Reinforcement-Learning-Based Output-Feedback Control of Nonstrict Nonlinear Discrete-Time Systems With Application to Engine Emission Control

Peter Shih, Brian C. Kaul, Sarangapani Jagannathan, *Senior Member, IEEE*, and James A. Drallmeier

***Abstract*—A novel reinforcement-learning-based output adaptive neural network (NN) controller, which is also referred to as the adaptive-critic NN controller, is developed to deliver the desired tracking performance for a class of nonlinear discrete-time systems expressed in nonstrict feedback form in the presence of bounded and unknown disturbances. The adaptive-critic NN controller consists of an observer, a critic, and two action NNs. The observer estimates the states and output, and the two action NNs provide virtual and actual control inputs to the nonlinear discrete-time system. The critic approximates a certain *strategic* utility function, and the action NNs minimize the *strategic* utility function and control inputs. All NN weights adapt online toward minimization of a performance index, utilizing the gradient-descent-based rule, in contrast with iteration-based adaptive-critic schemes. Lyapunov functions are used to show the stability of the closed-loop tracking error, weights, and observer estimates. Separation and certainty equivalence principles, persistency of excitation condition, and linearity in the unknown parameter assumption are not needed. Experimental results on a spark ignition (SI) engine operating lean at an equivalence ratio of 0.75 show a significant (25%) reduction in cyclic dispersion in heat release with control, while the average fuel input changes by less than 1% compared with the uncontrolled case. Consequently, oxides of nitrogen (NO_x) drop by 30%, and unburned hydrocarbons drop by 16% with control. Overall, NO_x's are reduced by over 80% compared with stoichiometric levels.**

***Index Terms*—Adaptive critic, discrete-time system, engine emission control, nonstrict nonlinear output feedback, reinforcement learning control.**

I. Introduction

ADAPTIVE neural network (NN) backstepping control of nonlinear discrete-time systems in strict feedback form given by

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

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

has been addressed in the literature [1]–[3], where $x_i(k) \in \Re$ is the state, $u(k) \in \Re$ is the control input, the term $\bar{x}_i(k) = [x_1(k), \ldots, x_i(k)]^{\mathrm{T}} \in \Re^i$, and $i = 1, \ldots, (n-1)$. For strict feedback nonlinear systems [1], the nonlinearities $f_i(\bar{x}_i(k))$ and $g_i(\bar{x}_i(k))$ depend on states $x_1(k), \ldots, x_i(k)$, i.e., $\bar{x}_i(k)$. However, for nonstrict feedback nonlinear systems, $f_i(\bar{x}_i(k))$ and $g_i(\bar{x}_i(k))$ depend on both $\bar{x}_i(k)$ and $x_{i+1}(k)$, and there are no currently available control design schemes. Even if the nonstrict feedback nonlinear discrete-time system is transformed into an equivalent form, the nonlinearities are still dependent upon all the states. Available [1]–[3] methods applied to nonlinear discrete-time systems in nonstrict feedback form will result in a noncausal controller (current control input depends on future system states), even if the system is of second order and when the adaptive NN backstepping approach is utilized. Finally, no optimization is carried out in control designs for strict feedback discrete-time systems, whereas a simple tracking error is utilized as a performance measure.

Available NN controller designs employ online NN training based on classical adaptive control [3], where a short-term system performance measure is defined by using the tracking error. By contrast, the reinforcement-learning-based adaptive-critic NN approach [4] has emerged as a promising tool to develop optimal NN controllers due to its potential to find approximate solutions to dynamic programming, where a *strategic* utility function, which is considered as the long-term system performance measure, can be optimized. In supervised learning, an explicit signal is provided by the teacher to guide the learning process, whereas in the case of reinforcement learning, the role of the teacher is more evaluative than instructional in nature. The critic NN monitors the system states and approximates the *strategic* utility function, with a potential for a look-ahead and better training of the action NN that generates the control action to the system.

There are many variants of adaptive-critic NN controller architectures [4]–[9] using state feedback, even though few results [6]–[9], [17] address convergence. These controllers are limited to affine nonlinear discrete-time systems. Such

Manuscript received February 7, 2007; revised October 24, 2007, February 18, 2008, and March 22, 2008. First published March 24, 2009; current version published September 16, 2009. This work was supported in part by the National Science Foundation (NSF) under Grant ECCS 0327877 and Grant ECCS 0621924, NSF I/UCRC Award for IMS, GAANN Program, and the Intelligent Systems Center. This paper was recommended by Associate Editor F. L. Lewis.

P. Shih and S. Jagannathan are with the Department of Electrical and Computer Engineering, Missouri University of Science and Technology, Rolla, MO 65409 USA (e-mail: sarangap@mst.edu).

B. C. Kaul and J. A. Drallmeier are with the Department of Mechanical and Aerospace Engineering, Missouri University of Science and Technology, Rolla, MO 65409 USA.

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TSMCB.2009.2013272

adaptive-critic NN controller results are not available for nonlinear discrete-time systems in nonstrict feedback form. It is important to note that a nonstrict feedback system by no means can be represented as a strict feedback system through analytical manipulation.

In this paper, a novel adaptive-critic NN-based output-feedback controller is developed to control a class of nonlinear discrete-time systems of second order in nonstrict feedback form with bounded and unknown disturbances. The backstepping methodology [1], [2] is utilized for controller design with two action NNs being used to generate the virtual and actual control inputs, respectively. The critic NN approximates certain *strategic* utility function that is a variant of the standard Bellman equation. The two action NN weights are tuned by the critic NN signal to minimize the *strategic* utility function and their outputs (or control inputs). The NN observer generates the estimates of the system states and output, which are subsequently used in controller design. The proposed controller is *model free*, since the dynamics of nonlinear discrete-time systems are unknown, and NN weights are tuned online. Reinforcement learning is accomplished online unlike existing adaptive-critic schemes, where an iterative approach is normally utilized. Controller extensions to an nth-order nonstrict feedback nonlinear discrete-time system are also briefly discussed.

The main contributions of this paper can be summarized as follows: 1) The adaptive NN backstepping scheme is extended to nonstrict feedback nonlinear discrete-time systems using the adaptive-critic approach; optimization of a long-term performance index is undertaken in contrast with traditional adaptive NN backstepping schemes [1], [2] where no optimization is performed; 2) demonstration of the boundedness of the overall system is shown even in the presence of NN approximation errors and bounded unknown disturbances, unlike in existing adaptive-critic works [7]–[9] where convergence is presented under ideal circumstances and in an iterative manner; 3) stability proof is inferred, even with an NN observer, by relaxing the separation principle via novel weight update rules and by selecting the Lyapunov function consisting of system estimation errors, tracking, and NN weight estimation errors for the adaptive-critic approach. Such mathematically proven stable output-feedback control approaches using adaptive critics are not presented in literature.

The proposed controller is evaluated on a spark ignition (SI) engine model operating lean, which is a practical nonstrict feedback nonlinear discrete-time system of second order. The controller allows the engine to operate in the lean regime, where small stoichiometric ratio of fuel to air is injected in each cycle. Operating an engine lean will result in cyclic dispersion in heat release that makes the engine performance to degrade, ultimately becoming unstable. The controller enables the engine to operate leaner compared with the uncontrolled case by reducing the heat-release dispersion while minimizing the fuel intake, which is the control input. Consequently, fuel conversion efficiency is improved, and engine-out emissions decrease due to lean operation. Although an SI engine with a three-way catalyst cannot be operated lean, the objective is to control an SI engine that is used for other applications such as scooters and lawn mowers, where a three-way catalyst is not normally used. Alternatively, the proposed scheme could be used with the new generation of lean NO_x catalyst systems that are currently under development.

In our previous work [18], an adaptive backstepping approach is utilized to control an SI engine operating with high exhaust gas recirculation levels. This system is represented as a complex discrete-time system consisting of a combination of nonstrict feedback and affine nonlinear discrete-time systems, which is significantly different from the proposed work. On the other hand, in [16], an online tracking controller is introduced for a class of nonstrict feedback nonlinear discrete-time systems with the potential application to an engine operating lean. However, no optimization is carried out, since reinforcement learning is not employed. The results from this proposed work are compared with that of our previous works. Finally, the proposed control scheme can easily be extended to nth-order discrete-time systems.

II. Controller Design

A. Nonlinear Nonstrict Feedback Discrete-Time Systems

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

$$\begin{aligned} x_1(k+1) &= f_1\left(x_1(k), x_2(k)\right) \\ &\quad + g_1\left(x_1(k), x_2(k)\right) x_2(k) + d_1(k) \qquad (3) \\ x_2(k+1) &= f_2\left(x_1(k), x_2(k)\right) \\ &\quad + g_2\left(x_1(k), x_2(k)\right) u(k) + d_2(k) \qquad (4) \\ y(k) &= f_3\left(x_1(k), x_2(k)\right) \qquad (5) \end{aligned}$$

where $x_i(k) \in \Re$, $i = 1, 2$, are the states, $u(k) \in \Re$ is the system input, and $d_1(k) \in \Re$ and $d_2(k) \in \Re$ are the unknown but bounded disturbances whose bounds are given by $|d_1(k)| < d_{1m}$ and $|d_2(k)| < d_{2m}$, with d_{1m} and d_{2m} being the known positive scalars. Here, the nonlinearities are considered unknown. The system output is an unknown nonlinear function of the states in contrast with that in the available literature [11], [12] where the output is considered as a linear function of the states. Finally, only the output is considered measurable, whereas the states are not available for measurement. An additional constraint that has to be satisfied is that the states have to be close to their respective target values for the proposed application of SI engine control. Then, the fuel-to-air ratio, which is defined as the ratio of the second state to the first one has to be close to its target value. Since the nonstrict feedback nonlinear discrete-time system cannot be expressed in the form of strict feedback via analytical manipulation, a new control design is introduced by using an observer.

To overcome the immeasurable states, namely, $x_1(k)$ and $x_2(k)$, an observer is utilized, where the current heat-release output $y(k)$ is employed to estimate the future output $\hat{y}(k+1)$ and states $\hat{x}_1(k+1)$ and $\hat{x}_2(k+1)$. The design of the observer will be discussed next.

B. Observer Design

Consider the system output equation (4). Since the output function is considered unknown, we use a two-layer feedforward NN with semirecurrent architecture and novel weight tuning to construct the output as

$$y(k+1) = w_1^{\mathrm{T}} \phi\left(v_1^{\mathrm{T}} z_1(k)\right) + \varepsilon\left(z_1(k)\right) \tag{6}$$

where $z_1(k) = [x_1(k), x_2(k), y(k), u(k)]^{\mathrm{T}} \in R^4$ is the network input; $y(k+1)$ and $y(k)$ are the future and current output values; $w_1 \in \Re^{n_1}$ and $v_1 \in \Re^{2\times n_1}$ denote the ideal output and constant hidden-layer weights, respectively; $u(k)$ is the control input; $\phi(v_1^{\mathrm{T}} z_1(k))$ represents the hidden-layer activation function; n_1 is the number of hidden-layer nodes; and $\varepsilon(z_1(k)) \in \Re$ is the approximation error. For convenience, the two equations can be represented as

$$\phi_1(k) = \phi\left(v_1^{\mathrm{T}} z_1(k)\right) \tag{7}$$

$$\varepsilon_1(k) = \varepsilon\left(z_1(k)\right). \tag{8}$$

Rewrite (6) using (7) and (8) to obtain

$$y(k+1) = w_1^{\mathrm{T}} \phi_1(k) + \varepsilon_1(k). \tag{9}$$

The states $x_1(k)$ and $x_2(k)$ are not measurable; therefore, $z_1(k)$ is not available either. Using the estimated states and measured output $\hat{x}_1(k)$, $\hat{x}_2(k)$, and $y(k)$, respectively, instead of $x_1(k)$, $x_2(k)$, and $y(k)$, the proposed observer is given as

$$\begin{aligned}\hat{y}(k+1) &= \hat{w}_1^{\mathrm{T}}(k) \phi\left(v_1^{\mathrm{T}} \hat{z}_1(k)\right) + l_1 \tilde{y}(k)\\ &= \hat{w}_1^{\mathrm{T}}(k) \hat{\phi}_1(k) + l_1 \tilde{y}(k)\end{aligned} \tag{10}$$

where $\hat{z}_1(k) = [\hat{x}_1(k), \hat{x}_2(k), y(k), u(k)]^{\mathrm{T}} \in R^4$ is the NN input vector using the estimated states, $\hat{y}(k+1)$ and $\hat{y}(k)$ are the estimated future and current outputs, $\hat{w}_1(k)$ is the actual weight matrix, $\hat{\phi}_1(k)$ is the hidden-layer activation function, $l_1 \in R$ is the observer gain, and $\tilde{y}(k)$ is the output estimation error defined as

$$\tilde{y}(k) = \hat{y}(k) - y(k). \tag{11}$$

It is demonstrated in [13] that if the hidden-layer weights v_1's are chosen initially at random and are kept constant, and the number of hidden-layer nodes is sufficiently large, then the approximation error $\varepsilon(z_1(k))$ can be made arbitrarily small so that the bound $\|\varepsilon(z_1(k))\| \leq \varepsilon_{1m}$ holds for all $z_1(k) \in S$ in a compact set, since the activation function vector forms a basis to the nonlinear function that the NN approximates. Now, we choose, at our convenience, the observer structure as a function of output estimation errors and known quantities as

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

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

where $l_2 \in R$ and $l_3 \in R$ are the design constants, and f_{10}, f_{20}, and g_{20} are the known nominal values for the unknown nonlinear functions. These nominal values can be obtained by a variety of ways, including Taylor series expansion and without ignoring higher order terms lumped as uncertain higher-order nonlinear terms. The expansion of the nonlinear functions is not required, and the higher-order terms are not ignored, except that their nominal values have to be known. The reason for requiring limited information from this unknown system is the uncertain system dynamics and their output relationship that is considered nonlinear and unknown. As a consequence, it is not possible to design an observer if everything is considered unknown. At least, some information in the form of nominal values has to be given to design an observer that is consistent with all available control literature [14]. The experimental section indeed presents how these nominal values can be obtained for a practical system.

