g_{2M} > |g_2(k)| > 0$, respectively, where $g_{1M} \in R^+$ and $g_{2M} \in R^+$.

Define the error between actual and desired air as

$$e_1(k) = x_1(k) - X_{1d}, \tag{22}$$

where X_{1d} is the desired constant value of the mass of the air. Hence Eq. (22) is rewritten as

$$e_1(k+1) = x_1(k+1) - X_{1d} = f_1(k) + g_1(k)x_2(k) - X_{1d} + d_1^{'}(k). \tag{23}$$

By viewing $x_2(k)$ as a virtual control input to Eq. (23), there exists a desired virtual control input given by

$$x_{2d}(k) = g_1^{-1}(k)\left(-f_1(k) + X_{1d}\right) + l_1 e_1(k). \tag{24}$$

where $l_1 \in R$ is a design constant, such that the error, $e_1(k)$, is bounded.

Since $f_1(k)$ and $g_1(k)$ are unknown, $x_{2d}(k)$ cannot be implemented in practice. From Eq. (24), the unknown part $\left(1/g_1(k)\right)\left(-f_1(k)+X_{1d}\right)$ is a smooth function of $x_1(k), x_2(k), x_3(k)$ and X_{1d}. By utilizing NN to approximate the unknown part in Eq. (24), $x_{2d}(k)$ can be expressed as

$$x_{2d}(k) = w_1^T \phi\left(v_1^T z_1(k)\right) + \varepsilon_1\left(z_1(k)\right) + l_1 e_1(k), \tag{25}$$

where

$$z_1(k) = \left[x_1(k), x_2(k), x_3(k), X_{1d}\right]^T, \tag{26}$$

is the input to the first NN, $w_1 \in R^{n_1}$ and $v_1 \in R^{4\times n_1}$ represent the matrix of target weights of the output and hidden layer, respectively, n_1 is the number of hidden layer nodes, $\phi_1(\cdot) \in R^{n_1}$ is the activation function vector, and $\varepsilon_1\left(z_1(k)\right) \in R$ is the NN reconstruction error. It is demonstrated in [Igelnik and Pao 1995] that, if the hidden layer weight, v_1, is chosen initially at random and kept constant and the number of hidden layer nodes is sufficiently large, the NN reconstruction error $\varepsilon_1\left(z_1(k)\right)$ can be made arbitrarily small since the activation function forms a basis.

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

$$\hat{x}_{2d}(k) = \hat{w}_1^T(k)\phi\left(v_1^T z_1(k)\right) + l_1 e_1(k) = \hat{w}_1^T(k)\phi(k) + l_1 e_1(k), \tag{27}$$

where $\hat{w}_1(k) \in R^{n_1}$ is the actual weights of the output layer to be tuned. The hidden layer weight, v_1, is randomly chosen initially and kept constant. For convenience, $\phi\left(v_1^T z_1(k)\right)$ is written as $\phi(k)$. Let the weights estimation error be

$$\tilde{w}_1(k) = \hat{w}_1(k) - w_1 . \tag{28}$$

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

$$e_2(k) = x_2(k) - \hat{x}_{2d}(k) . \tag{29}$$

Equation (23) becomes

$$\begin{aligned} e_1(k+1) &= f_1(k) + g_1(k)(e_2(k) + \hat{x}_{2d}(k)) - X_{1d} + d_1^{'}(k) \\ &= g_1(k)(l_1 e_1(k) + e_2(k) + \zeta_1(k) + d_1(k)), \end{aligned} \tag{30}$$

where

$$\zeta_1(k) = \tilde{w}_1^T(k)\phi(k), \text{ and } d_1(k) = d_1^{'}(k) \big/ g_1(k) - \varepsilon_1(z_1(k)) . \tag{31}$$

*Step 3: (*Design of the control input $u(k)$)

Writing the error in the total mass of fuel in the cylinder from Eq. (21) as

$$e_2(k+1) = x_2(k+1) - \hat{x}_{2d}(k+1) = f_2(k) + g_2(k)u(k) - \hat{x}_{2d}(k+1) + d_2^{'}(k) . \tag{32}$$

Similarly, choosing the desired control input by using the second NN as

$$\begin{aligned} u_d(k) &= \left(1 \big/ g_2(k)\right)\left(-f_2(k) + \hat{x}_{2d}(k+1)\right) + l_2 e_2(k) \\ &= w_2^T(k)\sigma\left(v_2^T z_2(k)\right) + \varepsilon_2(z_2(k)) + l_2 e_2(k) \end{aligned} . \tag{33}$$

where the input to the second NN, $z_2(k) \in R^6$, is taken as

$$z_2(k) = \left[z_1^T(k), e_1(k), e_2(k) \right]^T, \tag{34}$$

Assumption 2: The NN reconstruction errors $\varepsilon_1(z_1(k))$ and $\varepsilon_2(z_2(k))$ are bounded over the compact set S by ε_{1m} and ε_{2m}, respectively.

The actual control input is selected as

$$u(k) = \hat{w}_2^T(k)\sigma(k) + l_2 e_2(k), \tag{35}$$

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

Substituting Eq. (33) and (35) into Eq. (32) yields

$$e_2(k+1) = g_2(k)\left(l_2 e_2(k) + \zeta_2(k) + d_2(k)\right). \tag{36}$$

where

$$\tilde{w}_2(k) = \hat{w}_2(k) - w_2,\ \zeta_2(k) = \tilde{w}_2^T(k)\sigma(k), \text{ and } d_2(k) = d_2^{'}(k)/g_2(k) - \varepsilon_2(z_2(k)). \tag{37}$$

In order to ensure the stability of the closed loop system, suitable NN weight updating rules are now necessary and they are presented next.

B. Closed-loop System Stability Analysis

Before we present the NN weight tuning rules and stability analysis, the following mild assumption is stated.

Assumption 3: Both the ideal weights and the activation functions for all NNs are bounded by known positive values so that

$$\|w_1\| \le w_{1\max},\ \|w_2\| \le w_{2\max},\ \|\phi(\cdot)\| \le \phi_{\max} \text{ and } \|\sigma(\cdot)\| \le \sigma_{\max}. \tag{38}$$

Theorem 3.1: Consider the system given by Eq. (20) and (21). Assume that the Assumptions 1 through 3 hold. Take the first NN weight tuning as

$$\hat{w}_1(k+1) = \hat{w}_1(k) - \alpha_1 \phi(k)\left(\hat{w}_1^T(k)\phi(k) + l_1 e_1(k)\right). \tag{39}$$

with the second NN weight tuning be provided by

$$\hat{w}_2(k+1) = \hat{w}_2(k) - \frac{\alpha_2}{l_2}\sigma(k)\left(\hat{w}_2^T(k)\sigma(k) + l_2 e_2(k)\right). \tag{40}$$

where $\alpha_1 \in R, \alpha_2 \in R$ are NN learning rates, $l_1 \in R$ and $l_2 \in R$ are controller gains. The errors $e_1(k)$ and $e_2(k)$, the NN weights estimates, $\hat{w}_1(k)$ and $\hat{w}_2(k)$ are bounded provided the design parameters are selected as

(a) $0 < \alpha_1 \left\|\phi(\cdot)\right\|^2 < 1,$ (41)

(b) $0 < \alpha_2 \left\|\sigma(\cdot)\right\|^2 < l_2,$ (42)

(c) $0 < |l_1| < 1 \Big/ \left(\sqrt{12} g_{1M}\right),$ (43)

(d) $0 < l_2 < \left(-1 + \sqrt{\left(4 + g_{2M}^2\right)}\right) \Big/ \left(6 g_{2M}\right).$ (44)

Moreover, the equivalence ratio error is bounded; hence, the actual equivalence ratio is also bounded. The heat release dispersion is reduced.

Proof: See Appendix.

Remark 2: Our control scheme requires the need for the measurement of the total mass of the air and fuel, $x_1(k)$ and $x_2(k)$ respectively. Universal Exhaust Gas Oxygen (UEGO) sensors are commercially available to measure the equivalence ratio of the exhaust. The air intake sensor will provide the mass of new air entering the engine and the mass of fuel injected is also available. Using this information and performing experiments on an engine, one can infer the values of $x_1(k)$ and $x_2(k)$. Alternatively, an observer (He and Jagannathan 2005) can be utilized to estimate the values of $x_1(k)$ and $x_2(k)$.

Remark 3: It is important to notice that in this theorem there is no certainty equivalence assumption or the need for the persistence of excitation condition in contrast with standard adaptive discrete-time control (Lewis et al. 1999).

Remark 4: Controller singularity problem $\hat{g}(.) \to 0$ that is commonly encountered in the literature (Khalil 2002) is avoided.

IV. Simulation

The purpose of the simulation is to verify that the cyclic dispersion in heat release of a SI engine using the proposed controller is indeed small and acceptable at high EGR levels. The simulation parameters are selected as the following: 1000 cycles are considered at equivalence ratio of one with variation of 1%, R =14.6, F = 0.15, mass of new air = 9.1295, $\phi_u = 0.82$, $\phi_l = 0.79$, molecular weight of fuel = 114, molecular weight of EGR=30.4, molecular weight of air = 28.84, total gas mole in cylinder = 0.5, and ratio of hydrogen/carbon for fuel = 1.87. Initial conditions for all residues including air, fuel and inert are selected to be equal to zero. The desired mass of air is taken as $X_{1d} = 9.0123$ and the desired mass of fuel is calculated as $X_{2d} = 0.6173$. The gains of controllers are selected as $l_1 = l_2 = 0.1$, respectively. Both NNs contain 15 nodes in the hidden layer. The learning rate is selected as $\alpha_1 = 0.01$ and $\alpha_2 = 0.001$. The initial weights are selected uniformly within an interval of [0, 1] and all the activation functions are selected as hyperbolic tangent sigmoid functions.

EGR was assumed to be a mixture of gases with a molecular weight of 30.4 and fuel had a hydrogen-carbon ratio of 1.87. All initial values of air, fuel and inert gases were chosen to be zero. The neural networks were designed to have 15 neurons each in the hidden layer with learning rates of 0.01 each.

The activation functions used were the hyperbolic tangent sigmoid functions. The simulation was run for 1000 cycles of engine operation by varying the EGR value from 19% to 29%. The attached plots show the results obtained from the simulation runs for EGR levels of 27%. The dispersion without control illustrated in Figure 1 is significant and will make the engine performance unsatisfactory, whereas with control as seen in Figure 2 the dispersion is controlled within a relatively tight bound.

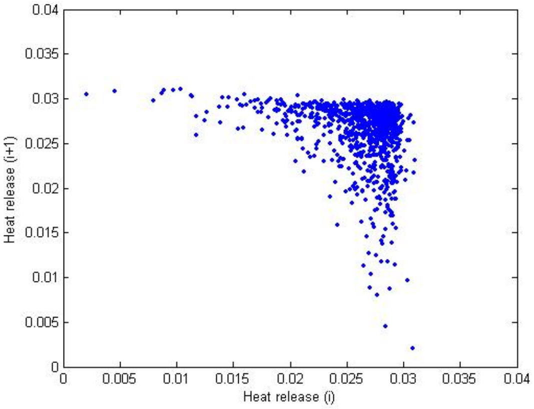

Fig. 1. Heat release without control (27% EGR).

It was observed that with the neural network controller applied, the engine exhibits minimal dispersion with high EGR levels even with perturbation on the residual gas fraction being unknown as depicted in Figure 2. The reduction in dispersion physically translates into fewer misfires and improved drivability even with high EGR levels in the engine. Heat release values are normalized.

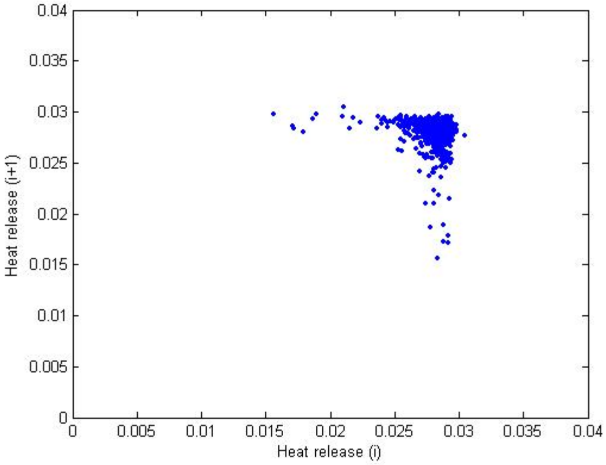

Fig. 2. Heat release with control (27% EGR).

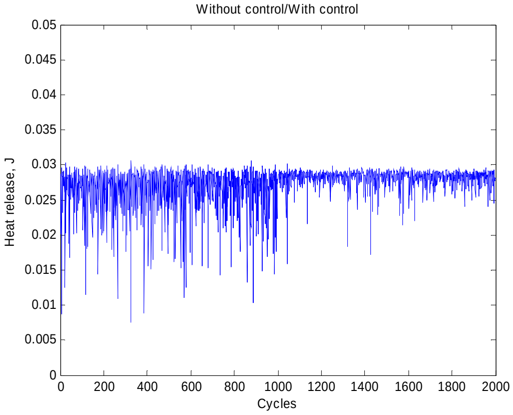

Fig. 3. Heat release with control action beginning at 1001st cycle.

Heat release variations with and without control are indicated as shown in Figure 3. With control applied, the variations in heat release have been minimized. Variations in combustion efficiency are minimized as shown in Figure 4. The associated total fuel (new plus residual) and air plots with and without control for this EGR level are shown in

Figure 5 and 6, respectively. These indicate that the cyclic dispersion in heat release is minimized by the control. As a result, the variations in residual fuel and air are minimized.

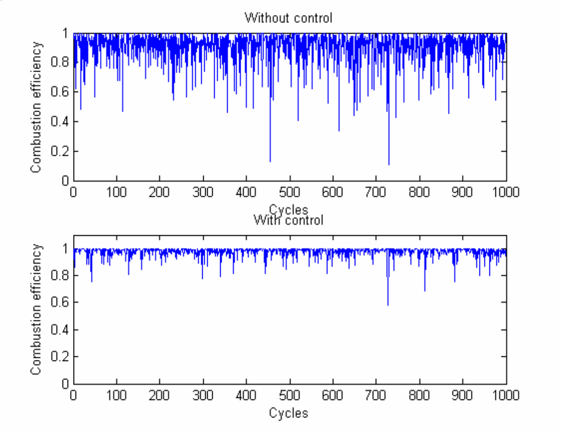

Fig. 4. Combustion efficiency with/without control.

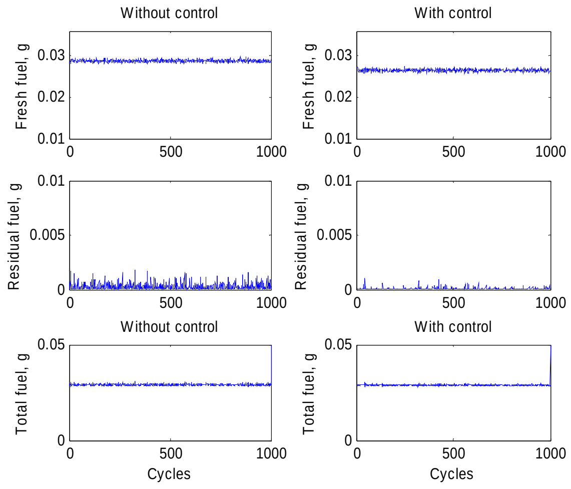

Fig. 5. Total new and residual fuel without/with control action.

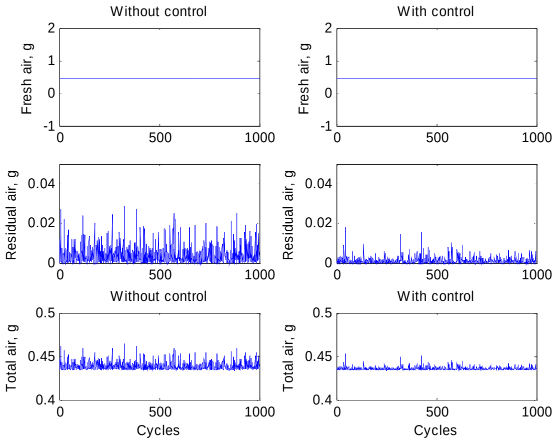

Fig. 6. Total new and residual air without/with control action.

Reductions in cyclic dispersion are observed due to minimization in combustion efficiency variations. These indicate that the cyclic dispersion in heat release is minimized by the controller. As a result, the variations in residual fuel and air are minimized. Although exhibiting very similar dynamics, the return maps of heat release are quantitatively different between the simplified model used for controller development and the actual engine as presented next. This can be attributed to the fact that the engine model simply considers mass conservation and places all complexities of the fluid mechanics and combustion into a phenomenological nonlinear combustion efficiency term. In spite of this simplicity in the model, the designed controller performs highly satisfactorily on the actual engine as will be seen in the next few sections.

V. Output-feedback NN Controller Design

In the last section, the NN controller design is discussed by assuming that the total air and fuel are available for measurement through sensors. The total amount of fuel and air in a given cylinder is normally not directly measurable which necessitates the development of output feedback control schemes.

Very few adaptive output feedback controller designs in discrete time are proposed for the signal-input-single-out (SISO) nonlinear systems; for instance [Yeh and Kokotovic 1995]. 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 NN-based output feedback controller is proposed next by using the NN universal approximation property.

In this section, a direct adaptive NN controller is proposed for stable operation of the SI engine under high levels of EGR. Neural networks are employed to learn the unknown nonlinear dynamics since the residual gas and combustion efficiency are unknown. A backstepping approach in discrete-time is used to design the control input (injected fuel) to the total fuel system. The total fuel is then treated as the virtual control signal to the air system so that both the air and fuel states are bounded tightly to their respective targets. Consequently, the cyclic dispersion is reduced and the engine is stable. A separate control loop is designed for maintaining EGR levels. No exact knowledge of the engine dynamics is needed making the NN controller model-free. The stability analysis of the closed-loop control system is given and the boundedness of the closed loop signals is shown. The NN weights are tuned on-line, with no off-line learning phase required.

Remark 3: In (3) through (6), states $x_1(k)$ and $x_2(k)$ are not measurable whereas the output $y(k)$ is available for measurement if a thermocouple or a pressure sensor is utilized. The control objective then is to operate the engine with high EGR levels with $y\left(k\right)$ as the feedback parameter and without knowing precisely the engine dynamics. 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.

A. Engine Dynamics using Nominal Values

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

$$x_1(k+1) = F(k)[x_1(k) - R.y(k) + r_{O_2}(k) + r_{N_2}\left(k\right)] + x_{1new}(k) + d_1(k) \tag{45}$$

$$x_2(k+1) = F(k)(x_2(k) - y(k)) + x_{2new}(k) + u(k) + d_2^{'}(k) \tag{46}$$

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 since a separate control loop, most likely a standard controller, designed to control EGR levels makes the inert gases evolve into a stable value. It is important to notice that engine-out emissions are the main focus of this paper using a controller and not maintaining how close the EGR level to its target value. Therefore, it is not included here.

Consider,

$$x_{1new}(k) = x_{1new0} + \Delta x_{1new}(k) \tag{47}$$

$$x_{2new}(k) = x_{2new0} + \Delta x_{2new}(k) \tag{48}$$

$$F(k) = F_0(k) + \Delta F(k) \tag{49}$$

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,

$$0 \le \left|\Delta x_{1new}(k)\right| \le \Delta x_{1newM} \tag{50}$$

$$0 \le \left|\Delta x_{2new}(k)\right| \le \Delta x_{2newM} \tag{51}$$

$$0 \le \left|\Delta F(k)\right| \le \Delta F_M \tag{52}$$

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

$$\begin{aligned} x_1(k+1) = (F_0(k) + \Delta F(k))[x_1(k) - R.CE(k)x_2\left(k\right) \\ +r_{O_2}(k) + r_{N_2}\left(k\right)] + x_{1new0} + \Delta x_{1new}(k) + d_1(k) \end{aligned} \tag{53}$$

$$x_2(k+1) = (F_0(k) + \Delta F(k))(1 - CE(k))x_2\left(k\right) + x_{2new0} + \Delta x_{2new}(k) + u(k) + d_2^{'}(k) \tag{54}$$

B. 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

$$y(k+1) = x(k+1)CE(k+1) \tag{55}$$

From (6), the heat release for the next cycle $y(k+1)$ can be approximated by using a one layer neural network as

$$y(k+1) = w_1^T \phi_1(v_1^T z_1(k)) + \varepsilon_1(z_1(k)) \tag{56}$$

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 [Igelnik and Pao 1995].

For simplicity, we define

$$\phi_1(z_1(k)) = \phi_1(v_1^T z_1(k)) \tag{57}$$

$$\varepsilon_1(k) = \varepsilon_1(z_1(k)) \tag{58}$$

Given (56) and (57), (56) is re-written as

$$y(k+1) = w_1^T \phi_1(z_1(k)) + \varepsilon_1(k) \tag{59}$$

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,

$$\hat{y}(k+1) = \hat{w}_1^T \phi_1(v_1^T \hat{z}_1(k)) + l_1\tilde{y}(k) = \hat{w}_1^T(k)\phi_1(\hat{z}_1(k)) + l_1\tilde{y}(k) \tag{60}$$

where $\hat{y}(k+1)$ is the predicted heat release, $\hat{w}_1(k) \in R^{n_i}$ 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

$$\tilde{y}(k) = \hat{y}(k) - y(k) \tag{61}$$

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 in the following form as

$$\hat{x}_1(k+1) = x_{1new0}(k) + F_o \hat{x}_1(k) - R \cdot F_o \cdot \hat{y}(k) + l_2 \tilde{y}(k) \tag{62}$$

$$\hat{x}_2(k+1) = F_o(\hat{x}_2(k) - \hat{y}(k)) + (x_{2new0}(k) + u(k)) + l_3 \tilde{y}(k) \tag{63}$$

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 (62) 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 (62) and (63) represent the dynamics of the observer to estimate the states of $x_1(k)$ and $x_2(k)$.

Define the state estimation errors as:

$$\tilde{x}_i(k) = \hat{x}_i(k) - x_i(k), i = 1, 2 \tag{64}$$

Combining (53) through (63), we obtain the estimation error dynamics as

$$\begin{aligned} \tilde{x}_1(k+1) = {} & F_o \tilde{x}_1(k) + (l_2 - R \cdot F_o)\tilde{y}(k) - \Delta x_{1new}(k) \\ & -\Delta F(k) x_1(k) + R\Delta F(k) y(k) - F_o(r_{O_2} + r_{N_2}) - \Delta F(r_{O_2} + r_{N_2}) - d_1(k) \end{aligned} \tag{65}$$

$$\tilde{x}_2(k+1) = F_o \tilde{x}_2(k) + (l_3 - F_o)\tilde{y}(k) - \Delta F(k)(x_2(k) - y(k)) - \Delta x_{2new}(k) - d_2(k) \tag{66}$$

$$\begin{aligned}\tilde{y}(k+1) &= \hat{w}_1^T(k)\phi_1(\hat{z}_1(k)) + l_1\tilde{y}(k) - w_1^T\phi_1(z_1(k)) - \varepsilon_1(k) \\ &= (\hat{w}_1(k) - w_1)^T\phi_1(\hat{z}_1(k)) + w_1^T(\phi_1(\hat{z}_1(k)) - \phi_1(z_1(k))) - \varepsilon_1(k) \\ &= \tilde{w}_1^T(k)\phi_1(\hat{z}_1(k)) + \tilde{w}_1^T(k)\phi_1(\tilde{z}_1(k)) - \varepsilon_1(k) \\ &= \zeta_1(k) + w_1^T\phi_1(\tilde{z}_1(k)) - \varepsilon_1(k)\end{aligned} \tag{67}$$

where

$$\tilde{w}_1(k-1) = \hat{w}_1(k) - w_1, \tag{68}$$

and

$$\zeta_1(k) = \tilde{w}_1^T(k)\phi_1(z_1(k)) \tag{69}$$

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

C. 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 1 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.

Step 1: Virtual controller design.

