Engine idle-speed system modelling and control optimization using artificial intelligence

P K Wong[1*], **L M Tam**[1], **K Li**[1], and **C M Vong**[2]
[1]Department of Electromechanical Engineering, Faculty of Science and Technology, University of Macau, Taipa, Macao
[2]Department of Computer and Information Science, Faculty of Science and Technology, University of Macau, Taipa, Macao

The manuscript was received on 6 March 2009 and was accepted after revision for publication on 30 June 2009.

DOI: 10.1243/09544070JAUTO1196

Abstract: This paper proposes a novel modelling and optimization approach for steady state and transient performance tune-up of an engine at idle speed. In terms of modelling, Latin hypercube sampling and multiple-input and multiple-output (MIMO) least-squares support vector machines (LS-SVMs) are proposed to build an engine idle-speed model based on experimental sample data. Then, a genetic algorithm (GA) and particle swarm optimization (PSO) are applied to obtain an optimal electronic control unit setting automatically, under various user-defined constraints. All of the above techniques mentioned are artificial intelligence techniques. To illustrate the advantages of the MIMO LS-SVM, a traditional multilayer feedforward neural network (MFN) is also applied to build the engine idle-speed model. The modelling accuracies of the MIMO LS-SVM and MFN are also compared. This study shows that the predicted results using the estimated model from the LS-SVM are in good agreement with the actual test results. Moreover, both the GA and PSO optimization results show an impressive improvement on idle-speed performance in a test engine. The optimization results also indicate that PSO is more efficient than the GA in an idle-speed control optimization problem based on the LS-SVM model. As the proposed methodology is generic, it can be applied to different engine modelling and control optimization problems.

Keywords: idle-speed control, least-squares support vector machines, control optimization, genetic algorithm, particle swarm optimization

1 INTRODUCTION

Nowadays, the automotive engine is controlled by the electronic control unit (ECU), and the engine performance at idle speed is significantly affected by the set-up of control parameters in the ECU. In modern spark ignition engines, an efficient idle-speed performance is required to fulfil the ever-increasing requirements on fuel consumption, vehicle driveability, and pollutant emissions. Basically, the idle-speed control problem is a compromise among low engine speed for fuel saving, minimum emissions, and disturbance rejection ability [**1**]. From the control point of view, the primary difficulty with the idle-speed control (ISC) problem is that the engine at idle is subject to step disturbances from unknown external loads and accessory loads such as air-conditioning or power steering loads, etc. These disturbances decrease engine speed rapidly and therefore must be rejected.

Currently, the engine idle-speed control parameters in the ECU for production cars are almost formulated in control maps (look-up tables). There are many maps around the target idle speed for the engineer to set, such as fuel and ignition maps. Current practice of engine idle performance tune-up relies on the experience of the automotive engineer who handles a huge number of combinations of engine control parameters. Moreover, engine idle-speed tune-up is done empirically through testes on the dynamometer (dyno) [**2**]. As a result, a lot of time

*Corresponding author: Department of Electromechanical Engineering, Faculty of Science and Technology, University of Macau, Taipa, Macao.
email: fstpkw@umac.mo

and resources are consumed, while the optimal parameters may not be obtained.

Recently, researches have described the use of some advanced idle-speed controllers, such as an online proportional–integral–derivative (PID) tuning controller [**3**], adaptive control algorithm [**4**], model-based control algorithm [**5**], and robust control algorithm [**6**, **7**], to regulate the air control valve (bypass air valve (BPAV) or electronic throttle) to achieve a satisfactory idle-speed response. Nevertheless, the limitations and problems of these advanced controllers are as follows.

1. No matter how advanced the algorithms are, the development of the control systems must call for exact engine model and base system parameters for system simulation and dynamic analysis. An exact engine idle-speed model is also important for model-based control algorithms. However, a modern automotive engine model is a complex multivariable non-linear function, which is very difficult to be determined. Therefore the models used in these sophisticated controllers are mostly empirical models, which are derived from resorting to some simple physics laws combined with identification procedures for estimation of several unknown parameters [**8**]. This kind of engine model is quite simple as compared with the real engine system [**4**] and only suitable for several models of vehicle engines. In fact, many coefficients in the empirical models are also difficult to determine. Therefore the empirical models cannot reflect the actual performance of the controller and cannot let the engineer truly optimize the controller parameters.
2. The control variables and control objectives of these intelligent controllers are incomprehensible. In the engine idle-speed system, ignition timing, air control valve opening area or duty cycle, and fuel injection time are significant factors affecting the internally developed torque and engine speed. However, since too many assumptions are made to simplify the empirical models, most of the sophisticated controllers mentioned above only consider the air control valve and/or ignition advance as direct control parameters [**3**, **5**–**8**]. Very little research considers the fuel injection time, because all of these advanced controllers assume that the base fuel map is perfectly fine-tuned, and the fuel injection time is adjusted according to the manifold air pressure, which is indirectly controlled by the air control valve. In fact, a base fuel map is quite important to engine idle-speed stability, fuel consumption, and emission quality, but it is very difficult to get a perfect base map. Moreover, most of the idle-speed control objectives in the available literatures only focus on an idle-speed response subject to external disturbances [**3**, **4**, **6**–**8**]; there is no comprehensive discussion about fuel consumption and emission quality together, but these two factors are very important to the ISC problem as well.
3. These intelligent controllers are still under investigation. Therefore, look-up tables together with a typical PID controller are still popular today when considering gasoline engine idle-speed control [**9**].

In view of the above limitations of ISC map calibration and advanced idle-speed controllers, development of a comprehensive idle-speed system modelling and control optimization method can contribute to existing engine ISC system tune-up and its controller development. The objective of this research is not to design an advanced idle-speed controller but to develop a comprehensive and reliable engine modelling methodology, such that the development of many intelligent idle-speed controllers can couple with any tailor-made and practical engine models built from this methodology for controller design and simulation. In terms of ISC system tune-up for current automobiles, by integrating proper computer aided optimization methods with the engine models built, an optimal ECU set-up can also be easily determined. The car engine is only required to go through a dyno test for verification after obtaining a satisfactory set-up from the integrated approach. Hence, the number of unnecessary dyno tests for the trial set-up can be drastically reduced in order to save a large amount of time and money for testing.

According to the above objectives, modelling of the engine idle-speed system is an important part of this research. To model a complicated system exactly, whose domain information is insufficient, black-box identification techniques are usually employed. These techniques can quickly derive models from experimental data [**10**]. The most common black-box modelling technique for automotive engine modelling uses neural networks. It is well known that a neural network is a universal estimator. Recent researchers have described the use of neural networks for modelling some engine performances [**11**, **12**] based on experimental data. There are, in general, however, two main drawbacks for neural networks [**13**].

1. The architecture, including the number of hidden neurons, has to be determined a priori or

modified while training by heuristics, which results in a non-necessarily optimal network structure.

2. The training process (i.e. the minimization of the residual squared error cost function) in neural networks can easily become stuck by local minima. Various ways of preventing local minima, like early stopping, weight decay, etc., are employed. However, those methods greatly affect the generalization of the estimated function, i.e. the capacity to handle new input cases.

With an emerging artificial intelligence technique of least-squares support vector machines (LS-SVMs) [**14**], combining the advantages of neural networks (handling large amount of highly non-linear data) and non-linear regression (high generalization), the previous drawbacks from neural networks are overcome. The main advantages of LS-SVM over neural networks are good generalization, a guarantee of a global solution having a minimal fitting error, and the fact that the architecture of the model must not be determined before training [**15**].

In view of the advantages of LS-SVMs, this paper proposes to use this novel approach to model the multivariable engine ISC system. The model then serves as an objective function for optimization.

2 PROPOSED MODELLING AND OPTIMIZATION FRAMEWORK

A schematic illustration of the framework and overall methodology is shown in Fig. 1. The upper branch in Fig. 1 shows the steps required to build the LS-SVM model. The experiments are still required, but only to provide sufficient data for LS-SVM training. The design of experiments is used additionally to streamline the process of creating representative sampling data points to train the model. Once the engine idle-speed performance model obtained is evaluated, it is then possible to use a computer aided technique to search for the best engine control parameters automatically based on the model, if the application is required. As the model derived by an LS-SVM is difficult to differentiate or is even non-differentiable, gradient information of this model cannot be easily obtained. Practically, direct search methods are employed to determine the suboptimal solution for this kind of model, because they do not require any gradient information. A genetic algorithm (GA) and particle swarm optimization (PSO) are two widely used direct search techniques, so both optimization algorithms are proposed for this constrained multivariable optimization problem. The optimal set points by both optimizers are then sent back to