C. Observer Error Dynamics

Define the state estimation and output errors as

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

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

Combine (3)–(6) and (12)–(15) to obtain the estimation and output error as

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

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

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

Now select the weight tuning of the observer NN as

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

where $\alpha_1 \in R$ and $l_4 \in R$ are the design constants.

Remark: The observer structure has direct implication on the stability, since the observer dynamics have to be considered in the Lyapunov proof. Normally, when the observer error dynamics are derived, the control input is eliminated, whereas, due to the expansion of the unknown dynamics into a known and an unknown part, the elimination of the control input is not possible here, making the stability proof difficult.

However, it will be shown that, by using the aforementioned weight tuning, the separation principle is relaxed, and the closed-loop signals will be bounded in Theorem 2. Next, we present the following theorem, where it is demonstrated that the state and output estimation errors and the observer NN weight estimation errors are bounded, provided that the control input is bounded, and in Theorem 2, it is relaxed. The following mild assumptions are required.

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

Definition 1 [10], [14]: The state vector of the closed-loop system is said to be uniformly ultimately bounded (UUB) if there exists a compact set $S \subset \Re^n$ so that, for all $x_0 \in S$, there exists a bound $\mu \geq 0$ and a number $N(\mu, x_0)$ such that $\|x(k)\| \leq \mu$ for all $k \geq k_0 + N$.

Theorem 1 (Observer Stability): Consider the system given by (3)–(5), and let the disturbance bounds d_{1m} and d_{2m} be known constants. Let the observer NN weight tuning be given by (19). The state estimation errors $\tilde{x}_1(k)$ and $\tilde{x}_2(k)$, output estimation error $\tilde{y}(k)$, and observer NN weight estimation error $\tilde{w}_1(k)$ are UUB, provided that the input is bounded, with the bounds specifically given by (35)–(38), provided that the observer design parameters are selected as

1) $$0 < \alpha_1 \left\|\phi_1(k)\right\|^2 < 1 \tag{20}$$

2) $$|l_1| < \frac{1}{\sqrt{3}} \tag{21}$$

3) $$|l_2| < \frac{\sqrt{3}}{3} \tag{22}$$

4) $$|l_3| < \frac{\sqrt{3}}{3} \tag{23}$$

5) $$|l_4| < \frac{1}{\sqrt{6}} \tag{24}$$

where α_1 is the NN adaptation gain, and l_1, l_2, l_3, and l_4 are the observer parameters.

Proof: Define the Lyapunov function

$$\begin{aligned} J(k) &= \sum_{i=1}^{4} J_i(k) \\ &= \frac{\gamma_1}{\alpha_1}\tilde{w}_1^{\mathrm{T}}(k)\tilde{w}_1(k) + \frac{\gamma_2}{3}\tilde{x}_1^2(k) + \frac{\gamma_3}{2}\tilde{x}_2^2(k) + \frac{\gamma_4}{3}\tilde{y}^2 \end{aligned} \tag{25}$$

where $0 < \gamma_i$, $i \in \{1, 2, 3, 4\}$, are the auxiliary constants. Take the first difference of the first term, and substitute (19) to get

$$\begin{aligned} J_1(k) &= \frac{\gamma_1}{\alpha_1}\tilde{w}_1^{\mathrm{T}}(k)\tilde{w}_1(k) \\ \frac{\alpha_1}{\gamma_1}\Delta J_1(k) &= \tilde{w}_1^{\mathrm{T}}(k+1)\tilde{w}_1(k+1) - \tilde{w}_1^{\mathrm{T}}(k)\tilde{w}_1(k) \\ &= \left[\tilde{w}_1^{\mathrm{T}}(k) - \alpha_1\Big(\hat{w}_1^{\mathrm{T}}(k)\hat{\phi}_1(k) + l_4\tilde{y}(k)\Big)^{\mathrm{T}} \hat{\phi}_1^{\mathrm{T}}(k)\right] \\ &\quad * \left[\tilde{w}_1(k) - \alpha_1\hat{\phi}_1(k)\Big(\hat{w}_1^{\mathrm{T}}(k)\hat{\phi}_1(k) + l_4\tilde{y}(k)\Big)\right] \\ &\quad - \tilde{w}_1^{\mathrm{T}}(k)\tilde{w}_1(k) \\ &= \alpha_1^2 \left\|\phi_1(k)\right\|^2 \Big(\hat{w}_1^{\mathrm{T}}(k)\hat{\phi}_1(k) + l_4\tilde{y}(k)\Big)^2 \\ &\quad - 2\alpha_1\tilde{w}_1(k)\hat{\phi}_1(k)\Big(\hat{w}_1^{\mathrm{T}}(k)\hat{\phi}_1(k) + l_4\tilde{y}(k)\Big) \\ &\quad + \alpha_1\Big(\hat{w}_1^{\mathrm{T}}(k)\hat{\phi}_1(k) + l_4\tilde{y}(k)\Big)^2 \\ &\quad - \alpha_1\Big(\hat{w}_1^{\mathrm{T}}(k)\hat{\phi}_1(k) + l_4\tilde{y}(k)\Big)^2 \end{aligned}$$

$$\begin{aligned} &= -\alpha_1\Big(1 - \alpha_1\left\|\hat{\phi}_1(k)\right\|^2\Big) \\ &\quad * \Big(\hat{w}_1^{\mathrm{T}}(k)\hat{\phi}_1(k) + l_4\tilde{y}(k)\Big)^2 \\ &\quad + \alpha_1\Big(\Big(\zeta_1(k) + w_1^{\mathrm{T}}\hat{\phi}_1 + l_4\tilde{y}(k)\Big) - \zeta_1(k)\Big)^2 \\ &\quad - \alpha_1\zeta_1^2(k) \\ &= -\alpha_1\Big(1 - \alpha_1\left\|\hat{\phi}_1(k)\right\|^2\Big) \\ &\quad * \Big(\hat{w}_1^{\mathrm{T}}(k)\hat{\phi}_1(k) + l_4\tilde{y}(k)\Big)^2 \\ &\quad + \alpha_1\Big(w_1^{\mathrm{T}}\hat{\phi}_1(k) + l_4\tilde{y}(k)\Big)^2 - \alpha_1\zeta_1^2(k) \end{aligned} \tag{26}$$

where $\zeta_1(k) = \tilde{w}_1^{\mathrm{T}}(k)\hat{\phi}_1(k) = \hat{w}_1^{\mathrm{T}}(k)\hat{\phi}_1(k) - w_1^{\mathrm{T}}\hat{\phi}_1(k)$.

Invoke the Cauchy–Schwarz inequality defined as

$$(a_1b_1 + \cdots + a_nb_n)^2 \leq \left(a_1^2 + \cdots + a_n^2\right)\left(b_1^2 + \cdots + b_n^2\right) \tag{27}$$

and simplifying the first difference in (26) to get

$$\begin{aligned} \Delta J_1(k) \leq& -\gamma_1\left(1 - \alpha_1\left\|\hat{\phi}_1(k)\right\|^2\right)\Big(\hat{w}_1(k)\hat{\phi}_1(k) + l_4\tilde{y}(k)\Big)^2 \\ &+ 2\gamma_1(w_{1m}\hat{\phi}_{1m})^2 + 2\gamma_1 l_4^2\tilde{y}^2(k) - \gamma_1\zeta_1^2(k) \end{aligned} \tag{28}$$

where the ideal weights and activation functions are bounded by $\|w_1\| \leq w_{1m}$ and $\|\tilde{\phi}_1\| \leq \tilde{\phi}_{1m}$, respectively.

Take the second term and substitute (16) to derive

$$\begin{aligned} \Delta J_2(k) \leq& \gamma_2 l_2^2\tilde{y}^2(k) + \gamma_2\tilde{x}_2^2(k) \\ &+ \gamma_2(w_{1m}\phi_{1m} + f_{10} + \varepsilon_{1m} + d_{1m})^2 - \frac{\gamma_2}{3}\tilde{x}_1^2(k). \end{aligned} \tag{29}$$

Take the third term in (25), substitute (17), and assume that the input is bounded such that $u_{\max} > |u(k)|$ to get

$$\begin{aligned} \Delta J_3(k) \leq& \gamma_3\left(f_{20} + (g_{20} + g_{2\max})u_{\max} + f_{2\max} + d_{2m}\right)^2 \\ &+ \gamma_3 l_3^2\tilde{y}^2(k) - \frac{\gamma_3}{2}\tilde{x}_2^2(k). \end{aligned} \tag{30}$$

Take the fourth and final term and substitute (18) to obtain

$$\Delta J_4(k) \leq \gamma_4\zeta_1^2(k) + \gamma_4 l_1^2\tilde{y}(k) + \gamma_4(w_{1m}\tilde{\phi}_{1m} + \varepsilon_{1m}) - \frac{\gamma_4}{3}\tilde{y}^2(k). \tag{31}$$

Combine (28)–(31) and simplify the first difference to get the first difference of the Lyapunov function as

$$\begin{aligned} \Delta J(k) \leq& -\gamma_1\left(1 - \alpha_1\left\|\hat{\phi}_1(k)\right\|^2\right)\Big(\hat{w}_1(k)\hat{\phi}_1(k) + l_4\tilde{y}(k)\Big)^2 \\ &- \left(\frac{\gamma_3}{2} - \gamma_2\right)\tilde{x}_2^2(k) - \frac{\gamma_2}{3}\tilde{x}_1^2(k) \\ &- \left(\frac{\gamma_4}{3} - 2\gamma_1 l_4^2 - \gamma_2 l_2^2 - \gamma_3 l_3^2 - \gamma_4 l_1^2\right)\tilde{y}^2(k) \\ &- (\gamma_1 - \gamma_4)\zeta_1^2(k) + D_M^2 \end{aligned} \tag{32}$$

where D_M^2 is defined as

$$\begin{aligned} D_M^2 =& 2\gamma_1(w_{1m}\hat{\phi}_{1m})^2 + \gamma_2(w_{1m}\phi_{1m} + f_{10} + \varepsilon_{1m} + d_{1m})^2 \\ &+ \gamma_3\left(f_{20} + (g_{20} + g_{2\max})u_{\max} + f_{2\max} + d_{2m}\right)^2 \\ &+ \gamma_4(w_{1m}\tilde{\phi}_{1m} + \varepsilon_{1m}). \end{aligned} \tag{33}$$

Select

$$\gamma_3 > 2\gamma_2 \quad \gamma_4 > 6\gamma_1 l_4^2 + 3\gamma_2 l_2^2 + 3\gamma_3 l_3^2 + 3\gamma_4 l_1^2 \qquad \gamma_1 > \gamma_4. \tag{34}$$

This implies that $\Delta J(k) < 0$ as long as (20)–(24) and the following hold:

$$|\,\tilde{x}_1(k)| > \frac{D_M}{\sqrt{\frac{\gamma_2}{3}}} \tag{35}$$

or

$$|\,\tilde{x}_2(k)| > \frac{D_M}{\sqrt{\frac{\gamma_3}{2} - \gamma_2}} \tag{36}$$

or

$$|\,\tilde{y}(k)| > \frac{D_M}{\sqrt{\frac{\gamma_4}{3} - 2\gamma_1 l_4^2 - \gamma_2 l_2^2 - \gamma_3 l_3^2 - \gamma_4 l_1^2}} \tag{37}$$

or

$$|\,\zeta_1(k)| > \frac{D_M}{\sqrt{\gamma_1 - \gamma_4}}. \tag{38}$$

According to a standard Lyapunov extension theorem [14], this demonstrates that the estimation errors, the output error, and the NN observer weight estimation errors are UUB. ■

Remark 1: In the aforementioned theorem, the state and output estimation errors and the NN weights of the observer are shown to be bounded, provided that the input is bounded. The separation principle needs to be asserted for controller design if the system under consideration is linear. Unfortunately, the separation principle does not hold for nonlinear systems. Therefore, in the next section, the boundedness of the closed-loop system is demonstrated, where the observer and all the controller signals, including the control input, are proven to be bounded. Here, the assumption that the control input is bounded is relaxed. Note that the boundedness of the control input in the aforesaid theorem for proving the observer stability may not be a so stringent assumption, since, for identification purposes alone, inputs are considered to be bounded in order to show the boundedness of the identification error [14]. In a sense, the observer is like an identifier expecting the inputs to be bounded in order to reconstruct the states. The need for the boundedness of the control input for proving the stability of the observer is the direct consequence of the observer structure, as mentioned before. In any case, this assumption will be relaxed next.

Remark 2: Equations (20)–(24), along with (35)–(38), are used to ensure that the bounds are positive. In other words, to guarantee that a selection of $\gamma_1, \ldots, \gamma_4$ satisfying (35)–(38) exists, the observer gain selections can be obtained as

$$\gamma_4 > 6\gamma_1 l_4^2 + 3\gamma_2 l_2^2 + 3\gamma_3 l_3^2 + 3\gamma_4 l_1^2 > 3\gamma_4 l_1^2 \Rightarrow |l_1| < \frac{1}{\sqrt{3}}.$$

Similarly, using

$$\begin{cases} \gamma_4 > 6\gamma_1 l_4^2 + 3\gamma_2 l_2^2 + 3\gamma_3 l_3^2 + 3\gamma_4 l_1^2 \\ \gamma_1 > \gamma_4 \end{cases}$$
$$\Rightarrow \gamma_4 > 6\gamma_1 l_4^2 > 6\gamma_4 l_4^2 \Rightarrow |l_4| < \frac{1}{\sqrt{6}}. \tag{39}$$

Additionally, by choosing $|l_2| < \sqrt{3}/3$ and $|l_3| < \sqrt{3}/3$, $\gamma_1, \ldots, \gamma_4$ can be assured to be positive.