Define the error between actual and desired air as

$$e_1(k) = x_1(k) - x_{1d} \tag{70}$$

which can be rewritten as

$$\begin{aligned} e_1(k+1) &= x_1(k+1) - x_{1d} \\ &= F(k)[x_1(k) - R.CE(k)x_2(k) + r_{O_2}(k) + r_{N_2}(k)] - x_{1d} \quad + x_{1new}(k) + d_1(k) \end{aligned} \tag{71}$$

For simplicity let us denote

$$f_1(k) = F(k)[x_1(k) + r_{O_2}(k) + r_{N_2}(k)] + x_{1new}(k) - x_{1d} \tag{72}$$

$$g_1(k) = R \cdot F(k)CE(k) \tag{73}$$

Then the system error equation can be expressed as

$$e_1(k+1) = f_1(k) - g_1(k)x_2(k) + d_1(k) \tag{74}$$

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

$$x_{2d}(k) = \frac{f_1(k)}{g_1(k)} \tag{75}$$

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

$$x_{2d}(k) = w_2^T \phi_2(v_2^T x(k)) + \varepsilon_2(x(k)) = w_2^T \phi_2(x(k)) + \varepsilon_2(x(k)) \tag{76}$$

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, *n₂* 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

$$\hat{x}_{2d}(k) = \hat{w}_2^T(k)\phi_2(v_2^T\hat{x}(k)) = \hat{w}_2^T(k)\phi_2(\hat{x}(k)) \tag{77}$$

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

$$\tilde{w}_2(k) = \hat{w}_2(k) - w_2 \tag{78}$$

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

$$e_2(k) = x_2(k) - \hat{x}_{2d}(k) \tag{79}$$

Equation (77) can be expressed using (79) for $x_2(k)$ as

$$e_1(k+1) = f_1(k) - g_1(k)(e_2(k) + \hat{x}_{2d}(k)) + d_1(k), \tag{80}$$

or equivalently

$$\begin{aligned} e_1(k+1) &= f_1(k) - g_1(k)(e_2(k) + x_{2d}(k) - x_{2d}(k) + x_{2d}(k)) + d_1(k) \\ &= f_1(k) - g_1(k)(e_2(k) + x_{2d}(k) - x_{2d}(k) + \hat{x}_{2d}(k)) + d_1(k) \\ &= -g_1(k)(e_2(k) - x_{2d}(k) + \hat{x}_{2d}(k)) + d_1(k) \\ &= -g_1(k)(e_2(k) + \hat{w}_2^T(k)\phi_2(\hat{x}(k)) - w_2^T\phi_2(x(k)) - \varepsilon_2(x(k))) + d_1(k) \end{aligned} \tag{81}$$

Similar to (77), (81) can be further expressed as

$$e_1(k+1) = -g_1(k)(e_2(k) - \zeta_2(k) + w_2^T\phi_2(\tilde{x}(k)) - \varepsilon_2(x(k))) + d_1(k) \tag{82}$$

where

$$\zeta_2(k) = \tilde{w}_2^T(k)\phi_2(\hat{x}(k)) \tag{83}$$

$$w_2^T\phi_2(\tilde{x}(k)) = w_2^T(\phi_2(\hat{x}(k)) - \phi_2(x(k))) \tag{84}$$

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

Rewriting the error $e_2(k)$ from (79) as

$$\begin{aligned} e_2(k+1) &= x_2(k+1) - \tilde{x}_{2d}(k+1) \\ &= (1-CE(k))F(k)x_2(k) + (MF(k)+u(k)) - \hat{x}_{2d}(k+1) + d_2(k) \end{aligned} \tag{85}$$

For simplicity, let us denote,

$$x_2(k+1) = F(k)(1-CE(k))x_2(k) + x_{2new}(k) \tag{86}$$

Equation (85) can be written as

$$e_2(k+1) = f_2(k) + u(k) - \hat{x}_{2d}(k+1) + d_2(k) \tag{87}$$

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 (75) and (77), it can be clear that $\hat{x}_{2d}(k+1)$ is a smooth nonlinear function of the state $x(k)$ and virtual control input $\hat{x}_{2d}(k+1)$. Another NN can be used to approximate the value of $\hat{x}_{2d}(k+1)$ by viewing it as a first order predictor since the proposed NNs use a semi-recurrent architecture which makes them dynamic NNs. Other methods via filtering approach [Lewis et al. 2002] 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

$$u_d(k) = (-f_2(k) + \hat{x}_{2d}(k+1)) = w_3^T \phi_3(v_3^T z_3(k)) + \varepsilon_3(z_3(k)) = w_3^T \phi_3(z_3(k)) + \varepsilon_3(z_3(k)) \tag{88}$$

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 (77). 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

$$z_3(k) = [x(k), \hat{x}_{2d}(k)]^T \in R^3 \tag{89}$$

and

$$\hat{z}_3(k) = [\hat{x}(k), \hat{x}_{2d}(k)]^T \in R^3 \tag{90}$$

Now define

$$\hat{e}_1(k) = \hat{x}_1(k) - x_{1d}, \tag{91}$$

and

$$\hat{e}_2(k) = \hat{x}_2(k) - x_{2d}, \tag{92}$$

The actual control input is now selected as

$$u(k) = \hat{w}_3^T(k)\phi_3(v_c^T \hat{z}_3(k)) + l_4\hat{e}_2(k) = \hat{w}_3^T(k)\phi_3(\hat{z}_3(k)) + l_4\hat{e}_2(k) \tag{93}$$

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 (61), combining (77), (76) with (93) yields

$$e_2(k+1) = l_4\hat{e}_2(k) + \xi_3(k) + w_3^T\phi_3(\tilde{z}(k)) - \varepsilon_3(z_3(k)) + d_2(k) \tag{94}$$

where

$$\tilde{w}_3(k) = \hat{w}_3(k) - w_3 \tag{95}$$

$$\xi_3(k) = \tilde{w}_3^T(k)\phi_3(\hat{z}_3(k)) \tag{96}$$

and

$$w_3^T\phi_3(\tilde{z}(k)) = w_3^T(\phi_3(\hat{z}_3(k)) - \phi_3(z_3(k))) \tag{97}$$

Equations (82) and (94) represent the closed-loop error dynamics. It is required to show that the estimation error (61) and (64), the system errors (82) and (94) and the NN weight matrices $\hat{w}_1(k)$, $\hat{w}_2(k)$, and $\hat{w}_3(k)$ are bounded. Fig. 7 shows the block diagram of the final structure of the designed neuro-controller.

Assumption 4 (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

$$\|w_1\| \le \|w_{1m}\|,\ \|w_2\| \le \|w_{2m}\|,\ \|w_3\| \le \|w_{3m}\|, \tag{98}$$

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

$$\left\|\phi_i(\cdot)\right\| \le \phi_{im}, i = 1, 2, 3 \tag{99}$$

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

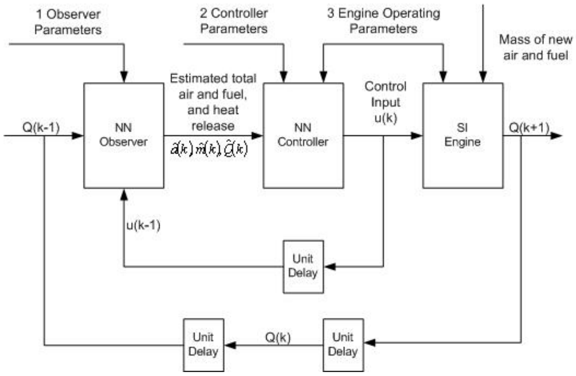

Fig.7. Neuro-controller structure.

*Assumption 5 (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 5.1: Consider the system given in (3) – (5) and let the Assumptions 4 and 5 hold. Let the unknown disturbances be bounded by $\left|d_1(k)\right| \le d_{1m}$ and $\left|d_2(k)\right| \le d_{2m}$, respectively. Let the observer NN weight tuning be given by

$$\hat{w}_1(k+1) = \hat{w}_1(k) - \alpha_1\phi_1(\hat{z}_1(k))(\hat{w}_1^T(k)\phi_1(\hat{z}_1(k) + l_5\tilde{y}(k)) \tag{100}$$

with the virtual control NN weight tuning be provided by

$$\hat{w}_2(k+1) = \hat{w}_2(k) - \alpha_2\phi_2(\hat{x}(k))(\hat{w}_2^T(k)\phi_2(\hat{x}(k) + l_6\hat{e}_1(k)) \tag{101}$$

and the control NN weight tuning be provided by

$$\hat{w}_3(k+1) = \hat{w}_3(k) - \alpha_3 \phi_3(\hat{z}_3(k))(\hat{w}_3^T(k)\phi_3(\hat{z}_3(k) + l_7 \hat{e}_2(k)) \qquad (102)$$

where $\alpha_1 \in R, \alpha_2 \in R, \alpha_3 \in R$ and $l_5 \in R, l_6 \in R, l_7 \in R$ are design parameters. Let the system observer be given by (60), (62) and (63), virtual and actual control inputs be defined as (77) and (93), respectively. The estimation errors (65) through (67), the tracking errors (82) and (94), 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

$$\text{(a)} \;\; 0 < \alpha_i \left\| \phi_i(k) \right\|^2 < 1, \quad i = 1, 2, 3 \qquad (103)$$

$$\text{(b)} \;\; l_3^2 < 1 - \frac{(l_1 - R \cdot F_0)^2}{6R^2 \cdot \Delta F_m{}^2} - \frac{(l_2 - F_0)^2}{6\Delta F_m{}^2} - 4l_5^2 \qquad (104)$$

$$\text{(c)} \;\; l_6^2 < \min\left(\frac{(1 - F_0{}^2)}{18R^2 \cdot \Delta F_m^2}, \frac{1}{18R^2} \right), \qquad (105)$$

$$\text{(d)} \;\; l_4^2 + 6l_7^2 < \min\left(\frac{(1 - F_0{}^2)}{6\Delta F_m^2}, \frac{1}{3} \right), \qquad (106)$$

Remark: For general nonlinear discrete-time systems, the design parameters can be selected using a priori values. Given specific values of R, F_0 and ΔF_m, the design parameters can be derived as $l_i, i = 1, 2, \ldots, 7$. For instance, given $R = 14.6$, $F_0 = 0.14$, and $\Delta F_m = 0.02$, 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 (103) – (106).

VI. Experimental Results

The experimental setup involves a Cooperative Fuel Research (CFR) engine shown in Fig. 8 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.

Fig. 8. Cooperative fuel research (CFR) engine.

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 communications are completed. The algorithm designed uses

15 neurons to approximate the output, though it was observed that even 100 controller nodes and 100 observer nodes calculations are complete within 1.2 milliseconds, well within the available time of 17.667 milliseconds.

The control input is an adjustment to the nominal fuel required at a given equivalence ratio. Fuel injection is controlled by a TTL signal to a fuel injector driver circuit. Pressure measurements come from a charge amplifier which receives pressure transducer signals from a piezoelectric transducer located inside the cylinder.

Fig. 9. Octagon Systems PC770 single board computer.

An engine-to-PC interface board was designed as shown in Fig. 9 to manage the shaft encoder signals, pressure measurements, and fuel injector signal since timing is crucial to correct engine operation. The board uses a microcontroller to communicate between the TTL and analog signals of the engine hardware and a parallel digital I/O port of the PC. A high speed 8-bit A/D converts the pressure measurements. Pressure measurements are sent to the PC where heat release is calculated before being sent to the controller. Fuel pulse width is sent to the microcontroller from the PC.

The controller algorithm and data structures are implemented in C and compiled to run on an x86 PC. The controller was compiled using the same structure and parameters as for simulation. Configuration files allow the controller parameters to be modified without recompiling.

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 one, 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.

$$\%EGR = 100 \times \left(\frac{m_{EGR}}{m_f + m_a + m_{EGR}} \right) \tag{107}$$

Heat release time series and return maps were generated for both controlled and uncontrolled cases for each of several EGR set points: 0%, 5%, 10% and so on. Before the control is applied, 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°.

NN weight values are all initialized at zero. Heat release return maps in Figs. 10 and 11 depict the performance of the proposed NN controller for the 10% EGR case. It is important to observe that the return maps of heat release with no control is slightly below the target value whereas with the application of control the heat release return maps is around the target value. Moreover, misfires are minimized. The return maps at 10% EGR show distinct cyclic dispersion during no control and a significant decrease in those dispersed data points during control.

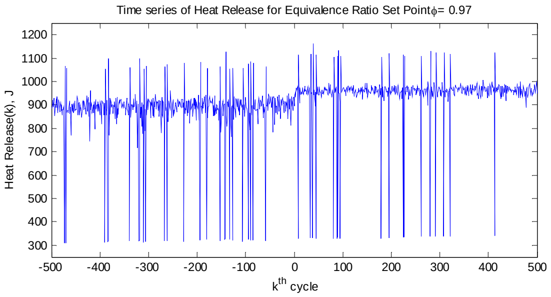

Fig. 10. Heat release time series at 10% EGR.

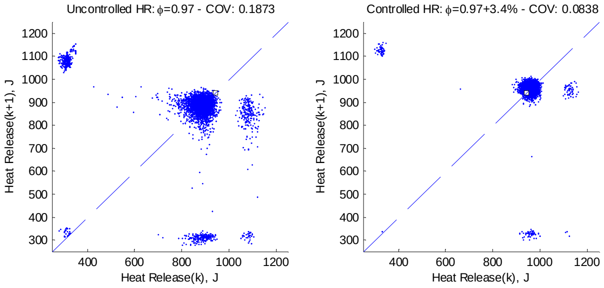

Fig. 11. Uncontrolled and controlled heat release return maps plotting current cycle y(k) against next cycle y(k+1) at 10% EGR.

It can be seen that the mean heat release increases with control, which corresponds to a slightly higher equivalence ratio. The equivalence ratio for EGR operation is intended to be held fixed at one. When using fuel as the control input, the controller must change the fuel to affect the engine, which 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 one. 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.

The dynamics exhibited in the experimental results are quite similar to the simulation results. As mentioned earlier, the higher level of dispersion with higher EGR levels seen in the experimental results is attributable to complex engine processes that are not taken into account by the simplified model. The model is a simplified representation of the engine processes for the purpose of real-time control. Scaling and measurement uncertainties also contribute to the difference. Therefore, at 10% EGR levels, misfires are still noted with control in comparison with simulation results.

The coefficient of variation, COV, in integrated cycle work is often used to establish variability in engine output. With the integrated cycle work obtained from the cyclic cylinder pressure-volume results, the COV is obtained by dividing the standard deviation in cycle work by the mean over all of the cycles observed. The COV for all of the EGR return maps is listed in Table I. 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 I. Coefficient of Variation of the Return Maps

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

Results from the engine-out emission data at the chosen set points show a sharp reduction in NOx as EGR increases. However, unburned hydrocarbons appear to increase with higher EGR which can be attributed to an increase in misfires at high EGR levels. Experimental results indicate an 80% drop of NO_x at stoichiometric levels (2153 PPM at 0% EGR to 436 PPM) 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). A 26% increase in unburned hydrocarbons is seen and this is due to the increased number of misfires at 10% EGR leading to more unburned fuel.

VII. Conclusions

A suite of novel NN controller schemes is presented to reduce the cyclic dispersion in heat release at high EGR levels. The first control scheme utilizes both the NN approximation property and a backstepping type approach for maintaining a fixed air to fuel ratio by altering the fuel injected into the cylinder as the control input. The stability analysis of the closed-loop control system was conducted and the boundedness of the closed loop signals was demonstrated. Simulation results by using an experimentally validated model 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. Using the nonlinear backstepping-like controller, the cyclic dispersion could be reduced significantly, resulting in the potential for decreased emissions and improved fuel economy.

The second control scheme uses heat release as the feedback variable and estimates the total air and fuel in the cylinder at any given time. 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. Both in model simulation and engine experimentation the controller leads to a noticeable decrease in cyclic dispersion. Even though this controller was designed for the model heat release output which does not exhibit all the nonlinearities of actual engine heat release, the controller was still able to minimize heat release error. 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 of the SI engine.

Acknowledgments

This work was supported in part by the National Science Foundation under Grants ECCS#0327877, ECCS#0621924, NSF I/UCRC grant on Intelligent Maintenance Systems and Department of Education through GAANN program.

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), 1998, pp. 2811 – 2819.

[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, J. A. Drallmeier, and C. S. Daw, “Nonlinear cycle dynamics in lean spark ignition combustion,” presented at the 27th Symposium (International) of Combustion, 1999.

[9] 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.

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

[11] H. K. Khalil, Nonlinear Systems, 3rd ed., Prentice Hall, (2002).

[12] 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, to appear Dec 2005.

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

[14] 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.

[15] 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.

Appendix

Proof of Theorem 3.1

Define the Lyapunov function candidate

$$J(k) = e_1^2(k)/4g_{1M}^2 + e_2^2(k)/3k_2 g_{2M}^2 + (1/\alpha_1)\tilde{w}_1^T(k)\tilde{w}_1(k) + (1/\alpha_2)\tilde{w}_2^T(k)\tilde{w}_2(k). \qquad \text{(A.1)}$$

where g_{1M} and g_{2M} are the upper bounds for $g_1(k)$ and $g_2(k)$ given a compact set (see Assumption 1), and k_2, α_1 and α_2 are design parameters (see Theorem 3.1). The first difference of the Lyapunov function is given by

$$\Delta J(k) = \Delta J_1(k) + \Delta J_2(k) + \Delta J_3(k) + \Delta J_4(k). \tag{A.2}$$

The first term $\Delta J_1(k)$ is obtained using (23) as

$$\begin{aligned}
\Delta J_1(k) &= \left(1/4g_{1M}^2\right)\left(e_1^2(k+1) - e_1^2(k)\right) \\
&= \left(1/4g_{1M}^2\right)\left(\left(g_1(k)\left(k_1 e_1(k) + e_2(k) + \zeta_1(k) + d_1(k)\right)\right)^2 - e_1^2(k)\right) \\
&\le \left(k_1^2 - 1/4g_{1M}^2\right)e_1^2(k) + e_2^2(k) + \zeta_1^2(k) + d_1^2(k).
\end{aligned} \tag{A.3}$$

Now taking the second term in the first difference (A.2) and substituting (36) yields

$$\begin{aligned}
\Delta J_2(k) &= \left(1/3k_2 g_{2M}^2\right)\left(e_2^2(k+1) - e_2^2(k)\right) \\
&= \left(\left(g_2(k)\left(k_2 e_2(k) + \zeta_2(k) + d_2(k)\right)\right)^2 - e_2^2(k)\right) \\
&\le \left(1/k_2\right)\left(k_2^2 - 1/3g_{2M}^2\right)e_2^2(k) + \zeta_2^2(k)/k_2 + d_2^2(k)/k_2
\end{aligned} \tag{A.4}$$

Taking the third term in (A.2) and substituting the weights updates from (39) and simplifying to get

$$\begin{aligned}
\Delta J_3(k) &= \frac{1}{\alpha_1}\tilde{w}_1^T(k+1)\tilde{w}_1(k+1) - \frac{1}{\alpha_1}\tilde{w}_1^T(k)\tilde{w}_1(k) \\
&= \frac{1}{\alpha_1}\left[\left(I - \alpha_1\phi(k)\phi^T(k)\right)\tilde{w}_1(k) - \alpha_1\phi(k)\left(w_1^T(k)\phi(k) + k_1 e_1(k)\right)\right]^T \times \\
&\left[\left(I - \alpha_1\phi(k)\phi^T(k)\right)\tilde{w}_1(k) - \alpha_1\phi(k)\left(w_1^T(k)\phi(k) + k_1 e_1(k)\right)\right] - \frac{1}{\alpha_1}\tilde{w}_1^T(k)\tilde{w}_1(k) \\
&= -\left(2 - \alpha_1\phi^T(k)\phi(k)\right)\zeta_1^2(k) - 2\left(1 - \alpha_1\phi^T(k)\phi(k)\right)\left(w_1^T\phi(k) + k_1 e_1(k)\right)\zeta_1(k) \\
&+ \alpha_1\phi^T(k)\phi(k)\left(w_1^T\phi(k) + k_1 e_1(k)\right)^2
\end{aligned} \tag{A.5}$$

Taking the forth term in (A.1) and substituting the weights updates from (40) and simplifying to get

$$\begin{aligned}
\Delta J_4(k) &= \frac{1}{\alpha_2}\tilde{w}_2^T(k+1)\tilde{w}_2(k+1) - \frac{1}{\alpha_2}\tilde{w}_2^T(k)\tilde{w}_2(k) \\
&= \frac{1}{\alpha_2}\left[\left(I - \frac{\alpha_2}{k_2}\sigma(k)\sigma^T(k)\right)\tilde{w}_2(k) - \frac{\alpha_2}{k_2}\sigma(k)\left(w_2^T\sigma(k)+k_2e_2(k)\right)\right]^T \times \\
&\left[\left(I - \frac{\alpha_2}{k_2}\sigma(k)\sigma^T(k)\right)\tilde{w}_2(k) - \frac{\alpha_2}{k_2}\sigma(k)\left(w_2^T\sigma(k)+k_2e_2(k)\right)\right] - \frac{1}{\alpha_2}\tilde{w}_2^T(k)\tilde{w}_2(k) \\
&= -(1/k_2)\left(2 - \frac{\alpha_2}{k_2}\sigma^T(k)\sigma(k)\right)\zeta_2^2(k) \\
&-2(1/k_2)\left(1 - \frac{\alpha_2}{k_2}\sigma^T(k)\sigma(k)\right)\left(w_2^T\sigma(k)+k_2e_2(k)\right)\zeta_2(k) \\
&+\frac{\alpha_2}{k_2^2}\sigma^T(k)\sigma(k)\left(w_2^T\sigma(k)+k_2e_2(k)\right)^2
\end{aligned} \tag{A.6}$$

Combining (A.3) through (A.6) to get the first difference of (A.2) and simplifying to get

$$\begin{aligned}
\Delta J(k) &= \Delta J_1(k) + \Delta J_2(k) + \Delta J_3(k) + \Delta J_4(k) \\
&\le \left(k_1^2 - \frac{1}{4g_{1M}^2}\right)e_1^2(k) + \frac{1}{k_2}\left(k_2^2 + k_2 - \frac{1}{3g_{2M}^2}\right)e_2^2(k) \\
&-\left(1-\alpha_1\phi^T(k)\phi(k)\right)\left(\zeta_1^2(k) + 2\left(w_1^T(k)\phi(k)\right)\zeta_1(k)\right) \\
&-\frac{1}{k_2}\left(1 - \frac{\alpha_2}{k_2}\sigma^T(k)\sigma(k)\right)\left(\zeta_2^2(k) + 2\left(w_2^T\sigma(k)+k_2e_2(k)\right)\zeta_2(k)\right) \\
&+\alpha_1\phi^T(k)\phi(k)\left(w_1^T\phi(k)+k_1e_1(k)\right)^2 \\
&+\frac{\alpha_2}{k_2^2}\left(w_2^T\sigma(k)+k_2e_2(k)\right)^2\sigma^T(k)\sigma(k) + d_1^2(k) + \frac{d_2^2(k)}{k_2}
\end{aligned} \tag{A.7}$$

The above (A.7) can be expressed in a more compact form as

$$\begin{aligned}\Delta J(k) \le {} & \left(3k_1^2 - 1/4g_{1M}^2\right)e_1^2(k) + \left(1/k_2\right)\left(3k_2^2 + k_2 - 1/3g_{2M}^2\right)e_2^2(k) \\ & -\left(1-\alpha_1\phi^T(k)\phi(k)\right)\left(\zeta_1(k) + w_1^T\phi(k) + k_1e_1(k)\right)^2 \\ & -\left(1/k_2\right)\left(1 - \frac{\alpha_2}{k_2}\sigma^T(k)\sigma(k)\right)\left(\zeta_2(k) + w_2^T\sigma(k) + k_2e_2(k)\right)^2 \\ & + d_1^2(k) + d_2^2(k)/k_2 + 2w_{1\max}^2\phi_{\max}^2 + w_{2\max}^2\sigma_{\max}^2/k_2\end{aligned} \tag{A.8}$$

This implies that $\Delta J < 0$ as long as (38) through (41) hold and

$$|e_1(k)| > \frac{2D_M}{\sqrt{1-12k_1^2g_{1M}^2}}\,, \tag{A.9}$$

or

$$|e_2(k)| > \frac{\sqrt{3k_2}\,g_{2M}D_M}{\sqrt{\left(1-3k_2g_{2M}^2 - 9k_2^2g_{2M}^2\right)}}\,, \tag{A.10}$$

or

$$\|\zeta_1(k)\| > \frac{D_M}{\sqrt{\left(1-\alpha_1\|\phi(k)\|^2\right)}} + \frac{2k_1D_M}{\sqrt{1-12k_1^2g_{1M}^2}} + w_{1\max}\phi_{\max}\,, \tag{A.11}$$

or

$$\|\zeta_2(k)\| > \frac{k_2D_M}{\left(k_2-\alpha_2\|\sigma(k)\|^2\right)} + \frac{\sqrt{3k_2}\,k_2g_{2M}D_M}{\sqrt{\left(1-3k_2g_{2M}^2 - 9k_2^2g_{2M}^2\right)}} + w_{2\max}\sigma_{\max}\,, \tag{A.12}$$

where

$$D_M^2 = d_1^2(k) + d_2^2(k)/k_2 + 2w_{1\max}^2\phi_{\max}^2 + w_{2\max}^2\sigma_{\max}^2/k_2\,. \tag{A.13}$$

According to a standard Lyapunov extension theorem (Lewis et al. 1999), this demonstrates that the system errors and the errors in weight estimates are bounded. The boundedness of $\|\zeta_1(k)\|$ and $\|\zeta_2(k)\|$ implies that $\|\tilde{w}_1(k)\|$ and $\|\tilde{w}_2(k)\|$ are bounded, or equivalently the weight estimates $\hat{w}_1(k)$ and $\hat{w}_2(k)$ are bounded.

