View metadata, citation and similar papers at core.ac.uk

brought to you by CORE

provided by Missouri University of Science and Technology (Missouri S&T): Scholars' Mine

Missouri University of Science and Technology
Scholars' Mine

Electrical and Computer Engineering Faculty Research & Creative Works

Electrical and Computer Engineering

01 Jan 2006

Neural Network-Based Output Feedback Controller for Lean Operation of Spark Ignition Engines

Brian C. Kaul

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

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

Jonathan B. Vance

et. al. For a complete list of authors, see https://scholarsmine.mst.edu/ele_comeng_facwork/1169

Follow this and additional works at: https://scholarsmine.mst.edu/ele_comeng_facwork

Part of the Aerospace Engineering Commons, Computer Sciences Commons, Electrical and Computer Engineering Commons, Mechanical Engineering Commons, and the Operations Research, Systems Engineering and Industrial Engineering Commons

Recommended Citation

B. C. Kaul et al., "Neural Network-Based Output Feedback Controller for Lean Operation of Spark Ignition Engines," *Proceedings of the 2006 American Control Conference*, Institute of Electrical and Electronics Engineers (IEEE), Jan 2006.
The definitive version is available at https://doi.org/10.1109/ACC.2006.1656497

This Article - Conference proceedings is brought to you for free and open access by Scholars' Mine. It has been accepted for inclusion in Electrical and Computer Engineering Faculty Research & Creative Works by an authorized administrator of Scholars' Mine. This work is protected by U. S. Copyright Law. Unauthorized use including reproduction for redistribution requires the permission of the copyright holder. For more information, please contact scholarsmine@mst.edu.

Proceedings of the 2006 American Control Conference
Minneapolis, Minnesota, USA, June 14-16, 2006

Neural Network-based Output Feedback Controller for Lean Operation of Spark Ignition Engines

Jonathan B. Vance, P. He, Brian Kaul, S. Jagannathan, and James A. Drallmeier

Abstract— **Spark ignition (SI) engines running at very lean conditions demonstrate significant nonlinear behavior by exhibiting cycle-to-cycle dispersion of heat release even though such operation can significantly reduce NO_x emissions and improve fuel efficiency by as much as 5-10%. A suite of neural network (NN) controller without and with reinforcement learning employing output feedback has shown ability to reduce the nonlinear cyclic dispersion observed under lean operating conditions. The neural network controllers consists of three NN: a) A NN observer to estimate the states of the engine such as total fuel and air; b) a second NN for generating virtual input; and c) a third NN for generating actual control input. For reinforcement learning, an additional NN is used as the critic. The uniform ultimate boundedness of all closed-loop signals is demonstrated by using Lyapunov analysis without using the separation principle. Experimental results on a research engine at an equivalence ratio of 0.77 show a drop in NO_x emissions by around 98% from stoichiometric levels. A 30% drop in unburned hydrocarbons from uncontrolled case is observed at this equivalence ratio.**

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 HC. Current efforts aim to decrease emissions and minimize the fuel consumption. To address these requirements, lean combustion control technology has received increasing preference [1]. A difficulty of operating an engine at extreme lean conditions is that significant cyclic dispersion [2] in heat release is exhibited, causing engine instability and poor performance.

Several control schemes have been proposed to stabilize the 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.* [3] developed a feedback control approach, which uses the fuel as the control variable to reduce the cyclic dispersion. However, no system stability is guaranteed in both [1] and [3] since stability analysis for nonlinear unknown engine dynamics is difficult. He *et al.* [4] 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 and using total mass of air and fuel (system states), which are extremely difficult if not impossible to measure. In [5], another control scheme is presented using the state feedback.

Output feedback controller schemes are necessary when certain states of the plant become unavailable. Moreover, the separation principle does not hold for nonlinear systems, since an exponentially decaying state estimation error can lead to instability at finite time [6]. Consequently, the output feedback control design is quite difficult.

To make the controller implementation more practical, a heat-release-based neuro-output feedback controller is introduced in discrete-time to reach stable operation of a spark ignition (SI) engine at lean conditions. The 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 a NN controller. Non-catalytic SI engine designs (e.g. generator sets and other industrial applications) could make use lean operation to reduce engine-out NO_x and improve fuel efficiency.

Moreover, the proposed controller is designed for a class of nonlinear discrete-time systems in nonstrict feedback form. A persistency of excitation condition is not required, certainty equivalence and separation principle are not needed and linearity in the unknown parameter assumption is not used. A uniform ultimate boundedness (UUB) of all the signals is demonstrated. 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 [6-8, 10-11] where the system output is a known linear function.

