Neural Comput & Applic (2016) 27:79–92
DOI 10.1007/s00521-014-1555-7

Model predictive engine air-ratio control using online sequential extreme learning machine

Pak Kin Wong · Hang Cheong Wong · Chi Man Vong · Zhengchao Xie · Shaojia Huang

Received: 10 September 2013 / Accepted: 31 January 2014 / Published online: 8 February 2014

Abstract Air-ratio is an important engine parameter that relates closely to engine emissions, power, and brake-specific fuel consumption. Model predictive controller (MPC) is a well-known technique for air-ratio control. This paper utilizes an advanced modelling technique, called online sequential extreme learning machine (OSELM), to develop an online sequential extreme learning machine MPC (OEMPC) for air-ratio regulation according to various engine loads. The proposed OEMPC was implemented on a real engine to verify its effectiveness. Its control performance is also compared with the latest MPC for engine air-ratio control, namely diagonal recurrent neural network MPC, and conventional proportional–integral–derivative (PID) controller. Experimental results show the superiority of the proposed OEMPC over the other two controllers, which can more effectively regulate the air-ratio to specific target values under external disturbance. Therefore, the proposed OEMPC is a promising scheme to replace conventional PID controller for engine air-ratio control.

Keywords Online sequential extreme learning machine · Nonlinear model predictive control · Automotive engine · Air-ratio

P. K. Wong (✉) · H. C. Wong · Z. Xie · S. Huang
Department of Electromechanical Engineering, Faculty of Science and Technology, University of Macau, Taipa, Macao
e-mail: fstpkw@umac.mo

C. M. Vong
Department of Computer and Information Science, Faculty of Science and Technology, University of Macau, Taipa, Macao

1 Introduction

Air-ratio (also known as lambda) is an engine parameter indicating the amount of difference between the actual available air–fuel mixture and the stoichiometric ratio of the fuel being used [1]. The air-ratio plays an important role in engine performance control because it relates closely to engine emissions, power, and brake-specific fuel consumption. The rapid growth in the number of vehicles makes vehicular emissions become a major source of air pollution, which impacts on the public health and environment significantly, especially in urban areas. Studies [2–6] show that every year about hundred thousand of mortalities all over the world are related to vehicular emissions, resulting in billions of economic loss. To reduce the amount of toxic elements in vehicular emissions, the three-way catalytic converter is currently the most effective after-treatment device. When the air-ratio is stoichiometry (i.e., air-ratio = 1.0), the conversion efficiency of the three-way catalytic converter can reach as high as 98 %. However, the conversion efficiencies for carbon monoxide (CO), hydrocarbons (HC), and nitrogen oxides (NO_X) drop more than 50 % when the air-ratio deviates more than 2 % from stoichiometry, causing more severe air pollution. Moreover, air-ratio can significantly affect brake-specific fuel consumption. Recent studies [7, 8] have showed that the air-ratio should be regulated to a value around 1.05 for achieving the best brake-specific fuel consumption. Furthermore, the maximum engine power can be achieved when the air-ratio is around 0.95. Therefore, optimal air-ratio control according to various operating conditions is significant and desired.

In the existing literature, there are a lot of control strategies on air–fuel ratio (AFR) rather than air-ratio. These strategies include sliding mode control [9], radial-basis-function neural network feedforward–feedback control [10], and model predictive control (MPC) using neural network-based models [11, 12]. Nevertheless, AFR control is ineffective for engine performance control because it does not consider the fuel variation. On the contrary, air-ratio is a fuel-independent index that can reflect the engine performance more effectively. Thus, this study focuses on air-ratio control rather than AFR. As air-ratio is associated with AFR, the control strategies developed for AFR can also be adopted to air-ratio. In the aforementioned researches, the latest and the most appropriate technique for AFR control is neural network-based MPC [11, 12], which is very robust and suitable for multivariable, time-varying, and delay systems, such as modern engine systems [13]. It is well known that a reliable engine performance prediction model is the core component of the MPC. However, the engine models developed in [10–12] were surrogate models. That is, the models were trained from the data generated by empirical equations rather than a real engine. In fact, many assumptions have been made in the empirical equations, and many coefficients in the equations are difficult to determine for a real engine [14]. Therefore, the neural network prediction models derived from the data generated by empirical equations cannot effectively reflect the actual performance of the controller in real engines. Moreover, the types of neural networks used in [11, 12] were radial-basis-function (RBF) neural network and diagonal recurrent neural network (DRNN). Both of these methods employ back-propagation (BP) algorithm that suffers from slow error convergence [15] and local minima [16]. Hence, the prediction models in these previous studies are not suitable for practical use.

To overcome the issues of BP, extreme learning machine (ELM) was proposed [17–20] that has a simple and effective modelling algorithm for training single hidden layer feedforward networks. In ELM, the training involves only the output weights; the parameters in the hidden nodes are generated randomly and unnecessary for adjustment. Many recent studies [16–18, 21–23] already showed that ELM can achieve better generalization and faster learning time than BP algorithms, and many variants of ELM have been introduced [15, 24–27]. However, classical ELM is only suitable for off-line batch training; it can only build the engine air-ratio model when all training data are ready. When a new datum arrives, the model cannot be updated dynamically, and a retraining from scratch is necessary. In practice, the air-ratio model in the MPC is required to be updated for any changes in engine performance due to engine aging or fair user modification on the engine. In other words, the training data for engine air-ratio performance will come sequentially in real time, so an online variant of ELM is necessary. Thankfully, online sequential extreme learning machine (OSELM) was developed in [15] for online learning of ELM. OSELM retains all the properties of ELM with advantages of efficient update and faster convergence speed. Moreover, its output weights can also be updated even when the data come one by one or chunk by chunk. This is a very favorable feature for real-time control applications. In order to develop a reliable prediction model for chaotic air-ratio behavior with online update ability, OSELM is therefore employed in this study.

In view of the deficiencies of the existing works, this paper proposes a nonlinear MPC algorithm for air-ratio control based on OSELM model, called online sequential extreme learning machine MPC (OEMPC). The OSELM air-ratio model can be updated continually to compensate for the air-ratio performance variation. This is a novel nontrivial application of OSELM. Based on the multiple-step-ahead prediction of the air-ratio, an optimal control signal is obtained to regulate the air-ratio to the desired values. The performance of the proposed OEMPC is compared with both the latest and conventional AFR control techniques, namely diagonal recurrent neural network MPC (DNMPC) [12] and proportional–integral–derivative (PID) controller in production cars, respectively. To the best knowledge of the authors, this paper is the first attempt at extending OSELM to the domain of automotive engine air-ratio modelling and control.

2 Online sequential extreme learning machine (OSELM)

OSELM is a simple and efficient online sequential modelling algorithm that can learn the training data one by one or chunk by chunk as well as discarding the data that are sequentially redundant. The OSELM modelling algorithm and update procedure are presented as below [15, 19]:

Given a chunk of initial training dataset $\mathbf{D}_0$ of N_0 input vectors $\boldsymbol{x}_n$, $n = 1$ to N_0, along with N_0 corresponding scalar-valued output y_n. The input vector $\boldsymbol{x}_n \in R^m$ contains the previous measured engine parameters, including engine speed (ES), throttle position (TP), and air-ratio at a previous time instant. The corresponding air-ratio at that time is defined as the output $y_n \in R$. The OSELM air-ratio model with L hidden nodes and activation function $g(\boldsymbol{x})$ can be mathematically expressed by:

$$\hat{y}_n = f(\boldsymbol{x}_n) = \sum_{i=1}^{L} \beta_i g(\boldsymbol{a}_i, b_i, \boldsymbol{x}_n), \tag{1}$$

where $\hat{y}_n = f(\boldsymbol{x}_n)$ is the prediction of the true air-ratio, $\boldsymbol{a}_i = [a_0,\ldots, a_L]$ and b_i are the centers and variances of the ith RBF hidden node because RBF kernel is chosen in this study, and β_i is the weight connecting the ith hidden node

