



Initial-training-free online sequential extreme learning machine based adaptive engine air–fuel ratio control

Pak Kin Wong¹ · Xiang Hui Gao² · Ka In Wong³ · Chi Man Vong⁴ · Zhi-Xin Yang¹

Received: 24 October 2017 / Accepted: 4 August 2018 / Published online: 11 August 2018
© Springer-Verlag GmbH Germany, part of Springer Nature 2018

Abstract

In modern automotive engines, air–fuel ratio (AFR) strongly affects exhaust emissions, power, and brake-specific consumption. AFR control is therefore essential to engine performance. Most existing engine built-in AFR controllers, however, are lacking adaptive capability and cannot guarantee long-term control performance. Other popular AFR control approaches, like adaptive PID control or sliding mode control, are sensitive to noise or needs prior expert knowledge (such as the engine model of AFR). To address these issues, an initial-training-free online sequential extreme learning machine (ITF-OSELM) is proposed for the design of AFR controller, and hence a new adaptive AFR controller is developed. The core idea is to use ITF-OSELM for identifying the AFR dynamics in an online sequential manner based on the real-time engine data, and then use the ITF-OSELM model to calculate the necessary control signal, so that the AFR can be regulated. The contribution of the proposed approach is the integration of the initial-training-free online system identification algorithm in the controller design. Moreover, to guarantee the stability of the closed-loop control system, a stability analysis is also conducted. To verify the feasibility and evaluate the performance of the proposed AFR control approach, simulations on virtual engine and experiments on real engine have been carried out. Both results show that the proposed approach is effective for AFR regulation.

Keywords Automotive engine · Air–fuel ratio · Online sequential extreme learning machine · Adaptive control

Abbreviations

a_i	Input weight of the i th hidden node
b_i	Bias of the i th hidden node
e_{k+1}	Error between system output and reference
\hat{e}_{k+1}	Error between system output and model prediction
$g(\mathbf{x}_k)$	Part of system to be identified
$\hat{g}(\mathbf{x}_k, \boldsymbol{\beta}_g)$	Approximating function for function $g(\mathbf{x}_k)$
$G(a_i, b_i, \mathbf{x}_k)$	Mapping function of the i th hidden node
\mathbf{H}_g	Hidden layer output for function g
\mathbf{H}_φ	Hidden layer output for function φ
\mathbf{I}	Identity matrix

P_0	Initial updating term for ITF-OSELM
P_{k+1}	Updated term by using the $(k + 1)$ th arriving training data
t	Time
u_k	Control signal of the k th step
v_i	Time varying factor
\mathbf{x}_k	System state at the k th step
y_{k+1}	System output for control signal u_k
\hat{y}_{k+1}	Model prediction for control signal u_k
$y_{r(k+1)}$	Tracking reference of the $(k + 1)$ th step
$\boldsymbol{\beta}^0$	Initial output weights
$\boldsymbol{\beta}^{(k+1)}$	Updated output weights by using the $(k + 1)$ th arriving training data
$\boldsymbol{\beta}_g$	Output weights of approximating function \hat{g}
$\boldsymbol{\beta}_\varphi$	Output weights of approximating function $\hat{\varphi}$
γ	Regularization factor
$\lambda(t)$	Measured lambda value at time t
$\lambda_d(t)$	Desired lambda value at time t
ρ	Forgetting factor
$\varphi(\mathbf{x}_k)$	Part of system to be identified
$\hat{\varphi}(\mathbf{x}_k, \boldsymbol{\beta}_\varphi)$	Approximating function for function $\varphi(\mathbf{x}_k)$

✉ Pak Kin Wong
fstpkw@umac.mo

¹ Department of Electromechanical Engineering, University of Macau, Macau, China

² Key Laboratory of Machine Learning and Computational Intelligence, College of Mathematics and Information Science, Hebei University, Baoding, Hebei, China

³ Institute for the Development and Quality, Macau, China

⁴ Department of Computer and Information Science, University of Macau, Macau, China

1 Introduction

In automotive engineering area, air–fuel ratio (AFR) is defined as the ratio of the amount of air to the amount of fuel consumed by an engine for combustion. As the engine combustion process is very sensitive to AFR, slight variations of AFR can already alter the engine behavior among highest power, lowest emissions and best fuel economy. For example, as shown in Fig. 1 [1], the amount of engine emissions for gasoline is minimum when AFR is at 14.7, which is the stoichiometric ratio of gasoline indicating that correct amount of air and fuel are supplied to produce a complete combustion; while peak torque usually occurs at AFR values around 12.5 (i.e., 85% of the stoichiometric ratio). The conversion efficiency of the three-way catalytic converter for emission reduction also maximizes at stoichiometric AFR. In order to achieve different engine performance objectives, such as more engine power and lower engine emissions, different AFRs should be targeted. Precise control of AFR is therefore essential for maintaining the best engine performance at various operating conditions and driving requirements.

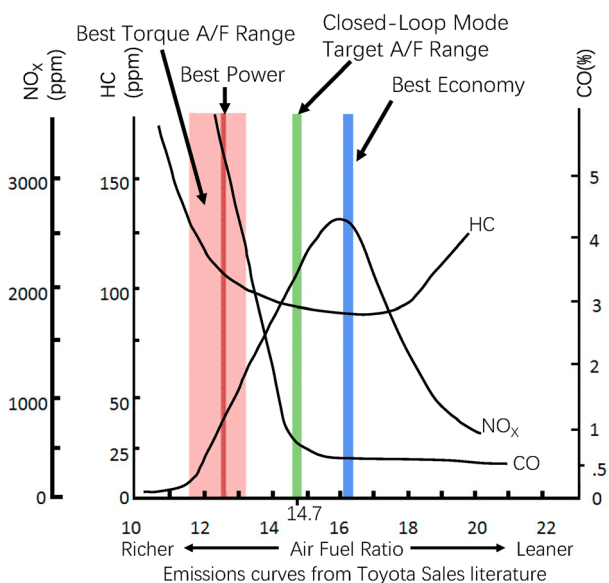
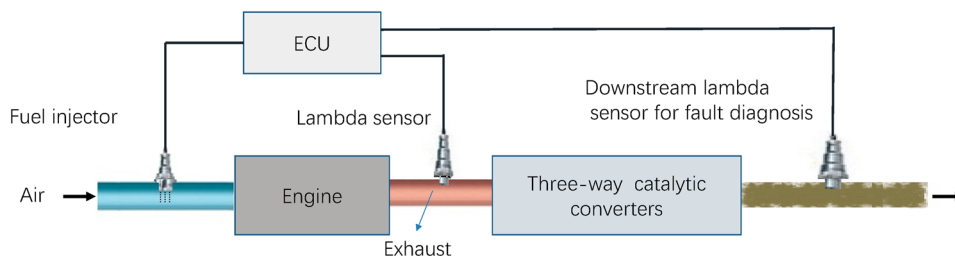


Fig. 1 Engine performance against air–fuel ratio for gasoline [1]

Fig. 2 Closed-loop AFR control



Practically, AFR control is achieved by using closed-loop feedback controllers. The feedback of AFR usually comes from the lambda sensors (also known as oxygen sensors) installed on the vehicle tailpipe. It is worth noting that lambda is a normalized AFR value of which the stoichiometry is always 1 regardless the fuel types. As shown in Fig. 2, commonly two lambda sensors are used on a vehicle, one before and one after the three-way catalytic converter, to ensure that the three-way catalytic converter of the vehicle works properly. For AFR control, only the lambda signal from the upstream lambda sensor is acquired as it is closer to the engine combustion chamber. In the closed-loop AFR control system as illustrated in Fig. 2, the lambda signal is firstly sent to the electronic control unit (ECU) of the engine and compared with the target AFR preset in the ECU. Then, the error is received by the built-in controller in the ECU to calculate for the necessary amount of fuel. Finally, the control signal of the fuel injection is sent to the fuel injector to adjust the AFR.