From (A.8) and (A.9), $e_1(k)$ and $e_2(k)$ are bounded. Using (19) and (29), $x_1(k)$ and $x_2(k)$ approach to X_{1d} and $\hat{x}_{2d}(k)$, respectively. The objective of our control scheme is to bound $x_1(k)$ and $x_2(k)$ close to their respective targets X_{1d} and X_{2d}. Then the equivalent ratio ($\phi(k) = \frac{1}{R}\frac{x_2(k)}{x_1(k)}$) will be close to the desired equivalence ratio ($\phi_d = \frac{1}{R}\frac{X_{1d}}{X_{2d}}$), and the combustion efficiency $CE(k)$ is held constant ($CE(k)$ is the function of $\phi(k)$ alone, see equation (12)). Since heat release $Q(k) = x_2(k)CE(k)$, the heat release is bounded and its variations are reduced provided the bounds are small. Since the bounds are a function of design parameters k_1, k_2, α_1 and α_2 and by suitably selecting these, the variations can be reduced. Consequently, it has been shown that $x_1(k)$ is bounded close to X_{1d} and $x_2(k)$ is bounded close to $\hat{x}_{2d}(k)$. In order to prove $x_2(k)$ is bounded close to X_{2d}, the difference between $\hat{x}_{2d}(k)$ and $x_{2d}(k)$ has to be considered as

$$\hat{x}_{2d}(k) - x_{2d}(k) = (\hat{w}_1(k) - w_1(k))^T \phi(k) - \varepsilon_1(k) = \zeta_1(k) - \varepsilon_1(k). \tag{A.14}$$

Since $\zeta_1(k)$ and $\varepsilon_1(k)$ are bounded, $\hat{x}_{2d}(k)$ is bounded close to $x_{2d}(k)$. Since $e_2(k)$ is bounded, and $x_2(k)$ is bounded close to $\hat{x}_{2d}(k)$, and $\hat{x}_{2d}(k)$ is bounded close to $x_{2d}(k)$, it can be concluded that $x_2(k)$ is bounded close to $x_{2d}(k)$. By suitably selecting X_{1d} according the following equation

$$x_{2d}(k) = \frac{1}{g_1(k)}\left(- f_1(k) + X_{1d}\right) + k_1 e_1(k) = X_{2d} + \delta(k) = R \times \phi_d \times X_{1d} + \delta(k). \qquad \text{(A.15)}$$

where $\delta(k)$ is a small and known bounded value and $x_{2d}(k)$ is forced to be close to X_{2d} at steady state. Then it follows that $x_2(k)$ is bounded close to X_{2d}. Since both $x_1(k)$ and $x_2(k)$ are bounded close to their targets X_{1d} and X_{2d}, respectively, the equivalence ratio is bounded close to its desired value. Consequently, the combustion efficiency, $CE(k)$, is close to its desired value and the heat release ($Q(k) = x_2(k)CE(k)$) is close to its target and the heat release dispersion is reduced.

PAPER 4

Neural Network Controller Development and Implementation for Spark Ignition Engines with High EGR Levels

J. Vance[1], A. Singh[1], B. Kaul[2], S. Jagannathan[1], Sr. Member, IEEE and J. Drallmeier[2]

[1]Department of Electrical and Computer Engineering

[2]Department of Mechanical and Aerospace Engineering

Abstract — Past 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 preventing commercial engines to operate with high EGR levels. A neural network (NN)-based output feedback controller is developed to reduce cyclic variation in the heat release under 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 stability analysis of the closed loop system is given and the boundedness of the control input is demonstrated by relaxing separation principle, persistency of excitation condition, certainty equivalence principle and linear in the unknown parameter assumptions. Online training is used for the adaptive NN and no offline training phase is needed. This online learning feature and model-free approach is used to demonstrate the applicability of the controller on a different engine with minimal effort.

Simulation results demonstrate that the cyclic dispersion is reduced significantly using the proposed controller when implemented on an engine model that has been validated experimentally. For a single cylinder research engine fitted with a modern four valve head (Ricardo engine), experimental results at 15% EGR indicate that cyclic dispersion was reduced 33% by the controller, an improvement of fuel efficiency by 2%, and a 90% drop in NO_x from stoichiometric operation without EGR was observed. Moreover, unburned hydrocarbons drop by 6% due to NN control as compared to the uncontrolled scenario due to the drop in cyclic dispersion. Similar performance was observed with the controller on a different engine.

I. Nomenclature

CFR	Cooperative Fuel Research
COV	coefficient of variation
IMEP	Mean effective pressure, $Work/Disp.Volume$
NO_x	Nitrogen oxide compounds
uHC	Unburned hydrocarbons
$CE(k)$	Combustion efficiency
$d_1(k)$	Unknown disturbance in air
$d_2(k)$	Unknown disturbance in fuel
$F(k)$	Fraction of unreacted gas and fuel remaining from previous cycle
$r_{H_2O}(k)$	Mass of water
$r_{O_2}(k)$	Mass of oxygen
$r_{N_2}(k)$	Mass of nitrogen
$r_{CO_2}(k)$	Mass of carbon dioxide
R	Stoichiometric air-fuel mass ratio
$u(k)$	Mass change fuel input
$x_1(k)$	Mass of air
$x_2(k)$	Mass of fuel
$x_3(k)$	Mass of EGR
$\varphi(k)$	Equivalence ratio

φ_l, φ_u Lower 10 and upper 90 percent locations of the combustion efficiency function

φ_m Midpoint between φ_l and φ_u

II. Introduction

One of the most interesting challenges facing the automotive industry today is the development of energy generation techniques that have a low impact on the environment. 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). Global warming and its impact on the environment have shifted the focus of the automotive industry. Current efforts are directed at reducing the total amount of emissions and fuel consumption. The engine control system can be classified into three categories [1]: the spark advance (SA) control, the air-fuel ratio (A/F) control and the exhaust gas recirculation (EGR) control. Partial recirculation of exhaust gases, a technique introduced in the early 70's, has continued to receive attention [2].

Operating a spark ignition engine with EGR can reduce the NO_X as well as improve the fuel efficiency. 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. Additionally, improved brake specific fuel consumption with EGR dilution is a result of reduced pumping work, reduced heat transfer to the walls due to decreased burned gas temperature, and to a lesser extent, a reduction in dissociation at high temperatures in the burned gases. EGR dilution has the advantage over lean combustion of maintaining stoichiometric operation so that current three-way catalyst technology can be used. These advantages which come with dilute engine operation are the primary motivation of this work.

However, increased dilution of the intake charge through EGR also reduces the combustion rate, which makes stable combustion [2, 4-6] 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. Therefore, commercial engines do not operate with high levels of EGR due to cyclic dispersion.

Several researchers [3, 7-9] have studied lean combustion engine control technology but few results have been reported for the EGR case. Investigation of the onset of complex dynamic behavior in a SI engine with high levels of simulated EGR (added nitrogen) as compared to the lean equivalence ratio case has demonstrated a bifurcation phenomenon similar to when the engine was operating under lean conditions [2]. Therefore, it is envisioned that by applying neural network (NN) controller similar to that of lean operation, the cyclic dispersion resulting from high levels of EGR dilution can be minimized, increasing the engine's EGR tolerance, potentially further reducing engine out NO_X and unburned HC while improving fuel efficiency.

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 on a per-cycle basis which necessitates the development of output feedback control schemes.

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

In this paper, a direct adaptive NN controller is proposed for stable operation of the SI engine under high levels of EGR. The SI engine dynamics is modeled as a nonlinear discrete-time system in *nonstrict feedback form* [16]. Neural networks are employed to learn the unknown nonlinear dynamics since the residual gas and combustion efficiency are unknown. A backstepping approach [14-15] in discrete-time is used to design the control input (injected fuel) to the total fuel system. The total fuel is then treated as the virtual control signal to the air system so that both the air and fuel states are bounded tightly to their respective targets. A separate control loop is designed for maintaining EGR levels. Consequently, the cyclic dispersion is reduced and the engine is stable even when an exact knowledge of engine dynamics is not known to the controller making the NN controller model-free.

This stability permits higher levels of diluents to be considered for a specific engine, further enhancing NO_X reduction and fuel efficiency than would be realized on an uncontrolled engine. The stability analysis of the closed-loop control system is given and the boundedness of the closed loop signals is shown since a stable open loop system can still become unstable with a controller. The NN weights are tuned on-line, with no offline learning phase required. Moreover, separation principle, persistency of excitation condition, certainty equivalence and linear in the unknown parameters assumptions are relaxed. Performance of the NN controller is evaluated on different engines and results show satisfactory performance of the controller.

III. Engine as a Nonlinear Discrete-time System

A. Non-strict Nonlinear System Description

Consider the following non-strict feedback nonlinear system described by the following equations

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

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

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

Equations (1) and (2) represent a discrete-time nonlinear system in nonstrict 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. Controller results from strict feedback nonlinear systems cannot be directly extended to nonstrict feedback nonlinear systems due to non causal controller design issues. Next the engine dynamics is presented in the nonstrict feedback form and subsequently the NN controller development is introduced. The dynamic NN architecture acts as a one-step predictor overcoming the non causal design. The SI engine dynamic model is discussed next.

B. Engine Dynamics

Daw, Finney, Green, Kennel and Thomas (1996) [4] and Daw et al. (1998) [5] 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. Actual variations in parameters due to complex processes like temperature and pressure effects, turbulence, fuel vaporization, etc. are not directly calculated, but modeled as stochastic effects through random noise on parameters such as injected air-fuel ratio and residual fraction. The model for the EGR case is shown below.

$$\begin{aligned} x_1(k+1) = {} & F(k)[x_1(k) - R.CE(k)x_2\left(k\right) + r_{O_2}(k) + r_{N_2}\left(k\right)] \\ & + x_{1new}(k) + d_1(k) \end{aligned} \tag{3}$$

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

$$\begin{aligned} x_3(k+1) = {} & F(k)(r_{CO_2}(k) + r_{H_2O}(k) + r_{N_2}(k) + x_3(k) \\ & + EGR(k)) \end{aligned} \tag{5}$$

$$y(k) = x_2(k)CE(k) \tag{6}$$

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

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

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

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

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

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

Equations for $x_1(k),\ x_2(k)$ 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 assumed to be proportional to the mass of fuel burned, which is given by $y(k)$. The term $CE(k)$ is defined as the combustion efficiency, which is bounded above as $0<CE_{\min}<CE(k)<CE_{\max}$ where $CE_{\max}$ is the maximum combustion efficiency denoted here as a constant. The term $F(k)$ is the residual gas fraction, which is bounded as $0<F_{\min}<F(k)<F_{\max}$ whereas R is the stoichiometric air-fuel ratio, which is given by ≈15.13 for iso-octane. The term $u(k)$ is the small change in fuel per cycle and $\varphi(k)$ is the equivalence ratio. Additionally, $\varphi_l,\varphi_u,\varphi_m$ are equivalence ratio system parameters for the lower 10 and upper 90 percent and midpoint locations of the combustion efficiency function. The terms $r_{H_2O}(k),\ r_{O_2}(k),\ r_{N_2}(k)$, 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 left over after this is the residual inert nitrogen. The terms γ, $\gamma_{H_2O,}\gamma_{O_2,}\gamma_{N_2}$, and γ_{CO_2} are constant parameters which are determined from stoichiometry and fuel properties such as the hydrogen/carbon ratio and the molecular weight. The terms $d_1^{'}(k)$ and $d_2^{'}(k)$ are unknown, 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 not measurable.

Remark 1: In (3) through (6), states $x_1(k)$ and $x_2(k)$ are not available for feedback control since they are not measured whereas the output $y(k)$ is available for measurement. The control objective then is to operate the engine with high EGR levels with $y(k)$ as the feedback parameter and without knowing precisely the engine dynamics. It is important to note that the output is a nonlinear function of the states unlike in many papers [10-11] where the output is considered as a linear function of system states.

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

C. Engine Dynamics Using Nominal Values

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

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

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

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 from the literature designed to control EGR levels makes the inert gases evolve into a stable value. One can observe that (5) is a stable system and standard control results [17] can be applied. Therefore, it is sufficient to use (3) and (4) in order to minimize cyclic dispersion and (5) is not included in the proposed controller design.

Consider,

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

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

$$F(k) = F_0(k) + \Delta F(k) \tag{17}$$

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,

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

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

$$0 \le \left|\Delta F(k)\right| \le \Delta F_M \tag{20}$$

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

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

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

It is important to note that the closed-loop stability analysis has to be performed with the proposed NN controller even though many of the engine terms are considered bounded above since a stable open-loop system can still become unstable with a controller unless the NN weight update laws are properly selected. Moreover, a Lyapunov-based stability analysis is needed in order to show the relaxation of the separation principle for the observer and certainty equivalence principle for the controller. Next the NN observer design is introduced.

IV. Neural Network-based observer design

First, a semi-recurrent 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

$$y(k+1) = x_2(k+1)CE(k+1) \tag{23}$$

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

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

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, then the approximation error $\varepsilon_1(z_1(k))$ can be made arbitrarily small over the compact set since the activation functions form a basis according to [13].

For simplicity, we define

$$\phi_1(z_1(k)) = \phi_1(v_1^T z_1(k)) \tag{25}$$

$$\varepsilon_1(k) = \varepsilon_1(z_1(k)) \tag{26}$$

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

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

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

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

where $\hat{y}(k+1)$ is the predicted heat release, $\hat{w}_1(k) \in R^{n_i}$ 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

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

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 in the following form:

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

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

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:

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

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

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

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

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

where

$$\tilde{w}_1(k) = \hat{w}_1(k) - w_1 \tag{36}$$

and

$$\zeta_1(k) = \tilde{w}_1^T(k)\phi_1(\hat{z}_1(k)) \tag{37}$$

and for the purpose of simplicity, $(\phi_1(\hat{z}_1(k)) - \phi_1(z_1(k)))$ is written as $(\phi_1(\tilde{z}_1(k))$. Next the NN controller design is presented and the NN weight updates for both NN observer and controller are discussed.

V. 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}, $y\left(k\right)$ 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

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

For simplicity let us denote

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

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

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} 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 control 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

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

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

$$e_1(k+1) = -g_1(k)(e_2(k) - \zeta_2(k) + w_2^T\phi_2(\tilde{x}(k)) - \varepsilon_2(x(k))) + d_1(k) \tag{50}$$

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

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

For simplicity, let us denote:

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

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 $x\left(k\right)$ and virtual control input $\hat{x}_{2d}(k)$. Another NN can be used to approximate the value of $\hat{x}_{2d}(k+1)$ and to generate a suitable control input by using this value since a second control NN with semi-recurrent architecture can be viewed as a first order predictor. Other methods via filtering approach [18] do exist in the literature in order to obtain this value future value which can subsequently be used by a second control NN.

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 that both $x_1(k)$ and $x_2(k)$ cannot be measured, $z_3\left(k\right)$ 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

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

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.

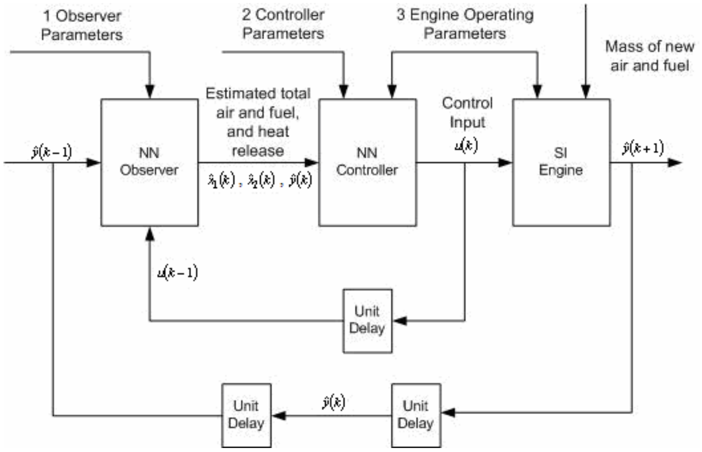

Fig.1. Neuro-controller structure.

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 when 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 $\left|d_1(k)\right| \le d_{1m}$ and $\left|d_2(k)\right| \le d_{2m}$, 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 with the bounds specifically given by (A.17) through (A.24) provided the design parameters are selected as

$$\text{(a)}\ \ 0 < \alpha_i \left\|\phi_i(k)\right\|^2 < 1, \quad i = 1, 2, 3 \tag{71}$$

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

(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)

Proof: See Appendix**.**

Remark 3: For general nonlinear discrete-time systems, the design parameters can be selected using a priori values. Given specific values of *R*, *F₀*, and ΔF_m, the design parameters can be derived as $l_i, i = 1$ to 7. For instance, given R = 14.6, F_0 = 0.14, and Δ F_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: A well-defined controller is developed in this paper since a single NN is utilized to approximate two nonlinear functions thereby avoiding division by zero.

Remark 5: It is important to note that in this theorem there is no persistency of excitation condition (PE) condition for the NN observer and NN controller in contrast with standard work in the discrete-time adaptive control [19] since the first difference of the Lyapunov function in the Appendix does not require the PE condition on input signals to prove the boundedness of the weights. Even though the input to the hidden-layer weight matrix is not updated and only the hidden to the output-layer weight matrix alone is tuned, the NN method relaxes the linear in the unknown parameter assumption. Additionally, certainty equivalence principle is not used in the proof.

Remark 6: Generally, the separation principle used for linear systems does not hold for nonlinear systems and hence it is relaxed in this paper for the controller design since the

Lyapunov function is a quadratic function of system errors and weight estimation errors of the observer and controller NNs.

Remark 7: It is important to notice that the NN outputs are not fed as delayed inputs to the network whereas the outputs of each layer are fed as delayed inputs to the same layer. The NN weight tuning proposed in the (68) through (70) render a semi-recurrent architecture due to the proposed weight tuning law even though feed forward NNs are utilized in the observer and controller. This semi-recurrent NN architecture renders a dynamic NN which is capable of predicting the state one step-ahead overcoming the non causal controller design.

VI. Simulation Results

In an initial phase to test the effectiveness of the control scheme, the Daw model was used to simulate the engine dynamics under high levels of diluent. The model input parameters were calibrated by comparing return maps of heat release to return maps from the single cylinder engines discussed in the next section. This approach, used in prior lean combustion work [9], provided a basis for choosing the nominal input mass of fuel and air as well as the residual gas fraction and stochastic variation. The controller was then applied to the simulation model to investigate the reduction in cyclic variability.

The simulation parameters selected were as follows: An equivalence ratio of one was maintained with stochastic 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 to satisfy (72) through (74). EGR was assumed to be an inert mixture with a molecular weight of 30.4. The residuals are assumed to be a mixture of fuel, air and inert gases. The composition of the residuals is determined based on the stoichiometry of the prior cycle to establish the fraction of the inert from both EGR and combustion. The NNs were designed to have 15 neurons each in the hidden layer with learning rates of 0.01 each so that (71) is satisfied. Here the heat release value is normalized.

The activation functions used were the hyperbolic tangent sigmoid functions. Simulations ran for 1000 cycles of engine operation at each EGR value ranging from 19% to 24%. The attached plots show the results obtained from the simulation runs at EGR levels of 24%. The dispersion of heat release in the return map of Fig. 3 is less than that seen in Fig. 2, according to the lower coefficient of variation which has reduced from 0.0741 to 0.0159.

The coefficient of variation metric – hereafter referred to as COV – is used to quantify cyclic dispersion due to heat release. It is calculated as the standard deviation of a set of heat release data divided by the mean heat release for that set. A larger COV indicates that heat release values were more dispersed on the return map. With regard to COV, a goal for this controller implementation is to observe a reduction in COV when the control loop is closed on the engine.

The coefficient of variation, COV, in integrated cycle work is often used to establish variability in engine output. With the integrated cycle work obtained from the cyclic cylinder pressure-volume results, the COV is obtained by dividing the standard deviation in cycle work by the mean over all of the cycles observed. It was observed that with the NN applied, the engine model exhibits minimal dispersion with high EGR levels even with perturbation on the residual gas fraction being unknown. The reduction in dispersion physically translates into fewer partial burning cycles even with high EGR levels in the engine.

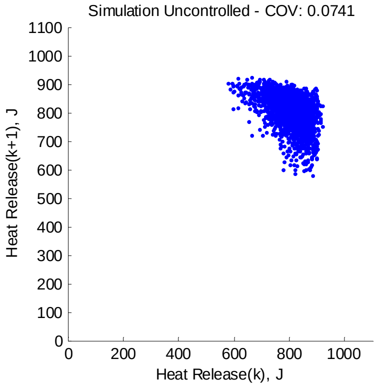

Fig. 2. Heat release return map without control (24% EGR).

Although exhibiting very similar dynamics, the return maps of heat release are quantitatively different between the simplified model used for controller development and testing and the actual engine as presented next. This can be attributed to the fact that the engine model simply considers mass conservation of the fuel, air, and combustion product species and places all complexities of the fluid mechanics and combustion into a phenomenological nonlinear combustion efficiency term and stochastic variations. In spite of this simplicity in the model, the designed controller performs highly satisfactorily on the actual engine as will be seen in the next few sections.

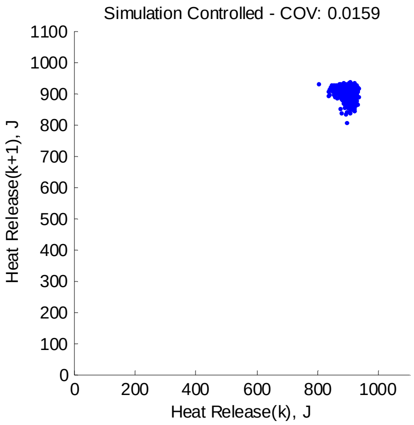

Fig. 3. Heat release return map with control (24% EGR).

VII. Controller Hardware Design

The experimental setup involves two research engines. The first is a Cooperative Fuel Research (CFR) engine and the second is a Ricardo Hydra engine with a modern four valve Ford Zetec head. Both engines are operated at 1000 RPM while multiple load set points are tested through the addition of diluent. Being single cylinder engines, dynamics introduced by multiple cylinders are avoided.

For each engine, 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 at this engine speed. 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 accurate metering of an average EGR flow 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, though it can be seen from Fig. 5 that even at 100 controller nodes and 100 observer nodes calculations are complete within 1.2 milliseconds, well within the available time of 17.667 milliseconds.

