INTERNATIONAL JOURNAL OF
INFORMATION AND SYSTEMS SCIENCES
Volume 3, Number 1, Pages 129-149

COMPARISON OF SINGLE-DIMENSIONAL AND MULTI-DIMENSIONAL OPTIMIZATION APPROACHES IN ADAPTIVE MODEL PREDICTIVE CONTROL FOR AIR-FUEL RATIO OF SI ENGINES

Y. J. ZHAI, DING-LI YU AND KE-LI WANG

Abstract With development of fast modern computers, it has become possible to extend model predictive control (MPC) method to automotive engine control systems, which is traditionally applied to plants with dynamics slow enough to allow computations between samples. In this paper MPC based on an adaptive neural network model is attempted for air fuel ratio (AFR), in which the model is adapted on-line to cope with nonlinear dynamics and parameter uncertainties. A radial basis function (RBF) network is employed and the recursive least squares (RLS) algorithm is used for weight updating. Based on the adaptive model, a MPC strategy for controlling air-fuel ratio is realized to a nonlinear simulation of the engines. Finally, both single-variable and multi-variable optimizations algorithms are used to find the optimal solution of MPC problems and are compared in term of their control performance and time consumption.

Key Words, Air-fuel ratio control, SI engine, Adaptive neural networks, Nonlinear programming, Model predictive control

1. Introduction

Reducing pollutant emissions is an imperative and a continuous challenge for the automotive industry. For spark-ignition (SI) engines, maintaining the air/fuel ratio (AFR) at stoichiometric value (14.7) for both steady state and transient operation is the best solution to obtain the best balance between power output and fuel consumption. AFR can also influence the effect of emission control because its stoichiometric value ensures the maximum efficiency of three-way catalysts (TWC). Variations of greater than 1% below 14.7 can result in significant increase of CO and HC emissions. An increase of more than 1% will produce more NO_x up to 50% [1,2]. However, the dynamics of air manifold and fuel injection of SI engines are very fast, severely nonlinear and constraints are imposed on the states and inputs [3-6]. Therefore, they present a considerable problem to any control engineers.

Many of the current production fuel injection controllers utilize feed-forward control based on a mass airflow sensor located upstream of the throttle plus a proportional integral (PI) type feedback control. The feed-forward control with look-up tables requires a laborious process of calibration and tuning. Furthermore, it is difficult to apply this method since it needs the output magnitude information that is not available in the A/F ratio control [7]. A variety of researches have been conducted during past decade on advanced control strategies on AFR. Onder and Geering made an LQR regulator to

Received by the editors January 10, 2004

improve the air-fuel ratio control [8]. It obtained fairly good AFR when throttle angle ranging from 4° to 8°, but is impractical due to heavy computations resulting from the high order of linearized model. Choi and Hendrick made an attempt at developing an observer-based fuel injection control algorithm to improve AFR control by using sliding mode [9]. This analytic design method is in good agreement with the binary nature of the oxygen sensor signal. However, the controller is effective only when the throttle change is not rapid, since the controller depends mainly on feedback sensor information [7]. Manzie implemented a model predictive control (MPC) scheme for maintaining the air-fuel ratio, the RBF network was used as an observer of the air system and a linear predictive control algorithm was realized by using the active set method to solve quadratic programs. However, it is only effective for a small region around a specific operating point due to the highly nonlinear dynamics of both air intake and fuel injection. Therefore, the next generation of model based engine control on AFR should achieve a good level of accuracy within a wide range of engine operating conditions, with a limited computational demand.

A nonlinear MPC control scheme for air-fuel ratio based on a RBF model is presented in this paper. The RBF network is on-line adapted to model engine parameter uncertainties and severe nonlinear dynamics in different operating regions. Based on the multiple-step-ahead prediction of the air fuel ratio, an optimal control is obtained to maintain the stoichiometric value when engine speed and load change. In the next section of this paper, two types of nonlinear optimization algorithms are implemented to generate the optimal control signals of fuel injection according to the inputs from the RBF neural network model: (1) Secant Method *and* (2) Reduced Hessian Method. In both case 1 and 2, satisfactory AFR control results are obtained by using MPC scheme. Finally, the comparisons between two algorithms are presented according to the performance and time cost.

2. SI Engine Dynamics

Engine control system analysis and design based on engine simulation models are much more economical than using a real engine test bed in both industrial practice and scientific research. The developed control will then be evaluated on real test engine under realistic model-plant mismatch and noise provided the test engine is available. The engine model adopted in this paper is referred to as the mean value engine model (MVEM) developed by Hendricks [10], which is a widely used benchmark for engine modeling and control. The three distinct subsystems of this model are the fuel injection, manifold filling and the crankshaft speed dynamics and those systems are modeled independently. Since this MVEM can achieve a steady state accuracy of about $\pm 2\%$ over the entire operating range of the engine, it is extremely useful for validation of control strategies using simulation. A full description of the MVEM can be found in [11].

2.1. Manifold Filling Dynamics

The intake manifold filling dynamics are analyzed from the viewpoint of the air mass conservation inside the intake manifold. It includes two nonlinear differential equations, one for the manifold pressure and the other for the manifold temperature. The manifold pressure is mainly a function of the air mass flow past throttle plate, the air mass flow into the intake port, the exhaust gas re-circulation (EGR) mass flow, the EGR temperature and the manifold temperature. It is described as

$$\dot{p}_i = \frac{\kappa R}{V_i}\left(-\dot{m}_{ap}T_i + \dot{m}_{at}T_a + \dot{m}_{EGR}T_{EGR}\right) \tag{1}$$

The manifold temperature dynamics are described by the following differential equation

$$\dot{T}_i = \frac{RT_i}{p_i V_i}\left[-\dot{m}_{ap}(\kappa - 1)T_i + \dot{m}_{at}(\kappa T_a - T_i) + \dot{m}_{EGR}(\kappa T_{EGR} - T_i)\right] \tag{2}$$

In equation (1) and (2), the air mass flow dynamics in the intake manifold can be described as follows. The air mass flow past throttle plate $\dot{m}_{at}$ is related with the throttle position and the manifold pressure. The air mass flow into the intake port $\dot{m}_{ap}$ is represented by a well-known speed-density equation:

$$\dot{m}_{at}(u, p_i) = m_{at1}\frac{p_a}{\sqrt{T_a}}\beta_1(u)\beta_2(p_r) + m_{at0} \tag{3}$$

$$\dot{m}_{ap}(n, p_i) = \frac{V_d}{120RT_i}(\eta_i \cdot p_i)n \tag{4}$$

where

$$\beta_1(u) = 1 - \cos(u) - \frac{u_0^2}{2!} \tag{5}$$

$$\beta_2(p_r) = \begin{cases} \sqrt{1 - \left(\dfrac{p_r - p_c}{1 - p_c}\right)^2}, & if \quad p_r \geq p_c \\ 1, & if \quad p_r < p_c \end{cases} \tag{6}$$

$$p_r = \frac{p_i}{p_a} \tag{7}$$

and $m_{at0}, m_{at1}, u_0, p_c,$ are constants. Additionally, instead of directly model the volumetric efficiency η_i, it is easier to generate the quantity $\eta_i \cdot p_i$ which is called normalized air charge. The normalized air charge can be obtained by the steady state engine test and is approximated with the polynomial equation (8)

$$\eta_i \cdot p_i = s_i(n)p_i + y_i(n) \tag{8}$$

where $s_i(n)$ and $y_i(n)$ are positive, weak functions of the crankshaft speed and $y_i << s_i$

2.2. Crankshaft Speed Dynamics

The crankshaft speed is derived based on the conservation of the rotational energy on the crankshaft

$$\dot{n} = -\frac{1}{In}\left(P_f(p_i, n) + P_p(p_i, n) + P_b(n)\right) + \frac{1}{In}H_u\eta_i(p_i, n, \lambda)\dot{m}_f(t - \Delta\tau_d) \tag{9}$$

Both the friction power P_f and the pumping power P_p are related with the manifold

pressure p_i and the crankshaft speed n. The load power P_b is a function of the crankshaft speed n only. The indicated efficiency η_i is a function of the manifold pressure p_i, the crankshaft speed n and the air fuel ratio λ.

