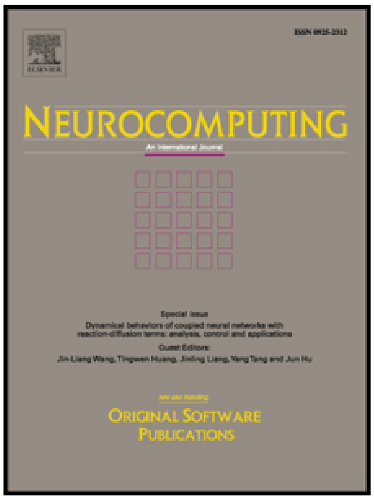
Accepted Manuscript



Online Extreme Learning Machine Based Modeling and Optimization for Point-by-point Engine Calibration

Pak Kin Wong , Xiang Hui Gao , Ka In Wong , Chi Man Vong

PII: S0925-2312(17)31410-8
DOI: 10.1016/j.neucom.2017.02.104
Reference: NEUCOM 18800

To appear in: *Neurocomputing*

Received date: 5 September 2016
Revised date: 23 December 2016
Accepted date: 2 February 2017

Online Extreme Learning Machine Based Modeling and Optimization for Point-by-point Engine Calibration

Pak Kin Wong[a*], Xiang Hui Gao[a], Ka In Wong[a], and Chi Man Vong[b]

[a]*Department of Electromechanical Engineering, University of Macau, Macau*

[b]*Department of Computer and Information Science, University of Macau, Macau*

*Corresponding author. *E-mail address: fstpkw@umac.mo*

Abstract

An online extreme learning machine (ELM) based modeling and optimization approach for point-by-point engine calibration is proposed to improve the efficiency of conventional model-based calibration approach. Instead of building hundreds of local engine models for every engine operating point, only one ELM model is necessary for the whole process. This ELM model is firstly constructed for a starting operating point, and calibration of this starting point is conducted by determining the optimal parameters of the model. This ELM model is then re-used as a base model for a nearby target operating point, and optimization is performed on the model to search for its best parameters. With a design of experiment strategy on the best parameters obtained, new measurements from the target operating point can be collected and used to update the model. By repeating the optimization and model update procedures, the optimal parameters for the target point can be found after several iterations. By using the model of this target point as the base model for another nearby operating point and repeating the same process again, calibration for all the operating points can be done online efficiently. The contribution of the proposed method is to save the number of experiments in the calibration process. To verify the effectiveness of the proposed approach, experiments on a commercial engine simulation software have been conducted. Three variants of online ELM are utilized in the model update process for comparison. The

results show that engine calibration can be carried out with much fewer measurements and time using the proposed approach, and the initial training free online ELM is the most efficient online modeling method for this application.

Keywords: Engine calibration; engine modeling; engine optimization; initial-training-free online extreme learning machine

1. Introduction

Automotive engines are intricate mechanical systems that consist of various actuators and electronic components. Due to the complex and nonlinear behavior of engine dynamics, these engine components are mainly controlled using sets of look-up tables that are stored in the electronic control unit (ECU). These tables, which are also called the engine maps, contain the engine control parameters at different operating points. For better understanding, an example of engine map and ECU is given in Fig. 1, in which a fuel map of an engine is shown. Here in Fig. 1, each cell in the map represents an engine operating point that is defined by engine speed (x-axis) and load (y-axis), and its value represents the amount of fuel to be injected to the engine at that operating point. Obviously, to maximize the engine performance, the values in the engine maps have to be carefully adjusted. The process of determining the optimal settings for these engine maps (i.e. optimal ECU setup) is called engine calibration.

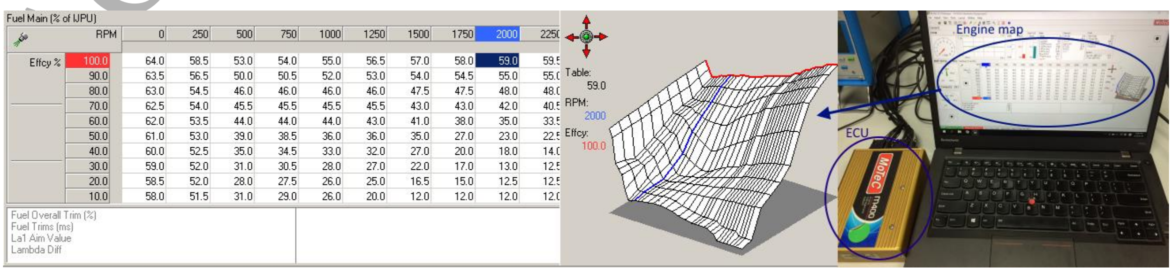


Fig. 1. An example of an engine map in an ECU undertaking a calibration

In the traditional calibration process, calibration engineers determine the optimal settings manually through hundreds of trial-and-error experiments. In each experiment, the engineers adjust the control parameters based on their experience, and then the corresponding engine performances (i.e. responses) are obtained for evaluation via measurement which requires expensive equipment and sensors together with a lot of consumable item and human resource. Thus, the engine calibration has not been an easy task. Nowadays, however, calibration of modern engines has become more challenging even for the most experienced engineers. It is because many more sub-systems have been introduced on today's engines to cope with environmental concerns and different customers' demands, and this increase of adjustable parameters inevitably increases the experimental efforts. To alleviate this problem, model-based calibration approach [1-3] has recently been adopted on modern engines. Instead of running tons of experiments, a Design of Experiment (DoE) [4, 5] procedure is firstly conducted to generate some representative combinations of control parameters so that the engine nature could be captured effectively with much fewer experiments. Based on the DoE framework, data are then acquired from the test engines to build black-box or grey-box models using data-driven methods (e.g., polynomials or neural networks). Optimization is finally performed with the models and optimizer to obtain the optimal settings. Consequently, the amount of resources in measurement can be considerably saved in terms of fuel, time and labor. However, since the optimization is done on the models but not the engines, whether the optimal results obtained are reliable for the real engines depends greatly on the accuracies of the models built. The accuracy or the generalization of the data-driven engine model plays an important role in this approach. Therefore, the global-local model structure [6, 7], which is shown in Fig. 2, has also been utilized to increase the generalization of the engine models. Owing to the fact that the control parameters in the engine maps are associated with engine operating points, local models can be constructed for all the operating points first, and by means of a bilinear interpolation, the local models can be combined to form a global model output, e.g., the global model output for point p' is determined by the area interpolation of the local model outputs of p_1 to p_4. In most cases, this combination of local models is more accurate than a single model that fits all the data in one process.

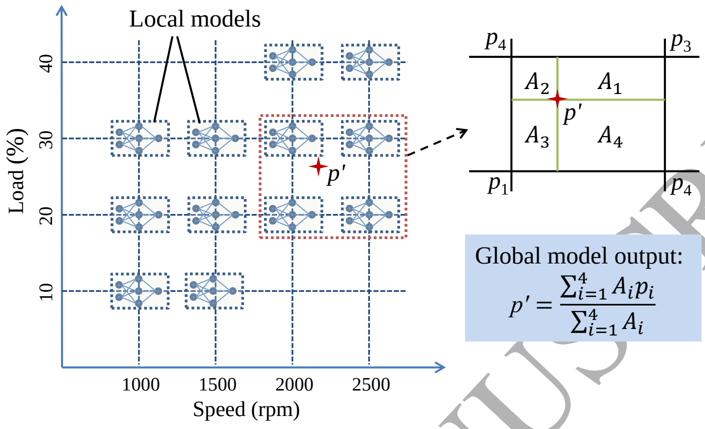


Fig. 2. Global-local engine model structure [6].