In most existing engines, proportional-integral-derivative (PID) controllers are adopted for this closed-loop AFR control. However, it is well-known that tuning for the parameters of the PID controller is a tedious process and is engine dependent. Moreover, even if the PID controller is well-tuned for an engine, it still cannot guarantee long-term AFR control performance due to its use of fixed gains and the presence of uncertainty with engine aging. Therefore, various intelligent techniques have been developed in the past decade for AFR control [2, 3]. An adaptive PID controller [4] and a parameter-varying filtered PID strategy for AFR control [5] were respectively proposed to adjust the controller gains, so that the controller characteristics could be varied in accordance with the engine conditions. However, noise disturbance and the highly nonlinear nature of AFR dynamics still limit the performance of such linear-based PID controllers. To deal with this problem, sliding-mode control (SMC) was widely studied [6–9]. An observer-based fuel injection control algorithm based on SMC strategy was proposed [6] and the results showed that it was better than engine factory controller. Later, an adaptive SMC strategy for AFR control [7] was proposed, in which an adaptive update law was derived for the fuel parameters. More recently, a SMC for AFR control of a dual-fuel engine [8], a second-order SMC strategy for AFR control of lean-burn engines [9], and a fuzzy sliding-mode

strategy for AFR control of lean-burn spark ignition engines [10], were also developed, which assume that the engine model (a simplified engine model) is known. All these studies showed promising control result of SMC-based controller. Nevertheless, one major limitation for all these SMC-based AFR control approaches is that prior expert knowledge of the engine is required for development of the controller. When there is no prior knowledge available for the target engine or the available knowledge is not sufficient, it is almost impossible to construct a reliable controller for practical use. The automotive engine is a complicated system. So far, there is no exact engine model available yet. Without exact engine model, reliable prior information of the target engine is difficult to know. In addition, the ECU that controls the engine has limited memory size and computation capability, which also make the other complicated control approaches, e.g., composite learning from adaptive dynamic surface control [11] and adaptive fuzzy back-stepping control [12], hard to be applied to real engine control.

In order to address the above issues, the initial-training-free online extreme learning machine (ITF-OELM) developed by the authors in 2016 [13] for general purpose is considered. ITF-OELM is a variant of extreme learning machine (ELM). One advantageous feature of ITF-OELM is that it can train and update the model anytime when necessary without the need of any pre-acquired data. Due to the advantage of initial training free, ITF-OELM is simpler and easier to be implemented, and more suitable for learning problems in which it is difficult to collect sufficient amount of training data in advance. This advantage indicates that ITF-OELM can easily be implemented in real automotive ECUs. Since the ITF-OELM can identify the relationship of an unknown system based on its input–output data, it can be used to identify the AFR dynamics online adaptively based on real-time engine data. However, the ITF-OELM used in this paper does not need a mechanism for change detection which is different from the original ITF-OELM. It is because the threshold of the change detection mechanism in [13] is difficult to determine. Moreover, the change detection mechanism will result in frequently retraining of engine AFR model. The simulation results in [13] show that the controller needs some time steps to settle every time when the model retraining is required. AFR is very sensitive to engine performance, a small change of AFR may cause engine stall [1]. To avoid the risk of engine stall before settle down, the change detection mechanism in ITF-OELM is ignored in this work. To distinguish the ITF-OELM in this paper from the author's previous work, the ITF-OELM in this paper is called initial-training-free online sequential extreme learning machine (ITF-OSELM).

With the adaptively updated ITF-OSELM model instead of frequently retrained ITF-OELM model, the necessary control signal in each combustion iteration can be properly

calculated, and the AFR can eventually be regulated. Unlike the ELM-based direct adaptive controllers proposed in [14, 15] that the controller parameters are directly updated from an adaptation error, the proposed controller is an indirect adaptive neural controller, in which the adaptation of the controller parameters is done in two stages: (1) online estimation of the plant and online computation of the controller parameters based on the current estimated plant model; (2) ITF-OSELM is used for identifying the AFR dynamics in an online sequential manner based on the real-time engine data, and then the ITF-OSELM model is used to calculate the necessary control signal in order to adjust the engine AFR to the desired level accurately. The proposed approach also differs from the ELM-based indirect adaptive controller developed in [16], in which a sliding mode controller is combined but the plant model is known.

In summary, existing PID-based engine AFR controllers are lack of adaptive capability, and other adaptive approaches generally require a dynamic model of the engine. However, since the automotive engine is a complicated system, no exact engine dynamic model is available in the literature yet. That means the parameters of the existing controllers need to be updated periodically to ensure the AFR controller performance. So the main motivation of this paper is to develop an automotive engine AFR controller that can adaptively adjust its parameters to maintain the AFR performance when the engine condition changes due to various uncertainties. The main contributions of this paper include: (1) an online identification strategy for the automotive engine AFR system using ITF-OSELM is designed; (2) an adaptive AFR control strategy based on the identified AFR system is proposed; (3) the proposed adaptive controller is applied to AFR control for a real automotive engine, while many existing studies are simulation work only.

The organization of this paper is as follows. The proposed ITF-OSELM based adaptive engine air–fuel ratio regulation is derived in Sect. 2. To verify the feasibility of the proposed approach, both simulations on virtual engine and experiments on real engine are carried out. Evaluation of the control performance is also conducted by comparing with the built-in PID controller in the ECU. The simulation and experiment studies are presented respectively in Sects. 3 and 4. The conclusions are given in Sect. 5.

2 Adaptive AFR control using ITF-OSELM

2.1 Problem formulation

Theoretically, the dynamics of engine AFR can be described by the following approximated discrete system, in which the control appears linearly [17]:

$$y_{k+1} = g(\mathbf{x}_k) + \varphi(\mathbf{x}_k)u_k \tag{1}$$

where y is the system output (i.e., lambda), u is the control input (i.e., fuel injection), $g(\cdot)$ and $\varphi(\cdot)$ are unknown nonlinear functions with unknown parameters, k is the time step, and $\mathbf{x}_k = [y_k, \dots, y_{k-n+1}, u_{k-1}, \dots, u_{k-n+1}]$ with n being the system order.

The following assumptions are made in accordance with the AFR dynamics:

1. The system is observable. This is valid as the lambda is measurable.
2. $\varphi(\cdot)$ is bounded away from zero. This means that change in fuel injection can always influence the AFR.
3. The system is bounded-input bounded-output stable. This is valid as for any input within the known stable bound, the corresponding engine response is always bounded and stable.
4. For a known bounded desired reference output, y_r , there exists a bounded control input that can drive the system output to follow the reference output.
5. There exist exact parameters such that both $g(\cdot)$ and $\varphi(\cdot)$ can be approximated with zero error.