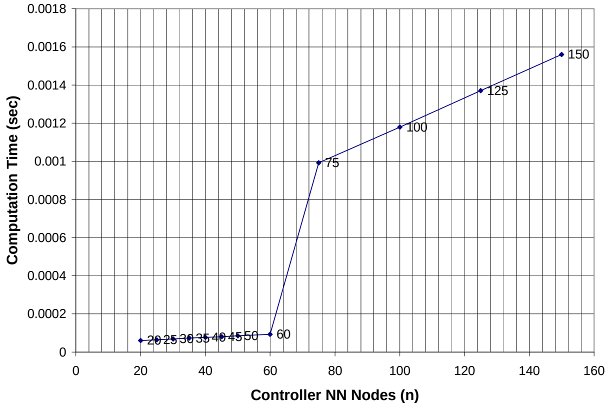

Fig. 4. Neural network controller runtimes varying nodes.

Since the number of nodes required in a multilayer NN for a given approximation error is not clear in the literature, the plot in Fig. 4 illustrates that even with large number of hidden-layer NNs the proposed controller can be implemented on the embedded hardware. However, it was found from offline analysis that the improvement in approximation accuracy is not significant beyond 15 hidden-layer nodes and therefore the hidden-layer NN nodes in the observer and controller are limited to 15.

The control input is an adjustment to the nominal fuel required at a given equivalence ratio. Fuel injection is controlled by a TTL signal to a fuel injector driver circuit. Pressure measurements come from a charge amplifier which receives pressure transducer signals from a piezoelectric transducer located inside the cylinder.

An engine-to-PC interface board was designed to manage the shaft encoder signals, pressure measurements, and fuel injector signal since timing is crucial to correct engine operation. The board uses a microcontroller to communicate between the TTL and analog signals of the engine hardware and a parallel digital I/O port of the PC. A high speed 8-bit A/D converts the pressure measurements. Pressure measurements are sent to the PC where heat release is calculated before being sent to the controller. Fuel pulse width for the next engine cycle is sent to the microcontroller from the PC.

VIII. Experimental Results

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 following equation shows how the mass of nitrogen, m_{EGR}, is chosen to give a desired percentage of EGR.

$$\%EGR = 100 \times \left(\frac{m_{EGR}}{m_f + m_a + m_{EGR}} \right) \tag{75}$$

As mentioned before in the simulation section, COV is a metric that can quantify a reduction in cyclic dispersion when viewing a return map. The following return maps for the two engines on which the controller was operated have COV information. It is shown that with control, the COV is reduced. Again, this reduction in COV means that the engine is more stable in the presence of high intake EGR.

Typically, COV values less than 10% are considered acceptable for production engines. The ideal COV would be 0%. More realistically, the cyclic dispersion of an engine cannot be reduced to less than the case where equivalence ratio is stoichiometric and no EGR is present due to ever present stochastic effects. The experimental results will show that the controller can reduce the cyclic dispersion, measured as a reduction in the COV metric.

Figs. 5 through 11 are results from engine controller tests on the CFR engine. The uncontrolled engine equivalence ratio was near stoichiometric at 0.97. The controller pushed the equivalence ratio to 1.0, due to the behavior of the control input $u(k)$. Equivalence ratios experienced for both uncontrolled and controlled scenarios are near stoichiometric although the control input $u(k)$ slightly modifies the effective equivalence ratio over time.

Heat release time series and return maps were generated for both controlled and uncontrolled cases for each of three EGR set points: 0%, 5%, and 10%. These EGR values correspond to average IMEP load values of 528 kPa, 476 kPa and 410 kPa, respectively. 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°.

NN weight values are all initialized at zero. Heat release return maps in Figs. 5 and 6 depict the performance of the proposed NN controller for the 0% EGR case. It is important to observe that the return maps of heat release with no control is slightly below the target value whereas with the application of control the heat release return maps is around the target value. Moreover, no misfire is noted. Figs. 7 and 8 show a decrease in cyclic dispersion for 5% EGR which corroborates the results seen in simulation. During the absence of control there is much cyclic dispersion and occasional misfires, and during control the misfires and dispersion are reduced.

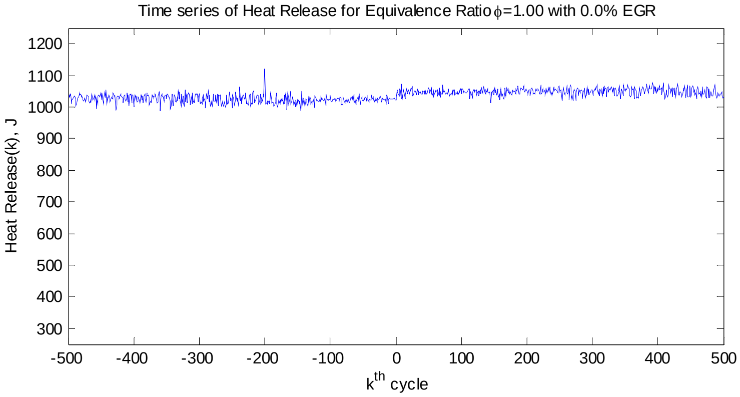

Fig. 5. CFR engine heat release (in joules) time series at 0% EGR.

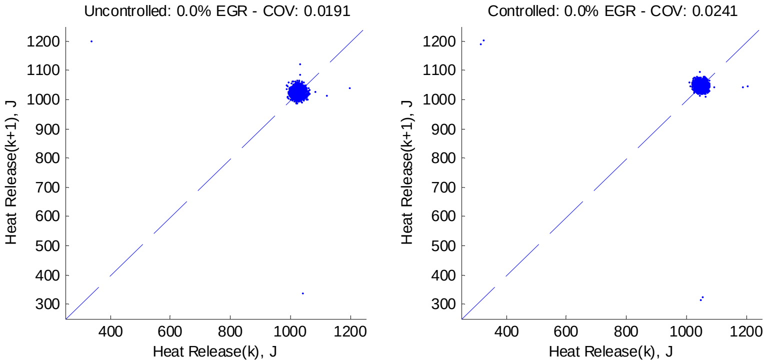

Fig. 6. Uncontrolled and controlled heat release return maps plotting current cycle y(k) against next cycle y(k+1) at 0% EGR on CFR engine.

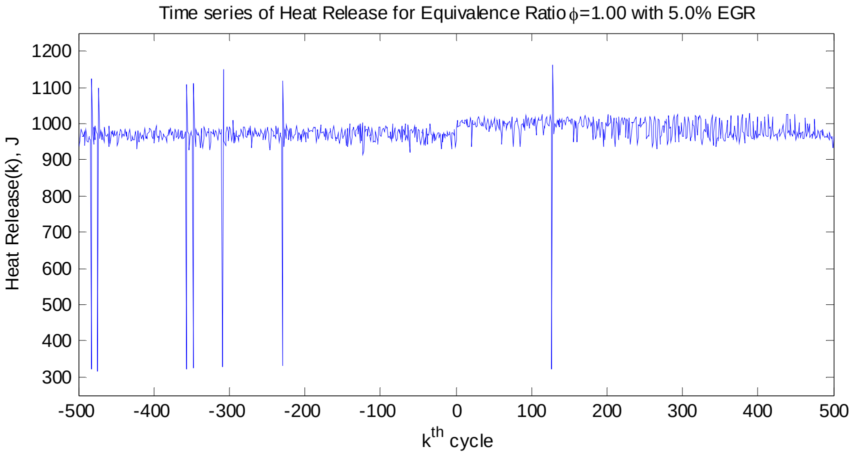

Fig. 7. CFR engine heat release time series at 5% EGR.

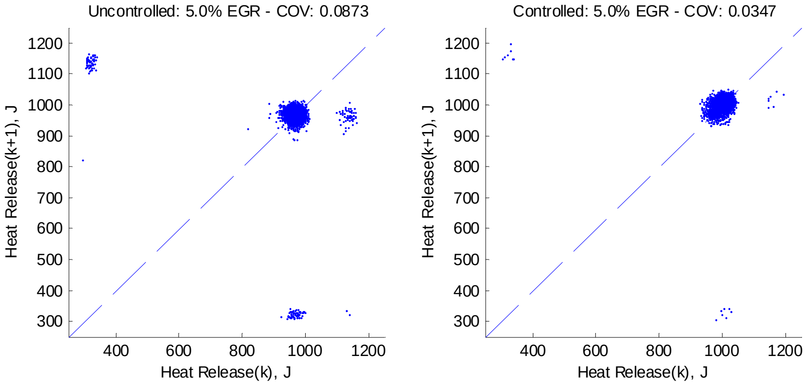

Fig. 8. Uncontrolled and controlled heat release return maps plotting current cycle y(k) against next cycle y(k+1) at 5% EGR on CFR engine.

The return maps at 10% EGR show distinct cyclic dispersion during no control and a significant decrease in those dispersed data points during control.

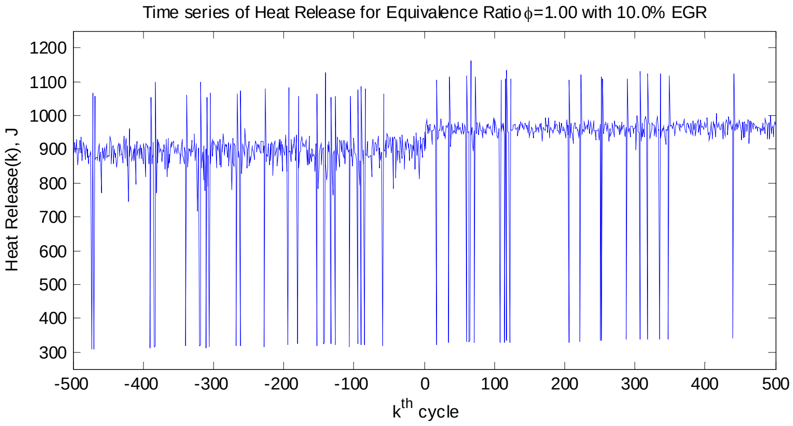

Fig. 9. CFR engine heat release time series at 10% EGR.

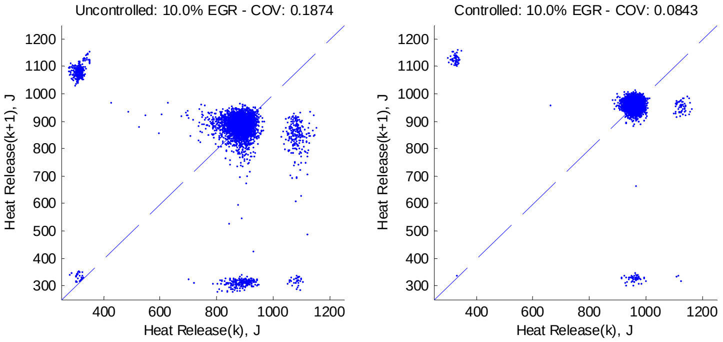

Fig. 10. Uncontrolled and controlled heat release return maps plotting current cycle y(k) against next cycle y(k+1) at 10% EGR on CFR engine.

It can be seen that the mean heat release increases with control, which corresponds to a slightly higher equivalence ratio. The equivalence ratio for EGR operation is intended to be held fixed at 1.00. 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.

The COV for the EGR return maps is listed in Table I for the CFR engine. As the EGR percentage of cylinder contents is increased from 0% to 10%, the coefficient of variation increases for uncontrolled engine operation. The increased coefficient of variation indicates increased cyclic dispersion as seen in the uncontrolled EGR return maps. The coefficient of variation decreases when control is applied in the presence of EGR, a decrease of 55% at 10% EGR, meaning the controller has made the engine more stable. Consequently a 25% drop in unburned hydrocarbons is observed with 10% EGR for this engine. Additionally, 80% drop in NOx from stoichiometric levels is noted.

Table I. Coefficient of Variation for CFR

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

Figs. 11 through 18 are data collected from engine controller operation on the Ricardo research engine. Performance of the controller was similar in that decreases in cyclic dispersion for high EGR cases are seen. Higher EGR is possible with the Ricardo engine because it is a faster-burning engine and hence is more tolerant of diluent addition. The fuel control input does not increase as much on the Ricardo engine as with the CFR engine, so the time series of heat release plots do not exhibit increase when control is activated. Total fuel input during control was not more than 1.5% from the nominal fuel for the desired stoichiometric equivalence ratio.

Aside from the stoichiometric operating condition with no EGR added to the Ricardo engine, three cases of high EGR were tested. EGR levels of 0%, 12.9%, 15.2%, and 18.5% were used to obtain data. These values correspond to IMEP load values of 881 kPa, 716 kPa, 594 kPa, and 297 kPa, respectively.

In Fig. 11 the time series of heat release is plotted, showing the last 500 cycles of the uncontrolled data set with the transition to the first 500 cycles of data with the controller enabled. As seen in Fig. 12, there is very little cyclic dispersion at this stoichiometric case without EGR present, indicated by the low COV value of 2.6%. This set point can be considered optimal in that the engine is operating under ideal conditions. When the controller is enabled, very little improvement can be made to reduce the stochastic dispersion beyond the engine's most stable operating point.

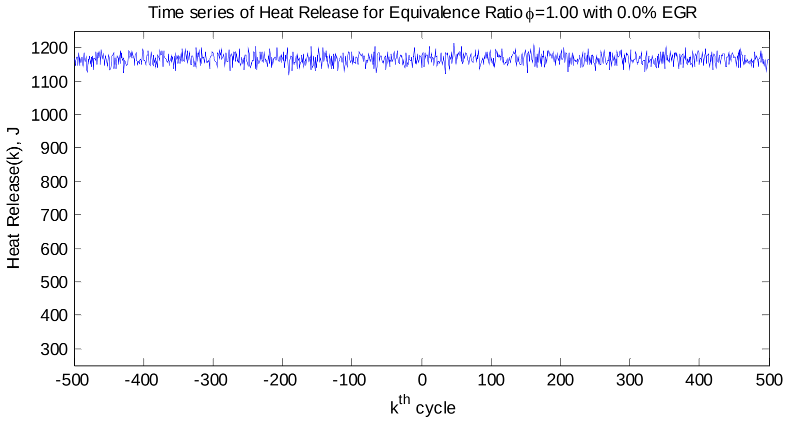

Fig. 11. Ricardo engine heat release time series at 0 % EGR.

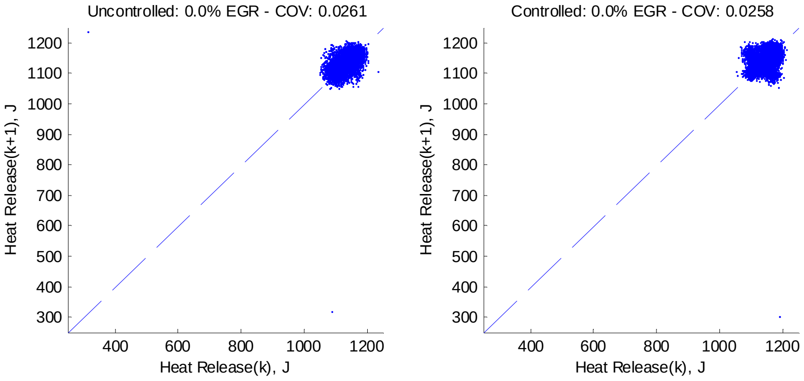

Fig. 12. Uncontrolled and controlled heat release return maps plotting current cycle y(k) against next cycle y(k+1) at 0% EGR on Ricardo.

In Figs. 13 and 14, the effect of about 12.9% EGR causes the engine to become less stable. The controller causes a reduction in cyclic dispersion by comparing the COV values for uncontrolled and controlled data in Fig. 14. The COV falls from 0.0462 to 0.0352.

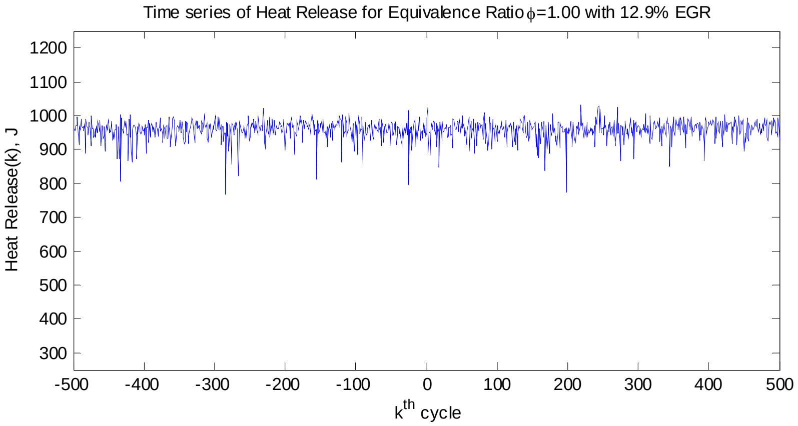

Fig. 13. Ricardo engine heat release time series at 12.9% EGR for Ricardo.

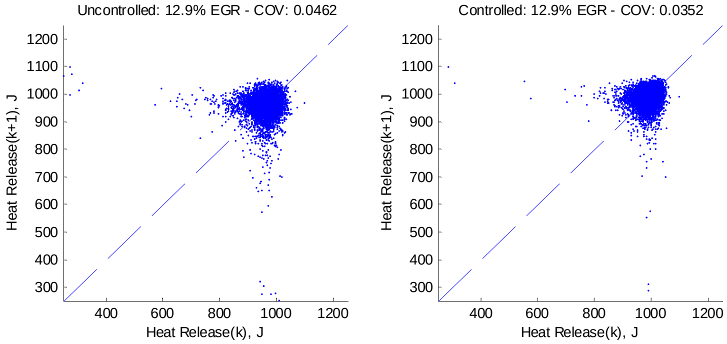

Fig. 14. Uncontrolled and controlled heat release return maps plotting current cycle y(k) against next cycle y(k+1) at 12.9% EGR on Ricardo.

One can see from Figs. 17 and 18 that 15.2% EGR causes the engine to become much more unstable than for the case of 12.9% EGR. The controller yields a significant improvement, reducing the dispersion by 33% from 0.1345 to 0.0891.

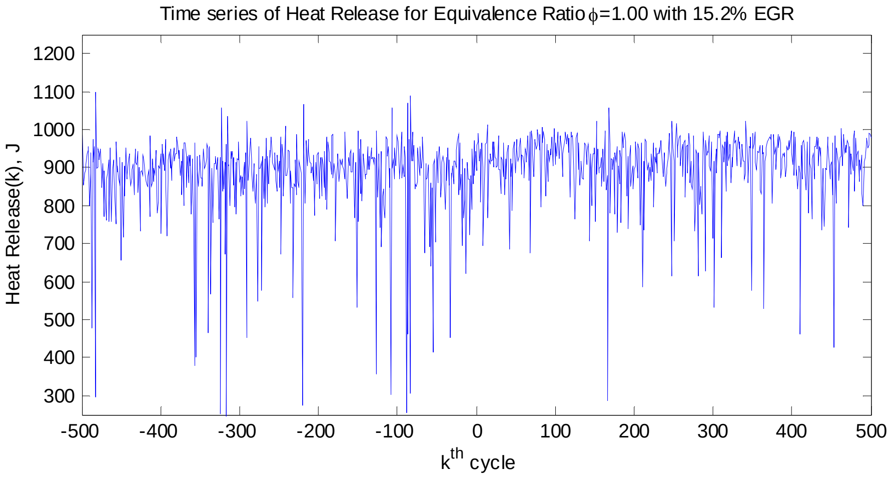

Fig. 15. Ricardo engine heat release time series at 15.2% EGR for Ricardo.

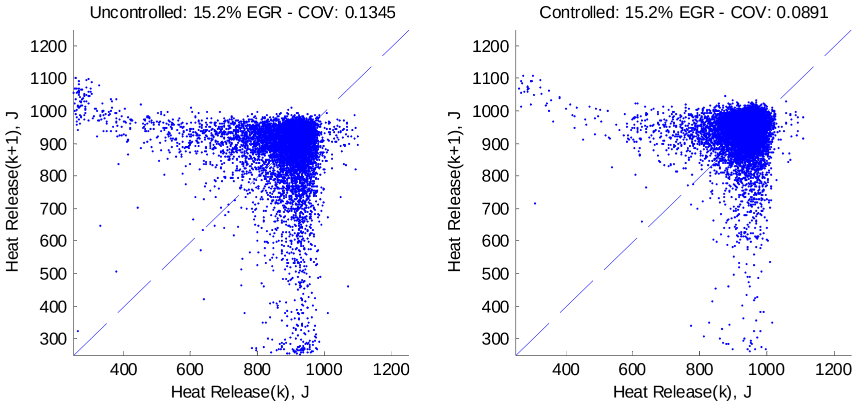

Fig. 16. Uncontrolled and controlled heat release return maps plotting current cycle y(k) against next cycle y(k+1) at 15.2% EGR on Ricardo.

In Figs. 17 and 18, a case of very high EGR is shown for the Ricardo engine. At 18.5% EGR the engine becomes very unstable, exhibiting significant cyclic dispersion of heat release. However, the controller is still able to reduce dispersion from 0.3733 to 0.3419.

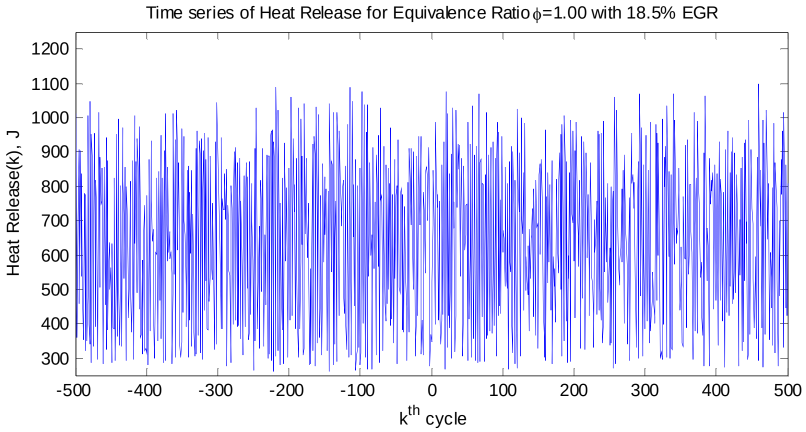

Fig. 17. Ricardo engine heat release time series at 18.5% EGR for Ricardo.

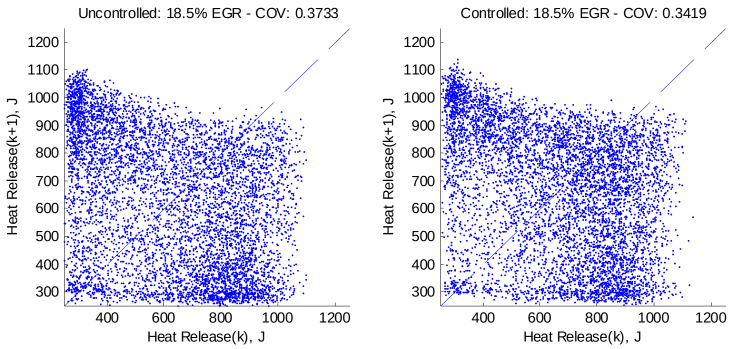

Fig. 18. Uncontrolled and controlled heat release return maps plotting current cycle y(k) against next cycle y(k+1) at 18.5% EGR on Ricardo.

The COV values for the Ricardo engine data are shown in Table II. The controller reduces COV for every case, corresponding to a decrease in cyclic dispersion. Hence, the controller can make the engine more stable in the presence of high EGR.

Table II. Coefficient of Variation for Ricardo

EGR	Uncontrolled	Controlled
	COV	COV
0.0%	0.0261	0.0258
12.9%	0.0462	0.0352
15.2%	0.1345	0.0891
18.5%	0.3733	0.3419

The engine-out emissions of unburned hydrocarbons and NO_x for the Ricardo engine are shown in Table III. Prefixes of ‘u’ and ‘c’ given before the emission type represent uncontrolled and controlled, respectively. For all cases of EGR, unburned hydrocarbons diminish with the controller enabled. This can be expected since the controller is reducing the number of partial-burns encountered in the heat release. Table data for NO_x shows a significant reduction at high levels of EGR. The controller makes engine operation more stable which increases the mean heat release. The higher burn temperatures within the cylinder during control cause the NO_x to slightly increase when the controller is activated compared to without control. At 15% EGR, a drop of 90% NO_x from stoichiometric levels is observed. At this EGR level, an improvement in fuel conversion efficiency of 2% is also noted. This improvement is the direct result of reduced cyclic dispersion. Even further improvement in fuel efficiency should be possible with further reduction in dispersion as this control scheme is refined.