Although the model-based approach is more effective than the trial-and-error approach, there are still two major concerns for this offline approach. Firstly, the computational cost for performing the optimization process could be very high as dozens or even hundreds of local models are required for all representative operating points. Secondly, to accomplish the goal of high quality calibration, sufficient data must be collected at every operating point for construction of local models. As the engineers may not know the range of the optimal parameters for some operating points in prior, many unnecessary measurements may have been taken that could make unnecessary expense on the engine test bench. Fortunately, these limitations can be addressed if the model-based calibration approach can be carried out in an online manner. For instance, since the engine behavior of two nearby operating points should be quite similar, a model can be built based on data from one operating point first and updated online using new arriving data when it comes to the next nearby operating point. In this sense, it is not necessary to collect all the data from every operating point and build the corresponding local models in advance of the optimization process. Moreover, if optimization is performed instantly on the model during the online modeling process, the direction to the optimal parameters can be easily observed by tracing the results from the iterative process. Thus, by following the same direction to take new measurements, the usefulness of the data can be guaranteed and the number of unnecessary measurements can be reduced significantly. In view of these advantages, this paper aims to develop a new online modeling and optimization framework for engine calibration to further improve the efficiency of conventional model-based approach.

In order to achieve this objective, it is essential to employ an efficient online learning algorithm that can learn the local engine models incrementally as new measurements are taken. Recent studies [8-10] showed that, among various data-driven methods, online extreme learning machine (ELM) is the most appealing one for learning the dynamic behavior of engines, thanks to its extremely fast learning speed and universal approximation capability. An ELM is a single-hidden-layer feedfoward network (SLFN). As proved in [11] and [12], SLFNs with piecewise continuous activation functions are capable of approximating any continuous functions. One key idea of online ELM is that the parameters (input weights and bias) of the hidden layer need not be tuned in the learning process, so only the output weights of the model need be adjusted when new data is available, resulting in very low computational cost that matches the requirement for this application. In the literature, there already exist some variants of online ELM. The original online ELM is called online sequential extreme learning machine (OS-ELM) [13], but it is only the recursive version of batch ELM [12, 14], which tends to build a model that fits all the data it has seen. Thus, if OS-ELM is used in this application, the idea of local model will be omitted and the generalization of the final model will be poor. An improved version of OS-ELM, namely weighted OS-ELM (WOS-ELM) [15], introduces a forgetting factor to the updating rule so that the old learnt data are gradually ignored. Despite its forgetting behavior, the convergence speed of WOS-ELM for updating the model is slow if there are too much useless or outdated learnt data needed be forgotten. Initial-training-free online ELM (ITF-OELM) [16], which was proposed by the authors of this paper, has the ability to start a learning process in a fully online manner. That means when the operating point changes, a new model can be trained immediately without going through the whole forgetting procedure as compared to WOS-ELM. As a result, ITF-OELM is selected in this study for online modeling of the engine.

Since ITF-OELM was originally developed for adaptive control problems, it cannot be directly applied to the present problem. To extend ITF-OELM to the engine calibration problem, it is necessary to derive an appropriate strategy to integrate with ITF-OELM to decide that what parameters should be tested for new measurements when the calibration for one operating point is finished and the point of interest is moved to the next one. As mentioned before, the engine dynamics should be similar for two nearby operating points. Therefore, one simple way is to use the optimal parameters obtained from the original operating point directly on the next new operating point as the starting parameters, and by measuring the corresponding engine output to update the model and performing optimization on the updated model, a new best set of parameters can be obtained. Then repeating this procedure until the model is accurate enough to reflect the engine output, the final optimal parameters could be determined. In fact, this iterative process is the principle of online optimization. However, as the model is only updated by one measurement each time, if it happens in some cases that the update of model is stuck in the same position during the online learning process, then the updated model is never able to reflect the true engine response of the new operating point. For better explanation, an example is given in Fig. 3, where M1 represents an original engine model, u' is the best parameters on M1, y' is the actual response of the engine for u', and M2 is the model obtained by updating M1 using the new measurement (u', y'). Here, since the optimization is only performed on the model, if the best parameters on M2 is u' again, then the model will just keep on updated by the data point (u', y'); it could never completely converge to the real engine response. In other words, the optimal parameters for the engine at this operating point could never be found but are stuck at u' which is only the optimal of the model. To overcome this issue, a new strategy is also designed in this study to ensure that the model can gradually and completely converge to the engine response of the new operating point. Considering that online ELM can update the model regardless the size of new arrival data, in addition to the best point determined by the optimization on the model, several points around of it could also be utilized for measurement during the iterative model update process. Under this chuck-by-chuck updating manner, the model can converge towards the area of the optimal parameters rather than a single point and thus the update of the model is seldom stuck before the optimal area is reached.

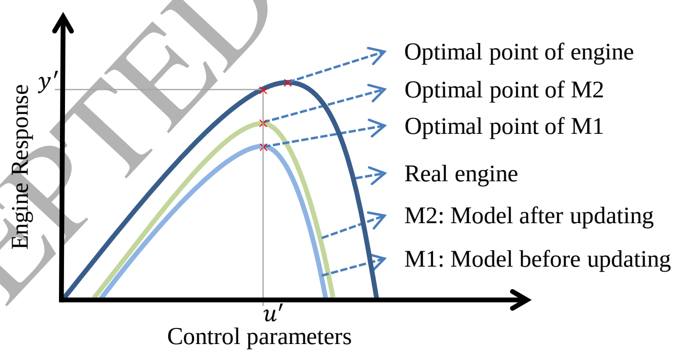


Fig. 3. Limitation of one-by-one model update strategy for online optimization

In short, a novel online ELM based modeling and optimization approach for point-by-point calibration is proposed in this paper. Here point-by-point refers to the scheme that the calibration is done from one operating point to another. For a starting operating point, a local model is constructed using ELM and optimization is performed to calibrate this point.

Then, measurements are taken at the next target operating point based on the optimal parameters obtained from the starting operating point, so that the model can be updated by ITF-OELM. A chunk-by-chunk updating strategy is used for model update to ensure that the area for new measurements can gradually converge to the area of optimal parameters of the target operating point, so that the number of unnecessary measurements can be reduced. When the model is updated sufficiently, the optimal parameters of this target point can be found and this set of optimal parameters can then be used as the starting parameters for the next operating point. Eventually, calibration for all the operating points of the engine can be done in an online manner efficiently.

The organization of this paper is as follows. A brief review of online ELM is provided in Section 2. The proposed online modeling and optimization approach for point-by-point calibration is detailed in Section 3. To verify the proposed approach, experiments on a commercial engine simulation software are conducted and the results are presented in Section 4. The final conclusions are given in Section 5.