2.3. Fuel Injection Dynamics

According to Hendrick's identification experiments with SI engine, the fuel flow dynamics could be described as following equations [12]

$$\dot{m}_{ff} = \frac{1}{\tau_f}\left(-\dot{m}_{ff} + X_f \dot{m}_{fi}\right) \tag{10}$$

$$\dot{m}_{fv} = \left(1 - X_f\right)\dot{m}_{fi} \tag{11}$$

$$\dot{m}_f = \dot{m}_{fv} + \dot{m}_{ff} \tag{12}$$

where the model is based on keeping track of the fuel mass flow. The parameters in the model are the time constant for fuel evaporation, τ_f, and the proportion X_f of the fuel which is deposited on the intake manifold, $\dot{m}_{ff}$, or close to the intake valves, $\dot{m}_{fv}$. These parameters are operating point dependent and thus the model is nonlinear in spite of its linear form, which could be approximately expressed in terms of the states of the model as

$$\tau_f(p_i, n) = 1.35 \times (-0.672n + 1.68) \times (p_i - 0.825)^2 + (-0.06 \times n + 0.15) + 0.56 \tag{13}$$

$$X_f(p_i, n) = -0.277 p_i - 0.055n + 0.68 \tag{14}$$

2.4. Air Fuel Ratio Measurement

The expanded simulation model of SI engine is given in Figure 1.The output of the intake manifold sub-model is the air mass flow into intake port $\dot{m}_{ap}$, and the output of the fuel injection sub-model is the engine port fuel mass flow $\dot{m}_f$, therefore, the AFR could be calculated using equation (15)

$$\lambda = \frac{\dot{m}_{ap}}{\dot{m}_f} \tag{15}$$

In practice, time delay of injection systems should also be considered. Manzie's research [1,2] has shown there are three causes of time delay for injection systems: the two engine cycle delay between the injection fuel and the expulsion from the exhaust valves, the propagation delay for the exhaust gases to reach the oxygen sensor and the sensor output delay. It has been found that the engine speed has more influence on these delays than the manifold pressure. Therefore, the following equation can be used to represent the delays of injection systems

$$t_d = 0.045 + \frac{10\pi}{n} \tag{16}$$

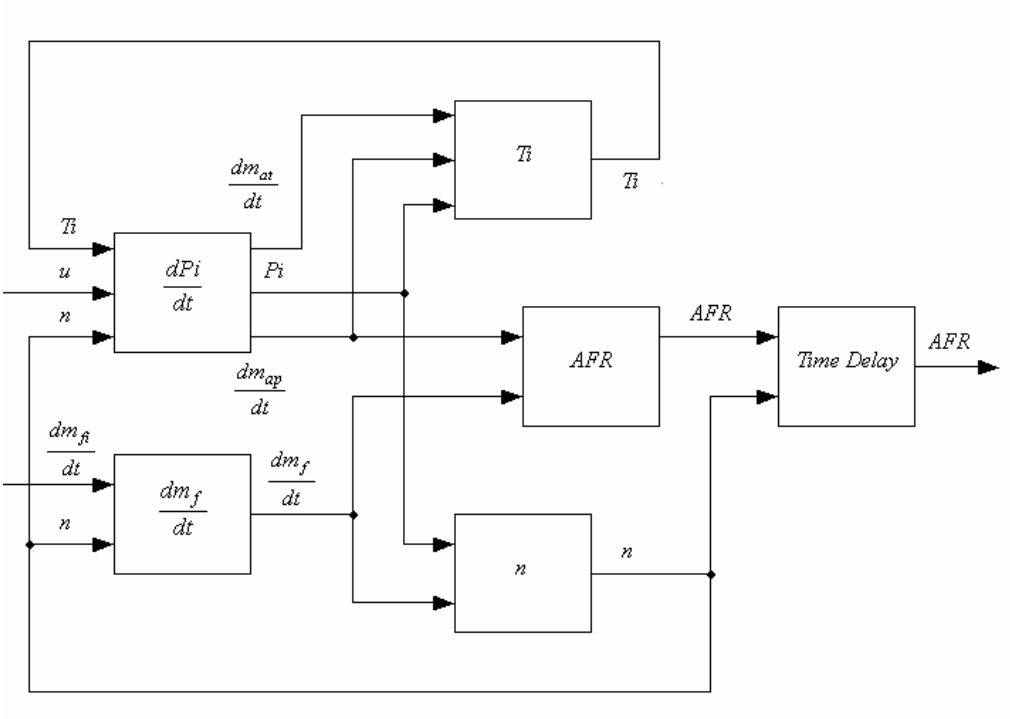


Figure 1. The Engine Simulation Model

3. Adaptive Neural Network Model

The advantage of using adaptive neural network is that it can track the time-varying properties of the process to provide efficient information to the controller, under circumstances where the process parameters are changing [13]. Radial basis function networks (RBFN) with Gaussian transfer function are chosen in this application as it has been shown that RBFN could map a nonlinear function arbitrarily well, and possess the best approximation property [14]. The *K*-means algorithm is used for center selection and ρ-nearest neighbor heuristic method used to determine the width σ for each hidden node in the RBFN.

3.1. RBF Network Structure

The RBFN, as shown in Figure 2, consists of three layers: input layer, hidden layer and output layer.

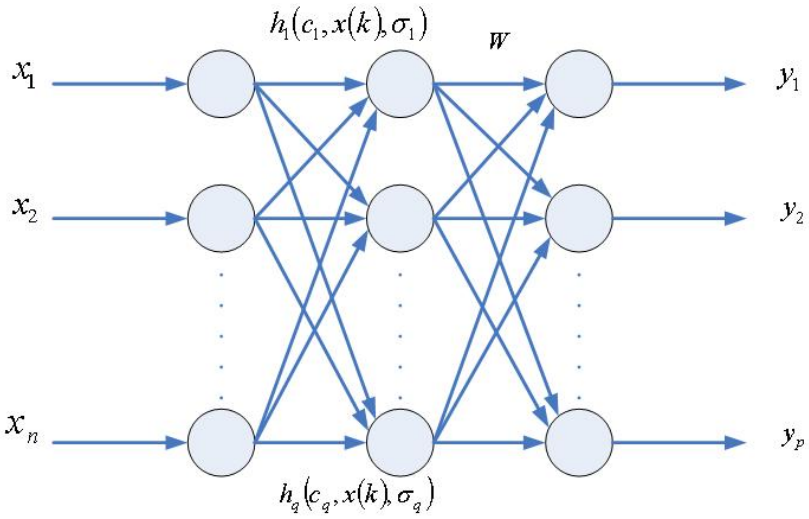


Figure 2. RBFN Structure

where $x(k) \in R^n$, $h(k) \in R^q$, $y(k) \in R^p$, c_i is the *i*th column vector of $C \in R^{n \times q}$, *i = 1, ...,q,* and $W(k) \in R^{p \times q}$.