D. Strategic Utility Function for Critic NN Design

The purpose of the critic NN is to approximate the long-term performance index (or strategic utility function) of the nonlinear system through online weight adaptation. The critic signal also tunes the two action NNs. The tuning will ultimately minimize the strategic utility function and NN outputs (control inputs) so that closed-loop stability is inferred.

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

$$p(k) = \begin{cases} 0, & \text{if } (|\tilde{y}(k)|) \le c \\ 1, & \text{otherwise} \end{cases} \tag{40}$$

where $c \in \Re$ is a user-defined threshold. The utility function $p(k)$ represents the current performance index. In other words, $p(k) = 0$ and $p(k) = 1$ refer to the good and unsatisfactory tracking performances at the kth time step, respectively. The threshold value "c" should be selected by keeping in mind the speed of convergence and tracking-error bounds. An additional remark is added after the theorem later in this section.

The long-term *strategic* utility function $Q(k) \in \Re$ is defined as

$$Q(k) = \beta^N p(k+1) + \beta^{N-1} p(k+2) + \cdots + \beta^{k+1} p(N) + \cdots \tag{41}$$

where $0 < \beta < 1$ is the discount factor, and N is the horizon index. The term $Q(k)$ is viewed here as the long-term system performance measure for the controller, since it is the sum of all future system performance indices. Minimization of the long-term measure (41) is accomplished by minimizing (41) with respect to the control input. This is done by selecting a quadratic performance index consisting of the critic NN signal for tuning the action NN weights via minimization of the quadratic index. After some manipulation, (41) can also be expressed as $Q(k) = \min_{u(k)}\{\beta Q(k-1) - \beta^{N+1} p(k)\}$, which is similar to the standard Bellman equation.

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

$$\hat{Q}(k) = \hat{w}_2^{\mathrm{T}}(k)\phi\left(v_2^{\mathrm{T}}\hat{z}_2(k)\right) = \hat{w}_2^{\mathrm{T}}(k)\hat{\phi}_2(k) \tag{42}$$

where $\hat{Q}(k) \in \Re$ is the critic signal, $\hat{w}_2(k) \in \Re^{n_2}$ is the tunable weight matrix, $v_2 \in \Re^{2\times n_2}$ represents the constant input weight matrix selected initially at random [13], $\hat{\phi}_2(k) \in \Re^{n_2}$ is the activation function vector in the hidden layer, n_2 is the number of hidden-layer nodes, and $\hat{z}_2(k) = [\hat{x}_1(k), \hat{x}_2(k)]^{\mathrm{T}} \in R^2$ is the NN input vector.

We define the prediction error as

$$e_c(k) = \hat{Q}(k) - \beta\left(\hat{Q}(k-1) - \beta^N p(k)\right) \tag{43}$$

where the subscript "c" stands for the "critic," since $Q(k)$ is unavailable at the kth time instant. Define a quadratic objective function to minimize based on the prediction error

$$E_c(k) = \frac{1}{2}e_c^2(k). \tag{44}$$

The weight update rule for the critic NN is obtained using gradient adaptation, which is given by

$$\hat{w}_2(k+1) = \hat{w}_2(k) + \Delta\hat{w}_2(k) \tag{45}$$

$$\Delta\hat{w}_2(k) = \alpha_2\left[-\frac{\partial E_c(k)}{\partial\hat{w}_2(k)}\right] \tag{46}$$

or

$$\hat{w}_2(k+1) = \hat{w}_2(k) - \alpha_2\hat{\phi}_2(k) \times\left(\hat{Q}(k) + \beta^{N+1}p(k) - \beta\hat{Q}(k-1)\right)^{\mathrm{T}} \tag{47}$$

where $\alpha_2 \in \Re$ is the NN adaptation gain. Next, the design of the virtual and actual control input selection is introduced using the backstepping methodology.

E. Virtual Controller Design

In this section, the design of the virtual control input is discussed. Before we proceed, the following mild assumption is needed. Then, the systems of nonlinear equations are rewritten.

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

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

$$\Phi(\cdot) = f_1\left(x_1(k), x_2(k)\right) + g_1\left(x_1(k), x_2(k)\right)x_2(k) + x_2(k). \tag{48}$$

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

$$x_1(k+1) = \Phi(\cdot) - x_2(k) + d_1(k) \tag{49}$$

$$x_2(k+1) = f_2(\cdot) + g_2(\cdot)u(k) + d_2(k). \tag{50}$$

Our goal is to stabilize the system output $y(k)$ around a specified target point y_d by controlling the input. The secondary objective is to make $x_1(k)$ approach its target $x_{1d}(k)$. At the same time, all signals in systems (3) and (4) must be UUB, all the NN weights must be bounded, and a performance index must be minimized.

Define the tracking error as

$$e_1(k) = x_1(k) - x_{1d}(k) \tag{51}$$

where $x_{1d}(k)$ is the desired trajectory. Using (49), (51) can be expressed as

$$\begin{aligned} e_1(k+1) &= x_1(k+1) - x_{1d}(k+1) \\ &= \left(\Phi(\cdot) - x_2(k) + d_1(k)\right) - x_{1d}(k+1). \end{aligned} \tag{52}$$

By viewing the second state $x_2(k)$ as a virtual control input (which is typical in backstepping design), a desired virtual control signal can be designed as

$$x_{2d}(k) = \Phi(\cdot) - x_{1d}(k+1) + l_5\hat{e}_1(k) \tag{53}$$

where l_5 is a gain constant. Since $\Phi(\cdot)$ is an unknown function, $x_{2d}(k)$ in (53) cannot be implemented in practice. We invoke the NN universal approximation property to estimate this unknown nonlinear function

$$\Phi(\cdot) = w_3^{\mathrm{T}}\phi\left(v_3^{\mathrm{T}}z_3(k)\right) + \varepsilon\left(z_3(k)\right) \tag{54}$$

where $z_3(k) = [x_1(k), x_2(k)]^{\mathrm{T}} \in \Re^2$ is the input vector, $w_3^{\mathrm{T}} \in \Re^{n_2}$ and $v_3^{\mathrm{T}} \in \Re^{2\times n_3}$ are the ideal and constant input weight matrices, respectively, $\phi(v_3^{\mathrm{T}}z_3(k)) \in \Re^{n_3}$ is the activation function vector in the hidden layer, n_3 is the number of hidden-layer nodes, and $\varepsilon(z_3(k))$ is the functional estimation error. Similar to the case of observer and critic NN design, using the results from [13], the hidden-layer weights v_3's are chosen initially at random and are kept constant, and the number of hidden-layer nodes is chosen to be sufficiently large in order to make the approximation error $\varepsilon(z_3(k))$ arbitrarily small so that the bound $\|\varepsilon(z_3(k))\| \le \varepsilon_{3m}$ holds for all $z_3(k) \in S$ in a compact set where ε_{3m} is the upper bound.

Rewriting (53) using (54), the virtual control signal can be rewritten as

$$x_{2d}(k) = w_3^{\mathrm{T}}\phi\left(v_3^{\mathrm{T}}z_3(k)\right) + \varepsilon\left(z_3(k)\right) - x_{1d}(k+1) + l_5\hat{e}_1(k). \tag{55}$$

Replacing the actual states with the estimated ones, (55) becomes

$$\begin{aligned} \hat{x}_{2d}(k) &= \hat{w}_3^{\mathrm{T}}(k)\phi\left(v_3^{\mathrm{T}}\hat{z}_3(k)\right) - x_{1d}(k+1) + l_5\hat{e}_1(k) \\ &= \hat{w}_3^{\mathrm{T}}(k)\hat{\phi}_3(k) - x_{1d}(k+1) + l_5\hat{e}_1(k) \end{aligned} \tag{56}$$

where $\hat{z}_3(k) = [\hat{x}_1(k), \hat{x}_2(k)]^{\mathrm{T}} \in \Re^2$ is the NN input vector using the estimated states, and $\hat{e}_1(k) = \hat{x}_1(k) - x_{1d}(k)$. Define

$$e_2(k) = x_2(k) - \hat{x}_{2d}(k). \tag{57}$$

Equation (52) can be rewritten using (57)

$$\begin{aligned} e_1(k+1) &= \left(\Phi(\cdot) - x_2(k) + d_1(k)\right) - x_{1d}(k+1) \\ &= \Phi(\cdot) - \left(e_2(k) + \hat{x}_{2d}(k)\right) + d_1(k) - x_{1d}(k+1) \\ &= \Phi(\cdot) - \hat{x}_{2d}(k) - e_2(k) - x_{1d}(k+1) + d_1(k). \end{aligned} \tag{58}$$

Replace (56) into (58), and then, substitute (54) into the combined equation (58)

$$\begin{aligned} e_1(k+1) =& \Phi(\cdot) - \left(\hat{w}_3^{\mathrm{T}}(k)\hat{\phi}_3(k) - x_{1d}(k+1) + l_5\hat{e}_1(k)\right) \\ &- e_2(k) - x_{1d}(k+1) + d_1(k) \\ =& \left(w_3^{\mathrm{T}}\phi_3(k) + \varepsilon_3(k)\right) - \hat{w}_3^{\mathrm{T}}(k)\hat{\phi}_3(k) \\ &- l_5\hat{e}_1(k) - e_2(k) + d_1(k) \\ =& w_3^{\mathrm{T}}\left(\hat{\phi}_3(k) - \tilde{\phi}_3(k)\right) - \hat{w}_3^{\mathrm{T}}(k)\hat{\phi}_3(k) + \varepsilon_3(k) \\ &- l_5\hat{e}_1(k) - e_2(k) + d_1(k) \\ =& w_3^{\mathrm{T}}\left(\hat{\phi}_3(k) - \tilde{\phi}_3(k)\right) - \hat{w}_3^{\mathrm{T}}(k)\hat{\phi}_3(k) + \varepsilon_3(k) \\ &- l_5\hat{e}_1(k) - e_2(k) + d_1(k) \\ =& -\tilde{w}_3^{\mathrm{T}}\hat{\phi}_3(k) - w_3^{\mathrm{T}}\tilde{\phi}_3(k) + \varepsilon_3(k) \\ &- l_5\hat{e}_1(k) - e_2(k) + d_1(k) \\ =& -\zeta_3(k) - w_3^{\mathrm{T}}\tilde{\phi}_3(k) + \varepsilon_3(k) \\ &- l_5\hat{e}_1(k) - e_2(k) + d_1(k) \end{aligned} \tag{59}$$

where

$$\zeta_3(k) = \tilde{w}_3^{\mathrm{T}}(k)\hat{\phi}_3(k) = \hat{w}_3^{\mathrm{T}}(k)\hat{\phi}_3(k) - w_3^{\mathrm{T}}\hat{\phi}_3(k) \tag{60}$$

$$\tilde{\phi}_3(k) = \phi\left(v_3\hat{z}_3(k)\right) - \phi\left(v_3 z_3(k)\right). \tag{61}$$

Let us define

$$e_{a1}(k) = \hat{w}_3^{\mathrm{T}}(k)\hat{\phi}_3(k) + \left(\hat{Q}(k) - Q_d(k)\right) \tag{62}$$

where $\hat{Q}(k)$ is defined in (42), $Q_d(k)$ represents the desired strategic utility, and the subscript "$a1$" represents the error for the first action NN, i.e., $e_{a1}(k) \in \Re$. The desired *strategic* utility function $Q_d(k)$ is selected as "0" [4] to indicate perfect tracking at all steps, whereas the first term in (60) is essentially the action NN output or virtual control input. Thus, (62) becomes

$$e_{a1}(k) = \hat{w}_3^{\mathrm{T}}(k)\hat{\phi}_3(k) + \hat{Q}(k). \tag{63}$$

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

$$E_{a1}(k) = \frac{1}{2}e_{a1}^2(k). \tag{64}$$

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

$$\hat{w}_3(k+1) = \hat{w}_3(k) + \Delta\hat{w}_3(k) \tag{65}$$

where

$$\Delta\hat{w}_3(k) = \alpha_3\left[-\frac{\partial E_{a1}(k)}{\partial\hat{w}_3(k)}\right] \tag{66}$$

or, in other words

$$\hat{w}_3(k+1) = \hat{w}_3(k) - \alpha_3\hat{\phi}_3(k)\left(\hat{Q}(k) + \hat{w}_3^{\mathrm{T}}(k)\hat{\phi}_3(k)\right) \tag{67}$$

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

F. Actual Controller Design

Choose the following desired control input:

$$u_d(k) = \frac{1}{g_2(k)}\left(-f_2(k) + \hat{x}_{2d}(k+1) + l_6 e_2(k)\right). \tag{68}$$

Note that $u_d(k)$ is noncausal, since it depends upon the future value of $\hat{x}_{2d}(k+1)$. We solve this problem by using a semirecurrent NN, since it can be a one-step predictor. The term $\hat{x}_{2d}(k+1)$ depends on state $x(k)$, virtual control input $\hat{x}_{2d}(k)$, desired trajectory $x_{1d}(k+2)$, and system errors $e_1(k)$ and $e_2(k)$. By taking the independent variables as the input to an NN, $\hat{x}_{2d}(k+1)$ can be approximated during control input selection. Consequently, in this paper, a feedforward NN with a properly chosen weight tuning law rendering a semirecurrent or dynamic NN can be used to predict the future value. Alternatively, the value can be obtained by employing a filter [14]. The first layer of the second NN generates $\hat{x}_{2d}(k+1)$ using the system errors, state estimates, and past value $\hat{x}_{2d}(k)$ as inputs. The output of the first layer is used by the second layer to generate a suitable control input. The results in the simulation section show that the overall controller performance is satisfactory. On the other hand, one can use a single-layer dynamic NN to generate the future value of $\hat{x}_{2d}(k)$, which 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.