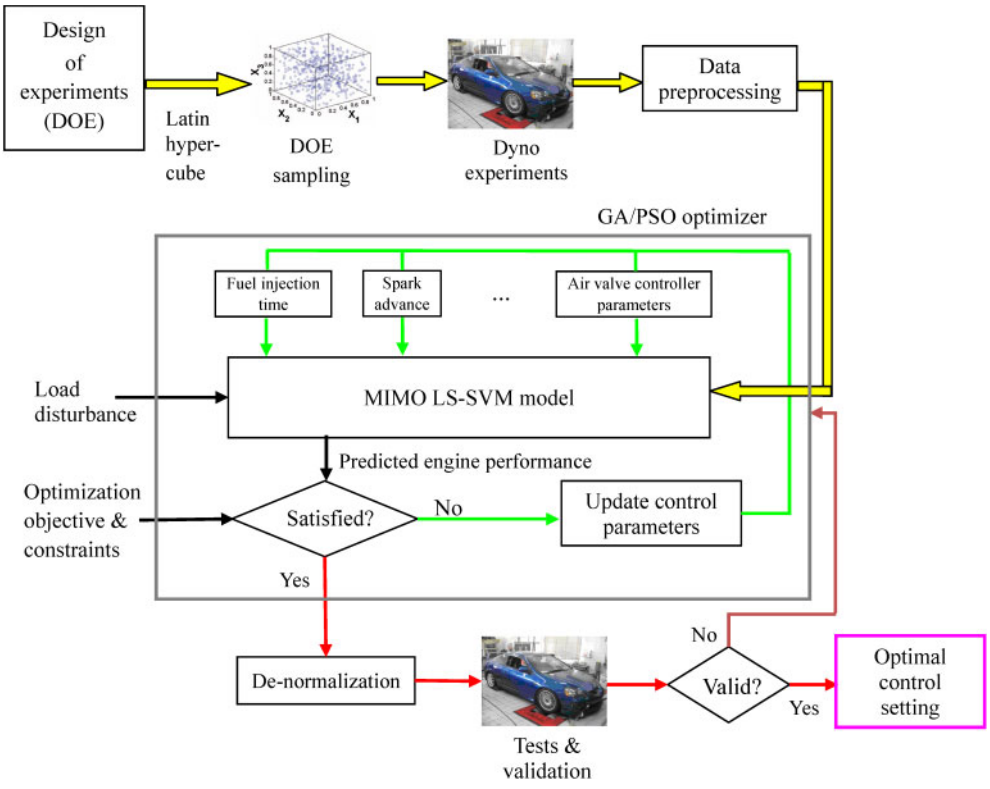

Fig. 1 Framework for optimization of engine idle-speed control parameters

the ECU to carry out evaluation tests and a comparison. The feasibility and efficiency of the two optimization approaches can then be examined. All of the techniques mentioned above are artificial intelligence techniques.

The purpose of this paper is to demonstrate the effectiveness of the proposed LS-SVM + GA/PSO methods on engine ISC systems. It is important to note that there are no apparent limits or constraints in the number of input–output variables, idle-speed controller types, and the formulation of the optimization objective function. Hence, the methodology is generic, and its effectiveness is demonstrated in later sections through a case study of the ISC optimization problem. It is believed that the proposed methods can be applied to different engine modelling and control optimization problems.

3 IDLE-SPEED MODEL IDENTIFICATION

3.1 LS-SVM formulation for multivariable function estimation

The classical LS-SVM modelling algorithm is only a multi-input but single-output modelling method. However, the practical engine idle-speed system modelling is a multi-input/multi-output (MIMO) modelling problem. Hence, a new MIMO LS-SVM modelling algorithm based on classical LS-SVM is proposed in this paper. The concept is presented below.

Consider a training dataset, $\mathbf{D} = \{(\boldsymbol{x}_k, \boldsymbol{y}_k)\}_{k=1}^{N}$, with N data points, where $\boldsymbol{x}_k \in R^n$ represents the kth engine set-up and $\boldsymbol{x}_k$ is an n-dimension system input vector, and $\boldsymbol{y}_k \in \mathrm{R}^m$ is the kth engine output based on $\boldsymbol{x}_k$, $k = 1$ to N. Here, $\boldsymbol{y}_k$ is an m-dimension system output vector, $\boldsymbol{y}_k = [y_{k,1},\ldots,y_{k,m}]$. For example, $y_{k,1}$ could be the minimum idle speed and $y_{k,m}$ could be the fuel consumption. For the automotive engine, each output performance in the dataset $\boldsymbol{y}_k$ is usually an individual variable and able to be measured separately, so the training dataset $\mathbf{D}$ can be arranged as

$$\mathbf{D} = \{\mathbf{d}_1, \ldots, \mathbf{d}_h, \ldots, \mathbf{d}_m\} \tag{1}$$

where $\mathbf{d}_h = \{(\boldsymbol{x}_k, \boldsymbol{y}_{k,h})\}_{k=1}^{N}$; $h \in [1,m]$. In this case, for each single output dimension in $\boldsymbol{y}_k$, it forms a new training dataset $\mathbf{d}_h$. Consequently, the MIMO training dataset $\mathbf{D}$ is separated into m multi-input but single-output subtraining datasets $\mathbf{d}_h$. For each multi-input single-output dataset $\mathbf{d}_h$, LS-SVM deals with the following optimization problem in the primal weight space

$$\begin{bmatrix} \min\limits_{\boldsymbol{w}_h, b_h, \boldsymbol{e}_h} J_P(\boldsymbol{w}, \boldsymbol{e}_h) = \frac{1}{2}\boldsymbol{w}_h^{\mathrm{T}}\boldsymbol{w}_h + \gamma_h \frac{1}{2}\sum\limits_{k=1}^{n} e_{k,h}^2 \\ \text{such that } e_{k,h} = y_{k,h} - [\boldsymbol{w}_h^{\mathrm{T}}\phi(\boldsymbol{x}_k) + b_h], \quad k = 1, \ldots, N \end{bmatrix} \tag{2}$$

where $\gamma_h \in R$ is a scalar for the regularization factor (which is a hyper-parameter for tuning), $\boldsymbol{w}_h \in R^{n_h}$ is the weight vector of the target function, $\boldsymbol{e}_h = [e_{1,h},\ldots, e_{N,h}]$ is the residual vector, and $\phi : R^n \rightarrow R^{n_h}$ is a non-linear mapping. The estimated model $\boldsymbol{M}_h(\boldsymbol{x})$ is considered as

$$\boldsymbol{M}_h(\boldsymbol{x}) = \boldsymbol{w}_h^{\mathrm{T}}\phi(\boldsymbol{x}) + b_h \tag{3}$$

However, $\boldsymbol{w}_h$ may be in very high or even infinite dimensions that cannot be solved directly. In order to resolve the problem, the Lagrangian of equation (2) is constructed to derive the dual problem and the Lagrangian is as follows

$$\begin{aligned} &L_h(\boldsymbol{w}_h, b_h, \boldsymbol{e}_h, \boldsymbol{\alpha}_h) \\ &\quad = J_p(\boldsymbol{w}_h, \boldsymbol{e}_h) - \sum_{k=1}^{n} \alpha_{k,h}\{\boldsymbol{w}_h^{\mathrm{T}}\phi(\boldsymbol{x}_k) + b_h + e_{k,h} - y_{k,h}\} \end{aligned} \tag{4}$$

where $\alpha_{k,h} \in \boldsymbol{\alpha}_h$ are Lagrange multipliers. The conditions for optimality are given by

$$\begin{aligned} &\frac{\partial L_h}{\partial \boldsymbol{w}_h} = 0 \rightarrow \boldsymbol{w}_h = \sum_{k=1}^{n} \alpha_{k,h}\phi(\boldsymbol{x}_k) \\ &\frac{\partial L_h}{\partial b_h} = 0 \rightarrow \sum_{k=1}^{n} \alpha_{k,h} = 0 \\ &\frac{\partial L_h}{\partial \boldsymbol{e}_h} = 0 \rightarrow \alpha_{k,h} = \gamma_h e_{k,h} \\ &\frac{\partial L_h}{\partial \boldsymbol{\alpha}_h} = 0 \rightarrow \boldsymbol{w}_h^{\mathrm{T}}\phi(\boldsymbol{x}_k) + b_h + e_{k,h} - y_{k,h} = 0, \quad k = 1,\ldots,N \end{aligned} \tag{5}$$

After elimination of the variables $\boldsymbol{w}_h$ and $\boldsymbol{e}_h$ in equation (4) using the results from equation (5), the LS-SVM dual formulation of the non-linear function estimation is then expressed as follows [**14**]

$$\begin{bmatrix} \text{Solve in } \boldsymbol{\alpha}_h, b_h : \\ \begin{bmatrix} 0 & 1_v^{\mathrm{T}} \\ 1_v & \boldsymbol{\Omega} + \frac{1}{\gamma_h}\mathbf{I}_N \end{bmatrix} \begin{bmatrix} b_h \\ \boldsymbol{\alpha}_h \end{bmatrix} = \begin{bmatrix} 0 \\ \boldsymbol{y}_h \end{bmatrix} \end{bmatrix} \tag{6}$$

where $\boldsymbol{y}_h = [y_{1,h},\ldots, y_{N,h}]^{\mathrm{T}}$ and $\boldsymbol{\alpha}_h = [\alpha_{1,h},\ldots, \alpha_{N,h}]^{\mathrm{T}}$. The kernel trick is employed as follows

$$\begin{aligned}\boldsymbol{\Omega}_{k,l} &= \phi(\boldsymbol{x}_k)^{\mathrm{T}}\phi(\boldsymbol{x}_l) \\ &= K(\boldsymbol{x}_k,\boldsymbol{x}_l), \quad k,l=1,\ldots,N\end{aligned} \tag{7}$$

The resulting LS-SVM model for the function estimation is constructed by substituting the results of equation (5), i.e. $\boldsymbol{w}_h = \sum_{k=1}^{N} \alpha_{k,h}\phi(\boldsymbol{x}_k)$, into equation (3) and becomes

$$\begin{aligned}\boldsymbol{M}_h(\boldsymbol{x}) &= \sum_{k=1}^{N} \alpha_{k,h}\phi(\boldsymbol{x}_k)^{\mathrm{T}}\phi(\boldsymbol{x}) + b_h \\ &= \sum_{k=1}^{N} \alpha_{k,h}K(\boldsymbol{x}_k,\boldsymbol{x}) + b_h \\ &= \sum_{k=1}^{N} \alpha_{k,h}\exp\left(-\frac{\|\boldsymbol{x}_k-\boldsymbol{x}\|^2}{\sigma_h^2}\right) + b_h\end{aligned} \tag{8}$$

where $\alpha_{k,h}$, $b_h \in R$ are the solutions of equation (6), $\boldsymbol{x}_k$ is the kth engine set-up in the training dataset $\mathbf{d}_h$, $\boldsymbol{x}$ is the new input set-up for the engine idle performance prediction, and the radial basis function is chosen as the kernel function K, which is the common choice for modelling. In the radial basis function, σ_h specifies the kernel sample variance, which is also a hyperparameter for tuning, and $\|\cdot\|$ means Euclidean distance. After inferring m pairs of hyperparameters (γ_h, σ_h) by a well-known technique, Bayesian inference [**14**], m individual training datasets are used for calculating m individual sets of support vectors $\alpha_{k,h}$ and threshold values b_h. Finally, m individual sets of $\boldsymbol{M}_h(\boldsymbol{x})$ can be constructed based on equation (8). The whole MIMO modelling algorithm is shown in Fig. 2.

