International Journal of Neural Systems, Vol. 21, No. 3 (2011) 213–224

DOI: 10.1142/S012906571100278X

World Scientific
www.worldscientific.com

NONLINEAR TORQUE AND AIR-TO-FUEL RATIO CONTROL OF SPARK IGNITION ENGINES USING NEURO-SLIDING MODE TECHNIQUES

TING HUANG
Department of Electrical and Computer Engineering
University of Illinois at Chicago, Chicago, IL 60607, USA
thuang20@uic.edu

HOSSEIN JAVAHERIAN
Propulsion Systems Research Lab, GM R&D Center
Warren, MI 48090, USA
hossein.javaherian@gm.com

DERONG LIU
Key Laboratory of Complex Systems and Intelligence Science
Institute of Automation, Chinese Academy of Sciences
Beijing 100190, China
derong.liu@ia.ac.cn

This paper presents a new approach for the calibration and control of spark ignition engines using a combination of neural networks and sliding mode control technique. Two parallel neural networks are utilized to realize a neuro-sliding mode control (NSLMC) for self-learning control of automotive engines. The equivalent control and the corrective control terms are the outputs of the neural networks. Instead of using error backpropagation algorithm, the network weights of equivalent control are updated using the Levenberg-Marquardt algorithm. Moreover, a new approach is utilized to update the gain of corrective control. Both modifications of the NSLMC are aimed at improving the transient performance and speed of convergence. Using the data from a test vehicle with a V8 engine, we built neural network models for the engine torque (TRQ) and the air-to-fuel ratio (AFR) dynamics and developed NSLMC controllers to achieve tracking control. The goal of TRQ control and AFR control is to track the commanded values under various operating conditions. From simulation studies, the feasibility and efficiency of the approach are illustrated. For both control problems, excellent tracking performance has been achieved.

Keywords: Air-to-fuel ratio control; automotive engine control; sliding mode control; torque control.

1. Introduction

In an effort to design more advanced engine control algorithms with the objective of reduced emissions and improved performance, we develop and evaluate a learning control technique based on sliding mode control methods (SLMCs). The theory of sliding mode control has been developed and widely used for more than three decades due to its robustness to system parameter uncertainties and external disturbances.[14,19,23–26,34,38,41] Essentially, SLMCs utilize a high-speed switching control law to drive state trajectory of the nonlinear system onto a specified surface in the state space, called the sliding surface, and to maintain the system state trajectory on this surface for all subsequent times. The system dynamics restricted to this surface result in highly robust control systems. By proper design of the sliding surface, SLMCs achieve the conventional goals

of control such as stabilization, tracking and regulation. In practical control applications, however, the problem of chattering and difficulty in the calculation of equivalent control limit the implementation of SLMCs. The equivalent control cannot be calculated accurately because only partial knowledge about the system is available. The chattering brought by the high frequency oscillations of the controller output results in low control accuracy and potential instability. The most common approach to reducing the chattering is using a saturation function.[44] In order to compensate for the uncertainties of the plant and reduce the computational burden, we use an estimation technique based on a neural network for the calculation of the equivalent control. Meanwhile, the continuous output of the corrective controller replaces the discontinuous sign term in conventional SLMCs for the elimination of chattering effects.

Automotive engines are known to be complex nonlinear dynamical systems. The control problems of automotive engines have been investigated by many researchers (see, e.g.,[2,27–29,33,37,43] and the references cited therein). Almost every branch of the modern and classical control theory has been researched for the control of automotive engines. The present study considers the neuro-sliding mode control (NSLMC) for both engine torque (TRQ) and air-to-fuel ratio (AFR) control of automotive engines. A good deal of work has been reported for the engine control using SLMCs, a few examples of which are cited in.[4,22,31,32,35,39,42] The application of model-based SLMCs is the emphasis in these works. SLMCs also have been applied to motion control and robotics.

Integration of a neural network into SLMCs can alleviate the problems associated with SLMCs,[21] which can be classified into three main categories. The first method is the use of different kinds of neural networks to approximate the plant nonlinearities or uncertainties and subsequently improve the control performance.[10,18,43] The second method utilizes a neural network for the adaptation of the SLMCs parameters where the SLMCs parameters are progressively updated.[20,27] The third approach is the use of neural network together with SLMCs either in parallel to act as the compensator for the conventional SLMCs controller[1,24] or to compute the equivalent and corrective control.[6,13,40] In Refs. 11 and 36), two parallel neural networks were used to realize the equivalent control and the corrective control of sliding mode control (SLMC) design. The distinction between these two works is in the error term for updating the neural network of the equivalent control. In (Ref. 11), the corrective control is handled as the error while in (Ref. 36), it is the sliding function S. However, the speed of convergence of either algorithm is slower than the one proposed in this paper where no special parameter tuning is needed.

The present work uses two parallel neural networks to realize the equivalent control and the corrective control of the SLMC design. The calculation of the equivalent control is realized by adaptively learning without *a priori* knowledge of the plant dynamics. The proposed adaptation scheme directly results in a chatter-free control action for the corrective control. The method has been successfully applied for the TRQ control[16] in our earlier phases of the engine control and calibration work. However, the AFR regulation is a more difficult control problem due to variable time delay, uncertain external disturbances, and pure dynamic effects of the process. Consequently, special considerations include the characteristics of the AFR process and control structure need to be taken. The distinct feature of the present technique is that the learning and control are carried out simultaneously, which allows the neural network controller to be further refined and improved in real-time vehicle operation through continuous learning and adaptation. For practical reasons, during the initial stage of the neural network controller learning, it is not practical to use on-line learning at the initial stage. At the initial stage, off-line engine data for initial simulation studies must be used to avoid damages to the engine if it is controlled by a randomly chosen initial controller. We will therefore first develop models of the engine for the purpose of initial controller training. In the real-time implementation, models will be replaced by the actual processes.

This paper is organized as follows: In Sec. 2, neural network models of the test engine are developed. In Sec. 3, SLMC will be briefly introduced. In Sec. 4, neuro-sliding mode controller will be developed. In Sec. 5, simulation results for engine TRQ and exhaust AFR tracking control using NSLMC will be presented. In Sec. 6, the paper will be concluded with some remarks.