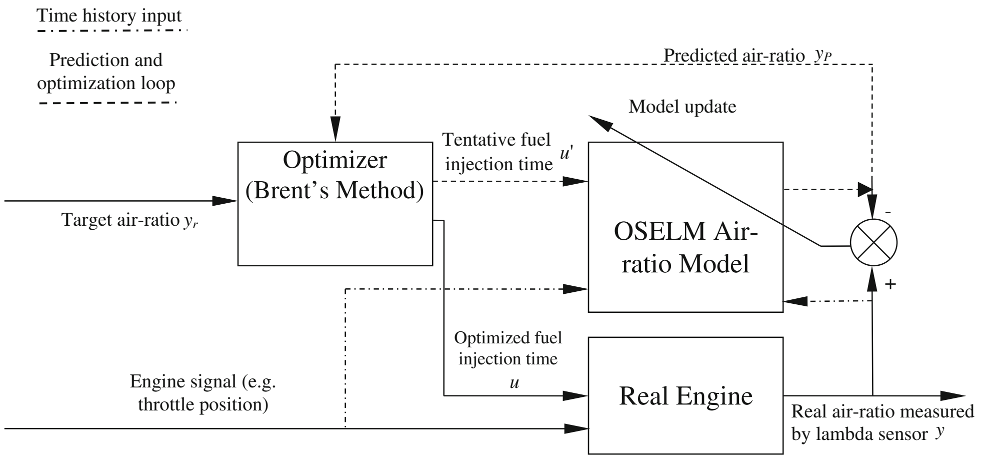

Time history input
Prediction and optimization loop
Predicted air-ratio y_P
Model update
Tentative fuel injection time u'
Optimizer (Brent's Method)
Target air-ratio y_r
OSELM Air-ratio Model
-
+
Optimized fuel injection time u
Engine signal (e.g. throttle position)
Real Engine
Real air-ratio measured by lambda sensor y

Fig. 1 Structure of OEMPC for engine air-ratio control

and the output node. Generally, the air-ratio model for predicting an unseen input datum $\boldsymbol{x}$, which contains the recent measured engine parameters, can be expressed as

$$\hat{y} = y_p = f(\boldsymbol{x}) = \sum_{i=1}^{L} \beta_i \exp\left(-\frac{\|\boldsymbol{a}_i - \boldsymbol{x}\|^2}{b_i^2}\right), \tag{2}$$

It has been theoretically shown that all weight vectors connecting the hidden nodes and the input nodes, $\boldsymbol{a}_i$ and b_i, can be randomly initialized and no update is needed. Only the weights connecting the hidden nodes and the output nodes, β_i, are required to be optimized with Eq. (3) [15, 19].

$$\boldsymbol{\beta}^{(0)} = \boldsymbol{K}_0^{-1}\boldsymbol{G}_0^T\boldsymbol{y}^{(0)}, \tag{3}$$

where $\boldsymbol{\beta}^{(0)} = [\beta_1, \beta_2,\ldots, \beta_L]$, $\boldsymbol{y}^{(0)} = [y_1, y_2,\ldots, y_{N0}]$, $\boldsymbol{G}_0$ is the hidden layer matrix that can be computed by Eq. (4), and $\boldsymbol{K}_0$ is defined by Eq. (5):

$$G_0 = \begin{bmatrix} \exp\left(-\frac{\|\boldsymbol{a}_1-\boldsymbol{x}_1\|^2}{b_i^2}\right) & \cdots & \exp\left(-\frac{\|\boldsymbol{a}_L-\boldsymbol{x}_1\|^2}{b_i^2}\right) \\ \vdots & \ddots & \vdots \\ \exp\left(-\frac{\|\boldsymbol{a}_1-\boldsymbol{x}_{N_0}\|^2}{b_i^2}\right) & \cdots & \exp\left(-\frac{\|\boldsymbol{a}_L-\boldsymbol{x}_{N_0}\|^2}{b_i^2}\right) \end{bmatrix}, \tag{4}$$

$$\boldsymbol{K}_0 = \boldsymbol{G}_0^T\boldsymbol{G}_0. \tag{5}$$

When there is a new chunk of dataset $\boldsymbol{D_1}$ of N_1 input vectors $\boldsymbol{x}_n$, $n = N_0 + 1$ to $N_0 + N_1$, along with N_1 corresponding scalar-valued output y_n, the air-ratio model is required to be updated to adapt the new scenario. As mentioned before, only the weights connecting the hidden nodes and the output nodes, β_i, are required to be updated, which is accomplished by the following equation [15, 19]:

$$\beta^{(1)} = \beta^{(0)} + \boldsymbol{K}_1^{-1}\boldsymbol{G}_1^T\left(y^{(1)} - \boldsymbol{G}_1\beta^{(0)}\right), \tag{6}$$

where $\boldsymbol{\beta}^{(1)} \in R^L$ are the updated weights, $\boldsymbol{y}^{(1)} = [y_{N0+1}, y_{N0+2},\ldots, y_{N0+N1}]$, $\boldsymbol{G}_1$ is another hidden layer matrix which can be computed by Eq. (7), and $\boldsymbol{K}_1$ is defined by Eq. (8):

$$\boldsymbol{G}_1 = \begin{bmatrix} \exp\left(-\frac{\|\boldsymbol{a}_1-\boldsymbol{x}_{N_0+1}\|^2}{b_i^2}\right) & \cdots & \exp\left(-\frac{\|\boldsymbol{a}_L-\boldsymbol{x}_{N_0+1}\|^2}{b_i^2}\right) \\ \vdots & \ddots & \vdots \\ \exp\left(-\frac{\|\boldsymbol{a}_1-\boldsymbol{x}_{N_0+N_1}\|^2}{b_i^2}\right) & \cdots & \exp\left(-\frac{\|\boldsymbol{a}_L-\boldsymbol{x}_{N_0+N_1}\|^2}{b_i^2}\right) \end{bmatrix}, \tag{7}$$

$$\boldsymbol{K}_1 = \boldsymbol{K}_0 + \boldsymbol{G}_1^T\boldsymbol{G}_1. \tag{8}$$

For the $k + 1$ chunk of dataset $\boldsymbol{D}_{k+1}$ of N_{k+1} input vectors $\boldsymbol{x}_n$, $n = N_k + 1$ to $N_k + N_{k+1}$, along with N_{k+1} corresponding scalar-valued output y_n, the optimal weights connecting the hidden node and the output nodes, β_i, are given by [15, 19]:

$$\boldsymbol{\beta}^{(k+1)} = \boldsymbol{\beta}^{(k)} + \boldsymbol{K}_{k+1}^{-1}\boldsymbol{G}_{k+1}^T\left(\boldsymbol{y}^{(k+1)} - \boldsymbol{G}_{k+1}\boldsymbol{\beta}^{(k)}\right), \tag{9}$$

where $\boldsymbol{y}^{(k+1)} = [y_{Nk+1}, y_{Nk+2},\ldots, y_{Nk+Nk+1}]$ and $\boldsymbol{K}_{k+1}$ is defined by Eq. (10):

$$\boldsymbol{K}_{k+1} = \boldsymbol{K}_k + \boldsymbol{G}_{k+1}^T\boldsymbol{G}_{k+1}. \tag{10}$$

In order to update $\boldsymbol{\beta}^{(k+1)}$ from $\boldsymbol{\beta}^{(k)}$ efficiently, $\boldsymbol{K}_{k+1}^{-1}$ is required to be computed efficiently, which could be derived using the Woodbury formula [28]:

$$\begin{aligned} \boldsymbol{K}_{k+1}^{-1} &= \left(\boldsymbol{K}_k + \boldsymbol{G}_{k+1}^T\boldsymbol{G}_{k+1}\right)^{-1}, \\ &= \boldsymbol{K}_k^{-1} - \boldsymbol{K}_k^{-1}\boldsymbol{G}_{k+1}^T\left(\boldsymbol{I} + \boldsymbol{G}_{k+1}\boldsymbol{K}_k^{-1}\boldsymbol{G}_{k+1}^T\right)^{-1} \times \boldsymbol{G}_{k+1}\boldsymbol{K}_k^{-1}. \end{aligned} \tag{11}$$

To avoid the term $\boldsymbol{G}_0^T\boldsymbol{G}_0$ in Eq. (5) being singular, $rank(\boldsymbol{G}_0)$ must be equal to L. In order to make $rank(\boldsymbol{G}_0) = L$, the OSELM must be initialized with a training dataset of N_0 data such that $N_0 > L$. The number of hidden nodes L in this study was chosen as 20 based on the model selection procedure presented in [15].