In Fig. 2, a set of LS-SVM models are generated to predict the engine response under different combinations of control variables. Each model represents one engine output performance, which is included in the objective function for optimization.

3.2 Experimental set-up and data sampling for the case study

To demonstrate the effectiveness of the whole modelling and optimization approach, an optimization of the engine idle-speed controller together with the ECU set-up was selected as a case study. The test car used in the case study was a Honda Integra DC5 Type-R with a K20A DOHC i-VTEC engine. The manifold absolute pressure (MAP) of the test engine at idle was controlled by a BPAV. Although the use of an electronic throttle becomes more and more prevalent, the method is equally applicable to both the electronic throttle and BPAV.

Taking the demonstration purpose into account, an aftermarket programmable ECU, MoTeC M800, was selected as the calibration and optimization test bed. The fuel injection time, valve timing and ignition control signals were stored in its look-up tables. Moreover, a PID controller using the engine idle speed as the feedback signal was applied to the BPAV due to its popularity among modern spark ignition automobiles. The control scheme of MoTeC M800 is shown in Table 1.

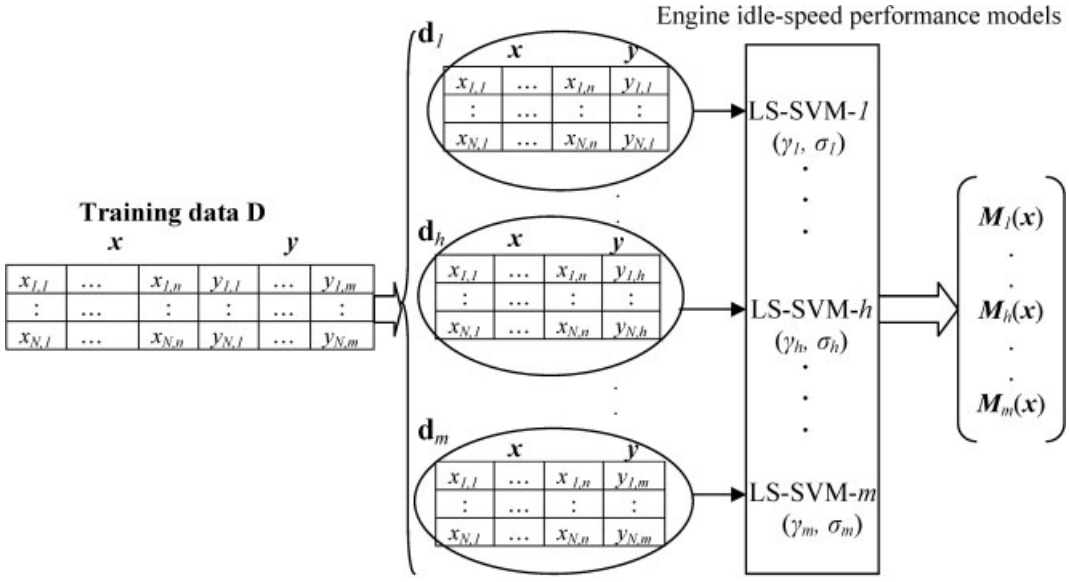

Fig. 2 MIMO LS-SVM modelling framework

Table 1 Control scheme of MoTeC M800

Control variable	Type of control scheme
Fuel injection time	Open loop 3-D look-up table
Valve timing	Open loop 2-D look-up table
Ignition advance	Open loop 3-D look-up table
MAP	PID BPAV controller

To check the robustness of the control settings, the test car was loaded on a chassis dyno (Fig. 3) and a constant step load provided by the dyno was applied to the engine. Then the engine speed and lambda (λ) variations were measured. The engine idle performance data were also recorded by the ECU at a logging rate of 20 Hz. Figure 4 shows the constant step load applied by the dyno in the course of data sampling. The magnitude of the constant load was measured from the total accessory loads of the test car including the electrical, air-conditioning, and power-steering loads. Because of a high-speed cam used in the test engine, the aimed engine idle speed was set to be 1200 r/min in the case study. It is well known that setting a stable idle speed for a high-speed cam is a challenging job.

In this case study, the following engine idle-speed system parameters were selected to be the input and output variables of the MIMO LS-SVM model

$$\boldsymbol{x} = <\boldsymbol{F}_{i,j}, \boldsymbol{I}_{i,j}, \boldsymbol{V}_j, Pro, Int, Der, Nor, L>$$
$$\boldsymbol{y} = <IAE_R, IAE_\lambda, \sum F, R_{\min}, T_{\text{rise}}> \tag{9}$$

There are two types of variables in equation (9): input variables and output variables. It was assumed that the input variables are

$\boldsymbol{F}_{i,j}$ = fuel injection time at the corresponding MAP i and idle speed j (ms, i = [20, 30, 40, 50], j = [500, 1000, 1500])

$\boldsymbol{I}_{i,j}$ = ignition advance at the corresponding MAP i and idle speed j (degree before top dead centre (BTDC), i = [20, 30, 40, 50], j = [500, 1000, 1500])

Fig. 3 Experimental set-up for data sampling and programmable ECU

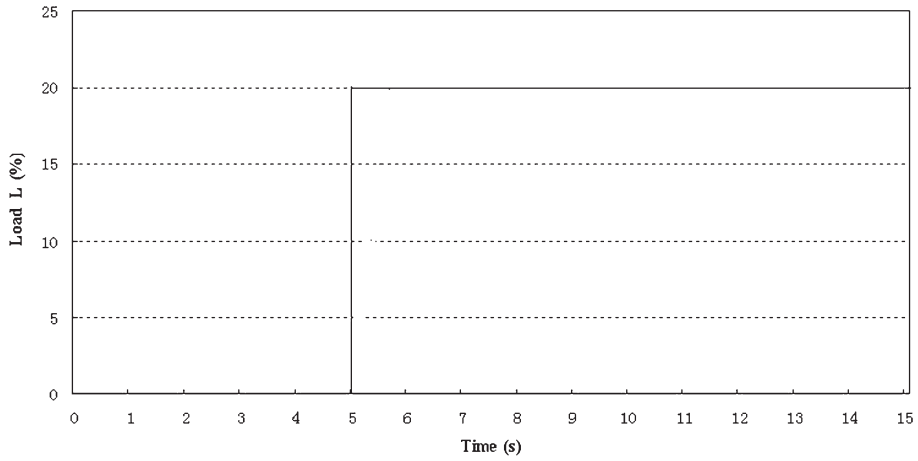

Fig. 4 Constant step load applied by the chassis dyno in the case study

$\boldsymbol{V}_j$ = intake valve open timing at the corresponding idle speed j (degree BTDC, j = [500, 1000, 1500])
Pro = proportional gain of the idle air valve controller
Int = integral gain of the idle air valve controller
Der = derivative gain of the idle air valve controller
Nor = normal position of the idle air valve (percentage of wide open)
L = constant step load applied to the engine (percentage of the dyno full load)

It was also assumed that the output variables are:

IAE_R = integral absolute error of the engine idle speed (r/min), which is calculated by

$$IAE_R = \sum_{t=0}^{t_f} |R_t - R_{\text{aim}}| \tag{10}$$

where, for this case study, $t_f = 15$ s and $R_{\text{aim}} = 1200$ r/min.
IAE_λ = integral absolute error of lambda

$$IAE_\lambda = \sum_{t=0}^{t_f} |\lambda_t - \lambda_{\text{aim}}| \tag{11}$$

where, for this case study, $\lambda_{\text{aim}} = 1$.
ΣF = overall fuel consumption (ms), where the fuel consumption is proportional to the fuel injection time, and hence the overall fuel consumption can be estimated by the summation of fuel injection from $t = 0$ to $t_f = 15$ s with sampling time = 0.05 s

$R_{\min}$ = minimum idle speed to which the engine falls when a step load is applied (r/min)
T_{rise} = recovery time to aimed speed when a step load is applied (s)

A design of experiment technique, Latin hypercube sampling (LHS) [**16**], was employed to choose a representative set of operating points for generating training samples. A total of 200 sets of representative combinations of input variables were selected and downloaded to the ECU to produce 200 sets of output performance data. Figures 5 to 7 show an example of output performance of $\boldsymbol{D}_{\text{best}}$, which is the best performance among the 200 sample datasets. The ECU set points of $\boldsymbol{D}_{\text{best}}$ are shown in Table 2.