If the NN input is assumed to be $z_4(k) = [x_1(k), x_2(k), e_1(k), e_2(k), \hat{x}_{2d}(k), x_{1d}(k+2)]^{\mathrm{T}} \in \Re^6$, then $u_d(k)$ can be approximated as

$$u_d(k) = w_4^{\mathrm{T}}\phi\left(v_4^{\mathrm{T}}z_4(k)\right) + \varepsilon\left(z_4(k)\right) = w_4^{\mathrm{T}}\phi_4(k) + \varepsilon_4(k) \tag{69}$$

where $w_4 \in \Re^{n_4}$ and $v_4 \in \Re^{6\times n_4}$ denote the constant ideal output and hidden-layer weight matrices, respectively, $\phi_4(k) \in \Re^{n_4}$ is the activation function vector, n_4 is the number of hidden-layer nodes, and $\varepsilon(z_4(k))$ is the estimation error so that the bound $\|\varepsilon(z_4(k))\| \leq \varepsilon_{4m}$ holds for all $z_4(k) \in S$ in a compact set. Again, we hold the input weights constant and adapt the output weights only. We also replace the actual states with estimated ones to design the control input as

$$u(k) = \hat{w}_4^{\mathrm{T}}(k)\phi\left(v_4^{\mathrm{T}}\hat{z}_4(k)\right) = \hat{w}_4^{\mathrm{T}}(k)\hat{\phi}_4(k) \tag{70}$$

where $\hat{z}_4(k) = [\hat{x}_1(k), \hat{x}_2(k), \hat{e}_1(k), \hat{e}_2(k), \hat{x}_{2d}(k), x_{1d}(k+2)]^{\mathrm{T}} \in \Re^6$ is the input vector. Rewrite (57) and substitute (68)–(70) to get

$$\begin{aligned} &e_2(k+1) \\ &\quad = x_2(k+1) - \hat{x}_{2d}(k+1) \\ &\quad = \left(f_2(\cdot) + g_2(\cdot)\hat{w}_4^{\mathrm{T}}(k)\hat{\phi}_4(k) + d_2(k)\right) - \hat{x}_{2d}(k+1) \end{aligned}$$

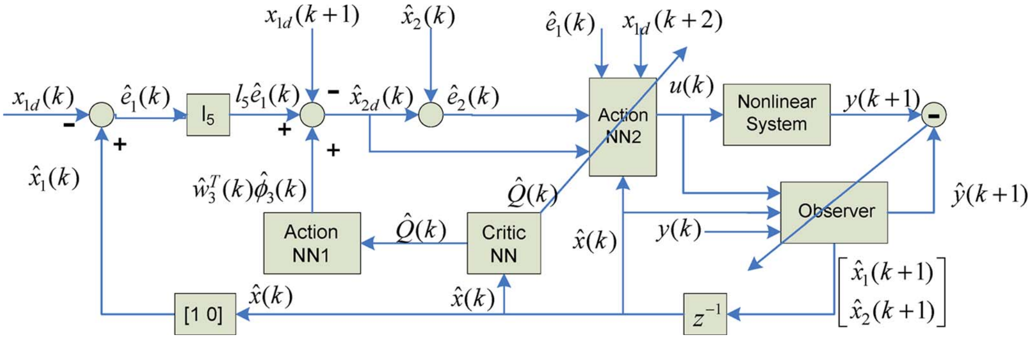

Fig. 1. Adaptive-critic NN-based controller schematic.

$$\begin{aligned}
&= f_2(\cdot) + g_2(\cdot)\left(\tilde{w}_4^{\mathrm{T}}(k)\hat{\phi}_4(k) + w_4^{\mathrm{T}}\phi_4(k) + w_4^{\mathrm{T}}\tilde{\phi}_4(k)\right)\\
&\quad + d_2(k) - \hat{x}_{2d}(k+1)\\
&= f_2(\cdot) + g_2(\cdot)\left(w_4^{\mathrm{T}}(k)\phi_4(k)\right)\\
&\quad + g_2(\cdot)\left(\zeta_4(k) + w_4^{\mathrm{T}}\tilde{\phi}_4(k)\right) + d_2(k) - \hat{x}_{2d}(k+1)\\
&= f_2(\cdot) + g_2(\cdot)\left(u_d(k) - \varepsilon_4(k)\right)\\
&\quad + g_2(\cdot)\left(\zeta_4(k) + w_4^{\mathrm{T}}\tilde{\phi}_4(k)\right) + d_2(k) - \hat{x}_{2d}(k+1)\\
&= l_6 e_2(k) - g_2(\cdot)\varepsilon_4(k) + g_2(\cdot)\zeta_4(k)\\
&\quad + g_2(\cdot)w_4^{\mathrm{T}}\tilde{\phi}_4(k) + d_2(k) \qquad (71)
\end{aligned}$$

where

$$\zeta_4(k) = \tilde{w}_4^{\mathrm{T}}(k)\hat{\phi}_4(k) = \hat{w}_4^{\mathrm{T}}(k)\hat{\phi}_4(k) - w_4^{\mathrm{T}}(k)\hat{\phi}_4(k) \qquad (72)$$

$$\tilde{\phi}_4(k) = \hat{\phi}_4(k) - \phi_4(k). \qquad (73)$$

Equations (59) and (71) represent the closed-loop error dynamics. Next, we derive the weight update law for the second action NN. Define

$$e_{a2}(k) = \hat{w}_4^{\mathrm{T}}(k)\hat{\phi}_4(k) + \hat{Q}(k) \qquad (74)$$

where $e_{a2}(k) \in \Re$, and the subscript "$a2$" stands for the second action NN. The first term in (72) is the NN output or control input to the nonlinear system. Here, the desired *strategic* utility function $Q_d(k)$ is "0" to indicate perfect tracking at all steps. Following a similar design, choose a quadratic objective function to minimize

$$E_{a2}(k) = \frac{1}{2}e_{a2}^2(k). \qquad (75)$$

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

$$\hat{w}_4(k+1) = \hat{w}_4(k) + \Delta\hat{w}_4(k) \qquad (76)$$

with

$$\Delta\hat{w}_4(k) = \alpha_4\left[-\frac{\partial E_{a2}(k)}{\partial \hat{w}_4(k)}\right] \qquad (77)$$

or

$$\hat{w}_4(k+1) = \hat{w}_4(k) - \alpha_4\hat{\phi}_4(k)\left(\hat{w}_4^{\mathrm{T}}(k)\hat{\phi}_4(k) + \hat{Q}(k)\right). \qquad (78)$$

The proposed controller structure is shown in Fig. 1. Next, in the following theorem, it is demonstrated that the closed-loop system is UUB under some mild assumption that will be stated next.

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

$$\|w_1\| \le w_{1m} \quad \|w_2\| \le w_{2m} \quad \|w_3\| \le w_{3m} \quad \|w_4\| \le w_{4m} \qquad (79)$$

where $w_{1m} \in R^+$, $w_{2m} \in R^+$, $w_{3m} \in R^+$, and $w_{4m} \in R^+$ represent the bounds on the unknown target weights, where the Frobenius norm [14] is used.

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

$$\begin{aligned}
\left\|\tilde{\phi}_1(\cdot)\right\| &\le \tilde{\phi}_{1m} \quad \left\|\tilde{\phi}_2(\cdot)\right\| \le \tilde{\phi}_{2m}\\
\left\|\tilde{\phi}_3(\cdot)\right\| &\le \tilde{\phi}_{3m} \quad \left\|\tilde{\phi}_4(\cdot)\right\| \le \tilde{\phi}_{4m} \qquad (80)
\end{aligned}$$

where $\hat{\phi}_{1m}, \tilde{\phi}_{1m} \in R^+$, $\hat{\phi}_{2m}, \tilde{\phi}_{2m} \in R^+$, $\hat{\phi}_{3m}, \tilde{\phi}_{3m} \in R^+$, and $\hat{\phi}_{4m}, \tilde{\phi}_{4m} \in R^+$ are the upper bounds.

Theorem 2 (Closed-Loop–Observer–Controller Stability): Consider the system given by (3) and (4), and let the disturbance bounds d_{1m} and d_{2m} be the known constants. Let the observer, critic, virtual control, and control input NN weight tuning be given by (19), (47), (67), and (78), respectively. Let the virtual control and actual control inputs be given by (56) and (70), the tracking errors $e_1(k)$ and $e_2(k)$ and weight estimates $\hat{w}_1(k)$, $\hat{w}_2(k)$, $\hat{w}_3(k)$, and $\hat{w}_4(k)$ be UUB, with the bounds specifically given by (B.17) (shown below), and the design parameters be selected as

$$0 < \alpha_1 \|\phi_1(k)\|^2 < 1 \qquad (81)$$

$$0 < \alpha_2 \|\phi_2(k)\|^2 < 1 \qquad (82)$$

$$0 < \alpha_3 \|\phi_3(k)\|^2 < 1 \qquad (83)$$

$$0 < \alpha_4 \|\phi_4(k)\|^2 < 1 \qquad (84)$$

$$|l_1| < \frac{1}{\sqrt{3}} \tag{85}$$

$$|l_2| < \frac{\sqrt{3}}{3} \tag{86}$$

$$|l_3| < \frac{\sqrt{3}}{3} \tag{87}$$

$$|l_4| < \frac{1}{\sqrt{6}} \tag{88}$$

$$|l_5| < \frac{1}{\sqrt{5}} \tag{89}$$

$$|l_6| < \frac{\sqrt{3}}{3} \tag{90}$$

$$0 < \beta < \frac{\sqrt{2}}{2} \tag{91}$$

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

Proof: See Appendix B. ■

Remark 3: The proposed adaptive-critic NN controller scheme can be implemented in real-time manner rather than in an offline iterative manner which is normally observed in most adaptive-critic NN methods [4]. Suitable utility functions are selected for both the critic and action NNs so that their weights are tuned by minimizing the quadratic performance indices. The observer NN estimates the states, whereas the critic NN, using the binary signal of the system tracking errors as the long-term utility function, generates a signal that is used to tune the two action NNs. The first action NN is utilized here to generate the virtual control input, whereas and the second one generates the actual control input. In addition, the two NN weights are tuned by using the gradient-descent-based rule while minimizing the control inputs and long-term utility function. Normally, a single critic is used to tune an action NN for all adaptive-critic controller schemes in the literature [14], whereas in this paper, a single critic is employed to tune two action NNs due to the class of nonlinear discrete-time systems under consideration. If the nonlinear discrete-time system is an affine one, then a single action NN is sufficient [17].

Remark 4: 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 in the aforementioned theorem, since the Lyapunov function is a quadratic function of the system and weight estimation errors of the observer and controller NNs. Consequently, the need for the boundedness of the control input for the previous theorem is relaxed in this theorem.

Remark 5: It is important to note that, in this theorem, persistency of excitation (PE) condition for the NN observer and the NN controller and linearity in parameter assumption are not needed, since the first difference does not require the PE condition 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 is tuned, the NN method relaxes the linearity in the unknown parameter assumption. Additionally, the certainty equivalence principle is not used. Overall, the approach renders a well-defined control.

Remark 6: The NN weight tuning in (23), (45), (65), and (76) results in a semirecurrent NN, even though a feedforward NN architecture is utilized. Here, 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. This semirecurrent NN architecture renders a dynamic NN that is capable of predicting the state one step ahead.

Remark 7: It is only possible to show the boundedness of all the closed-loop signals by using an extension of Lyapunov stability [14] due to the presence of approximation errors and bounded disturbances, which is consistent with the literature.

Remark 8: In order to guarantee that a selection of $\gamma_1, \ldots, \gamma_{10}$ satisfying (B.16) (shown below) exists, the observer and controller gains have to be selected as

$$\gamma_{10} > 6\gamma_3 l_4^2 + 3\gamma_8 l_2^2 + 3\gamma_9 l_3^2 + 3\gamma_{10} l_1^2$$
$$\Rightarrow \gamma_{10} > 3\gamma_{10} l_1^2 \Rightarrow |l_1| < \frac{1}{\sqrt{3}}.$$

Now, using

$$\begin{cases} \gamma_{10} > 6\gamma_3 l_4^2 + 3\gamma_8 l_2^2 + 3\gamma_9 l_3^2 + 3\gamma_{10} l_1^2 \\ \gamma_3 > \gamma_{10} \end{cases}$$
$$\Rightarrow \gamma_{10} > 6\gamma_3 l_4^2 > 6\gamma_{10} l_4^2 \Rightarrow |l_4| < \frac{1}{\sqrt{6}}.$$

Similarly, using $\gamma_1 > 5\gamma_1 l_5^2 \Rightarrow |l_5| < (1/\sqrt{5})$.

Also, $\gamma_2 > 3\gamma_1 + 3\gamma_2 l_6^2 \Rightarrow \gamma_2 > 3\gamma_2 l_6^2 \Rightarrow |l_6| < \sqrt{3}/3$.