II. CONTROLLER DESIGN

A. Background

1) Engine Dynamics

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

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

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

$$y(k) = x_2(k)CE(k), \quad (3)$$

This work was supported in part by the U.S. Department of Education GAANN Fellowship, Intelligent Systems Center and by the NSF grant #0327877.

J. B. Vance, P. He, S. Jagannathan are with the Department of Electrical and Computer Engineering, and J. A. Drallmeier is with the Department of Mechanical Engineering at the University of Missouri-Rolla (contact author's e-mail: jbvance@umr.edu).

1-4244-0210-7/06/$20.00 ©2006 IEEE

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

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

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: In system (1)-(3), states of $x_1(k)$ and $x_2(k)$ are typically immeasurable and only output $y(k)$ is 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 [6-8,10-11], 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 Another Form*

Substituting (3) into both (1) and (2), obtain equations

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

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

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. The bounds are given by

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

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

and

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

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

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

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

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

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

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

B. *NN Observer Design*

A neural network predicts the heat release in the subsequent time interval. The observer has 35 hidden layer nodes with sigmoid activation function. 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

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

where $z_1(k) = [x_1(k), x_2(k), y(k), u(k)]^T \in R^4$ 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 $\varepsilon(z_1(k)) \in R$ is the functional approximation error. As demonstrated in [12], 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 $\varepsilon(z_1(k))$ can be made arbitrarily small over the compact set since the activation function forms a basis.

For simplicity define

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

and

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

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

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

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

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

where $\hat{y}(k+1)$ is the predicted heat release, $\hat{w}_1(k)\in R^{n_1}$ are output layer weights, $\hat{z}_1(k)=[\hat{x}_1(k),\hat{x}_2(k),\hat{y}(k),u(k)]^T\in R^4$ 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 $\phi_1(\hat{z}_1(k))$ 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

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

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

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

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

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

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

and

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

where

$$\tilde{w}(k)=\hat{w}(k-1)-w_1\,,\tag{30}$$

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

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

C. Adaptive NN Output Feedback Controller

To stabilize the engine due to cyclic dispersion in heat release at lean conditions, the control objective is to 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_{21d}, $y(k)$ 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 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

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

For simplicity, 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

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

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

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

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

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

where the input is 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 hidden layer nodes, the hidden layer activation function of the input and hidden layer weights, $\phi_2\left(v_2^Tx(k)\right)$, is abbreviated 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

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

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

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

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

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

or, equivalently,

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

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

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

where

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

and

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

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

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

for simplicity,

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

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 since a first order predictor is sufficient to obtain this value. The feedforward NN with the proposed weight tuning generates a dynamic NN which is used to obtain the future value. Alternatively, a first order filter is used to obtain the value in [9].

Using the second action neural network, we can now select the desired control input as

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

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 $\phi_3(z_3(k))$, $\varepsilon_3(z_3(k))$ 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

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

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

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

where

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

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

and

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

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 $\hat{w}_1(k)$, $\hat{w}_2(k)$, and $\hat{w}_3(k)$ 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

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

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

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

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

Assumption 2 (Bounded NN Approximation Error): The NN approximation errors $\varepsilon_1(z_1(k))$, $\varepsilon_2(x(k))$, 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 (1)-(3) and let the Assumptions 1 and 2 hold. Let the unknown disturbances be bounded by $|d_1(k)| \le d_{1m}$ and $|d_2(k)| \le d_{2m}$, respectively. Let the observer weight tuning be given by

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

with the virtual control NN weight tuning be provided by

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

and the control input weight be tuned by

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

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 $\hat{w}_1(k)$, $\hat{w}_2(k)$, and $\hat{w}_3(k)$ are UUB provided the design parameters are selected as:

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

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

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

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

Remark 3: Given specific values of R , F_0 , and ΔF_m , we can derive the design parameters of l_i, i=1,6,7. 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 NN control scheme and the weight updating rules in *Theorem 1* with the parameter selection based on (65) through (68), the state $x_2(k)$ approaches the operating point x_{2d} .

Remark 5: It is important to note that in this theorem there is no persistence of excitation condition, certainty equivalence and linearity in the unknown parameter assumptions for the NN observer and NN controller. In the proof, the Lyapunov function consists of the observer estimation errors, system errors, and the NN estimation errors and therefore separation principle is not used. The proof obviates the need for the certainty equivalence assumption, and it allows weight-tuning algorithms to be derived during the proof, not selected a priori.

III. Reinforcement Weight Updating

In this section, we develop alternate weight updating rules based on reinforcement learning where actor-critic architecture is utilized. The critic NN is trained online to approximate the *strategic* utility function (long-term system performance index). Then the critic signal, with a potential for estimating the future system performance, is employed to tune the two action NNs to minimize the strategic utility function and the unknown system estimation errors so that closed-loop stability is inferred.