In order to have a fair comparison with the engine idle performances under different input set-ups, all the engine training data were recorded 5 seconds before the load was applied and 10 seconds after that point. Figure 5 shows the idle-speed regulation performance of $\boldsymbol{D}_{\text{best}}$. When the load is applied, the engine speed first falls to 602 r/min and then takes 1.05 s to recover. The integral absolute error, which represents the idle-speed regulation ability, in the 15 s test period is 6192.36 r/min. Meanwhile, the engine lambda value, as shown in Fig. 6, rises first, which is due to the fact that if the engine speed suddenly drops, the fuel injection time also drops accordingly (Fig. 7). When the BPAV controller starts to take action, it tends to open widely to increase the MAP, resulting in an increase in the amount of fuel injected into the intake manifold. In this way, more

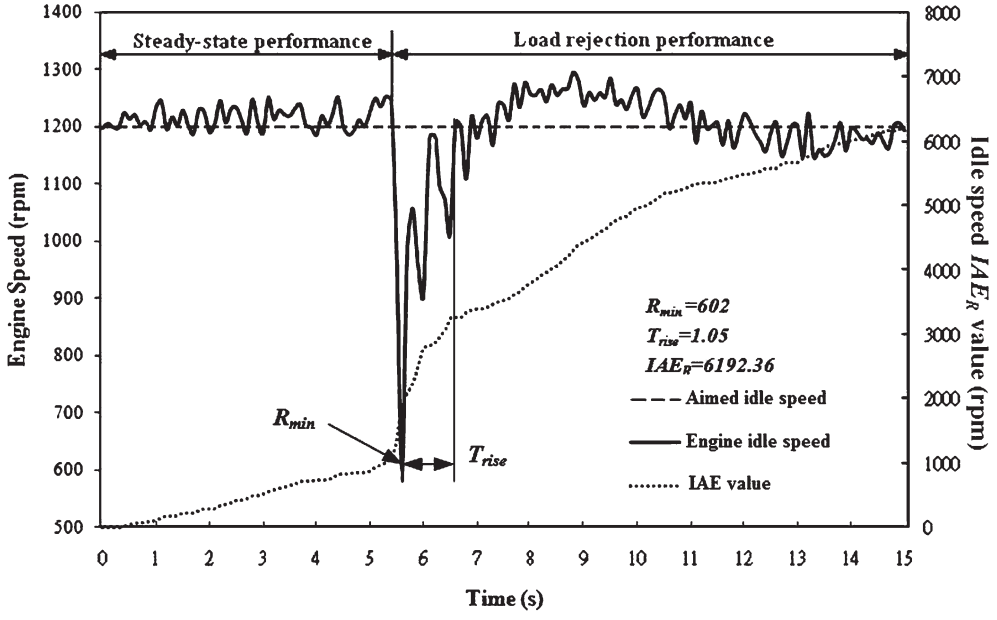

Fig. 5. Idle-speed regulation performance in $\boldsymbol{D}_{\text{best}}$

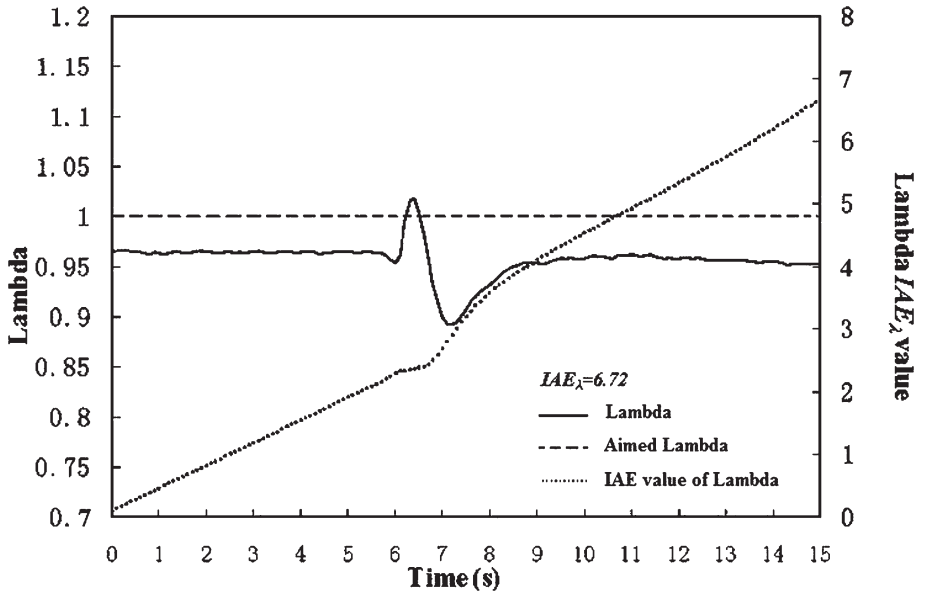

Fig. 6 Lambda performance in D_{best}

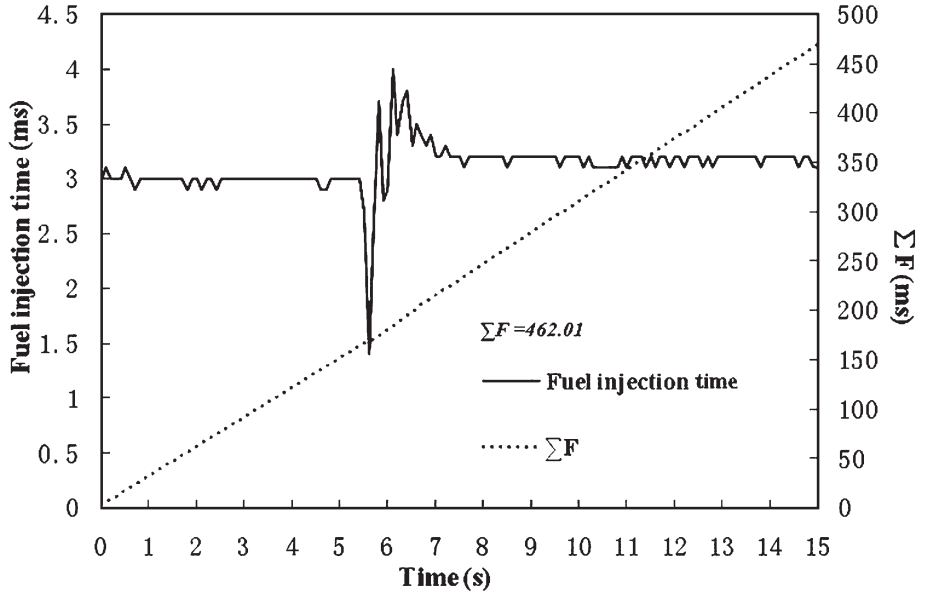

Fig. 7 Fuel consumption in D_{best}

Table 2 ECU set points of D_{best}

Engine speed (r/min)	$F_{i,j}$ MAP (kPa) 20	30	40	50	$I_{i,j}$ MAP (kPa) 20	30	40	50	V_j
500	3.04	3.24	3.36	4.02	7.9	8.6	9.4	10	2
1000	2.89	3.27	3.39	4.06	14.5	15	15.7	16	6
1500	3.04	3.31	3.36	4.09	15.9	16.8	17.3	18	8
BPAV controller parameters		*Pro* 2.24		*Int* 0.87		*Der* 1.32		*Nor* 40.24	

air–fuel mixture can be breathed into the combustion chamber to generate more torque in order to counteract the load and regain the aimed speed. This is noted that there is a time delay between the fuel injection and the lambda value. This is because the lambda value is measured by an oxygen sensor in the exhaust pipe and the value can only represent the stoichiometric ratio in the previous combustion cycle.

3.3 Application of the MIMO LS-SVM and modelling results

In the current application, $\boldsymbol{M}_h(\boldsymbol{x})$, $h = 1, 2, \ldots, 5$, in equation (8) stands for the performance functions of IAE_R, IAE_λ, ΣF, $R_{\min}$, T_{rise} respectively. After collection of the sample dataset $\mathbf{D}$, for every data subset $\mathbf{d}_h \in \mathbf{D}$, it was randomly divided into two sets: TRAIN_h for training and TEST_h for testing, where TRAIN_h contains 80 per cent of $\mathbf{d}_h$ and TEST_h holds the remaining 20 per cent. Then TRAIN_h was sent to the LS-SVM module for training, which had been implemented using LS-SVMlab 1.5 [**17**], a Matlab toolbox under MS Windows XP.

In order to have a more accurate modelling result, the input data of the training dataset is conventionally normalized before training [**18**]. This prevents any input parameter from dominating the output value. For all input values, it is necessary to be normalized within the range [0, 1], i.e. unit variance, through the following equation

$$v^* = \frac{v - v_{\min}}{v_{\max} - v_{\min}} \tag{12}$$

For example, $v \in [2, 35]$, $v_{\min} = 2$, and $v_{\max} = 35$. The limits for each input control parameter are determined via a number of experiments, expert knowledge, and manufacturer data sheets. After obtaining the optimal setting, each set point should go through a denormalization using the inverse of equation (12) in order to obtain the actual value v. The process flow of the normalization and denormalization is shown in Fig. 1.

To verify the accuracy of each function of $\boldsymbol{M}_h(\boldsymbol{x})$, an error function is proposed. For a certain function $\boldsymbol{M}_h(\boldsymbol{x})$, the corresponding validation error is

$$E_h = \sqrt{\frac{1}{N}\sum_{k=1}^{N}\left[\frac{y_{k,h} - \boldsymbol{M}_h(\boldsymbol{x}_k)}{y_{k,h}}\right]^2} \tag{13}$$

where $\boldsymbol{x}_k$ is the engine control parameters of the kth data point in TEST_h, $y_{k,h}$ is the actual performance output value in the data point $\boldsymbol{d}_k$ ($\boldsymbol{d}_k(\boldsymbol{x}_k,\ y_{k,h})$ represents the kth data point in $\mathbf{d}_h$), and N is the number of data points in the test set. The error E_h is a root-mean-square of the difference between the true value $y_{k,h}$ and its corresponding estimated value from $\boldsymbol{M}_h(\boldsymbol{x}_k)$. The difference is also divided by $y_{k,h}$, so that the result is normalized within the range [0, 1]. Hence, the accuracy rate for each output function $\boldsymbol{M}_h(\boldsymbol{x})$ is calculated using the following formula

$$\text{Accuracy}_h = (1 - E_h) \times 100\ \% \tag{14}$$

All the output functions were evaluated one by one against their own test sets TEST_h using equation (14). Table 3 shows that the predicted results are in good agreement with the actual experiment results under their hyperparameters (γ_h, σ_h) inferred using Bayesian inference. Hence, the idle-speed system model that was built is reliable and can be used for optimization. However, it is believed that the model accuracy could be improved by increasing the number of training data.

