

Research Article

Model Predictive Engine Air-Ratio Control Using Online Sequential Relevance Vector Machine

Hang-cheong Wong,¹ Pak-kin Wong,¹ and Chi-man Vong²

¹ Department of Electromechanical Engineering, Faculty of Science and Technology, University of Macau, Taipa 999078, Macau

² Department of Computer and Information Science, Faculty of Science and Technology, University of Macau, Taipa 999078, Macau

Correspondence should be addressed to Hang-cheong Wong, hcwong@umac.mo

Received 17 January 2012; Accepted 24 March 2012

Academic Editor: Qingsong Xu

Copyright © 2012 Hang-cheong Wong et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Engine power, brake-specific fuel consumption, and emissions relate closely to air ratio (i.e., lambda) among all the engine variables. An accurate and adaptive model for lambda prediction is essential to effective lambda control for long term. This paper utilizes an emerging technique, relevance vector machine (RVM), to build a reliable time-dependent lambda model which can be continually updated whenever a sample is added to, or removed from, the estimated lambda model. The paper also presents a new model predictive control (MPC) algorithm for air-ratio regulation based on RVM. This study shows that the accuracy, training, and updating time of the RVM model are superior to the latest modelling methods, such as diagonal recurrent neural network (DRNN) and decremental least-squares support vector machine (DLSSVM). Moreover, the control algorithm has been implemented on a real car to test. Experimental results reveal that the control performance of the proposed relevance vector machine model predictive controller (RVMMPC) is also superior to DRNNMPC, support vector machine-based MPC, and conventional proportional-integral (PI) controller in production cars. Therefore, the proposed RVMMPC is a promising scheme to replace conventional PI controller for engine air-ratio control.

1. Introduction

Engine power, brake-specific fuel consumption, and emissions relate closely to air ratio among all the engine variables [1]. Air ratio is also called lambda. It indicates the amount that the actual available air-fuel ratio mixture differs from the stoichiometric air-fuel ratio of the fuel being used. Manzie et al. [2, 3] mentioned that if the air-fuel ratio is 1% lower than its stoichiometric ratio (e.g., 14.7:1 for gasoline), carbon monoxide (CO) and hydrocarbon (HC) emissions will be significantly increased. An air-fuel ratio that is 1% higher than the stoichiometric ratio produces more nitrogen oxides (NO_x), up to 50%. Modern automotive engines are controlled by the electronic control unit (ECU) which usually uses look-up tables with compensation of a proportional-integral (PI) closed-loop controller for lambda regulation. Since the nature of engine combustion is multivariable, time-varying, time-delay, and chaotic, look-up tables with PI

controller cannot produce desirable and accurate lambda control [2, 3].

So far, there are only a few papers focusing on air-ratio control, but some control strategies were developed for air-fuel ratio (AFR) control in the past decade, including the sliding mode control [4], radial basis function neural-network feed-forward feedback control [5], and model predictive control (MPC) using neural network-based models [6–8]. In the aforementioned researches, the most appropriate and the latest technique is MPC based on diagonal recurrent neural network (DRNN) [8] because of its fast computational time. The MPC is very robust and suitable for a multivariable, time-varying, and delay system that matches the characteristic of modern engine AFR control systems [9]. A reliable engine performance model is a core component of the MPC. However, the engine models developed in [5–8] were surrogate models, which were trained from the data generated by empirical equations. Moreover, there were

many assumptions in the empirical equations. In fact, many coefficients in the empirical equations are also difficult to determine for a real engine [10]. Therefore the neural-network prediction models derived from the data generated by empirical equations cannot reflect the actual performance of the controller in real engines. Meanwhile, their control target and engine models only focused on AFR instead of air ratio. AFR control is ineffective for engine performance control because it does not consider the fuel variation whereas air-ratio is a fuel independent index. Furthermore, in [5–8], only simulation tests were presented and the testing on real car engines has never been done. Besides, the inherent drawbacks of the neural network (NN) would make itself difficult put into practice including multiple local minima, user burden on selection of optimal NN structure, and overfitting. Another practical challenge to the modelling of engine lambda performance is that the lambda model is required to be updated for any changes in engine performance such as the engine aging or fair user modification on it. Therefore, the current research objective is to develop a reliable and nonlinear time-series prediction model for chaotic engine lambda behaviour and the model should have ability of online update as well.

Least-squares support vector machine (LS-SVM) is an alternative technique of nonlinear modelling [11, 12], which combines the advantages of NNs (handling large amount of highly nonlinear data) and nonlinear regression (high generalization). In recent years, LS-SVM has been successfully applied to a wide range of engineering applications [10, 13–17]. Hence it is believed that LS-SVM can also be applied to estimate the lambda model. However, LS-SVM is just an offline algorithm which cannot continually update the lambda prediction models with the subsequent samples for correction. Although there is an online version of LS-SVM [18] based on pruning, namely, incremental least-squares support vector machine (DLSSVM), it still suffers from two drawbacks. The first one is that there are two user-defined hyperparameters in DLSSVM which seriously affect the model accuracy and generalization. The optimization of these two hyperparameters is time-consuming and the optimality of these hyperparameter values is not guaranteed after incremental update. Since the number of support vectors typically grows linearly with the size of the training dataset, DLSSVM makes unnecessarily liberal use of basis functions which results in long computational time for real-time control applications.

To overcome the deficiencies of neural networks and LS-SVM, Tipping and Faul [19] proposed an advanced modelling technique, namely, relevance vector machine (RVM). RVM is an online machine learning technique which utilizes more flexible and sparser model without setting additional regularization parameters. RVM takes shorter training time than NN. In addition, the model updating time of RVM is also typically fast so that the RVM model can be continually updated whenever new lambda samples arrive.

In view of the deficiencies of the existing work and the chaotic nature of engine combustion [20] as well as the advantages of RVM, a promising avenue of research is to apply relevance vector machine and model predictive

controller together to complicated engine systems and test it on a real car engine for practical examination of the MPC. This paper presents a nonlinear MPC algorithm for air-ratio control based on an RVM model for a real car engine. The method is new for real-time engine lambda control. The RVM model can be updated continually to model the engine performance variation and severely nonlinear and chaotic natures. Based on the multiple-step-ahead prediction of the lambda, an optimal control signal is obtained to regulate the lambda to the desired value upon the change of engine operating condition. In order to show the advantages of the presented control algorithm over the existing algorithms, the control performance of the proposed relevance vector machine model predictive controller (RVMMPC) is compared with the latest neural network-based method for air-fuel ratio control, diagonal recurrent neural network MPC (DRNNMPC) [8] and conventional PI controller in production cars. Besides, it is interesting in extending the incremental least-squares support vector machine (DLSSVM) [18] to MPC and making comparison with the RVMMPC for engine air-ratio. To the best knowledge of the authors, this research is the first attempt at extending online LS-SVM and RVM to the domain of automotive engine air-ratio modelling and control.

2. Relevance Vector Machine

In the first part of this section, the algorithm of RVM modelling is briefly reviewed. In order to handle the online update of RVM model, an online sequential algorithm is adopted and described in the last part of this section.

2.1. RVM Modelling. Given a training dataset \mathbf{D} of N input vectors \mathbf{x}_n , $n = 1$ to N , along with N corresponding scalar-valued output y_n . The input vector $\mathbf{x}_n \in \mathbb{R}^m$ contains the previous measured engine time-series parameters including fuel injection time, throttle position, and air ratio at a specific time instant. The corresponding air ratio at that time is defined as the output $y_n \in \mathbb{R}$ which is assumed to contain zero-mean Gaussian noise with variance σ^2 . Hence, the probability of prediction error ε_n for y_n is a Gaussian distribution of zero mean and variance σ^2 , that is, $p(\varepsilon_n | \sigma^2) = N(0, \sigma^2)$, with

$$y_n = f(\mathbf{x}_n, \mathbf{w}) + \varepsilon_n, \quad (1)$$

$$p(y_n | \mathbf{x}_n, \mathbf{w}, \sigma^2) = N(\hat{y}_n, \sigma^2), \quad (2)$$

where $\hat{y}_n = f(\mathbf{x}_n, \mathbf{w})$ is the output prediction of the true value y_n , and $\mathbf{w} = [w_0, \dots, w_N]$ is the weight vector for the RVM model.

Generally, the prediction can be represented by

$$\begin{aligned} \hat{y} &= f(\mathbf{x}, \mathbf{w}) \\ &= w_0 + \sum_{n=1}^N w_n K(\mathbf{x}, \mathbf{x}_n) \\ &= \mathbf{w} \boldsymbol{\phi}(\mathbf{x}), \end{aligned} \quad (3)$$

where $K(\mathbf{x}, \mathbf{x}_i)$ is a basis function and $\phi(\mathbf{x}) = [1K(\mathbf{x}, \mathbf{x}_1) \cdots K(\mathbf{x}_n, \mathbf{x}_N)]^T$. In the current research, radial basis function (RBF) is chosen as the basis function K because it is commonly used for modelling problem [10].

Under the aforementioned formulation, the likelihood function of the output vector $\mathbf{y} = y_n, n = 1$ to N , is constructed by a multivariate Gaussian:

$$p(\mathbf{y} | \mathbf{w}, \sigma^2) = (2\pi)^{-N/2} \sigma^{-N} \exp\left\{-\frac{\|\mathbf{y} - \hat{\mathbf{y}}\|^2}{2\sigma^2}\right\}, \quad (4)$$

where $\hat{\mathbf{y}} = \Phi \mathbf{w}^T$ and Φ represents an $N \times (N + 1)$ design matrix with the notation $\Phi = [\phi(\mathbf{x}_1), \dots, \phi(\mathbf{x}_N)]^T$, and $\phi(\mathbf{x}_n) = [1K(\mathbf{x}_n, \mathbf{x}_1) \cdots K(\mathbf{x}_n, \mathbf{x}_N)]^T, n = 1$ to N .