$$C = \begin{bmatrix} c_{1,1} & \cdots & c_{1,q} \\ \vdots & \ddots & \vdots \\ c_{n,1} & \cdots & c_{n,q} \end{bmatrix} \tag{17}$$

$$W = \begin{bmatrix} w_1 \\ \vdots \\ w_p \end{bmatrix} = \begin{bmatrix} w_{1,1} & \cdots & w_{1,q} \\ \vdots & \ddots & \vdots \\ w_{p,1} & \cdots & w_{p,q} \end{bmatrix} \tag{18}$$

- x_i the *i*th node in the input layer, *i = 0, 1, ...n*
- h_i the output of the *i*th node in the hidden layer, *i = 1, 2, ...q.*
- c_i the *i*th centre in the hidden layer, where $c_i \in R^n$, *i = 1, 2, ..., q.*
- y_i the output of the *i*th node in the output layer, *i = 1, 2 , ..., p.*
- $w_{i,j}$ the weight linking the *j*th node in the hidden layer to the *i*th node in the output layer, *i = 1, 2, ..., p.* and *j = 1,2, ..., q.*

In mathematical terms, we have following equations to describe the RBFN.

$$y(k) = W \cdot h(k) \tag{19}$$

$$h(k) = f[z(k)] \tag{20}$$

$$z_i(k) = \left\| x(k) - [c_i]^T \right\| = \sqrt{[x(k) - c_i]^T [x(k) - c_i]} \tag{21}$$

where *i = 1, 2, ..., q.* and $f(\cdot)$ is the nonlinear activation function in hidden layer. The Gaussian-basis function is given by

$$f[z(k), \sigma] = e^{\frac{-z^2(k)}{\sigma^2}} \tag{22}$$

where σ is a positive scalar called a width, which is a distance scaling parameter to determine over what distance in the input space the unit will have a significant output.

3.1.1. *K*-means Algorithm

The *K*-means algorithm is based on minimization of a performance index, which is defined as the sum of the squared distance from all data points in cluster domains to their corresponding cluster centers.

Procedure:

- Choose *K* initial cluster centers $c_1(1)$, $c_2(1)$, …, $c_K(1)$, where *K* is equivalent to q
- At the *i*th iteration step, distribute the sample $\{x\}$ into $S_j(i)$ among the q cluster domains. $S_j(i)$ denotes the set of samples whose cluster is $c_j(i)$

$$x \in S_j(i) \ \ if \ \ \left\| x - c_j(i) \right\| < \left\| x - c_i(i) \right\| \tag{23}$$

- where *j = 1, 2, ..., K,* and *i = 1, 2, ..., K,* $i \neq j$
- Update the cluster centers.

$$c_j(i+1) = \frac{1}{N_j} \sum_{J}^{N_J} S_j(i) \tag{24}$$

  where N_j is the number of elements in $S_j(i)$.
- Repeat step 2 to step 3 until $c_j(i+1) = c_j(i)$.

3.1.2. ρ -Nearest Neighbors Method

Once the unit centers are established, the width σ of each unit can be determined through ρ -nearest neighbor heuristic, where σ for each hidden node are set the average distance from the centre to the ρ nearest centers.

$$\sigma_i = \left[\frac{1}{\rho} \sum_{j=1}^{\rho} \left\| c_i - c_j \right\|^2 \right]^{\frac{1}{2}} \tag{25}$$

where *i = 1, 2, ..., q,* c_j are the ρ -nearest neighbors of c_i. For nonlinear function approximation ρ depends on the problem and requires to be experimented.

3.2. Recursive Least Squares (RLS) Algorithm

Generally, the training algorithms for RBFN are least squares (LS) and recursive least

squares (RLS) method. However, the LS algorithm is used only for off-line learning. To deal with the ageing effect on engine control, RLS algorithm is adopted in this application for online learning. The algorithm of RLS could be summarized as follows [15]:

$$g(k) = \frac{P(k-1)h(k)}{\mu + h^T(k)P(k-1)h(k)} \tag{26}$$

$$P(k) = \mu^{-1}\left[P(k-1) - g(k)h^T(k)P(k-1)\right] \tag{27}$$

$$w(k) = w(k-1) + g(k)\left(y(k) - w^T(k-1)h(k)\right) \tag{28}$$

where $w(k)$ and $h(k)$ represent the RBF network weights and activation function outputs at iteration k, P and g are middle terms. μ here is called *forgetting factor* ranging from 0 to 1. The parameters g, w and P are updated orderly at each sampling time with the change of the activation function output $h(k)$.

3.3. Data Collection

In engine data collection, the training data must be representative of typical plant behavior in order to analyze the performance of different adaptive engine models in practical driving conditions. This means that input and output signals should adequately cover the region in which the system is going to be controlled [16]. A set of random amplitude signal (RAS) combining short pulse width (transient state) and long pulse width (steady state) was designed for throttle angle and fuel injection, therefore the RBFN model after trained would produce adequate transient and steady state performance.

Throttle angle was bounded between 20 and 40 *degree* and the range of fuel injection is from 0.0014 to 0.0079*(kg/s)*, the sample time is set to be 0.1*s*. The excitation signal is shown in Figure 3 partially, consisting of two parts. The length of square waves is set 0.3*s* in the first part and 1.5*s* in the second part. A set of 3000 data samples of AFR obtained was divided into two groups. The first 1500 data samples were used for training RBFN model and the rest would be remained for model validation.

3.4. Engine Modeling

Given the expanded engine model as shown Figure 1 in Section 2, the RBFN engine model has three inputs, fuel injection rate m_{fi}, throttle angle θ and air-fuel ratio y, and one output that is air-fuel ratio. Different orders of network inputs and different number of hidden layer nodes have been used in training experiments and the second-order structure with 12 hidden nodes is chosen after experiments as shown in Figure 4, which gives the minimum prediction error. The centers c and the width σ in hidden layer nodes of the RBFN were determined using *K*-means algorithm and ρ-nearest neighborhood heuristic respectively and ρ here is set to 3. RLS algorithm was used for training the neural network and the corresponding parameters were set as follows:

$$\mu = 0.99, \quad w(0) = 2.22 \times 10^{-16} \times U_{n_h \times 2}, \quad P(0) = 1 \times 10^4 \times I_{n_h \times n_h}$$

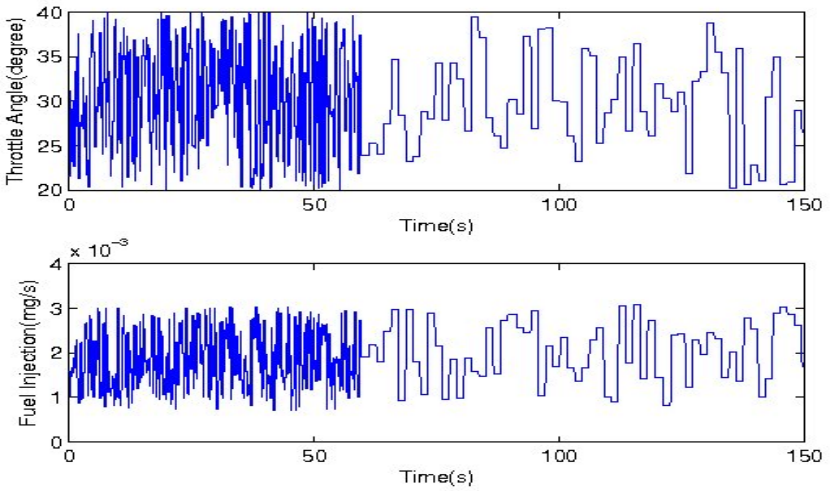


Figure 3. Training Data with Mixed Pulse Width

where I is the identity matrix and U stands for a matrix whose components are ones.