2. Brief review of online ELM

This section gives a brief review of the original batch ELM and three online ELM variants, including OS-ELM, WOS-ELM and ITF-OELM, for later use in engine modeling of the proposed approach.

2.1. Batch ELM

ELM is a data-driven method that aims to train SLFNs with less human intervention. Huang et al. [12, 14] proved that only the output weights of an SLFN need to be determined, while the input weights and biases in the hidden layer can be generated randomly and remain fixed.

For an SLFN with L hidden nodes, the output function is:

$$f(\mathbf{x}) = \sum_{i=1}^{L} \beta_i h_i(\mathbf{x}) = \sum_{i=1}^{L} \beta_i G(\boldsymbol{a}_i, b_i, \mathbf{x}) = \boldsymbol{h}(\mathbf{x})\boldsymbol{\beta}, \tag{1}$$

where $G(\cdot)$ is the activation function used in the hidden nodes, $\boldsymbol{a}_i$ is the input weight vector, b_i is the bias, $\boldsymbol{h}(\mathbf{x}) = [h_1(\mathbf{x}), h_2(\mathbf{x}), \dots, h_L(\mathbf{x})]$ is the hidden layer output vector with respect to the input $\mathbf{x}$, and $\boldsymbol{\beta} = [\beta_1, \beta_2, \dots, \beta_L]^{\mathrm{T}}$ is the vector containing the output weights between the hidden nodes and the output nodes.

Now, given a training dataset with N samples, to train the SLFN using ELM is essentially to solve a multiple linear regression problem of the following form:

$$\boldsymbol{H}\boldsymbol{\beta} = \boldsymbol{T}, \tag{2}$$

where $\boldsymbol{H} = [\boldsymbol{h}(\mathbf{x}_1), \boldsymbol{h}(\mathbf{x}_2), \dots, \boldsymbol{h}(\mathbf{x}_N)]^{\mathrm{T}}$ is a $N \times L$ matrix of the hidden layer output, and $\boldsymbol{T} = [t_1, \cdots t_N]$ is the vector of target values. The solution of $\boldsymbol{\beta}$ to Equation (2) is:

$$\boldsymbol{\beta} = \boldsymbol{H}^{\dagger}\boldsymbol{T}, \tag{3}$$

where $\boldsymbol{H}^{\dagger}$ is the Moore-Penrose generalized inverse of matrix $\boldsymbol{H}$. When $\boldsymbol{H}^{\mathrm{T}}\boldsymbol{H}$ is nonsingular, the orthogonal projection method can be used to calculate $\boldsymbol{H}^{\dagger}$, that is:

$$\boldsymbol{H}^{\dagger} = (\boldsymbol{H}^{\mathrm{T}}\boldsymbol{H})^{-1}\boldsymbol{H}^{\mathrm{T}}. \tag{4}$$

To improve the generalization performance and robustness of the solution, a small regularization factor γ is usually employed to the diagonal of $\boldsymbol{H}^{\mathrm{T}}\boldsymbol{H}$:

$$\boldsymbol{\beta} = (\boldsymbol{H}^{\mathrm{T}}\boldsymbol{H} + \gamma \boldsymbol{I})^{-1}\boldsymbol{H}^{\mathrm{T}}\boldsymbol{T}. \tag{5}$$

According to Bartlett's theory [17], this resulting solution tends to have better and more stable prediction performance, as verified in [12, 18].

2.2. Online ELM

Different from batch ELM, online ELM can continuously improve the model accuracy by incrementally learning new data. Several variants of online ELM have been developed with their own merits, and three of them are introduced here.

2.2.1. Online sequential extreme learning machine (OS-ELM)

OS-ELM is the original version of online ELM, which can learn data not only one-by-one but also chunk-by-chunk with fixed or varying chunk size [13]. It consists of two phases: initialization phase and sequential learning phase. In the first phase, a base ELM model is trained using a small chunk of initial training samples. For instance, given an initial training dataset $\boldsymbol{S}_0$ with N_0 training samples, according to batch ELM, if $\boldsymbol{H}_0^{\mathrm{T}}\boldsymbol{H}_0$ is nonsingular, the output weights can be obtained as:

$$\boldsymbol{\beta}^0 = \boldsymbol{P}_0\boldsymbol{H}_0\boldsymbol{T}_0 \tag{6}$$

$$\boldsymbol{P}_0 = (\boldsymbol{H}_0^{\mathrm{T}}\boldsymbol{H}_0)^{-1}. \tag{7}$$

Then, in the second phase, recursive least-squares (RLS) algorithm is used to update the output weights for new arriving training data [13]:

$$\boldsymbol{\beta}^{(k+1)} = \boldsymbol{\beta}^{(k)} + \boldsymbol{P}_{k+1}\boldsymbol{H}_{k+1}^{\mathrm{T}}\left(\boldsymbol{T}_{k+1} - \boldsymbol{H}_{k+1}\boldsymbol{\beta}^{(k)}\right) \tag{8}$$

$$\boldsymbol{P}_{k+1} = \boldsymbol{P}_k - \boldsymbol{P}_k\boldsymbol{H}_{k+1}^{\mathrm{T}}\left(\boldsymbol{I} + \boldsymbol{H}_{k+1}\boldsymbol{P}_{\boldsymbol{k}}\boldsymbol{H}_{k+1}^{\mathrm{T}}\right)^{-1}\boldsymbol{H}_{k+1}\boldsymbol{P}_k, \tag{9}$$

where $k+1$ indicates the $(k+1)th$ arriving training data with k starting from zero, and $\boldsymbol{H}_{k+1}$ is the hidden layer output for the $(k+1)th$ arriving training data.

2.2.2. Weight online sequential extreme learning machine (WOS-ELM)

WOS-ELM is a revised version of OS-ELM, in which a forgetting factor is adopted in the weight updating algorithm for different purposes, such as dealing with class imbalance learning problems [15]. WOS-ELM follows the same initialization phase in OS-ELM, but in the second phase, the updating rule of (9) is revised as follows:

$$\boldsymbol{P}_{k+1} = \frac{1}{\rho}\left(\boldsymbol{P}_k - \boldsymbol{P}_k\boldsymbol{H}_{k+1}^T\left(\rho\boldsymbol{I}_{N_{k+1}} + \boldsymbol{H}_{k+1}\boldsymbol{P}_k\boldsymbol{H}_{k+1}^T\right)^{-1}\boldsymbol{H}_{k+1}\boldsymbol{P}_k\right), \tag{10}$$

where $0 < \rho \leq 1$ is the forgetting factor that gives the more recent arriving data higher contribution to the output weight adjustment. It should be noticed that when $\rho = 1$, the weight updating algorithm is the same as that of OS-ELM.