2. Neural Network Modelling of TRQ and AFR Dynamics

2.1. *Engine description*

A test vehicle with a V8 engine and 4-speed automatic transmission is instrumented with engine and transmission torque sensors, wide-range AFR ratio sensors in the exhaust pipe located before and after the catalyst on each bank, and exhaust gas pressure and temperature sensors. The vehicle is equipped with a dSPACE rapid prototyping controller for data collection and controller implementation. Data are collected at each engine firing event under various driving conditions, such as federal test procedure (FTP cycles), as well as the more aggressive driving patterns (known as US06 driving cycles), for a length of about 95,000 samples during each test. The production engine is run under closed-loop fuel control using switching oxygen sensors. During testing of the control algorithms, the closed-loop fuel control system uses information from the wide-range AFR sensors. The dSPACE is interfaced with the powertrain control module in a by-pass mode. Neural networks offer one of the most attractive techniques to the design of the highly nonlinear and dynamic models[5,36] in such cases as AFR and TRQ estimation processes. Due to distinct characteristics of the TRQ and AFR processes, two neural network models are built for these processes with structures compatible with the mathematical engine models developed by Dobner[8,9] and others.

2.2. *Identification of TRQ using feedforward neural network*

In view of the fact that the torque generation process appears to be quasi-static and governed mostly by nonlinear properties, it is assumed that a simple multilayer feedforward network[17] should be sufficient for the process characterization and identification. The system model is represented by a neural network containing four input neurons, one hidden nonlinear layer using *tansig* function (i.e., the hyperbolic tangent function) in Matlab[7] with 10 neurons, and one linear output neuron.[16] The inputs to the model are: TPS (throttle position), MAP (manifold absolute pressure), RPM (engine speed) and SPA (spark advance) where TPS is the control signal and the other three inputs are reference signals compatible with the control signal at any operating conditions. The output is TRQ. The target is TRQ* which represents the measured values of engine torque generated by the engine using conventional engine controllers in open- and closed-loop operation. The TRQ network was initialized to small random weights between −0.18 and 0.18. The error backpropagation learning algorithm[11] in the batch mode was utilized for the training of the weights. The size of learning for TRQ modeling is 10000 events. The learning rate is set 0.01 at first. Then, the learning algorithm will automatically adjust the learning rate according to the performance. All the values of input and output are normalized between the range of −1 and 1 for the convenience of neural network training in both TRQ and AFR modelling. Figure 1 shows the validation result for the modelling of TRQ. Figure 2 shows the

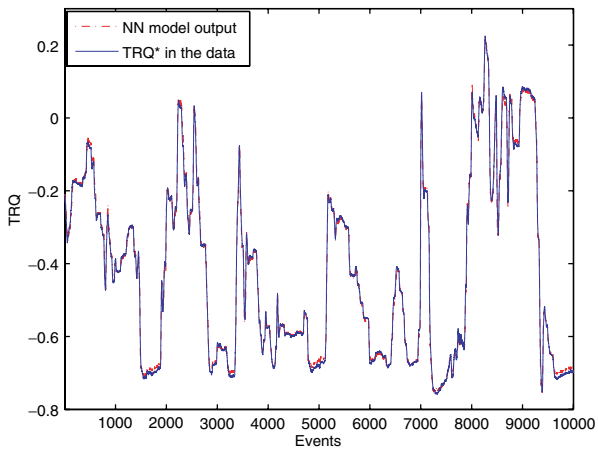

Fig. 1. Illustration of engine torque TRQ validation data.

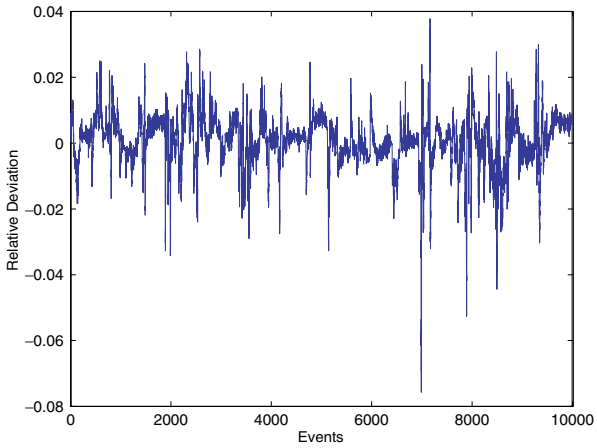

Fig. 2. IIllustration of relative deviation of the model from TRQ*.

relative deviation of the model from TRQ*, that is, $(\mathrm{TRQ} - \mathrm{TRQ}^*)/\max(\mathrm{TRQ}^*)$. Both figures demonstrate a high degree of model accuracy.

2.3. *Identification of AFR using nonlinear output error model*

Considering the AFR process dominated by purely dynamic effects, we choose nonlinear output error (NNOE) model[30] to model AFR dynamics. In practice, NNOE model has been used frequently for function approximation.[3,30] The general form of the NNOE model is:

$$\hat{y}(t|\theta) = F[\hat{y}(t-1|\theta), \quad \hat{y}(t-2|\theta), \ldots, \hat{y}(t-m|\theta), \\ u(t-1), \ldots, u(t-n)] \tag{1}$$

where $\hat{y}(t|\theta)$ is the output, θ is the adjustable parameter, $u(t)$ is a vector with several inputs, the indices $m = 2$, $n = 3$ define the lag space dimensions of external inputs and feedback variables, and F is a nonlinear mapping function realized by a recurrent neural network. We choose FPW (fuel pulse width) as the control input which is generated using our neural sliding mode controller. We also choose MAP and RPM as reference inputs. Thus, the inputs $u(t)$ to the model are: MAP, RPM and FPW. The input terms in (1) have the following structure:

$$\begin{pmatrix} u(t-1) \\ u(t-2) \\ u(t-3) \end{pmatrix} = \begin{pmatrix} \mathrm{MAP}(t-1) \\ \mathrm{RPM}(t-1) \\ \mathrm{FPW}(t-1) \\ \mathrm{MAP}(t-2) \\ \mathrm{RPM}(t-2) \\ \mathrm{FPW}(t-2) \\ \mathrm{MAP}(t-3) \\ \mathrm{RPM}(t-3) \\ \mathrm{FPW}(t-3) \end{pmatrix}.$$

The output quantity is the exhaust AFR. The target is AFR* which is obtained from the measured values of air fuel ratio by a wide-range AFR sensor in the dataset. The initial weights of the network were initialized to small random values between −0.18 and 0.18. Levenberg-Marquardt optimization algorithm was employed to update the weights of the NNOE neural model of AFR. The size of learning for AFR modeling is 5000 events. Figure 3 shows the validation result for the modelling of AFR. Figure 4 shows the relative deviation of the model from AFR*, that is, $(\mathrm{AFR} - \mathrm{AFR}^*)/\max(\mathrm{AFR}^*)$. Both figures indicate a very good match between the real vehicle data and the NNOE model output.