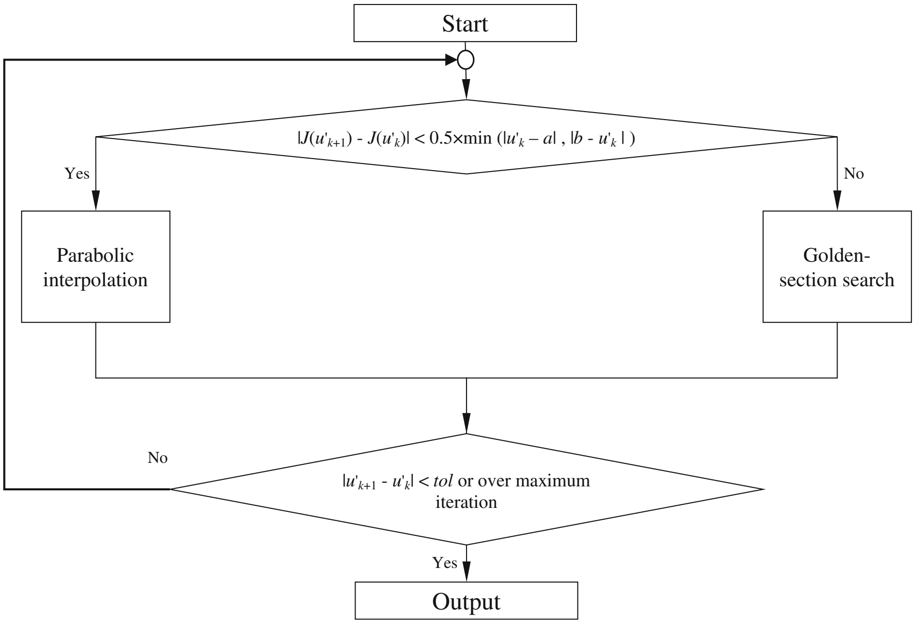

Start
|J(u'k+1) - J(u'k)| < 0.5×min (|u'k – a| , |b - u'k |)
Yes
No
Parabolic interpolation
Golden-section search
No
|u'k+1 - u'k| < tol or over maximum iteration
Yes
Output

Fig. 2 General working principle of Brent's method

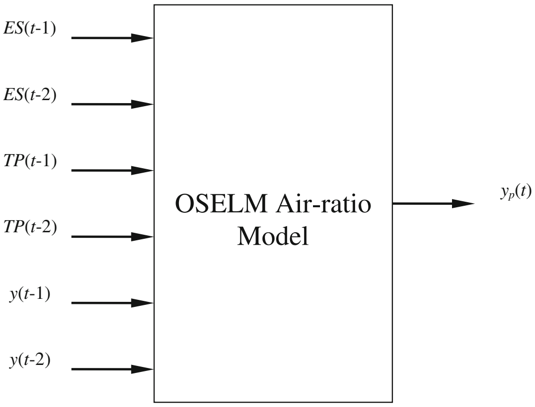


Fig. 3 Structure of OSELM air-ratio model

3 MPC with OSELM model

The structure of the proposed OEMPC is shown in Fig. 1. The controller consists of the OSELM air-ratio model and the optimizer based on Brent's method [29]. The OSELM air-ratio model predicts the engine response over a specified time horizon. The predictions are used by the optimizer to determine the tentative fuel injection time u', which minimizes the following performance criterion over the specified time horizon, and then the optimal fuel injection time signal u is sent to the engine.

$$\min J(u') = \sum_{j=N_1}^{N_2} \left(y_r(t+j) - y_p(t+j)\right)^2 + \rho \sum_{j=1}^{N_u} \left(u'(t+j-1) - u'(t+j-2)\right)^2, \quad (12)$$

where N_1 and N_2 define the prediction horizon. t and N_u are the time step and the control horizon, respectively. ρ is a user-defined weight that penalizes excessive movement of the control signal (i.e., the fuel injection time). The variables $u'(t+j-1)$ and $u'(t+j-2)$ in the second part of Eq. (12) are the tentative fuel injection time at the time steps $t+j-1$ and $t+j-2$, respectively. The second part of Eq. (12) ensures the stability of the controller output. $y_r(t+j)$ is the target air-ratio at the time step $t+j$, and y_p is the predicted air-ratio by the OSELM model which is computed from Eq. (2) at the time step $t+j$, from which the input vector $\boldsymbol{x}$ consists of three time series of engine parameters: engine speed ES$(t+j-1)$, ES$(t+j-2)$, etc.; throttle position TP$(t+j-1)$, TP$(t+j-2)$, etc; and previous measured air-ratio $y(t+j-1)$, $y(t+j-2)$, etc. In other words,

$$y_p(t+j) = \sum_{i=1}^{L} \beta_i \exp\left(-\frac{\|\boldsymbol{a}_1 - \boldsymbol{x}\|^2}{b_i^2}\right). \quad (13)$$

3.1 Single-dimensional optimization approach

The original optimization problem involved in this paper is multidimensional and constrained with the tentative control signals, $u'(t)$, $u'(t+1)$,…, $u'(t+N_u-1)$, over the control horizon N_u, which can minimize the objective function $J(u')$ of Eq. (12). Then the predicted air-ratios, $y_p(t+N_1)$, $y_p(t+N_1+1)$,…, $y_p(t+N_2)$, can trace the target air-ratios, $y_r(t+N_1)$, $y_r(t+N_1+1)$,…, $y_r(t+N_2)$, by using the optimized fuel injection time series. Each fuel injection time is normally bounded within the range from 2 to 15 ms. However, the multidimensional optimization

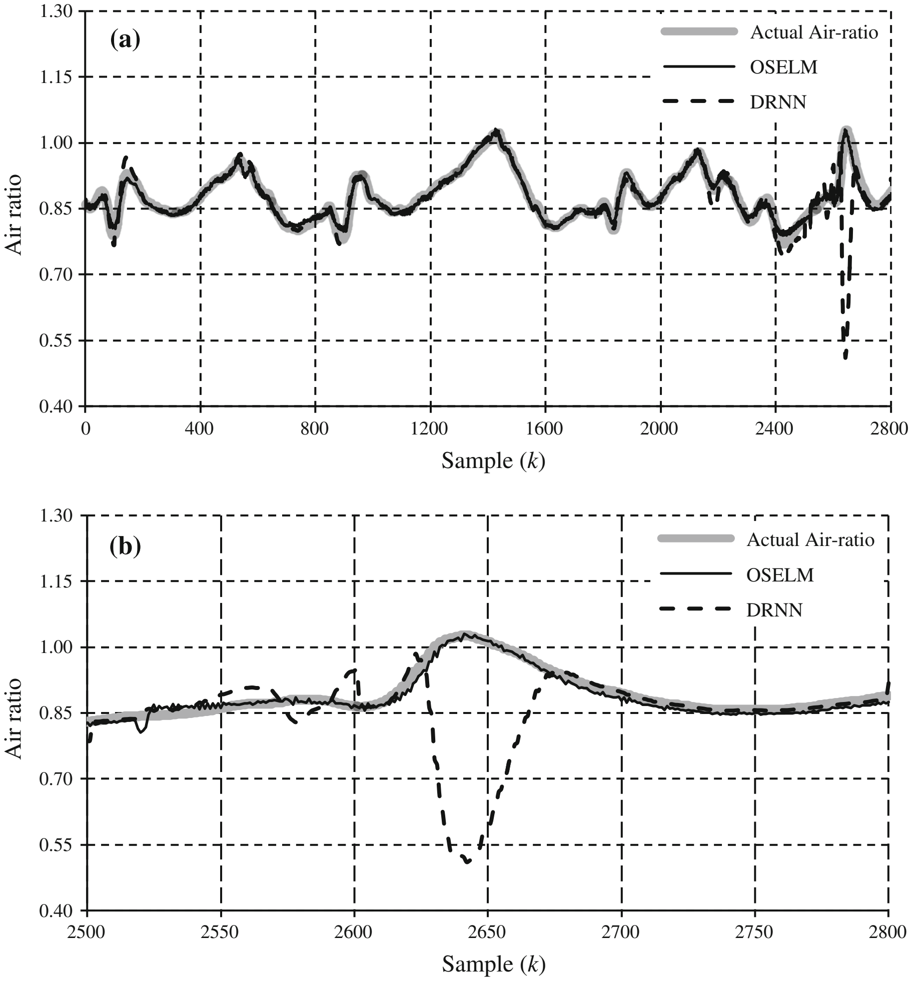


Fig. 4 **a** Comparison between predicted air-ratios and the corresponding actual air-ratios; **b** in detail

always requires heavy computation, especially when constraints exist. Real-time control applications often put emphasis on computational speed. The research of [11] showed that the one-dimensional approach is efficient for real-time AFR control and that the overall tracking error is similar to that using multidimensional optimization approach. Therefore, the optimization problem to be solved is reduced to one-dimensional. In this paper, the control signal u is assumed to remain constant over the control horizon. Therefore, the tentative control signal in the objective function is also constant over the control horizon, i.e., $u'(t) = u'(t + 1),\ldots, = u'(t + N_u - 1)$. In this way, only one parameter $u'(t)$ is needed to be determined, and the final fuel injection time at each time step u is set to be the optimal value of $u'(t)$.