2.2.3. Initial training free online ELM

In both OS-ELM and WOS-ELM, there is a stint that the number of distinct initial training data must not be less than the number of hidden nodes to guarantee that the hidden layer output matrix $\boldsymbol{H}$ is full rank and the term $\boldsymbol{H}^{\mathrm{T}}\boldsymbol{H}$ is nonsingular. To tackle this issue, ITF-OELM [16] implements the regularization factor in the learning phase of WOS-ELM. Furthermore, ITF-OELM combines both the initialization phase and the sequential learning phase into one online learning process, with the following special model initialization conditions:

$$\boldsymbol{\beta}^0 = \boldsymbol{0} \tag{11}$$

$$\boldsymbol{P}_0 = (\lambda\boldsymbol{I}_L)^{-1}. \tag{12}$$

That means it is not necessary to collect any initial training data before online learning. This advantage is very useful for this application because a new learning process can be started at any time without concerning the availability of initial training data. The update rule of ITF-OELM is the same as Equations (8) and (10).

3. Proposed approach for point-by-point engine calibration

This section contains the details of the proposed online ELM based modeling and optimization approach for engine calibration.

3.1. Problem formulation

Referring to [19], the relationship between the inputs and outputs of an engine at different steady-state operating conditions can be described as:

$$\boldsymbol{y} = F(\boldsymbol{u}, \boldsymbol{p}), \tag{13}$$

where $\boldsymbol{u}$ is the vector of control parameters, the $\boldsymbol{y}$ is the vector of engine performance characteristics, and $\boldsymbol{p}$ indicates the operating conditions.

The purpose of calibration is to search for the optimal input vector $\boldsymbol{u}^*$ that could make the output vector be the optimal output $\boldsymbol{y}^*$, that is,

$$\boldsymbol{y}^* = F(\boldsymbol{u}^*, \boldsymbol{p}). \tag{14}$$

If F is available, then the optimal control parameters $\boldsymbol{u}^*$ can be easily determined by using any optimization method, so the primary step in model-based calibration approach is to approximate F. However, since the engine performance varies more distinctly with the operating conditions than with the control parameters, it is very difficult to obtain a high generalization model of F for calibration purpose. Yet if for each time only one specific operating condition is considered, then Equation (13) can be simplified into:

$$\boldsymbol{y}_p = F_p(\boldsymbol{u}_p), \tag{15}$$

where the subscript p denotes the operating point being considered. It is obvious that approximating Equation (15) is much easier than Equation (13) as the model only needs to find out the relationship between the control parameters and the engine performance without concerning the operating conditions. Thus, to achieve high quality calibration, models are constructed to approximate F_p for different operating points instead of F for all operating points, and the objective is to search for the optimal $\boldsymbol{u}_p^*$ for each operating point p.

3.2. Framework of proposed approach

As F_p varies with different operating points and there are too many operating points for an ECU, dozens or hundreds of models may be required for the whole calibration process. Aiming to further reduce the computational cost and experimental efforts for model construction, a new ELM based modeling and optimization approach is proposed in this study. It is an iterative process as shown in Fig. 4.

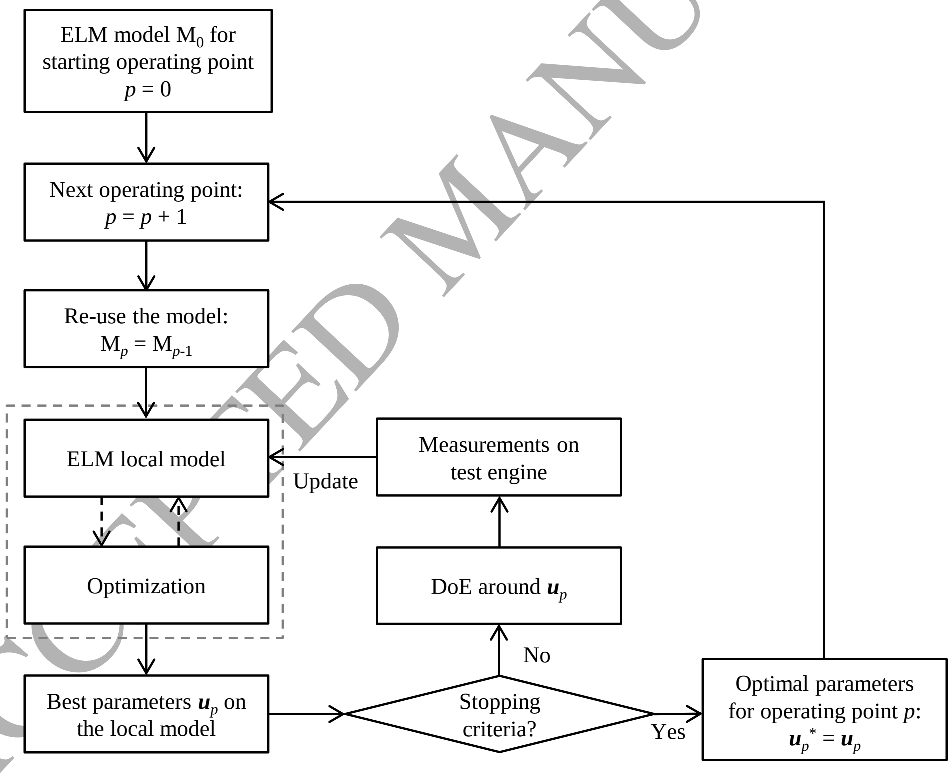


Fig. 4. Workflow of proposed approach for point-by-point calibration

To begin with, a starting operating point is selected and an ELM model is constructed for this point for performing optimization to determine its optimal parameters. Once the optimal settings of this starting point are found, the point of interest is moved to the next operating point. The trained model of the starting operating point is then used as the initial model for the new operating point, and its optimal parameters are as well used as the starting parameters for measurements on the new operating point. As mentioned in the introduction section, several points around the parameters should be utilized for new measurements, so a simple DoE strategy is carried out around the starting parameters to determine what other parameters should be tested. With the new measurements from the engine, the ELM model is updated and optimization is conducted again on the updated model in order to determine the best parameters on this model. Then, by repeating the steps that DoE is performed on the best parameters obtained from the model and the corresponding new measurements are used to update the model, the ELM model can gradually approximate the real engine behavior towards the optimal area, with better and better accuracy. The optimal parameters should soon be determined after several iterations when the stopping criteria is reached. The stopping criteria can be set as the tolerance of the best parameters obtained from the model between two successive iterations. Finally, once the optimal parameters for this operating point are found, the point of interest can be moved to the next operating point, and again, the model of the last operating point is used as the starting model for the new target operating point and its optimal parameters is also used as the starting parameters for the new operating point. This procedure can be continued until the calibration for all representative operating points is completed.

Here within the framework of this proposed approach, three key components must be carefully designed in order to achieve the goal of high quality calibration. They are the online modeling, the online optimization and the simple DoE strategy. The details for these three components are provided in the following sub-sections.

3.2.1. Online modeling

For the purpose of reducing unnecessary measurements, the model of the previous operating point is used as the initial model for approximating the engine behaviors of the target operating point. However, since in some cases the initial model may contain too much old learnt information that is useless or outdated for the new target operating point, it may take a relatively long time to finish the online modeling iterations because the accuracy of the model will not be good enough until all the useless information is forgotten. In order to accelerate the procedure, ITF-OELM is considered in this study, because all the useless information can be omitted in just one step due to its attractive feature that a new learning process can be started immediately at any time when necessary. The procedure for online modeling of the engine behavior at an operating point using ITF-OELM involves on two phases, which are summarized in the following:

Model resetting phase:

(a) Re-use the fixed parameters from the model of the previous operating point, i.e., the hidden node parameters $(\boldsymbol{a}_i, b_i)$, the activation functions $G(\cdot)$ and the regularization factor γ;

(b) Reset $\boldsymbol{\beta}^{(0)} = 0$ and $\boldsymbol{P}_0 = (\gamma \boldsymbol{I})^{-1}$ to start a new learning process.