A. The Strategic Utility Function

The utility function $p(k) \in \Re$ is defined based on the current system errors and it is given by

$$p(k) = \begin{cases} 0, & if \left(\left| \hat{e}_1(k) \right| + \left| \hat{e}_2(k) \right| \right) \le c\,, \\ 1, & otherwise \end{cases} \tag{69}$$

where $c \in \Re$ is a pre-defined threshold. The utility function $p(k)$ is viewed as the current system performance index; $p(k) = 0$ and $p(k) = 1$ refers to the good and poor tracking performance respectively.

The long-term system performance measure or the *strategic* utility function $Q(k) \in \Re$, is defined as

$$Q(k) = \alpha^N p(k+1) + \alpha^{N-1} p(k+2) + \cdots + \alpha^{k+1} p(N), \tag{70}$$

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

B. Design of the Critic NN

The critic NN is used to approximate the *strategic* utility function $Q(k)$. We define the prediction error as

$$e_c(k) = \hat{Q}(k) - \alpha\left(\hat{Q}(k-1) - \alpha^N p(k) \right), \tag{71}$$

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

$$\hat{Q}(k) = \hat{w}_3^T(k) \phi_3\left(v_3^T x(k) \right) = \hat{w}_3^T(k) \phi_3(k)\,, \tag{72}$$

and $\hat{Q}(k) \in \Re$ is the critic signal, $\hat{w}_3(k) \in \Re^{n_3}$ and $v_3 \in \Re^{2 \times n_3}$ represent the matrix of weight estimates, $\phi_3(k) \in \Re^{n_3}$ is the activation function vector in the hidden layer, n_3 is the number of the nodes in the hidden layer, and the critic NN input is given by $\hat{x}(k) \in \Re^2$. The objective function to be minimized by the critic NN is defined as

$$E_c(k) = \frac{1}{2} e_c^2(k) \cdot \tag{73}$$

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

$$\hat{w}_3(k+1) = \hat{w}_3(k) + \Delta \hat{w}_3(k), \tag{74}$$

where

$$\Delta \hat{w}_3(k) = \alpha_3 \left[-\frac{\partial E_c(k)}{\partial \hat{w}_3(k)} \right], \tag{75}$$

or

$$\hat{w}_3(k+1) = \hat{w}_3(k) - \alpha_3 \phi_3(k) \left(\hat{Q}(k) + \alpha^{N+1} p(k) - \alpha \hat{Q}(k-1) \right)^T, \tag{76}$$

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

C. Weight Updating Rule for the First Action NN

The first action NN $\hat{w}_1^T(k)\phi_1(k)$ weight is tuned by using the functional estimation error, $\zeta_1(k)$, and the error between the desired *strategic* utility function $Q_d(k) \in \Re$ and the critic signal $\hat{Q}(k)$. Define

$$e_{a1}(k) = \zeta_1(k) + (\hat{Q}(k) - Q_d(k)), \tag{77}$$

where $\zeta_1(k)$ is defined in (31), $e_{a1}(k) \in \Re$, and the subscript "a1" stands for the "first action NN".

The value for the desired *strategic* utility function $Q_d(k)$ is taken as "0" [9], i.e., to indicate that at every step, the nonlinear system can track the reference signal well. Thus, (34) becomes

$$e_{a1}(k) = \zeta_1(k) + \hat{Q}(k), \tag{78}$$

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

$$E_{a1}(k)=\frac{1}{2}e_{a1}^{2}(k), \tag{79}$$

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

$$\hat{w}_1(k+1)=\hat{w}_1(k)+\Delta\hat{w}_1(k), \tag{80}$$

where

$$\Delta\hat{w}_1(k)=\alpha_1\left[-\frac{\partial E_{a1}(k)}{\partial\hat{w}_1(k)}\right], \tag{81}$$

or

$$\hat{w}_1(k+1)=\hat{w}_1(k)-\alpha_1\phi_1(k)\left(\hat{Q}(k)+\zeta_1(k)\right), \tag{82}$$

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

The NN weight updating rule in (82) cannot be implemented in practice since the target weight w_1 is unknown. However, using (18), the functional estimation error $\zeta_1(k)$ is given by

$$\zeta_1(k)=-\hat{e}_1(k+1)-l_1\hat{e}_1(k)-\hat{e}_2(k)+\varepsilon_1(x(k))+d_1(k)\cdot \tag{83}$$