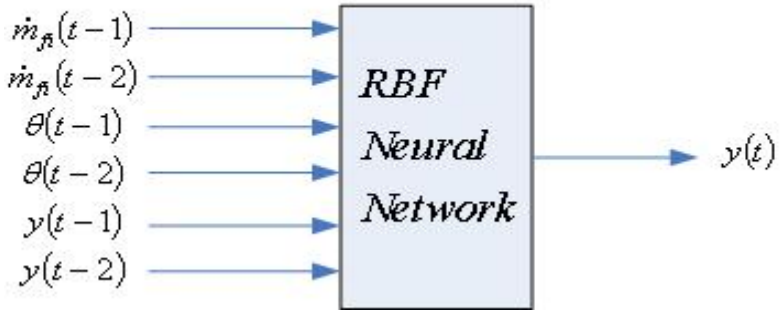


Figure 4. RBFN Structure

All input and output data of the RBFN have been scaled to the range of (0,1) before they are used for training and validation. The linear scale is used as follows:

$$u_s(k) = \frac{u(k) - u_{\min}}{u_{\max} - u_{\min}}, \quad y_s(k) = \frac{y(k) - y_{\min}}{y_{\max} - y_{\min}} \tag{29}$$

where $u_{\min}$, $u_{\max}$ are the minimum, maximum input among the data set while u_s is the scaled input. The same is for the output.

The training data set with 1500 samples are used to train the RBF network model. Then,

the test data set is applied to the trained model and the model predicts results for $k = 1 \sim 1300$ are displayed in Figure 5.

From the simulation result in Figure 5, we can see the good match between the real engine data and the RBFN output during the model validation phase, the AFR in the figure is normalized value and the mean absolute error (MAE) is 0.0265.

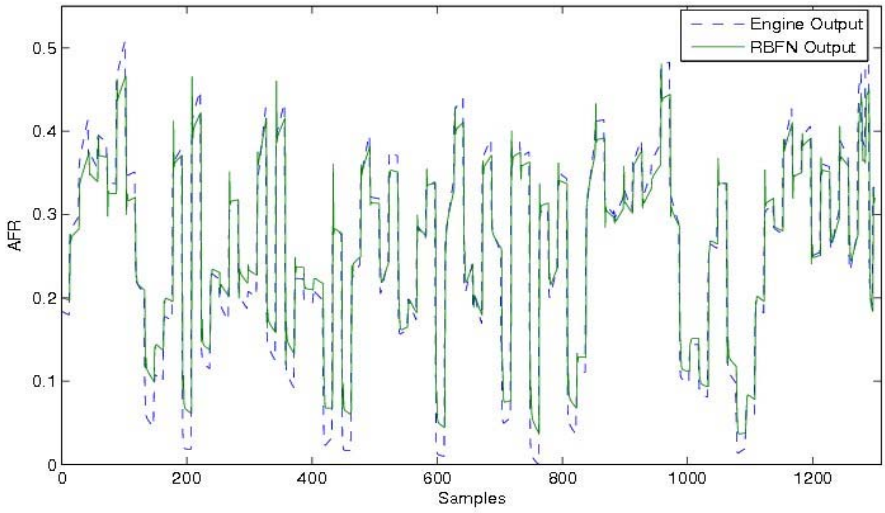


Figure 5. Illustration of AFR Validation Data

4. MPC of Air Fuel Ratio

4.1. Control System Structure

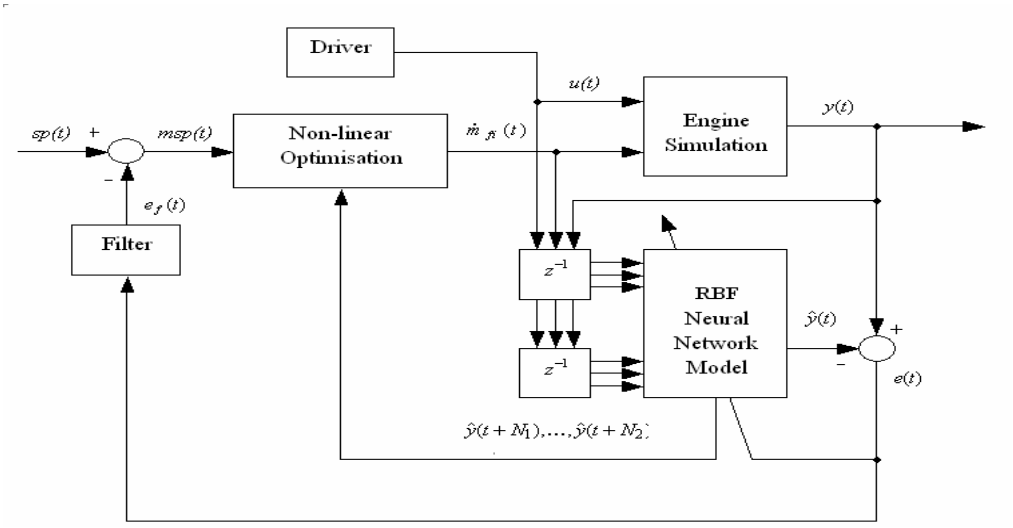


Figure 6. Configuration of Model Predictive Control on AFR

The idea of model predictive control with neural network has been introduced in details by Draeger [17]. The strategy is shown in Fig. 6. The obtained adaptive RBF neural network is used to predict the engine output for N_2 steps ahead. The nonlinear optimizer minimizes the errors between the set point and the engine output by using the cost function,

$$J(k) = \sum_{i=k+N_1}^{k+N_2} \left[msp(i) - \hat{y}(i)\right]^2 + \xi \sum_{i=k}^{k+N_u} \left[\dot{m}_{fi}(i) - \dot{m}_{fi}(i-1)\right]^2 \tag{30}$$

Here, N_1 and N_2 define the prediction horizon. ξ is a control weighting factor which penalizes excessive movement of the control input, the fuel injection $\dot{m}_{fi}$. N_u is the control horizon. Then the remaining main problem of MPC is to solve the nonlinear optimization problem, i.e. in each sample period, calculate a series of optimal $\dot{m}_{fi}(k), \dot{m}_{fi}(k+2), \cdots, \dot{m}_{fi}(k+N_2-1)$, from which the neural network model generates outputs to minimize $J(k)$ in (30). Finally the first control variable $\dot{m}_{fi}(k)$ is used to control the process and this procedure is repeated in the next sample period.

4.2. Single-Dimensional Optimization Approach

As second-order RBFN structure was chosen to achieve the minimum prediction error in engine modeling, the optimization problem involved in the paper is multi-dimensional and constrained. That is, we are going to find the future input $\dot{m}_{fi}(k), \dot{m}_{fi}(k+2), \cdots, \dot{m}_{fi}(k+N_2-1)$ that can minimize $J(k)$ such that the predicted outputs $\hat{y}(k), \hat{y}(k+1), \cdots, \hat{y}(k+N_2)$ coincides with the modified set-point input $mspi(k)$, $mspi(k+1)$, , $\cdots, mspi(k+N_2)$, here the fuel injection rate is bounded within the region from 0.0014 to 0.0079*(kg/s). Sequential Quadratic Programming (SQP)* can be used to acquire the accurate solution, which is perhaps one of the best methods of optimization, would be shown in next section. However, the multi-dimensional optimization always requires heavy computation, especially when constraints exist.