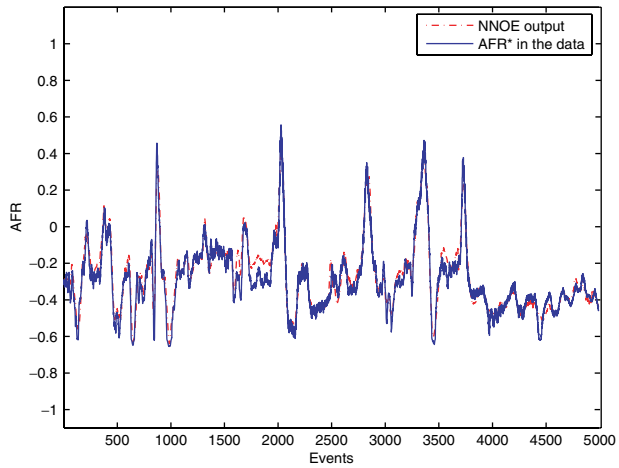

Fig. 3. Illustration of engine air-to-fuel ratio AFR validation data.

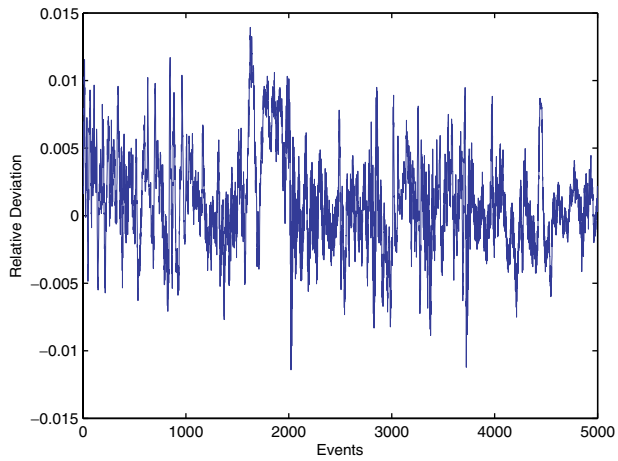

Fig. 4. Illustration of relative deviation of the model from AFR*.

3. Brief Introduction of Sliding Mode Control

The aim of SLMCs is to drive the system states to the sliding surface $S = 0$ and remain on the surface. Once the states are on the sliding surface, the system is insensitive to parameter variations or external disturbances. The sliding mode control design approach consists of two steps. The first step is to select a sliding surface that models the desired closed-loop performance in the state variable space according to design specifications. The second step is concerned with the selection of a control law which will drive

the system state trajectory toward the sliding surface and remain on it.

Consider the following nonlinear, multi-input multi-output system:

$$\dot{X} = G(X, U) \tag{2}$$

where $X \in R^p$ and $U \in R^q$ are the states and the control, respectively. The control problem is to find a control law so that the states X can track the desired trajectory X_d. Let the tracking error of the system be

$$e = X_d(t) - X(t). \tag{3}$$

The sliding surface S is defined in the state space by the scalar function $S(e) = 0$, where

$$S(e) = c^T e + d^T \dot{e} \tag{4}$$

$c = [c_1 \ c_2 \cdots c_n]^T$, $d = [d_1 \ d_2 \cdots d_n]^T$, $e = [e_1 \ e_2 \cdots e_n]^T$, and n is the number of parameters. The vector c and d are chosen so that $S(e) = 0$, which means the subsequent system is stable.[41] Hence, the control input can drive the system (2) to converge on the sliding surface.

The design of SLMC is based on the selection of Lyapunov function. The control should be chosen such that the candidate Lyapunov function satisfy Lyapunov stability criteria. In general, the control input of SLMCs for the system based on Lyapunov stability criteria is:

$$U(t) = U_{\text{eq}}(t) + U_c(t) \tag{5}$$

where U_{eq} represents the equivalent control which is the control action necessary to maintain an ideal sliding motion on the sliding surface and U_c represents the corrective control which drives the phase trajectory towards the sliding surface. The calculation of the equivalent control requires a good mathematical model of the system, G in Eq. (2), which is hard to obtain. The corrective control is given by $K\text{sign}(S)$ in conventional SLMCs which exhibits high frequency oscillations in its output, resulting in the chattering problem. The introduction of the boundary layer as a substitute for the sign function could provide a chattering-free system.[44] However, a finite steady error would always exist. To eliminate the chattering effect, we suggest that instead of the sign function, a saturation or a sigmoid function[12] is used. In our design, the shifted sigmoid function $T(S)$ is used to compute the corrective control:

$$U_c(t) = KT(S) \tag{6}$$

where $T(S)$ is chosen as follows:

$$T(S) = \frac{1 - e^{-S}}{1 + e^{-S}}. \tag{7}$$

Since the output of the corrective controller is continuous and variable, the chattering is effectively eliminated.

4. Neuro-Sliding Mode Control

In this study, considering the difficulty of the calculation of the equivalent control, we use neural networks to generate both the equivalent control and the corrective control in the SLMC. The combination of neural networks and SLMCs assures the desirable properties of both neural network and SLMCs are captured. Neural network- based equivalent control and corrective control possess the ability for self-adaptation to system uncertainties and also robustness to parameter variations and external disturbances. In this work, the goal of NSLMC design is to minimize the value of sliding function S such that the system states reach the sliding surface as soon as possible. Two neural networks in parallel are used to realize the equivalent control and corrective control of SLMC design as in Fig. 5 which shows where neural network 1 (NN1) is used to estimate the equivalent control (U_{eq}), and neural network 2 (NN2) is employed to generate the corrective control (U_c).