Model updating phase:

(a) Calculate the hidden layer output $\boldsymbol{H}$ of the arriving engine measurements;

(b) Adjust $\boldsymbol{P}_k$ and $\boldsymbol{\beta}^{(k)}$ using Equations (8) and (10).

When the first set of new measurements arrives, the model resetting phase is performed first so that all the useless information is forgotten immediately. This resetting phase is only conducted once for each operating point. After the model is reset, the model updating phase is carried out to update the model. This model updating phase is then used in all the remaining iterations until the optimal parameters of the engine at the given operating point are obtained.

It can be learnt from model resetting phase (a) that the hidden nodes of the model are kept the same for all operating points in the online modeling process. This can only be done with online ELM because these nodes are independent of the data and always remain fixed even after the model is updated. This is also the reason why online ELM is preferred than other data-driven methods, as some of the data-driven methods update the input weights with the seen data and useless information may thereby be inherited to the next operating point. In addition, ELM allows the use of many different activation functions, so for different engine outputs, different activation functions can be used to further improve its accuracy, whereas some data-driven methods may not have such merit.

3.2.2. Online optimization

In the proposed approach, each time when the model is updated, optimization is performed instantly to determine the best parameters of the model so that they can be tested on the engine to obtain new measurements for another round of model update. This iterative loop is indeed an online optimization process. Comparing with offline optimization, online optimization can ensure that the measurements on the engine are all meaningful as these measurements are taken towards the optimal area. To carry out the online optimization process, two things are required: an objective function and an optimization method.

Since engine calibration is a multi-objective and multi-variable problem and the optimal engine performance outputs may conflict with each other, the weighted-sum method is adopted for the design of objective function so that the users can adjust the trade-off between each engine performance output according to their needs. Hence, the following objective function is considered:

$$\boldsymbol{u}_p = \text{argmin} \sum_{i=1}^{m} w^i \left(\frac{y_t^i - \hat{y}^i}{y_t^i}\right)^2, \tag{16}$$

where m is the total number of engine performance outputs to be optimized, y_t^i is the desired value for the ith engine performance output, $\hat{y}^i$ is the prediction from the engine model, and w^i is the user-defined weight for the ith engine performance output.

To solve Equation (16) for the best parameters $\boldsymbol{u}_p$, traditional gradient-based optimization methods may not be suitable as the prediction model for the engine performance outputs is a highly nonlinear data-driven model that those gradient-based methods can easily fall into local minima. A famous meta-heuristic optimization method, particle swarm optimization (PSO) [20], is therefore employed in this study to search for the optimal parameters as it does not need to take any gradient information of the problem while still being able to determine the optimal sets efficiently.

It is worth mentioning that in engine calibration problems, the best values for some engine performance outputs, such as the maximum engine torque and the minimum exhaust emissions, are often unknown. Therefore, for those performance outputs without exact desired values, y_t^i can be set to a higher value (for maximizing) or a lower value (for minimizing) that the engine cannot achieve, so as to provide a direction for the optimization method to work on.

3.2.3. Simple DoE strategy for parameter selection

In traditional offline calibration, DoE is the first step of the whole calibration process. It mainly refers to the design of plans on what combinations of the control parameters should be tested on the engines in order to reduce the amount of experiments required for high degree-of-freedom engines. For online engine calibration with the proposed approach, a simple DoE strategy is also designed to determine what parameters should be tested for model update in each iteration.

As indicated in the Fig. 3, if only the measurement of the best parameters is used to improve the model accuracy, the model tends to be updated towards this data point of best parameters and easily gets stuck at this data point. To solve this issue, for each parameter, the deviations from its best value in both sides are also utilized for measurement so that the model can tend to be updated towards the area of the best parameters. Under this simple DoE strategy, there are totally $(2n+1)$ data points required for measurement in each iteration, where n is the number of the control parameters needed be optimized. To demonstrate this strategy clearly, an example is shown in Fig. 5.

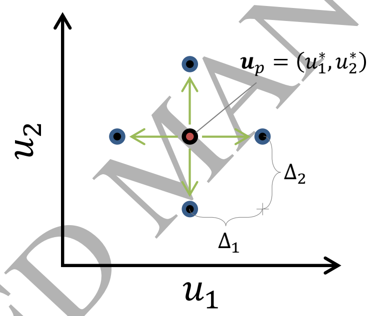


Fig. 5. DoE for measurement around the best parameters

In this example, two control parameters, u_1 and u_2, are considered, i.e., $\mathrm{n}=2$ and the total number of measurements required in each iteration is 5. The point in the center represents the tentative position of the best parameters, $\boldsymbol{u}_p = (u_1^*, u_2^*)$, found on the model, and the four points around, which are $(u_1^* + \Delta_1, u_2^*)$, $(u_1^* - \Delta_1, u_2^*)$, $(u_1^*, u_2^* + \Delta_2)$ and $(u_1^*, u_2^* - \Delta_2)$ respectively, represent the position of other measurements to be tested. If $\boldsymbol{u}_p$ is the not the optimal for the real engine, then the performance of at least one of the other four measurements around should be better than that of the center one. Thus, when these measurements around are used to update the model, not only the model accuracy can be improved, but the best parameters of the model can also be moved towards the optimal area of the real engine response.

3.3. Summary of proposed approach

In brief, the proposed approach (1) regards the model and the optimal of the last operating point as the starting information for the next operating point; (2) employs ITF-OELM to update the model online when new measurements of the target operating point are available; (3) takes advantage of online optimization to guide the collection of new measurements that mainly land in the interesting area; and (4) utilizes a simple DoE strategy to decide what parameters should be tested for new measurements on the target operating point. It possesses the following advantages:

i. The model of the last operating point is utilized so that the computational cost can be eased.
ii. Updating the model with measurements of the target operating point gradually improves the model accuracy for precise calibration.
iii. Online optimization guarantees that the measurement in each iteration is useful with respect to the optimization objective, so the number of unnecessary measurement can be significantly reduced.
iv. The simple DoE strategy around the best parameters avoids the model being stuck in the same position while ensuring that the model converges towards the true engine response, so that the optimal parameters for real engine can be found.

4. Experiments

To verify the effectiveness of the proposed approach, three experiments considering different calibration objectives are designed, in which the model of a starting operating point is re-used for the calibration of a target operating point. Details of the experiments and the corresponding results are presented in this section.

4.1. Experimental scenarios

The scenarios of the three experiments are summarized shown in Table 1. In experiment A, calibration of engine air-fuel ratio (AFR) is conducted, in which the target is to regulate the lambda value to 1 as close as possible in order to achieve maximum conversion efficiency of the three-way catalytic converter [9]. In experiment B, calibration of engine torque is conducted, where the objective is to maximize the engine torque. In this case, the optimal value of maximum torque is unknown. Finally, in experiment C, the objective is to optimize both the engine AFR and torque simultaneously. The optimal values of these two engine outputs are potentially conflicting, so this experiment mainly aims to check whether the proposed approach can also deal with the multi-objective engine calibration problem.