Table III. EGR NOx and uHC Emissions Data for Ricardo

EGR	(u) NO_x (PPM)	(c) NO_x (PPM)	(u) uHC (PPM)	(c) uHC (PPM)
0.0%	5443	4995	3399	3532
12.9%	572	628	3905	3825
15.2%	204	306	4420	4162
18.5%	43	50	10370	9051

Results from the controller implementation on two different engines exemplify the controller’s flexibility. Only engine parameters such as fuel injector information and cylinder geometry had to be changed to extend the controller from the CFR engine to the

Ricardo engine. No offline NN training is required and the controller is model-free. Finally, the task of identifying stabilizing initial weights for the observer and controller NNs, a well known problem in the literature [19], is overcome by initializing the NN weights to zero.

IX. Conclusions

A novel NN controller scheme is presented to reduce the cyclic dispersion in heat release at high EGR levels for a SI engine which is modeled as a non strict feedback nonlinear discrete-time system. The proposed control scheme utilizes both the NN approximation property and a backstepping type approach for maintaining a fixed air to fuel ratio by altering the fuel injected into the cylinder as the control input. 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. Using the nonlinear backstepping like controller, the cyclic dispersion could be reduced significantly, resulting in the potential for decreased emissions and improved fuel economy. Even though the controller was designed for the model heat release output which does not exhibit all the nonlinearities of actual engine heat release, the controller was still able to minimize heat release error. Persistency of excitation condition is not needed, separation principle and certainty equivalence principle are relaxed and linear in the unknown parameter assumption is not used.

Experimental results indicate that the controller can improve engine stability and reduce unburned hydrocarbons at high levels of EGR where significant reductions in NO_x can be realized. Furthermore, the controller is flexible enough to be implemented on two spark ignition research engines.

Appendix

Proof of Theorem: Define the Lyapunov function

$$\begin{aligned} J(k) &= \sum_{i=1}^{3} \frac{\gamma_i}{\alpha_i} \tilde{w}_i^T(k)\tilde{w}_i(k) + \frac{\gamma_4}{\alpha_4}\tilde{x}_1^2(k) + \frac{\gamma_5}{5}\tilde{x}_2^2(k) \\ &\quad + \frac{\gamma_6}{3}\tilde{y}^2(k) + \frac{\gamma_7}{3}e_1^2(k) + \frac{\gamma_8}{4}e_2^2(k) \end{aligned} \tag{A.1}$$

where $0 < \gamma_i$, $i = 1, 5, 8$ are auxiliary constants, the NN weights estimation errors $\tilde{w}_1, \tilde{w}_2$ and $\tilde{w}_3$ are defined in (36), (46) and (63) respectively, the observation errors $\tilde{x}_1(k), \tilde{x}_2(k)$ and $\tilde{y}(k)$ are defined in (32) and (29), the system errors $e_1(k)$ and $e_2(k)$ are defined in (38) and (47), respectively, α_i, $i = 1,2,3$ are NN adaptation gains. The Lyapunov function (A.1) consisting of the system errors, observation errors and the weights estimation errors obviates the need for separation principle and certainty equivalence principle (CE).

The first difference of Lyapunov function is given by

$$\Delta J(k) = \sum_{i=1}^{8} \Delta J_i(k) \tag{A.2}$$

The first term of $\Delta J_1\left(k\right)$ is obtained by using (68) as:

$$\begin{aligned}\Delta J_1(k) &= \frac{\gamma_1}{\alpha_1}[\tilde{w}_1^T(k+1)\tilde{w}_1(k+1) - \tilde{w}_1^T(k)\tilde{w}_1(k)] \\ &\le -\gamma_1(1-\alpha_1\left\|\phi_1(\cdot)\right\|^2)(\hat{w}_1^T(k)\phi_1(\cdot) + l_5\tilde{y}(k))^2 \\ &\quad -\gamma_1\zeta_1^2(k) + 2\gamma_1 l_5^2\tilde{y}^2(k) + 2\gamma_1(w_1^T\phi_1(\cdot))^2\end{aligned} \tag{A.3}$$

where $\zeta_1(k)$ is defined in (37).

Now taking the second term in the first difference (A.1) and substituting (69) into (A.2), we get

$$\begin{aligned}\Delta J_2(k) &= \frac{\gamma_2}{\alpha_2}[\tilde{w}_2^T(k+1)\tilde{w}_2(k+1) - \tilde{w}_2^T(k)\tilde{w}_2(k)] \\ &\le -\gamma_1(1-\alpha_2\left\|\phi_2(\cdot)\right\|^2)(\hat{w}_2^T(k)\phi_2(\cdot) + l_6\tilde{x}_1(k) + l_6 e_1(k))^2 \\ &\quad -\gamma_2\zeta_2^2(k) + 3\gamma_2 l_6^2\tilde{x}_1^2(k) + 3\gamma_2 l_6^2 e_1^2(k) + 3\gamma_2(w_2^T\phi_2(\cdot))^2\end{aligned} \tag{A.4}$$

Taking the third term in the first difference (A.1) and substituting (70) into (A.2), we get

$$\begin{aligned}\Delta J_3(k) &= \frac{\gamma_3}{\alpha_3}[\tilde{w}_3^T(k+1)\tilde{w}_3(k+1) - \tilde{w}_3^T(k)\tilde{w}_3(k)] \\ &\le -\gamma_3(1-\alpha_3\left\|\phi_3(\cdot)\right\|^2)(\hat{w}_3^T(k)\phi_3(\cdot) + l_7\tilde{x}_2(k) + l_7 e_2(k))^2 \\ &\quad -\gamma_3\zeta_3^2(k) + 3\gamma_2 l_7^2\tilde{x}_2^2(k) + 3\gamma_3 l_7^2 e_1^2(k) + 3\gamma_3(w_3^T\phi_3(\cdot))^2\end{aligned} \tag{A.5}$$

Similarly, we have

$$\begin{aligned}\Delta J_4(k) &\le \gamma_4[F_o^2\tilde{x}_1^2(k) + (l_2 - R\cdot F_o)^2\tilde{y}^2(k) + \Delta F^2(k)e_1^2(k)] \\ &\quad +\gamma_4[l_1^{'}(k))^2 e_2^2(k) + l_1^{'}(k))^2\zeta_2^2(k) + d_{11}^2(k) - \tilde{x}_1^2(k)]\end{aligned} \tag{A.6}$$

where

$$l_1^{'}(k) = R\cdot\Delta F(k).CE(k) \tag{A.7}$$

$$d_{11} = R\cdot\Delta F(k).CE(k).w_2^T\phi_2(\cdot) - \Delta x_{1new}(k) - d_1(k) - \left(\Delta F + F_o\right)(r_{O_2} + r_{N_2}) \tag{A.8}$$

and $\zeta_2(k)$ is defined in (51).

$$\begin{aligned}\Delta J_5(k) = \gamma_5[F_o^T \tilde{x}_2^2(k) + (l_2 - F_o)^2 \tilde{y}^2(k) + d_{21}^2(k) - \tilde{x}_2^2(k)] \\ +\gamma_5[((1 - CE(k)\Delta F(k))^2 (e_2^2(k) + \zeta_2^2(k))]\end{aligned} \tag{A.9}$$

where

$$d_{21} = -d_2(k) - \Delta F(k)(1 - CE(k)) \cdot w_2^T \phi_2(\cdot), \tag{A.10}$$

$$\Delta J_6(k) \le \gamma_6(\zeta_1^2(k) + l_3^2 \tilde{y}^2(k) + d_3^2(k) - \tilde{y}^2(k)) \tag{A.11}$$

$$\Delta J_7(k) \le \gamma_7(g_1^2(k) g_2^2(k) + g_1^2 \zeta_1^2(k) + d_{12}^2(k) - e_1^2(k)) \tag{A.12}$$

$$\Delta J_8(k) \le \gamma_8(l_4^2(k) e_2^2(k) + l_4^2 \tilde{x}_2^2(k) + \zeta_3^2(k) + d_{22}^2(k) - e_2^2(k)) \tag{A.13}$$

Combining (A.3) through (A.13) to get the first difference of the Lyapunov function and simplifying it, we get

$$\begin{aligned}\Delta J(k) \le -\gamma_1(1 - \alpha_1 \left\|\phi_1(\cdot)\right\|^2)(\hat{w}_1^T(k)\phi_1(\cdot) + l_5 \tilde{y}(k))^2 + l_6 \tilde{x}_1(k) \\ -\gamma_2(1 - \alpha_2 \left\|\phi_2(\cdot)\right\|^2)(\hat{w}_2^T(k)\phi_2(\cdot) + l_6 e_1(k))^2 + l_7 \tilde{x}_2(k) \\ -\gamma_3(1 - \alpha_3 \left\|\phi_3(\cdot)\right\|^2)(\hat{w}_3^T(k)\phi_3(\cdot) - (\gamma_1 - \gamma_6)\zeta_1^2(k) \\ -(\gamma_2 - \gamma_4(l_1^{'}(k))^2 + l_7 e_2(k))^2 - \gamma_5((1 - CE(k)\Delta F(k))^2 \\ -\gamma_7 g_1^2(k))\zeta_2^2(k) - (\gamma_2 - \gamma_4)\zeta_3^2(k) - ((1 - F_o^2)\gamma_4 \\ -3\gamma_2 l_6^2)\tilde{x}_1^2(k) - ((1 - F_o^2)\gamma_5 - 3\gamma_3 l_7^2 - \gamma_8 l_4^2)\tilde{x}_2^2(k) \\ -((1 - l_3^2)\gamma_6 - (l_1 - R \cdot F_o)^2 \gamma_4 - (l_2 - F_o)^2 \gamma_5 \\ -2\gamma_1 l_5^2)\tilde{y}^2(k) - (\gamma_7 - 3\gamma_2 l_6^2 - \gamma_4 \Delta F^2(k))e_1^2(k) \\ -((1 - l_4^2)\gamma_8 - 3\gamma_3 l_7^2 - \gamma_4(l_7^{'}(k))^2 - \gamma_7 g_1^2(k) e_2^2(k) \\ -\gamma_5((1 - CE(k)\Delta F(k))^2 + D_M^2\end{aligned} \tag{A.14}$$

where,

$$D_M^2 = 2\gamma_1 w_{1m}^2 \phi_{1m}^2 + 3\gamma_2 w_{2m}^2 \phi_{2m}^2 + 3\gamma_3 w_{3m}^2 \phi_{3m}^2 + \gamma_4 d_{11m}^2 + \gamma_5 d_{21m}^2 + \gamma_6 d_{3m}^2 + \gamma_7 d_{12m}^2 + \gamma_8 d_{22m}^2 \tag{A.15}$$

Choose $\gamma_1 = 2,\ \gamma_2 = 1,\ \gamma_3 = 2,\ \gamma_4 = \dfrac{1}{6R^2\Delta F_m^2},\quad \gamma_5 = \dfrac{1}{6\Delta F_m^2},\ \gamma_6 = 1,\quad \gamma_7 = \dfrac{1}{3R^2},$

and $\gamma_8 = 1,$ then, (A.14) is simplified as

$$
\begin{aligned}
\Delta J(k) \le & -2(1-\alpha_1\left\|\phi_1(\cdot)\right\|^2)(\hat{w}_1^T(k)\phi_1(\cdot)+l_5\tilde{y}(k))^2 \\
& -(1-\alpha_2\left\|\phi_2(\cdot)\right\|^2)(\hat{w}_2^T(k)\phi_2(\cdot)+l_6\tilde{x}_1(k)+l_6e_1(k))^2 \\
& -2(1-\alpha_3\left\|\phi_3(\cdot)\right\|^2)(\hat{w}_3^T(k)\phi_3(\cdot)+l_7\tilde{x}_2(k)+l_7e_2(k))^2 \\
& -\zeta_1^2(k)-\frac{1}{3}\zeta_2^2(k)-\zeta_3^2(k)-\left(\frac{(1-F_o{}^2)}{6R^2\Delta F_m^2}-3l_6^2\right)\tilde{x}_1^2(k) \\
& -\left(\frac{(1-F_o{}^2)}{6\Delta F_m^2}-6l_7^2-l_4^2\right)\tilde{x}_2^2(k) \\
& -\left(\left(1-l_3^2\right)-\frac{(l_1-R\cdot F_o)^2}{6R^2\Delta F_m^2}-\frac{(l_2-F_o)^2}{6\Delta F_m^2}-4l_5^2\right)\tilde{y}^2(k) \\
& -\left(\frac{1}{6R^2}-3l_6^2\right)e_1^2(k)+D_M^2-\left((1-l_4^2)-6l_7^2-\frac{2}{3}\right)e_2^2(k)
\end{aligned}
\tag{A.16}
$$

This implies $\Delta J(k) < 0$ as long as (70) - (72) hold and

$$\left|\zeta_1(k)\right| > D_M, or \left|\zeta_2(k)\right| > \sqrt{3}D_M, or \left|\zeta_3(k)\right| > D_M, \tag{A.17}$$

or

$$\left|\tilde{x}_1(k)\right| > \frac{D_M}{\sqrt{\dfrac{(1-F_o^2)}{6R^2\Delta F_m^2}-3l_6^2}}, or \left|\tilde{x}_2(k)\right| > \frac{D_M}{\sqrt{\dfrac{(1-F_o^2)}{6\Delta F_m^2}-6l_7^2-l_4^2}} \tag{A.18}$$

or

$$\left|\tilde{y}(k)\right| > \frac{D_M}{\sqrt{(1-l_3^2)-\dfrac{(l_1-R\cdot F_o)^2}{6R^2\Delta F_m^2}-\dfrac{(l_2-F_o)^2}{6\Delta F_m^2}-4l_5^2}}, \tag{A.19}$$

or

$$\left|e_1(k)\right| > \frac{D_M}{\sqrt{\frac{1}{6R^2} - 3l_6^2}}, \left|e_2(k)\right| > \frac{D_M}{\sqrt{\left(\frac{1}{3} - l_4^2\right) - 6l_7^2}} \tag{A.20}$$

According to a standard Lyapunov extension theorem [17,19], this demonstrates that the system tracking error and the weight estimation errors are *UUB*. The boundedness of $\|\zeta_1(k)\|$, $\|\zeta_2(k)\|$ and $\|\zeta_3(k)\|$ implies that the weight estimation errors $\|\tilde{w}_1(k)\|$, $\|\tilde{w}_2(k)\|$, and $\|\tilde{w}_3(k)\|$ are bounded, and this further implies that the weight estimates $\|\hat{w}_1(k)\|, \|\hat{w}_2(k)\|,$ and $\|\hat{w}_3(k)\|$ are bounded. Therefore all the signals in the closed-loop system are bounded.

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), 1998, pp. 2811 – 2819.

[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.,

[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. Jagannathan, “Control of a class of nonlinear systems using multilayered neural networks,” *IEEE Transactions on Neural Networks*, vol.12, no. 5, September 2001.

[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.

[19]S. Jagannathan, *Neural Network Control of Nonlinear Discrete-time Systems,* CRC Press, Boca Raton, FL 2006.

PAPER 5

Reinforcement Learning-based State-feedback Control of Nonaffine Nonlinear Discrete-time Systems with Application to Engine Spark Timing Control

J. Vance, Q. Yang, and S. Jagannathan

Abstract — A nonaffine, nonlinear, discrete-time system is represented by the nonlinear auto regressive moving average with eXogenous input (NARMAX) model with unknown nonlinear functions. Controlling such systems is extremely difficult and challenging. An equivalent affine-like representation for the tracking error dynamics is first developed from the original nonlinear nonaffine system, whereby a novel controller based on reinforcement learning is proposed for the affine-like nonlinear error dynamic system. The control scheme consists of two NNs. One NN is designated as the critic NN, which approximates a predefined long-term cost function, whereas an action NN is employed to derive a near optimal control signal for the system to track a desired trajectory while minimizing the cost function simultaneously. NN weights are tuned online with derived rules. By using the standard Lyapunov approach, the uniformly ultimate boundedness (UUB) of the tracking error and weight estimates is shown with and without input constraints. Simulation of the control scheme is performed on a nonaffine, nonlinear, discrete-time engine model where cycle-by-cycle control of spark timing reduces cyclic dispersion. Simulation results show the satisfactory performance of the proposed controller.

I. Introduction

Design of control laws for nonaffine, nonlinear, discrete-time systems is difficult because conventional nonlinear design techniques may fail. Past literature [5]-[8] has reported the design of adaptive NN controllers to affine nonlinear discrete-time systems. However, for an unknown nonaffine nonlinear discrete-time system, such controller techniques cannot be directly employed. Further, reinforcement learning control techniques of nonaffine nonlinear discrete-time is not available whereas they are recently introduced for affine nonlinear discrete-time systems [8].

One of the most popularly used nonaffine nonlinear discrete-time representation is nonlinear autoregressive moving average with eXogenous input (NARMAX) model, which is introduced and studied in [1]-[3]. Due to the difficulty in developing the controller design for nonaffine nonlinear discrete-time systems, an affine-like representation is first obtained and subsequently a controller is designed [1]-[3]. However, certain stringent assumptions are exerted, (e.g. boundedness of control input changes), which limits its applicability for many practical applications. Moreover, disturbances are not considered in the development. In this paper, an affine-like representation is first derived by using Mean Value Theorem from the original NARMAX model.

On the other hand, several appealing online neural network controller design methods were introduced in [4]-[6], which were also referred to as forward dynamic programming (FDP) or adaptive critic designs (ACD). The central theme of this family of methods is that the optimal control law and cost function are approximated by parametric structures, such as neural networks (NNs), which are trained over time along with the feedback from the plant. In other words, instead of finding the exact minimum, the ACDs approximate the Bellman equation: $J\left(x(k)\right)=\min_{u(k)}\left\{J\left(x(k+1)\right)+U\left(x(k),u(k)\right)\right\}$, where $x(k)$ is the state and $u(k)=u(x(k))$ is a control law at time step *k*. The strategic utility function $J(x(k),u)=J(x(k))$ represents the cost or performance measure associated with going from *k* to final step *N*, while $U(x(k),x(k+1))$ is the utility function denoting the cost incurred in going from $x(k)$ to $x(k+1)$ by using control $u(k)$.

In the ACD literature, NNs are widely used for approximation. A new NN learning algorithm based on gradient descent rule is introduced in [7]. However, it employs a simplified binary reward or cost function that is a variant of the standard Bellman equation. The work in [8] proposes a near optimal controller design using standard Bellman equation, but the method is only applicable to general affine nonlinear discrete-time systems.

In this paper, the control scheme is applied to a nonaffine, nonlinear, discrete-time system representation for a spark ignition engine. Past works with the engine dynamics based on the Daw model in [14] and [15] have focused on control of the affine fuel input. Modification to the combustion efficiency function of the Daw model gives spark timing as a nonaffine control input.

The desired spark ignition engine control scenario aims to reduce cyclic variation in heat release at lean equivalence ratios by controlling spark timing on a cycle-to-cycle basis. Cyclic variations in combustion mean that undesirable partial burns are present, which lead to increased emissions of NOx (nitrogen oxides) and unburned HC (hydrocarbons), both of which lead to ground-level ozone that is harmful to plants, animals, humans, and materials [18].

For retarded spark timing, where spark is occurring after MBT (maximum brake torque), better mixing leads to lower cyclic variation and reduced HC (hydrocarbon) emissions [16]. Advanced spark timing causes combustion to occur with less time for fuel-air mixing. Spark timing that is advanced too far leads to misfires, and the pronunciation of prior cycle effects becomes apparent [17]. Leaner fuel mixtures will require advanced spark timing from MBT since maximum combustion pressure should occur at TDC (top-dead-center) [13]. Variations in the combustion at lean operating conditions lead to cyclic variations in IMEP (indicated mean effective pressure, work/displacement volume) and heat release, which can be determined to give cycle-by-cycle feedback. Since misfires are formed if spark timing is advanced too far, constraints have to be placed on the control input.

Additionally, this controller is tested on a second-order system that is defined differently, which can clearly demonstrate the performance of the controller when a constraint is placed on the control input. The nonlinear, nonaffine, discrete-time system is made to track a sinusoidal reference signal, and a limit on the input forces the controller to maintain tracking as closely as possible.

This system can be represented as a non-affine nonlinear discrete system with quadratic-performance index as the cost function. The entire closed-loop system consists of two NNs: an action NN to derive the optimal (or near optimal) control signal to track not only the desired system output but also to minimize the long-term cost function and an adaptive critic NN to approximate the long-term cost function $J(x(k))$ and to tune the action NN weights. Further, actuator constraints are incorporated as saturation nonlinearities during the controller development in order to prevent misfires if spark timing is taken as the control input. Closed-loop stability is demonstrated by using standard Lyapunov approach with and without input constraints.

II. Spark Time Control: A Non-affine Nonlinear Discrete-time System

A. Engine Model

A non-strict, non-affine, nonlinear system is given below.

$$\begin{aligned} x_1(k+1) &= f_1(x_1, x_2) + g_1(x_1, x_2)x_2(k) \\ x_2(k+1) &= f_2(x_1, x_2) + g_2(x_1, x_2) \\ y(k+1) &= f_3(x_1, x_2, u) \end{aligned} \tag{1}$$

Spark ignition engine dynamics can be presented in the form of (1) when the control input is spark timing, rather than fuel as in [14]. Writing the engine dynamics of the Daw model, the system equations follow

$$x_1(k+1) = F(k)\left(x_1(k) - R\cdot\eta_c(k)x_2(k)\right) + x_{1new}(k) + d_1(k) \tag{2}$$

$$x_2(k+1) = F(k)\left(1-\eta_c(k)\right)x_2(k) + x_{2new}(k) + d_2(k) \tag{3}$$

$$y(k) = \eta_c(k)x_2(k) \tag{4}$$

where the combustion efficiency function $\eta_c(k)$ is a function of air, fuel, and spark timing as shown in (5). In equations (2) through (4) $x_1(k)$ is air, $x_2(k)$ is fuel, $F(k)$ is residual gas fraction, R is stoichiometric fuel-air ratio, $d_1(k)$ is air disturbance, and $d_2(k)$ is fuel disturbance. The heat release output of the cycle is $y(k)$, which is without units and a product of fuel $x_2(k)$and combustion efficiency $\eta_c(k)$. The spark timing input to the engine system, $\theta_s(k)$, appears in the combustion efficiency function which is directly multiplied to obtain heat release for a given engine cycle as

$$\eta_c(k) = f\left(x_1(k), x_2(k), \theta_s(k)\right) \in \Re \tag{5}$$

The heat release is a function of the combustion efficiency, past heat release values, and the control input.

$$y(k+1) = f\left(\eta_c(k), y(k), y(k-1), \ldots, y(k-n), u(k)\right) \in \Re \tag{6}$$

Total fuel and air within the cylinder are not measurable on a cycle-by-cycle basis because the residual fuel, residual air, and residual products from combustion are unknown.

B. System Dynamics

The engine system model in (6) with disturbance can be written in NARMAX form [1] as follows

$$\begin{aligned} y(k+\tau) &= f(\eta_c(k), \overline{y}_k, \overline{u}_{k-1}, u(k), \overline{d}_{k+\tau-1}) \\ &= f(w_k, u(k), \overline{d}_{k+\tau-1}) \end{aligned} \tag{7}$$

where $w_k = \left[\eta_c(k), \overline{y}_k^T, \overline{u}_{k-1}^T\right]^T$ and $\overline{u}_{k-1} = \left[u(k-1), ..., y(k-n+1)\right]^T$ denotes the system inputs and $\overline{y}_k = \left[y(k), ..., y(k-n+1)\right]^T$ denotes the system outputs. The term $\overline{d}_{k+\tau-1} = \left[d(k+\tau-1), ..., d(k)\right]^T$ is the disturbance, and τ represents the system delay, or the relative degree of the system [2]. Note that the output $y(k)$ is considered measurable with initial values in a compact set Ω_{y_0}. Furthermore, several mild assumptions are needed in order to proceed.