Similar to Theorem 1, by selecting $|l_2| < (\sqrt{3}/3)$ and $|l_3| < (\sqrt{3}/3)$, the bounds are guaranteed to be positive. Finally, the selection of the observer gains from this theorem appears to be consistent with that of Theorem 1.

Remark 9: Although the proposed scheme is shown for a second-order nonstrict feedback nonlinear discrete-time system, the proposed backstepping scheme can be extendable to an nth-order system with modifications.

Remark 10: The need for an exact model of the nonlinear discrete-time system in many existing reinforcement learning or adaptive-critic approaches [5], [6] is relaxed in this paper. The action NN will learn the unknown system dynamics through the feedback signals from the closed-loop system. The proposed actor–critic architecture will render a model-free approach [5], [6].

Remark 11: There is no explicit offline training phase, and the updating of the NNs is performed in an online manner. This is in contrast with many reinforcement learning designs where some *a priori* training is needed. Additionally, the proposed methodology does not require the stop/reset strategy utilized by certain adaptive-critic schemes [5], [6].

Remark 12: Equations (47)–(56) relate the selection of adaptation, observer, and controller gains, whereas (57) provides how the discount factor can be chosen in order to ensure stability and convergence. Such a relationship does not exist in the existing adaptive-critic literature where the discount factor and adaptation gains are selected by a trial-and-error procedure [4].

Remark 11: With the proposed approach, learning can be performed simultaneously in both the critic and action NNs, which is in contrast with some of the available schemes where learning is first accomplished by the critic NN and then by the action NN [4].

Remark 12: The selection of the user-defined threshold "c" has implications in the stability of the closed-loop system, since it affects the critic NN performance. This value of "c" is normally selected to be small; otherwise, the performance is considered unacceptable, and the strategic utility function is then constructed. Unless this value of "c" is chosen to be small, the action NN weights are not close to their near-optimal weights. These affect the tracking error and NN weight estimation error bounds. However, there is a tradeoff between speed of convergence and value of "c." A small value of "c" takes longer to converge and vice versa. More effort is needed to understand the effect of "c" on tracking and NN weight estimation error bounds, which will be a part of future work.

Corollary 1: The proposed adaptive-critic NN controller and the weight update rule with parameter selection based on (81)–(91) cause the state $x_2(k)$ to approach the desired virtual control input $x_{2d}(k)$.

Proof: Combining (55) and (56), the difference between $\hat{x}_{2d}(k)$ and $x_{2d}(k)$ is given by

$$\hat{x}_{2d}(k) - x_{2d}(k) = \tilde{w}_3(k)\phi_3(k) - \varepsilon\left(z_3(k)\right) = \zeta_3(k) - \varepsilon_3(k) \tag{92}$$

where $\tilde{w}_3(k) \in \Re^{n_3}$ is the first action NN weight estimation error, and $\zeta_3(k) \in \Re$ is defined in (60). Since both $\zeta_3(k) \in \Re$ and $\varepsilon_3(k)$ are bounded, $\hat{x}_{2d}(k)$ is bounded close to $x_{2d}(k)$. In Theorem 1, we show that $e_2(k)$ is bounded, i.e., the state $x_2(k)$ is bounded to the virtual control signal $\hat{x}_{2d}(k)$. Thus, the state $x_2(k)$ is bounded to the desired virtual control signal $x_{2d}(k)$. ■

III. Results and Analysis

Lean operation of an SI engine allows low emissions and improved fuel efficiency. However, lean operation destabilizes the engine due to the cyclic dispersion in heat release that causes misfires. The adaptive-critic NN controller is designed to stabilize the SI engine operating at lean conditions. In our previous works, an adaptive NN controller approach [18] was used to control the engine operating lean, with fuel being the control input. However, the control input changes by more than 2.5%, causing a shift in the equivalence ratio (ratio of total fuel to air) or operating regime of the engine that is undesirable. This calls for a new controller, such as the one proposed in this paper.

A. Daw Engine Model

SI engine dynamics can be expressed, according to the Daw model, as a class of nonlinear systems in nonstrict feedback form [15]

$$\begin{aligned} x_1(k+1) =& AF(k) + F(k)x_1(k) \\ &- R \cdot F(k)CE(k)x_2(k) + d_1(k) \end{aligned} \tag{93}$$

$$\begin{aligned} x_2(k+1) =& \left(1 - CE(k)\right) F(k)x_2(k) \\ &+ \left(MF(k) + u(k)\right) + d_2(k) \end{aligned} \tag{94}$$

$$y(k) = x_2(k)CE(k) \tag{95}$$

$$\varphi(k) = R\frac{x_2(k)}{x_1(k)} \tag{96}$$

$$CE(k) = \frac{CE_{\max}}{1 + 100^{-(\varphi(k)-\varphi_m)/(\varphi_u-\varphi_l)}} \tag{97}$$

$$\varphi_m = \frac{\varphi_u - \varphi_l}{2} \tag{98}$$

where $x_1(k)$ and $x_2(k)$ are the total masses of air and fuel in each cylinder that is unknown, and $AF(k)$ and $MF(k)$ represent the mass flow rates of new air and nominal fuel, respectively. The term $\varphi(k)$ is defined as the fuel-to-air ratio or equivalence ratio, and the $y_1(k)$ variable is the heat release at the kth instance. The term φ_m relates the upper φ_u and lower φ_l values of certain system parameters, as given by (98), and it is used to compute combustion efficiency (97). The combustion efficiency $CE(k)$ is a function of both states, and it is within the range of $0 < CE_{\min} < CE(k) < CE_{\max}$ that is typically unknown. Moreover, the unknown residual gas fraction $F(k)$ is bounded by $0 < F_{\min} < F(k) < F_{\max}$. The unknown residual gas fraction is defined as the ratio of total fuel to air remaining in the engine cylinder after combustion (the fuel and air that were not burned during combustion). The term $F(k)$ is a function of both $x_1(k)$ and $x_2(k)$. The sum of the nominal fuel and the control input will be considered as the total fuel input per cycle. Finally, R is the air-to-fuel ratio constant at stoichiometric conditions.

The terms $d_1(k)$ and $d_2(k)$ are unknown disturbances yet upper bounded by $|d_1(k)| < d_{1m}$ and $|d_2(k)| < d_{2m}$, with d_{1m} and d_{2m} being the positive scalars. The engine dynamics (93)–(97) can be represented in the general form given by (3)–(5), with the nonlinearities being defined in (99).

To implement the observer, replace the following from the Daw model into the general form as

$$\begin{aligned} f_1(\cdot) =& AF(k) + F(k)x_1(k) \\ g_1(\cdot) =& - R \cdot F(k)CE(k) \\ f_2(\cdot) =& \left(1 - CE(k)\right) F(k)x_2(k) + MF(k) \\ g_2(\cdot) =& 1 \end{aligned} \tag{99}$$

$$\begin{aligned} f_{10} =& AF_0 + F_0\hat{x}_1(k) \\ g_{10} =& - R \cdot F_0CE_0 \\ f_{10} =& \left(1 - CE_0\right) F_0\hat{x}_2(k) + MF_0 \\ g_{10} =& 1. \end{aligned} \tag{100}$$

Equations (93)–(97), which represent the engine operating lean, are a nonlinear discrete-time system in nonstrict feedback form, since $F(k)$ and $CE(k)$ are a function of $x_1(k)$ and $x_2(k)$. To implement the controller, replace (99) in place of $f_1(\cdot)$ and $g_1(\cdot)$, and combine them similar to (48) to define

$$\Phi(\cdot) = AF(k) + F(k)x_1(k) - R \cdot F(k)CE(k)x_2(k) + x_2(k). \tag{101}$$

To calculate the nominal values for (3) and (4), we run the engine at the desired equivalence ratio. That will give us the nominal fuel, air, and equivalence ratio—MF_0, AF_0, and φ_0. From those, combustion efficiency CE_0 is calculated.

B. Simulation Results

The controller is easily simulated in C in conjunction with the Daw model. The learning rates for the observer (81), critic (82), virtual control input (83), and control input (84) networks are 0.01, 0.01, 0.01, and 0.01, respectively. The gains l_1, l_2, l_3, l_4, l_5, and l_6 are selected as 0.05, 0.05, 0.04, 0.05, 0.2, and 0.1. The constant "c" is chosen as 0.001 for the simulation and experimental work. The system constants $CE_{\max}$, φ_l, and φ_u are chosen as 1, 0.66, and 0.73 based on the physics of the engine system. The critic constants β and N are 0.4 and 4 based on the conditions from Theorem 1. All NNs use 20 neurons with hyperbolic tangent sigmoid activation functions in the hidden layer.

The maximum moles that a single cylinder can hold is set as 0.021 to match the experimental engine constraint shown in the next section. Using this constant along with the following equations:

$$\varphi = R\left(\frac{MF}{AF}\right) \tag{102}$$

$$tm = \frac{MF}{mw_{\text{fuel}}} + \frac{AF}{mw_{\text{air}}} \tag{103}$$

where mw_{fuel} and mw_{air} are the molecular weights of fuel and air, respectively, and tm is the maximum moles that each cylinder is capable of holding, for each equivalence ratio set point, φ, MF, and AF can be calculated.

The last two system variables, namely, disturbances and stochastic effects, are modeled as follows. First, we assume that a Gaussian distribution governs the two effects. We may inject disturbances to the two states in (93) and (94) due to $d_1(k)$ and $d_2(k)$, but a simpler method is to perturb the equivalence ratio (96). This simplification is sufficient because the states are not measurable; therefore, the disturbances are increasingly complex and immeasurable. Stochastic effects alter the output, and through the combustion efficiency equation (97) and, finally, the output equation (95), this single perturbation effectively models the last two system variables. The final model uses a Gaussian distribution noise that is injected into (96) centered around the target equivalence ratio and deviation of 1% of the target equivalence ratio. The resulting simulation output matches the output observed from the Ricardo engine. All simulations ran for 5000 cycles uncontrolled first and then 5000 cycles controlled.

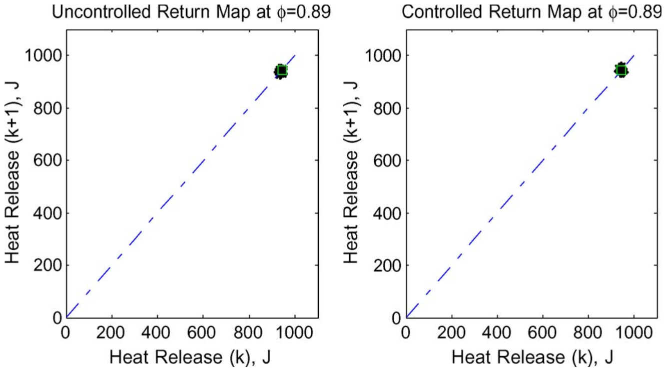

Fig. 2. Uncontrolled and controlled heat-release return maps at $\varphi = 0.89$. Heat release at $k + 1$th instance is plotted against heat release at kth instance.

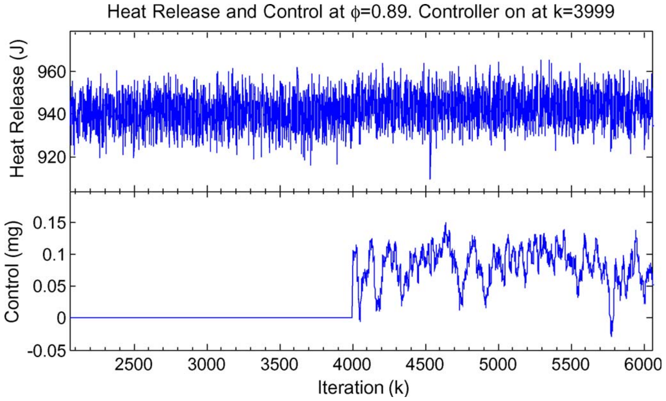

Fig. 3. Heat-release and control input at $\varphi = 0.89$. The controller turns on at $k = 4000$. Note the almost instant learning convergence of the controller.

Fig. 2 shows two heat-release return maps, i.e., one controlled and the other uncontrolled, for an equivalence ratio of 0.89. Each subfigure shows heat release for the next time step versus the current time step. Points centered along the 45° line represent heat-release values that are equal to the next-step heat release. Note the clustering of the points around the mean heat release of 870 J. The square represents the target heat release. The relatively high equivalence ratio exhibits little dispersion, which is indicated by little or no stray points away from the central cluster. The left uncontrolled plot is similar to the right controlled plot, because the controller is quiescent due to the simulated engine that is performing well. There are no complete misfires, but the heat-release variation can be clearly seen. Fig. 3 shows the time series of the heat release and control input at the same equivalence ratio. The controller activates after several thousand cycles, which is indicated by the fluctuation of the control output. The controller converges quickly and to a stable operation point. The presence of spikes in the control output indicates a decline in heat release such as a misfire, translating into additional fuel control to counteract.

Figs. 4 and 5 show another set point at 0.79. Similar features appear compared with the previous equivalence ratio, except with higher frequency and amplitude of dispersion. Improvements shown reflect the assertion of the control action.

In order to quantify the performance of the controller, we compare the coefficient of variation (COV), which is the normalized standard deviation divided by the mean of the heat

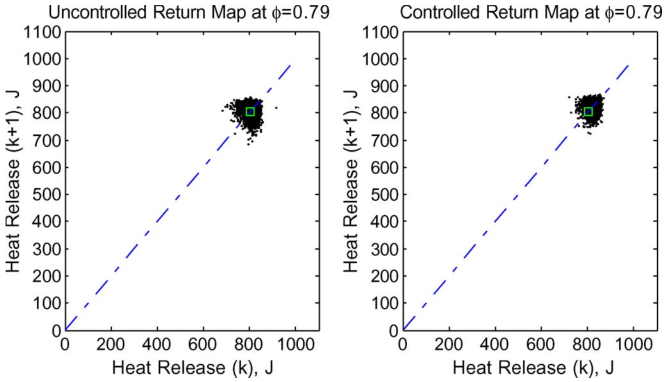

Fig. 4. Uncontrolled and controlled heat-release return maps at $\varphi = 0.79$.