Table 1. Experimental scenerios

Experiment	Objectives
A	Target lambda equal to 1
B	Maximum engine torque
C	Lambda as close as to 1 & maximum engine torque

In the experiments, two operating points were selected for demonstration purpose, one of which is the source operating point being used as the starting point, while the other is the new target operating point to be calibrated. These selected operating points are defined by throttle opening and engine speed, as shown in Table 2. Moreover, two important control parameters, namely the fuel injection time and the ignition time, were chosen for calibration because of ease of illustration, as these two parameters greatly affect the engine AFR and torque. It should be noted that this proposed approach can also deal with more control parameters easily; for more parameters only the number of input nodes of ELM need be increased.

Table 2. Selected operating points

Operating point	Throttle opening (%)	Engine speed (RPM)
Source	25	3000
Target	40	4000

The proposed approach was implemented using Simulink in MATLAB. The experiments were then carried out by co-simulating the proposed approach with a popular commercial engine simulation software called GT-Power. All these were executed on a computer with Intel Core i5-3470 CPU and 4 GB ram under Windows 7 operating system platform.

4.2. Experimental procedure

The purpose of the experiments is to evaluate how well the proposed approach can perform for determining the optimal settings for the target operating point based on the information from the source operating point. The procedure for each of the experiments is the same, which is summarized as follows. At first, the conditions for the source operating point are set in GT-Power so that data can be collected from the simulated engine to create an ELM model. PSO is then used on the model to obtain the optimal settings for this operating point. After that, the operating point is changed to the target operating point by changing the operating conditions in GT-Power. The ELM model from the source operating point is then re-used for calibration of this target operating point. By sending the optimal parameters of the source operating point to the simulated engine, the first set of measurements at the target operating point can be obtained. With the new sets of measurements from GT-Power and by following the steps of the proposed approach, the optimal parameters of the target operating point can be determined.

4.3. Parameter settings

Three versions of online ELM are compared in each of the experiments, and their parameters are provided in Table 3. Since the performance of ELM is sensitive to the activation function, to improve the accuracy, the ELM models for engine AFR (lambda) and torque are respectively constructed using different activation functions.

Table 3. ELM model parameters

ELM variants	Number of hidden nodes	Activation function		Regularization factor	Forgetting factor
		Lambda	Torque		
OS-ELM	100	Sigmoid	Sine	-	-
WOS-ELM				-	0.9
ITF-OELM				0.0001	1.0

The optimization algorithm, PSO, also requires some user-defined parameters. To maintain the computational speed while ensuring a reliable results, the parameters of PSO are selected by trial-and-error, as summarized in Table 4.

Table 4. Parameters of PSO

Parameters	Population	Self-adjustment	Social-adjustment	Max-iteration
Value	50	2	2	100

In addition, experiment C involves multiple objectives, so in the objective function for experiment C, weights should be assigned to each of the two objectives. For demonstration purpose, the weights for both lambda and torque are set to be 0.5.

4.4. Calibration of source operating point

The first step of the calibration is to construct the models of the source operating point. 77 steady-state measurements have therefore been collected from GT-Power, in which the two control parameters, fuel injection time and ignition timing, were varied from 6 to 12 milliseconds (ms) and 5 to 15 degrees before top dead center (°BTDC) respectively. With the measurements, the models for both lambda and torque were constructed, and the resulting models are shown in Fig. 6 and Fig. 7 respectively.

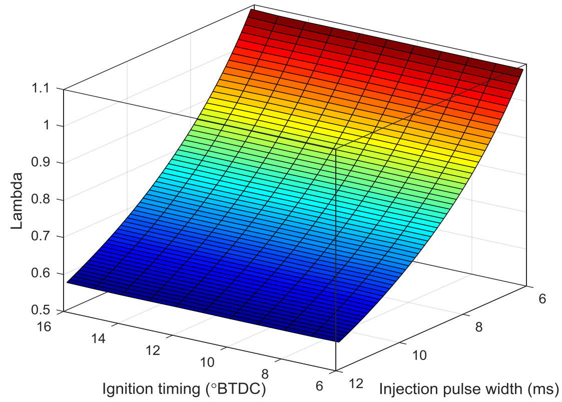


Fig. 6. ELM lambda model of the source operating point

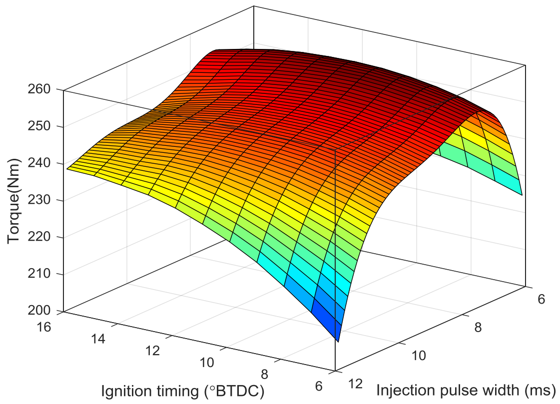


Fig. 7. ELM torque model of the source operating point

Applying PSO to these models and the objective function, the optimal parameters obtained for experiments A, B and C are (10.5 °BTDC, 6.67 ms), (9.56 °BTDC, 7.36 ms) and (10.3 °BTDC, 6.92 ms) respectively.

4.5. Calibration of target operating point

After the calibration of the source operating point is done, the calibration for the target operating point is conducted. Beginning with the models of the source operating point, the best parameters for the target operating point in the first iteration can be obtained and used as the starting parameters. This first set of parameters should be the same as the optimal settings of the source operating point since the model has not been updated yet. Then, following the simple DoE strategy presented in Section 3.2.3, the measurements around the starting parameters are collected for model update. The parameter Δ_1 for ignition timing is set to be 1°, and the parameter Δ_2 for injection pulse width is set to be 0.2 *ms*. The same values of Δ_1 and Δ_2 are used for all the three experiments. The choices for Δ_1 and Δ_2 are actually subject to the decision of the users. Optimal selection of the deviation is left for future study. The calibration results of target operating point are given in the following sub-section.

4.5.1. Results of experiment A

The results of experiment A are shown in Fig. 8. It can be seen that in this experiment all the three versions of online ELM can converge towards the best lambda value, which is 1, but with different speeds. OS-ELM, which tends to build a model that can fit to all seen data, fails to eliminate the error even after 11 iterations. It should be noted that, for two control parameters, five measurements of the target operating point are gathered in each iteration to update the engine model, so totally 50 measurements have been used to update the model using OS-ELM, but the results are still not satisfactory with these amount of new measurements. On the other hand, WOS-ELM, with the use of forgetting factor, can gradually reach the target lambda value in around 8 iterations, but it is still much slower than ITF-OELM, which can achieve the target lambda value in just 3 iterations. This demonstrates that the model of the source operating point may contain too much useless information such that WOS-ELM needs to take 5 more iterations before the useless information is forgotten. Furthermore, as more iteration it takes, the more experiments are necessary. Hence, this experiment also demonstrates that ITF-OELM is more efficient than the other two online ELMs in terms of cost and time for this application.