Assumption 1: The unknown nonlinear function $f(\cdot)$ in (7) is continuous and differentiable.

Assumption 2: The disturbance $d(k)$ is bounded with a known bound $|d(k)| \le d_M$, and the partial derivative $|\partial f / \partial d(k)| \le D_M$ is also bounded, with D_M a positive constant.

With assumption 2, by using Mean Value Theorem, equation (1) can be rephrased as

$$\begin{aligned} y(k+\tau) &= f(w_k, u(k), \bar{d}_{k+\tau-1}) = f(w_k, u(k), 0) + \delta_f^T \bar{d}_{k+\tau-1} \\ &= f(w_k, u(k), 0) + \delta_{d_k} \end{aligned} \tag{8}$$

where

$$\delta_f = \left[\left. \frac{\partial f}{\partial d(k+\tau-1)} \right|_{d(k+\tau-1)=d_\xi(k+\tau-1)}, \ldots, \left. \frac{\partial f}{\partial d(k)} \right|_{d(k)=d_\xi(k)} \right]^T,$$

$$\delta_{d_k} = \delta_f^T \bar{d}_{k+\tau-1},$$

and $d_\xi(k)$ is between 0 and $d(k)$, or $d_\xi(k) = 0 + \lambda(d(k) - 0)$, $\lambda \in [0,1]$. Through this paper, they have the same meaning, and we will present by the former fashion for simplicity.

Lemma 1: δ_{d_k} is bounded by $|\delta_{d_k}| \le \tau D_M d_M$.

Proof: This lemma can be straightforwardly verified from (8) and assumption 2.

Our objective is to design a control law to drive the system output $y(k)$ to track a desired trajectory $y_d(k)$. Before we proceed, let us construct the following virtual system, which is free of disturbance.

$$y_n(k+\tau) = f(w_k, u(k), 0) \tag{9}$$

Assumption 3: $\partial f / \partial u(k) = g(k)$ is bounded and satisfies $0 < g_{\min} \le g(k) \le g_{max}$, where $g_{\min}$ and $g_{\max}$ are positive constants [8].

Assumption 4: Virtual system (9) is invertibly stable [9], which means bounded system output implies bounded system input.

From Assumptions 3 and 4, we can draw the conclusion that for any output trajectory $y_n(k+\tau) = f(w_k, u(k), 0)$, there exists a unique and continuous (smooth) function $u(k) = f^{-1}(w_k, y_n(k+\tau), 0)$ [2], [11].

III. Controller Methodology

A. Optimal Control

In this paper, we consider the long-term cost function as

$$\begin{aligned} J(k) = J(y(k), u) &= \sum_{i=t_0}^{\infty} \gamma^i r(k+i) \\ &= \sum_{i=t_0}^{\infty} \gamma^i [q(y(k+i)) + u^T(k+i) R u(k+i)] \end{aligned} \tag{10}$$

where $J(k)$ stands for $J(x(k), u)$ for simplicity, u is a given control policy, R is a positive design constant and γ $(0 \le \gamma \le 1)$ is the discount factor for the infinite-horizon problem [8]. One can observe from (10) that the long-term cost function is the discounted sum of the immediate cost function or Lagrangian expressed as

$$\begin{aligned} r(k) &= q(y(k)) + u^T(k) R u(k) \\ &= (y(k) - y_d(k))^T Q (y(k) - y_d(k)) + u^T(k) R u(k) \\ &= Qe^2(k) + Ru^2(k) \end{aligned} \tag{11}$$

where Q is a positive design constant. In this paper, we are using a widely applied standard quadratic cost function defined based on the tracking error $e(k) = y(k) - y_d(k)$ in contrast with [7]. The immediate cost function $r(k)$ can be also viewed as the system performance index for the current step.

The basic idea in the adaptive critic or reinforcement learning design is to approximate the long-term cost function $J(k)$ (or its derivative, or both), and generate the control signal minimizing the cost. By learning online through an algorithm, the online approximator will converge to the optimal cost function and the controller will in turn generate an optimal signal. As a matter of fact, for an optimal control law, which can be expressed as $u^*(k) = u^*(y(k))$, the optimal long-term cost function can be written as $J^*(k) = J^*(y(k), u^*(y(k))) = J^*(y(k))$, which is just a function of the current system output [10]. Next, one can state the following assumption.

Assumption 5: The optimal cost function $J^*(k)$ is finite and bounded over the compact set $S \subset R$ by J_M.

B. Affine-like Dynamics

To avoid the complexity of non-affine nonlinear systems, an affine-like representation is desirable for the controller design. By applying the Taylor series expansion of system (9) with respect to $u(k)$ around $u(k-1)$ yields

$$\begin{aligned} y(k+\tau) &= f(w_k, u(k), 0) + \delta_{d_k} \\ &= f(w_k, u(k-1), 0) + \frac{\partial f(w_k, u(k-1), 0)}{\partial u} \Delta u(k) \\ &\quad + \frac{1}{2} \cdot \frac{\partial^2 f(w_k, u_\mu(k), 0)}{\partial u^2} \Delta u^2(k) + \delta_{d_k} \\ &= F(w_k, u(k)) + G(w_k) \Delta u(k) + \delta_{d_k} \end{aligned} \tag{12a}$$

$$F(w_k,u(k)) = f(w_k,u(k-1),0) + \frac{1}{2}\cdot\frac{\partial^2 f(w_k,u_\mu(k),0)}{\partial u^2}\Delta u^2(k) \tag{12b}$$

$$G(w_k) = \frac{\partial f(w_k,u(k-1),0)}{\partial u} \tag{12c}$$

where $u_\mu(k)$ is between $u(k)$ and $u(k+1)$ (or $u_\mu(k) = u(k+1) + \lambda(u(k+1) - u(k))$, $\lambda \in [0,1]$) by using Mean Value Theorem. In other words, there are no higher order terms in the Taylor series expansion missing, since they are incorporated into the second derivative. By observing (12), we have the equation similar to the virtual system as

$$y_n(k+\tau) = F(w_k,u(k)) + G(w_k)\Delta u(k) \tag{13}$$

Lemma 2: Consider any desired system trajectory $y_d(k) \in R$ and corresponding nominal desired control input $u_d(k) = f^{-1}(w_k, y_d(k+\tau),0)$, there exists $u_\xi(k)$ between any input $u_n(k)$ and $u_d(k)$ to the system such that

$$\begin{aligned} F(w_k,u_n(k)) = F(w_k,u_d(k)) + \frac{\partial F(w_k,u_\xi(k))}{\partial u} \\ \times\frac{\partial f^{-1}(w_k,y_\xi(k+\tau),0)}{\partial y}\cdot(y_n(k+\tau) - y_d(k+\tau)) \end{aligned} \tag{14}$$

where $u_\xi(k) = f^{-1}(w_k, y_\xi(k+\tau),0)$.

Lemma 3: Consider the output of the virtual system $y_n(k+\tau) = f(w_k,u_n(k),0)$ for a given input $u_n(k)$, then there exists $y_\varsigma(k+\tau)$ between $y_n(k+\tau)$ and $y_d(k+\tau)$ such that

$$
\begin{aligned}
u_n(k) &= f^{-1}(w_k, y_n(k+\tau), 0) \\
&= f^{-1}(w_k, y_d(k+\tau), 0) + \frac{\partial f^{-1}(w_k, y_\varsigma(k+\tau), 0)}{\partial y}(y_n(k+\tau) - y_d(k+\tau)) \\
&= u_d(k) + \frac{\partial f^{-1}(w_k, y_\varsigma(k+\tau), 0)}{\partial y}(y_n(k+\tau) - y_d(k+\tau))
\end{aligned}
\tag{15}
$$

Proof: Lemmas 2 and 3 can be readily obtained by using Chain Rule and Mean Value Theorem. Further, we have following lemma.

Lemma 4: Consider system (13) with Lemma 2 and 3, we have

$$
\frac{\partial F(w_k, u_\xi(k))}{\partial u} \cdot \frac{\partial f^{-1}(w_k, y_\xi(k+\tau), 0)}{\partial y} + G(w_k)\frac{\partial f^{-1}(w_k, y_\varsigma(k+\tau), 0)}{\partial y} = 1 \tag{16}
$$

Proof: i) If $y_n(k+\tau) = y_d(k+\tau)$, then $y_\xi(k+\tau) = y_\varsigma(k+\tau) = y_d(k+\tau)$. Therefore, (16) could be obtained by differentiating (13) with respect to $y_n(k+\tau)$.

ii) If $y_n(k+\tau) \neq y_d(k+\tau)$, then from (13), one has

$$
\begin{aligned}
y_n(k+\tau) &= F(w_k, u_n(k)) + G(w_k)(u_n(k) - u(k-1)) \\
&= F(w_k, u_d(k)) + \frac{\partial F(w_k, u_\xi(k))}{\partial u} \\
&\quad \times \frac{\partial f^{-1}(w_k, y_\xi(k+\tau), 0)}{\partial y} \cdot (y_n(k+\tau) - y_d(k+\tau)) \\
&\quad + G(w_k)(u_d(k) - u(k-1)) \\
&\quad + \frac{\partial f^{-1}(w_k, y_\varsigma(k+\tau), 0)}{\partial y}(y_n(k+\tau) - y_d(k+\tau)) \\
&= F(w_k, u_d(k)) + G(w_k)(u_d(k) - u(k-1)) \\
&\quad + (\frac{\partial F(w_k, u_\xi(k))}{\partial u}\frac{\partial f^{-1}(w_k, y_\xi(k+\tau), 0)}{\partial y} \\
&\quad + \frac{\partial f^{-1}(w_k, y_\varsigma(k+\tau), 0)}{\partial y}) \times (y_n(k+\tau) - y_d(k+\tau))
\end{aligned}
\tag{17a}
$$

Substituting $y_d(k+\tau) = F(w_k, u_d(k)) + G(w_k)(u_d(k) - u(k-1))$ into (17) yields

$$\frac{\partial F(w_k,u_\xi(k))}{\partial u}\cdot\frac{\partial f^{-1}(w_k,y_\xi(k+\tau),0)}{\partial y}+G(w_k)\frac{\partial f^{-1}(w_k,y_\varsigma(k+\tau))}{\partial y}=1 \qquad \text{(17b)}$$

Lemma 5: For any $y_\varsigma(k+\tau)\in S$ and corresponding control input $u_\varsigma(k)=f^{-1}(w_k,y_\varsigma(k+\tau),0)$, the following statement holds

$$\frac{\partial f(w_k,u_\varsigma(k),0)}{\partial u}\cdot\frac{\partial f^{-1}(w_k,y_\varsigma(k+\tau))}{\partial y}=1 \qquad \text{(18)}$$

Proof: It can be straightforward to verify (11) by differentiating $y_\varsigma(k+\tau)=f(w_k,f^{-1}(w_k,y_\varsigma(k+\tau)),0)$ with respect to $y_\varsigma(k+\tau)$.

Therefore, substituting (14) into (12) produces the system dynamics in terms of the tracking error as

$$\begin{aligned} e(k+\tau) &= y(k+\tau)-y_d(k+\tau)\\ &= F(w_k,u(k))+G(w_k)\Delta u(k)+\delta_{d_k}-y_d(k+\tau)\\ &= F(w_k,u_d(k))+\frac{\partial F(w_k,u_\xi(k))}{\partial u}\\ &\quad\times\frac{\partial f^{-1}(w_k,y_\xi(k+\tau),0)}{\partial y}\cdot(y(k+\tau)-y_d(k+\tau))\\ &\quad+G(w_k)\Delta u(k)+\delta_{d_k}-y_d(k+\tau) \end{aligned} \qquad \text{(19)}$$

Making use of Lemma 4, (19) can be written as

$$\begin{aligned} e(k+\tau) &= F(w_k,u_d(k))+G(w_k)\Delta u(k)+\delta_{d_k}-y_d(k+\tau)\\ &\quad+(1-G(w_k)\frac{\partial f^{-1}(w_k,y_\varsigma(k+\tau),0)}{\partial y})\cdot(y(k+\tau)-y_d(k+\tau))\\ &= F(w_k,u_d(k))+G(w_k)\Delta u(k)+\delta_{d_k}-y_d(k+\tau)\\ &\quad+(1-G(w_k)\frac{\partial f^{-1}(w_k,y_\varsigma(k+\tau),0)}{\partial y})\cdot e(k+\tau) \end{aligned} \qquad \text{(20)}$$

Combining (18) and (20), one has

$$e(k+\tau)=\frac{\partial f(w_k,u_\varsigma(k),0)}{\partial u}(\frac{F(w_k,u_d(k))+\delta_{d_k}-y_d(k+\tau)}{G(w_k)}+\Delta u(k)) \tag{21}$$

By defining $\partial f(w_k,u_\varsigma(k),0)/\partial u=\kappa_k$, (21) can be rephrased as

$$\begin{aligned} e(k+\tau)&=\frac{\kappa_k}{G(w_k)}(F(w_k,u_d(k))-y_d(k+\tau))+\kappa_k\Delta u(k)+\frac{\kappa_k}{G(w_k)}\delta_{d_k} \\ &=F_a(w_k,y_d(k+\tau),\kappa_k)+\kappa_k\Delta u(k)+\delta_{\kappa_k} \end{aligned} \tag{22a}$$

$$F_a(w_k,y_d(k+\tau),\kappa_k)=\frac{\kappa_k}{G(w_k)}(F(w_k,u_d(k))-y_d(k+\tau)) \tag{22b}$$

$$\delta_{\kappa_k}=\frac{\kappa_k}{G(w_k)}\delta_{d_k} \tag{22c}$$

Notice that $0<g_{\min}\le\kappa_k\le g_{max},0<g_{\min}\le G(w_k)\le g_{max}$ due to Assumption 3. By referring to Lemma 1, one also observes that δ_{κ_k} is bounded above by $\left|\delta_{\kappa_k}\right|\le g_{max}\tau D_M d_M/g_{\min}$.

By rewriting the non-affine system into an equivalent affine-like representation (22) in terms of error dynamics, the difficulty of designing controllers for nonaffine nonlinear discrete-time systems could be avoided.

C. Online Controller

Proceeding from last section, the purpose of this study is to design an online reinforcement learning NN controller for the equivalent error dynamics (22), such that 1) all the signals in the closed-loop system remain UUB; 2) the output $y(k)$ follows a desired trajectory $y_d(k)\in S$; and 3) the long-term cost function (4) is minimized so that a near optimal control input can be generated [8]. Here, the “online” means the learning of the controller takes place “in real-time” by interacting with the plant, instead of in an offline or iterative manner.

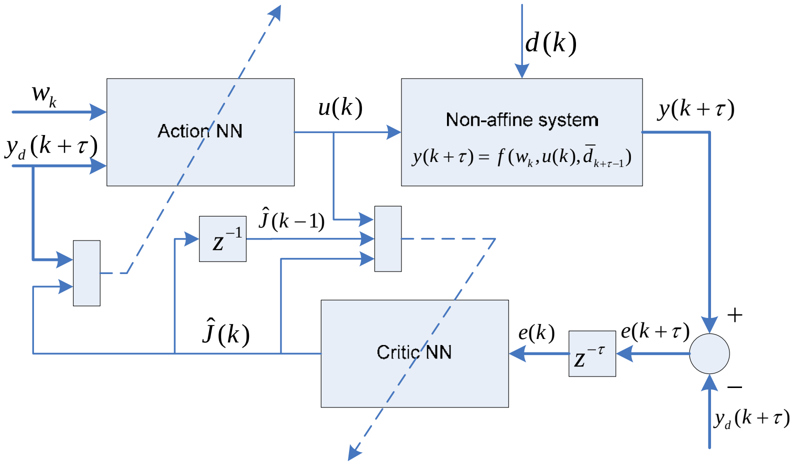

Fig. 1. Online reinforcement learning neural controller structure.

The block diagram of the proposed controller is shown in Fig. 1, where the action NN is providing a near optimal control signal to the nonlinear system while the critic NN approximates the long-term cost function. The two NN weights are initialized to zero and trained online without any offline learning phase.

In our controller architecture, we consider the action and the critic NN having two layers, and the output of the NN can be given by $Y=W^T\phi(V^TX)$, where V and W are the hidden layer and output layer weights respectively. X is the input vector of the NN and we choose $\phi(X)=2/(1+e^{-X})-1$ as the activation function.

We know that any continuous function $f(X)\in C^N(S)$ can be written as

$$f(X)=W^T\phi(V^TX)+\varepsilon(X) \tag{23}$$

with $\varepsilon(X)$ a NN functional reconstruction error vector. In our design, W is adapted online but V is initially selected at random and held fixed during entire learning process. It is demonstrated in [13] that if the number of hidden layer neurons is sufficiently large, the NN approximation error $\varepsilon(X)$ can be made arbitrarily small since the activation function vector forms a basis. Furthermore, in this paper, a novel tuning algorithm is proposed making the NN weights robust and PE condition unnecessary. Before we proceed, the following mild assumption is needed.

Assumption 6: The desired trajectory of the system output, $y_d\left(k\right)$, is bounded over the compact subset of R.

1) The Action NN Design

Consider system (22), to eliminate the tracking error, a desired control law is given by

$$u_d(k) = u(k-1) - \frac{1}{\kappa_k} F_a(w_k, y_d(k+\tau), \kappa_k) \tag{24}$$

By this means, the tracking error will go to zero after τ steps if no disturbance presents.

However, since both $F_a(w_k, y_d(k+\tau), \kappa_k)$ and κ_k are unknown smooth nonlinear functions, the desired feedback control $u_d(k)$ cannot be implemented directly. Instead, an action NN is employed to produce the control signal. From (24) and considering Assumption 3 and 4, the desired control signal can be approximated by the action NN as

$$u_d(k) = w_a^T \phi_a(v_a^T s(k)) + \varepsilon_a(s(k)) = w_a^T \phi_a(k) + \varepsilon_a(k) \tag{25}$$

where $s(k) = \left[w_k^T, y_d(k+\tau) \right]^T$ is the action NN input vector. n_a is the number of neurons in the hidden layer, and $w_a \in R^{n_a \times 1}$, $v_a \in R^{2n \times n_a}$ denote the desired weights of the output and hidden layer, respectively, with $\varepsilon_a(k) = \varepsilon_a(s(k))$ as the action NN approximation error. Since v_a is fixed, for simplicity purpose, the hidden layer activation function vector $\phi_a(v_a^T s(k)) \in R^{n_a}$ is written as $\phi_a(k)$.

Considering the fact that the desired weights are unavailable for us, the actual NN weights have to be trained online and its actual output can be expressed as

$$u(k) = \hat{w}_a^T(k)\phi_a(v_a^T s(k)) = \hat{w}_a^T(k)\phi_a(k) \tag{26}$$

where $\hat{w}_a(k) \in R^{n_a \times 1}$ is the actual weight matrix of the output layer at instant *k*.

Using the action NN output as the control signal, and substituting (25) and (26) into (22) yields

$$\begin{aligned} e(k+\tau) &= F_a(w_k, y_d(k+\tau), \kappa_k) + \kappa_k \Delta u(k) + \delta_{\kappa_k} \\ &= \kappa_k(u(k) - u_d(k)) + \delta_{\kappa_k} \\ &= \kappa_k(\tilde{w}_a^T(k)\phi_a(k) - \varepsilon_a(k)) + \delta_{\kappa_k} \\ &= \kappa_k \zeta_a(k) + d_a(k) \end{aligned} \tag{27}$$

where

$$\tilde{w}_a(k) = \hat{w}_a(k) - w_a \tag{28}$$

$$\zeta_a(k) = \tilde{w}_a^T(k)\phi_a(k) \tag{29}$$

$$d_a(k) = -\kappa_k \varepsilon_a(k) + \delta_{\kappa_k} \tag{30}$$

Next the critic NN design with updating rule is followed.

2) The Critic NN Design

As stated above, the proposed controller can stabilize the closed-loop system along with minimizing the cost function. In this regard, a critic NN is employed to approximate the unknown long-term cost function $J\left(k\right)$ for current stage.

First, the prediction error generated by the critic or the Bellman error [9] is defined as

$$e_c(k) = \gamma \hat{J}(k) - \hat{J}(k-1) + r(k) \tag{31}$$

where the subscript “c” stands for the “critic” and

$$\hat{J}\left(k\right) = \hat{w}_c^T(k)\phi_c(v_c^T e(k)) = \hat{w}_c^T(k)\phi_c(k) \tag{32}$$

$\hat{J}\left(k\right) \in R$ is the critic NN output, which is for approximating $J\left(k\right)$. $\hat{w}_c\left(k\right) \in R^{n_c \times 1}$ and $v_c \in R^{1 \times n_c}$ represent the actual weight matrices of the output and hidden layer, respectively. The term n_c denotes the number of the neurons in the hidden layer. Similar to HDP, the tracking error $e(k)$ is selected as the critic NN input. Again, the activation function vector of the hidden layer $\phi_c(v_c^T e(k)) \in R^{n_c}$ is simply denoted as $\phi_c(k)$. Provided with enough number of hidden layer neurons, the optimal long-term cost function $J*\left(k\right)$ can be approximated with arbitrarily small approximation error $\varepsilon_c(k)$ as

$$J*(k) = w_c^T \phi_c(v_c^T e(k)) + \varepsilon_c(e(k)) = w_c^T \phi_c(k) + \varepsilon_c(k) \tag{33}$$

Similarly, the critic NN weight estimation error can be defined as

$$\tilde{w}_c(k) = \hat{w}_c(k) - w_c \tag{34}$$

where the approximation error is given by

$$\zeta_c(k) = \tilde{w}_c^T(k)\phi_c(k) \tag{35}$$

Thus, we obtain

$$\begin{aligned} e_c(k) &= \gamma \hat{J}(k) - \hat{J}(k-1) + r(k) \\ &= \gamma\zeta_c(k) + \gamma J^*(k) - \zeta_c(k-1) - J^*(k-1) \\ &\quad + r(k) - \varepsilon_c(k) + \varepsilon_c(k-1) \end{aligned} \tag{36}$$

Next we propose the weight tuning algorithms for both NNs.

3) Weight Updating for the Critic NN

Following the discussion from the last section, the objective function to be minimized by the critic NN can be defined as a quadratic function of Bellman error as

$$E_c(k) = \frac{1}{2}e_c^T(k)e_c(k) = \frac{1}{2}e_c^2(k) \tag{37}$$

Using a standard gradient-based adaptation method, the weight updating algorithm for the critic NN is given by

$$\hat{w}_c(k+\tau) = \hat{w}_c(k) + \Delta\hat{w}_c(k) \tag{38}$$

where

$$\Delta\hat{w}_c(k) = \alpha_c\left[-\frac{\partial E_c(k)}{\partial\hat{w}_c(k)}\right] \tag{39}$$

with $\alpha_c \in R$ is the adaptation gain.

Combining (31), (32), (37) with (39), the critic NN weight updating rule can be obtained by using the chain rule as [8]

$$\begin{aligned}\Delta\hat{w}_c(k) &= -\alpha_c\frac{\partial E_c(k)}{\partial\hat{w}_c(k)} = -\alpha_c\frac{\partial E_c(k)}{\partial e_c(k)}\frac{\partial e_c(k)}{\partial\hat{J}(k)}\frac{\partial\hat{J}(k)}{\partial\hat{w}_c(k)} \\ &= -\alpha_c\gamma\phi_c(k)(\gamma\hat{J}(k)+r(k)-\hat{J}(k))\end{aligned} \tag{40}$$

4) Weight Updating for the Action NN

The objective for adapting the action NN is to track the desired output and to lower the cost function simultaneously. Therefore, the error for the action NN can be formed by combining the functional estimation error $\zeta_a(k)$, and the critic signal $\hat{J}(k)$. Let

$$\begin{aligned}e_a(k) &= \sqrt{\kappa_k}\zeta_a(k)+\left(\sqrt{\kappa_k}\right)^{-1}\left(\hat{J}(k)-J_d(k)\right) \\ &= \sqrt{\kappa_k}\zeta_a(k)+\left(\sqrt{\kappa_k}\right)^{-1}\hat{J}(k)\end{aligned} \tag{41}$$

where $\zeta_a(k)$ is defined in (29). The desired long-term cost function $J_d(k)$ is nominally defined and is considered to be zero (“0”), which means as low as possible [8].