The likelihood function, (4), is complemented by a prior over the weights, $\mathbf{w} = \{w_p\}, p = 0$ to N , to control the complexity of the model and avoid overfitting. The prior is a zero-mean Gaussian distribution and is defined over every weight w_n as follows:

$$p(\mathbf{w} | \boldsymbol{\alpha}) = (2\pi)^{-N/2} \prod_{p=0}^N \alpha_p^{1/2} \exp\left(-\frac{\alpha_p w_p^2}{2}\right). \quad (5)$$

The hyperparameter vector $\boldsymbol{\alpha} = [\alpha_0 \cdots \alpha_N]^T$ that controls how far each weight w_p is allowed to deviate from zero. Given $\boldsymbol{\alpha}$, using Bayes' rule, the posterior over \mathbf{w} is expressed as

$$p(\mathbf{w} | \mathbf{y}, \boldsymbol{\alpha}, \sigma^2) = \frac{p(\mathbf{y} | \mathbf{w}, \sigma^2) p(\mathbf{w} | \boldsymbol{\alpha})}{p(\mathbf{y} | \boldsymbol{\alpha}, \sigma^2)} = N(\mathbf{w} | \boldsymbol{\mu}, \boldsymbol{\Sigma}). \quad (6)$$

The posterior mean $\boldsymbol{\mu}$ and covariance $\boldsymbol{\Sigma}$ for \mathbf{w} are given as follows [19]:

$$\boldsymbol{\Sigma} = (\mathbf{A} + \sigma^{-2} \Phi^T \Phi)^{-1}, \quad (7)$$

$$\boldsymbol{\mu} = \mathbf{w} = \sigma^{-2} \boldsymbol{\Sigma} \Phi^T \mathbf{y},$$

where \mathbf{A} is defined as $\text{diag}(\alpha_0 \cdots \alpha_N)$.

The posterior mean $\boldsymbol{\mu}$ is an estimation of the weight vector \mathbf{w} for prediction. The only unknowns in (7) are the hyperparameters $\boldsymbol{\alpha}$ which can be estimated via a type-II maximum likelihood procedure [19]. It is called sparse Bayesian learning which is formulated as the local maximization with respect to $\boldsymbol{\alpha}$ of the marginal likelihood or, equivalently, its logarithm $L(\boldsymbol{\alpha})$:

$$L(\boldsymbol{\alpha}) = \log p(\mathbf{y} | \boldsymbol{\alpha}, \sigma^2) = \log \int_{-\infty}^{\infty} p(\mathbf{y} | \mathbf{w}, \sigma^2) p(\mathbf{w} | \boldsymbol{\alpha}) d\mathbf{w}$$

$$= -\frac{1}{2} [N \log 2\pi + \log |\mathbf{C}| + \mathbf{y}^T \mathbf{C}^{-1} \mathbf{y}], \quad (8)$$

where $\mathbf{C} = \sigma^2 \mathbf{I} + \Phi \mathbf{A} \Phi^T$.

By maximizing (8) over $\boldsymbol{\alpha}$, the most probable values $\boldsymbol{\alpha}_{\text{MP}}$ can be generated. Then, \mathbf{A} can be obtained by substituting $\boldsymbol{\alpha}_{\text{MP}}$ into (7). Afterwards, the covariance $\boldsymbol{\Sigma}$ and posterior

mean $\boldsymbol{\mu} = [\mu_0, \mu_1, \dots, \mu_N]^T$ can be estimated. Therefore, the RVM model f can be found by setting $\mathbf{w} = \boldsymbol{\mu}$:

$$\hat{y} = y_p = f(\mathbf{x}, \mathbf{w}) = f(\mathbf{x}, \boldsymbol{\mu}),$$

$$y_p = \mu_0 + \sum_{n=1}^N \mu_n K(\mathbf{x}, \mathbf{x}_n), \quad (9)$$

$$y_p = \mu_0 + \sum_{n=1}^N \mu_n \exp\left(-\frac{\|\mathbf{x} - \mathbf{x}_n\|^2}{\sigma^2}\right),$$

where \hat{y} is the prediction of the output lambda with the unseen input data \mathbf{x} containing the recent measured engine data series, which is explained in Section 4.1. One crucial observation is that the optimal values of many hyperparameters α_p are typically infinite. Considering that $w_p = \mu_p \propto (1/\alpha_p)$, this leads to a parameter posterior infinitely peaked at zero for many weights w_p . In consequence, the posterior mean $\boldsymbol{\mu}$ consists of very few nonzero elements. This results in good sparseness for RVM whereas LS-SVM and NN do not have this advantage. A good sparseness implies that the computational time for prediction can be shortened.

2.2. Online RVM Modelling. In order to train and update the RVM model continually, an online algorithm for RVM [19] is employed. The algorithm starts with an empty set of basis function for the RVM model and sequentially adds basis functions $\phi_n = [1K(\mathbf{x}_n, \mathbf{x}_1) \cdots K(\mathbf{x}_n, \mathbf{x}_N)]^T$ to increase the marginal likelihood and modify their corresponding weights w_n . Within the same principal framework, the likelihood can also be updated by deleting those basis functions which subsequently become redundant. Removing a basis function ϕ_n implies that the corresponding \mathbf{x}_n is no longer important and can be excluded from the design matrix Φ in constructing the RVM model f , which can be simply done by setting the corresponding hyperparameter α_p equal to infinity (practically a very large value). In this way, the corresponding weight value $w_p = \mu_p \propto (1/\alpha_p)$ becomes zero. Since w_0 is a bias corresponding to no training data, only the weights w_1, w_2, \dots, w_N are considered. The data \mathbf{x}_n with nonzero weights $w_n, n = 1$ to N , are referred to relevance vectors. Since the basis functions are sequentially added to or deleted from the lambda model by RVM, the likelihood can be continually updated and hence this mechanism makes online update of the lambda model feasible.

In the following discussions, the constraints to add and delete basis function from the RVM lambda model are given, and then the detailed online RVM algorithm is presented.

2.2.1. Constraints for the Basis Functions Update. In (8), the covariance matrix \mathbf{C} includes all basis functions in the RVM prediction model. When it is necessary to remove a basis function ϕ_n from \mathbf{C} , the new covariance matrix \mathbf{C}_{-n} with the influence of removed basis function ϕ_n can be expressed as follows:

$$\mathbf{C}_{-n}(\boldsymbol{\alpha}) = \sigma^2 \mathbf{I} + \sum_{j \neq n} \alpha_j^{-1} \phi_j \phi_j^T. \quad (10)$$

It was shown in [21] that $L(\boldsymbol{\alpha})$ has a unique maximum with respect to $\boldsymbol{\alpha}$:

$$\alpha_n = \frac{S_n^2}{Q_n^2 - S_n}, \quad \text{if } Q_n^2 > S_n, \quad (11)$$

$$\alpha_n = \infty, \quad \text{if } Q_n^2 \leq S_n, \quad (12)$$

where

$$S_n = \phi_n^T \mathbf{C}_{-n}^{-1} \phi_n, \quad (13)$$

$$Q_n = \phi_n^T \mathbf{C}_{-n}^{-1} \mathbf{y}, \quad (14)$$

with $\phi_n = [1K(\mathbf{x}_n, \mathbf{x}_1) \cdots K(\mathbf{x}_n, \mathbf{x}_N)]^T$, $n = 1$ to N .

The results of (11) and (12) imply that if ϕ_n is currently included in the lambda model (i.e., $\alpha_n < \infty$) and $Q_n^2 \leq S_n$, then ϕ_n can be deleted by setting α_n to ∞ . On the other hand, if ϕ_n is currently excluded from the lambda model (i.e., $\alpha_n = \infty$) and $Q_n^2 > S_n$, then ϕ_n can be added by setting α_n to some optimal finite values. With these constraints, an online RVM algorithm can be implemented here in after.

2.2.2. Online Sequential Algorithm of RVM. The steps for the implementation of the online RVM algorithm are described as follows.

- (1) Initialize σ^2 to some sensible values (e.g., $\text{var}[\mathbf{y}] \times 0.1$) and all α_n are notionally set to infinity.
- (2) Initialize S_n and Q_n with a single basis function ϕ_n from (13) and (14) and compute new α_n from (11) which can be simplified as follows.

$$\alpha_n = \frac{\|\phi_n\|^2}{\|\phi_n^T \mathbf{y}\| / \|\phi_n\|^2 - \sigma^2}. \quad (15)$$

- (3) Explicitly compute $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ (which are scalars initially), along with initial values of S_n and Q_n for all N basis functions ϕ_n using (7) and (10)–(14).
- (4) Select a candidate basis function ϕ_n from the set of all N basis functions.
- (5) Compute $\theta_n = Q_n^2 - S_n$.
- (6) If $\theta_n > 0$ and $\alpha_n < \infty$ (i.e., ϕ_n is included in the model), then reestimate α_n using (11).
- (7) If $\theta_n = 0$ and $\alpha_n = \infty$, then add ϕ_n to the model with updated α_n .
- (8) If $\theta_n < 0$ and $\alpha_n < \infty$, then delete ϕ_n from the model and set $\alpha_n = \infty$.
- (9) Estimate the noise level, and update σ^2 as follows:

$$\sigma^2 = \frac{\|\mathbf{y} - \hat{\mathbf{y}}\|^2}{\sum_{n=1}^N \alpha_n \boldsymbol{\Sigma}_{nn}}. \quad (16)$$

- (10) Recompute or update $\boldsymbol{\Sigma}$, $\boldsymbol{\mu}$ and all S_n and Q_n using (7) and (10)–(14).

- (11) If converged then terminate, otherwise go to Step 4.

Remarks. For the initialization in Step (2), a potential basis function could be the one with the largest normalized projection onto the output vector which gives the largest initial likelihood. In Step (4), a candidate basis function ϕ_n , both included in and excluded from the current RVM model, must be selected for updating but the selection is purely at random. In Step (10), $\boldsymbol{\Sigma}$, $\boldsymbol{\mu}$ and all S_n and Q_n are recomputed in full forms of (7) and (10)–(14). In Step (11), attaining a local maximum of the marginal likelihood must be judged. The online algorithm terminates when the changes in $L(\boldsymbol{\alpha})$ for all basis functions in the RVM model are smaller than 10^{-6} and all other $\theta_n \leq 0$ [19].

The aforementioned online sequential algorithm for RVM ensures to increase the marginal likelihood at each step until a local maximum is attained. Although adding and deleting basis functions appear notionally to be independent, posterior statistics for all basis functions are being maintained concurrently; that is, all elements of $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ corresponding to removed basis functions are trivially zero.