Practical applications often place emphasis on computation speed on the premise that all the performance requirements are met. Therefore, we chose the simplest structure in this paper and assumed that the input $\dot{m}_{fi}$ will remain constant over the prediction horizon: $\dot{m}_{fi}(k) = \dot{m}_{fi}(k+2) = \cdots, \dot{m}_{fi}(k+N_2-1)$, in this case， there is only one parameter $\dot{m}_{fi}(k)$ that we are going to find. The optimization problem to be solved is reduced as one-dimensional. *Secant Method* is chosen to find the solution of this *nonlinear programming (NLP)* problem and our experiments show that it is more efficient and reliable in this application if compared with the other interpolation methods.

4.2.1. Problem Formulation

The general nonlinear programming problem could be defined as,

$$\min_{x \in R^n} \; J(x) \tag{31}$$

subject to

$$\begin{aligned} c_{eq}(x) &= 0 \\ c_{in}(x) &\le 0 \end{aligned} \tag{32}$$

where $J : R^n \to R$ is the objective function, $c_{eq} : R^n \to R^m$ and $c_{in} : R^n \to R^p$ are constraint functions. All of these functions are smooth. Only inequality constraint applied in our case, as fuel injection rate is bounded within a region.

The *Secant Method* is to find the improved design vector X_{i+1} from the current design vector X_i using the formula

$$X_{i+1} = X_i + \zeta_i^* S_i \tag{33}$$

where S_i is the know search direction and ζ_i^* is the optimal step length found by solving the one-dimensional minimization problem as

$$\zeta_i^* = \min_{\zeta_i} [J(X_i + \zeta_i S_i)] \tag{34}$$

Here the objective function *J* is to be evaluated at any trial step length t_0 as

$$J(t_0) = J(X_i + t_0 S_i) \tag{35}$$

Similarly, the derivative of the function *J* with respect to ζ corresponding to the trial step length t_0 is to be found as

$$\left.\frac{dJ}{d\zeta}\right|_{\zeta=t_0} = S_i^T \Delta J\big|_{\zeta=t_0} \tag{36}$$

4.2.2. Secant Method

The necessary condition for $J(\zeta)$ to have a minimum of ζ^* is that $J'(\zeta^*) = 0$. The *Secant Method* seeks to find the root of this equation [18]. The equation is given with the form as follows,

$$J'(\zeta) = J'(\zeta_i) + s(\zeta - \zeta_i) = 0 \tag{37}$$

where *s* is the slope of the line connecting the two points $(A, J'(A))$ and $(B, J'(B))$, where *A* and *B* denote two different approximations to the correct solution, ζ^*. The slope *s* can be expressed as

$$s = \frac{J'(B) - J'(A)}{B - A} \tag{38}$$

Equation 37 approximates the function $J'(\zeta^*)$ between A and B as a linear equation and the solution of equation 37 gives the new approximation to the root of $J'(\zeta^*)$ as

$$\zeta_{i+1} = \zeta_i - \frac{J'(\zeta_i)}{s} = A - \frac{J'(A)(B - A)}{J'(B) - J'(A)} \tag{39}$$

The iteration process given in equation 39 is known as the *Secant Method*. See Figure 7.

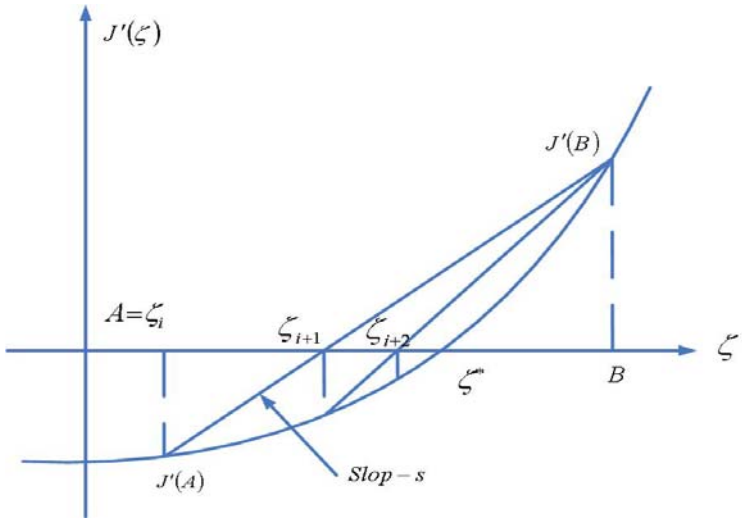


Figure 7. Iterative Process of *Secant Method*

4.2.3. Simulation Result Using Secant Method

In the simulation, the set-point of the system is set to be the constant stoichiometric value 14.7. The throttle angle θ is set as disturbance, a change from 25° to 30° with 0.5% uncertainty as shown in Figure 8. This is to evaluate the tracking performance and the robustness to throttle angle change of the designed system.

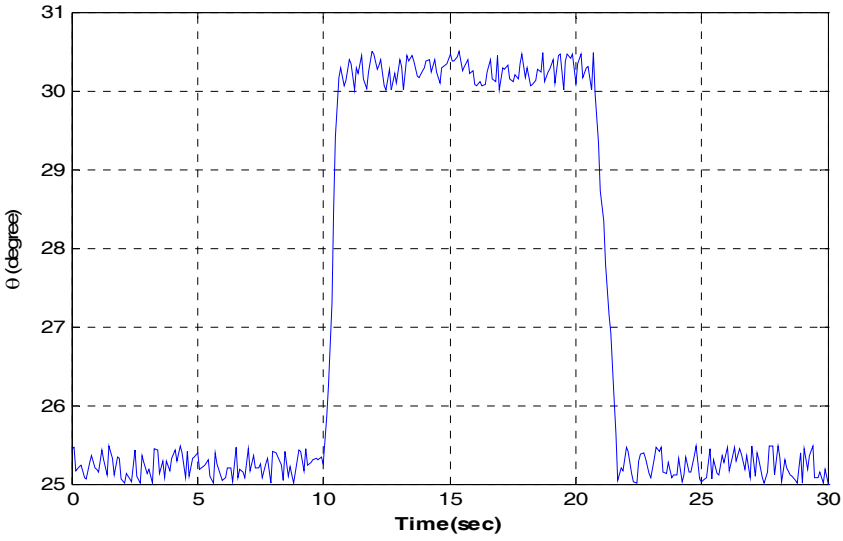


Figure 8. Throttle Angle Pattern in Simulations

The AFR is to be controlled between the $\pm 1\%$ bounds of the stoichiometric

value(14.7). Choosing the sampling time to be 0.1s. The parameters of nonlinear optimization were chosen as $N_1 = 1$, $N_2 = 6$, $\xi = 1$, $N_u = 0$, then the MPC of SI engines can be considered as a sub-problem of NLP problems.

$$\min_{x \in R^n} f(\dot{m}_{fi})$$

subject to

$$\dot{m}_{fi}{}^{l} \le \dot{m}_{fi} \le \dot{m}_{fi}{}^{u}$$

where $f : R^n \to R$, $\dot{m}_{fi}{}^{l}$ and $\dot{m}_{fi}{}^{u}$ represent the lower bound and the upper bound of the control variable $\dot{m}_{fi}$.