Hence, the weights of the action NN $\hat{w}_a(k)$ are tuned to minimize the error

$$E_a(k) = \frac{1}{2}e_a^T(k)e_a(k) \tag{42}$$

Combining (27), (29), (41), (42) and using the chain rule yields

$$\begin{aligned}\Delta\hat{w}_a(k) &= -\alpha_a\frac{\partial E_a(k)}{\partial\hat{w}_a(k)} = -\alpha_a\frac{\partial E_a(k)}{\partial e_a(k)}\frac{\partial e_a(k)}{\partial\zeta_a(k)}\frac{\partial\zeta_a(k)}{\partial\hat{w}_c(k)} \\ &= -\alpha_a\phi_a(k)(\kappa_k\zeta_a(k)+\hat{J}(k))^T \\ &= -\alpha_a\phi_a(k)(e(k+\tau)-d_a(k)+\hat{J}(k))^T\end{aligned} \tag{43}$$

where $\alpha_a \in R^+$ is the adaptation gain of the action NN. Since $d_a(k)$ is typically unavailable, similar to the ideal case, we assume the $d(k)$ and the mean value of $\varepsilon_a(k)$

over the compact subset of R to be zero [8], and obtain the weight updating algorithm for the action NN as

$$\hat{w}_a(k+\tau) = \hat{w}_a(k) - \alpha_a \phi_a(k)(e(k+\tau) + \hat{J}(k))^T \tag{44}$$

IV. Control without Saturation

First, the design of the controller is addressed when no input constraints are asserted. The following mild assumptions are needed before we proceed.

Assumption 7: Let the unknown desired output layer weights for the action and critic NNs be upper bounded such that

$$\|w_a\| \le w_{am}\text{, and } \|w_c\| \le w_{cm} \tag{45}$$

where $w_{am} \in R^+$ and $w_{cm} \in R^+$ represent the bounds on the unknown target weights. Here $\|\cdot\|$ stands for the Frobenius norm [14].

Assumption 8: The activation functions for the action and critic NNs are bounded by known positive values, such that

$$\|\phi_a(k)\| \le \phi_{am},\ \|\phi_c(k)\| \le \phi_{cm} \tag{46}$$

where $\phi_{am}, \phi_{cm} \in R^+$ is the upper bound. It is easily satisfied, since hyperbolic tangent sigmoid transfer function is chosen.

Assumption 9: The NN approximation errors $\varepsilon_a(k)$ and $\varepsilon_c(k)$ are bounded above over the compact set $S \subset R$ by ε_{am} and ε_{cm} [10].

Lemma 6: With the Assumption 3, 9, the term $d_a(k)$ in (30) is bounded over the compact set $S \subset R$ by

$$\left|d_a(k)\right| \le d_{am} = g_{\max}\varepsilon_{am} + g_{max}\tau D_M d_M / g_{\min} \tag{47}$$

Combining Assumptions 1, 3, and 4 and Facts 1 and 2, the main result for this section is introduced next.

Theorem 1 (No Actuator Constraints): Consider the nonlinear discrete-time system given by (8) whose dynamics can be expressed as (22). Let the Assumptions 1 through 9 hold with the disturbance bound d_M, a known constant. Let the control input be provided by the action NN (26) with the critic NN output given by (32). Further, let the weights of the action NN and the critic NN be tuned by (40) and (44), respectively, and let us assume there are no constraints on the input. Then, the tracking error, $e(k)$, and the NN weight estimates of the action and critic NNs, $\tilde{w}_a(k)$ and $\tilde{w}_c(k)$, are *UUB*, provided the controller design parameters are selected as

(a) $$0 < \alpha_a \phi_a^2(k) < \frac{g_{\min}}{g_{\max}^2} \tag{48}$$

(b) $$0 < \alpha_c \phi_c^2(k) < 1/\gamma^2 \tag{49}$$

(c) $$\gamma > \frac{1}{2} \tag{50}$$

where α_a and α_c are NN adaptation gains, and γ is employed to define the *strategic* utility function.

Proof: See Appendix.

Remark: It is demonstrated the boundedness of tracking error and NN weights without using the persistency of excitation condition. Additionally, the tracking error can be made small by appropriately selecting the control gains.

Remark: A well-defined controller is developed in this paper since a single NN is utilized to approximate two nonlinear functions.

Remark: It is important to note that in this theorem there is no linearity in the parameters assumption, in contrast with standard work in the discrete-time adaptive control. Additionally, certainty equivalence principle is not used.

Remark: Compared to other adaptive critic or reinforcement learning schemes [4]-[7], the proposed approach ensures closed-loop stability using the Lyapunov approach even though gradient based adaptation is employed.

V. Control with Actuator Saturation

When the actuator constraints are applied, the control design given in the previous section has to be modified as follows.

A. Design of the Auxiliary Tracking Error System

The control input $v(k)$ is defined as the output of the action NN

$$v(k) = \hat{w}_a^T(k)\phi_a(k) \tag{51}$$

The actual control input after the incorporation of saturation constraints is selected as

$$u(k) = \begin{cases} v(k), & \text{if } \|v(k)\| \le u_{\max} \\ u_{\max}\operatorname{sgn}(v(k)), & \text{if } \|v(k)\| > u_{\max} \end{cases} \tag{52}$$

where $u_{\max} \in \Re^{+}$ is the upper bound for the control input $u(k)$. Then, the closed-loop system (27) becomes

$$e(k+\tau) = \kappa_k \zeta_a(k) + d_a(k) + \kappa_k \Delta u(k) \tag{53}$$

where the control signal difference $\Delta u(k) = u(k) - v(k)$. To remove the effect of $\Delta u(k) \in \Re$, which can be seen as a disturbance, we define now

$$e_u(k) = e(k) - \kappa_k \Delta u(k) \tag{54}$$

From (53), we have

$$e_u(k+\tau) = \kappa_k \zeta_a(k) + d_a(k) \tag{55}$$

In the remainder of this section, (55) is used to focus on designing NN algorithms to guarantee the stability of the auxiliary error, $e_u(k)$. Once $e_u(k)$ is proven stable, the stability of $e(k)$ follows.

B. Adaptive Critic Design with Saturation

With the presence of saturation, the critic NN design is the same as that of the design presented in the last section. However, the action NN design contains the auxiliary error signal $e_u(k)$ instead of the tracking error, $e(k)$. Hence, the action NN weight update is given by

$$\hat{w}_a(k+\tau) = \hat{w}_a(k) - \alpha_a \phi_a(k)(e_u(k+\tau) + \hat{J}(k))^T \tag{56}$$

C. Closed-loop System Stability Analysis

Theorem 2 (with Actuator Constraints): Consider the nonlinear discrete-time system given by (8) whose dynamics can be expressed as (22). Let the Assumptions 1 through 9 still hold. Let the control input be provided by the action NN (26), with the critic NN output being (32). Further, let the weights of the action NN and the critic NN be tuned by (56) and (44), respectively. Then, the auxiliary tracking error, $e_u(k)$, and the NN weight estimates of the action and critic NNs, $\tilde{w}_a(k)$ and $\tilde{w}_c(k)$, are *UUB*, with the bounds specifically given by (A.13) – (A.15), provided the controller design parameters are selected as (48) – (50).

Proof: See Appendix.

Corollary 1: Let the hypotheses presented in Theorem 2 hold. The tracking error, $e(k)$, is also UUB.

Proof: See Appendix.

VI. Simulations

A. Spark Ignition Engine Model

Control of fuel for operating a spark ignition engine has been simulated in [14] and [15], but here spark timing is taken as the desired control variable. The nonlinear, discrete-time engine model used in those papers and created in [19] does not use spark timing information. In order to simulate the controller with spark timing, the model was changed to use a new function for combustion efficiency. The sigmoid-like combustion efficiency function of the aforementioned sources follows.

$$CE(k) = \frac{CE_{\max}}{1+100^{-(\varphi(k)-\varphi_m)/(\varphi_u-\varphi_l)}} \tag{57}$$

The combustion efficiency function of (57) is used as a model for combustion efficiency at discrete spark times. This method is used to produce a two-input function with equivalence ratio and spark-timing as inputs and a combustion efficiency output.

Engine data was recorded at several operating points for a Ricardo Hydra research engine with a Ford Zetec head. This engine is port fuel-injected and has one cylinder. For data collection, the engine was motored at 1,000 RPM by a dynamometer, and the pressure in the intake manifold (manifold absolute pressure, or MAP) was maintained around 90 kPa – roughly a mid-load operating condition. An in-cylinder pressure sensor samples pressure at every crank angle of the cycle. These pressure measurements combined with knowledge of the cylinder geometry are used to determine heat release for the engine cycle.

The combustion efficiency function of the Daw model is modified to include spark timing information by collecting engine data at a predetermined equivalence ratio and spark timing, and then the combustion efficiency is determined. Combustion efficiency is proportional to amount of fuel in combustion and heat release. Heat release is determined from in-cylinder pressure measurements during combustion, and fuel input per cycle is known. Equivalence ratios from 1.00 down to 0.70 with spark timings ranging from -40 to +20 degrees relative to MBT were used as operating points for data collection as seen in Table I for equivalence ratio 0.80.

Engine data was obtained for equivalence ratios of {1.00, 0.90, 0.80, 0.76, 0.74, 0.72}. Based on the model combustion efficiency function from [19], data for other equivalence ratios of {0.60, 0.50, 0.40, 0.30} were extrapolated for each of the tested spark timings. This allowed for a full set of data to train the neural network that would approximate the combustion efficiency function with spark timing as an input, in addition to equivalence ratio.

Table I. Combustion Efficiency Data for Equ. Ratio 0.80

Equ. Ratio	Spark Time from MBT	Combustion Efficiency
0.79966	0	0.69337
0.80005	1	0.69101
0.80014	5	0.68546
0.8011	10	0.67473
0.80498	20	0.65049
0.79778	-1	0.69638
0.79574	-5	0.70452
0.79585	-10	0.71324
0.7895	-20	0.71773
0.777	-30	0.66821
0.78991	-40	0.57791

A two-layer neural network with 6 hidden layer nodes is trained with back-propagation to obtain a combustion efficiency function as seen in Fig. 2. The neural network equations are

$$\begin{aligned} z &= \sigma\left(V^T x + b_v\right) \\ y &= W^T z + b_w \end{aligned} \tag{51}$$

where y is the combustion efficiency output, $x \in \Re^{2x1}$ is the input vector containing spark timing and equivalence ratio,

$$x = \left[\theta_s \quad \varphi\right]^T \tag{53}$$

and the sigmoid activation function of the first layer is

$$\sigma(x) = \frac{2}{1+e^{-2x}} - 1 \tag{52}$$

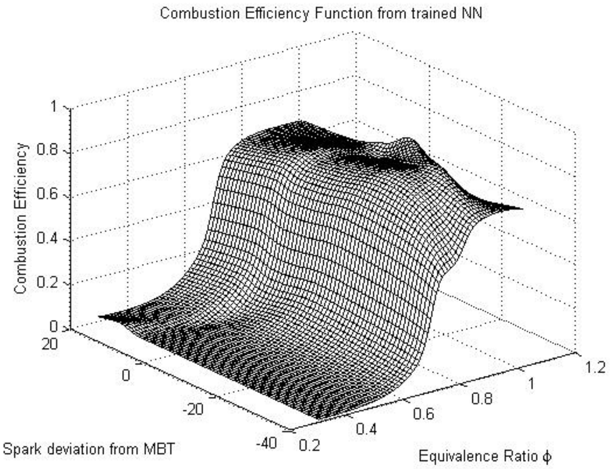

Fig. 2. Combustion efficiency function based on trained neural network derived from engine data.

Engine model simulations run for 1,500 cycles. Residual gas fraction $F(k) = 0.9$ and stoichiometric fuel-air ratio is $R = 15.13$. Adaptation gains for weight updating are selected as $\alpha_c = 0.09$ and $\alpha_a = 0.09$. Cost function parameters are Q=0.1 and R=0.1. The action NN is set with 20 hidden layer nodes, and the critic NN is set with 20 hidden layer nodes. The neuron activation functions are hyperbolic tangent sigmoids in order to ensure the NN approximation capability.

The simulation results are given in return maps of heat release. Simulation returns heat release for every engine cycle simulated. For a set of heat release data, the current cycle, *k*, is plotted against the next cycle, *k+1*, to show the cyclic dispersion of the heat release from cycle to cycle. Ideally, the heat release data in a return map would be concentrated on the diagonal axis. This would indicate that current-cycle and next-cycle heat release are the same, but that is not the case due to cycle-to-cycle effects of residuals within the cylinder and the dynamic behavior of combustion. Units of heat release on each axis are in Joules that are scaled to match the Ricardo engine data recorded for training the combustion efficiency function with spark timing.

The first simulation depicts the control at a lean equivalence ratio of 0.78. COV (coefficient of variation) is the metric used to determine a reduction in the cyclic dispersion between heat release return maps. A reduced COV value indicates a reduction in cyclic dispersion. In Fig. 3 the return map for the uncontrolled case is shown and in Fig. 4 the return map for the controlled case is shown. A modest drop from 0.103 to 0.094 in COV is shown.

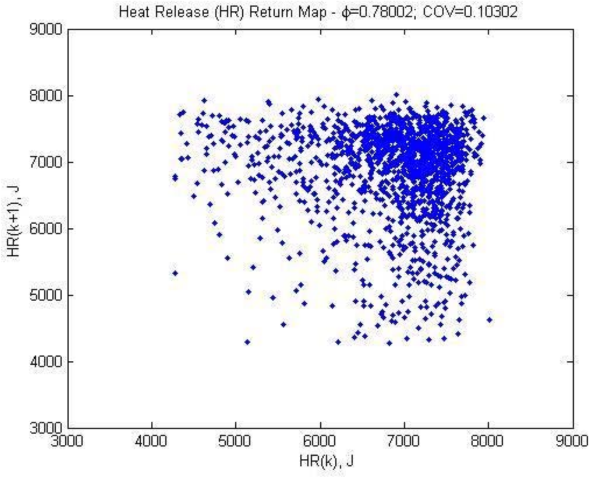

Fig. 3. Simulation at equivalence ratio 0.78 with open-loop control.

The second simulation shows control at lean equivalence ratio 0.85. In Fig. 5 the return map for the uncontrolled case is shown and in Fig. 6 the return map for the controlled case is shown. A reduction from 0.043 to 0.041 in COV is shown.

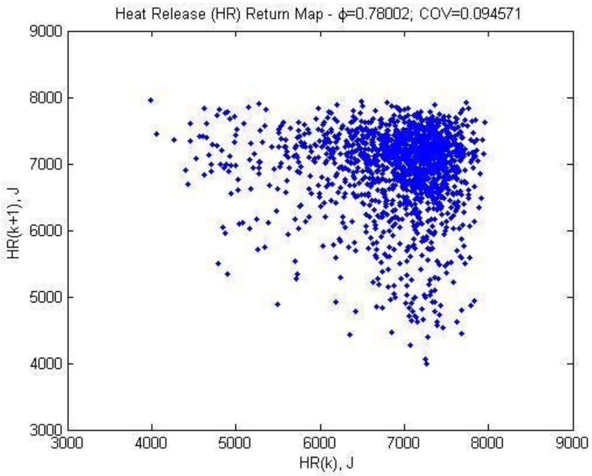

Fig. 4. Simulation of equivalence ratio 0.78 with closed-loop control.

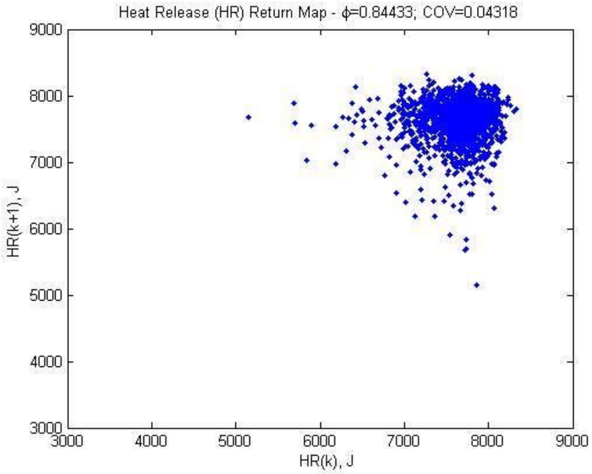

Fig. 5. Simulation at equivalence ratio 0.85 with open-loop control.

Table II contains the COV values for the reported simulation results and the corresponding percent reduction. The results indicate a modest improvement in reduction in cyclic dispersion in heat release with control, which demonstrates the satisfactory performance of the controller.

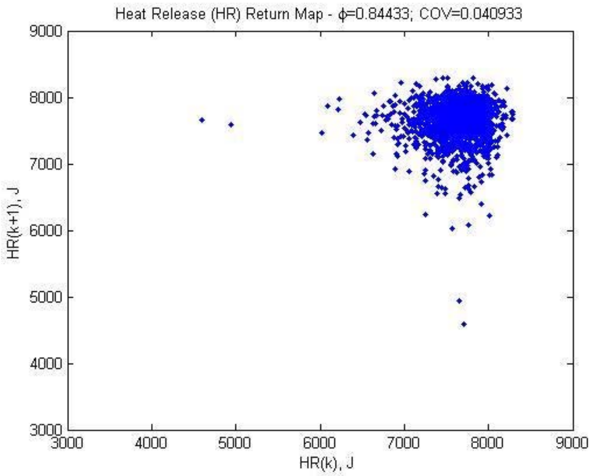

Fig. 6. Simulation at equivalence ratio 0.85 with closed-loop control.

Table II. COV Reduction

Equivalence ratio φ	COV open-loop	COV closed-loop	% COV reduced
0.78	0.103	0.094	8.7%
0.85	0.043	0.041	4.7%

B. Second-order Nonlinear Discrete-time System

Another simulation using the controller designed in this paper involves a fictional second-order discrete-time system as shown in the following equation.

$$\begin{aligned} \mathrm{y}(\mathrm{k}+1) = {} & 0.2\cos\left(0.8\left(y(k)+y(k-1)\right)\right) \\ & +0.4\sin\left(0.8\left(y(k)+y(k+1)\right)+2u(k)+u(k-1)\right) \\ & +0.1\left(9+y(k)+y(k-1)\right) \\ & +\frac{2\left(u(k)+u(k-1)\right)}{1+\cos\left(y(k)\right)}+d(k) \end{aligned} \tag{53}$$

One can see that the system has a nonaffine control input that appears within functions of this system. In the following simulation, the controller must have the system in (53) track the reference signal defined as

$$y_d\left(k\right)=0.8+0.2\sin\left(2\pi kT\right) \tag{54}$$

where $T=0.01$ seconds. The simulation runs for 25 seconds and the controller parameters are selected as γ =0.8, α_a =0.01, α_c =0.09, Q=0.01, R=0.01, and d_M =0.01. Both the critic NN and the adaptive NN are given 10 hidden-layer nodes. Fig. 7 shows the output of the simulation where the desired and actual outputs are plotted together. There is no constraint placed on the input in Fig. 7, and the system output tracks the desired signal with small error.

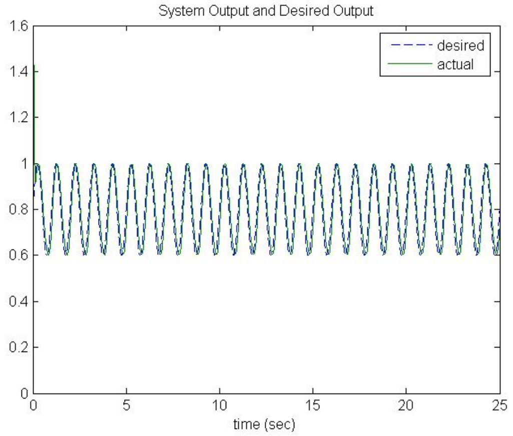

Fig. 7. Output tracking of system given in equation (53).

To check the performance of the controller, a bound of $\left|u\right|<0.25$ was placed on the control input. There is some error visible in tracking the desired signal of Fig. 8 since the control input is bounded as seen in Fig. 9. After learning the desired signal, the controller is able to continue tracking even with the input limit constraint.

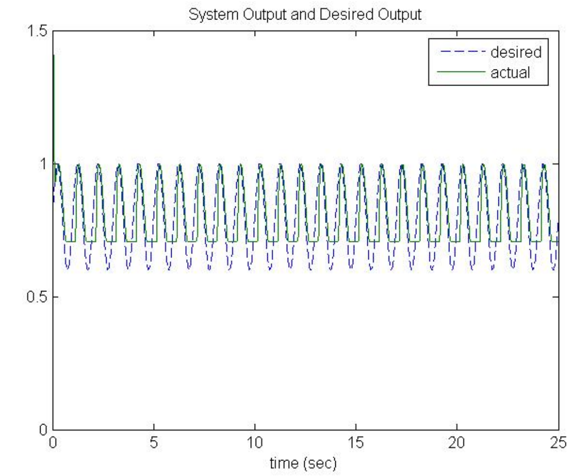

Fig. 8. Output tracking of system given in equation (53) with input constraint applied.

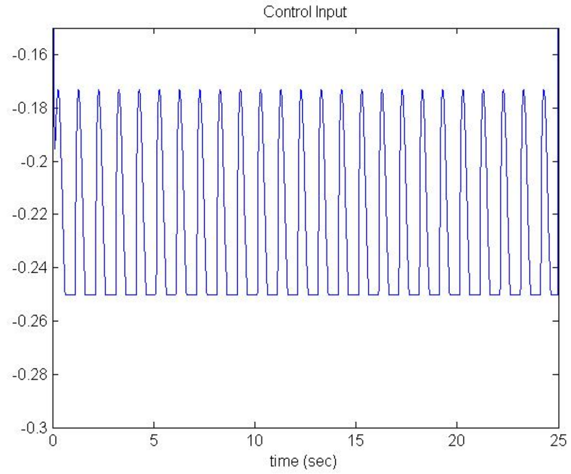

Fig. 9. Control input of simulation performed on system equation (53) with input constraint.

In addition to the limited inputs seen in Fig.9., the critic NN weights are shown in Fig. 10. The norm of the critic weights are visibly bounded by looking at the Fig. 10. This suggests that the controller is stable.

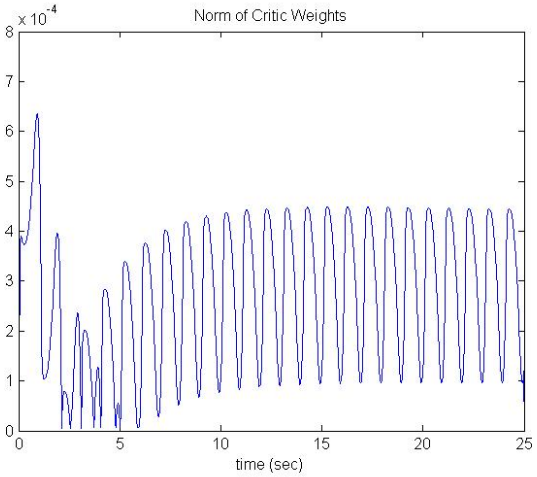

Fig. 10. Norm of critic NN weights from simulation of equation (53) with input constraint.

VII. Conclusions

The model integrates all higher order terms of the Taylor expansion without losing any information. Bounded disturbance is also considered within the model. Based on this effort, a novel reinforcement online learning scheme is designed to deliver a desired performance by using neural networks. The controller is updated in an online fashion without offline phase. To guarantee the boundedness of the closed-loop signals for this approximation based optimal controller methodology, the uniform ultimate boundedness of the closed-loop tracking errors and NN weight estimates is verified by using standard Lyapunov analysis in the presence of bounded disturbances and approximation errors.

Simulation on both a nonlinear engine model with nonaffine control input and a second order nonlinear model based on sinusoids with nonaffine control input demonstrates that the controller with input constraints imposed can track a desired signal. The controller remains stable even with limitations placed on the input.