Under these assumptions, if both $g(\cdot)$ and $\varphi(\cdot)$ can be exactly approximated using $\hat{g}(\cdot)$ and $\hat{\varphi}(\cdot)$, then the following control law can be used to exactly track the desired reference y_r :

$$u_k = \frac{y_{r(k+1)} - \hat{g}(\mathbf{x}_k)}{\hat{\varphi}(\mathbf{x}_k)}, \tag{2}$$

such that,

$$y_{k+1} = g(\mathbf{x}_k) + \varphi(\mathbf{x}_k) \frac{y_{r(k+1)} - \hat{g}(\mathbf{x}_k)}{\hat{\varphi}(\mathbf{x}_k)} \cong y_{r(k+1)}, \tag{3}$$

when $\hat{g}(\cdot) \cong g(\cdot)$ and $\hat{\varphi}(\cdot) \cong \varphi(\cdot)$. Therefore, the core idea of the proposed control strategy is to adaptively learn the two unknown functions online by ITF-OSELM.

2.2 Identifying engine AFR system using ITF-OSELM

In the control law of Eq. (2), two functions $\hat{g}(\mathbf{x}_k)$ and $\hat{\varphi}(\mathbf{x}_k)$ should be known to calculate the fuel injection time (i.e. control signal) u_k , so that the system output y_{k+1} can track the desired reference $y_{r(k+1)}$. Therefore, the system identification of the two functions is the key to implement the adaptive control.

Since both $g(\mathbf{x}_k)$ and $\varphi(\mathbf{x}_k)$ in Eq. (1) are unknown, generally two networks $\hat{g}(\mathbf{x}_k, \beta_g)$ and $\hat{\varphi}(\mathbf{x}_k, \beta_\varphi)$ are necessary to approximate these two functions based on the system inputs and outputs. However, there is only one single output y_{k+1}

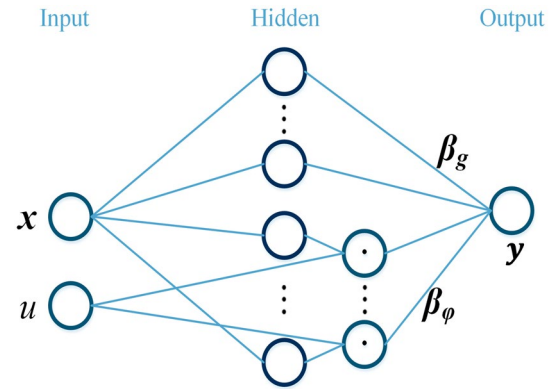


Fig. 3 Network structure for simultaneous approximation of $g(\cdot)$ and $\varphi(\cdot)$

that can be observed each time from the system, so it is practically difficult to explicitly derive the approximation errors of the two functions (i.e., $|\hat{g}(\cdot) - g(\cdot)|$ and $|\hat{\varphi}(\cdot) - \varphi(\cdot)|$) for the system of interest. Fortunately, by considering the approximate system output given as:

$$\hat{y}_{k+1} = \hat{g}(\mathbf{x}_k, \beta_g) + \hat{\varphi}(\mathbf{x}_k, \beta_\varphi)u_k, \tag{4}$$

it is feasible to use only one single network with special structure design to estimate the two functions simultaneously. This special network structure is provided in Fig. 3.

In the structure of Fig. 3, the hidden nodes of the network are separated into two groups, one of which is for $\hat{g}(\cdot)$ and the other is for $\hat{\varphi}(\cdot)$. For simplicity purpose, in this paper, each group contains L hidden nodes, so there are totally $2L$ hidden nodes in the hidden layer. The hidden layer output matrixes for these two groups are denoted as \mathbf{H}_g and \mathbf{H}_φ respectively. It has to be noted that \mathbf{H}_g is different from \mathbf{H}_φ because at the k th step, each i th hidden node output in \mathbf{H}_g is the result of feature mapping (a_i, b_i, \mathbf{x}_k) , while that in \mathbf{H}_φ is the product of the feature mapping $G(a_i, b_i, \mathbf{x}_k)$ and the control input u_k (i.e., $G(a_i, b_i, \mathbf{x}_k) \cdot u_k$). The mapping function $G(a_i, b_i, \mathbf{x}_k)$ is sigmoid, where \mathbf{x}_k is the input instance, a_i is the input weight and b_i is the bias of the i th hidden node. The parameters a_i and b_i are generated randomly. Since at each time step, there is only one sample available, so the resulting hidden layer matrix can be denoted as: $\mathbf{H} = [\mathbf{H}_g, \mathbf{H}_\varphi] = [G(a_1, b_1, \mathbf{x}_k), \dots, G(a_L, b_L, \mathbf{x}_k), G(a_{L+1}, b_{L+1}, \mathbf{x}_k) \cdot u_k, \dots, G(a_{L+L}, b_{L+L}, \mathbf{x}_k) \cdot u_k]$.

The objective of ITF-OSELM is to adjust the weights of this single network by minimizing the approximate error between the exact system output y_{k+1} and the approximated output \hat{y}_{k+1} , i.e.:

$$\text{Minimize: } \hat{y}_{k+1} - y_{k+1} = \|\mathbf{H}\beta - y_{k+1}\| = \left\| [\mathbf{H}_g, \mathbf{H}_\varphi] \begin{bmatrix} \beta_g \\ \beta_\varphi \end{bmatrix} - y_{k+1} \right\|. \tag{5}$$

The procedure for online system identification using ITF-OSELM involves on two phases: (1) initialization phase

and (2), online learning phase, which are summarized in the following:

Initialization phase:

Assign random values for input weights, and set the output weights $\beta^0 = [\beta_g^0, \beta_\varphi^0] = 0$ and the updating term $P_0 = (\gamma I)^{-1}$.

Online learning phase:

For the $(k + 1)$ th arriving training data,

1. Calculate the hidden layer output matrix $H_{k+1} = [H_{g(k+1)}, H_{\varphi(k+1)}]$;
2. Update the output weights $\beta^{(k+1)}$ using the following equations.

$$\beta^{(k+1)} = \beta^{(k)} + P_{k+1} H_{k+1}^T (y_{k+1} - H_{k+1} \beta^{(k)}) \tag{6}$$

$$P_{k+1} = P_k - P_k H_{k+1}^T (I + H_{k+1} P_k H_{k+1}^T)^{-1} H_{k+1} P_k. \tag{7}$$

ITF-OSELM is simpler than the original online sequential extreme learning machine (OS-ELM) [18] and its improved version regularized OS-ELM (ReOS-ELM) [19], and thus it is easier to use. Furthermore, like weighted OS-ELM (WOS-ELM) [20], a forgetting factor can also be introduced for some applications, so that the old learnt data are gradually ignored. It follows the same initialization phase in ITF-OSELM, but in the second phase, the updating rule of Eq. (7) is revised as follows:

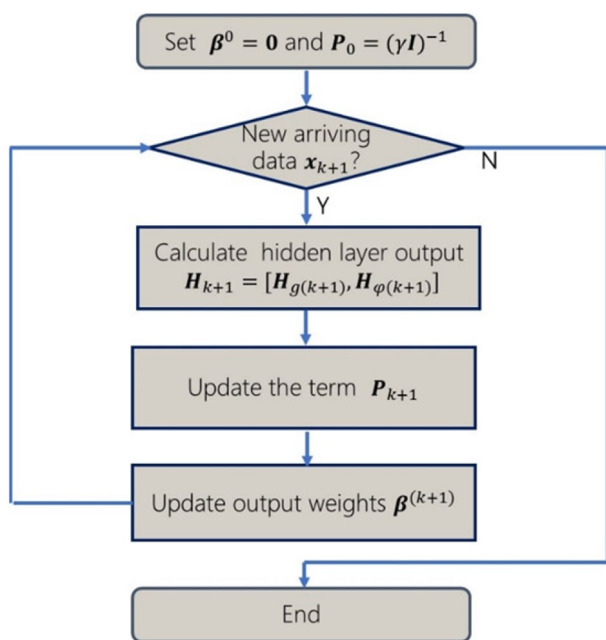


Fig. 4 Flowchart of ITF-OSELM

$$P_{k+1} = \frac{1}{\rho} \left(P_k - P_k H_{k+1}^T \left(\rho I_{N_{k+1}} + H_{k+1} P_k H_{k+1}^T \right)^{-1} H_{k+1} P_k \right) \tag{8}$$

where $0 < \rho \leq 1$ is the forgetting factor. This factor can give the more recent arriving data higher contribution to the adjustment of output weights. It should be noted that when $\rho = 1$, the weight updating algorithm is the same as that of original ITF-OSELM. The above procedure is summarized in a flowchart as shown in Fig. 4.