3.2 Brent's method

There are many optimization techniques available for MPC, and each technique has its pros and cons. A well-known technique—Brent's method—was selected as the MPC optimizer in this study for illustrative purpose. Brent's method is a robust and efficient optimization method. It combines the typical parabolic interpolation and golden-section search. The objective function in each iteration is approximated by interpolating a parabola through three existing points. The minimum point of the parabola is taken as a guess for the minimum point if certain criteria are met. Otherwise, golden-section search is carried out. The advantage of this method is that the high convergence rate of parabolic interpolation can be maintained without losing the robustness of golden-section search [29]. The general working principle of Brent's method is shown in Fig. 2. The detailed optimization procedure of Brent's method was presented in [29] and is not presented herein. There are three parameters of Brent's method, which are the initial interval of the input variable, [a, b], that is the limit of the fuel injection time, the tolerance, tol, and the maximum iteration for stopping the optimization procedure. The three variables were set at [2, 15], 0.05, and 50, respectively, because the fuel injection time varies within 2–15 ms from 0 to 100 % throttle.

Table 1 Accuracy and computation time of different air-ratio models

Air-ratio model	LMAE	Model training time (s)	Average model updating time (s)	Cumulative model updating time (s)
EM*($\boldsymbol{x}$)	2.2424	0.1783	0.0060	0.1685
NN*($\boldsymbol{x}$)	1.8323	4.5609	0.1689	6.0339

4 Implementation and evaluation of OSELM air-ratio model

4.1 OSELM air-ratio model implementation

The objective of the OSELM air-ratio model is to predict the future air-ratio y_p from the inputs of three time series of engine parameters: ES, TP, and previous measured air-ratio y. The structure of the OSELM model was chosen to be second order (i.e., 2 past time steps), which gives the minimum prediction error [11, 12], and the structure is shown in Fig. 3.

To obtain the engine data for building the air-ratio model, 5,800 data samples including air-ratio, ES, and TPs were collected. These 5,800 data samples were collected over a Honda DC5 Type-R test car with K20A i-VTEC engine connected to a MoTeC M800 programmable ECU with nonfactory calibration data. A dyno test was done for collecting the data samples. The first 3,000 data samples were used as initial training dataset $\boldsymbol{D}_0$ to build the air-ratio model. The last 2,800 data samples were used as test dataset **TEST**. The first 2,700 data samples of **TEST** were also regarded as update dataset **UPDATE** for real-time updates. The air-ratio model was updated every 100 measured air-ratio data samples during prediction. In other words, **UPDATE** was divided into 27 datasets, $\boldsymbol{D}_k$, for $k = 1$–27. Each dataset $\boldsymbol{D}_k$ was used as test data first to test the immediate model accuracy, and then $\boldsymbol{D}_k$ was used again as update data for updating the model. The remaining 100 data samples of **TEST** were the last dataset for evaluating the accuracy of the air-ratio model after finishing the last update (i.e., updated by 2,601–2,700 data samples) in the test. With respect to the training dataset $\boldsymbol{D}_0$, the test dataset **TEST** was the unseen case for testing the generalization of the built air-ratio model.

4.2 Evaluation of engine air-ratio model

To illustrate the accuracy, superiority, and online update ability of the proposed OSELM air-ratio model, its prediction result was compared with those obtained from the latest modelling algorithm for AFR control, DRNN [12]. For a fair comparison between the two modelling algorithms, both modelling algorithms were implemented using MATLAB 2011b and executed on a 3.4 GHz Intel Core i7 PC with eight GB RAM onboard.

After obtaining the air-ratio model EM($\boldsymbol{x}$) with OSELM through the modelling algorithm presented in Sect. 2 over $\boldsymbol{D}_0$,

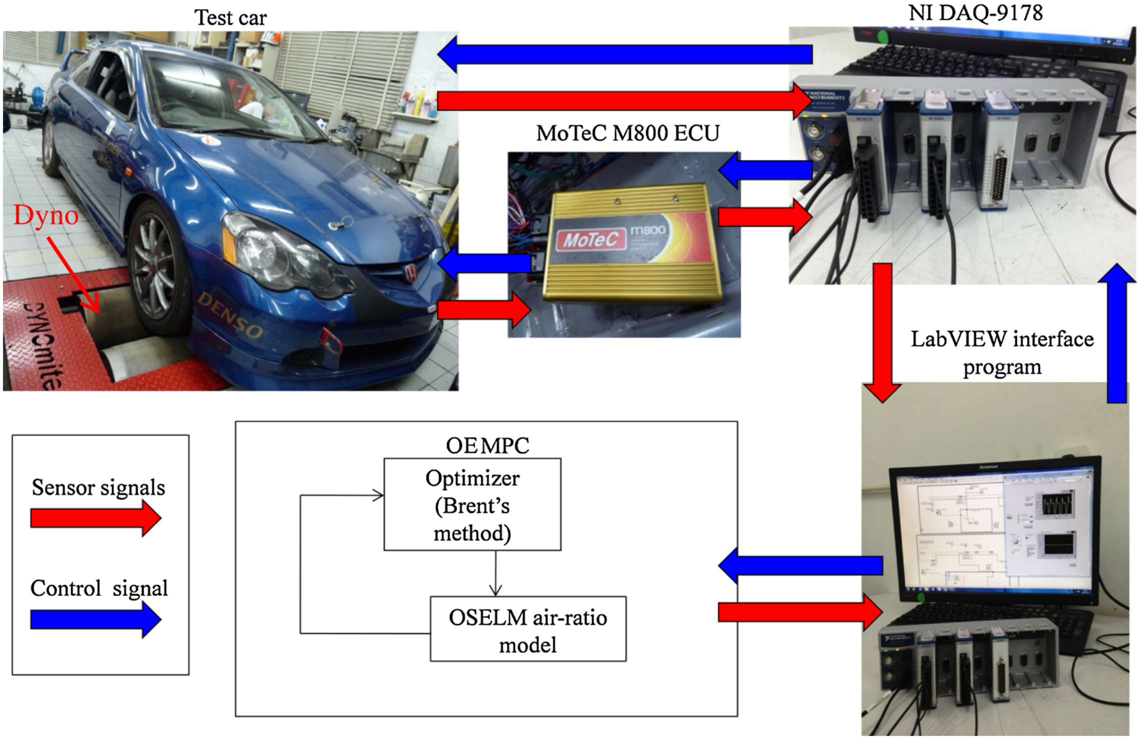


Fig. 5 Experimental setup and signal flow between test car and OEMPC

Fig. 6 Test cycle: Throttle position versus time

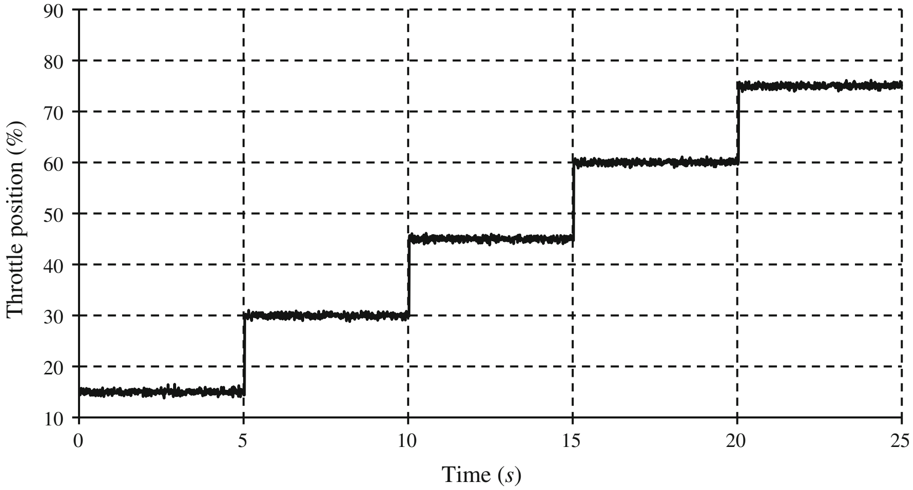
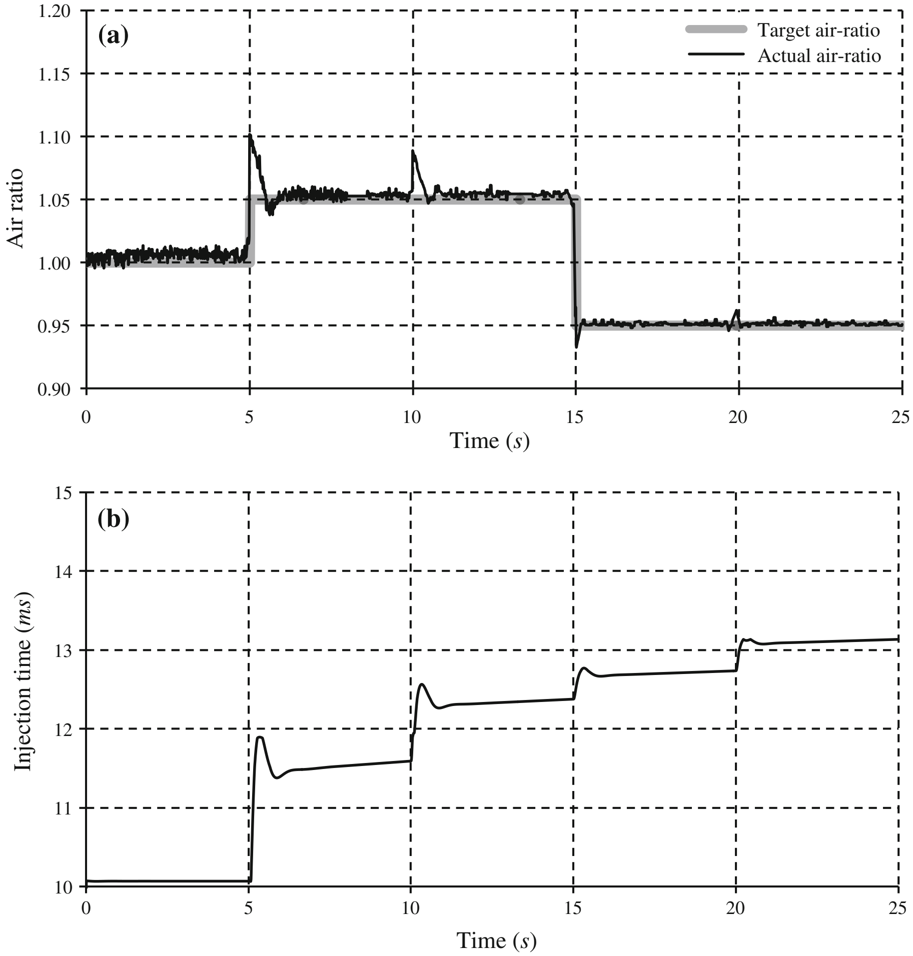