Substituting (83) into (82), we get

$$\begin{aligned}\hat{w}_1(k+1)&=\hat{w}_1(k)-\alpha_1\phi_1(k)\hat{Q}(k)\\&-\alpha_1\phi_1(k)\left(-\hat{e}_1(k+1)-l_1\hat{e}_1(k)-\hat{e}_2(k)+\varepsilon_1(x(k))+d_1(k)\right)\cdot\end{aligned} \tag{84}$$

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

$$\hat{w}_1(k+1)=\hat{w}_1(k)-\alpha_1\phi_1(k)\hat{Q}(k)+\alpha_1\phi_1(k)\left(\hat{e}_1(k+1)+l_1\hat{e}_1(k)+\hat{e}_2(k)\right)\cdot \tag{85}$$

Equation (85) is the adaptive critic based weight updating rule for the first action NN. Similarly, the weight updating rule for the second action NN $\hat{w}_2^T(k)\phi_2(k)$ is given next.

D. Weight Updating Rule for the Second Action NN

Define

$$e_{a2}(k)=\sqrt{g_2(k)}\zeta_2(k)+\frac{\hat{Q}(k)}{\sqrt{g_2(k)}}, \tag{86}$$

where $\zeta_2(k)$ is defined in (45), $g_2(k)\in\Re^+$ and $e_{a2}(k)\in\Re$, the subscript "a2" stands for the "second action NN". Following the similar design procedure and taking the bounded unknown disturbance $d_2(k)$ and the NN approximation error $\varepsilon_2(z(k))$ to be zeros, the second action NN $\hat{w}_2^T(k)\phi_2(k)$ weight updating rule is given by

$$\hat{w}_2(k+1)=\hat{w}_2(k)-\alpha_2\phi_2(k)\left(\hat{Q}(k)+\hat{e}_2(k+1)-l_2\hat{e}_2(k)\right), \tag{87}$$

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

IV. Controller Hardware Design

The research engine on which the controller operates is motored at 1000 RPM, and fires one cylinder to eliminate the dynamics introduced by multiple cylinders. 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.

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 before, during, and after combustion, which is a cycle window from 345° to 490°, for a total of 290 pressure measurements. At 1000 RPM pressure measurements must be made every 83.3 microseconds.

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 equipped with the engine. Pressure measurements come from a charge amplifier which receives pressure transducer signals from within 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 they are used to calculate heat release that is inputted to the controller algorithm. Fuel pulse width is adjusted from the nominal value according to the value returned from the controller algorithm and sent to the microcontroller. The received fuel pulse width is used in the following engine cycle.

The controller algorithm and data structures are implemented in C and compiled to run on an x86 PC. The hidden layer nodes parameter *n* was set to 35 for all of the NN for controller compilation. This value was chosen after experimentation showed that more nodes give no further improvement and that the hardware was sufficiently capable of processing the networks. Configuration files allow the controller gains and engine parameters to be modified without recompiling.

V. Experimental Results

The engine load is not varied during experimentation so that the performance of the controller on reducing emissions can be observed at a desired operating point. At 1000 RPM the pressure in the intake manifold is around 80 kPa which is roughly a mid-load operating condition. Full load would be atmospheric pressure and low load would be around 40 kPa. Since the controller is working to reduce effects of the unknown non-linear dynamics caused by residuals left in the cylinder after combustion, the engine speed was held constant.

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

$$\varphi=R\left(MF\middle/AF\right), \tag{88}$$

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 pressure measurements. This is necessary to remove any bias generated by the charge amplifier from which pressure is measured.

Uncontrolled and controlled heat release data were collected at equivalence ratios 0.79, 0.77, and 0.75. NO_x and unburned hydrocarbons (uHC) emissions data were also collected for both uncontrolled and controlled engine operation.

"Uncontrolled" 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" 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.

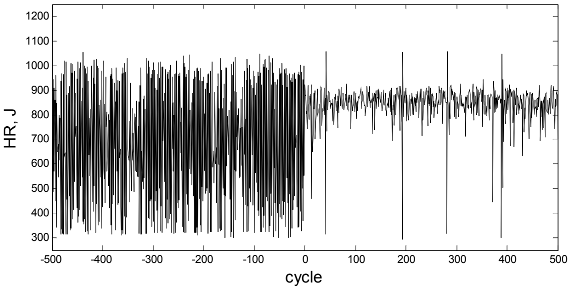

Fig. 1. Time series of heat release at equivalence ratio 0.77.