4.1. *Neural computation of the equivalent control*

The structure will be chosen as a two-layer feedforward neural network with one hidden layer and one

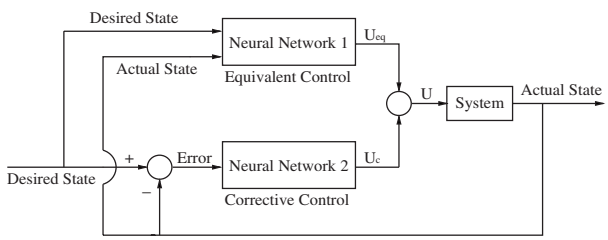

Fig. 5. Structure of neuro-sliding mode control.

output layer. The inputs to the neural network are the desired target and actual values of the states. The output of the neural network is the equivalent control U_{eq}. The weight adaptation of the neural network is based on a minimization of the cost function as follows:

$$E = \frac{1}{2}(U_{\text{eq}} - \hat{U}_{\text{eq}})^2 = \frac{1}{2}\zeta^2 \tag{8}$$

where $\hat{U}_{\text{eq}}$ is the estimated value of the equivalent control and $\zeta = U_{\text{eq}} - \hat{U}_{\text{eq}}$.

The Levenberg-Marquardt (L-M) algorithm is used to update the weights of NN1 instead of the error backpropagation (BP) algorithm. The selection of L-M algorithm is based on the fact that the L-M algorithm is widely accepted as the most efficient one in the sense of realization accuracy for nonlinear least squares.[15]

The formula for updating weights is given as follows:

$$\triangle W = [J^T(W)J(W) + \mu I]^{-1} J^T(W)\zeta \tag{9}$$

where the parameter μ is adjustable, W is adjustable weight vector and $J(W)$ is the Jacobian matrix. $J(W)$ can be expressed as follows:

$$J(W) = \left[\frac{\partial \zeta}{\partial W_1} \quad \frac{\partial \zeta}{\partial W_2} \quad \cdots \quad \frac{\partial \zeta}{\partial W_n}\right]^T. \tag{10}$$

From Eq. (8),

$$\frac{\partial \zeta}{\partial W_i} = \frac{\partial (U_{\text{eq}} - \hat{U}_{\text{eq}})}{\partial W_i} = -\frac{\partial \hat{U}_{\text{eq}}}{\partial W_i}. \tag{11}$$

Equation (11) can be calculated using the standard BP algorithm. Thus, the Jacobian matrix can be computed by Eq. (10) and Eq. (11).

From Eq. (8), we find that the desired equivalent control is unknown. To overcome this problem, it is suggested that $U_{\text{eq}} - \hat{U}_{\text{eq}}$ is replaced by the value of sliding function S since the characteristics of $U_{\text{eq}} - \hat{U}_{\text{eq}}$ and S are similar, that is, when S is close to 0, $U_{\text{eq}} - \hat{U}_{\text{eq}} \to 0$.[40]

4.2. *Neural computation of the corrective control*

The structure of the NN2 is decided by the design of SLMC. From Eq. (4) and Eq. (6), the gains of SLMC are represented as the weights of neural network 2 as shown in Fig. 6. In this way, the gains of SLMC are adapted gradually to the best values. The cost function is chosen to drive the states converging to sliding surface as follows:

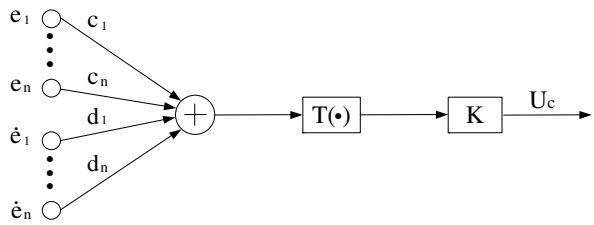

Fig. 6. Structure of neural network #2 for the corrective control.

$$J_c = \frac{1}{2}SS^T. \tag{12}$$

Minimization of J_c results in minimization of S. To minimize J_c, the weights are changed in the direction of the negative gradient,

$$\triangle K = -\mu \frac{\partial J_c}{\partial K} \tag{13}$$

$$\triangle c_i = -\mu \frac{\partial J_c}{\partial c_i} \tag{14}$$

$$\triangle d_i = -\mu \frac{\partial J_c}{\partial d_i} \tag{15}$$

where K is defined in Eq. (6), c_i and d_i are both defined in Eq. (4), and μ is the learning rate.

The gradient descent for c_i can be derived using Eq. (4) as:

$$\begin{aligned}\triangle c_i &= -\mu \frac{\partial J_c}{\partial c_i} = -\mu \frac{\partial J_c}{\partial S}\frac{\partial S}{\partial c_i} = -\mu S \frac{\partial S}{\partial c_i} \\ &= -\mu \cdot S \cdot e_i.\end{aligned} \tag{16}$$

The gradient descent for d_i can be derived using Eq. (4) as:

$$\begin{aligned}\triangle d_i &= -\mu \frac{\partial J_c}{\partial d_i} = -\mu \frac{\partial J_c}{\partial S}\frac{\partial S}{\partial d_i} = -\mu S \frac{\partial S}{\partial d_i} \\ &= -\mu \cdot S \cdot \dot{e}_i.\end{aligned} \tag{17}$$

The gradient descent for K can be derived as:

$$\triangle K = -\mu \frac{\partial J_c}{\partial K} = -\mu \frac{\partial J_c}{\partial S}\frac{\partial S}{\partial K} = -\mu S \frac{\partial S}{\partial K}. \tag{18}$$

From Eq. (4),

$$S(e) = c^T e + d^T \dot{e} = c^T(X_d - X) + d^T \dot{e}. \tag{19}$$

That is, from Eq. (2), Eq. (5), Eq. (6) and Eq. (19)

$$\frac{\partial S}{\partial K} = -c^T \frac{\partial X}{\partial K} = -c^T \frac{\partial X}{\partial U}\frac{\partial U}{\partial U_c}\frac{\partial U_c}{\partial K}$$
$$= -c^T \frac{\partial X}{\partial U} T(S) \tag{20}$$

where X is the state of the system and $T(S)$ is defined in Eq. (7). Since TRQ and AFR dynamics are represented by neural networks, according to the BP algorithm, $\partial X/\partial U$ can be derived easily. Finally,

$$\triangle K = -\mu S c^T \frac{\partial X}{\partial U} T(S). \tag{21}$$

The most significant feature of an SLMC is its robustness. When a system is in a sliding mode, it is insensitive to noise or external disturbances. Therefore, it is a good candidate for tracking control of uncertain nonlinear systems.[21]

5. Simulation Studies of NSLMC of a V8 Engine Control

The objective of the present engine controller design is to provide control signals, so that the TRQ generated by the engine will track the TRQ measurement as in the dataset and the AFR will track the target values also as in the dataset. The measured TRQ values in the data are generated by the engine using the existing controller. Our learning controller will assume no knowledge about the control signals provided by the existing controller. It will generate a set of control signals that are independent of the control signals in the measured data. Based on the data collected, we use our learning controller to generate control signals TPS and FPW with the goal of producing exactly the same TRQ and AFR as in the data set. That is to say, we require our system outputs follow the cooresponding measured values as in the dataset and build controllers that provide control signals that achieve the target TRQ and AFR control performance of the engine.

