

Model Based Calibration Techniques for Medium Speed Engine Optimization: Investigations on Common Modeling Approaches for Modeling of Selected Steady State Engine Outputs

Christian Friedrich, Matthias Auer, and Gunnar Stiesch
MAN Diesel & Turbo

ABSTRACT

Due to the increasing number of engine setting parameters to be optimized, model based calibration techniques have been introduced to medium speed engine testing to keep the number of engine tests low. Polynomials in combination with d-optimal test plans have been proven to be a good choice for modeling the stationary behavior of selected engine outputs. Model approaches like artificial neural networks (ANNs) have been rarely used for medium speed purposes since they require quite high amounts of testing data for model training. To evaluate the potential of these model approaches radial basis function networks, a subclass of neural networks, as well as Gaussian processes have been investigated as alternatives to polynomials. A manageable amount of tests according to an adapted d-optimal test plan was carried out at a test bench. Based on the test results polynomials, radial basis function networks (RBFs) as well as Gaussian process models (GPMs) have been fitted for selected engine outputs. Furthermore, an extensive appraisal and comparison of these model approaches, using statistical criteria for regression evaluation, was executed.

The results show that radial basis function networks or Gaussian processes are promising alternatives to polynomials even if the amount of testing data is limited. The quality of the approximation results of the model types is on a comparable level. However, an advantage of radial basis function networks and Gaussian processes is the higher flexibility allowing a better representation of local nonlinear behavior of modeled engine outputs in comparison to polynomials.

CITATION: Friedrich, C., Auer, M., and Stiesch, G., "Model Based Calibration Techniques for Medium Speed Engine Optimization: Investigations on Common Modeling Approaches for Modeling of Selected Steady State Engine Outputs," *SAE Int. J. Engines* 9(4):2016, doi:10.4271/2016-01-2156.

INTRODUCTION

Stricter emission legislation with simultaneous demand for higher fuel efficiency led to sharply increased numbers of engine setting parameters due to the usage of new technologies in modern medium speed engines in the recent years. Modern technologies introduced to medium speed engines are for example common rail diesel injection (CR) allowing to apply multiple injection strategies to the engine, variable valve trains (VVT) and variable turbine geometry (VTG) improving the gas exchange at different load points and exhaust gas aftertreatment systems like diesel particle filters (DPF) or selective catalytic reduction (SCR) to reduce the engine out emissions.

Since all of these technologies have elements or actuators to be adjusted according the required engine power, fuel efficiency and exhaust emissions limits during engine calibration, the number of calibration parameters apparently increased [1, 2].

To manage this enormous calibration effort model based techniques utilizing design of experiments (DoE) for efficient test planning have been introduced to keep the number of tests at the engine test bench within limited scope. Aim of these techniques is to build a data driven

model describing the steady state behavior of selected engine outputs depending on chosen setting parameters with a limited number of required engine tests. By the help of these models predictions of engine outputs can be done at setting parameter combinations that have not been measured at the test bench. Furthermore, these models can be used for executing numerical optimizations regarding the calibration goals, e.g. minimizing fuel consumption in compliance with the emissions limits given by the emission legislation. Various publications like [3, 4, 5, 6, 7, 8, 9] have shown that the model-based calibration methodology was adopted and has proven very successful for various calibration tasks in the automotive industry. However, the amount of several hundred data points used to fit flexible models like RBF-networks or Gaussian process models exceeds the capacity of low and medium speed engines.

The successful introduction of model based calibration techniques to low and medium speed engines can be seen in [10, 11, 12]. Low and medium speed engines differ from automotive engines in many aspects. The higher absolute fuel oil consumption and duration to reach steady engine operation is limiting the amount of measurable data points for model based calibration techniques. Until now, almost only polynomial model approaches have been used in context

of low and medium speed engines for model based calibration in the past. Main purpose of low and medium speed engines running at engine speeds from 100 to 500 respectively 500 to 1000 rpm are propulsion or generator applications in ships or power plants.

The aim of the study presented in this paper was to investigate the alternative modeling approaches RBF-networks and Gaussian process models commonly used for automotive applications regarding their applicability to medium speed engines. A similar study comparing different modeling approaches for automotive engines is given in [9]. In difference to [9] global models have been investigated within this study since engine load was chosen as a model input parameter and the measurements were taken at different operation points. The modeling was done using the MATLAB® Model Based Calibration Toolbox. Gaussian process models are recently available in the MATLAB® Model Based Calibration Toolbox and thus available to a broad range of users.

MODELING TECHNIQUES

In this section the fundamentals of the three model types polynomials, radial basis function networks and Gaussian process models, which will be compared within this study, will be described. Furthermore, statistical criteria for model evaluation will be explained.