The system output under the developed MPC is displayed in Figure 9, together with the associated manipulated variable $\dot{m}_{fi}$ displayed in Figure 10. The mean absolute error (MAE) of the AFR tracking is 0.4464. One can see that the air-to-fuel ratio is regulated within a $\pm 1\%$ neighborhood of stoichiometric. This performance is much better than that of PI controller [19] that is widely used in automotive industry.

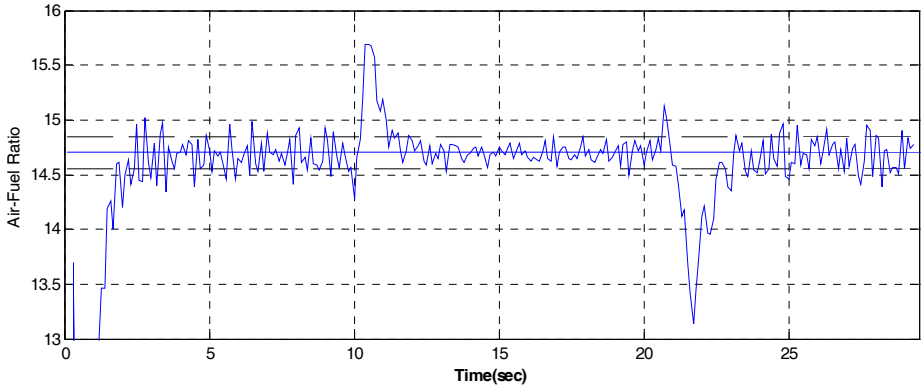


Figure 9. MPC on AFR Using Secant Method

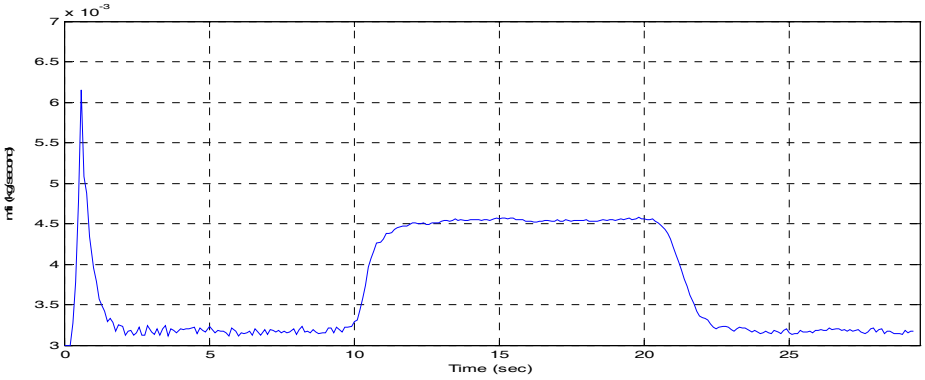


Figure 10. Fuel Injection Using Secant Method

The time cost in optimization in each sample period is shown in Figure 11. The mean time cost in one sample period is 0.0277 seconds. Since the whole simulation was running in ***Matlab*** environment, we feel that the further reduction on time cost of optimization could be achieved if optimization algorithm is realized by ***C*** code in real application.

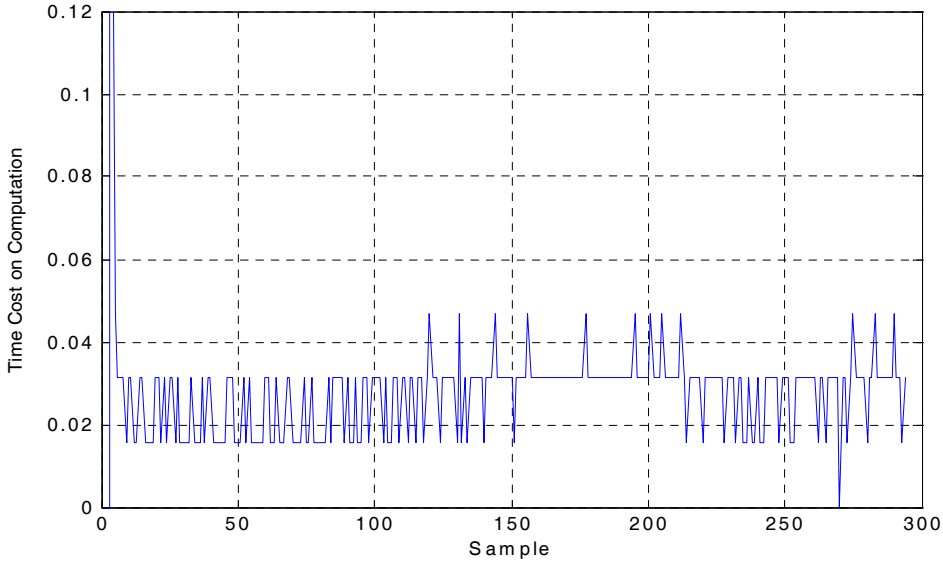


Figure 11. Time Cost on Optimization Using Secant Method

4.3. Multi-Dimensional Optimization Approach

The multi-dimensional approach for MPC was implemented using *Reduced Hessian Method* and is compared with *Secant Method*, in terms of the control performance and time consumptions on optimization. The *Reduced Hessian Method* is reviewed in the following.

4.3.1. Reduced Hessian Method

By applying SQP, the general nonlinear programming problem reduces to solving the following quadratic programming (QP) at each iteration. Find d that minimizes

$$g(x_k)^T d + \frac{1}{2} d^T W(x_k) d \tag{40}$$

subject to

$$\begin{aligned} c_{eq}(x_k) + A_{eq}(x_k)^T d &= 0 \\ c_{in}(x_k) + A_{in}(x_k)^T d &\le 0 \end{aligned} \tag{41}$$

Here, g denotes the gradient of f, W denotes the Hessian Matrix (with respect to x) of the Lagrangian function $L(x,\lambda) = f(x) + \lambda^T c(x)$, $A_{eq}(x)$ and $A_{in}(x)$ stand for the $n \times m$ and $n \times p$ matrices of constraint gradients,

Among different SQP methods, the reduced Hessian method is a newly developed algorithm for solving NLP problems subject to equality constraints ($c_{eq}(x) = 0$) [20,21].

It has been shown that the method is robust and less expensive to implement [22]. In order to illustrate how it can be implemented into MPC of SI engines, the basic idea of the reduced Hessian method is discussed in this section. In the following parts of this paper, *c(x)* and *A(x)* are used to represent $c_{eq}(x)$ and $A_{eq}(x)$ respectively.