5.1. *NSLMC for engine control*

The neural network for the equivalent control for both TRQ and AFR is chosen as a 2-9-1 structure with 2 inputs and 9 hidden layer neurons. Both the hidden layer and the output layer use the *tansig* function. The block diagram of the present NSLMC controller for TRQ control is shown in Fig. 7. The

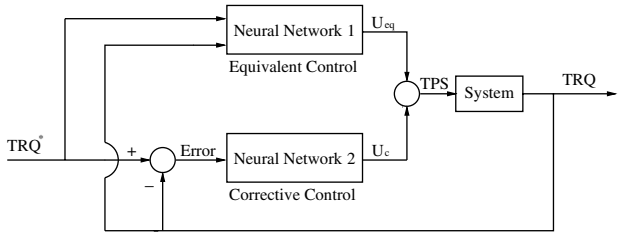

Fig. 7. Structure of NSLMC for the TRQ control.

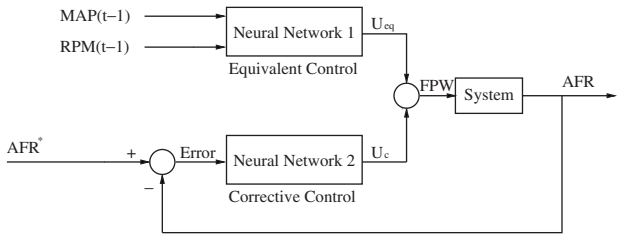

Fig. 8. Structure of NSLMC for the AFR control.

inputs to the NN1 are TRQ* (TRQ* is the commanded TRQ value) and the measured TRQ. The sum of two outputs of the NN1 and NN2 is TPS which is the control signal for the TRQ control loop. The block diagram for the AFR control is shown in Fig. 8. The inputs to the NN1 are MAP(t-1) and RPM(t-1) from the data. The sum of the two outputs of the NN1 and NN2 is FPW (fuel pulsewidth) which is the control signal for the AFR control loop. AFR* is the desired AFR value. Both TRQ* and AFR* are read from the data set, indicating the desired values for the present control algorithm to track. The TRQ and AFR dynamic systems are rpresented by the neural network models we built in Sec. 2.

The sliding surface S is defined as:

$$S = c_1 e + d_1 \dot{e} \tag{22}$$

where e represents the difference between the target (TRQ* or AFR*) and the actual values (TRQ or AFR). The sliding surface chosen in this way will lead to control objectives of TRQ which follows TRQ* and AFR which follows AFR*. The overall procedures of NSLMC technique for engine control are given as follows:

Step 1. Initialize: Set all weights of NN1 to small random values in the range of [−0.18 0.18]. The weights of NN2: c_i, d_i and K can be selected randomly, that is, they do not have to be selected with big initial values. The

range of [1 3] works well. In the simulations, c_i, d_i and K are set as 1 for TRQ control and 1, 1, 2, respectively, for AFR control. Set learning rate $0 < \mu < 1$. All values of signals were normalized to a range between -1 and 1 for convenience in the neural network training.

Step 2. Compute the equivalent control U_{eq} from NN1.

Step 3. Compute the corrective control U_c from NN2.

Step 4. Apply the sum of equivalent control and corrective control to the engine system.

Step 5. Measure the state of the engine system (TRQ or AFR).

Step 6. Adjust the weights of NN1 and NN2 according to the weight adaptation rules described in Sec. 4.

Step 7. Repeat by going to Step 2 until the convergence criterion is achieved.

5.2. *Simulation results*

We randomly choose to use 4000 points from the data (1000-5000 in the data set) for TRQ control and 3000 points (25000-28000 in the data set) for AFR control. Figure 9 and 13 show the controller performance for TRQ and AFR, respectively. Both figures show that excellent tracking control performance is achieved. We note that, at the present stage of the research, we have not attempted to regulate the AFR at the stoichiometric value but to track a given command. In these experiments, we simply try to track the measured engine-out AFR values, so that the control signal obtained can be directly validated against the measured control signals in the vehicle. Figures 10 and 14 show the output of the equivalent control and corrective control compared to the TPS or FPW values in the dataset. From the figures, we can see that most of the time, the corrective control is around zero. Only when the system states deviate from the sliding mode, the controller takes action to pull the system states back to the sliding surface. The trajectory of the sliding function S is as in Figs. 11 and 15. From the figures, we can see that only after 5 events, the sliding function S reaches the acceptable value (under 0.005 is generally regarded as acceptable). Figure 12 demonstrates good generalization ability of the NLSMC controller for TRQ

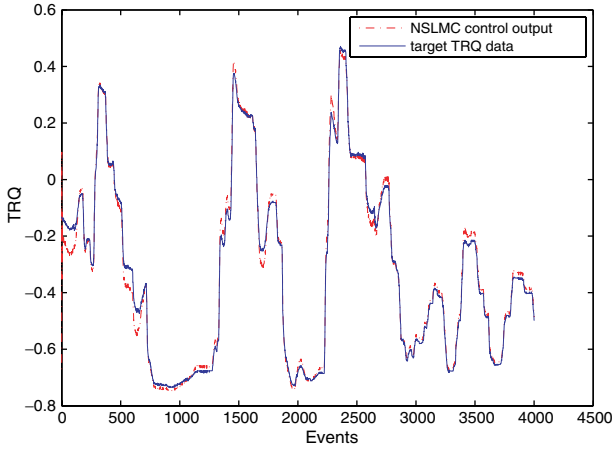

Fig. 9. Control effect of NSLMC for TRQ control.