3. MPC with RVM Model

The structure of the proposed RVMMPCC is shown in Figure 1. The controller consists of the RVM engine lambda model and the optimizer based on Brent's method [22]. The RVM engine lambda 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' , that minimizes the following performance criterion over the specified time horizon, and then the optimal fuel injection time signal u is sent to the engine:

$$\begin{aligned} \min J(u') = & \sum_{j=N_1}^{N_2} (y_r(t+j) - y_p(t+j))^2 \\ & + \rho \sum_{j=1}^{N_u} (u'(t+j-1) - u'(t+j-2))^2, \end{aligned} \quad (17)$$

where N_1 and N_2 define the prediction horizon. t is the time step. N_u is the control horizon. ρ is a user-defined control weighting factor which 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 (17) are the tentative fuel injection time at the time step $t+j-1$ and $t+j-2$, respectively. The second part of (17) ensures the stability of the controller output. $y_r(t+j)$ is the target lambda at the time step $t+j$, and y_p is the predicted lambda by the RVM model which is computed from (9) at the time step $t+j$, from which the input vector \mathbf{x} consists of three time series of engine control parameters: $u'(t+j-1)$, $u'(t+j-2)$, ..., throttle position $\text{TP}(t+j-1)$, $\text{TP}(t+j-2)$, ..., and

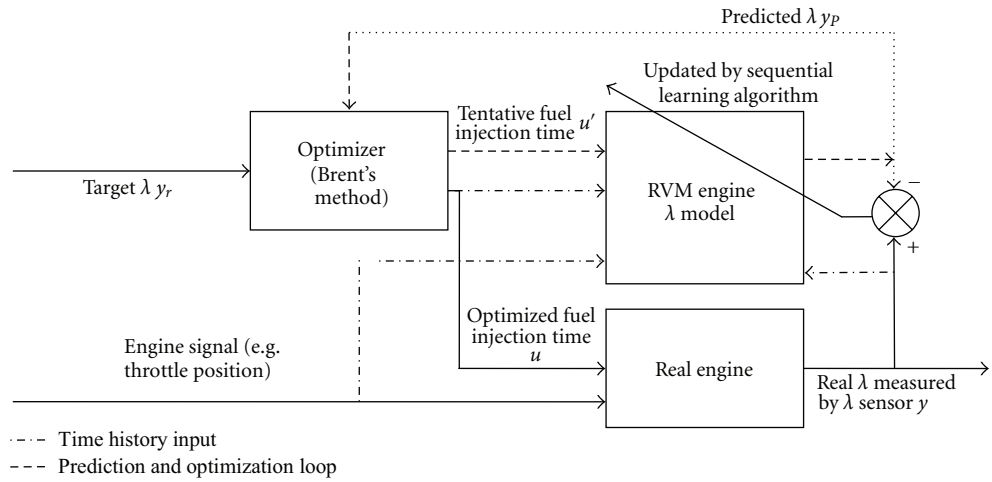


FIGURE 1: Structure of RVMMPCC for engine lambda control.

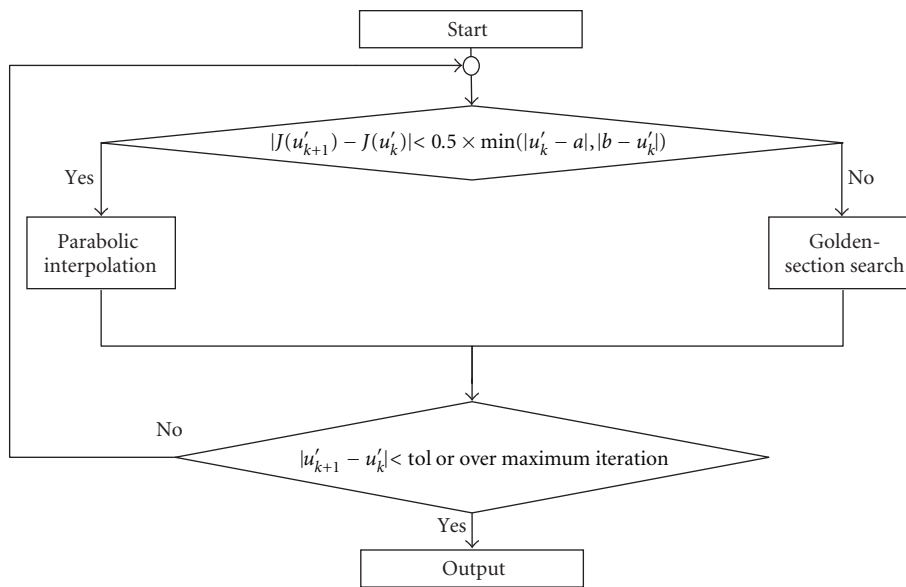


FIGURE 2: General working principle of Brent's method.

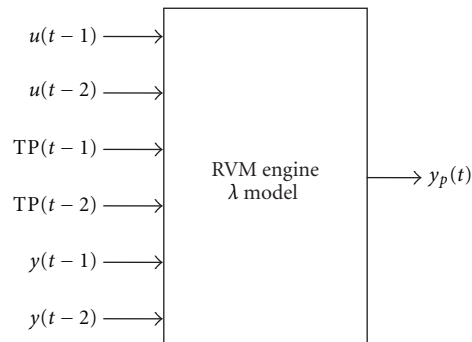


FIGURE 3: Structure of RVM engine lambda model.

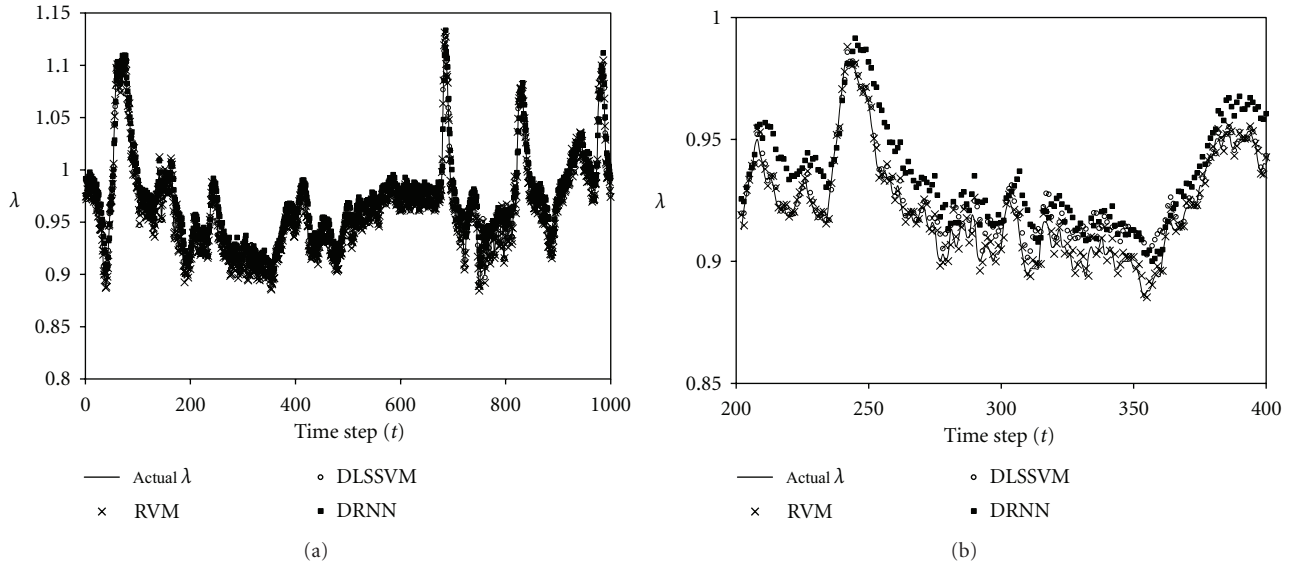


FIGURE 4: (a) Comparison between predicted lambda and the corresponding actual lambda; (b) in detail.

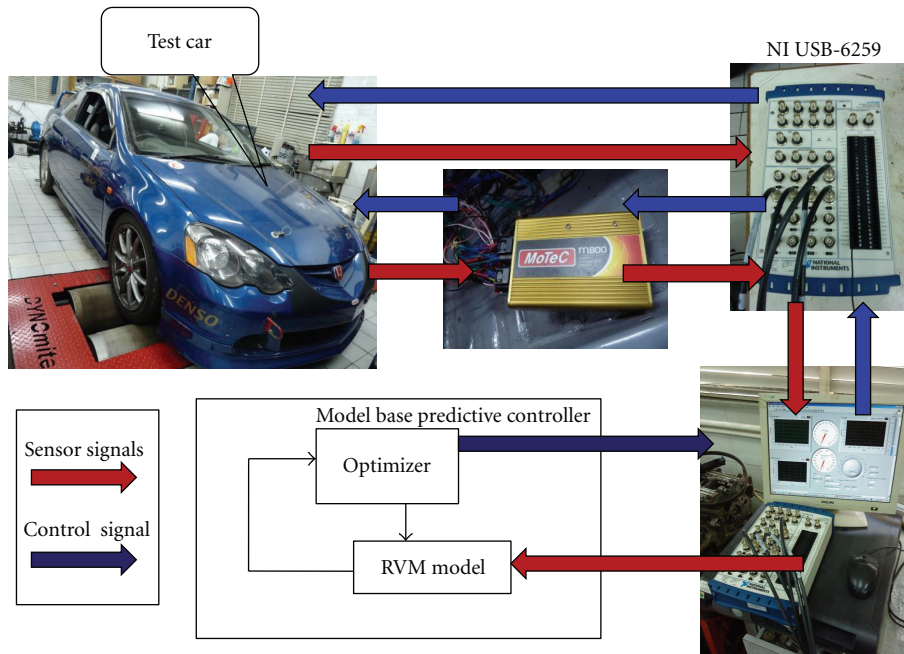


FIGURE 5: Experimental setup.

previous measured lambda $y(t+j-1)$, $y(t+j-2)$ In other words,

$$y_p(t+j) = \mu_0 + \sum_{n=1}^N \mu_n \exp\left(-\frac{\|\mathbf{x} - \mathbf{x}_n\|}{\sigma^2}\right). \quad (18)$$