3.4 Application of neural network and modelling results

To illustrate the advantages of MIMO LS-SVM regression, the modelling results were compared with those obtained from training a multilayer feed forward neural network (MFN) with back-propagation. Since the MFN is similar to the MIMO LS-SVM and is also a well-known universal estimator, the results from the MFN can be considered as a rather standard benchmark.

Table 3 Accuracy of different output functions $\boldsymbol{M}_h(\boldsymbol{x})$ and its corresponding hyperparameters

Engine output function $\boldsymbol{M}_h(\boldsymbol{x})$	γ_h	σ_h	Training error with TRAIN_h (%)	Average accuracy with TEST_h (%)
$\boldsymbol{M}_1(\boldsymbol{x})$	2796.40	70.04	1.75	95.72
$\boldsymbol{M}_2(\boldsymbol{x})$	190.53	53.54	2.43	96.14
$\boldsymbol{M}_3(\boldsymbol{x})$	1546.34	1264.26	6.45	94.52
$\boldsymbol{M}_4(\boldsymbol{x})$	2426.72	61.86	1.34	95.41
$\boldsymbol{M}_5(\boldsymbol{x})$	3349.90	44.69	3.65	93.52
Overall average			3.12	95.06

A neural network with five output neurons $NET_h = \{NET_1, NET_2, NET_3, NET_4, NET_5\}$, which respectively represent the performance functions of IAE_R, IAE_λ, ΣF, $R_{\min}$, T_{rise}, was built based on the same training dataset. The neural network consists of 32 input neurons (i.e. the input parameters $\boldsymbol{x}$), 5 output neurons, and 50 hidden neurons. Normally, 50 hidden neurons can provide enough capability to approximate a highly non-linear function. The activation function used inside the hidden neurons was the Tan-Sigmoid transfer function, while for the output neurons, a pure linear filter was employed (Fig. 8).

The training method employed the standard back-propagation algorithm (i.e. a gradient descent towards the negative direction of the gradient), so that the results of the MFN can be considered as a standard. The learning rate of the weight update was set to be 0.05. The same test sets were also chosen so that the accuracies of the engine idle-speed performance functions built by the MIMO LS-SVM and MFN can be compared reasonably. The training results of the neural network are shown in Table 4. The average accuracy of each output shown in Table 4 is calculated by using equation (14).

3.5 Comparison of modelling results

With reference to Tables 3 and 4, the MIMO LS-SVM outperforms the MFN by about 11.40 per cent in overall average accuracy under the same test sets. In the MIMO LS-SVM, two sets of hyperparameters (γ_h, σ_h) are required. These hyperparameters can be inferred automatically by using Bayesian inference, which totally eliminates the user burden. In the MFN, the learning rate and number of hidden neurons are required from the users. Surely, these parameters can also be solved by 10-fold cross-validation. However, the users have to prepare a grid of guessed values for these parameters, and the grid may not cover the best values for the parameters. Therefore, the MIMO LS-SVM could often produce a better generalization rate over the MFN, as indicated in Tables 3 and 4. The MFN produces less training errors than the MIMO LS-SVM because there is no regularization factor controlling the tradeoff between training error and generalization. In contrast, the MIMO LS-SVM produces better generalization due to the regularization factor γ_h introduced in the training error function.

4 IDLE-SPEED CONTROL OPTIMIZATION

After obtaining the idle-speed model, it is then possible to use the GA and PSO to search the best engine ISC set-up automatically. In this application, the engine set-up involves 31 variables, so it is a large-scale and challenging optimization problem. Nowadays, the GA has become a well-known technique for

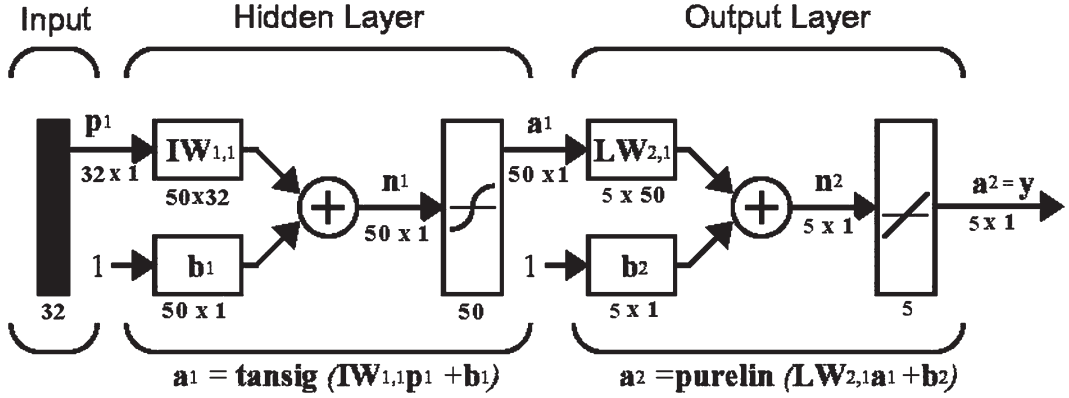

Fig. 8 Architecture of the MFN

Table 4 Accuracy of different output functions NET_h

Engine output function $M_h(\boldsymbol{x})$	Training error with TRAIN_h (%)	Average accuracy with TEST_h (%)
$\boldsymbol{M}_1(\boldsymbol{x}) = NET_1(\boldsymbol{x})$	0.16	86.43
$\boldsymbol{M}_2(\boldsymbol{x}) = NET_2(\boldsymbol{x})$	0.24	84.35
$\boldsymbol{M}_3(\boldsymbol{x}) = NET_3(\boldsymbol{x})$	0.65	80.37
$\boldsymbol{M}_4(\boldsymbol{x}) = NET_4(\boldsymbol{x})$	0.82	81.72
$\boldsymbol{M}_5(\boldsymbol{x}) = NET_5(\boldsymbol{x})$	0.03	85.44
Overall average	0.38	83.66

solving many engineering optimization problems. Moreover, it has been proved effective in solving both constrained and un-constrained objective functions with discontinuous, non-differentiable, stochastic, and highly non-linear, etc., properties [**19**, **20**]. On the other hand, PSO is a relatively new algorithm proposed by Kennedy and Eberhart in 1995 [**21**]. As a form of swarm intelligence, PSO is a population-based stochastic optimization technique inspired by the social behaviour of bird flocking or fish schooling. Basically, PSO is also an evolutionary computation technique similar to the GA. However, it has no evolutionary operators of crossover and mutation as compared with the GA. From the user point of view, PSO can reduce the time and the burden for users trying different evolutionary operators to find the optimal solution.

Since both the GA and PSO have their own pros and cons, these two optimization methods are studied in this paper. However, direct search methods can easily suffer from a local optimum. In order to minimize the chance of obtaining a local optimum, different and arbitrarily generated initial populations are used. To achieve such a purpose, 10 sets of independent optimization runs were carried out for each method. The optimization results of both optimization algorithms are also compared at the end of this section.

4.1 Objective function for engine ISC optimization

An objective function was designed to evaluate the idle performance under different control set-ups. In the case of engine ISC optimization, a complete evaluation function should encompass the following [**22**].

1. Idle-speed regulation. The engine idle speed must be capable of maintaining close to the target point with deviation as low as possible. Essentially, the better the idle-speed regulation and disturbance rejection ability, the lower the aimed idle speed can be set.
2. Robustness of load disturbance. In a vehicle, the disturbance is due to electrical loads (e.g. switching on or off the air-conditioning, window heating, light, etc.), power steering, and low-speed manoeuvring. These are events that, when they occur, may cause the engine to stall.
3. Fuel economy and emissions. On average, vehicles consume about 30 per cent of their fuel in idling during city driving [**23**], so minimization of fuel consumption and pollutant emissions at idle speed is very important.

Hence a well-considered objective function, f_{obj}, is formulated in the following equation, where it is shown that the larger the fitness value calculated by the objective function, the better the engine idle-speed performance

$$\begin{aligned} f_{\text{obj}} &= \text{Max}[\text{Fitness}] \\ &= \text{Max}[-w_{IAE_R}\tan^{-1}(\boldsymbol{M}_1(\boldsymbol{x})) \\ &\quad - w_{IAE_\lambda}\tan^{-1}(\boldsymbol{M}_2(\boldsymbol{x})) - w_{\sum F}\tan^{-1}(\boldsymbol{M}_3(\boldsymbol{x})) \\ &\quad + w_{R_{\min}}\tan^{-1}(\boldsymbol{M}_4(\boldsymbol{x})) - w_{T_{\text{rise}}}\tan^{-1}(\boldsymbol{M}_5(\boldsymbol{x}))] \end{aligned} \tag{15}$$

Subject to

$$2 \leqq \boldsymbol{F}_{i,j} \leqq 5 \text{ ms}$$

$$0 \leqq \boldsymbol{I}_{i,j} \leqq 30^\circ \text{ BTDC}$$

$$-30 \leqq \boldsymbol{V}_j \leqq 30^\circ \text{ BTDC}$$

$$L = 20\%$$

where $\boldsymbol{M}_1(\boldsymbol{x})$ represents the idle-speed regulation quality, $\boldsymbol{M}_2(\boldsymbol{x})$ represents the idle-speed emission quality (ideally when the stoichiometric ratio $\lambda = 1$, the catalytic converter gains the maximum conversion efficiency of the exhaust gas), $\boldsymbol{M}_3(\boldsymbol{x})$ is employed to assess the idle-speed fuel consumption, and $\boldsymbol{M}_4(\boldsymbol{x})$ and $\boldsymbol{M}_5(\boldsymbol{x})$ are used together for assessing the idle-speed load rejection ability. In this case study, w_{IAE_R}, w_{IAE_λ}, $w_{\sum F}$, $w_{R_{\min}}$, and $w_{T_{\text{rise}}}$ were set to be 3, 2, 3, 4, and 1 respectively. Each performance index is also transformed to a scale of (0, $\pi/2$) by $\tan^{-1}(\cdot)$ in equation (15). This ensures that each index has the same contribution to the objective function. The objective function is manipulated by the two optimizers for generating the best ISC settings.