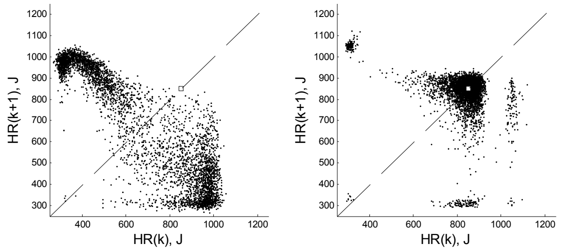

Fig. 2. Return maps of heat release at equivalence ratio 0.77.

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, the percentage increase in equivalence ratio during control is due to the mean value of fuel increasing from the nominal value injected for the cycles without controller operation.

Fig. 1 shows the time series of heat release for an equivalence ratio of 0.77. 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. 2 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 indicated work per cycle calculated for each return map.

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. At this equivalence ratio, coefficient of variation decreases from 38.7% to 13.6% when control has been applied. A decrease in cyclic dispersion is shown by the drop in coefficient of variation (COV).

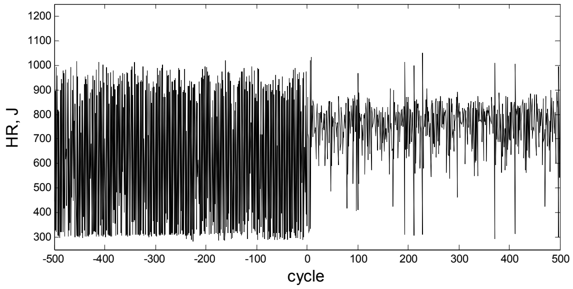

Fig. 3. Time series of heat release at equivalence ratio 0.75.

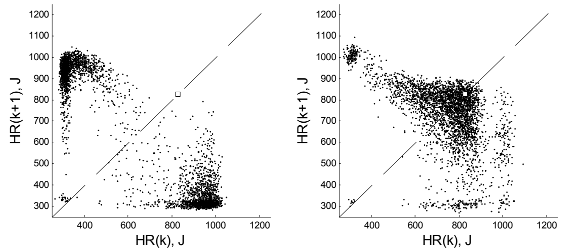

Fig. 4. Return maps of heat release at equivalence ratio 0.75.

Fig. 3 is a time series of heat release plot like that of Fig. 1, but the equivalence ratio has been lowered to 0.75. Notice that the heat release has become more unstable both with and without control. As equivalence ratio is further decreased, heat release becomes so unstable that misfires and partial burns are more prevalent than proper fires. Fig. 4 is the set of uncontrolled and controlled return maps obtained at equivalence ratio 0.75. Note that the COV decreases when control is applied. The cyclic dispersion decrease is evident.

The COV for all of the uncontrolled and controlled heat release return maps is shown in Table 1. For each equivalence ratio, the uncontrolled COV is greater than the uncontrolled COV since cyclic dispersion reduced when control was applied. The most significant decrease in cyclic dispersion was observed at an equivalence ratio 0.77. This reduction is 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 1. COEFFICIENT OF VARIATION FOR LEAN SET-POINTS

$\varphi_{\text{set-point}}$	Uncontrolled COV	Controlled COV
0.79	0.2302	0.2047
0.77	0.3851	0.1364
0.75	0.4631	0.2073

TABLE 2. EMISSIONS DATA FOR LEAN SET-POINTS

$\varphi_{\text{set-point}}$	(u) NO_x (PPM)	(c) NO_x (PPM)	(u) uHC (PPM)	(c) uHC (PPM)
0.79	159.01	351.17	81.417	77.678
0.77	92.833	48.199	387.31	283.70
0.75	130.00	54.512	913.29	386.09

Emissions data are given in Table 2. Exhaust gas analyzers were used to measured parts-per-million (PPM) of nitrogen oxides (NO_x) and PPM C_3 unburned hydrocarbons (uHC). The (u) and (c) prefixes in the column headings stand for uncontrolled and controlled, respectively. Looking at the uncontrolled and controlled data independently uHC increases as equivalence ratio decreases due to more abundant partial fuel burns. NO_x decreases at lower equivalence ratios because of lower combustion temperatures. uHC will tend to increase as equivalence ratio is decreased because of higher cyclic dispersion resulting from misfires and partial burns. To reduce uHC at lower equivalence ratios, cyclic dispersion must be decreased.

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 cannot 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.

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, 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.

[3] 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.

[4] P. He and S. Jagannathan, "Neuro emission controller for minimizing cyclic dispersion in spark ignition engines", in *Proc. Int. Joint Conf. Neural Networks*, Portland, OR, vol. 2, pp. 1535–1540, 2003.

[5] 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.

[6] Hassan K. Khalil, "Nonlinear Systems", 3rd Edition, Prentice Hall, 2002.

[7] 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.

[8] 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.

[9] 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.

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

[11] 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.

[12] 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.