3.1. Single-Dimension Optimization Approach. The original optimization problem involved in this paper is multidimensional and constrained with the tentative control signals over the control horizon N_u and tentative fuel injection times $u'(t)$, $u'(t+1)$, ..., $u'(t+N_u-1)$, which can minimize

the objective function $J(u')$ of (17). Then the predicted lambda values, $y_p(t+N_1)$, $y_p(t+N_1+1)$, ..., $y_p(t+N_2)$, can trace the target lambda values, $y_r(t+N_1)$, $y_r(t+N_1+1)$, ..., $y_r(t+N_2)$, by using the optimized fuel injection time series. Each fuel injection time is normally bounded within the range from 3 ms to 60 ms. However, the multi-dimensional optimization always requires heavy computation, especially when constraints exist. Real-time control applications often put emphasis on computational speed. The research of [7] also showed that the one-dimensional approach is efficient for real time AFR control and the overall

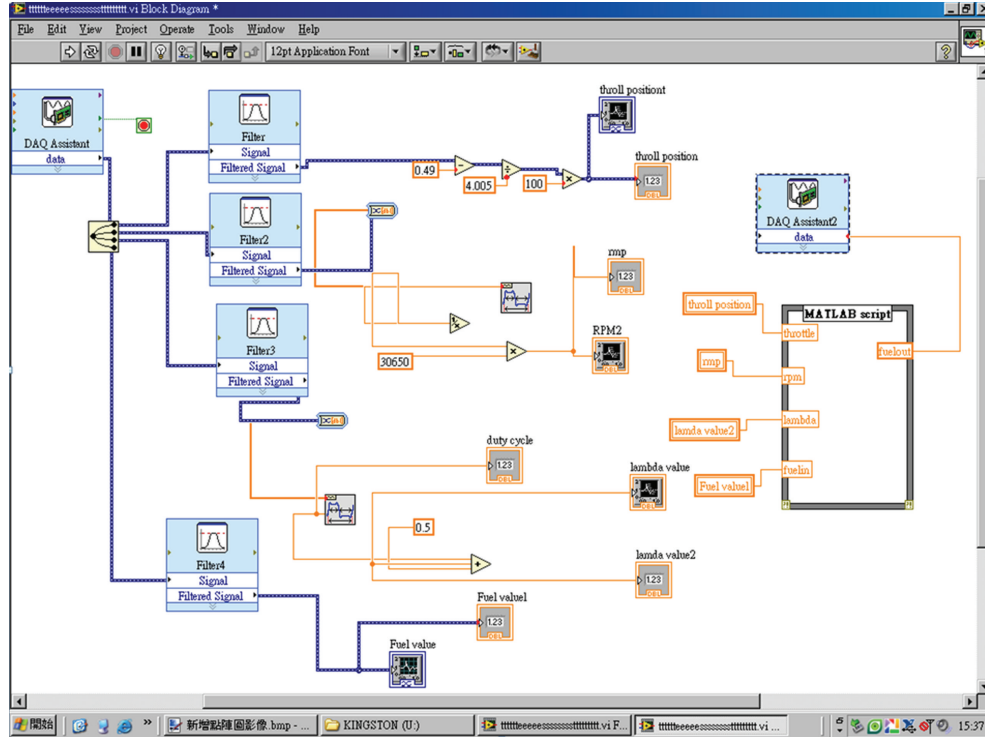


FIGURE 6: LabVIEW interface program.

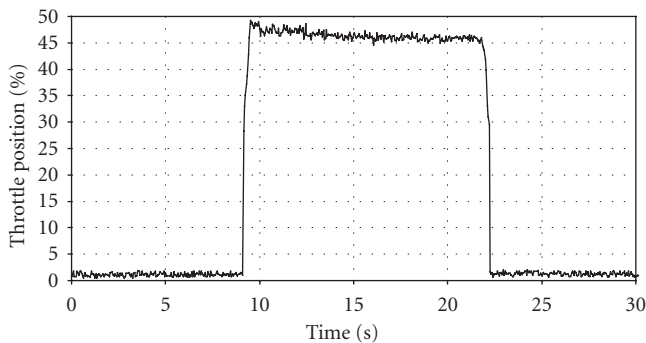


FIGURE 7: Throttle position against time in pilot test 1.

tracking error is similar to that using multidimensional optimization approach. Therefore, the optimization problem to be solved is reduced to onedimensional. 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, that is, $u'(t) = u'(t+1), \dots, = u'(t+N_u - 1)$. In this way, only one parameter $u'(t)$ is needed to determine, 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 methods available in the literature; it is impossible to examine all of them. For illustrative purpose, a well-known technique—Brent's method—is examined in this research. 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 an interpolating 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 [22]. According to this advantage, Brent's method was selected as the MPC optimizer in this study. The general working principle of Brent's method is shown in Figure 2. The detail optimization procedure of Brent's method was presented in [22] 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, as well as the tolerance, tol , for stopping the optimization procedure. The three variables are set at 3, 60, and 0.05, respectively, because the fuel injection time varies within 3 to 60 ms from 0 to 100% throttle.

4. Implementation and Evaluation of RVM Engine Air-Ratio Model

4.1. RVM Engine Air-Ratio Model Implementation. The objective of the RVM engine lambda model is to predict the future lambda y_p from three time series of inputs: fuel injection time u , throttle position TP, and previous measured lambda y . The structure of the RVM model was chosen to

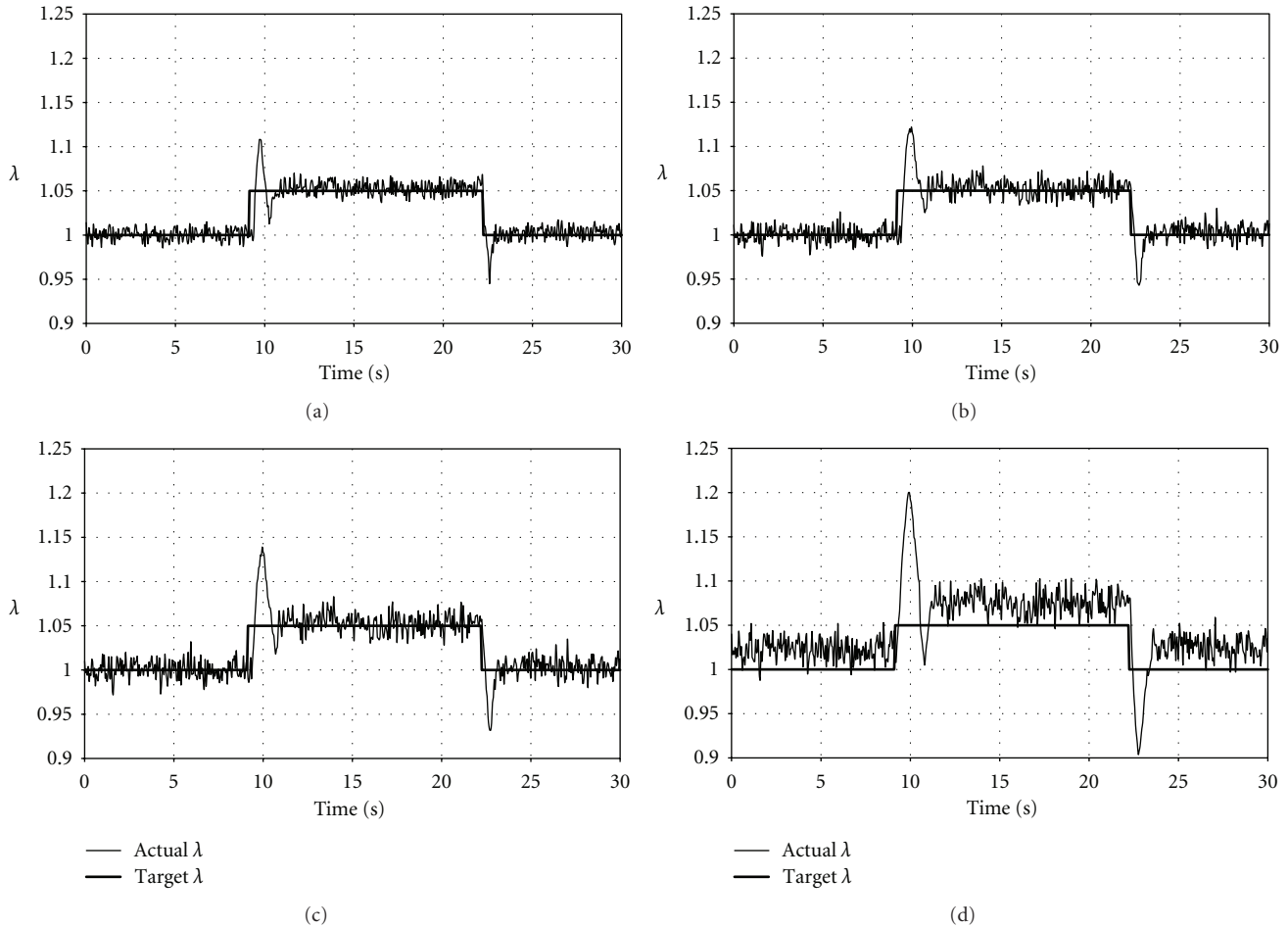


FIGURE 8: Lambda control results of pilot test 1 of (a) RVMMP, (b) DLSSVMMP, (c) DRNNMPC, and (d) PI controller.

be second order (i.e., 2 past time steps) which gives the minimum prediction error [7, 8], and the structure is shown in Figure 3.

To obtain the engine data for building the engine lambda model, 3000 data samples including lambda, fuel injection time, and throttle positions were collected. These 3000 lambda samples were collected over a Honda Type-R K20A i-VTEC engine controlled by a MoTeC M800 programmable electronic control unit with nonfactory calibration data. A dyno test was done for collecting the data samples. In the dyno test, the engine speed was ranged from 1500 r/min to 6500 r/min with random throttle positions bounded between 5% and 65%. The lambda data samples were collected using an in-car wide-band lambda sensor at a sampling frequency of 200 Hz. The first 2000 data samples were used as training dataset \mathbf{D} to build the lambda model. The last 1000 data samples were used as test dataset TEST and the first 900 data samples of TEST were regarded as update dataset UPDATE for real-time updates. The lambda model was updated every 100 measured lambda data samples during prediction. With respect to the training dataset

\mathbf{D} , the test dataset TEST is unseen cases for testing the generalization of the built lambda model.