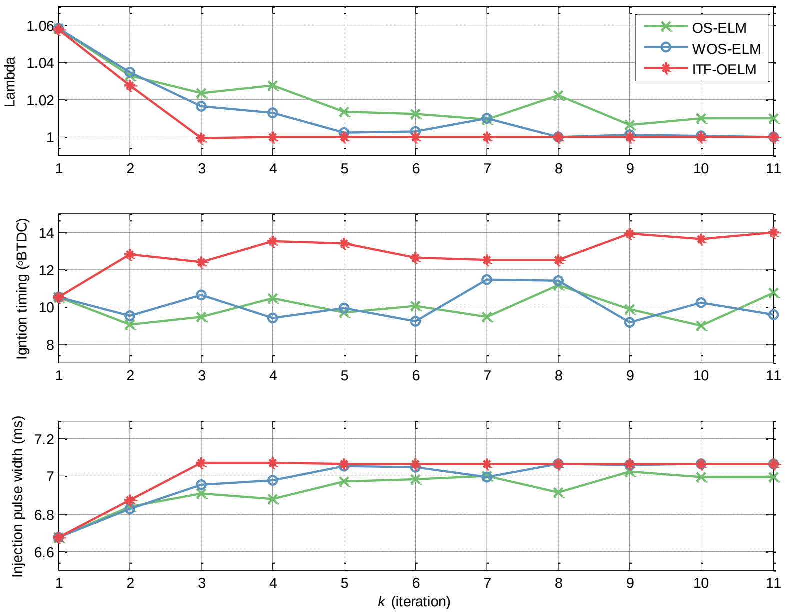


Fig. 8. Results of experiment A

4.5.2. Results of experiment B

The results of experiment B are given in Fig. 9, which shows that the optimal value from all the three ELM variants can converge towards an increasing torque, with a similar speed. It can be seen that, among these optimal results, the highest engine torque is achieved by the approach based on ITF-OELM. This once again shows the superior of ITF-OELM for this application. Nevertheless, comparing with experiment A, the ITF-OELM based calibration takes about 3 more iterations to finish the convergence. This can be explained by the fact that the behavior of engine torque is much more complicated and nonlinear than that of lambda, as shown in Fig. 6 and Fig. 7 previously.

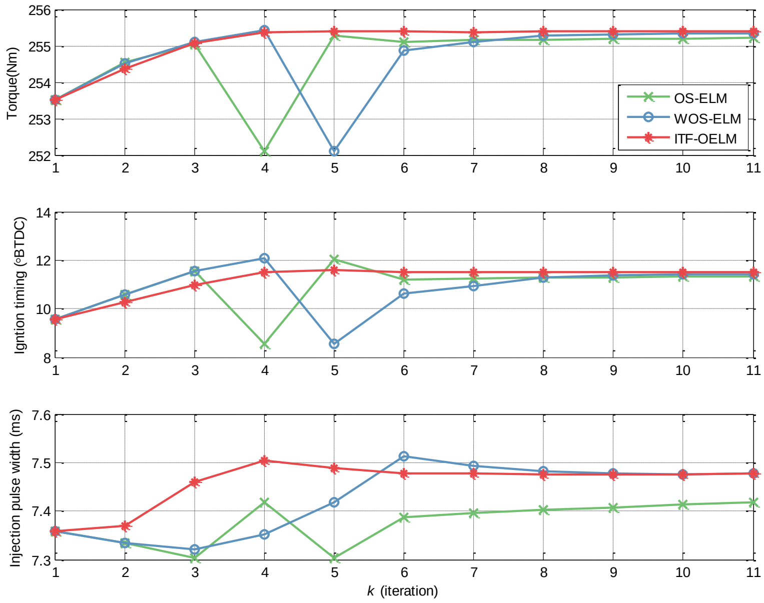


Fig. 9. Results of experiment B

4.5.3. Results of experiment C

The results of experiment C under the aforementioned weight setting are presented in Fig. 10. It can be observed that, comparing with single-objective calibration as provided in the previous two experiments, the multi-objective calibration here needs to take much more iterations to reach convergence. Moreover, the optimal results achieved in both lambda and torque in this case are also lower than that of single objective one in the previous cases. This is mainly because the two performance outputs are conflicting with each other (e.g. maximum torque usually occurs at lambda value of about 0.95). This is also the reason why user-defined weights are required in the objective function, since some may prefer higher torque rather than a better lambda.

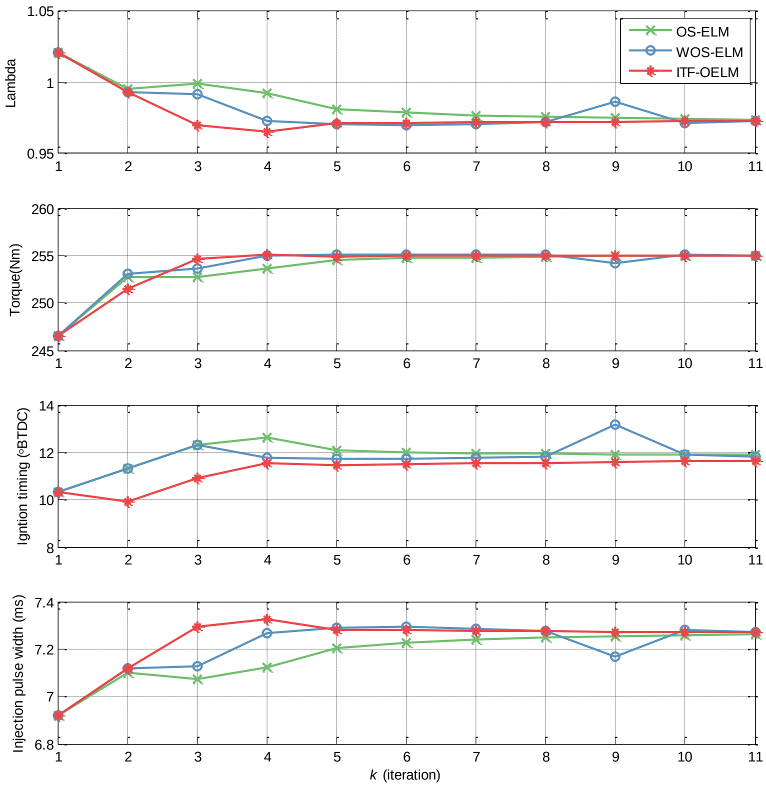


Fig. 10. Results of experiment C

4.6. Discussion of experimental results

All in all, the experimental results from Figs. 8-10 reveal that:

1. Using the best parameters of the source operating point as the starting parameters for the target operating point is actually feasible and can ensure that the measurement is near the optimal area of the target operating point, because the engine outputs of the first iteration, which are obtained by the best parameters of the source operating point, is quite close to the final optimal output achieved by the approach.
2. ITF-OELM has the fastest convergence speed among the three online ELM variants, and can achieve the best performance within just several stable iterations.