2.3 Design of ITF-OSELM-based AFR controller

A block diagram is provided in Fig. 5 showing how the AFR is controlled with ITF-OSELM. It should be noted that, the error \hat{e} used for model update is the prediction error between the real value of the system output and the prediction value of the ITF-OSELM model. This prediction error is also called model error, which can be reduced when more data of the system are available for model update. During the online learning process, more input–output dynamic data of the engine system are acquired to update the ITF-OSELM model, and hence the model error, and so as the tracking error, will gradually converge to zero as proved in Eq. (3).

2.4 Stability analysis

To verify the control strategy, the stability analysis from [21] can be referred, in which the error between the system output and the reference (i.e., tracking error) is defined as:

$$e_{k+1} = y_{k+1} - y_{r(k+1)}. \tag{9}$$

Then, introducing two approximated system outputs into Eq. (9) yields:

$$e_{k+1} = y_{k+1} - \hat{y}_{k+1} + \hat{y}_{k+1} - y_{r(k+1)}, \tag{10}$$

$$e_{k+1} = [g(\cdot) + \varphi(\cdot)u_k - \hat{g}(\cdot) - \hat{\varphi}(\cdot)u_k] + [\hat{g}(\cdot) + \hat{\varphi}(\cdot)u_k - y_{r(k+1)}]. \tag{11}$$

Finally, substituting the control law Eq. (2) into Eq. (11), the resulting tracking error becomes:

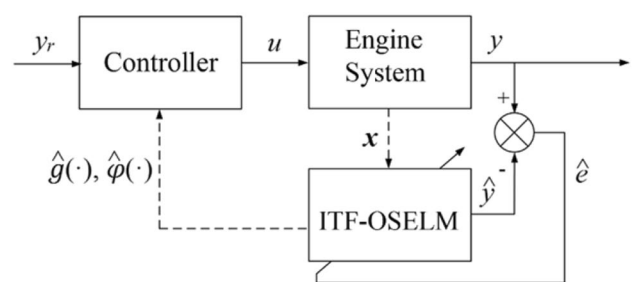


Fig. 5 Control block diagram

$$e_{k+1} = [g(\cdot) + \varphi(\cdot)u_k - \hat{g}(\cdot) - \hat{\varphi}(\cdot)u_k] + [y_{r(k+1)} - y_{r(k)}], \tag{12}$$

$$e_{k+1} = g(\cdot) - \hat{g}(\cdot) + [\varphi(\cdot) - \hat{\varphi}(\cdot)]u_k = y_{k+1} - \hat{y}_{k+1}. \tag{13}$$

If the assumptions in Sect. 2.1 are valid and the control signal is bounded, then the tracking error is equal to the prediction error as shown in Eq. (13). Since the control signal is fuel injection time, the control signal is bounded by the fuel injector hardware. With more training data in the online learning phase, the prediction error converges to zero. If the approximation errors of the two functions converge to zero, the tracking error also becomes zero. Therefore, the bound-input bound-output stability can be guaranteed. Since the weight updating algorithm in ITF-OSELM has the same form as recursive least squares (RLS), the convergence is guaranteed, and hence the stability of the controller can also be guaranteed.

3 Simulation verification

To verify the effectiveness of the proposed ITF-OSELM based adaptive controller before applying to complicated real engine applications, simulations on simulated engines were carried out for different kinds of evaluations.

The objective of Simulation I is to test the identifying capability of ITF-OSELM. Therefore, in this simulation, ITF-OSELM is employed to approximate the whole simulated engine, which is not time-varying.

The objective of Simulation II is to test the adaptive capability of ITF-OSELM based AFR controller. Therefore, in this simulation ITF-OSELM is used to identify the whole simulated engine with a time-varying factor.

Furthermore, to demonstrate the effectiveness of ITF-OSELM, the stochastic gradient back-propagation (SGBP) in [22] was implemented for the adaptive air–fuel ratio controller for comparison. Among the three existing online learning algorithms: OS-ELM, ReOS-ELM and SGBP, OS-ELM and ReOS-ELM need a base model to begin online learning, so only the similar initial-training-free online algorithm SGBP was selected for comparison. In addition, all the simulations in the following sections were implemented in MATLAB and executed on a PC with Intel Core i7 CPU and 4 GB RAM onboard.

3.1 Simulation I for time-invariant system

To simplify the problem, a simulated engine model of 465Q gasoline engine at speed of 3500 rpm and manifold pressure of 85 kPa [23], is used, given as:

$$y_{k+1} = \frac{0.2 \sin y_k + 3.5(9 - u_k)}{14.7}, \tag{14}$$

where the output y_{k+1} is lambda. AFR is measured by lambda sensor in real applications. Therefore, the AFR is also called lambda, which is the ratio of actual AFR to stoichiometry for a given mixture.

$$\lambda = \frac{AFR}{AFR_{stoich}}. \tag{15}$$

Lambda is 1.0 at stoichiometry; lambda less than 1.0 indicates a rich mixture and lambda bigger than 1.0 indicates a lean mixture. For pure gasoline, the AFR_{stoich} is approximately 14.7. In this simulation, the whole system is assumed unknown, and ITF-OSELM is to identify the whole unknown system of Eq. (14) in online manner.

In this simulation, the step response of the controller was tested. The desired lambda y_r was set to drop from 1.00 to 0.95 at some point and then jump to 1.05 afterward, and finally return back to 1.00. The formulation of the reference output in this case is:

$$y_r(k) = \begin{cases} 0.95 & 50 \leq k < 150 \\ 1.05 & 150 \leq k < 250 \\ 1 & \text{otherwise} \end{cases}. \tag{16}$$

Based on this reference, the results of simulations using SGBP and ITF-OSELM with different number of hidden nodes are provided in Figs. 6, 7, and 8, respectively. From Figs. 6 and 8, it can be seen that for both approaches, the actual lambda can quickly track the reference after the first several steps. Both SGBP and ITF-OSELM cannot approximate the AFR system well in the first several steps, because only a few of data can be available. When the reference change abruptly at the 50th, 150th and the 250th steps, the actual output of ITF-OSELM can track the desired reference well. However, the SGBP based approach spends several