4.2. Evaluation Criteria for Engine Air-Ratio Model. To illustrate the accuracy, superiority, and online update ability of the proposed RVM model, its prediction result was compared with those obtained from the latest methods, DLSSVM [18] and DRNN [8].

After obtaining the lambda model $RM(\mathbf{x})$ with RVM through the training algorithm presented in Section 2 over \mathbf{D} , $RM(\mathbf{x})$ was then updated for 9 times with UPDATE, 100 data samples each time, to build the updated RVM lambda model $RM^*(\mathbf{x})$. Apart from the RVM model, classical LS-SVM was employed to build the lambda model $DM(\mathbf{x})$ over \mathbf{D} . After that, $DM(\mathbf{x})$ was updated with UPDATE but employing the updating algorithm presented in [18] to build the incremental LS-SVM model $DM^*(\mathbf{x})$. Moreover, DRNN, the same modelling algorithm for AFR model predictive control presented in [8], was applied to estimate the lambda model, $NN(\mathbf{x})$. The network was trained for 60 epochs. After that, $NN(\mathbf{x})$ was updated with UPDATE but

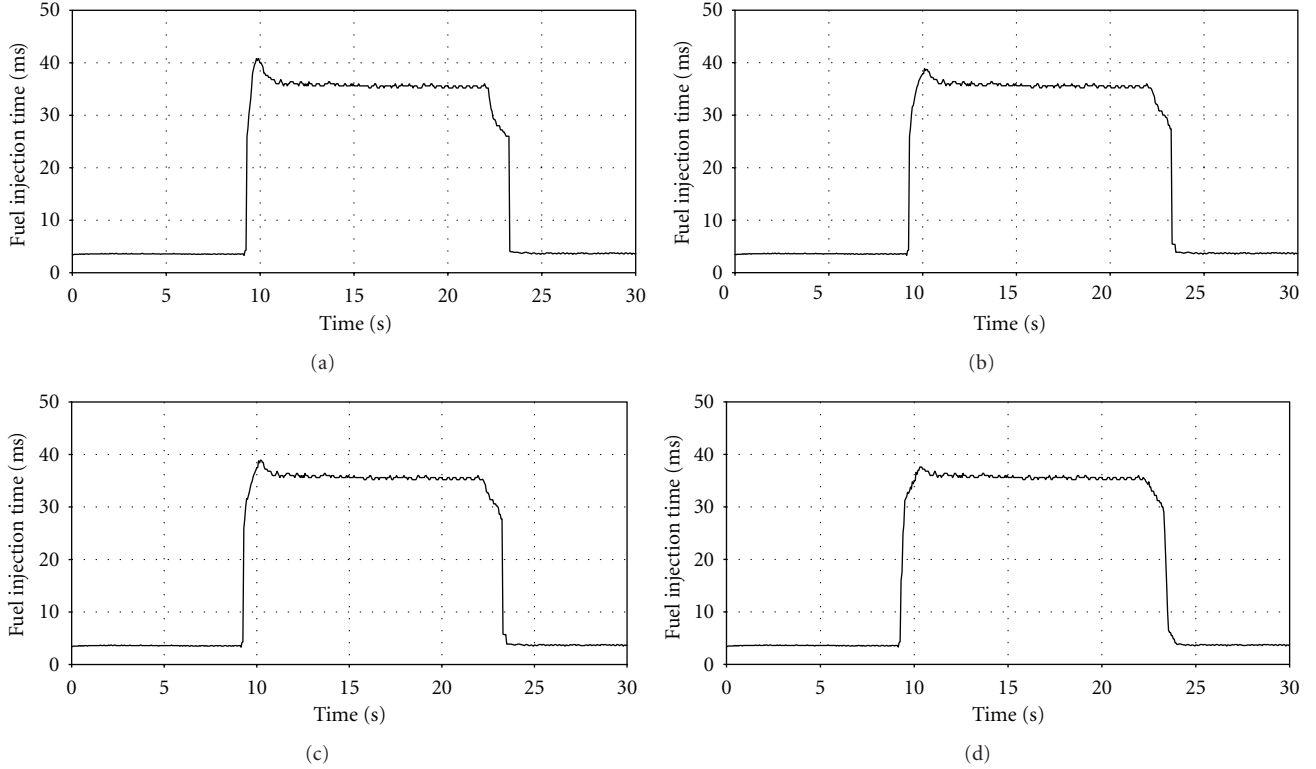


FIGURE 9: Fuel injection time of pilot test 1 of (a) RVMMPCC, (b) DLSSVMMPCC, (c) DRNNMPC, and (d) PI controller.

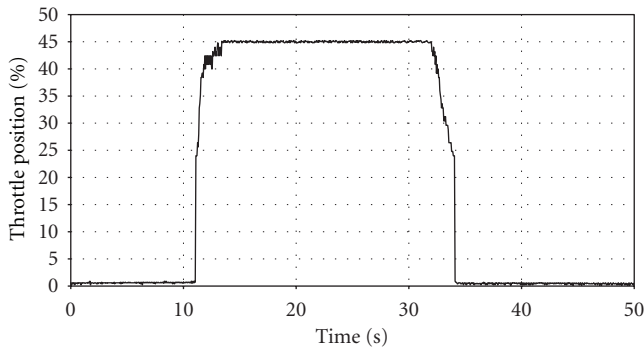


FIGURE 10: Throttle position against time in pilot test 2.

employing the dynamic back-propagation with automatic differentiation technique presented in [8] to build the online model $NN^*(\mathbf{x})$.

After constructing all $RM^*(\mathbf{x})$, $DM^*(\mathbf{x})$, and $NN^*(\mathbf{x})$, the performance of the three lambda models can be evaluated in terms of accuracy. Since the range of lambda for combustible mixture is very narrow, the prediction errors of the above three models are presented by logarithmic mean absolute error ($LMAE$), and they were evaluated one by one against the test dataset, TEST, using (19):

$$LMAE = -\log \left[\frac{1}{T} \sum_{k=1}^T |y_k - f^*(\mathbf{x}_k)| \right], \quad (19)$$

where $f^*(\mathbf{x}_k)$ represents either $RM^*(\mathbf{x}_k)$, $DM^*(\mathbf{x}_k)$, or $NN^*(\mathbf{x}_k)$, \mathbf{x}_k is the k th new input vector for lambda prediction, y_k is the corresponding actual lambda value of $f^*(\mathbf{x}_k)$, and T is the total number of predictions. The value of T is equal to 1000 in this case study. Besides $LMAE$, the accuracies of the three models were also evaluated by the linear regression R^2 value. The prediction results between the predicted lambda values and the corresponding actual lambda values over TEST are shown in Figure 4.

Table 1 shows the $LMAE$ and R^2 values of the three lambda models. According to the $LMAE$ values in Table 1, the RVM lambda model $RM^*(\mathbf{x})$ outperforms $DM^*(\mathbf{x})$ and $NN^*(\mathbf{x})$ by approximately 4% and 29%, respectively. It is believed that the accuracy of the RVM lambda model can be significantly increased if the model update is continually carried out. Moreover, as compared with DLSSVM, RVM has better sparseness. This property is very important for online control and system identification because the computational time and the size of memory are fewer.

In addition to the improvement on prediction accuracy, RVM can also reduce the time for training and updating the lambda model significantly. Table 2 shows the model training time, average model updating time, and cumulative model updating time after nine times of updates. In Table 2, the lambda model training and average model updating time of RVM are 1.52 s and 0.08 s, respectively. The lambda model training time of $RM^*(\mathbf{x})$ is the shortest among the three lambda models. The average model updating time of $RM^*(\mathbf{x})$, 0.08 s, is only about one-sixth of $DM^*(\mathbf{x})$,

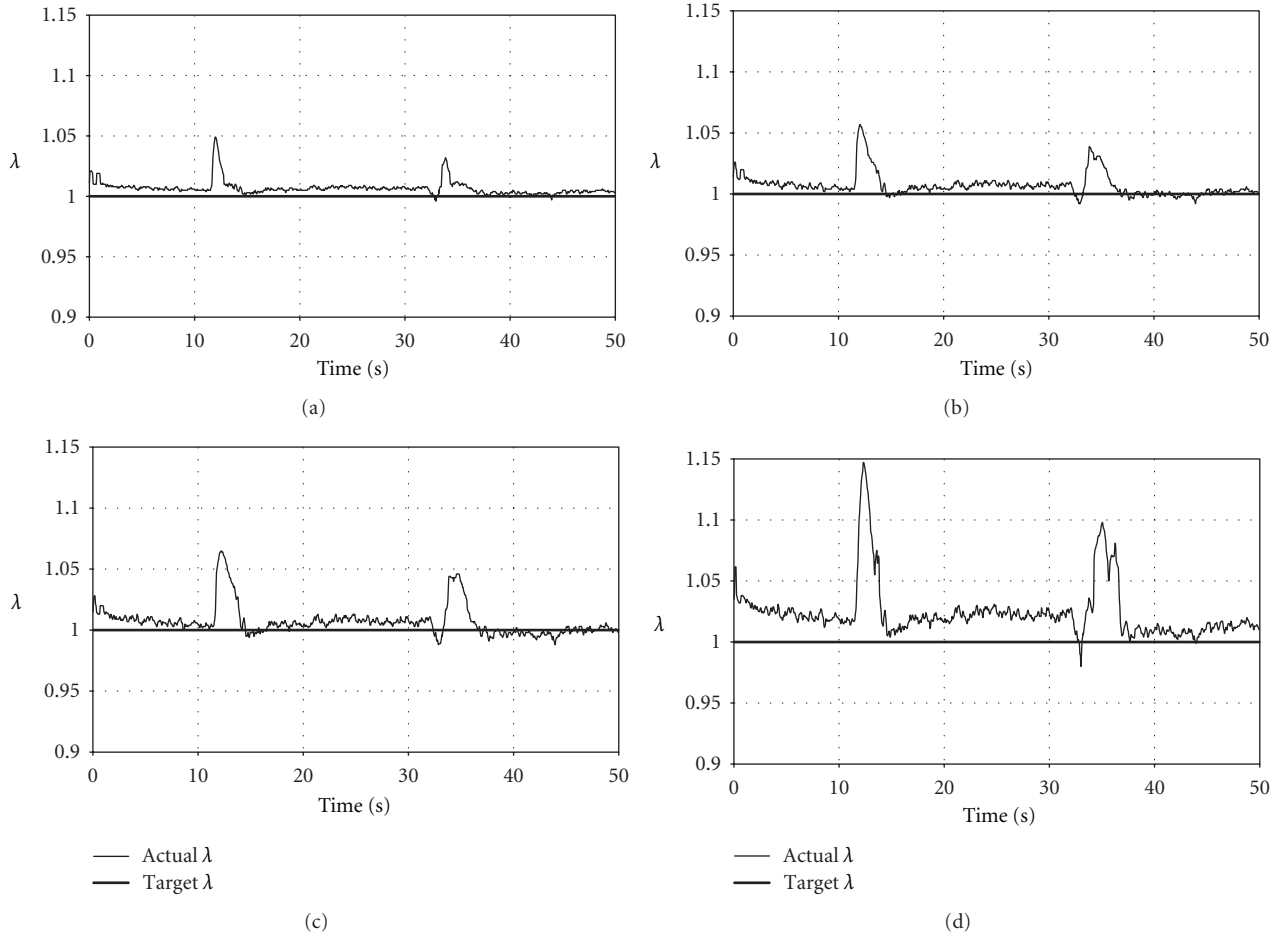


FIGURE 11: Lambda control results of pilot test 2 of (a) RVMMPCC, (b) DLSSVMMPCC, (c) DRNNMPCC, and (d) PI controller.