4.3.2 The Search Direction

Assuming

$$A_k^T Z_k = 0 \tag{42}$$

then Z_k is a basis for the tangent space of the constraints. Now the solution *d* in equation (40) can be stated as

$$d_k = Y_k p_Y + Z_k p_z \tag{43}$$

where Z_k is a matrix spanning the null space of A_k^T, Y_k is a matrix spanning the range of A_k, p_Y and p_z are vectors in R^m and R^{n-m}, respectively. The problem of (43) becomes solving Y_k, p_Y, Z_k, p_z. By grouping the components of *x* into *m* basic or dependent variables and *n-m* non-basic or control variables, and $A(x)^T$ can be written as follows.

$$A(x)^T = [C(x) \quad N(x)] = \left[\left(\frac{\partial c}{\partial x}\right)^T \left(\frac{\partial c}{\partial u}\right)^T \right] \tag{44}$$

C(x) is assumed to be non-singular. Then defining

$$Z_k = \begin{bmatrix} -C(x)^{-1} N(x) \\ I \end{bmatrix} \quad \text{or} \quad Z_k = \begin{bmatrix} -N(x) \\ C(x) \end{bmatrix} \tag{45}$$

$$Y_k = \begin{bmatrix} I \\ 0 \end{bmatrix} \tag{46}$$

According to (42), the following equation is obtained

$$p_Y = -\left[A_k^T Y_k\right]^{-1} c_k \tag{47}$$

Substituting p_Y into (43) and then into (41) to compute the minimum value under the assumption that $Z_k^T W_k Z_k$ is positive definite, the solution is

$$p_z = -(Z_k^T W_k Z_k)^{-1} \left[Z_k^T g_k + Z_k^T W_k Y_k p_Y \right] \tag{48}$$

$Z_k^T W_k Z_k$ is the reduced Hessian. In order to eliminate the computational work required to evaluate W_k and $Z_k^T W_k Z_k$, the BFGS algorithm is used to approximate the reduced Hessian $Z_k^T W_k Z_k$. As for the cross term $Z_k^T W_k Y_k p_Y$, there are two methods to deal with it. The first one is to omit it and the second one is to use a vector w_k as its approximation by means of finite-difference approximation or quasi-Newton approximation (Broyden's update).

4.3.3 Line Search and Stopping Criterion

Before updating x_* by using $x_{k+1} = x_k + \alpha_k d_k$, α_k needs to be tested according to

$$\phi_{\mu k}(x_k + \alpha_k d_k) \le \phi_{\mu k}(x_k) + 0.1\alpha_k D\phi_{\mu k}(x_k; d_k) \tag{49}$$

where $\phi_{\mu k}(x_k) = f_k + \mu_k \|c_k\|_1$ and $D\phi_{\mu k}(x_k; d_k) = g_k^T d_k - \mu_k \|c_k\|_1$ (μ_k is chosen by users). If the above formula is not satisfied, a new α_k should be chosen as

$$\alpha_k = \max\left\{\frac{-0.5D\phi_{\mu k}(x_k; d_k)\alpha_k^2}{\phi_{\mu k}(x_k + \alpha_k d_k) - \phi_{\mu k}(x_k) - \alpha_k D\phi_{\mu k}(x_k; d_k)}, 0.1\right\} \tag{50}$$

If a solution of this optimization problem is denoted by x_*, define $e_k = x_k - x_*$ and $\sigma_k = \max\{\|e_k\|, \|e_{k+1}\|\}$. If $\sigma_k \le tol$ (*tol* is defined by users.), the algorithm can be stopped.

4.3.4. Simulation Result Using Reduced Hessian Method

During the implementation of *Reduced Hessian Method*, the parameters of the nonlinear optimization were the same as one-dimensional case. To solve the MPC optimization problem by reduced Hessian method, two slack variables x_1 and x_2 are introduced to convert the inequality constraint to become two equality constraints. Correspondingly, a quadratic penalty function $x_1^2 + x_2^2$ with a parameter μ_s is added into the objective function *J* in (30). Then the MPC optimization problem becomes: Find a suitable control variable $\dot{m}_{fi}$ to minimize

$$f = J + \mu_s \cdot \left(x_1^2 + x_2^2\right) \tag{51}$$

subject to

$$\left(\dot{m}_{fi} - \dot{m}_{fi}{}^{l}\right) - x_1^2 = 0 \quad \text{and} \quad \left(\dot{m}_{fi} - \dot{m}_{fi}{}^{u}\right) + x_2^2 = 0 \tag{52}$$

Here $\dot{m}_{fi}{}^{l}$ and $\dot{m}_{fi}{}^{u}$ are respectively 0.0014 kg/sec and 0.0079 kg/sec. After being scaled, the two bounds become 0 and 1. In order to reduce the influence of the slack variables x_1 and x_2 on the new objective function, the parameter μ_s is chosen to be 10^{-10}. To stop the optimization program at a suitable time, the tolerance *tol* is set be 10^{-7}.

With the above modification and parameters setting, the simulation results are shown in Figure 12, 13, the tracking mean absolute error is 0.4465. The time cost in optimization is shown in Figure 14 for comparison with the performance of *Secant Method.*

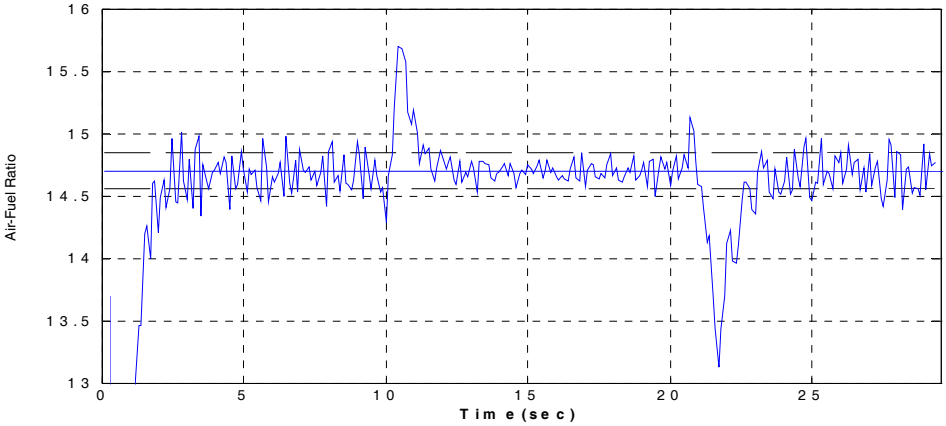


Figure 12. MPC on AFR Using Reduced Hessian Method

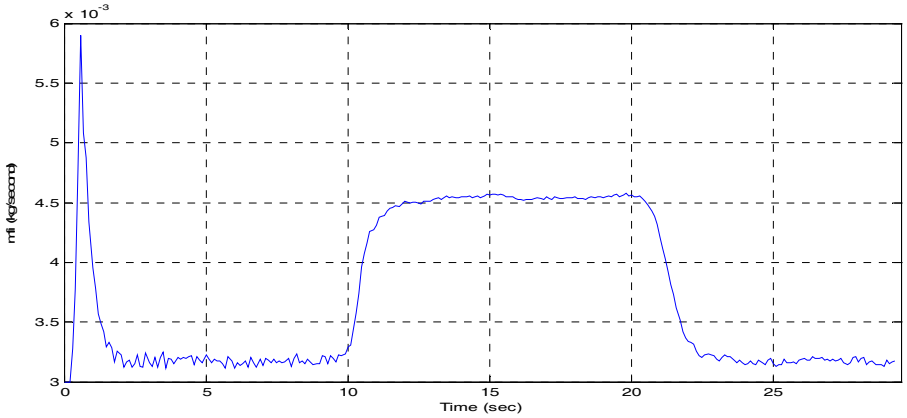


Figure 13. Fuel Injection Using Reduced Hessian Method

The simulation results show that *Reduced Hessian Method* has the similar tracking performance of *Secant Method*, however, its time consumption in optimization is much more than that of previous method. In our experiment, the mean time cost in one sample period using this method is 0.0473 seconds that is nearly twice as many as that used by *Secant Method*.

5. Conclusions

In this paper, Adaptive RBF model based MPC is applied to AFR control of automotive engines. The simulation results validated that the developed method can control the AFR to track the set-point value under disturbance of changing throttle angle.