4.2 GA optimization results

The GA optimization framework was implemented using Matlab. By testing various crossover and mutation methods for this application, the GA operators and parameters as shown in Table 5 were selected to ensure maximum efficiency and accuracy.

Table 6 shows 10 sets of optimization results obtained by the GA. The 10 sets of optimization results differ from each other. This is because the GA is initialized with random start points within the search space and the search result is very sensitive to

Table 5 GA operators and parameters

Number of generation	1000
Population size	50
Selection method	Standard proportional selection
Crossover method	Simple crossover with probability = 80%
Mutation method	Hybrid static Gaussian and uniform mutation with probability = 40% and standard deviation = 0.2

the initial values. The best fitness value out of 10 is −6.7010 and its optimal set points are shown in Table 7.

4.3 PSO results

The PSO framework was also implemented using Matlab. In PSO, the population is called a swarm and the individuals (i.e. different combinations of ECU set points $\boldsymbol{x}$) are called particles. Regarding an n-dimensional search space and a swarm consisting of M particles, the ith particle is represented by an n-dimensional vector $\boldsymbol{x}_i = (x_{i1}, x_{i2}, \ldots, x_{in})$, the velocity of this particle is an n-dimensional vector $\boldsymbol{v}_i = (v_{i1}, v_{i2}, \ldots, v_{in})$, and the best previous position encountered by this particle is described by $\boldsymbol{p}_i = (p_{i1}, p_{i2}, \ldots, p_{in})$. Let g represent the index of the particle that attains the best previous position among all the particles in the swarm. Then, the swarm is manipulated in accordance with the following equations [**24**, **25**]

$$\boldsymbol{v}_i(j+1) = w_c\boldsymbol{v}_i(j) + c_1 r_1[\boldsymbol{p}_i(j) - \boldsymbol{x}_i(j)] + c_2 r_2[\boldsymbol{p}_g(j) - \boldsymbol{x}_i(j)] \tag{16}$$

$$\boldsymbol{x}_i(j+1) = \boldsymbol{x}_i(j) + \boldsymbol{v}_i(j+1) \tag{17}$$

where i is the particle index $i = [1, 2, \ldots, M]$. The selection of the above parameters was widely studied in the relevant literature [**24**, **26**]. With reference to the literature, the PSO parameters as shown in Table 8 were selected for this case study.

Table 9 shows 10 sets of optimization results obtained by PSO. Although it is also initialized arbitrarily, the standard deviation of PSO in Table 9 is only 0.0229, which is much less than 0.0583 of the GA shown in Table 6. In other words, the PSO results are more stable than those of the GA. This is because PSO is somehow insensitive to the initial values. The best fitness value of PSO in Table 9 is −6.6666 and its optimal set points are shown in Table 10. It is also found that the best fitness value of PSO (−6.6666) is larger than that of the GA (−6.7010).

Table 8 PSO parameters

Number of generation	1000
Population size	50
w_c	0.9
c_1	2
c_2	2

Table 6 Optimization results of 10 independent runs of GA

Result set	IAE_R (r/min)	IAE_λ	ΣF (ms)	$R_{\min}$ (r/min)	T_{rise} (s)	Fitness
1	5665.17	5.97	273.92	741	1.52	−6.9174
2	6135.25	5.46	293.51	778	1.53	−6.8886
3	5787.51	4.94	277.23	846	1.57	−6.8454
4	**5013.49**	**5.14**	**351.31**	**672**	**1.15**	**−6.7010**
5	5822.16	4.65	276.14	934	1.44	−6.7949
6	5390.34	4.79	276.12	883	1.45	−6.8066
7	5882.28	4.89	271.56	931	1.42	−6.8254
8	6147.47	4.73	246.02	998	1.53	−6.8416
9	6193.63	4.38	252.31	1063	1.55	−6.8048
10	6467.42	4.57	259.57	1093	1.61	−6.8440
Standard deviation						0.0583

Table 7 Optimal set points recommended by GA

	$\boldsymbol{F}_{i,j}$ MAP (kPa)				$\boldsymbol{I}_{i,j}$ MAP (kPa)				
Engine speed (r/min)	20	30	40	50	20	30	40	50	$\boldsymbol{V}_j$
500	1.99	2.11	2.23	3.39	11.1	11.9	12.6	13.2	0.4
1000	1.91	2.13	2.25	3.34	13	14	15.4	14.9	8.1
1500	2.12	2.25	2.25	3.55	16.6	16.9	17.8	18.5	10.3
BPAV controller parameters		*Pro* 0.097		*Int* 0.072		*Der* 0.081		*Nor* 34.4	

Table 9 Optimization results of 10 independent runs of PSO

Result set	IAE_R (r/min)	IAE_λ	ΣF (ms)	R_{min} (r/min)	T_{rise} (s)	Fitness
1	4807.36	5.02	357.82	695	1.04	−6.6928
2	**4996.41**	**4.82**	**356.83**	**724**	**1.02**	**−6.6666**
3	5228.35	5.03	361.54	747	1.00	−6.6707
4	5095.62	5.02	362.66	740	1.00	−6.6725
5	4860.74	4.81	357.01	727	1.01	−6.6605
6	4908.31	4.90	369.43	717	0.97	−6.6453
7	4812.84	5.05	359.84	689	0.94	−6.6428
8	4736.69	4.87	345.53	665	0.92	−6.6189
9	4804.78	5.11	355.62	639	0.95	−6.6540
10	4915.12	5.14	337.77	609	1.03	−6.6935
Standard deviation						0.0229

Table 10 Optimal set points recommended by PSO

Engine speed (r/min)	$F_{i,j}$ MAP (kPa)				$I_{i,j}$ MAP (kPa)				V_j
	20	30	40	50	20	30	40	50	
500	1.96	2.05	2.17	3.28	10.8	11.7	12.6	13.2	0.0
1000	1.9	2.13	2.27	3.31	11.9	13.4	15.7	16.4	4.3
1500	2.12	2.18	2.23	3.47	14.7	15.2	16.7	17.3	6.7
BPAV controller parameters		*Pro*		*Int*		*Der*		*Nor*	
		0.08		0.041		0.032		30.6	

4.4 Evaluation of optimization results

To check the feasibility and efficiency of the methodology, the optimal settings found by the GA and PSO were then sent back to the ECU and evaluation tests were carried out using the chassis dyno. Figures 9 to 11 present the actual engine idle performance based on the optimal settings.

Figure 9 shows the idle-speed regulation performance using the optimal settings found by both optimizers. Before the load is applied, both engine idle speeds run steady and closely to the aimed speed. When the load is applied, the engine using the set points recommended by the GA falls to a minimum speed of 641 r/min and then takes 1.18 s to recover. On the contrary, the engine using the set points recommended by PSO falls to a minimum speed of 723 r/min and then takes 1.1 s to recover. As compared with the steady state idle-speed performance of $\boldsymbol{D}_{\mathbf{best}}$ in Fig. 5, the speed fluctuation at

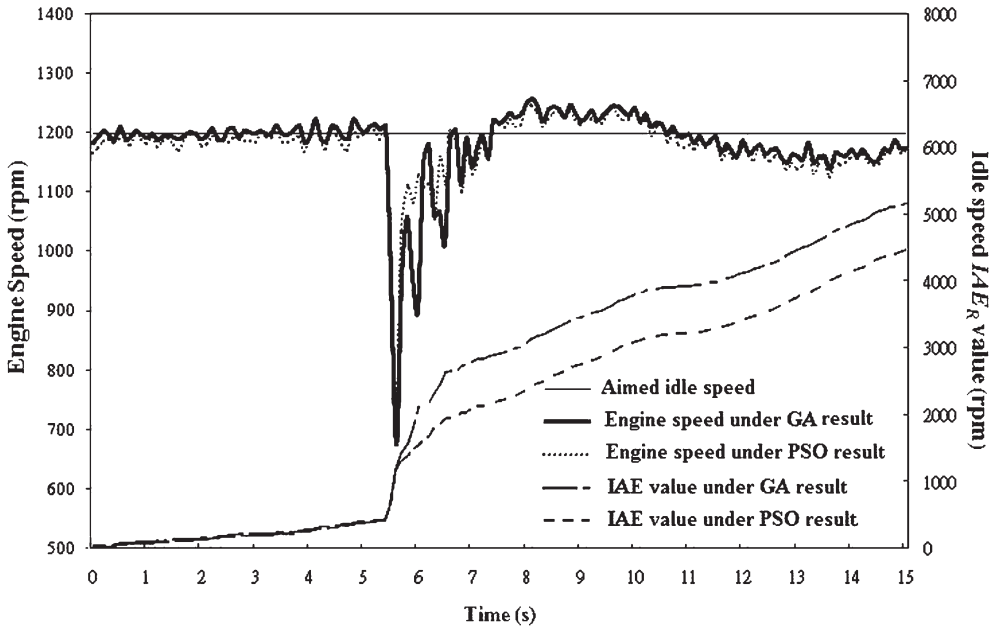

Fig. 9 Actual idle-speed regulation performance using the optimal settings of the GA and PSO