0.52 s. It is because in RVM, the updating procedure is applied to every basis function individually while DLSSVM has to work on a large matrix for its estimated inversion. Moreover, the average model updating time of $RM^*(\mathbf{x})$ is also about one-eleventh of $NN^*(\mathbf{x})$. This achievement may be accomplished by the effective online algorithm of RVM. The total cumulative model updating time saved by RVM from DLSSVM and DRNN after 9 update iterations is 3.99 s and 7.56 s, respectively. However, if the lambda model is often updated, the cumulative model updating time saved by RVM will be very significant. Obviously, RVM has more advantage than DLSSVM and DRNN algorithms. As a whole, the high accuracy and short updating time of the lambda model using RVM make online model predictive air-ratio control more feasible.

Experimental results show that the model accuracy, model training time, and updating time of RVM are the best among the aforementioned algorithms. Therefore, RVM was confidently selected to implement the model predictive engine air-ratio controller.

5. Implementation and Evaluation of RVMMPCC

5.1. Experimental Setup. The proposed RVMMPCC algorithm was implemented and tested on a Honda Type-R K20A i-VTEC engine with MoTeC M800 programmable ECU and National Instrument (NI) USB-6259. The model predictive control algorithm was implemented using MATLAB. MoTeC M800 is mainly used for engine control, whereas NI USB-6259 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 USB-6259 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 LabVIEW interface program are shown in Figures 5 and 6, respectively.

The initial offline training data for building the RVM model were obtained using a wide-band lambda sensor subject to random throttle positions and are discussed in

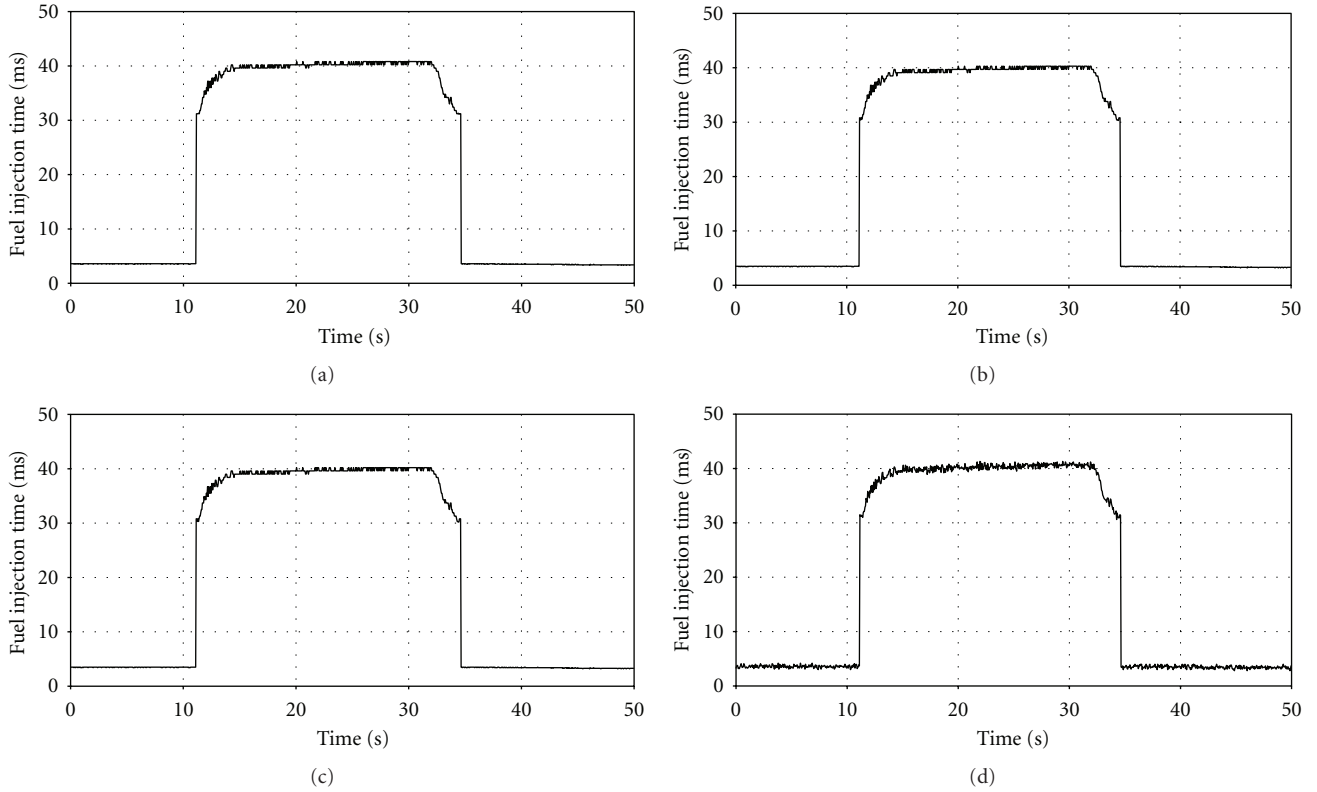


FIGURE 12: Fuel injection time of pilot test 2 of (a) RVMMPCC, (b) DLSSVMMPCC, (c) DRNNMPCC, and (d) PI controller.

TABLE 1: $LMAE$ and R^2 of different lambda models.

Lambda model	$LMAE$	Improvement in $LMAE$ (%)	R^2	Improvement in R^2 (%)
$RM^*(\mathbf{x})$	2.4112	—	0.9868	—
$DM^*(\mathbf{x})$	2.3166	4.08 ^a	0.9816	0.53 ^a
$NN^*(\mathbf{x})$	1.8674	29.12 ^b	0.9511	3.75 ^b

^aRVM over DLSSVM,

^bRVM over DRNN.

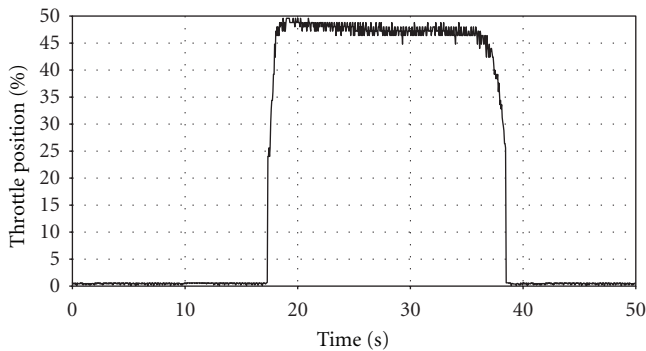


FIGURE 13: Throttle position against time in pilot test 3.

the pervious section. There were three pilot tests in this study. The tests evaluate the tracking error, robustness, and adaptability of the controllers respectively.

5.2. Pilot Test 1: System Tracking Ability. In pilot test 1, the test cycle is shown in Figure 7 where the throttle position changes from 0 to 50% throttle (i.e., partial throttle). In this test, the lambda value is needed to track the target values from the stoichiometric value (1.00) to a value for the best brake-specific fuel consumption (1.05) as the throttle position is changed. This lambda value for such partial throttle is a normal requirement for automobiles. After choosing the sampling time to be 0.005 s, the tracking ability of the RVMMPCC can be examined. By testing many values around the setting used in [8], the parameters of the optimizer were chosen as $N_1 = 1$, $N_2 = 10$, $\rho = 0.5$, and $N_u = 3$. With the test cycle shown in Figure 7 and the parameters chosen, the lambda control result and the corresponding fuel injection time are shown in Figures 8(a) and 9(a), respectively. In order to show the advantage of the RVMMPCC, its control result is compared with other two latest model predictive controllers, decremental LS-SVM model predictive controller (DLSSVMMPCC) and DRNN model predictive controller (DRNNMPCC), as well as the