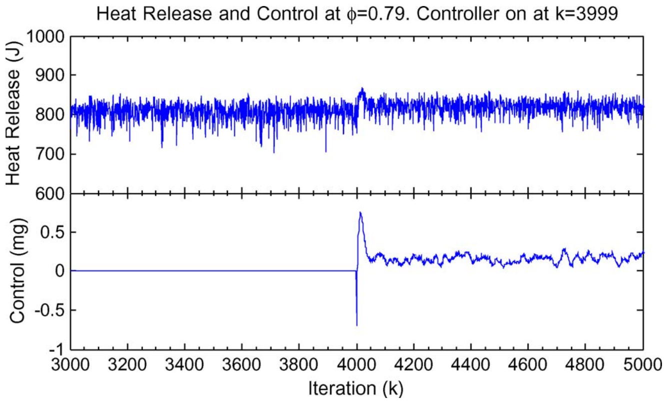

Fig. 5. Heat release and control input at $\varphi = 0.79$.

TABLE I
COV AND FUEL DATA FOR EACH OF THE SIX SET POINTS

	Covariance (COV)		%COV	%Fuel
φ	**Uncontrolled**	**Controlled**	**Change**	**Change**
0.89	0.0080	0.0077	-4.1	0.29
0.84	0.0090	0.0087	-3.3	0.11
0.79	0.0267	0.0221	-17.0	0.66
0.77	0.0475	0.0435	-8.3	0.48
0.75	0.1217	0.1071	-12.0	0.56
0.72	0.2373	0.2128	-10.3	0.48

release. As the COV decreases, the standard deviation decreases, which indicates that the engine heat release is more stable compared with that with a higher COV. The controller performs better, and the return map should consequently approach the target value. Table I tabulates all of the data from the simulation. The COV of each set point decreased drastically (shown with a negative sign) as the controller operated. The performance exceeded the improvement expected due to the slight increase in the mean fuel input. Next, we show that the experimental data support the simulation data.

C. Ricardo Engine

The experimental results are collected from a Ricardo Hydra engine with a modern four-valve Ford Zetec head. It contains a single cylinder running at 1000 rpm with shaft encoders to signal each crank angle degree and start of cycle. There are 720° per engine cycle.

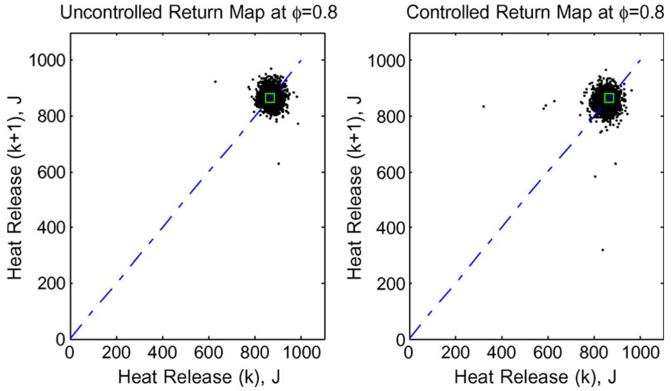

Fig. 6. Uncontrolled and controlled heat-release return maps at $\varphi = 0.8$. Heat release at $k + 1$th instance is plotted against heat release at kth instance.

In the cylinder, a piezoelectric pressure transducer records pressure every crank-angle degree. Combustion is considered to take place between 345° and 490°, for a total of 145 pressure measurements. The cylinder pressure is integrated along with volume during the 17.7-ms calculation window. All communications are completed at this time. The output of our controller is the fuel input. The fuel is controlled by a TTL signal to a fuel injector driver circuit.

All signals communicate through a custom interface board using a microcontroller. The board interfaces with the PC through a parallel port and with the engine hardware through an analog signal.

D. Experimental Results

All constants given in the simulation section are used in the experiment. The first operation for an engine run is to measure the air flow and nominal fuel. The desired equivalence ratio is given by (102), where MF is the nominal mass of fuel, AF is the nominal mass of air, and R is the constant.

These values are loaded into the controller. Ambient pressure is used to reference the in-cylinder pressure when the exhaust valve is fully open, and it is subtracted from the combustion pressure measurements. Uncontrolled and controlled data were collected at equivalence ratios of 0.8, 0.78, 0.75, and 0.72. The uncontrolled engine ran for approximately 5000 cycles, and then, the controller is turned on for another 5000 cycles. Steady state was ensured prior to data collection by measuring a stable exhaust temperature.

Fig. 6 shows two heat-release return maps, namely, one controlled and the other uncontrolled, for the equivalence ratio of 0.8. The target heat release is at 850 J. Fig. 7 shows the time series of the heat release and control input for the same equivalence ratio. Small changes indicate a quiescent controller due to the near-stoichiometric set point.

Now, denote the state and output tracking errors as

$$\begin{aligned}\hat{e}_1(k) &= \hat{x}_1(k) - x_{1d}(k)\\ \hat{e}_2(k) &= \hat{x}_2(k) - \hat{x}_{2d}(k)\\ \hat{e}_y(k) &= \hat{y}(k) - y(k)\end{aligned} \qquad (104)$$

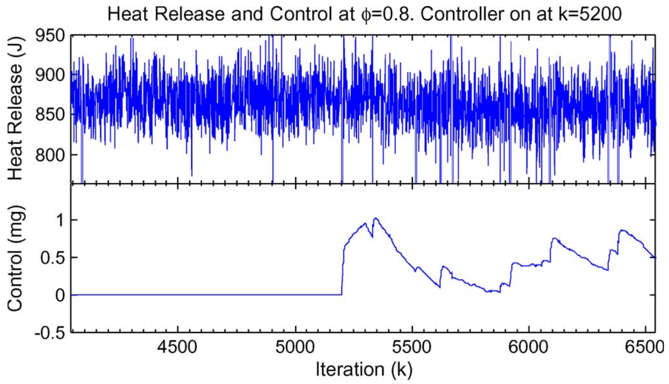

Fig. 7. Heat release and control input at $\varphi = 0.8$. The controller turns on at $k = 5200$. Note the almost instant learning convergence of the controller.

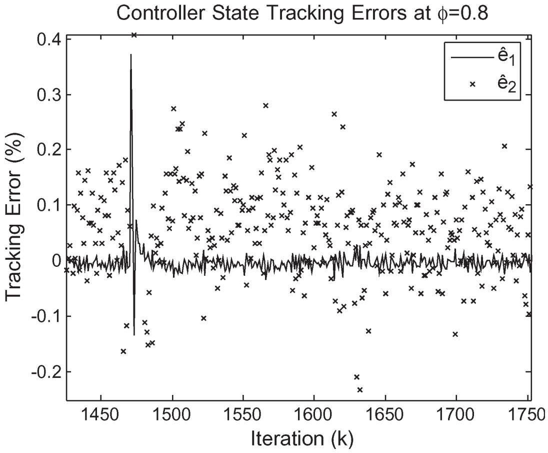

Fig. 8. State tracking errors.

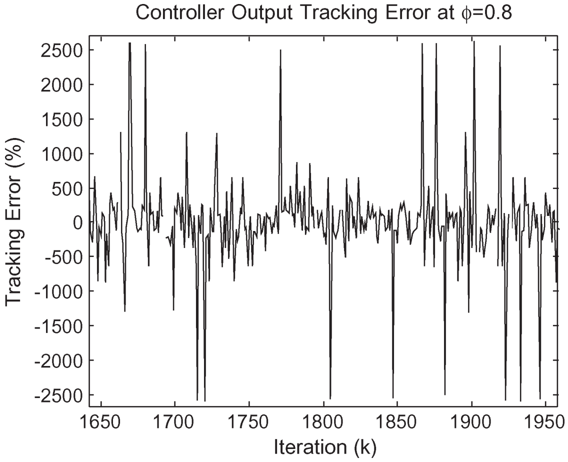

Fig. 9. Output tracking error.

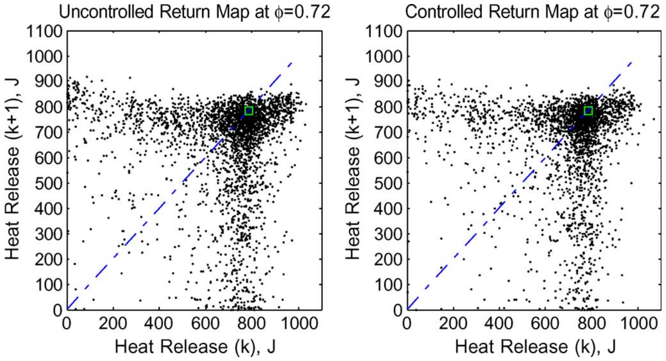

Fig. 10. Uncontrolled and controlled heat-release return maps at $\varphi = 0.72$.

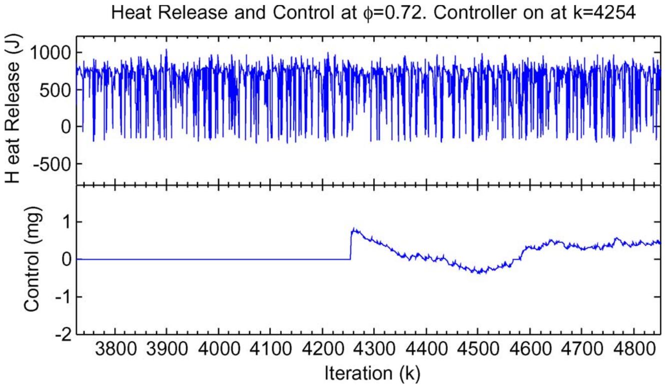

Fig. 11. Heat release and control input at $\varphi = 0.72$.

where $\hat{e}_1(k)$, $\hat{e}_2(k)$, and $\hat{e}_y(k)$ are the state-1, state-2, and output tracking errors, respectively.

Fig. 8 shows the controller state tracking errors at an equivalence ratio of 0.8. The range represents tracking error in percentage over and under the desired state trajectories. State-1 tracking error is considerably better than state-2 tracking error. The second state tracks within 0.3%; therefore, both are performing well. The spikes indicate unsuccessful tracking. Consequently, the observer and controller converged together to the desired and estimated states, generating a stable error system.

Fig. 9 shows the output tracking error in the same form as the state tracking error. An immediate observation shows an extremely high error rate. The observer performance is abysmal. Nonetheless, this signal that is fed into the NN controller allows for the critical performance factor, i.e., the state tracking errors, to converge and stabilize. It is not critical for one signal to track perfectly, but rather, it should be the system as a whole. Moreover, Theorem 1 proved the boundedness of the output estimation. In conjunction with the natural bound of the engine output, the tracking error will always be bounded. The extreme fluctuation of the observer output may be the key to the responsiveness of the controller as a whole.

Fig. 10 shows the return map of heat release for an equivalence ratio of 0.72. Note that as the equivalence ratio decreases, the return map spreads out, and dispersion increases. Fig. 11 shows the corresponding heat release and time series of the control input. Misfires increase in frequency, as shown by the negative heat-release spikes due to heat transfer from the cylinder to the environment without internal generation of useful work by combustion. Fig. 12 shows the increasing difficulty of the observer and controller to generate a low state tracking error compared with the previous case. As the engine operates in a leaner mode, the overall dispersion increases, thus degrading observer performance. Although the performance is reduced, the tracking error is well within the satisfactory performance. Fig. 13 shows the output tracking error. At the lower equivalence ratio, it is performing better than the previous equivalence ratio. This may be due to the memory effect of past

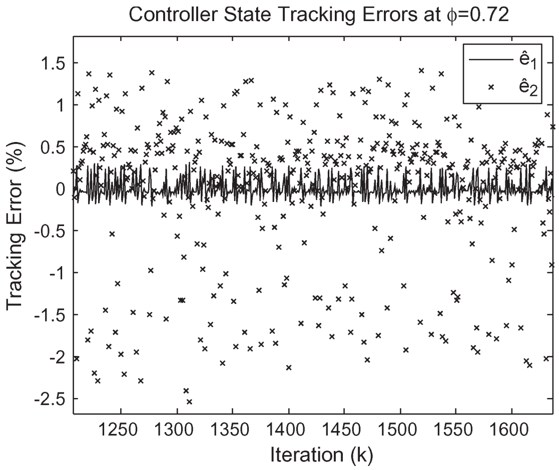

Fig. 12. State tracking error with the corresponding mean value.

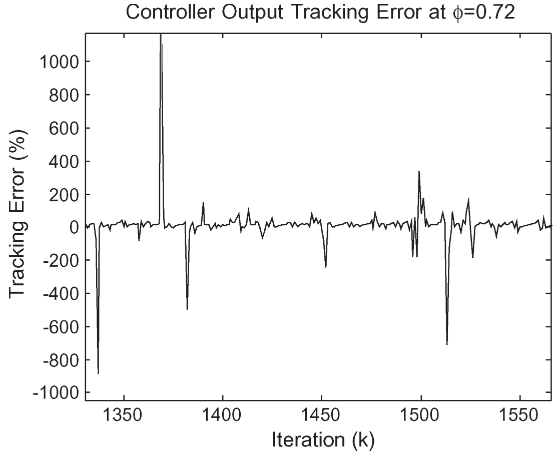

Fig. 13. Output tracking error.

engine cycles contributing to the residuals in the current cycle. At a near-stoichiometric ratio of fuel to air, little dispersion occurs, resulting in similar cylinder chemistry content before each power cycle. Stochastic effects dominate and destroy predictability. The high observer learning rate decimates the tracking ability. On the other hand, at lower equivalence ratios, higher dispersion and misfires create patterns of predictable residuals. The observer exploits the pattern recognition power of NN to drastically improve its performance.