Fig. 7 **a** Air-ratio control results and **b** corresponding fuel injection time of OEMPC in pilot test 1



EM($\boldsymbol{x}$) was then updated for 27 times with $\boldsymbol{D}_k$, for $k = 1$–27, 100 data samples each time, to build the updated OSELM air-ratio model EM*($\boldsymbol{x}$). Apart from the OSELM model, the DRNN modelling algorithm for model predictive AFR control presented in [12] was applied to build the air-ratio model, NN($\boldsymbol{x}$). The network was trained for 100 epochs. After that, NN($\boldsymbol{x}$) was updated with **UPDATE** but employing the dynamic BP with automatic differentiation technique presented in [12] to build the updated DRNN air-ratio model NN*($\boldsymbol{x}$).

After constructing EM*($\boldsymbol{x}$) and NN*($\boldsymbol{x}$), the performance of the two air-ratio models can be evaluated in terms of accuracy. Since the range of air-ratio for

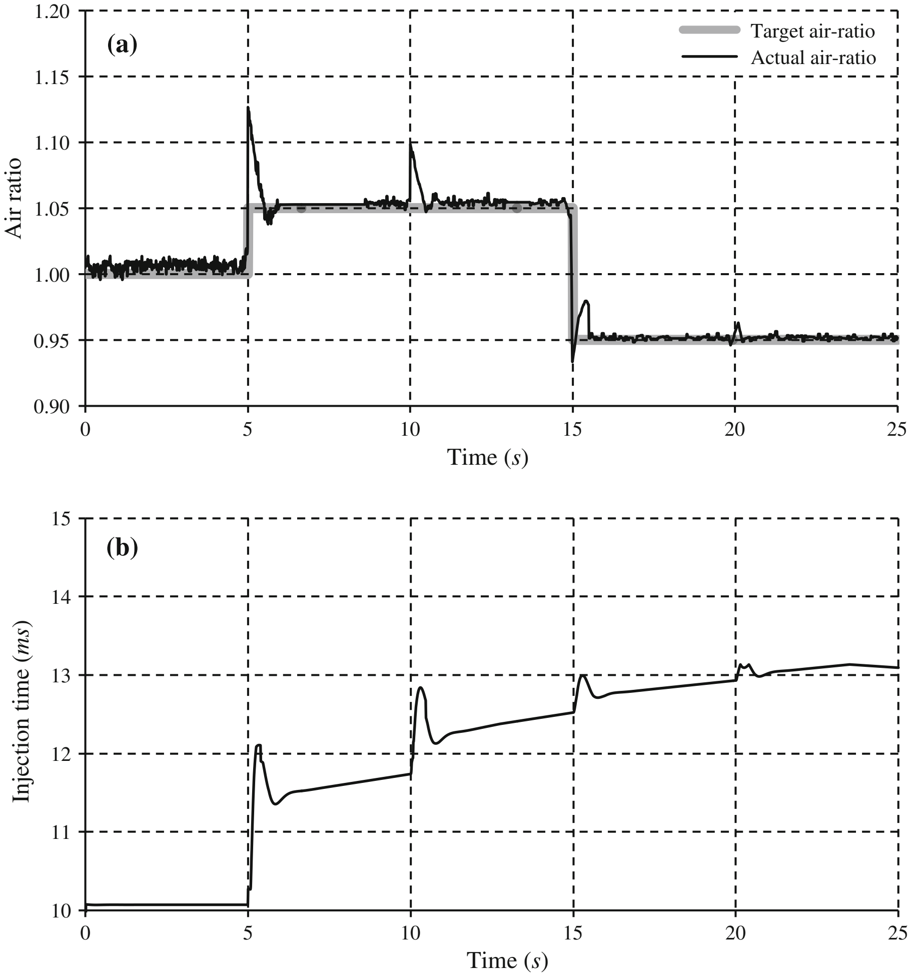


Fig. 8 **a** Air-ratio control results and **b** corresponding fuel injection time of DNMPC in pilot test 1

combustible mixture is very narrow, the prediction errors of the above two air-ratio models are presented by logarithmic mean absolute error (LMAE), and they were evaluated one by one against the test dataset, **TEST**, using Eq. (14):

$$\mathrm{LMAE} = -\log\left[\frac{1}{T}\sum_{k=1}^{T}\left|y_k - f*(\boldsymbol{x}_k)\right|\right], \quad (14)$$

where $f^*(\boldsymbol{x}_k)$ represents either EM*$(\boldsymbol{x}_k)$ or NN*$(\boldsymbol{x}_k)$, $\boldsymbol{x}_k$ is the kth new input vector for air-ratio prediction, y_k is the corresponding actual air-ratio of $f^*(\boldsymbol{x}_k)$, and T is the total number of predictions. The value of T is equal to 2,800 in this case study. The larger the LMAE, the higher the model accuracy is. The prediction results between the predicted air-ratios and the corresponding actual air-ratios over **TEST** are shown in Fig. 4.

Apart from the model accuracy, computation speed is also an important factor in real-time control applications. Table 1 shows the LMAE, model training time, and model updating time of both OSELM and DRNN air-ratio models. According to the LMAE values in Table 1, the OSELM air-ratio model EM*$(\boldsymbol{x})$ outperforms NN*$(\boldsymbol{x})$ by approximately 22 %. The trends and tracking errors in Fig. 4 also imply that the accuracy of the OSELM air-ratio model will be well maintained if the model update is continually carried out to adapt to any changes in engine performance due to engine aging or fair user modification on the engine. In addition to model accuracy, OSELM can also reduce the time for training and updating the air-ratio model significantly. In Table 1, the air-ratio model training time, average model updating time, and cumulative model updating time of EM*$(\boldsymbol{x})$ are 0.1783, 0.0060, and 0.1685 s, respectively. The average model updating time of EM*$(\boldsymbol{x})$, 0.0060 s, is only about 128 of NN*$(\boldsymbol{x})$. This achievement is accomplished by the effective online learning algorithm of OSELM. The total cumulative model updating time saved by OSELM from DRNN after 27 update iterations is 5.8654 s. However, if the air-ratio model is often updated, the cumulative model updating time saved by OSELM can be very significant. Obviously, OSELM is superior to DRNN algorithm. As a whole, the high accuracy and short updating time of the air-ratio model using OSELM make online model predictive air-ratio control more feasible.