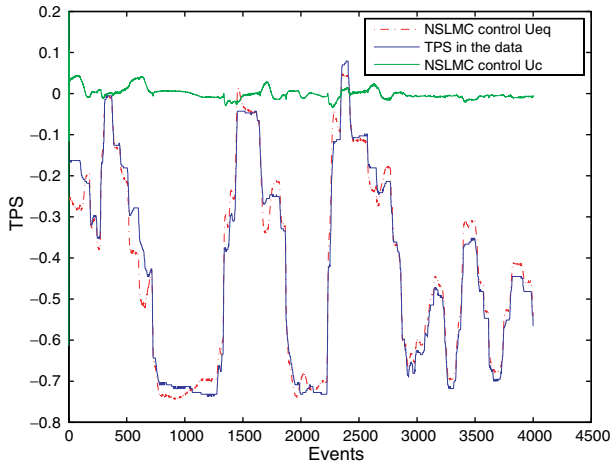

Fig. 10. Output of equivalent control, corrective control and TPS in the data.

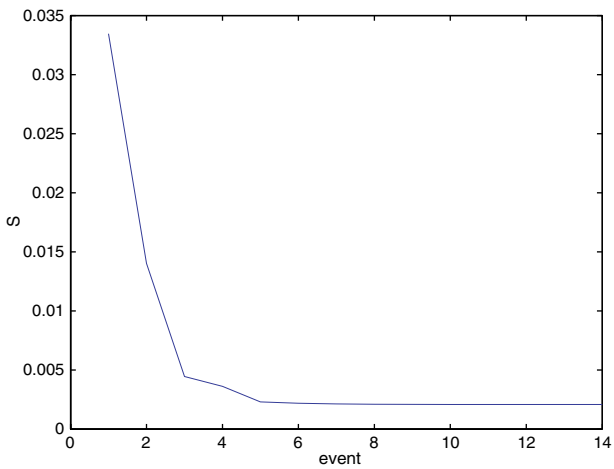

Fig. 11. Trajectory of sliding function S.

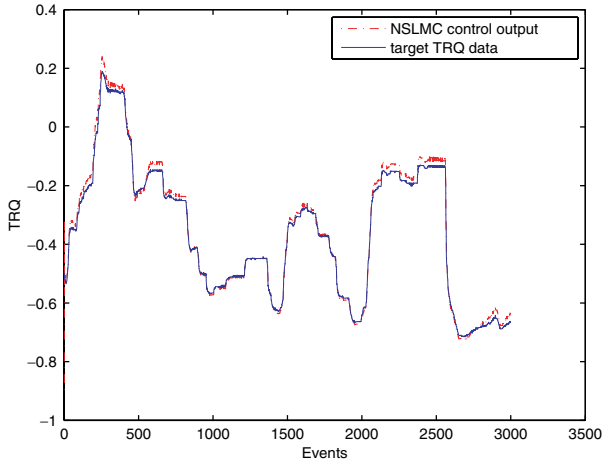

Fig. 12. NSLMC generalization ability for TRQ control.

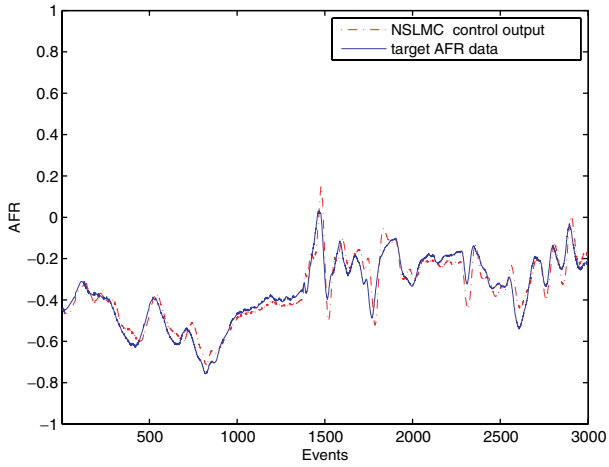

Fig. 13. Control effect of NSLMC for AFR control.

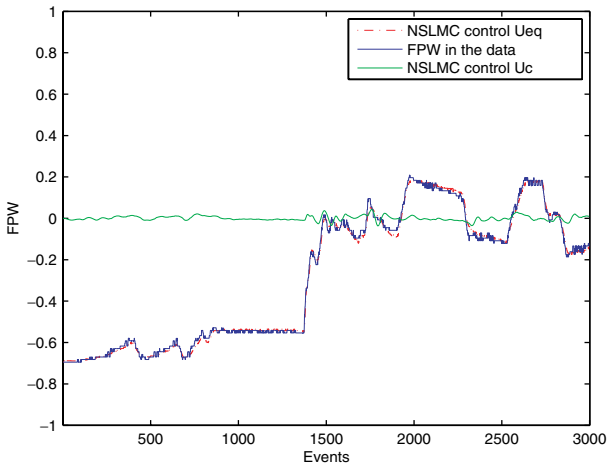

Fig. 14. Output of equivalent control, corrective control and FPW in the data.

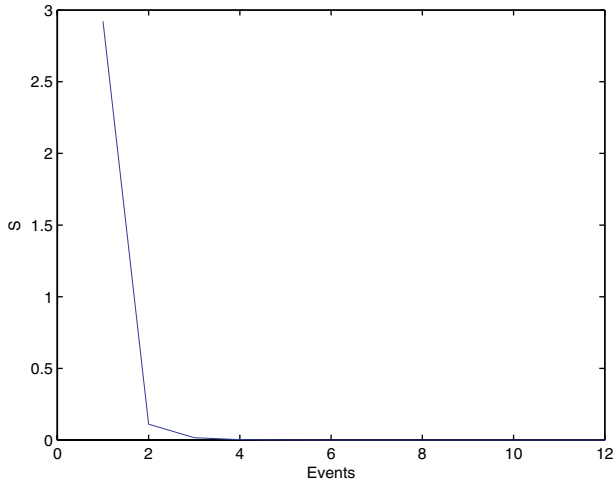

Fig. 15. Trajectory of sliding function S.