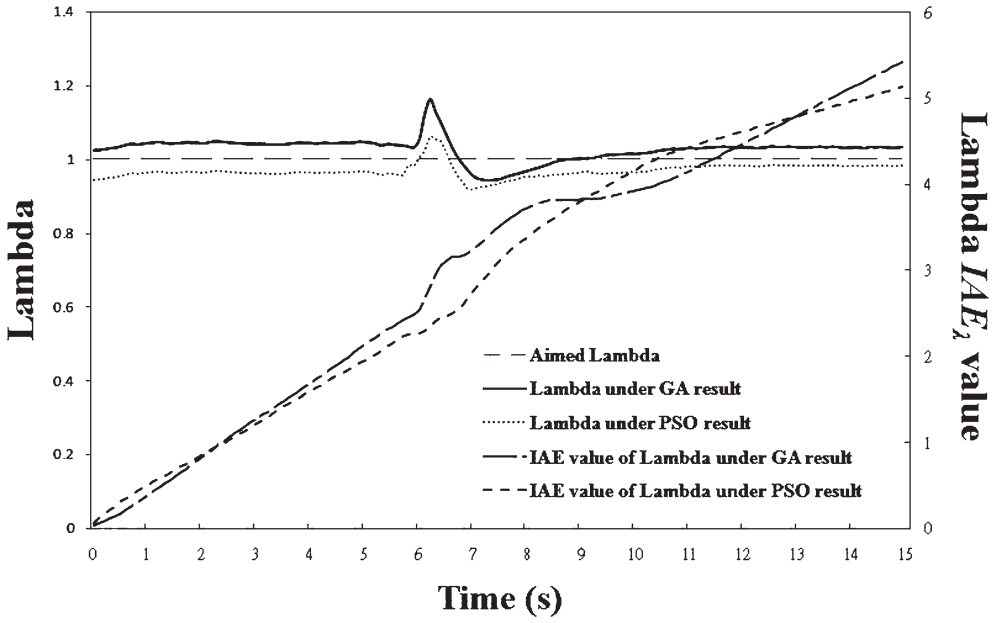

Fig. 10 Actual lambda performance using the optimal settings of the GA and PSO

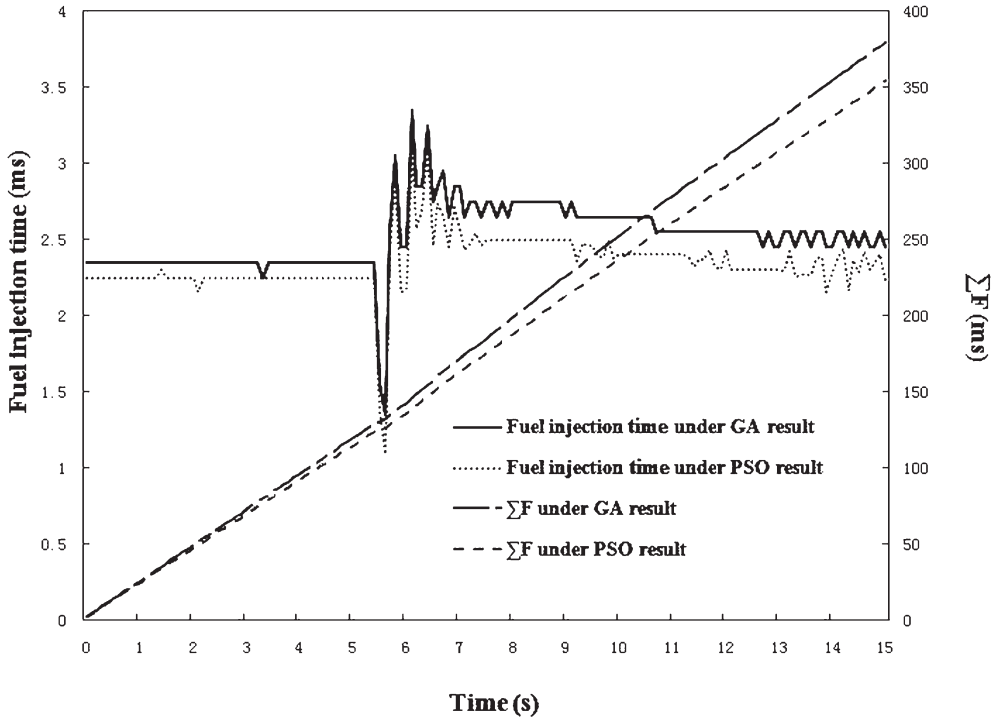

Fig. 11 Actual fuel consumption using the optimal settings of the GA and PSO

steady state can be improved using both optimal settings. Figure 10 shows the engine lambda performance using the optimal settings of the two optimizers. As compared with the lambda performance of the GA, the performance of PSO is closer to the target lambda value along the test. Figure 11 shows that the overall fuel consumption of PSO is also lower than that of the GA. Table 11 shows a comparison among the optimization results, actual test results, and the results of $\boldsymbol{D}_{\mathbf{best}}$. The accuracies in Table 11 show that both the GA and PSO results are in good agreement with the actual test results. These verify again that the engine idle-speed model built by the MIMO LS-SVM is accurate and reliable.

In Table 11, the actual test results using the set points produced by the GA outperform $\boldsymbol{D}_{\mathbf{best}}$ by

Table 11 Comparison between optimization results and actual test results and $\boldsymbol{D}_{\mathbf{best}}$

	IAE_R (r/min)	IAE_λ	ΣF (ms)	$R_{\min}$ (r/min)	T_{rise} (s)	Fitness
$\boldsymbol{D}_{\mathbf{best}}$	6192.36	6.72	462.01	602	1.05	−6.7972
GA optimization results (GA_O)	5013.49	5.14	351.31	672	1.15	−6.7010
PSO optimization results (PSO_O)	4996.41	4.82	356.83	724	1.02	−6.6666
GA actual test results (GA_A)	5127.36	5.42	379.34	641	1.18	−6.7490
PSO actual test results (PSO_A)	4465.42	5.12	354.06	723	1.10	−6.7036
Accuracy of GA results (GA_O relative to GA_A)	97.78%	94.83%	92.61%	95.16%	86.36%	99.29%
Accuracy of PSO results (PSO_O relative to PSO_A)	88.11%	94.14%	99.20%	99.76%	90.48%	99.45%
Actual improvement of GA (GA_A relative to $\boldsymbol{D}_{\mathbf{best}}$)	17.20%	19.35%	17.97%	6.48%	−12.38%	0.71%
Actual improvement of PSO (PSO_A relative to $\boldsymbol{D}_{\mathbf{best}}$)	27.89%	23.81%	23.38%	20.04%	−4.00%	1.38%
Comparison between PSO_A and GA_A (PSO_A relative to GA_A)	12.91%	5.54%	6.60%	12.74%	6.78%	0.84%

about 17.20, 19.35, 17.97, and 6.48 per cent in the actual idle-speed regulation ability, emission quality, fuel consumption, and minimum idle speed respectively, whereas the actual test results using the set points produced by PSO outperform $\boldsymbol{D}_{\mathbf{best}}$ by about 27.89, 23.81, 23.38, and 20.04 per cent in the same four performance indexes respectively. These verify that both optimization algorithms are effective. The recovery time of the idle speed is sacrificed in the objective function, $w_{T_{\text{rise}}}$ was set to be the lowest value in this case study, so the recovery time based on the GA and PSO optimal settings is 12.38 and 4.00 per cent longer than that of $\boldsymbol{D}_{\mathbf{best}}$ respectively. However, the recovery time is still acceptable.

Table 11 also indicates that the actual engine idle-speed regulation ability, emission quality, fuel consumption, minimum idle speed, and recovery time of PSO are respectively 12.91, 5.54, 6.60, 12.74, and 6.78 per cent better than the corresponding values of the GA. The above results show that PSO is superior to the GA in this case study. The reason may be that the mechanism of PSO can generate more diverse populations during the whole iteration process.

5 CONCLUSIONS

This paper proposes a novel methodology for engine idle-speed system modelling and optimization. The approach uses a novel MIMO LS-SVM + LHS framework for modelling and a multi-objective GA/PSO framework to manipulate the engine model built to determine the best combination of control parameters automatically. A case study successfully demonstrates its application to a real automotive engine. Evaluation tests show that the predicted results using the estimated model from the MIMO LS-SVM are in good agreement with the actual test results. In comparison with the ordinary neural network modelling approach, the novel MIMO LS-SVM outperforms by about 11.40 per cent in overall accuracy under the same test dataset. The optimization results in Tables 6, 9, and 11 show that PSO is superior to the GA under the LS-SVM model. An impressive improvement in engine idle performance is also achieved using the optimal setting generated by PSO. Both prediction and experimental results indicate that the proposed methodology (LHS + MIMO LS-SVM + PSO) can really produce reliable and high-quality engine idle-speed performance.

In addition, this research is also a first attempt at integrating a couple of paradigms (LHS, MIMO LS-SVM and GA/PSO) into a general framework for constrained multivariable optimization problems under insufficient system information. The idle-speed problem presented in this paper is to optimize a lot of ECU base map variables and air valve controller parameters for maximizing engine idle-speed regulation quality, load rejection ability, emission quality, and fuel economy all together. Therefore the variables and optimization objectives involved in this paper are more comprehensive and practical than the schemes presented in the existing literature.

From the perspective of automotive engineering, the integrated modelling and optimization methodology is a new approach and can be applied to the following engine set-up and control problems.