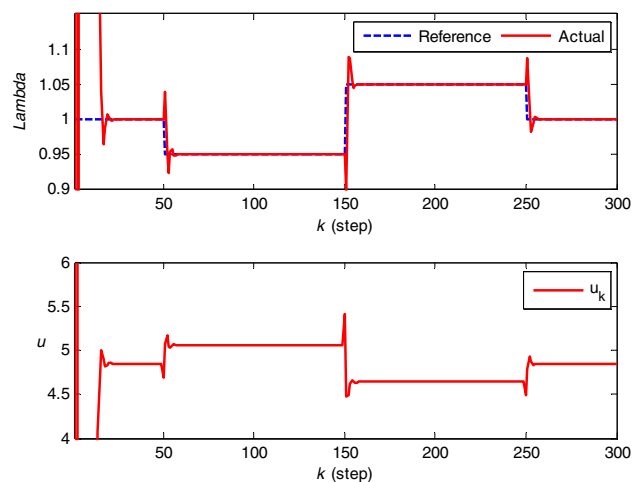


Fig. 6 Control performance of SGBP in Simulation I

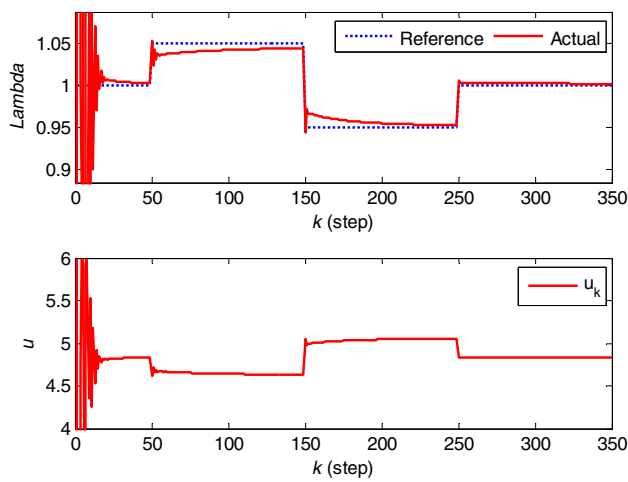


Fig. 7 Control performance of ITF-OSELM with 50 hidden nodes in Simulation I

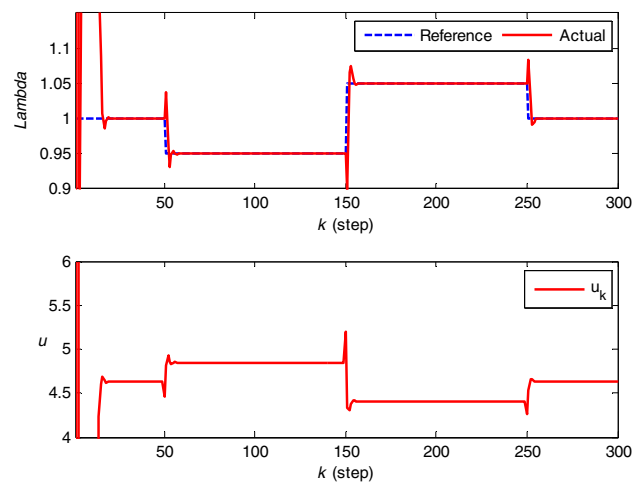


Fig. 9 Control performance of SGBP in Simulation II

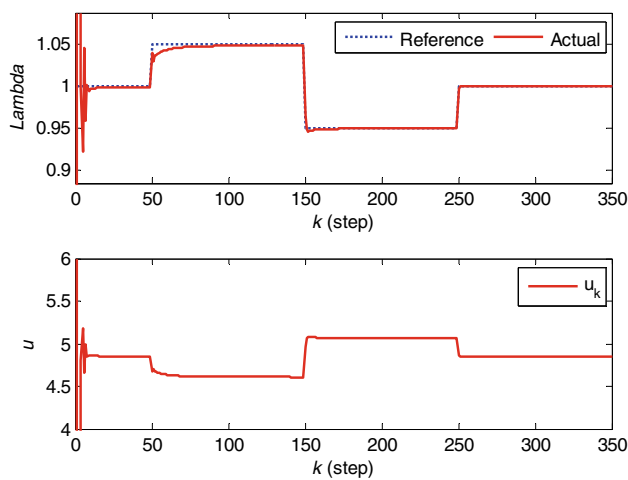


Fig. 8 Control performance of ITF-OSELM with 100 hidden nodes in Simulation I

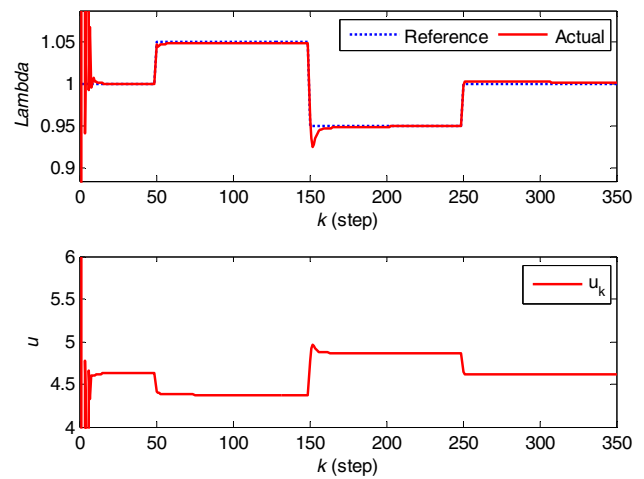


Fig. 10 Control performance of ITF-OSELM without forgetting factor in Simulation II

steps to catch the reference. This simulation results show the powerful capability of both SGBP and ITF-OSELM for online system identification and thus the adaptive control. However, the effectiveness of ITF-OSELM is better than SGBP based approach for the tracking performance.

In this simulation, the regularization factor is 0.001, and the activation function is sigmoid for hidden nodes. In Fig. 7, 50 hidden nodes are used. However, the tracking performance is not good. Therefore, to enhance the approximating capability of ITF-OSELM, 100 hidden nodes are employed to approximate the whole system. With 100 hidden nodes, the good tracking performance can be achieved easily as shown in Fig. 8. For fair comparison, the hidden node number for SGBP is also 100, but the learning rate is 0.002 in order to achieve the best tracking performance.

3.2 Simulation II for time-varying system

In order to further test the performance of the proposed approach, a time-varying system based on the engine model of Simulation I is also conducted. In this simulated engine, a time-varying factor v_t is introduced.

$$y_{k+1} = \frac{0.2 \sin y_k + v_t \times 3.5(9 - u_k)}{14.7} \tag{17}$$

The time varying factor v_t is set as 0.95, then the engine system will vary slightly at each step. The same reference and parameters in Simulation I are used, so that the results of the two simulations can be compared easily.

The results of Simulation II are given in Fig. 9 for SGBP, Fig. 10 for ITF-OSELM without forgetting factor

respectively. It can be seen that, the actual lambda for both approaches can also track the reference after the first several steps, even the varying factor is introduced. Comparing Figs. 6 and 8, the time varying system spends almost the same number of steps when the reference is changed at the first several steps. That means for time varying system, both SGBP and ITF-OSELM have powerful adaptive capability, but the ITF-OSELM based approach has better tracking performance.

For more evaluation, a forgetting factor (0.98) is introduced into ITF-OSELM, and the simulation results are given in Fig. 11. It can be seen that the tracking performance of ITF-OSELM with forgetting factor is better than that without forgetting factor.

In these two simulations, ITF-OSELM demonstrates its powerful online learning capability, so that the unknown system can be approximated just using several steps, and the ITF-OSELM based AFR controller has good tracking performance as well.