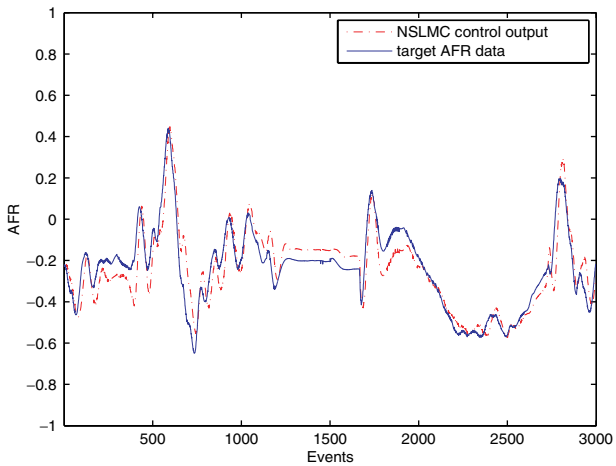

Fig. 16. NSLMC generalization ability for AFR control.

control and Fig. 16 demonstrates acceptable generalization ability of the NLSMC controller for AFR control.

5.3. *Discussions of the achieved results*

The figures shown in this section indicate that the present learning controller design based on the combination of neural network and sliding mode control is effective in training a neural network controller to track the desired TRQ and AFR sequences through proper control actions. The performance of TRQ controller is a little better than the performance of AFR controller mainly due to the fact that the process of AFR is far more complex than the TRQ process. Combustion TRQ is mostly governed

by process nonlinearities whereas the AFR process is mostly dominated by dynamic effects.

Figures 11 and 15 show the trajectory of the sliding function S. For a sliding mode-based control system, the dynamic behavior is determined by the sliding surface when the system is in the sliding mode. The sliding surface is chosen as $S(e) = c^T e + d^T \dot{e}$ which its value reflects the difference between the actual and the targeted trajectory. The sliding surface chosen in this way will lead to control objectives of TRQ which follows TRQ* and AFR which follows AFR*. When the sliding curve is near zero, it means that TRQ follows TRQ* and AFR tracks AFR*.

The adoption of L-M optimization method instead of BP for updating the weights of neural networks increases the convergence rates significantly. It took only 15 steps (about 1 minute) to achieve very good results for TRQ control and 20–30 steps (around 5 minutes) for AFR control (The configuration of computer is Pentium 4 with clock frequency of 3.2 GHz with 1GB RAM). Compared to the training technique used in (Ref. 36) which took one hour for TRQ control, the training speed is a lot quicker and there were no oscillations during the training.[16] The parameters of the neural network for the corrective control were set by trial and error in (Refs. 11 and 36) which is a time-consuming task, and they were selected as large values to achieve a fast convergence. In our scheme, the parameters values are selected as 1 or 2 which is enough to guarantee convergence. Since the controlled systems are modelled using neural network, the way of computation of K here is different from the method in (Refs. 11 and 36) where the computation of the gradient of K is acquired from the integral of $G(s)$. The integral of $G(s)$ would bring unpredictably big values in real experiments, resulting in potential instability. Boundary layer is set to assure stability in (Refs. 11 and 36).

6. Conclusions

Our research results have demonstrated that NSLMC provides a powerful alternative approach for engine calibration and control. The design is the combination of neural networks and sliding mode control. Two parallel neural networks are utilized to realize the equivalent control and the corrective control of SLMC. The successful application of NSLMC to TRQ and AFR control enables us to reach the following conclusions: (1) The proposed technique automatically learns the inherent dynamics and nonlinearities of engine processes from the real vehicle data and therefore no prior models and/or system characterization are necessary, (2) The present method involves neural network models of an engine for which a controller is designed and evaluated based on the models. Such a method will further advance the development of a virtual powertrain for performance evaluation of various control strategies in a test vehicle, and (3) Learning and control are carried out simultaneously with very fast convergence rates in the proposed controller. This allows the controller to be further refined and improved during real-world vehicle operation through continuous learning and adaptation. As such, this technique may offer an advantage as an enabling tool for real-time engine calibration and control.

The simulation results indicate that the proposed NSLMC are effective in achieving the TRQ and AFR tracking requirements through neural network learning. In a future work, we plan to implement the algorithms in an actual vehicle for further evaluation and possible refinements.

Acknowledgments

This work was supported by the National Science Foundation under Grants ECCS-0621694 and by the General Motors.

References

1. N. Al-Holou, T. Lahdhiri, D. S. Joo, J. Weaver and F. Al-Abbas, Sliding mode neural network inference fuzzy logic control for active suspension systems, *IEEE Trans. Fuzzy Syst.* **10** (2000) 259–268.
2. C. Alippi, C. D. Russis and V. Piuri, A neural-network based control solution to air-fuel ratio control for automotive fuel-injection systems, *IEEE Trans. Syst., Man, Cybern. C, Appl. Rev.* **33** (2003) 259–268.
3. I. Arsie, C. Pianese and M. Sorrentino, A procedure to enhance identification of recurrent neural networks for simulating air-fuel ratio dynamics in SI engines, *Eng. Appl. Artif. Intell.* **19** (2006) 65–77.
4. Q. R. Butt and A. I. Bhatti, Estimation of gasoline-engine parameters using higher Order sliding Mode, *IEEE Trans. Ind. Electron.* **55** (2008) 3891–3898.
5. Q. Cong, W. Yu and T. Chai T, Cascade process modeling with mechanism-based hierarchical neural networks, *Int. J. Neural Syst.* **20** (2010) 1–11.