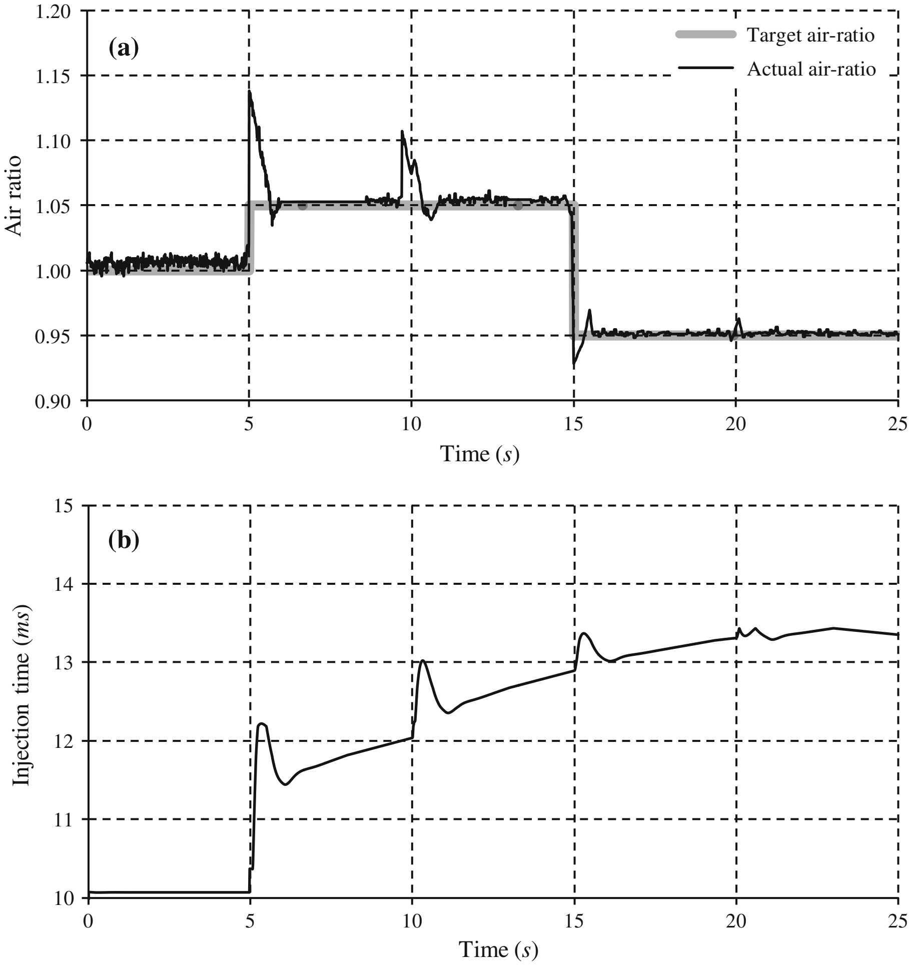


Fig. 9 **a** Air-ratio control results and **b** corresponding fuel injection time of PID controller in pilot test 1

Table 2 Control performance of pilot test 1 of different controllers

Controller	TI	Max. overshoot
OEMPC	2.3309	0.1074
(% of improvement)[a]	(1.8 %)	(15.23 %)
(% of improvement)[b]	(4.18 %)	(22.23 %)
DNMPC	2.2896	0.1267
(% of improvement)[c]	(2.3 %)	(8.25 %)
PID	2.2373	0.1381

[a] OEMPC over DNMPC, [b] OEMPC over PID, [c] DNMPC over PID

Hence, OSELM was confidently selected to implement the model predictive engine air-ratio controller in this research.

5 Implementation and evaluation of OEMPC

5.1 Experimental setup

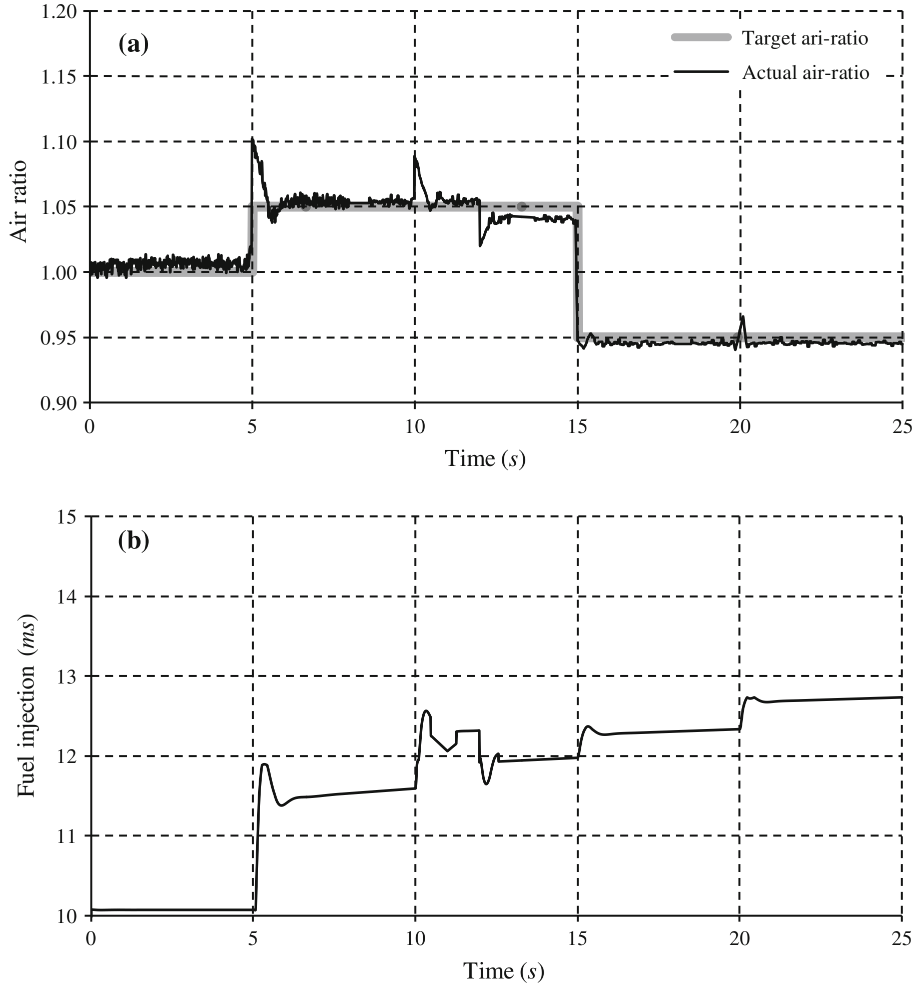
The proposed OEMPC algorithm was implemented and tested on a real car, Honda DC5 Type-R with K20A i-VTEC engine connected to a MoTeC M800 programmable electronic control unit (ECU) and National Instrument (NI) CompactDAQ chassis DAQ-9178. The MPC algorithm was implemented using MATLAB. MoTeC M800 is mainly used for engine control, whereas NI DAQ-9178 is used for sending control signal to the MoTeC ECU via a LabVIEW interface program according to the MATLAB MPC program embedded. In other words, NI DAQ-9178 serves as an interface between the MATLAB program and the MoTeC ECU. Apart from fuel injector control, the MoTeC ECU also contains many control maps, such as ignition map and valve timing map, to maintain the engine operation. The experimental setup and the signal flow between the test car and OEMPC are shown in Fig. 5.

The initial off-line training data for building the OSELM model were obtained using a wideband lambda sensor subject to random TPs and are discussed in Sect. 4. In this study, there are two pilot tests evaluating the tracking error and online update ability of the controllers.

5.2 Pilot test 1: tracking ability

In pilot test 1, the test cycle is shown in Fig. 6 where the TP gradually changes from 15 to 75 % throttle (the TP

Fig. 10 **a** Air-ratio control results and **b** corresponding fuel injection time of OEMPC in pilot test 2