1. Engine tune-up and ECU calibration. Compared to the conventional manual tuning approach for current production car engines, the proposed new methodology can greatly reduce the number of expensive dyno tests. This saves not only the time taken for optimal tune-up but also a large amount of resources. It is also believed that the optimization results can be further improved if more training data are added to the LS-SVM model.
2. Engine idle-speed system identification and controller design. The proposed modelling approach can be employed to build many types of practical

engine models exactly, and those models can be employed to reflect the true engine performance for advanced idle-speed controller design. In comparison with the traditional engine models used in the advanced controllers, such as neural networks and empirical models, the proposed modelling approach produces better generalization and accuracy. Moreover, the proposed modelling and/or optimization alogrithms can be used as core components by some advanced model reference controllers, such as model reference adaptive controllers, model identification adaptive controllers, and model-based predictive controllers, etc.

ACKNOWLEDGEMENTS

The research is supported by the University of Macau Research Grant UL011/09-Y1/EME/WPK01/FST and the Science and Technology Development Fund of Macau, Grant 019/2007/A.

REFERENCES

1 **Hrovat, D.** and **Sun, J.** Models and control methodologies for IC engine idle speed control design. *Control Engng Practice*, 1997, **5**(8), 1093–1100.
2 **Robert Bosch GmbH** *Gasoline-engine management,* 2nd edition, 2004 (Robert Bosch GmbH and Professional Engineering Publishing Limited, London).
3 **Howell, M. N.** and **Best, M. C.** On-line PID tuning for engine idle-speed control using continuous Action Reinforcement Learning Automata. *Control Engng Practice*, 2000, **8**, 147–154.
4 **Kim, D.** and **Park, J.** Application of adaptive control to the fluctuation of engine speed at idle. *Information Sci.*, 2007, **177**, 3341–3355.
5 **Manzie, C.** and **Watson, H. C.** A novel approach to disturbance rejection in idle speed control towards reduced idle fuel consumption. *Proc. IMechE, Part D: J. Automobile Engineering*, 2003, **217**(D8), 677–690. DOI: 10.1243/09544070360692078.
6 **Grizzle, J. W., Julia, B.,** and **Jing, S.** Idle speed control of a direct injection spark ignition scarified charge engine. *Int. J. Robust and Nonlinear Control*, 2001, **11**, 1043–1071.
7 **Petridis, A. P.** and **Shenton, A. T.** Inverse-NARMA: a robust control method applied to SI engine idle-speed regulation. *Control Engng Practice*, 2003, **11**, 279–290.
8 **Christian, B., Thomas, B., Aik, S.,** and **Petra, M.** A nonlinear model for design and simulation of automotive idle speed control strategies. In Proceedings of the 2006 American Control Conference, USA, 2006, pp. 3272–3277.
9 **Li, G. Y.** *Application of intelligent control and MATLAB to electronically controlled engines* (in Chinese), 2007 (Publishing House of Electronics Industry, China).
10 **Söderström, T.** and **Stoica, P.** *System identification,* 1st edition, 1989 (Prentice-Hall Press, Cambridge).
11 **Beham, M.** and **Yu, D. L.** Modelling a variable valve timing spark ignition engine using different neural networks. *Proc. IMechE, Part D: J. Automobile Engineering*, 2004, **218**(D10), 1159–1171.
12 **Celik, V.** and **Arcaklioglu, E.** Performance maps of a diesel engine. *Applied Energy*, 2005, **81**, 247–259.
13 **Haykin, S.** *Neural networks: a comprehensive foundation,* 2nd edition, 1999 (Prentice-Hall, Englewood Cliffs, New Jersey).
14 **Suykens, J., Gestel, T., Brabanter, J., Moor, B.,** and **Vandewalle, J.** *Least squares support vector machines,* 1st edition, 2002 (World Scientific Press).
15 **Liu, B., Su, H. Y.,** and **Chu, J.** New predictive control algorithms based on least squares support vector machines. *J. Zhejiang University Science*, 2005, **5**, 440–446.
16 **Lunani, M., Sudjianto, A.,** and **Johnson, P. L.** Generating efficient training samples for neural networks using Latin hypercube sampling. In Proceedings of the 1995 Artificial Neural Networks in Engineering Conference, 2005, pp. 209–214.
17 **Pelckmans, K., Suykens, J., Van, G., De Brabanter, J., Lukas, L., Hanmers, B., De Moor, B.,** and **Vandewalle, J.** *LS-SVMlab: a MATLAB/C toolbox for least square support vector machines*, 2003, available from http://www.esat.kuleuven.ac.be/sista/lssvmlab.
18 **Pyle, D.** *Data preparation for data mining,* 1st edition, 1999 (Morgan Kaufmann Press).
19 **Zhang, D., Xu, Z., Mechefske, C. M.,** and **Xi, F.** Optimum design of parallel kinematic toolheads with genetic algorithms. *Robotica*, 2004, **22**, 77–84.
20 **Wong, P. K., Mok, K. W.,** and **Vong, C. M.** Design and control of an electromechanical variable rotary valve system for four-stroke engines. In Proceedings of the 9th International Symposium on Advanced Vehicle Control, vol. II, Japan, 2008, pp. 887–892.
21 **Kennedy, J.** and **Eberhart, R. C.** Particle swarm optimization. In Proceedings of the International Conference on Neural Networks, 1995, pp. 1942–1948.
22 **Thornhill, M.** and **Thompson, S.** A comparison of idle speed control schemes. *Control Engng Practice*, 2000, **8**, 519–530.
23 **Jurgen, R.** *Automotive electronics handbook,* 1st edition, 1995 (McGraw-Hill Press).
24 **Clerc, M.** and **Kennedy, J.** The particle swarm-explosion, stability, and convergence in a multi-dimensional complex space. *IEEE Trans. Evol. Comput.*, 2002, **6**(1), 58–73.

25 **Hu, X., Eberhart, R. C.,** and **Shi, Y.** Engineering optimization with particle swarm. In Proceedings of the IEEE Swarm Intelligence Symposium, 2003, pp. 53–57.
26 **Trelea, I. C.** The particle swarm optimization algorithm: convergence analysis and parameter selection. *Information Processing Lett.*, 2003, **85**(6), 317–325.

APPENDIX

Notation

b_h	bias of the hth engine model
c_1	cognitive parameter
c_2	social parameter
$\mathbf{d}_h$	subtraining dataset for each single output dimension $\boldsymbol{y}_h$
$\boldsymbol{d}_k$	kth data point in $\mathbf{d}_h$
$\mathbf{D}$	training dataset
$\boldsymbol{D}_{\mathbf{best}}$	best performance set-up among the 200 sample datasets
Der	derivative gain of the idle air valve controller
$\boldsymbol{e}_h$	residual vector for the hth engine model
E_h	root-mean-square error of the hth engine model
$\boldsymbol{F}_{i,j}$	fuel injection time at the corresponding MAP i and idle speed j
h	engine performance model index
i	particle index of PSO
IAE_R	integral absolute error of engine idle speed
IAE_λ	integral absolute error of lambda
$\boldsymbol{I}_{i,j}$	ignition advance at the corresponding MAP i and idle speed j
Int	integral gain of the idle air valve controller
$\mathbf{I}_N$	N-dimensional identity matrix
j	iteration counter of PSO
K	predefined kernel function
L	constant step load
m	dimension of $\boldsymbol{y}_k$
$\boldsymbol{M}_h(\boldsymbol{x})$	hth engine performance model
n	dimension of $\boldsymbol{x}_k$
n_h	dimension of the unknown feature space
N	number of data points in $\mathbf{D}$
Nor	normal position of the idle air valve
$NET_h(\boldsymbol{x})$	hth engine performance model trained by the neural network
$\boldsymbol{p}_g$	best previous position among all particles
$\boldsymbol{p}_i$	best previous position encountered by the ith particle
Pro	proportional gain of the idle air valve controller
r_1, r_2	random numbers uniformly distributed between 0 and 1
R_{aim}	aimed idle speed
$R_{\min}$	minimum idle speed under a step load
R_t	engine idle speed at the corresponding time t
t_f	data recording time
TEST_h	test dataset for the hth engine model
TRAIN_h	training dataset for the hth engine model
T_{rise}	recovery time to aimed speed under a step load
v	input control parameter before normalization
v^*	normalized input control parameter
$v_{\max}$	upper limit of the input control parameter before normalization
$v_{\min}$	lower limit of the input control parameter before normalization
$\boldsymbol{v}_i$	velocity of the ith particle in PSO
$\boldsymbol{V}_j$	intake valve open timing at the corresponding idle speed j
w_c	inertial weight
$\boldsymbol{w}_h$	weight vector of the hth engine model
w_{IAE_R}	user-defined weight of engine idle-speed regulation
w_{IAE_λ}	user-defined weight of engine idle-speed emission quality
$w_{R_{\min}}$	user-defined weight of minimum idle speed
$w_{T_{\text{rise}}}$	user-defined weight of recovery time to aimed speed
$w_{\Sigma F}$	user-defined weight of engine idle-speed fuel economy
$\boldsymbol{x}$	input engine set-up of the engine performance model
$\boldsymbol{x}_i$	ith particle vector in PSO
$\boldsymbol{x}_k$	kth engine set-up in the training dataset $\mathbf{D}$
$\boldsymbol{y}$	output vector of the engine performance model
$\boldsymbol{y}_h$	hth engine output performance data in $\boldsymbol{y}$
$\boldsymbol{y}_k$	kth engine output performance training data based on the kth engine set-up $\boldsymbol{x}_k$

$y_{k,h}$	hth engine output performance data point in $\boldsymbol{y}_k$	λ_{aim}	target lambda value
		λ_t	engine lambda value at the corresponding time t
$\boldsymbol{\alpha}_h$	support vector of the hth engine model	σ_h	kernel sample variance of the hth engine model
γ_h	regularization scalar factor of the hth engine model	ΣF	overall fuel consumption
		$\boldsymbol{1}_v$	N-dimensional vector $= [1 \ \ldots \ 1]^{\mathrm{T}}$