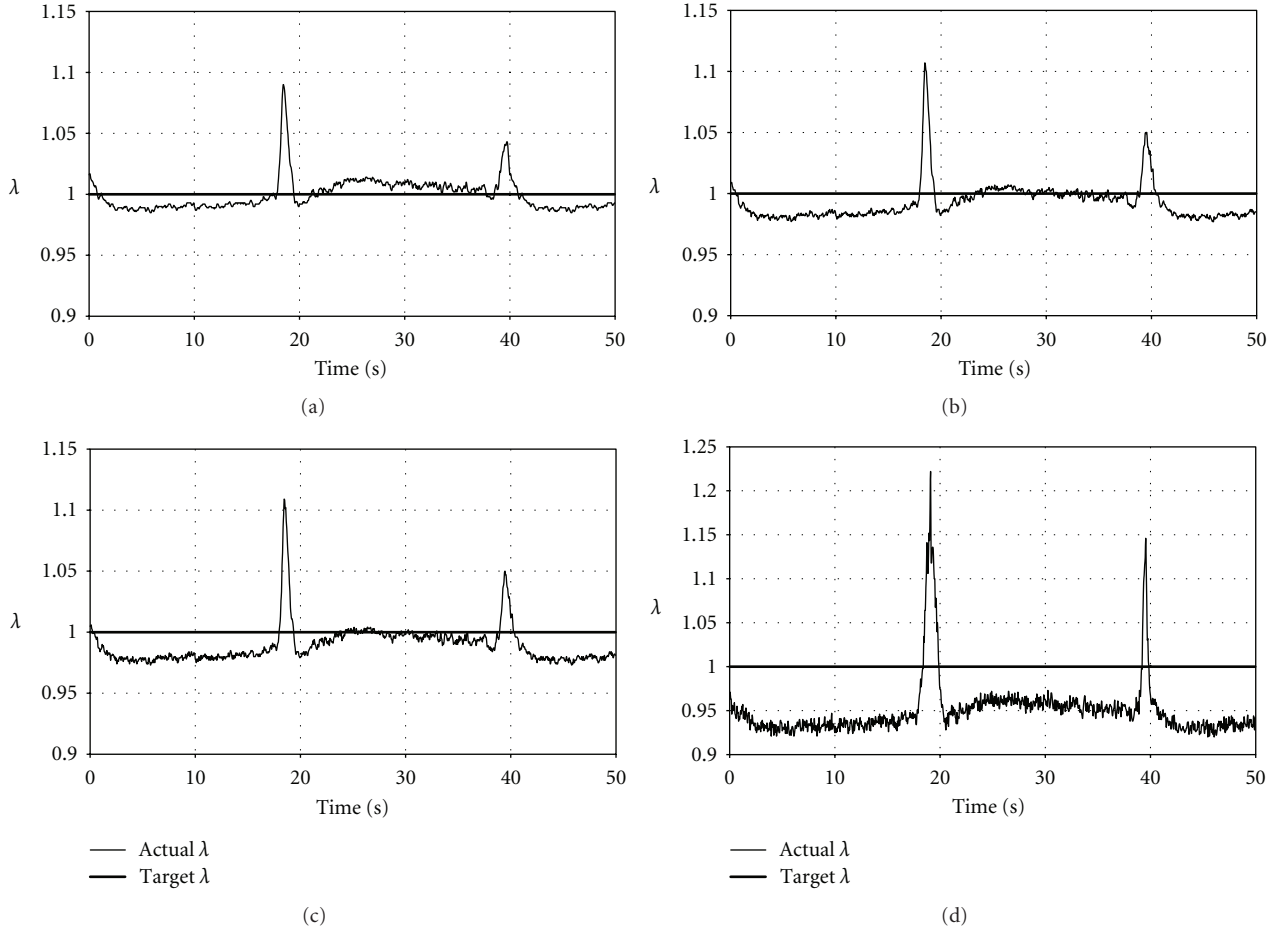


FIGURE 14: Lambda control results of pilot test 3 of (a) RVMMP, (b) DLSSVMMP, (c) DRNNMPC, and (d) PI control.

TABLE 2: Training and updating time of different lambda models.

Lambda model	Training time (s)	Average updating time (s)	Cumulative updating time (s)
$RM^*(\mathbf{x})$	1.52	0.08	0.71
$DM^*(\mathbf{x})$	2.02	0.52	4.70
$NN^*(\mathbf{x})$	3.52	0.92	8.27

PI control algorithm used in the existing automotive ECU. The engine lambda models used in the DLSSVMMP and DRNNMPC are the models mentioned in Section 4.2. The parameters of nonlinear optimization for the DLSSVMMP, and DRNNMPC were the same as those of RVMMP. The PI gains of the PI controller were obtained by the Ziegler-Nichols method. The lambda control results and the corresponding fuel injection time of the DLSSVMMP, DRNNMPC, and PI controller are shown in Figures 8(b)–8(d) and 9(b)–9(d), respectively.

Figure 8(a) shows that the RVMMP can control the lambda to follow the target lambda with the smallest tracking error and overshoot among all the controllers. The control performances of the four controllers are shown in Table 3. As the range of lambda for combustible mixture is very narrow, logarithmic mean absolute error is chosen as the logarithmic

tracking error (LTE) to evaluate the tracking ability of the controllers and is defined by

$$LTE = -\log \left[\frac{1}{T} \sum_{t=1}^T |y_t - y_r(t)| \right], \quad (20)$$

where t is time step, T is the total number of time step, y_t is the actual lambda at each time step, and $y_r(t)$ is the corresponding target lambda at each time step. Table 3 reveals that the LTE of the RVMMP outperforms the DLSSVMMP, DRNNMPC and PI controller by approximately 5%, 8%, and 38%, respectively.

5.3. Pilot Test 2: System Robustness. In pilot test 2, the test cycle is shown in Figure 10 where the throttle position changes from 0 to 50% throttle. This change can be viewed

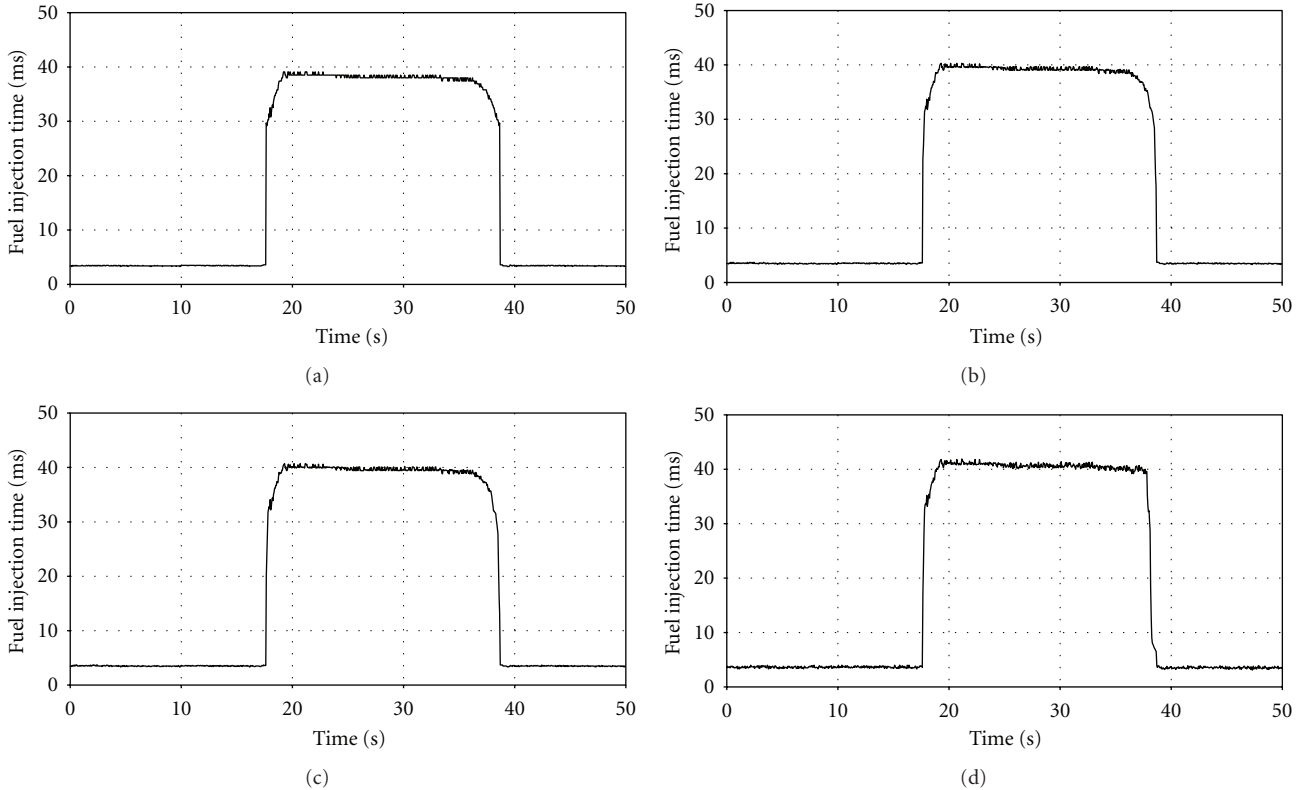


FIGURE 15: Fuel injection time of pilot test 3 of (a) RVMMP, (b) DLSSVMMP, (c) DRNNMPC, and (d) PI controller.

as a disturbance. In this test, the lambda value is needed to control within the $\pm 5\%$ bounds of the stoichiometric value (1.0). After choosing the sampling time to be 0.005 s, the effectiveness of the RVMMP can be examined. The parameters of the optimizer were chosen as $N_1 = 1$, $N_2 = 10$, $\rho = 0.5$, and $N_u = 3$ again. The lambda control result and the fuel injection time of the RVMMP are shown in Figures 11(a) and 12(a), respectively. Similar to pilot test 1, the control result is compared with the two model predictive controllers, DLSSVMMP and DRNNMPC, as well as typical PI controller. The control results and the corresponding fuel injection time of the DLSSVMMP, DRNNMPC, and PI controller are shown in Figures 11(b)–11(d) and 12(b)–12(d), respectively. Figure 11(a) shows that the RVMMP can regulate the lambda to the target lambda with the smallest deviation (i.e., minimum *LTE*) in the steady state among all the controllers. Moreover, the RVMMP can also achieve the smallest overshoot in the transient state among all the controllers. These results show the superior robustness of the RVMMP. The control performances of the four controllers are shown in Table 4, and the control performance of the RVMMP in terms of *LTE* is superior to the DLSSVMMP, DRNNMPC, and PI controller by 3%, 6%, and 35%, respectively.

5.4. Pilot Test 3: Online Update Ability. In the previous two tests, the “bypass air valve” of the test engine is 60% opening. In order to test the update ability of the RVMMP, the

“bypass air valve” was set from 60% to 30%. It is equivalent to the clogging of the engine intake filter as the engine aging. Normally, the lambda value must decrease under the same throttle position as lack of intake air. The testing procedure and the parameter setting are exactly the same as those in pilot test 2 except the change of the “bypass air valve” position. The variation of throttle position in the test cycle is shown in Figure 13. The lambda control result and the fuel injection time of the RVMMP are shown in Figures 14(a) and 15(a), respectively. Similar to pilot tests 1 and 2, the control result is compared with the two model predictive controllers, DLSSVMMP and DRNNMPC, as well as typical PI controller. The control results and the corresponding fuel injection time of the DLSSVMMP, DRNNMPC and PI controller are shown in Figures 14(b)–14(d) and 15(b)–15(d) respectively. Figures 14(a)–14(d) depict the lambda values decrease after changing the position of the “bypass air valve” due to fuel rich. Figures 14(a)–14(c) illustrate that the three controllers, RVMMP, DLSSVMMP, and DRNNMPC, can regulate the lambda with obviously less deviation from the stoichiometric value and overshoot than that of the PI controller because the engine lambda model can be self-updated for any changes in engine condition. The control performances of the four controllers are shown in Table 5, and the control performance of the RVMMP in terms of *LTE* outperforms the DLSSVM, DRNNMPC, and PI controller by approximately 6%, 10%, and 62%, respectively. This promising result indicates that

TABLE 3: Control performance of pilot test 1 of different controllers.