Fig. 14 shows a detailed view of 35 controlled cycles at an equivalence ratio of 0.72. The controller generates decreasing control during cycles when the heat release is steady, which is indicated by cycles from 4947 to 4954 and from 4963 to 4769. However, during misfires or extreme dispersion in heat release, the controller attempts to compensate for the drop in heat release by pushing the control up, which is indicated by cycles 4943, 4944, 4955, etc. Note the general increase in control during sequential or near-sequential misfires, such as between cycles 4955 and 4962. The controller compensates after a one cycle delay in the positive direction and attempts to recover the engine heat release toward the target point. It is difficult to determine success on cycles with no misfire, because no heat-release plots are available for the uncontrolled case during the same cycles when the controller is operating for

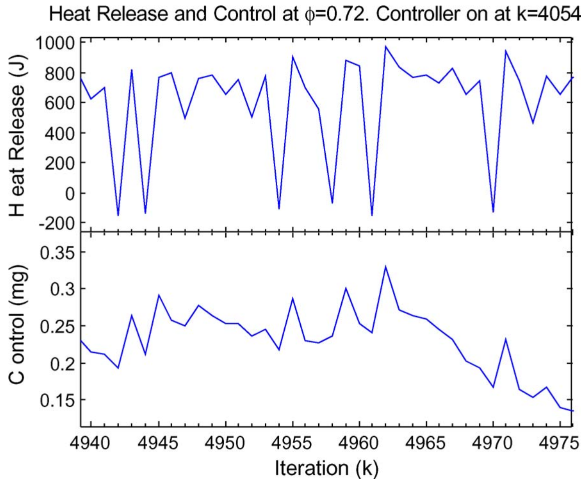

Fig. 14. Detailed view of 35 controlled cycles at $\varphi = 0.72$.

TABLE II
COV AND FUEL DATA FOR EACH OF THE FOUR SET POINTS

	COV		%COV	%Fuel
φ	**Uncontrolled**	**Controlled**	**Change**	**Change**
0.80	0.1140	0.0913	-19.9	0.76
0.78	0.1457	0.1318	-9.5	0.65
0.75	0.3438	0.2522	-26.6	2.48
0.72	0.6088	0.5712	-6.2	1.07

comparison. Overall, the controller performs according to the general expectation.

Table II shows the improved COV when the controller is in operation compared with the uncontrolled engine and also the corresponding change in nominal fuel. An improvement in the COV may be artificial due to an increase in fuel input. However, this is not the case for this controller. At all equivalence ratios except 0.75, the increase in fuel input is well within the tolerance of the equipment. On average, the COV decreased significantly by 16% compared with the controlled case, while the fuel change is minimal and less than 2.5%.

Due to reduced cyclic dispersion, fewer misfires, and low-energy cycles, a gain of approximately 8% in the indicated fuel conversion efficiency was observed for controlled-engine operation, which is significant.

IV. Conclusion

The presented controller successfully controlled an SI engine to reduce cyclic dispersion under lean operation. The system is modeled under a nonstrict feedback nonlinear discrete-time system. It converged upon a near-optimal solution through the use of a long-term strategic utility function, even though the exact dynamics are not known beforehand. It was experimentally shown that the COV was reduced when the controller was turned on. At the same time, the average fuel input did not change significantly; therefore, the improvements are solely due to the effects of the controller. The output is stable, as predicted by the Lyapunov proof.

We also provided the emission data for several set points in Appendix A. It is important to note that the emission-data

TABLE III
EMISSION DATA FOR SELECT EQUIVALENCE RATIOS

	Uncontrolled			Controlled			Change(%)		
φ	CO_2(%)	CO(%)	O_2(%)	CO_2(%)	CO(%)	O_2(%)	CO_2	CO	O_2
0.80	7.5	0.1	8.0	7.5	0.1	7.0	-0.5	0.0	-12.4
0.78	7.7	0.1	8.3	7.6	0.1	8.6	-0.5	0.0	3.9
0.75	8.9	0.1	7.7	9.1	0.1	7.5	1.8	0.0	-2.7

TABLE IV
uHC EMISSION DATA

	Uncontrolled		Controlled		Change(%)	
φ	uHC (ppm C_1)	NO_X (ppm)	uHC (ppm C_1)	NO_X (ppm)	uHC	NO_X
0.80	8732.00	1284	8919.00	761	2.1	-40.7
0.75	11154.00	215	9347.00	150	-16.2	-30.2

uncertainty may be 5% or more. Therefore, the presented data are used for indicating general trends, not as absolute improvement. However, lean operation, in general, is proven to decrease emissions compared with stoichiometric operation, regardless of the inaccuracies of emission-data collection presented. There was a significant reduction in both NO_x and unburned hydrocarbons (uHCs) between controlled and uncontrolled situations. However, the most significant drop is between lean and stoichiometric equivalence ratios. This is due to the controller's ability to successfully decrease dispersion.

APPENDIX A

Tables III and IV show the improvement in emissions for several equivalence ratios. The improvement is better than what we have seen before [16] using another controller that does not optimize any performance index. NO_x is reduced by around 30%–40% from the uncontrolled scenario. However, CO_2 remains unchanged, whereas O_2 decreased by about 4%–10%, with uHCs decreasing with control by 8% due to reduced cyclic dispersion.

APPENDIX B

Proof of Theorem 2: Define the Lyapunov function

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

where $0 < \gamma_i$, $i \in \{1, \ldots, 6\}$, are the auxiliary constants; the NN weight estimation errors $\tilde{w}_1^{\mathrm{T}}(k+1)$, $\tilde{w}_2^{\mathrm{T}}(k+1)$, $\tilde{w}_3^{\mathrm{T}}(k+1)$, and $\tilde{w}_4^{\mathrm{T}}(k+1)$ are defined in (19), (47), (67), and (78), by subtracting their respective ideal weights w_i, $i \in \{1, 2, 3, 4\}$, on both sides, respectively; the observation errors $\tilde{x}_1(k+1)$ and $\tilde{x}_2(k+1)$, are defined in (16) and (17), respectively; the system errors $e_1(k+1)$ and $e_2(k+1)$ are defined in (59) and (71), respectively; and α_i, $i \in \{1, 2, 3, 4\}$, are the NN adaptation gains. The Lyapunov function (B.1) obviates the need for the separation principle. Take the first term and the first difference using (59) to get

$$\begin{aligned} J_1(k) &= \frac{\gamma_1}{5} e_1^2(k) \\ \frac{5}{\gamma_1} \Delta J_1(k) &= e_1^2(k+1) - e_1^2(k) \\ &= \Big(-\zeta_3(k) - w_3^{\mathrm{T}} \tilde{\phi}_3(k) + \varepsilon_3(k) \\ &\qquad - l_5 \hat{e}_1(k) - e_2(k) + d_1(k) \Big)^2 - e_1^2(k) \\ &= \Big(-\zeta_3(k) - w_3^{\mathrm{T}} \tilde{\phi}_3(k) + \varepsilon_3(k) \\ &\qquad - l_5 \left(\tilde{x}_1(k) + e_1(k) \right) - e_2(k) + d_1(k) \Big)^2 \\ &\quad - e_1^2(k). \end{aligned} \tag{B.2}$$

Invoke the Cauchy–Schwarz inequality defined as

$$(a_1 b_1 + \cdots + a_n b_n)^2 \le \left(a_1^2 + \cdots + a_n^2 \right) \left(b_1^2 + \cdots + b_n^2 \right) \tag{B.3}$$

and simplify (B.2) to get

$$\begin{aligned} \frac{1}{\gamma_1} \Delta J_1(k) \le& \Big(\zeta_3^2(k) + l_5^2 \tilde{x}_1^2(k) + l_5^2 e_1^2(k) + e_2^2(k) \\ &+ \Big(\varepsilon_3(k) - w_3 \tilde{\phi}_3(k) + d_1(k) \Big)^2 \Big) - \frac{1}{5} e_1^2(k) \\ \Delta J_1(k) \le& \gamma_1 \zeta_3^2(k) + \gamma_1 l_5^2 \tilde{x}_1^2(k) + \gamma_1 l_5^2 e_1^2(k) + \gamma_1 e_2^2(k) \\ &+ \gamma_1 \Big(\varepsilon_3(k) - w_3 \tilde{\phi}_3(k) + d_1(k) \Big)^2 - \frac{\gamma_1}{5} e_1^2(k) \\ \le& \gamma_1 l_5^2 \tilde{x}_1^2(k) + \gamma_1 l_5^2 e_1^2(k) + \gamma_1 e_2^2(k) + \gamma_1 \zeta_3^2(k) \\ &+ \gamma_1 (\varepsilon_{3m} + w_{3m} \tilde{\phi}_{3m} + d_{1m})^2 - \frac{\gamma_1}{5} e_1^2(k) c. \end{aligned} \tag{B.4}$$

Take the second term, substitute (71), invoke the Cauchy–Schwarz inequality, and simplify

$$\begin{aligned} \Delta J_2(k) \le& 3 l_6^2 e_2^2(k) + 3 g_{2\max}^2 \zeta_4^2(k) \\ &+ \gamma_2 (d_{2m} + g_{2\max} \varepsilon_{4m} + g_{2\max} w_{4m} \tilde{\phi}_{4m})^2 - e_2^2(k). \end{aligned} \tag{B.5}$$

Take the third term, substitute (19), invoke the Cauchy–Schwarz inequality, and simplify

$$\Delta J_3(k) \leq -\gamma_3\left(1-\alpha_1\left\|\hat{\phi}_1(k)\right\|^2\right)\left(\hat{w}_1(k)\hat{\phi}_1(k)+l_4\tilde{y}(k)\right)^2 + 2\gamma_3(w_{1m}\hat{\phi}_{1m})^2 + 2\gamma_3 l_4^2\tilde{y}^2(k) - \gamma_3\zeta_1^2(k). \quad \text{(B.6)}$$

Take the fourth term, substitute (47), invoke the Cauchy–Schwarz inequality, and simplify

$$\begin{aligned}\Delta J_4(k) \leq & -\gamma_4\left(1-\alpha_2\left\|\hat{\phi}_2(k)\right\|^2\right) \\ & \times\left(\hat{Q}(k)+\beta^{N+1}p(k)-\beta\hat{Q}(k-1)\right)^2 \\ & -\gamma_4\zeta_2^2(k)+2\gamma_4\beta^2\zeta_2^2(k-1) \\ & +2\gamma_4\left(w_{2m}\hat{\phi}_{2m}(1+\beta)+\beta^{N+1}\right)^2. \quad \text{(B.7)}\end{aligned}$$

Take the fifth term (B.1), substitute (67), invoke the Cauchy–Schwarz inequality, and simplify

$$\Delta J_5(k) \leq -\gamma_5\left(1-\alpha_3\left\|\hat{\phi}_3(k)\right\|^2\right)\left(\hat{Q}(k)+\hat{w}_3^{\mathrm{T}}(k)\hat{\phi}_3(k)\right)^2 + 2\gamma_5\zeta_2^2(k)+2\gamma_5(w_{2m}\hat{\phi}_{2m}+w_{3m}\hat{\phi}_{3m})^2-\gamma_5\zeta_3^2(k). \quad \text{(B.8)}$$

Take the sixth term, substitute (78), invoke the Cauchy–Schwarz inequality, and simplify

$$\Delta J_6(k) = -\gamma_6\left(1-\alpha_4\left\|\hat{\phi}_4(k)\right\|^2\right)\left(\hat{w}_4^{\mathrm{T}}(k)\hat{\phi}_4(k)+\hat{Q}(k)\right)^2 + 2\gamma_6(w_{4m}\hat{\phi}_m+w_{2m}\hat{\phi}_{2m})^2+2\gamma_6\zeta_2^2(k)-\gamma_6\zeta_4^2(k). \quad \text{(B.9)}$$

Take the seventh term, set $\gamma_7 = 2\gamma_4\beta^2$, and simplify

$$\Delta J_7(k) = 2\gamma_4\beta^2\zeta_2^2(k) - 2\gamma_4\beta^2\zeta_2^2(k-1). \quad \text{(B.10)}$$

Take the eighth term, substitute (16), invoke the Cauchy–Schwarz inequality, and simplify

$$\Delta J_8(k) \leq \gamma_8 l_2^2\tilde{y}^2(k)+\gamma_8\tilde{x}_2^2(k) + \gamma_8(w_{3m}\phi_{3m}+f_{10}+\varepsilon_{3m}+d_{1m})^2-\frac{\gamma_8}{3}\tilde{x}_1^2(k). \quad \text{(B.11)}$$

Take the ninth term, substitute (17), invoke the Cauchy–Schwarz inequality, and simplify

$$\Delta J_9(k) \leq \gamma_9\left(f_{20}+(g_{20}+g_{2\max})w_{4m}\hat{\phi}_{4m}+f_{2\max}+d_{2m}\right)^2 + \gamma_9(g_{20}+g_{2\max})\zeta_4(k)+\gamma_9 l_3^2\tilde{y}^2(k)-\frac{\gamma_9}{3}\tilde{x}_2^2(k). \quad \text{(B.12)}$$