4 Experimental study on a real engine

The results of above simulations show the powerful online learning capability of ITF-OSELM and better adaptive control performance. However, the engine model is still simple and out of sensor noise and signal delay. To further verify the proposed ITF-OSELM based adaptive control for AFR regulation. A system identification and experiment on a real engine were also conducted. Different from the above simulations, the experiment on the real engine is more complicated and the environmental condition will change as the engine is operating. Moreover, the lambda

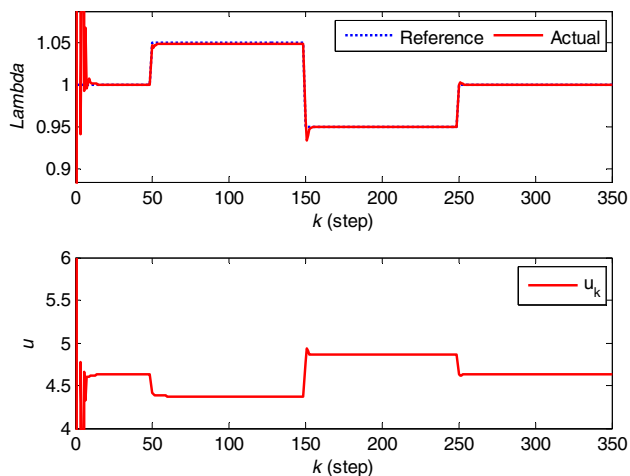


Fig. 11 Control performance of ITF-OSELM with forgetting factor (0.98) in Simulation II

sensor for providing the feedback lambda signal, y , has a noticeable delay. Two experiments were carried out. One is to test the online system identification of ITF-OSELM and the tracking performance of the proposed controller. The other experiment is to test the adaptive capability when the engine operating condition is changed. Furthermore, the engine built-in controller (i.e., look-up table and PID) is also applied for comparison.

4.1 Experimental setup

A Honda Integra DC5 sport car with a high performance electronic controlled, water-cooled, 4-cylinder gasoline engine is employed as shown in Fig. 12. The specification of the test engine is provided in Table 1. A MoTeC M400 programmable ECU, as shown in Fig. 13, is used as the base controller to maintain the basic engine operation.

To implement the proposed ITF-OSELM based adaptive AFR controller, three national instrument (NI) devices were used. A NI 9215 module was used to measure analog signal, a NI 9263 module was used to send analog signal, and a NI



Fig. 12 Honda K20A test engine

Table 1 Engine specification

Engine	Honda K20A—Type R
Type	Water-cooled, four-stroke, DOHC i-VTEC
Cylinder arrangement	Inline four-cylinder, transverse
Bore and stroke	86 × 86 mm
Displacement	1998 cc
Compression ratio	11.5:1
Valve train	Chain drive, 16 valves
Maximum power	160 kW at 8000 rpm
Maximum torque	196 N m at 7000 rpm
Type of fuel injection system	Port injection



Fig. 13 MoTeC M400 ECU for engine basic control

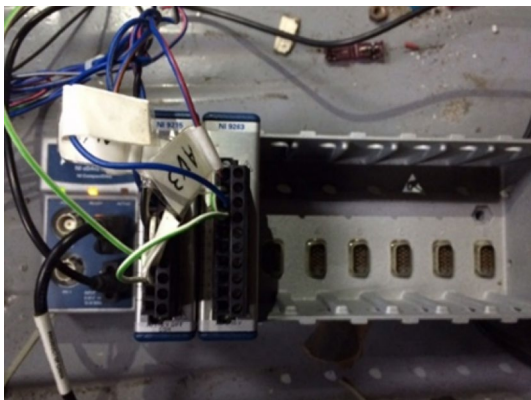


Fig. 14 NI devices for controller implementation

cDAQ-9178 chassis was employed as signal transmission between the NI module and computer. These NI devices were assembled and connected with the ECU as shown in Fig. 14.

A LabVIEW 2012 software is installed on a computer, which was a graphical programming platform for the user to control the NI devices. In other words, the NI devices serve as the interface between the engine signal and the computer, and LabVIEW program serves as the interface between the computer and users. In LabVIEW, a MATLAB plugin is available such that the MATLAB script can be embedded directly. A “MATLAB Script Node” module was used to embed the ITFO-ELM based controller in the LabVIEW program. The diagram of experiment setup is shown in Fig. 15 and the interface between LabVIEW and NI devices is given in Fig. 16. For controlling the fuel injection, instead of sending control signal directly to the

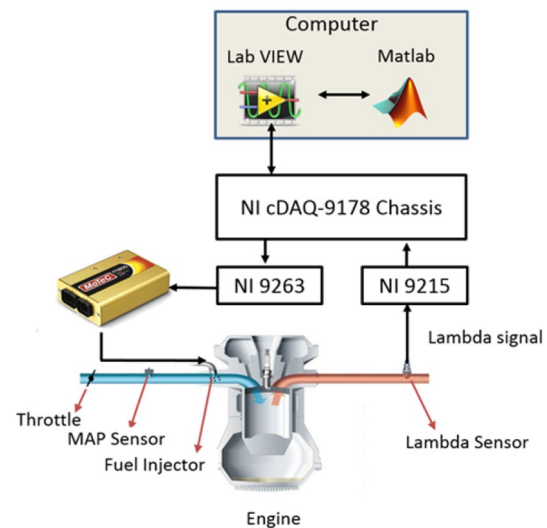


Fig. 15 Schematic diagram of experimental setup

engine fuel injectors, the control signal from the LabVIEW is sent to the programmable ECU and the ECU then converts the given control signal into the control signal of the fuel injectors.

With this program and the NI devices, the designed controller was successfully implemented on the engine. Then, the effectiveness of the controller could be tested and evaluated through this hardware and software system. In the implementation of ITF-OSELM based adaptive control, the hidden node number is 100, the regularization factor is 0.001, the forgetting factor is 0.98 and the activation function is sigmoid function, because they can lead to good tracking performances in Simulations I and II.

4.2 Experimental procedure

In the experiment, the control input from LabVIEW is fuel injection time and the engine output is lambda. At each control step, the fuel injection time and the lambda of last step is collected as the input of the ELM model as shown in Fig. 3 and the lambda of the current step is collected as the target value for training the ELM model. Then the updated ELM model is used to calculate the control input for the next step using Eq. (2). With the continual updating of ELM model with the online data, it can rapidly capture the dynamic behaviors of the engine AFR system and thus precisely control the actual lambda to the desired lambda.

Two different tests are performed to evaluate the effectiveness of the proposed approach.

Test I is designed to evaluate the tracking ability of the adaptive controller for the change of reference target, in which the operating condition is not changed. That means the dynamic system does not vary or changes slightly.

Fig. 16 Interface between LabVIEW and NI devices

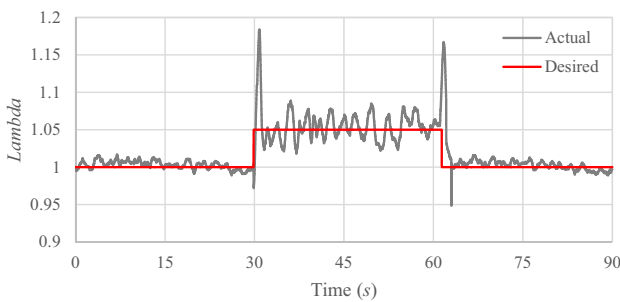
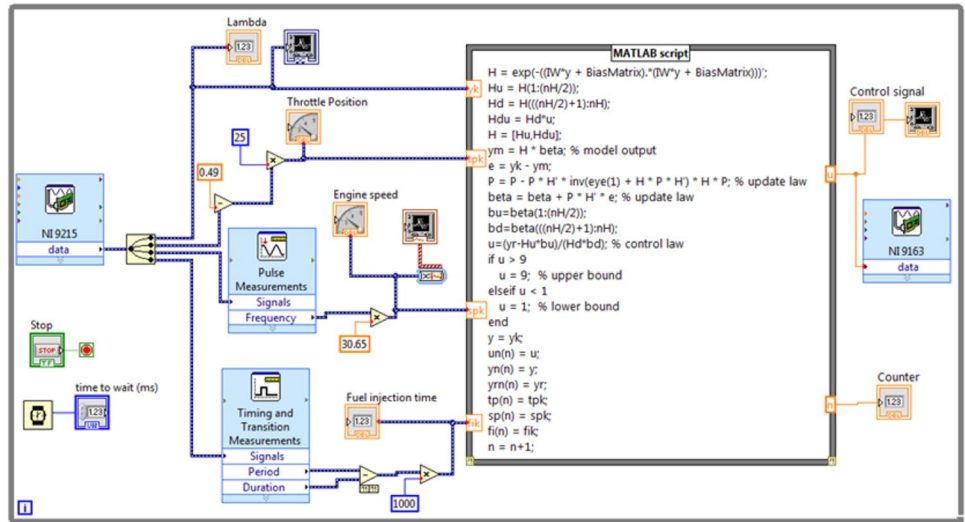


