

Development of Real-Time Capable Engine Plant Models for use in HIL Systems

Iakovos Papadimitriou, John Silvestri, Matt Warner and Benoit Despujols

Gamma Technologies Inc.

Copyright © 2008 SAE International

ABSTRACT

This paper presents a novel approach in real-time engine modeling. Unlike standard practices, which involve system level modeling, the presented methodology is a hybrid physical/system domain solution. Specifically, for each subsystem that the engine is divided into, a physical, map-based, or combination physical/map-based solution is chosen depending on the available computational power and the desired model detail. The resulting semi-physical engine models are suitable for real-time applications, such as Hardware-In-Loop (HiL) simulations, and, at the same time, re-usable to a large extent when model updates are required. In addition, since the proposed methodology allows for variable level of detail –from models as simple as pure map-based look-ups for torque, airflow, and exhaust temperature, all the way to models capable of predicting crank angle resolved cylinder pressure– it provides natural adjustability to the ongoing growth of computer power.

INTRODUCTION

The current trend for engine testing and verification as early as possible in the development process has uncovered the need for fast running engine simulations. Fast running engine models are needed not only by ECU (Electronic Control Unit) design groups for control strategy testing, but also by engine design groups for optimization of long drive cycles and vehicle integration. Currently, there is a barrier between these two groups mainly because of the lack of a unified approach in engine modeling. More precisely, during engine design, an engine model remains in the physical, component-level domain and fast execution is normally achieved through certain assumptions, such as steady air flow and map-based combustion. However, such assumptions are not sufficient to allow the model to run in real-time speeds at the currently available computer power. As a result, for control design a new system-level real-time capable model is developed. This new model is composed of a set of transfer functions, hence the physical representation of the engine is completely lost. Typically, the calibration process of such a model is a very time-consuming process and its accuracy is frequently questionable. In addition, a slight change in

the design of the engine requires the entire model re-calibration. Clearly, the challenge is to overcome the barrier between engine designer models and control designer models and create a methodology that favors model sharing and reusability.

Recently, several such attempts have been made [6,8,10] and the results are promising; however these attempts do not address all calibration issues that might occur and thus their accuracy, although acceptable, can be improved. Especially, with the development of new neural network architectures and training algorithms [1,2,3,7] and their application in computer programming [9], it is possible to combine them with well established physics laws [4,5] in order to provide fast-running semi-physical engine models.

In this paper is described a methodology that divides the engine into 4 major subsystems, as shown in Figure 1, and provides a modeling approach for each one of them. The resulting model is mostly based on the same equations that govern a typical detailed model and, at the same time, provides the ability for adjustable detail depending on the designer needs and the available computer power. The approach is applied on a diesel turbocharged engine and the real-time model is used in a simple HiL (Hardware-In-Loop) simulation.

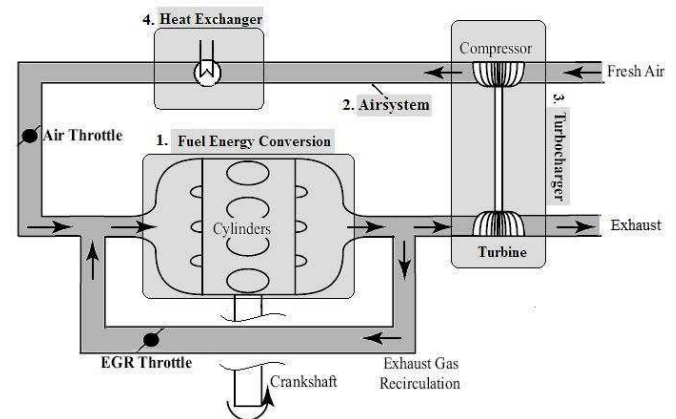


Figure 1. Schematic diagram of an engine.