Take the tenth term, substitute (18), invoke the Cauchy–Schwarz inequality, and simplify

$$\Delta J_{10}(k) \leq \gamma_{10}\zeta_1^2(k)+\gamma_{10}l_1^2\tilde{y}(k) + \gamma_{10}(w_{1m}\tilde{\phi}_{1m}+\varepsilon_{1m})-\frac{\gamma_{10}}{3}\tilde{y}^2(k). \quad \text{(B.13)}$$

Combine (B.4)–(B.13) to get the first difference of the Lyapunov function

$$\begin{aligned}\Delta J \leq & -\left(\frac{\gamma_1}{5}-\gamma_1 l_5^2\right)e_1^2(k)-\left(\frac{\gamma_2}{3}-\gamma_1-\gamma_2 l_6^2\right)e_2^2(k) \\ & -(\gamma_3-\gamma_{10})\zeta_1^2(k)-\left(\frac{\gamma_9}{3}-\gamma_8\right)\tilde{x}_2^2(k) \\ & -(\gamma_5-\gamma_1)\zeta_3^2(k) \\ & -\left(\gamma_6-\gamma_2 g_{2\max}^2-\gamma_9(g_{20}+g_{2\max})\right)\zeta_4^2(k) \\ & -\left(\frac{\gamma_8}{3}-\gamma_1 l_5^2\right)\tilde{x}_1^2(k) \\ & -\left(\gamma_4-2\gamma_5-2\gamma_6-2\gamma_4\beta^2\right)\zeta_2^2(k) \\ & -\left(\frac{\gamma_{10}}{3}-2\gamma_3 l_4^2-\gamma_8 l_2^2-\gamma_9 l_3^2-\gamma_{10}l_1^2\right)\tilde{y}^2(k)+D_m^2 \\ & -\gamma_3\left(1-\alpha_1\left\|\hat{\phi}_1(k)\right\|^2\right)\left(\hat{w}_1(k)\hat{\phi}_1(k)+l_4\tilde{y}(k)\right)^2 \\ & -\gamma_4\left(1-\alpha_2\left\|\hat{\phi}_2(k)\right\|^2\right) \\ & \times\left(\hat{Q}(k)+\beta^{N+1}p(k)-\beta\hat{Q}(k-1)\right)^2 \\ & -\gamma_5\left(1-\alpha_3\left\|\hat{\phi}_3(k)\right\|^2\right)\left(\hat{Q}(k)+\hat{w}_3^{\mathrm{T}}(k)\hat{\phi}_3(k)\right)^2 \\ & -\gamma_6\left(1-\alpha_4\left\|\hat{\phi}_4(k)\right\|^2\right)\left(\hat{w}_4^{\mathrm{T}}(k)\hat{\phi}_4(k)+\hat{Q}(k)\right)^2 \quad \text{(B.14)}\end{aligned}$$

where

$$\begin{aligned}D_m^2 = & \gamma_1(\varepsilon_{3m}+w_{3m}\tilde{\phi}_{3m}+d_{1m})^2 \\ & +\gamma_2(d_{2m}+g_{2\max}\varepsilon_{4m}+g_{2\max}w_{4m}\tilde{\phi}_{4m})^2 \\ & +2\gamma_3(w_{1m}\hat{\phi}_{1m})^3+2\gamma_4\left(w_{2m}\hat{\phi}_{2m}(1+\beta)+\beta^{N+1}\right)^2 \\ & +2\gamma_5(w_{2m}\hat{\phi}_{2m}+w_{3m}\hat{\phi}_{3m})^2 \\ & +2\gamma_6(w_{4m}\hat{\phi}_m+w_{2m}\hat{\phi}_{2m})^2 \\ & +\gamma_8(w_{3m}\phi_{3m}+f_{10}+\varepsilon_{3m}+d_{1m})^2 \\ & +\gamma_9\left(f_{20}+(g_{20}+g_{2\max})w_{4m}\hat{\phi}_{4m}+f_{2\max}+d_{2m}\right)^2 \\ & +\gamma_{10}(w_{1m}\tilde{\phi}_{1m}+\varepsilon_{1m})^2. \quad \text{(B.15)}\end{aligned}$$

Select

$$\begin{aligned} & \gamma_1 > 5\gamma_1 l_5^2 && \gamma_2 > 3\gamma_1+3\gamma_2 l_6^2 \\ & \gamma_3 > \gamma_{10} && \gamma_4 > 2\gamma_5+2\gamma_6+2\gamma_4\beta^2 \\ & \gamma_5 > \gamma_1 && \gamma_6 > \gamma_2 g_{2\max}^2+\gamma_9(g_{20}+g_{2\max}) \\ & \gamma_7 = 2\gamma_4\beta^2 && \gamma_8 > 3\gamma_1 l_5^2 \\ & \gamma_9 > 3\gamma_8 && \gamma_{10} > 6\gamma_3 l_4^2+3\gamma_8 l_2^2+3\gamma_9 l_3^2+3\gamma_{10}l_1^2. \quad \text{(B.16)}\end{aligned}$$

This implies that $\Delta J(k) < 0$ as long as (81)–(91) and the following hold:

$$|e_1(k)| > \frac{D_m}{\sqrt{\frac{\gamma_1}{5} - \gamma_1 l_5^2}}$$

or

$$|e_2(k)| > \frac{D_m}{\sqrt{\frac{\gamma_2}{3} - \gamma_1 - \gamma_2 l_6^2}}$$

or

$$|\zeta_1(k)| > \frac{D_m}{\sqrt{\gamma_3 - \gamma_{10}}}$$

or

$$|\zeta_2(k)| > \frac{D_m}{\sqrt{\gamma_4 - 2\gamma_5 - 2\gamma_6 - 2\gamma_4 \beta^2}}$$

or

$$|\zeta_3(k)| > \frac{D_m}{\sqrt{\gamma_5 - \gamma_1}}$$

or

$$|\zeta_4(k)| > \frac{D_m}{\sqrt{\gamma_6 - \gamma_2 g_{2\max}^2 - \gamma_9 (g_{20} + g_{2\max})}}$$

or

$$|\tilde{x}_1(k)| > \frac{D_m}{\sqrt{\frac{\gamma_8}{3} - \gamma_1 l_5^2}}$$

or

$$|\tilde{x}_2(k)| > \frac{D_m}{\sqrt{\frac{\gamma_9}{3} - \gamma_8}}$$

or

$$|\tilde{y}(k)| > \frac{D_m}{\sqrt{\frac{\gamma_{10}}{3} - 2\gamma_3 l_4^2 - \gamma_8 l_2^2 - \gamma_9 l_3^2 - \gamma_{10} l_1^2}}. \qquad \text{(B.17)}$$

According to a standard Lyapunov extension theorem [10], [14], this demonstrates that the state estimation errors, the output error, and the NN observer and controller weight estimation errors are UUB. ■

REFERENCES

[1] M. Krstic, I. Kanellakopoulos, and P. Kokotovic, *Nonlinear and Adaptive Control Design*. Hoboken, NJ: Wiley, 1995.

[2] H. K. Khalil, *Nonlinear Systems*, 3rd ed. Englewood Cliffs, NJ: Prentice–Hall, 2002.

[3] F. C. Chen and H. K. Khalil, "Adaptive control of a class of nonlinear discrete-time systems using neural networks," *IEEE Trans. Autom. Control*, vol. 40, no. 5, pp. 791–801, May 1995.

[4] J. Si, A. G. Barto, W. B. Powell, and D. Wunsch, *Handbook of Learning and Approximate Dynamic Programming*. Piscataway, NJ: IEEE Press, 2004.

[5] P. J. Werbos, *Neurocontrol and Supervised Learning: An Overview and Evaluation*. New York: Van Nostrand Reinhold, 1992.

[6] J. J. Murray, C. Cox, G. G. Lendaris, and R. Saeks, "Adaptive dynamic programming," *IEEE Trans. Syst., Man, Cybern. C, Appl. Rev.*, vol. 32, no. 2, pp. 140–153, May 2002.

[7] D. P. Bertsekas and J. N. Tsitsiklis, *Neuro-Dynamic Programming*. Belmont, MA: Athena Scientific, 1996.

[8] J. Si and Y. T. Wang, "On-line learning control by association and reinforcement," *IEEE Trans. Neural Netw.*, vol. 12, no. 2, pp. 264–276, Mar. 2001.

[9] X. Lin and S. N. Balakrishnan, "Convergence analysis of adaptive critic based optimal control," in *Proc. Amer. Control Conf.*, 2000, vol. 3, pp. 1929–1933.

[10] F. L. Lewis, S. Jagannathan, and A. Yesilderek, *Neural Network Control of Robot Manipulators and Nonlinear Systems*. London, U.K.: Taylor & Francis, 1999.

[11] N. Hovakimyan, F. Nardi, A. Calise, and N. Kim, "Adaptive output feedback control of uncertain nonlinear systems using single-hidden-layer neural networks," *IEEE Trans. Neural Netw.*, vol. 13, no. 6, pp. 1420–1431, Nov. 2002.

[12] A. N. Atassi and H. K. Khalil, "A separation principle for the stabilization of a class of nonlinear systems," *IEEE Trans. Autom. Control*, vol. 44, no. 9, pp. 1672–1687, Sep. 1999.

[13] B. Igelnik and Y. H. Pao, "Stochastic choice of basis functions in adaptive function approximation and the functional-link net," *IEEE Trans. Neural Netw.*, vol. 6, no. 6, pp. 1320–1329, Nov. 1995.

[14] S. Jagannathan, *Neural Network Control of Nonlinear Discrete-Time Systems*. London, U.K.: Taylor & Francis, 2006.

[15] 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, Stat. Phys. Plasmas Fluids Relat. Interdiscip. Top.*, vol. 57, no. 3, pp. 2811–2819, Mar. 1998.

[16] J. Vance, A. Singh, B. Kaul, S. Jagannathan, and J. Drallmeier, "Development and implementation of neural network controller for spark ignition engines with high EGR levels," *IEEE Trans. Neural Netw.*, vol. 18, no. 4, pp. 1083–1100, Jul. 2007.

[17] P. He and S. Jagannathan, "Reinforcement-based neuro-output feedback control of affine discrete-time systems with input constraints," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 35, no. 1, pp. 150–154, Feb. 2005.

[18] J. Vance, P. He, S. Jagannathan, and J. Drallmeier, "Neural network-based output feedback controller for lean operation of spark ignition engine," in *Proc. Amer. Control Conf.*, Minneapolis, MN, 2006, pp. 1012–1014.

Peter Shih was born in Taiwan on August 5, 1980. He received the B.S. degree in biomedical engineering from Washington University, St. Louis, MO, in 2002 and the M.S. degree in computer engineering from Missouri University of Science and Technology (formerly University of Missouri—Rolla), Rolla, in 2007.

He is a Software Engineer in Maryland.

Brian C. Kaul was born in St. Louis, MO, on December 9, 1978. He received the B.S. (summa cum laude), M.S., and Ph.D. degrees in mechanical engineering from Missouri University of Science and Technology (formerly University of Missouri—Rolla), Rolla, in 2001, 2003, and 2008, respectively.

He is currently with the Oak Ridge National Laboratory, Knoxville, TN.

Sarangapani Jagannathan (M'95–SM'99) received the B.S. degree in electrical engineering from the College of Engineering, Anna University, Madras, India, in 1987, the M.S. degree in electrical engineering from the University of Saskatchewan, Saskatoon, SK, Canada, in 1989, and the Ph.D. degree in electrical engineering from the University of Texas, Arlington, in 1994.

From 1986 to 1987, he was a Junior Engineer with Engineers India Ltd., New Delhi, India. From 1990 to 1991, he was a Research Associate and Instructor with the University of Manitoba, Winnipeg, MB, Canada. From 1994 to 1998, he was a Consultant with the Systems and Controls Research Division, Caterpillar Inc., Peoria, IL. From 1998 to 2001, he was with the University of Texas, San Antonio. Since September 2001, he has been with Missouri University of Science and Technology (formely University of Missouri–Rolla), Rolla, where he is currently a Rutledge–Emerson Distinguished Professor with the Department of Electrical and Computer Engineering and the Site Director for the National Science Foundation Industry/University Cooperative Research Center on Intelligent Maintenance Systems. He has coauthored more than 220 refereed conference and juried journal articles, several book chapters, and three books, which are entitled *Neural Network Control of Robot Manipulators and Nonlinear Systems* (Taylor & Francis, 1999), *Discrete-Time Neural Network Control of Nonlinear Discrete-Time Systems* (CRC, 2006), and *Wireless Ad Hoc and Sensor Networks: Performance, Protocols and Control* (CRC, 2007). He is the holder of 20 patents with several pending. His research interests include adaptive and neural network control, computer/communication/sensor networks, prognostics, and autonomous systems/robotics.

Prof. Jagannathan has served on a number of IEEE conference committees. He is currently an Associate Editor for the IEEE TRANSACTIONS ON NEURAL NETWORKS, the IEEE TRANSACTIONS ON CONTROL SYSTEMS TECHNOLOGY, and the IEEE SYSTEMS JOURNAL.

James A. Drallmeier received the Ph.D. degree in mechanical engineering from the University of Illinois, Urbana, in 1989.

Since 1989, he has been a Faculty Member with Missouri University of Science and Technology (formerly University of Missouri—Rolla), Rolla, where he is currently a Professor with the Department of Mechanical and Aerospace Engineering. He operates the Spray Dynamics and Internal Combustion Engine Laboratories. His research interests include combustion, laser-based measurement systems, and internal combustion engines. His current research includes studying two-phase flows, particularly sprays and thin shear-driven films and the dynamics of highly strained dilute intermittent combustion. He has been involved in developing and using laser-based diagnostic techniques for measuring spray and thin-film dynamics over the past two decades. Additionally, he has been active in studying fuel systems and mixture preparation for advanced engine designs.