6. F. P. Da, Decentralized sliding mode adaptive controller design based on fuzzy neural networks for interconnected uncertain nonlinear systems, *IEEE Trans. Neural Netw.* **11** (2000) 1471–1480.
7. H. Demuth and M. Beale, *Neural Network Toolbox User's Guide* (Natick, MA: MathWorks Inc., 1998).
8. D. J. Dobner, A mathematical engine model for development of dynamic engine control, *SAE Paper 800054* (1980).
9. D. J. Dobner, Dynamic engine models for control development Part I: Non-linear and linear model formation, *Int. J. Veh. Des.,* Special Publication SP4 (1983) 54–74.
10. H. Du and S. S. Nair, A neuro-sliding control approach for a class of nonlinear systems, *Proc. of 1st Int. Conf. on Knowledge-Based Intell. Elect. Syst.* (1997) 331–337.
11. E. Istook and T. Martinez, Improved backpropagation learning in neural networks with windowed momentum, *Int. J. Neural Syst.* **12** (2002) 303–318.
12. M. Ertugrul, O. Kaynak, A. Sabanovic and K. Ohnishi, A generalized approach for Lyapunov design of sliding mode controllers for motion control applications, *Proc. of Adv. Motion Control* (1996) 243–267.
13. M. Ertugrul and O. Kaynak, Neuro sliding mode control of robotic manipulators, *Mechatron* **10** (2000) 243–267.
14. A. Ferrara and P. Pisu, Minimum sensor second-order sliding mode longitudinal control of passenger vehicles, *IEEE Trans. Intell. Transp. Syst.* **5** (2004) 20–32 .
15. M. T. Hagan and M. B. Menhaj, Training feedforward networks with the Marquardt algorithm, *IEEE Trans. Neural Netw.* **5** (1994) 989–993.
16. T. Huang, D. Liu, H. Javaherian and N. Jin, Neural sliding-mode control of the engine torque, *Proc. of the 2008 IFAC Triennial World Congr.* (2008) 9453–9458.
17. H. T. Huynh, Y. Won and J. J. Kim, An improvement of extreme learning machine for compact single-hidden-layer feedforward neural networks, *Int. J. Neural Syst.* **18** (2008) 433–441.
18. C. L. Hwang, A novel Takagi-Sugeno-based robust adaptive fuzzy sliding-mode controller, *IEEE Trans. Fuzzy Syst.* **12** (2004) 676–687.
19. S. Janardhanan and B. Bandyopadhyay, Discrete sliding mode control of systems with unmatched uncertainty using multirate output feedback, *IEEE Trans. Automat. Contr.* **51** (2006) 1030–1035.
20. A. Karakasoglu and M. K. Sundareshan, A recurrent neural network based adaptive variable structure model-following control of robotic manipulators, *Automatica* **31** (1995) 1495–1507.
21. O. Kaynak, K. Erbatur and M. Ertugnrl, The fusion of computationally intelligent methodologies and sliding mode control-a survey, *IEEE Trans. Ind. Electron.* **48** (2001) 4–17.
22. M. K. Khan and S. K. Spurgeon, MIMO control of an IC engine using dynamic sliding modes, *Proc. of Intell. Syst. and Control* (2003).
23. A. J. Koshkouei and A. S. I. Zinober, Sliding mode control of discrete-time systems, *J. Dyn. Syst. Meas. Contr.* **122** (2000) 793–802.
24. M. Lee and H. S. Choi, A robust neural controller for underwater robot manipulators, *IEEE Trans. Neural Netw.* **11** (2000) 1465–1470.
25. P. Lee, S. Hong, Y. Lim, C. Lee, B. Jeon and J. Park, Discrete-time quasi-sliding mode control of an autonomous underwater vehicle, *IEEE J. Oceanic Eng.* **24** (1999) 388–395.
26. X. Li and S. Yurkovich, Neural network based, discrete adaptive sliding mode control for idle speed regulation in IC engines, *J. Dyn. Syst. Meas. Contr.* **122** (2000) 269–275.
27. F. J. Lin and R. J. Wai, Sliding-mode-controlled slider-crank mechanism with fuzzy neural network, *IEEE Trans. Ind. Electron.* **48** (2001) 60–70.
28. D. Liu, H. Javaherian, O. Kovalenko and T. Huang, Adaptive critic learning techniques for engine torque and air-fuel ratio control, *IEEE Trans. Syst. Man Cybern. Part B Cybern.* **38** (2008) 988–993.
29. J. J. Moskwa and J. K. Hedrick, Nonlinear algorithms for automotive engine control, *IEEE Control Syst. Mag.* **10** (1990) 88–93.
30. M. Norgaard, O. Ravn, N. L. Poulsen and L. K. Hansen, *Neural Networks for Modelling and Control of Dynamic Systems.* Berlin: Springer (2000).
31. S. Ouenou-Gamo, A. Rachid and M. Ouladsine, A nonlinear controller of a turbocharged diesel engine using sliding mode, *Proc. of IEEE Int. Conf. on Control Appl.* (1997) 803–805.
32. Y. Pan, U. Ozguner and O. H. Dagci, Variable-structure control of electronic throttle valve, *IEEE Trans. Ind. Electron.,* **55** (2008) 3899–3907.
33. S. Park, M. Yoon and M. Sunwoo, Feedback error learning neural networks for air-to-fuel ratio control in SI engines, *SAE Technical Paper 2003-01-0356* (2003).
34. G. G. Parma, B. R. De Menezes and A. P. Braga, Neural networks learning with sliding mode control: The sliding mode backpropagation algorithm, *Int. J. Neural Syst.* **9** (1999) 187–193.
35. J. K. Pieper and R. Mehrotra, Air/fuel ratio control using sliding mode methods, *Proc. of American Control Conf.* (1999) 1027–1031.
36. G. Puscasu, B. Codres, A. Stancu and G. Murariu, Nonlinear system identification based on internal recurrent neural networks, *Int. J. Neural Syst.* **19** (2009) 115–125.
37. B. Saerens, J. Vandersteen, T. Persoons, J. Swevers, M. Diehl and E. Van den Bulck, Minimization of the fuel consumption of a gasoline engine using

dynamic optimization, *Applied Energy* **86** (2009) 1582–1588.
38. J. J. E. Slotine, Sliding controller design for nonlinear systems, *Int. J. of Contr.* **40** (1984) 421–434.
39. J. S. Souder and J. K. Hedrick, Adaptive sliding mode control of air-fuel ratio in internal combustion engines, *Int. J. Robust Nonlinear Control* **14** (2004) 525–541.
40. C. Tsai, H. Chung and F. Yu, Neuro-sliding mode control with its applications to seesaw systems, *IEEE Trans. Neural Netw.* **15** (2004) 124–134.
41. V. I. Utkin, *Sliding Modes in Control and Optimization* (New York: Springer, 1992).
42. S. Wang and D. L. Yu, An application of second-order sliding mode control for IC engine fuel injection, *Canadian Conf. on Electr. and Comput. Eng.* (2006) 1035–1038.
43. M. Won, S. B. Choi and J. K. Hedrick, Air-to-fuel ratio control of spark ignition engines using Gaussian network sliding control, *IEEE Trans. Contr. Syst. Technol.* **6** (1998) 678–687.
44. D. Q. Zhang and S. K. Panda, Chattering-free and fast-response sliding mode controller, *IEE Proc. of Control Theory Appl.* **146** (1999) 171–177.