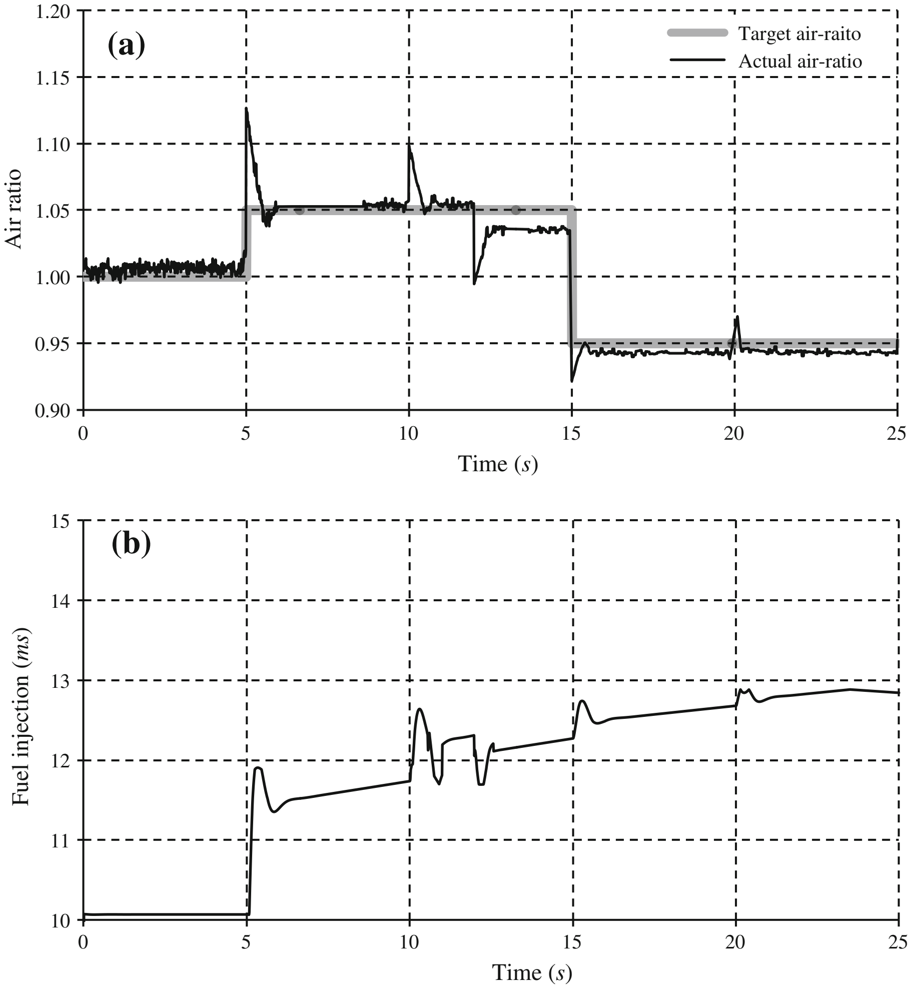
increases by 15 % every 5 s). Such test cycle is designed by referring to [12], which almost covers the whole operating condition. In this test, the air-ratio is required to track the target air-ratios from the stoichiometric value (1.00) for minimum emissions to a value for the best brake-specific fuel consumption (1.05) and then to a value for maximum engine power (0.95) as the TP is gradually changed from 15 to 75 % throttle. Such tracking of air-ratio changes is essential for automobiles to satisfy the emission, fuel consumption, and power requirements under different operating conditions [7, 8]. After choosing the sampling time to be 0.01 s, the tracking ability of the OEMPC can be examined. By testing many values around the parameters used in [11, 12], the parameters of the optimizer were chosen as $N_1 = 1, N_2 = 8, \rho = 0.75$, and $N_u = 5$. With the test cycle shown in Fig. 6 and the parameters chosen, the air-ratio control result and the corresponding fuel injection time of the OEMPC are shown in Fig. 7. To show the advantage of the OEMPC, its control result is compared with the latest model predictive controller, DRNN model predictive controller (DNMPC), as well as the PID controller used in the existing automotive ECU. The air-ratio model used in the DNMPC is the model mentioned in Sect. 4.2. The parameters of nonlinear optimization for the DNMPC were the same as those of OEMPC. The PID gains of the PID controller were obtained by the Ziegler–Nichols method. For a fair comparison, all the control algorithms were implemented using MATLAB 2011b and executed on a 3.4 GHz Intel Core i7 PC with eight GB RAM onboard. The air-ratio control results and the corresponding fuel injection time of the DNMPC and PID controller are shown in Figs. 8 and 9, respectively.

Figure 7a shows that the OEMPC can regulate the air-ratio to follow the target air-ratios with the smallest deviation among all the controllers. As the range of air-ratio for combustible mixture is very narrow, logarithmic mean absolute error is chosen as the tracking index (TI) to evaluate the tracking ability of the controllers. The index is defined by

$$\mathrm{TI} = -\log\left[\frac{1}{T_s}\sum_{t=1}^{T_s}|y_t - y_r(t)|\right], \tag{15}$$

where t is time step, T_s is the total number of time step in the test, y_t is the actual air-ratio at each time step, and $y_r(t)$ is the

Fig. 11 **a** Air-ratio control results and **b** corresponding fuel injection time of DNMPC in pilot test 2



corresponding target air-ratio at each time step. The target air-ratio varies according to the aforesaid practical operating conditions. In other words, when the TP is near 15 %, the target air-ratio is set to stoichiometry (1.00) to minimize the emissions, whereas the target air-ratio is set to a value of 1.05 to minimize the fuel consumption during partial throttle as the car is cursing. On the other hand, the target air-ratio is set to a value of 0.95 to maximize the engine power for 75 % throttle or above [7, 8].

The control performances of the three controllers are shown in Table 2, and the OEMPC outperforms the DNMPC and PID controller by approximately 2 and 4 %, respectively, in terms of *TI*. These results show the excellent tracking ability of the OEMPC. Apart from the superiority of *TI*, the maximum overshoot during transient state of the OEMPC also outperforms the DNMPC and PID controller by approximately 15 and 22 %, respectively.

5.3 Pilot test 2: online update ability

In the previous test, the “bypass air valve” of the test engine is 60 % opening. In order to test the update ability of the OEMPC, the “bypass air valve” was set from 60 to 30 % from 12 s to the end of the test. It is equivalent to the clogging of the engine intake filter as the engine is aging. Normally, the air-ratio must decrease under the same TP as lack of intake air. The testing procedure and the parameter setting are exactly the same as those in pilot test 1 except the sudden change of the “bypass air valve” position after 12 s. The air-ratio control result and the fuel injection time of the OEMPC are shown in Fig. 10. Similar to pilot test 1, the control result is compared with the DNMPC and typical PID controller. The control results and the corresponding fuel injection time of the DNMPC and PID controller are shown in Figs. 11 and 12, respectively. Figures 10a, 11a, and 12a depict that the air-ratios decrease after changing the position of the “bypass air valve” due to fuel rich. Figures 10a and 11a illustrate that the two controllers, OEMPC and DNMPC, can regulate air-ratios with obviously less deviation from the target air-ratios and overshoot than that of the PID controller because the air-ratio model can be self-updated for any changes in engine condition. The control performances of the three controllers are shown in Table 3, and the control performance of *TI* of the

Fig. 12 **a** Air-ratio control results and **b** corresponding fuel injection time of PID controller in pilot test 2

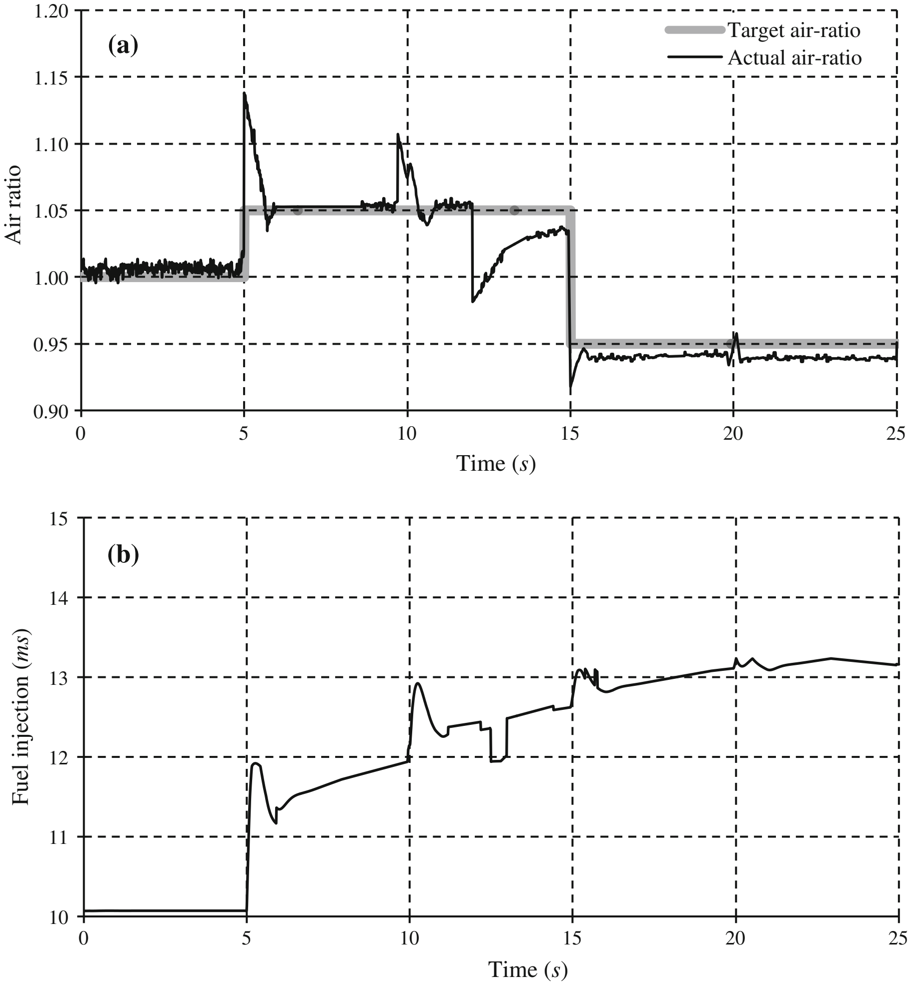


Table 3 Control performance of pilot test 2 of different controllers

Controller	TI	Max. overshoot
OEMPC	2.1948	0.1173
(% of improvement)[a]	(5.75 %)	(8.5 %)
(% of improvement)[b]	(14.02 %)	(15.06 %)
DNMPC	2.0754	0.1282
(% of improvement)[c]	(7.81 %)	(7.17 %)
PID	1.9250	0.1381

[a] OEMPC over DNMPC, [b] OEMPC over PID, [c] DNMPC over PID

OEMPC outperforms the DNMPC and PID controller by approximately 6 and 14 %, respectively. Moreover, the maximum overshoot during transient state of the OEMPC outperforms the DNMPC and PID controller by approximately 9 and 15 %, respectively. This promising result indicates that the OEMPC can regulate air-ratio very well even though the engine ages and undergoes external disturbance simultaneously.