Polynomials

Polynomials are one of the most commonly used model approaches in the context of model based engine calibration [7, 8]. According to [13] a general polynomial with degree l and dimension p for an objective $\hat{y}$ can be described by:

$$\begin{aligned}\hat{y} = & w_0 + \dots \\ & + \sum_{i=1}^p w_i u_i + \dots \\ & + \sum_{i_1=1}^p \sum_{i_2=1}^p w_{i_1, i_2} u_{i_1} u_{i_2} + \dots \\ & + \sum_{i_1=1}^p \dots \sum_{i_l=1}^p w_{i_1, \dots, i_l} u_{i_1} \dots u_{i_l}\end{aligned}\quad (1)$$

In this equation u_i with $i = \{1 \dots p\}$ are the model inputs, $w_p, \dots, w_{i_1, \dots, i_l}$ the so called model coefficients and w_0 is the offset or threshold. The offset and the coefficients have to be determined using measured data points. Considering M basis functions ϕ_i describing any combinations of the p model inputs $\mathbf{u}$ using the coefficients θ_i , equation (1) can also be simplified to:

$$\hat{y}(\mathbf{u}) = \theta_0 + \sum_{i=1}^M \theta_i \phi_i \quad (2)$$

To fit a polynomial the estimation of the coefficients θ_0 and θ_i with n measured data points has to be performed. n data points lead to an equation system with n equations that is linear in its coefficients.

$$\begin{bmatrix} y(1) \\ y(2) \\ \vdots \\ y(n) \end{bmatrix} = \begin{bmatrix} 1 & \phi_1(1) & \phi_2(1) & \dots & \phi_M(1) \\ 1 & \phi_1(2) & \phi_2(2) & \dots & \phi_M(2) \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & \phi_1(n) & \phi_2(n) & \dots & \phi_M(n) \end{bmatrix} \cdot \begin{bmatrix} \theta_0 \\ \theta_1 \\ \vdots \\ \theta_M \end{bmatrix} + \begin{bmatrix} e(1) \\ e(2) \\ \vdots \\ e(n) \end{bmatrix} \quad (3)$$

Equation (3) expressed with the vectors $\mathbf{y}$, $\boldsymbol{\theta}$, $\mathbf{e}$ and the so called design matrix $\boldsymbol{\phi}$ gives:

$$\mathbf{y} = \hat{\mathbf{y}}(\mathbf{u}) + \mathbf{e} = \boldsymbol{\phi}\boldsymbol{\theta} + \mathbf{e} \quad (4)$$

By the help of the Least Squares Regression the coefficients vector $\boldsymbol{\theta}$ is determined, minimizing the sum of the squared errors $e(i)$ between model prediction $\hat{y}(i)$ and measured data point $y(i)$ according to:

$$\sum_{i=1}^n e^2(i) = \sum_{i=1}^n [y(i) - \hat{y}(i)]^2 = (\mathbf{y} - \boldsymbol{\phi}\boldsymbol{\theta})^2 \rightarrow \min \quad (5)$$

Using polynomials it is very common to apply a stepwise regression in order to eliminate terms of the polynomial equation that can be identified as not statistical significant. The influence of a model term $\theta_i \phi_i$ of equation (2) can be evaluated by the quotient t_{θ_i} that is calculated according the following formula.

$$t_{\theta_i} = \frac{\bar{\theta}_i}{\sigma_{\theta_i}} \quad (6)$$

The bigger this quotient the higher the influence of the model term will be. $\bar{\theta}_i$ is the mean value of the coefficient θ_i in n linear equations (3) and σ_{θ_i} is the standard deviation. A model term can be rated as statistical significant, if the quotient t_{θ_i} is bigger than the quantile $t_{\alpha/2, n-(M-1)}$ of the Student's t-distribution with a significance level α (commonly $\alpha = 0.025$ for a 95% confidence interval):

$$t_{\theta_i} > t_{\alpha/2, n-(M-1)} \quad (7)$$

Radial Basis Function Networks (RBFs)

As already mentioned RBFs are a subclass of artificial neural networks (ANNs). Therefore the basic structure of an artificial neural network consisting of an input layer with p inputs, one hidden layer with M neurons representing M transfer functions Φ and an output layer is given in Figure 1.

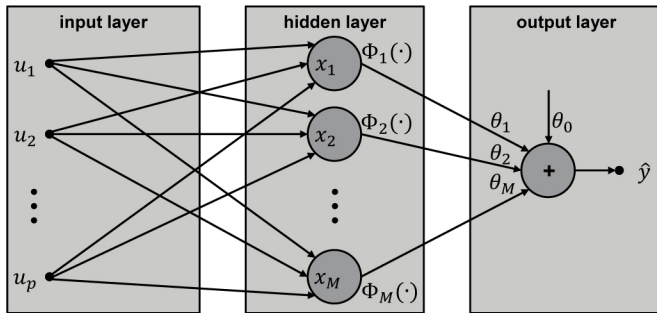


Figure 1. Structure of an artificial neural network with one hidden layer, p inputs and M neurons with the transfer function Φ according to [13].

As shown in Figure 1 the inputs $u_1, u_2, \dots, u_p$ are used to determine so called activations $x_1, x_2, \dots, x_M$ for each neuron in the hidden layer. If the activation x_i exceeds a defined threshold the neuron i will be activated and the neuron output will be given by applying the neurons transfer function Φ_i to the activation x_i . Finally the model output $\hat{y}$ arises by a weighted summation of the M neuron outputs $\Phi_1(x_1), \Phi_2(x_2), \dots, \Phi_M(x_M)$ using the coefficients $\theta_1, \theta_2, \dots, \theta_M$ and the offset or threshold θ_0 in the output layer [13]. The output $\hat{y}$ of the ANN can also be expressed with the simplified polynomial in equation (2).