3. For multi-objective optimization, different objectives are traded off against each other through the user-defined weights, and it is almost impossible for each objective to achieve its optimal performance as compared to the single objective optimization of each objective alone.

5. Conclusions

In order to achieve high quality calibration with fewer measurements, lower cost and less time, a novel online ELM based modeling and optimization approach for point-by-point calibration was proposed in this paper. An ELM model was firstly constructed for the starting operating point, and calibration of this starting point was done by performing optimization on the ELM model. This starting model and its optimal parameters could then be used as the base model and starting parameters for the next operating point to be calibrated so that the computational cost as well as the unnecessary measurements that are far away from the optimal area of target operating point could be greatly reduced. With the online optimization process and a simple DoE strategy, the most useful measurements could be collected from the test engines and used to update the ELM models of engine. Along with ongoing iterative updates, the model would eventually become very precise within the optimal area of target operating point. Thus, the calibration of the target operating point could be finished and the same process could be continued for the next operating point until all the operating points could be calibrated. To verify the effectiveness of proposed approach, three experiments were carried out, and three variants of online ELM, namely OS-ELM, WOS-ELM, and ITF-OELM, were utilized in the online modeling process for comparison purposes. The results showed that high quality calibration could be achieved using all the online ELM variants, for both single objective and multiple objectives. In addition, among the three variants, ITF-OELM is the most efficient online modeling method for the proposed approach. With the help of the proposed efficient online calibration approach, it is believed that the development time and cost of automotive engines can be significantly reduced.

Acknowledgements

This study is funded by the University of Macau Research Grant (Grant Nos: MYRG2014-00178-FST, MYRG2014-00083-FST and MYRG075(Y1-L2)-FST13-VCM) and the Science and Technology Development Fund of Macau S.A.R. (Grant No: FDCT/050/2015/A).

References

[1] P.K. Wong, L.M. Tam, K. Li, Automotive engine power performance tuning under numerical and nominal data, Control Eng Pract, 20 (2012) 300-314.

[2] J.D. Martinez-Morales, E.R. Palacios-Hernandez, G.A. Velazquez-Carrillo, Modeling and multi-objective optimization of a gasoline engine using neural networks and evolutionary algorithms, Journal of Zhejiang University-Science A, 14 (2013) 657-670.

[3] P.K. Wong, K.I. Wong, C.M. Vong, C.S. Cheung, Modeling and optimization of biodiesel engine performance using kernel-based extreme learning machine and cuckoo search, Renewable Energy, 74 (2015) 640-647.

[4] K. Röpke, C. von Essen, DoE in engine development, Quality and Reliability Engineering International, 24 (2008) 643-651.

[5] M.R. Kianifar, L.F. Campean, D. Richardson, Sequential DoE framework for steady state model based calibration, SAE International Journal of Engines, 6 (2013) 843-855.

[6] H. Sequenz, Emission Modelling and Model-Based Optimisation of the Engine Control, VDI Verlag, 2013.

[7] R. Isermann, Engine Modeling and Control: Modelling and Electronic Management of Internal Combustion Engines, Springer, 2014.

[8] A. Vaughan, S.V. Bohac, Real-time, adaptive machine learning for non-stationary, near chaotic gasoline engine combustion time series, Neural Networks, 70 (2015) 18-26.

[9] P.K. Wong, H.C. Wong, C.M. Vong, Z. Xie, S. Huang, Model predictive engine air-ratio control using online sequential extreme learning machine, Neural Computing and Applications, 27 (2016) 79-92.

[10] V.M. Janakiraman, X. Nguyen, D. Assanis, Stochastic gradient based extreme learning machines for stable online learning of advanced combustion engines, Neurocomputing, 177 (2016) 304-316.

[11] G.B. Huang, L. Chen, C.K. Siew, Universal approximation using incremental constructive feedforward networks with random hidden nodes, IEEE Trans. Neural Netw., 17 (2006) 879-892.

[12] G.B. Huang, H.M. Zhou, X.J. Ding, R. Zhang, Extreme learning machine for regression and multiclass classification, IEEE Trans. Syst. Man Cybern. Part B-Cybern., 42 (2012) 513-529.

[13] N.Y. Liang, G.B. Huang, P. Saratchandran, N. Sundararajan, A fast and accurate online sequential learning algorithm for feedforward networks, IEEE Trans. Neural Netw., 17 (2006) 1411-1423.

[14] G.B. Huang, Q.Y. Zhu, C.K. Siew, Extreme learning machine: Theory and applications, Neurocomputing, 70 (2006) 489-501.

[15] B. Mirza, Z. Lin, K.-A. Toh, Weighted online sequential extreme learning machine for class imbalance learning, Neural Processing Letters, 38 (2013) 465-486.

[16] X.H. Gao, K.I. Wong, P.K. Wong, C.M. Vong, Adaptive control of rapidly time-varying discrete-time system using initial-training-free online extreme learning machine,

Neurocomputing, 194 (2016) 117-125.

[17] P.L. Bartlett, The sample complexity of pattern classification with neural networks: The size of the weights is more important than the size of the network, IEEE Transactions on Information Theory, 44 (1998) 525-536.

[18] W.Y. Deng, Q.H. Zheng, C. Lin, Regularized extreme learning machine, Proceedings of the 2009 IEEE Symposium on Computational Intelligence and Data Mining, (2009) 389-395.

[19] D. Filev, Y. Wang, I. Kolmanovsky, Learning based approaches to engine mapping and calibration optimization, in: H. Waschl, I. Kolmanovsky, M. Steinbuch, L. del Re (Eds.) Optimization and Optimal Control in Automotive Systems, Springer, 2014, pp. 257-272.

[20] J. Kennedy, R. Eberhart, Particle swarm optimization, Proceedings of IEEE International Conference on Neural Networks, 4 (1995) 1942-1948.

Pak Kin Wong received the Ph.D. degree in Mechanical Engineering from The Hong Kong Polytechnic University, Hong Kong, in 1997. He is currently a Professor in the Department of Electromechanical Engineering and Associate Dean (Academic Affairs), Faculty of Science and Technology, University of Macau. His research interests include automotive engineering, fluid transmission and control, engineering applications of artificial intelligence, mechanical vibration and manufacturing processes for biomedical applications. He has published over 185 scientific papers in refereed journals, book chapters, and conference proceedings.

Xiang Hui Gao received the Master's degree in computer application technology from Hebei University, Hebei, China, in 2007. He is currently working towards the Ph.D. degree from the University of Macau, Macao, China. His research interests include machine learning, data mining, and engineering applications of artificial intelligence.

Ka In Wong received the B.S. degree in electromechanical engineering from the University of Macau, Macau, China, in 2012. He is currently working towards the Ph.D. degree from the University of Macau, Macao, China. His research interests include automotive engineering, biofuels and engineering applications of artificial intelligence.

Chi Man Vong received the M.S. and Ph.D. degrees in software engineering from the University of Macau, Macao, China, in 2000 and 2005, respectively. He is currently an Associate Professor with the Department of Computer and Information Science, Faculty of Science and Technology, University of Macau. His research interests include machine learning methods and intelligent systems.