Controller	<i>LTE</i>	Improvement in <i>LTE</i> (%)	Maximum overshoot	Improvement in maximum overshoot (%)
RVMMP	2.1328	—	0.0580	—
DLSSVMMP	2.0317	4.98 ^a	0.0720	19.44 ^a
DRNNMPC	1.9665	8.46 ^b	0.0890	34.83 ^b
PI	1.5485	37.73 ^c	0.1500	61.33 ^c

^a RVMMP over DLSSVMMP.^b RVMMP over DRNNMPC.^c RVMMP over PI.

TABLE 4: Control performance of pilot test 2 of different controllers.

Controller	<i>LTE</i>	Improvement in <i>LTE</i> (%)	Maximum overshoot	Improvement in maximum overshoot (%)
RVMMP	2.1699	—	0.0490	—
DLSSVMMP	2.1158	2.56 ^a	0.0570	14.04 ^a
DRNNMPC	2.0523	5.73 ^b	0.0650	24.62 ^b
PI	1.6092	34.84 ^c	0.1470	66.67 ^c

^a RVMMP over DLSSVMMP.^b RVMMP over DRNNMPC.^c RVMMP over PI.

TABLE 5: Control performance of pilot test 3 of different controllers.

Controller	<i>LTE</i>	Improvement in <i>LTE</i> (%)	Maximum overshoot	Improvement in maximum overshoot (%)
RVMMP	2.0042	—	0.0900	—
DLSSVMMP	1.8965	5.76 ^a	0.1070	15.89 ^a
DRNNMPC	1.8274	9.67 ^b	0.1090	17.43 ^b
PI	1.2386	61.81 ^c	0.2220	59.46 ^c

^a RVMMP over DLSSVMMP.^b RVMMP over DRNNMPC.^c RVMMP over PI.

the RVMMP can regulate air ratio very well even the engine ages and undergoes external disturbance simultaneously.

5.5. Discussion of Results. All the experimental results show that the overall lambda control performance of the RVMMP is better than those of DLSSVMMP, DRNNMPC, and conventional PI controller. There are two important factors affecting the control performance of the model predictive controllers which are the lambda model accuracy, and computational time. As presented in Sections 2.1 and 4.2, the sparseness, model accuracy and model updating time of RVM are better than those of DLSSVM and DRNN. Therefore, the model accuracy, and computational time of the RVMMP is the best among the three controllers. Overall, the RVMMP is the most suitable method for engine air-ratio control. Although Tables 3 to 5 show that the logarithmic tracking errors of the RVMMP have a small improvement only, and the lambda is a very delicate value, a small change can result in a big change of engine performance, so the *LTE* is not very large. For instance, the air

ratio is changed from 1 to 1.05; an increment of 0.05 results in changing from the best emission performance to the best brake-specific fuel consumption. Besides, [2, 3] stated that if AFR is 1% lower than its stoichiometric ratio, CO and HC emissions are significantly increased. An AFR that is 1% higher than the stoichiometric ratio produces more NO_x, up to 50%. As a result, the actual improvement achieved by the RVMMP is very significant.

6. Conclusions

This research is the first attempt at developing RVMMP for engine air-ratio control. This study is also the first to extend DLSSVM to MPC for engine air-ratio control. The RVM is trained by sequential learning algorithm and real engine data, which can let the engine lambda model update continually so as to maintain the lambda model accuracy for any changes in engine performance as the engine aging or fair user modification on it. With the highly accurate engine lambda model, the new MPC strategy can perform

lambda control effectively. The proposed intelligent control algorithm was successfully implemented and tested on a real automotive engine, whereas many previous researches were simulation tests only. Experimental results show that the lambda control performance of the RVMMPC is better than of DLSSVMMPC, DRNNMPC and conventional PI controller. Tables 3 to 5 reveal that the RVMMPC can effectively reduce the lambda deviation and overshoot from target lambda value up to 62% and 67%, respectively. Thus, the RVMMPC is a potential control scheme to replace conventional PI controller in the automotive ECU for engine lambda control. In the future, some advanced optimization algorithms for the RVMMPC and system stability will be studied.

Acknowledgments

This work is supported by the University of Macau Research Grant, Grant no. MYRG149(Y1-L2)-FST11-WPK. The authors would also like to thank the support from Mr. Adrian Kowalski.

References

- [1] W. H. Course and D. L. Anglin, *Automotive Mechanics*, McGraw-Hill, New York, NY, USA, 10th edition, 1993.
- [2] C. Manzie, M. Palaniswami, and H. Watson, "Gaussian networks for fuel injection control," *Journal of Automobile Engineering*, vol. 215, no. D8, pp. 1053–1068, 2001.
- [3] C. Manzie, M. Palaniswami, D. Ralph, H. Watson, and X. Yi, "Model predictive control of a fuel injection system with a radial basis function network observer," *Journal of Dynamic Systems, Measurement and Control*, vol. 124, no. 4, pp. 648–658, 2002.
- [4] S. W. Wang and D. L. Yu, "A new development of internal combustion engine air-fuel ratio control with second-order sliding mode," *Journal of Dynamic Systems, Measurement and Control*, vol. 129, no. 6, pp. 757–766, 2007.
- [5] Y. J. Zhai and D. L. Yu, "Radial-basis-function-based feedforward-feedback control for air-fuel ratio of spark ignition engines," *Proceedings of the Institution of Mechanical Engineers D*, vol. 222, no. 3, pp. 415–428, 2008.
- [6] S. W. Wang, D. L. Yu, J. B. Gomm, G. F. Page, and S. S. Douglas, "Adaptive neural network model based predictive control for air-fuel ratio of SI engines," *Engineering Applications of Artificial Intelligence*, vol. 19, no. 2, pp. 189–200, 2006.
- [7] Y. J. Zhai and D. L. Yu, "Neural network model-based automotive engine air/fuel ratio control and robustness evaluation," *Engineering Applications of Artificial Intelligence*, vol. 22, no. 2, pp. 171–180, 2009.
- [8] Y. J. Zhai, D. W. Yu, H. Y. Guo, and D. L. Yu, "Robust air/fuel ratio control with adaptive DRNN model and AD tuning," *Engineering Applications of Artificial Intelligence*, vol. 23, no. 2, pp. 283–289, 2010.
- [9] G. Y. Li, *Application of Intelligent Control and MATLAB to Electronically Controlled Engines*, Publishing House of Electronics Industry, 2007.
- [10] P. K. Wong, L. M. Tam, K. Li, and C. M. Vong, "Engine idle-speed system modelling and control optimization using artificial intelligence," *Proceedings of the Institution of Mechanical Engineers*, vol. 224, no. D1, pp. 55–72, 2010.
- [11] J. Suykens, J. De Brabanter, L. Lukas, and L. Vandewalle, *Least Squares Support Vector Machines*, World Scientific Press, 1st edition, 2002.
- [12] J. Vallyon and G. A. Horvath, "Sparse least squares support vector machine classifier," in *Proceedings of the IEEE International Joint Conference on Neural Networks*, vol. 1, pp. 543–548, Budapest, Hungary, July 2004.
- [13] C. M. Vong, P. K. Wong, and Y. P. Li, "Prediction of automotive engine power and torque using least squares support vector machines and Bayesian inference," *Engineering Applications of Artificial Intelligence*, vol. 19, no. 3, pp. 277–287, 2006.
- [14] P. K. Wong, C. M. Vong, L. M. Tam, and K. Li, "Data preprocessing and modelling of electronically-controlled automotive engine power performance using kernel principal components analysis and least squares support vector machines," *International Journal of Vehicle Systems Modelling and Testing*, vol. 3, no. 4, pp. 312–330, 2008.
- [15] T. Quan, X. Liu, and Q. Liu, "Weighted least squares support vector machine local region method for nonlinear time series prediction," *Applied Soft Computing Journal*, vol. 10, no. 2, pp. 562–566, 2010.
- [16] X. Wang, H. Zhang, C. Zhang, X. Cai, J. Wang, and J. Wang, "Prediction of chaotic time series using LS-SVM with automatic parameter selection," in *Proceedings of the 6th International Conference on Parallel and Distributed Computing, Applications and Technologies (PDCAT '05)*, pp. 962–965, Dalian, China, December 2005.
- [17] Z. Shi and M. Han, "Support vector echo-state machine for chaotic time-series prediction," *IEEE Transactions on Neural Networks*, vol. 18, no. 2, pp. 359–372, 2007.
- [18] P. K. Wong, H. C. Wong, and C. M. Vong, "Online time-sequence incremental and decremental least squares support vector machines for engine air-ratio prediction," *International Journal of Engine Research*, vol. 13, no. 1, pp. 28–40, 2011.
- [19] M. E. Tipping and A. C. Faul, "Fast marginal likelihood maximisation for sparse bayesian method," in *Proceedings of the 9th International Workshop on Artificial Intelligence and Statistics*, Key West, Fla, USA, January 2003.
- [20] Rototest Research Institute, "Performance measurements on chassis dynamometers," 2005, <http://www.rototest.com>.
- [21] A. C. Faul and M. E. Tipping, "Analysis of sparse Bayesian learning," *Advances in Neural Information Processing Systems*, vol. 14, pp. 383–389, 2002.
- [22] T. R. Chandrupatla, "An efficient quadratic fit—sectioning algorithm for minimization without derivatives," *Computer Methods in Applied Mechanics and Engineering*, vol. 152, no. 1–2, pp. 211–217, 1998.



Hindawi

Submit your manuscripts at
<http://www.hindawi.com>