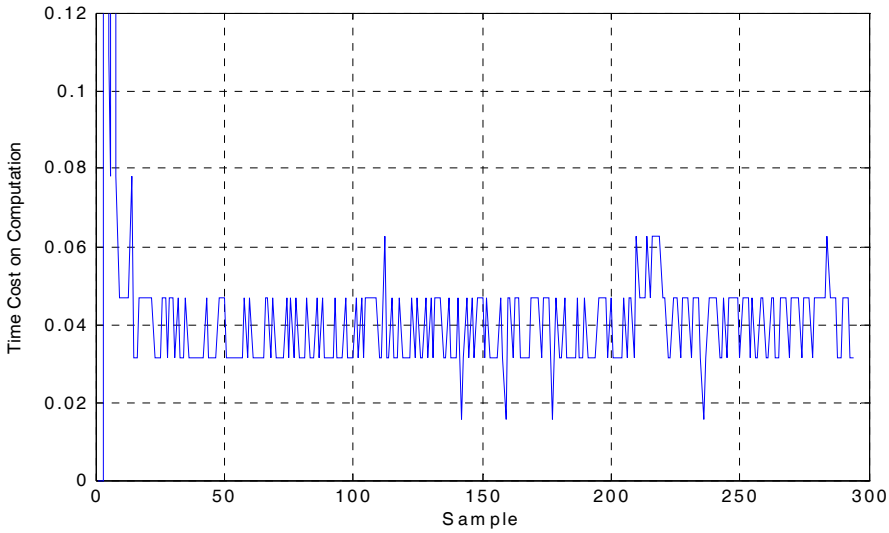


Figure 14. Time Cost on Optimization Using Reduced Hessian Method

To meet the requirement for fast optimization in engine control, a one-dimensional optimization method, *Secant Method*, is implemented in the MPC and is compared with the multi-dimensional method, *Reduced Hessian Method.* Simulations show a much shorter optimization time using *Secant Method* and the achieved tracking control with similar performance to that in *Reduced Hessian Method.*

In summary, the adaptive neural network model-based control on automotive engine with fast optimization algorithm has demonstrated attractive operating properties and merit further consideration for the advanced engine control.

REFERENCES

[1] Manzie C. Palaniswami M. and Watson H. Gaussian networks for fuel injection control. Proceedings of the Institution of Mechanical Engineers, Part D, Journal of Automobile Engineering, 2001, 215(D10), 1053-1068.

[2] Manzie C. Palaniswami M. Ralph D. Watson H. and Yi X. Model predictive control of a fuel injection system with a radial basis function network observer. Journal of Dynamic Systems Measurement and Control Transactions of the ASME, 2002, 124(4), 648-658.

[3] Balluchi A, Benvenuti L, Di Benedetto MD, Pinello C, Sangiovanni-Vincentelli AL. Automotive engine control and hybrid systems: Challenges and opportunities. Proceeding of the IEEE, 2000, 88(7), 888-912.

[4] De Nicolao G., Scattolini R. and Siviero C. Modelling the volumetric efficiency of IC engines: parametric, non-parametric and neural techniques. Control Eng. Practice, 1996, 4(10), 1405-1415.

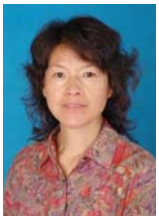
[5] Tan Yonghong and Mehrdad Saif Neural-networks-based nonlinear dynamic modeling for automotive engines. Neurocomputing, 2000, 30, 129-142.

[6] Vinsonneau J. A. F., Shields D. N., King P.J. and Burnham K. J. Polynomial and neural network

spark ignition engine intake manifold modeling. Proc. 16th Int. Conf. on Systems Engineering, ICSE 2003, 2,718-723.
[7] Mooncheol Won, Seibum B. Choi, and J. K. Hedrick Air-to-Fuel Ratio Control of Spark Ignition Engines Using Gaussian Network Sliding Control. IEEE Transactions on Control Systems Technology, Vol. 6, NO. 5, September 1998.
[8] Onder,C.H., and Geering, H.P. Model-based Multivariable Speed and Air-to-Fuel Ratio Control of an SI Engine. SAE Technical Paper No. 930859, 1993.
[9] Choi S. B. and Hendricks J. K. An observer-based controller design method for improving air-fuel characteristics of spark ignition engines. IEEE Transactions on Control Systems Technology, 1998, 6(3), 325-334.
[10] D.Cho and Hendricks J. K. Automotive Powertrain Modeling for Control. Transaction of the ASME, pp.568-576,1989.
[11] Hendricks, E., and Sorenson, S. C. Mean Value Modelling for Spark Ignition Engines. SAE Technical Paper No. 900616, Transactions of the SAE, 1990
[12] Hendricks, E., Engler, D., Fam, M. A Generic Mean Value Engine Model for Spark Ignition Engines. Proceedings of 41st Simulation Conference, DTU Lyngby, Denmark, SIMS 2000
[13] Chang Thoon Khin Process Fault-tolerant Control Based on Adaptive Neural Network, Liverpool John Moores University, 2001
[14] F.Girosi and T. Poggio Network and the Best Approximation Property, Biological Cybernetics, Vol. 63, pp.169-176, 1999.
[15] Zhi Tian, Kristine L. Bell, Harry L. Van Trees A Recursive Least Squares Implementation for LCMP Beamforming Under Quadratic Constraint. IEEE Transactions on Signal Processing, Vol 49. NO. 6. June 2001.
[16] Orlando De Jesus, Arjpolson Pukrittayakamee, Martin T. Hagan A Comparison of Neural Network Control Algorithms.IEEE Transactions on Neural Network, 2001
[17] A. Draeger, S. Engell, and H. Ranke Model Predictive Control Using Neural Networks. IEEE Control Systems Magazine, pp. 61-66, 1995.
[18] Singiresu S. Rao Engineering Optimization, John Wiley Sons, Inc., 1996
[19] Shiwei. Wang, Dingli, Yu, J.B. Gomm, G.F. Page, and S.S. Douglas Adaptive Neural Network Model Based Predictive Control for Air-fuel Ratio of SI Engine. Engineering Application of Artificial Intelligence 19, pp.189-200, 2006
[20] Biegler L. T., Nocedal J. and Schmid C., A reduced Hessian method for large scale constrained optimization. SIAM Journal of Optimization, 1995, 5(2), 314-347.
[21] Biegler L. T., Nocedal J., Schmid C. and Ternet D., Numerical Experience with a Reduced Hessian Method for Large Scale Constrained Optimization. Computational Optimization and Applications, 2000, 15(1), 45-67.
[22] Hendricks E., A generic mean value engine model for spark ignition engines. Proc. of 41st Simulation Conference, SIMS 2000, DTU, Lyngby, Denmark. 2000.

Yu-Jia Zhai, Control Systems Research Group, School of Engineering
Liverpool John Moores University, Byrom Street, Liverpool L3 3AF, U.K.
Email: y.zhai@2005.ljmu.ac.uk

Ding-Li Yu, Control Systems Research Group, School of Engineering
Liverpool John Moores University, Byrom Street, Liverpool L3 3AF, U.K.
Email: D.Yu@ljmu.ac.uk



Ke-Li Wang, Department of Inorganic Chemistry, Jilin Agricultural Polytechnic, Jilin Province, China. E-mail: wangkeli5@yahoo.com.cn She joined Jilin Agricultural Polytechnic since 1981, she teaches and conducts research in Inorganic Chemistry