5.4 Discussion of results

All experimental results show that the overall air-ratio control performance of the OEMPC is better than those of DNMPC and conventional PID controller. There are three important factors affecting the control performance of the model predictive controllers: air-ratio model accuracy, computational time, and adaptability. As presented in Table 1, the model accuracy and model updating time of the OSELM are better than those of DRNN. Moreover, the tracking errors from 12 to 25 s in Figs. 10a and 11a reveal that the model accuracy of the OSELM after update is obviously better than that of DRNN. This result shows that the online sequential learning algorithm of the OSELM is superior to the dynamic BP with automatic differentiation technique in DRNN. As a result, the overall control performance of the OSELM is better than the DNMPC. Furthermore, the PID controller cannot adapt to the changes of engine conditions, and no optimal PID gains can handle all situations. Therefore, PID controller is worse than both OEMPC and DNMPC in performance. In summary, the tracking and online update abilities of the OEMPC are the best among the three controllers. Hence, the OEMPC is the most suitable method for engine air-ratio control. Although Tables 2 and 3 show that the *TI* of the OEMPC has only been slightly improved, this small change in *TI* can practically result in a big change of engine performance because the air-ratio is a very delicate value. References [7, 8] stated clearly that if the air-ratio deviates 2 % from its stoichiometric ratio, CO, HC, and NO_X emissions are significantly increased due to the degradation of conversion efficiency of the three-way catalytic converter. As a result, the actual improvement achieved by the OEMPC is very significant to this application.

6 Conclusions

This research is the first attempt at developing OEMPC for engine air-ratio control. The OEMPC for engine air-ratio control is trained and updated sequentially by real-time engine data so as to maintain the air-ratio model accuracy for any changes in engine performance, such as engine aging or fair user modification. The proposed OEMPC was successfully implemented and tested on a real automotive engine, whereas many previous researches were simulation tests only. Experimental results show that the air-ratio control performance of the OEMPC is significantly better than those of DNMPC and conventional PID controller. Tables 2 and 3 reveal that the OEMPC can effectively reduce the air-ratio deviation and maximum overshoot up to 5.75 and 15.23 %, and 14.02 and 22.23 % than those of DNMPC and conventional PID controller, respectively. Therefore, the OEMPC is a promising control scheme to replace conventional PID controller in the automotive ECU for engine air-ratio control. In the future, advanced optimization algorithms for the OEMPC and system stability will be studied.

Acknowledgments The research was supported by the University of Macau Research Grant, Grant numbers: MYRG149(Y1-L2)-FST11-WPK and MYRG075(Y1-L2)-FST13-VCM. The research was also supported by FDCT Macau Grant: FDCT/075/2013/A. The authors would also like to thank the support from Mr. Ka In Wong.

References

1. Crouse WH, Anglin DL (1993) Automotive mechanics, 10th edn. McGraw-Hill, New York
2. Heinrich J, Schwarze PE, Stilianakis N, Momas I, Medina S, Totlandsdal AI, Bree LV, Kuna-Dibbert B, Krzyzanowski M (2005) Studies on health effects of transport-related air pollution. In: Krzyzanowski M, Kuna-Dibbert B, Schneider J (eds) health effects of transport-related air pollution. WHO Press, Geneva, pp 125–183
3. Campbell M, Bassil K, Morgan C, Lalani M, Macfarlane R, Bienefeld M (2007) Air pollution burden of illness from traffic in Toronto: problems and solutions. Toronto Public Health, Toronto
4. Millman A, Tang D, Perera FP (2008) Air pollution threatens the health of children in China. Pediatrics 122:620–628
5. Ji S, Cherry CR, Bechle MJ, Wu Y, Marshall JD (2012) Electric vehicles in China: emissions and health impacts. Environ Sci Technol 46:2018–2024
6. Yim SH, Barrett SR (2012) Public health impacts of combustion emissions in the United Kingdom. Environ Sci Technol 46:4291–4296
7. Gilles T (2011) Automotive service: inspection, maintenance, repair, 4th edn. Cengage Learning, Stamford
8. Żmudka Z, Postrzednik S (2011) Inverse aspects of the three-way catalytic converter operation in the spark ignition engine. J KONES Powertrain Transp 18:509–516
9. Wang SW, Yu DL, Gomm JB, Page GF, Douglas SS (2006) Adaptive neural network model based predictive control for air–fuel ratio of SI engines. Eng Appl Artif Intell 19:189–200
10. Zhai YJ, Yu DL (2008) Radial-basis-function-based feedforward-feedback control for air–fuel ratio of spark ignition engines. Proc Inst Mech Eng D J Automob Eng 222:415–428
11. Zhai YJ, Yu DL (2009) Neural network model-based automotive engine air/fuel ratio control and robustness evaluation. Eng Appl Artif Intell 22:171–180
12. Zhai YJ, Yu DW, Guo HY, Yu DL (2010) Robust air/fuel ratio control with adaptive DRNN model and AD tuning. Eng Appl Artif Intell 23:283–289
13. Li GY (2007) Application on intelligent control and MATLAB to electronically controlled engines, 1st edn. Publishing House of Electronics Industry, Beijing (in Chinese)
14. Wong PK, Tam LM, Li K, Vong CM (2010) Engine idle-speed system modelling and control optimization using artificial intelligence. Proc Inst Mech Eng D J Automob Eng 224:55–72
15. Liang NY, Huang GB, Saratchandran P, Sundararajan N (2006) A fast and accurate online sequential learning algorithm for feedforward networks. IEEE Trans Neural Netw 17:1411–1423
16. Wong KI, Wong PK, Cheung CS, Vong CM (2013) Modelling of diesel engine performance using advanced machine learning methods under scarce and exponential data set. Appl Soft Comput 13:4428–4441
17. Huang GB, Zhu QY, Siew CK (2004) Extreme learning machine: a new learning scheme of feedforward neural networks. In: Proceedings of International Joint Conference on Neural Networks (IJCNN2004), Budapest, pp 985–990
18. Huang GB, Zhu QY, Siew CK (2006) Extreme learning machine: theory and applications. Neurocomputing 70:489–501
19. Huang GB, Wang DH, Lan Y (2011) Extreme learning machines: a survey. Int J Mach Learn Cybernet 2:107–122
20. Huang GB, Zhou HM, Ding XJ, Zhang R (2012) Extreme learning machine for regression and multiclass classification. IEEE Trans Syst Man Cybern B Cybern 42:513–529
21. Chacko B, Krishnan VRV, Raju G, Anto PB (2012) Handwritten character recognition using wavelet energy and extreme learning machine. Int J Mach Learn Cybernet 3:149–161
22. Bueno-Crespo A, García-Laencina PJ, Sancho-Gómez J-L (2013) Neural architecture design based on extreme learning machine. Neural Netw 48:19–24
23. Wang X, Shao Q, Miao Q, Zhai J (2013) Architecture selection for networks trained with extreme learning machine using localized generalization error model. Neurocomputing 102:3–9
24. Miche Y, van Heeswijk M, Bas P, Simula O, Lendasse A (2011) TROP-ELM: a double-regularized ELM using LARS and Tikhonov regularization. Neurocomputing 74:2413–2421
25. Zhai J, Xu H, Wang X (2012) Dynamic ensemble extreme learning machine based on sample entropy. Soft Comput 16:1493–1502
26. Zhai J, Xu H, Li Y (2013) Fusion of extreme learning machine with fuzzy integral. Int J Uncertain Fuzziness Knowl Based Syst 21(Suppl. 2):23–34

27. Luo J, Vong CM, Wong PK (2013) Sparse Bayesian extreme learning machine for multi-classification. IEEE Trans Neural Netw Learn Syst. doi:10.1109/TNNLS.2013.2281839
28. Golub GH, Loan CFV (1996) Matrix computation, 3rd edn. Johns Hopkins University Press, Baltimore
29. Chandrupatla TR (1998) An efficient quadratic fit—sectioning algorithm for minimization without derivatives. Comput Methods Appl Mech Eng 152:211–217