Fig. 17 Tracking performance of ITF-OSELM based controller

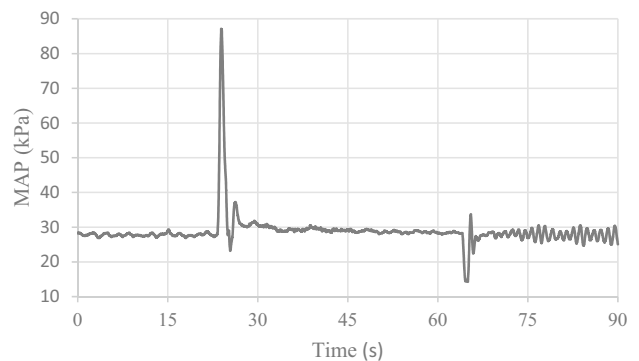


Fig. 19 Sudden changes of MAP

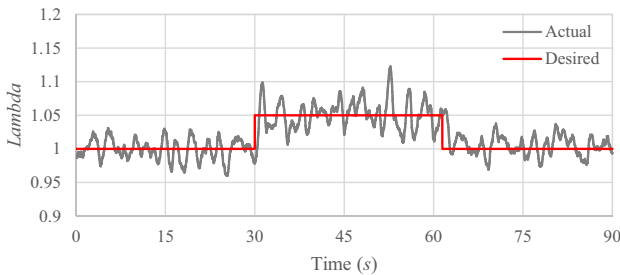


Fig. 18 Tracking performance of the engine built-in controller

Test II is designed to evaluate the tracking ability of the adaptive controller for the change of engine operating condition. It means the operating condition will change and thus the dynamic system also changes.

4.3 Experimental results of Test I

In Test I, the engine is started and warmed up until steady state is reached, which is indicated by the intake air temperature and engine temperature. Test I refers to an idle speed test,

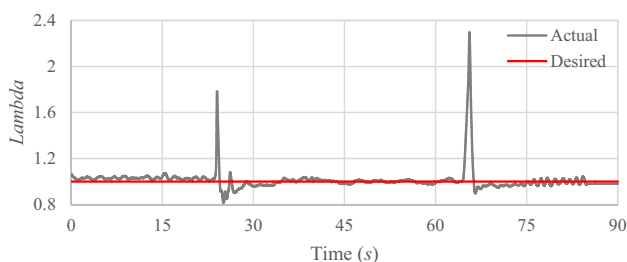
so the throttle position is fixed at 0%. In the beginning, the desired lambda value stays at a common value of 1.00 (maximum conversion efficiency of the three-way catalytic converter) for a period of time. Then the desired lambda value is changed to 1.05 (for best brake specific fuel consumption) for about 30 s and finally changed back to 1.00 so that the tracking performance could be reasonably evaluated.

The results of this test are given in Figs. 17 and 18 respectively for both the ITF-OSELM based approach and the engine built-in controller. The tracking performance in the real engine cannot achieve a perfect tracking performance as compared to the simulation because of the complexity of real engine and existence of sensor noise and signal delay. However, the proposed controller can still immediately adjust the fuel injection time to track the desired lambda with a minor fluctuation, when the desired lambda value is changed either from 1.00 to 1.05 or from 1.05 to 1.00. Anyway, the proposed approach can achieve an acceptable performance level.

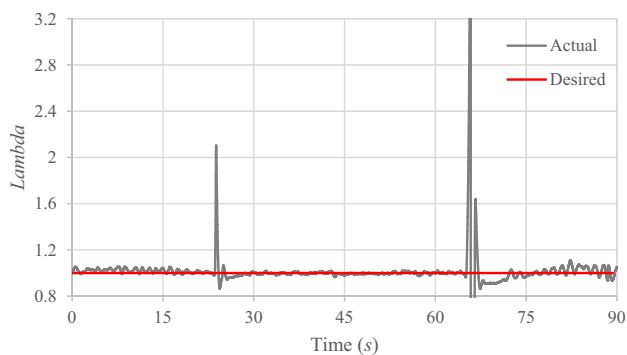
4.4 Experimental results of Test II

In Test II, the engine is warmed up until steady state. The manifold absolute pressure (MAP) is then changed by switching on/off V-TEC which means the valve timing and lift are suddenly changed. The changing of MAP is given in Fig. 19, in which the operating condition is changed at the 20th and 65th seconds, and the dynamic system is altered accordingly. The control objective is to hold the actual output to the desired reference 1.0.

The experimental results of this test are given in Fig. 20 for both the proposed approach and the engine built-in controller. For the abrupt change of MAP at the 20th second and the 65th second, the dynamics of AFR system alters a lot and it needs to take a short period of time for the proposed controller to adapt to the changed system. It can be seen that even when the MAP varies from 85 to 30 kPa, the proposed controller can still successfully regulate the fuel injection time to achieve the desired lambda. Moreover, the response time and the tracking performance are within an acceptable range. It is noteworthy that the lambda obtained by the engine built-in controller around the 65th second is too lean that almost stalls the engine whilst the proposed approach does not have this problem. Besides, the proposed approach has less tracking error than the engine built-in controller, when MAP is changed abruptly.



(a) ITF-OSELM based controller



(b) Engine built-in controller

Fig. 20 Tracking performance comparison between the proposed approach and engine built-in controller

To quantify the control performance of the controllers, the sum of absolute errors (SAE) of the control results are calculated using the following equation:

$$SAE = \sum_{t=0}^{End\ of\ test} |\lambda(t) - \lambda_d(t)|. \quad (18)$$

where $\lambda(t)$ and $\lambda_d(t)$ are the measured and desired lambda values at the time t .

The SAE of Test I for the proposed approach is 92.12, and 132.07 for the engine built-in controller. The SAE of Test II for the proposed approach is 333.47, and 465.41 for the engine built-in controller. By averaging the errors of these two cases, the proposed approach outperforms the engine built-in controller by 29.3%. This significant result, especially for such tough control tasks, shows that the proposed approach controller is very useful and effective for AFR control.

4.5 Discussion

The proposed ITF-OSELM based adaptive controller for engine AFR regulation has good tracking performance for the system without varying or with slow varying. The experimental results of Test I and Test II show the controller can successfully track the desired reference for this situation. It is noteworthy that the result in Test I is obtained without any prior training, whereas the result in Test II is obtained after taking the online training in Test I, so the tracking performance of ITF-OSELM at abrupt change in Test II is better than that in Test I. Overall, the results of both tests verify that ITF-OSELM has powerful online learning capability to identify the nonlinear dynamic system. This shows that the proposed approach is feasible for time-varying systems.