Limiting potential transfer functions to radial basis function with the two parameters center c and width σ and determining the neuron activations $x_1, x_2, \dots, x_M$ with a radial construction mechanism the subclass of RBFs is given. Within the scope of this paper a RBF is considered to consist of only one hidden layer. According to [13] the activation x_i of the neuron i can be calculated by the Euclidean distance of the input vector $\mathbf{u}$ and the RBF center $\mathbf{c}_i$ as:

$$x_i = \|\mathbf{u} - \mathbf{c}_i\| \quad (8)$$

Commonly used radial basis functions are listed in Table 1. Additional basis function can be found in [14]. Talking about RBF-networks within this paper, it is assumed, that a single network only varies in the number of neurons in the hidden layer but all neurons represent the same radial basis function type.

Table 1. Commonly used radial basis functions and their mathematical descriptions [14].

Name of the function	Mathematical description
Linear function	$\Phi_i(x_i) = \frac{x_i}{\sigma_i} \quad (9)$
Inverse multiquadratic function	$\Phi_i(x_i) = \frac{1}{\sqrt{x_i^2 - \sigma_i^2}} \quad (10)$
Logistic function	$\Phi_i(x_i) = \frac{1}{1 + e^{\left(\frac{x_i}{\sigma_i}\right)}} \quad (11)$
Gaussian function	$\Phi_i(x_i) = e^{-\left(\frac{x_i^2}{\sigma_i^2}\right)} \quad (12)$

Contrary to the estimation of the coefficients θ_i of polynomials with the Least Squares approach, the Regularized Least Squares method is utilized in context of RBF-network training [15]. Therefore the already known sum of squared errors (equation (5)) is augmented by a term $\lambda_{reg} \theta^2$ to the regularized sum of squared errors which will be minimized.

$$(\mathbf{y} - \boldsymbol{\phi}\boldsymbol{\theta})^2 + \lambda_{reg}\boldsymbol{\theta}^2 \rightarrow \min_{\boldsymbol{\theta}} \quad (13)$$

The term $\lambda_{reg} \theta^2$ with the regularization parameter λ_{reg} is smoothing the model course within the design space by punishing neuron outputs with high weights θ_i . In order to fit a radial basis function network to a given dataset the amount of neurons or RBFs M has to be determined. In addition the function parameters $\mathbf{c}_i$ and σ_i of each neuron as well as their weights θ_i and the offset or threshold θ_0 have to be determined. There are several numerical algorithms available to determine the mentioned parameters. Examples of such algorithms are the Orthogonal Least Squares (OLS) [16] and the Regularized Orthogonal Least Squares (ROLS) [15] algorithm. Details of these algorithms and further information on parameter estimation of RBF-networks can be found in [15, 16, 17, 18, 19, 20].

Gaussian Process Models (GPMs)

Another way of statistical learning is given by using Gaussian processes. Therefore a specific family of functions out of various family functions is determined utilizing the Bayesian approach where the mean of the specific family of functions is representing the measured dataset with the highest probability [21].

Recalling equations (4) and (5) a model that is linear in its coefficients $\boldsymbol{\theta}$ can be written as:

$$\mathbf{y} = \hat{\mathbf{y}}(\mathbf{u}) + \mathbf{e} = \boldsymbol{\phi}\boldsymbol{\theta} + \mathbf{e}$$

and the coefficients can be estimated with n observations by minimizing the sum of squares.

$$\sum_{i=1}^n e^2(i) = \sum_{i=1}^n [y(i) - \hat{y}(i)]^2 = (\mathbf{y} - \boldsymbol{\phi}\boldsymbol{\theta})^2 \rightarrow \min_{\boldsymbol{\theta}}$$

Introducing a kernel function $k(\mathbf{u}_a, \mathbf{u}_b)$ and the Gram matrix $\mathbf{K}$ according to [9]:

$$k(\mathbf{u}_a, \mathbf{u}_b) = \boldsymbol{\phi}(\mathbf{u}_a)^T \boldsymbol{\phi}(\mathbf{u}_b) = (\mathbf{K})_{a,b} \quad (14)$$

$\hat{\mathbf{y}}(\mathbf{u})$ in equation (4) can be represented by [9]:

$$\hat{\mathbf{y}}(\mathbf{u}) = \mathbf{k}(\mathbf{u})^T \mathbf{K}^{-1} \mathbf{y} \quad (15)$$

Equation (15) is derived by some mathematical transformations of equation (4) that can be found in detail in [22]. $\mathbf{k}(\mathbf{u})$ is a vector consisting of $k_i = k(\mathbf{u}, \mathbf{u}_i)$ elements with $i = 1, 2, \dots, n$. A benefit of equation (15) is given by the complete expression of the solution of