Appendix

Proof of Theorem 1: Define a Lyapunov candidate as

$$\begin{aligned} L(k) = \sum_{i=1}^{4} L_i = \frac{\gamma_1}{3} \sum_{j=0}^{\tau-1} e^2(k+j) + \frac{\gamma_2}{\alpha_a} \sum_{j=0}^{\tau-1} tr\left(\tilde{W}_a^T(k+j)\tilde{W}_a(k+j)\right) \\ + \frac{\gamma_3}{\alpha_c} \sum_{j=0}^{\tau-1} tr\left(\tilde{W}_c^T(k+j)\tilde{W}_c(k+j)\right) + \gamma_4 \sum_{j=0}^{\tau-1} \zeta_c{}^2(k+j) \end{aligned} \tag{A.1}$$

where $\gamma_i \in R^+$, $i = 1,2,3,4$ are design parameters. Hence, the first difference of the Lyapunov function is given by

$$\begin{aligned} \Delta L_1 &= \frac{\gamma_1}{2}\left(e^2(k+\tau) - e^2(k)\right) \\ &= \frac{\gamma_1}{2}\left((\kappa_k \zeta_a(k) + d_a(k))^2 - e^2(k)\right) \\ &\le -\frac{\gamma_1}{2} e^2(k) + \gamma_1 g_{\max}^2 \zeta_a^2(k) + \gamma_1 d_a^2(k) \end{aligned} \tag{A.2}$$

$$\begin{aligned} \Delta L_2 &= \frac{\gamma_2}{\alpha_a} tr\left(\tilde{W}_a^T(k+\tau)\tilde{W}_a(k+\tau) - \tilde{W}_a^T(k)\tilde{W}_a(k)\right) \\ &= \frac{\gamma_2}{\alpha_a} tr\left(2\tilde{W}_a^T(k+\tau)\Delta\hat{W}_a(k) + \Delta\hat{W}_a^T(k)\Delta\hat{W}_a(k)\right) \\ &= \frac{\gamma_2}{\alpha_a} tr\left(-2\tilde{W}_a^T(k+\tau)\alpha_a\phi_a(k)\left(\kappa_k\zeta_a(k) + \hat{J}(k) + d_a(k)\right)\right. \\ &\quad \left. + \Delta\hat{W}_a^T(k)\Delta\hat{W}_a(k)\right) \\ &= -2\gamma_2\kappa_k\zeta_a^2(k) - 2\gamma_2\zeta_a(k)\left(\hat{J}(k) + d_a(k)\right) \\ &\quad + \gamma_2\alpha_a\phi_a^2(k)\left(\kappa_k\zeta_a(k) + \hat{J}(k) + d_a(k)\right)^2 \end{aligned}$$

$$
\begin{aligned}
&\le -2\gamma_2 g_{\min}\zeta_a^2(k) - 2\gamma_2\zeta_a(k)\left(\hat{J}(k)+d_a\left(k\right)\right)\\
&\quad +\gamma_2\alpha_a\phi_a^2(k)g_{\max}^2\zeta_a^2(k) + \gamma_2\alpha_a\phi_a^2(k)\\
&\quad \times\left(\left(\hat{J}(k)+d_a(k)\right)^2 + 2\left(\hat{J}(k)+d_a(k)\right)\kappa_k\zeta_a(k)\right)\\
&= \gamma_2\left\{-g_{\min}\zeta_a^2(k) - \left(g_{\min}-\alpha_a\phi_a^2(k)g_{\max}^2\right)\zeta_a^2(k)\right.\\
&\quad -2\zeta_a(k)\left(I-\alpha_a\phi_a^2(k)\kappa_k\right)\left(\hat{J}(k)+d_a(k)\right)\\
&\quad \left.+\alpha_a\phi_a^2(k)\left(\hat{J}(k)+d_a(k)\right)^2\right\}\\
&= \gamma_2\left\{-g_{\min}\zeta_a^2(k) - \left(g_{\min}-\alpha_a\phi_a^2(k)g_{\max}^2\right)\right.\\
&\quad \times\left(\zeta_a(k)+\frac{\left(I-\alpha_a\phi_a^2(k)\kappa_k\right)}{g_{\min}-\alpha_a\phi_a^2(k)g_{\max}^2}\right)^2\\
&\quad \left.+\frac{1-\alpha_a\phi_a^2(k)g_{\min}}{g_{\min}-\alpha_a\phi_a^2(k)g_{\max}^2}\left(\hat{J}(k)+d_a(k)\right)^2\right\}
\end{aligned}
\qquad (A.3)
$$

Set $\gamma_2 = \gamma_2'\gamma_2''$, where $\gamma_2'' \dfrac{1-\alpha_a\phi_a^2(k)g_{\min}}{g_{\min}-\alpha_a\phi_a^2(k)g_{\max}^2} \le \dfrac{1}{2}$, therefore,

$$
\begin{aligned}
\Delta L_2 &\le -\gamma_2 g_{\min}\zeta_a^2(k) - \gamma_2\left(g_{\min}-\alpha_a\phi_a^2(k)g_{\max}^2\right)\\
&\quad \times\left(\zeta_a(k)+\frac{\left(I-\alpha_a\phi_a^2(k)\kappa_k\right)}{g_{\min}-\alpha_a\phi_a^2(k)g_{\max}^2}\right)^2 + \gamma_2'\frac{\left(\hat{J}(k)+d_a(k)\right)^2}{2}\\
&\le -\gamma_2 g_{\min}\zeta_a^2(k) - \gamma_2\left(g_{\min}-\alpha_a\phi_a^2(k)g_{\max}^2\right)\\
&\quad \times\left(\zeta_a(k)+\frac{\left(I-\alpha_a\phi_a^2(k)\kappa_k\right)}{g_{\min}-\alpha_a\phi_a^2(k)g_{\max}^2}\right)^2 + \gamma_2'\zeta_c^2(k)\\
&\quad +\gamma_2'\left(J^*(k)+d_a(k)\right)^2
\end{aligned}
\qquad (A.4)
$$

At the same time,

$$
\begin{aligned}
\Delta L_3 &= \frac{\gamma_3}{\alpha_c} tr\left(\tilde{W}_c^T(k+\tau)\tilde{W}_c(k+\tau) - \tilde{W}_c^T(k)\tilde{W}_c(k)\right) \\
&= \frac{\gamma_3}{\alpha_c} tr\left(-2\tilde{W}_c^T(k)\alpha_c\gamma\phi_c(k)e_c(k)\right) + \frac{\gamma_3}{\alpha_c} tr\left(\Delta\hat{W}_c^T(k)\Delta\hat{W}_c(k)\right) \\
&= -2\gamma_3\gamma\zeta_c(k)e_c\left(k\right) + \gamma_3\alpha_c\gamma^2 e_c^2(k)\phi_c^2(k) \\
&= -2\gamma_3 e_c(k)\left(e_c(k) - \gamma J*(k) + \zeta_c(k-1) + J*(k-1) - r(k)\right. \\
&\quad \left.+\varepsilon_c(k) - \varepsilon_c(k-1)\right) + \gamma_3\alpha_c\gamma^2 e_c^2(k)\phi_c^2(k) \\
&= -\gamma_3\left(1-\alpha_c\gamma^2\phi_c^2(k)\right)e_c^2(k) - \gamma_3 e_c^2(k) + 2\gamma_3 e_c(k)\left(\gamma J*(k)\right. \\
&\quad \left.-\zeta_c(k-1) - J*(k-1) + r(k) - \varepsilon_c(k) + \varepsilon_c(k-1)\right) \\
&= -\gamma_3\left(1-\alpha_c\gamma^2\phi_c^2(k)\right)e_c^2(k) - \gamma_3\gamma^2\zeta_c^2(k) \\
&\quad +\gamma_3\left(\gamma J*(k) - \zeta_c(k-1) - J*(k-1) + r(k) - \varepsilon_c(k) + \varepsilon_c(k-1)\right)^2 \\
&\le -\gamma_3\left(1-\alpha_c\gamma^2\phi_c^2(k)\right)e_c^2(k) - \gamma_3\gamma^2\zeta_c^2(k) + \frac{\gamma_3}{4}\zeta_c^2(k-1) \\
&\quad +\frac{\gamma_3}{4}\left(\gamma J*(k) - J*(k-1)\right)^2 + \frac{\gamma_3}{4}r(k) + \frac{\gamma_3}{4}\left(\varepsilon_c(k) - \varepsilon_c(k-1)\right)^2 \\
&\le -\gamma_3\left(1-\alpha_c\gamma^2\phi_c^2(k)\right)e_c^2(k) - \gamma_3\gamma^2\zeta_c^2(k) + \frac{\gamma_3}{4}\zeta_c^2(k-1) \\
&\quad +\frac{\gamma_3}{4}\left(\gamma J*(k) - J*(k-1)\right)^2 + \frac{\gamma_3}{4}Qe^2(k) \\
&\quad +\frac{\gamma_3}{4}\left(\zeta_a(k) + w_a{}^T\phi_a(k)\right)^T R\left(\zeta_a(k) + w_a{}^T\phi_a(k)\right) + \gamma_3\varepsilon_{cm}{}^2 \\
&\le -\gamma_3\left(1-\alpha_c\gamma^2\phi_c^2(k)\right)e_c^2(k) - \gamma_3\gamma^2\zeta_c^2(k) + \frac{\gamma_3}{4}\zeta_c^2\left(k-1\right) \\
&\quad +\frac{\gamma_3}{4}\left(\gamma J*(k) - J*(k-1)\right)^2 + \frac{\gamma_3}{4}Qe^2(k) + \frac{\gamma_3}{8}R\zeta_a^2(k) \\
&\quad +\frac{\gamma_3}{8}R\left(w_a{}^T\phi_a(k)\right)^2 + \gamma_3\varepsilon_{cm}{}^2
\end{aligned}
\tag{A.5}
$$

$$
\Delta L_4 = \gamma_4\left(\zeta_c^2(k) - \zeta_c^2(k-1)\right) \tag{A.6}
$$

Combining (A.1) - (A.6) yields

$$\begin{aligned}
\Delta L(k) \le & -\frac{\gamma_1}{3} e^2(k) + \gamma_1 g_{\max}^2 \zeta_a^2(k) + \gamma_1 d_a^2(k) \\
& -\gamma_2 g_{\min} \zeta_a^2(k) - \gamma_2 \left(g_{\min} - \alpha_a \phi_a^2(k) g_{\max}^2 \right) \\
& \times \left(\zeta_a(k) + \frac{\left(1 - \alpha_a \phi_a^2(k) \kappa_k\right)}{g_{\min} - \alpha_a \phi_a^2(k) g_{\max}^2} \right)^2 + \gamma_2' \zeta_c^2(k) \\
& + \gamma_2' \left(J^*(k) + d_a(k) \right)^2 - \gamma_3 \left(1 - \alpha_c \gamma^2 \phi_c^2(k) \right) e_c^2(k) \\
& - \gamma_3 \gamma^2 + \frac{\gamma_3}{4} \zeta_c^2(k-1) + \frac{\gamma_3}{4} \left(\gamma J^*(k) - J^*(k-1) \right)^2 \\
& + \frac{\gamma_3}{4} Q e^2(k) + \frac{\gamma_3}{8} R \zeta_c^2(k) \\
& + \frac{\gamma_3}{8} R \left(w_a{}^T \phi_a(k) \right)^2 + \gamma_3 \varepsilon_{cm}{}^2 + \gamma_4 \left(\zeta_c^2(k) - \zeta_c^2(k-1) \right) \\
= & -\left(\frac{\gamma_1}{2} - \frac{\gamma_3}{4} Q \right) e^2(k) \\
& -\left(\gamma_2 g_{\min} - \gamma_1 g_{\max}^2 - \frac{\gamma_3}{8} R \right) \zeta_a^2(k) \\
& -\left(\gamma_3 \gamma^2 - \gamma_2' - \gamma_4 \right) \zeta_c^2(k) - \left(\gamma_4 - \frac{\gamma_3}{4} \right) \zeta_c^2(k-1) \\
& -\gamma_3 \left(1 - \alpha_c \gamma^2 \phi_c^2(k) \right) e_c^2\left(k\right) - \gamma_2 \left(g_{\min} - \alpha_a \phi_a^2(k) g_{\max}^2 \right) \\
& \times \left(\zeta_a(k) + \frac{\left(1 - \alpha_a \phi_a^2(k) \kappa_k\right)}{g_{\min} - \alpha_a \phi_a^2(k) g_{\max}^2} \right)^2 + D_M^2
\end{aligned} \tag{A.7}$$

where

$$\begin{aligned}
D_M^2 = & \left(\gamma_1 + \frac{\gamma_2'}{2} \right) d_{am}^2 + \left(\frac{\gamma_2'}{2} + \frac{\gamma_3}{4} (\gamma + 1)^2 \right) J_M^2 \\
& + \frac{\gamma_3}{8} R w_{am}^2 \phi_{am}^2 + \gamma_3 \varepsilon_{cm}{}^2
\end{aligned} \tag{A.8}$$

For the standard Lyapunov analysis, equation (A.7) and (A.8) implies that $\Delta L \le 0$ as long as the conditions (32) – (35) are satisfied and following holds

$$\left\| e(k) \right\| \ge \frac{2 D_M}{\sqrt{2\gamma_1 - \gamma_3 Q}} \tag{A.9}$$

or

$$\left\|\zeta_a\left(k\right)\right\| \le \frac{2\sqrt{2}D_M}{\sqrt{8\gamma_2 g_{\min} - 8\gamma_1 g_{\max}^2 - \gamma_3 R}} \tag{A.10}$$

or

$$\left\|\zeta_c\left(k\right)\right\| \le \frac{D_M}{\sqrt{\gamma_3\gamma^2 - \gamma_2' - \gamma_4}} \tag{A.11}$$

According to the standard Lyapunov extension theorem [12], the analysis above demonstrates that the tracking error $\|e(k)\|$ and the weights of the estimation errors are UUB. Further, the boundedness of $\left\|\zeta_a\left(k\right)\right\|$ and $\left\|\zeta_c\left(k\right)\right\|$ implies that the weight estimations $\|\hat{w}_a(k)\|$ and $\|\hat{w}_c(k)\|$ are also bounded.

Proof of Theorem 2: The proof of Theorem 2 is similar to that of Theorem 1. Alternatively, define the Lyapunov candidate as

$$\begin{aligned} L(k) = \sum_{i=1}^{4} L_i = \frac{\gamma_1}{3}\sum_{j=0}^{\tau-1} e_u^2(k+j) + \frac{\gamma_2}{\alpha_a}\sum_{j=0}^{\tau-1} tr\left(\tilde{W}_a^T(k+j)\tilde{W}_a(k+j)\right) \\ + \frac{\gamma_3}{\alpha_c}\sum_{j=0}^{\tau-1} tr\left(\tilde{W}_c^T(k+j)\tilde{W}_c(k+j)\right) + \gamma_4\sum_{j=0}^{\tau-1}\zeta_c{}^2(k+j) \end{aligned} \tag{A.12}$$

The first difference of it is also similar to (A.7), except that the tracking error, $e(k)$, is replaced by $e_u(k)$. Thus, $\Delta L(k) \le 0$ as long as (48) – (50) are satisfied and

$$\|e_u(k)\| \ge \frac{2D_M}{\sqrt{2\gamma_1 - \gamma_3 Q}} \tag{A.13}$$

or

$$\left\|\zeta_a\left(k\right)\right\| \le \frac{2\sqrt{2}D_M}{\sqrt{8\gamma_2 g_{\min} - 8\gamma_1 g_{\max}^2 - \gamma_3 R}} \tag{A.14}$$

or

$$\left\|\zeta_c\left(k\right)\right\| \le \frac{D_M}{\sqrt{\gamma_3 \gamma^2 - \gamma_2' - \gamma_4}} \tag{A.15}$$

According to standard Lyapunov extension theorem [12], this demonstrates that the auxiliary error and the error in weight estimates are UUB. Further, the boundedness of $\left\|\zeta_a\left(k\right)\right\|$ and $\left\|\zeta_c\left(k\right)\right\|$ implies the boundedness of the weight estimates $\left\|\hat{w}_a(k)\right\|$ and $\left\|\hat{w}_c(k)\right\|$.

Proof of Corollary 1: The proof of Corollary 1 directly follows Theorem 2.

Case 1: $\left\|v(k)\right\| \le u_{\max}$

If $\left\|v(k)\right\| \le u_{\max}$, then $u(k) = v(k)$ and $\Delta u(k) = u(k) - v(k) = 0$. Hence, (54) implies $e(k) = e_u(k)$, and the boundedness of $e(k)$ directly follows the result of Theorem 1.

Case 2: $\left\|v(k)\right\| > u_{\max}$

If $\left\|v(k)\right\| > u_{\max}$, then $u(k) = u_{\max}\,\mathrm{sgn}(v(k))$. As the boundedness of the weight estimates $\left\|\hat{w}_a(k)\right\|$ over a compact set is guaranteed, the auxiliary control input $v(k)$ is also bounded, which implies the boundedness of $\Delta u(k)$. Notice from $0 < g_{\min} \le \kappa_k \le g_{max}$ and (54), the tracking error, $e(k)$, is also bounded.

References

[1] S. A. Billings and W. S. F. Voon, “A prediction-error and stepwise regression algorithm for nonlinear systems,” International Journal of Control ,44, pp. 803–822, 1986.

[2] O. Adetona, E. Garcia, and L. H. Keel, “A new method for the control of discrete nonlinear dynamic systems using neural networks,” IEEE Trans. Neural Networks, vol. 11, pp. 102–112, Jan. 2000.

[3] O. Adetona, S. Sathananthan, and L. H. Keel, “Robust adaptive control of nonaffine nonlinear plants with small input signal changes,” IEEE Trans. Neural Networks, vol. 15, pp. 408–416, Mar. 2004.

[4] J. Si, A. G. Barto, W. B. Powell, and D. Wunsch, Eds., “Handbook of Learning and Approximate Dynamic Programming,” Wiley-IEEE Press, 2004.

[5] D. Prokhorov and D. Wunsch, “Adaptive critic designs,” IEEE Trans. Neural Networks, Vol. 8, No.5, p.997-1007, 1997.

[6] P. J. Werbos, “Building and understanding adaptive systems: A statistical/numerical approach to factory automation and brain research,” IEEE Transactions on Systems, Man, and Cybernetics 17, pp. 7–20, 1987.

[7] J. Si and Y. T. Wang, “On-line learning control by association and reinforcement,” IEEE Trans. Neural Networks, vol. 12, no. 2, pp. 264–276, Mar.2001.

[8] Qinmin Yang and S. Jagannathan, “Online reinforcement learning-based neural network controller design for affine nonlinear discrete-time systems,” Proc. of American Control Conference, to appear 2007.

[9] M. S. Ahmed, “Neural-net-based direct adaptive control for a class of nonlinear plants,” IEEE Trans. On Automatic Control, vol. 45, pp. 119–124, 2000.

[10] D. P. Bertsekas, “Dynamic Programming and Optimal Control. Belmont,” MA: Athena Scientific, 2000.

[11] B. Igelnik and Y. H. Pao, “Stochastic choice of basis functions in adaptive function approximation and the functional-link net,” IEEE Trans. Neural Network, vol. 6, no. 6, pp. 1320–1329, Nov. 1995.

[12] S. Jagannathan, “Neural Network Control of Nonlinear Discrete-time Systems,” Taylor and Francis (CRC Press), Boca Raton, FL 2006.

[13] J.B. Heywood, *Internal combustion engine fundamentals*, MGraw-Hill, New York, 1998, pp. 425-426.

[14] J. B. Vance, B. C. Kaul, S. Jagannathan, and J. A. Drallmeier, “Output Feedback Controller for Operation of Spark Ignition Engines at Lean Conditions Using Neural Networks,” *IEEE Transaction on Control Systems and Technology,* to appear in 2007.

[15] J. B. Vance, A. Singh, B. C. Kaul, S. Jagannathan, and J. A. Drallmeier, “Neural Network Controller Development and Implementation for Spark Ignition Engines with High EGR Levels,” *IEEE Transaction on Neural Networks*, vol. 18, no. 4, July 2007.

[16] Stephen G. Russ, George A. Lavoie, and Wen Dai, “SI Engine Operation with Retarded Ignition: Part 1 – Cyclic Variations,” *SAE Technical Papers*, no. 1999-01-3506, 1999.

[17] A. Ramesh, O. Le Corre, and M. Tazerout, “Experimental Investigation on Cycle-By-Cycle Variations in a Natural-Gas-Fuelled Spark Ignition Engine,” *SAE Technical Papers*, no. 2001-28-0021, 2001.

[18] Environmental Protection Agency: “Air Quality Criteria for Ozone and Other Photochemical Oxidants,” EPA-600/8-84-020cF, 1986.

[19] 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.

SECTION

2. CONCLUSION

Results from the papers of this dissertation show that the designed controller schemes demonstrate good tracking performance and meet closed-loop stability requirements. The NN approximation property is used many times. To guarantee the boundedness of the closed-loop signals for these controller methodologies, the boundedness of the closed-loop tracking errors and NN weight estimates is verified by using standard Lyapunov analysis in the presence of bounded disturbances and approximation errors.

Simulation results by using the experimentally validated spark ignition engine model show that the performance of the proposed controller is highly satisfactory while meeting the closed loop stability even though all of the system dynamics are not known beforehand. When the controller design techniques are applied to a spark ignition engine model, the closed-loop system is able to reduce cyclic dispersion of heat release. Whether considering state feedback of the system states of fuel and air, or using output feedback of the heat release output, the controller shows satisfactory performance and stability. The approximation error of the neural networks is shown to be bounded using Lyapunov techniques. Estimated states of fuel and air enable controller design to follow backstepping technique.

Experimental data for the controllers employing output feedback, shows that real-time performance is achieved when applied to a spark ignition engine. Over several engine cycles, cyclic dispersion of heat release output is reduced as intended. The result is verified with fuel as the control input and for both cases where an EGR control loop is considered and omitted. NN weights are bounded and the overall engine system is stable.

Extending this dissertation to future work should include an examination of active EGR control where the amount of EGR introduced to the cylinder at each cycle is controlled by a pin valve apparatus at the intake. Similar to the perturbation of fuel from a nominal value, the EGR would deviate from a nominal value to influence the combustion. An output feedback control scheme would be applicable to this problem where the heat release of each engine cycle is obtained. The affine control input EGR appears in the inert gas state equation of the spark ignition engine model.

Further work with the results of this dissertation would lead to a control scheme where fuel, EGR, and spark timing are controlled together. Controlling those inputs simultaneously would require that stability is demonstrated for a feedback system with three control inputs that affect each other – one of which being nonaffine in the system model.

More experimental results are needed to form a picture of how this work can be applied to production automotive spark ignition engines. All experimental results are from single-cylinder research engines operating at a speed held fixed by a dynamometer. Data should be obtained from a multi-cylinder engine, but this will add new dynamics to the system as all of the cylinders are mechanically linked and undergo unique combustion. Finally, variation of the engine load over time to simulate driving conditions would also be required to evaluate how these controllers might perform on a production automotive engine. Spark ignition engine system changes over time are difficult to predict. The mid-load condition would experience hysterisis as transitions come from either high-load or low-load conditions.

Results show that the controller can improve spark ignition engine performance in controlled laboratory conditions. Extended controller development with an EGR input and with all inputs considered at once may lead to better performance. To supplement this dissertation, additional controller research and experiments with multi-cylinder and variable-load tests will uncover new and useful conclusions.

VITA

Jonathan Blake Vance was born January 26, 1981 in Metairie, Louisiana. He earned a Bachelor's of Science in Computer Engineering and a Bachelor's of Science in Electrical Engineering in 2003, a Master's of Science in Computer Engineering in 2005, and a Ph. D. in Electrical Engineering in December 2007, all from the University of Missouri-Rolla. He is a member of IEEE – Institute of Electrical and Electronics Engineers, Eta Kappa Nu – Electrical and Computer Engineering Honor Society, and Tau Beta Pi – The Engineering Honor Society.