5 Conclusions

In this paper, an ITF-OSELM based adaptive controller for AFR regulation is proposed. The unique feature of the proposed approach is that no engine AFR model and training data are required in advance. So far, there is no exact engine model available yet, and hence many model-based control approaches are practically very hard to be applied to engine control. In the proposed approach, ITF-OSELM is employed to identify the complicated engine AFR dynamics in an online manner. Therefore, no training data is required in advance. To avoid frequently retraining of engine AFR model, which may cause engine stall, the ITF-OSELM used in this paper does not need a mechanism for change detection which is also different from the original ITF-OELM in the author's previous work. In addition, the proposed

approach is relatively easy to be implemented as compared to other existing approaches.

In this research, single neural network online trained by ITF-OSELM is used to approximate two unknown functions, and then the learnt model by ITF-OSELM is used to calculate the control signal, so that the system output can track the desired reference.

In order to verify the proposed approach, both simulations on virtual engines and experiments on a real engine were conducted. The simulation results show that the ITF-OSELM based adaptive controller could achieve excellent tracking performance. Therefore, the controller was confidently implemented on a real test engine. Experiments were then set-up in which NI devices, LabVIEW, MATLAB and MoTeC programmable ECU were utilized. The experimental results demonstrate that the designed adaptive AFR controller can successfully track the desired reference when the AFR system is fixed or varying slowly.

The system of Eq. (1) for ITF-OSELM to approximate to is a general system. That is to say, the proposed ITF-OSELM based adaptive control strategy can be applied to many other similar applications.

Acknowledgements This study is funded by the University of Macau Research Grant under Grant numbers: MYRG2017-00135-FST, MYRG2016-00134-FST and MYRG2016-00212-FST and the Science and Technology Development Fund of Macau S.A.R. under Grant numbers: 050/2015/A, 012/2015/A and 015/2015/AMJ.

References

- EndTuning (2007) Air fuel ratios and stoichiometry. <http://www.endtuning.com/afr.html>. Accessed 10 Jan 2017
- Kumar M, Shen T (2015) Estimation and feedback control of air-fuel ratio for gasoline engines. *Control Theory Technol* 13:151–159. <https://doi.org/10.1007/s11768-015-4148-9>
- Ranga A, Surnilla G, Thomas J, Sanborn E, Linenberg M (2017) Adaptive algorithm for engine air–fuel ratio control with dual fuel injection systems. SAE technical paper 2017-01-0588. <https://doi.org/10.4271/2017-01-0588>
- Franceschi EM, Muske KR, Jones JCP, Makki I (2007) An adaptive delay-compensated PID air fuel ratio controller. SAE Paper no. 2007-01-1342. <https://doi.org/10.4271/2007-01-1342>
- Ebrahimi B, Tafreshi R, Masudi H, Franchek M, Mohammadpour J, Grigoriadis K (2012) A parameter-varying filtered PID strategy for air-fuel ratio control of spark ignition engines. *Control Eng Pract* 20:805–815. <https://doi.org/10.1016/j.conengprac.2012.04.001>
- Choi SB, Hedrick JK (1998) An observer-based controller design method for improving air/fuel characteristics of spark ignition engines. *IEEE Trans Control Syst Technol* 6:325–334. <https://doi.org/10.1109/87.668034>
- Yoon P, Sunwoo M (2001) An adaptive sliding mode controller for air-fuel ratio control of spark ignition engines. *Proc Inst Mech Eng Part D J Automob Eng* 215:305–315. <https://doi.org/10.1243/0954407011525548>
- Pace S, Zhu GMG (2012) Sliding mode control of both air-to-fuel and fuel ratios for a dual-fuel internal combustion engine. *ASME Trans J Dyn Syst Meas Control* 134:1–12. <https://doi.org/10.1115/1.4005513>
- Ebrahimi B, Tafreshi R, Mohammadpour J, Franchek M, Grigoriadis K, Masudi H (2014) Second-order sliding mode strategy for air-fuel ratio control of lean-burn SI engines. *IEEE Trans Control Syst Technol* 22:1374–1384. <https://doi.org/10.1109/TCST.2013.2281437>
- Wu HM, Tafreshi R (2018) Fuzzy sliding-mode strategy for air–fuel ratio control of lean-burn spark ignition engines. *Asian J Control* 20(1):149–158. <https://doi.org/10.1002/asjc.1544>
- Pan YP, Yu HY (2016) Composite learning from adaptive dynamic surface control. *IEEE Trans Autom Control* 61(9):2603–2609. <https://doi.org/10.1109/TAC.2015.2495232>
- Liu H, Pan YP, Li SG, Chen Y (2017) Adaptive fuzzy backstepping control of fractional-order nonlinear systems. *IEEE Trans Syst Man Cyber Syst* 47(8):2209–2217. <https://doi.org/10.1109/TSMC.2016.2640950>
- Gao XH, Wong KI, Wong PK, Vong CM (2016) Adaptive control of rapidly time-varying discrete-time system using initial-training-free online extreme learning machine. *Neurocomputing* 194:117–125. <https://doi.org/10.1016/j.neucom.2016.01.071>
- Rong HJ, Wei JT, Bai JM, Zhao GS, Liang YQ (2015) Adaptive neural control for a class of MIMO nonlinear systems with extreme learning machine. *Neurocomputing* 149:405–414. <https://doi.org/10.1016/j.neucom.2014.01.066>
- Rong HJ, Zhao GS (2013) Direct adaptive neural control of nonlinear systems with extreme learning machine. *Neural Comput Appl* 22(3–4):577–586. <https://doi.org/10.1007/s00521-011-0805-1>
- Rong HJ, Suresh S, Zhao GS (2011) Stable indirect adaptive neural controller for a class of nonlinear system. *Neurocomputing* 74(16):2582–2590. <https://doi.org/10.1016/j.neucom.2010.11.029>
- Wong KI (2017) Machine-learning-based modeling of biofuel engine systems with applications to optimization and control of engine performance. Dissertation, University of Macau
- Liang NY, Huang GB, Saratchandran P, Sundararajan N (2006) A fast and accurate online sequential learning algorithm for feed-forward networks. *IEEE Trans Neural Netw* 17:1411–1423. <https://doi.org/10.1109/TNN.2006.880583>
- Huynh HT, Won Y (2011) Regularized online sequential learning algorithm for single-hidden layer feedforward neural networks. *Pattern Recognit Lett* 32:1930–1935. <https://doi.org/10.1016/j.patrec.2011.07.016>
- Mirza B, Lin Z, Toh KA (2013) Weighted online sequential extreme learning machine for class imbalance learning. *Neural Process Lett* 38:465–486. <https://doi.org/10.1007/s11063-013-9286-9>
- Li XL, Jia C, Liu DX, Ding DW (2014) Adaptive control of nonlinear discrete-time systems by using OS-ELM neural networks. *Abstr Appl Anal* <https://doi.org/10.1155/2014/267609> (Article ID 267609)
- LeCun YA, Bottou L, Orr GB, Müller KR (2012) Efficient backprop. In: Montavon G, Orr GB, Müller KR (eds) *Neural networks: tricks of the trade*. Springer, Berlin, pp 9–48. https://doi.org/10.1007/3-540-49430-8_2
- Li GY (2007) Application of intelligent control and MATLAB to electronically controlled engines (in Chinese). Publishing House of Electronics Industry, Beijing

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.