FUEL ENERGY CONVERSION SUBSYSTEM MODELING

The system responsible for the conversion of fuel energy to useful kinetic and thermal energy is composed of the injectors, the cylinders and the pistons, the valves and possibly the ignition system – for SI engines only. The level of detail at which this system can be modeled varies depending on the desired results. The engineer may utilize three-dimensional fluid dynamics for the simulation of flow inside the combustion chamber, predictive combustion models, flexible body dynamics for the motion of the valves and the injectors; however real-time simulations require simpler models. Therefore, in this paper it is assumed that this entire system's dynamics are so fast so that they can accurately be captured based on a quasi-steady assumption. The system then becomes a black box that imposes air mass flow rate ($\dot{m}_a$) to the inlet and outlet of the cylinder block, mechanical torque (T_i) to the crank train and exhaust gas temperature (T_{exh}) to the exhaust port based on instantaneous values of certain quantities, such as engine speed, intake manifold pressure and temperature, exhaust manifold pressure, injected fuel mass and injection timing. In other words, the cylinder equations are:

$$\dot{m}_a = \frac{\rho_a \cdot V_d \cdot N}{2} \cdot \eta_{vol}(\mathbf{x}) \quad (1)$$

$$T_i = \frac{V_d}{4 \cdot \pi} \cdot imep(\mathbf{x}) \quad (2)$$

$$T_{exh} = T_{exh}(\mathbf{x}) \quad (3)$$

Where ρ_a is the inlet air density, V_d is the engine displacement, N is the cranktrain speed, η_{vol} is the volumetric efficiency, $imep$ is the indicated mean effective pressure and $\mathbf{x}$ is the vector of input quantities that affect cylinder performance.

One of the challenges of using this map-based cylinder model, well known as “Mean Value Cylinder Model”, is to provide the mapping of the input quantities $\mathbf{x}$ (e.g. engine speed etc) to the output quantities, namely volumetric efficiency η_{vol} , indicated mean effective pressure $imep$ and exhaust temperature T_{exh} . More precisely, in order to achieve accurate results, the input vector normally has more than 5 dimensions. In fact, depending on the complexity of the engine, the input vector may be ten-dimensional or even more. Therefore, look-up tables are not convenient since the number of entries required to accurately describe the system becomes forbiddingly high. Typical function approximations of the input variables, such as polynomials, which are frequently used in similar situations, are also problematic, since the fuel energy conversion subsystem is highly nonlinear. Therefore, in this paper, static artificial neural networks (ANN) are proposed in order to provide these three variables.

The proposed neural network architecture is a three-layer feedforward neural network with 2 hidden layers whose activation functions are tan-sigmoid and 1 output layer with linear activation function. The schematic of such a neural network with k inputs, n neurons in the first layer, m neurons in the second layer and 1 output (i.e. n - m -1 neural network) as well as the equation that describes it are shown below:

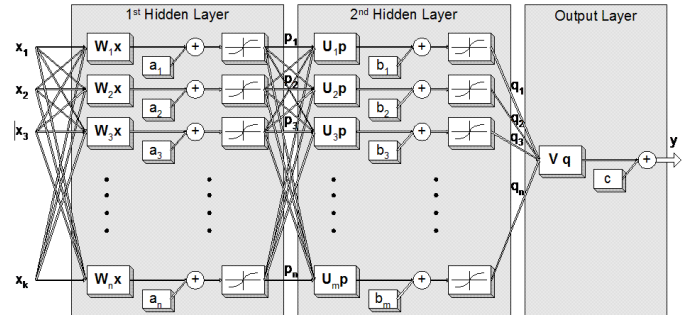


Figure 2. 3-layer feedforward neural network.

$$y = f(\mathbf{V} \times \mathbf{g}(\mathbf{U} \times \mathbf{h}(\mathbf{W} \times \mathbf{x} + \mathbf{a}) + \mathbf{b}) + \mathbf{c}) \quad (4)$$

Where:

$$h(z) = g(z) = -1 + \frac{2}{1 + e^{-2z}} \quad (5)$$

$$f(z) = z \quad (6)$$

$\mathbf{W}$ and $\mathbf{U}$ are the matrices of weights of the first and second hidden layer respectively, $\mathbf{a}$ and $\mathbf{b}$ are the vectors of biases of the first and second hidden layer respectively, $\mathbf{V}$ is the vector of weights of the output layer and c is the scalar bias of the output layer. $\mathbf{x}$ is the vector of inputs and y is the output

In order to train the neural network, i.e. to select “optimal” values for the weights and biases, the following objective function is used:

$$J = k_1 \cdot \sum_{i=1}^{nd} (y_i - \hat{y}_i)^2 + k_2 \cdot \left(\sum_{i=1}^k \sum_{j=1}^m W_{ij} + \sum_{i=1}^n \sum_{j=1}^m U_{ij} + \sum_{j=1}^n V_j + \sum_{j=1}^m a_j + \sum_{j=1}^n b_j + c \right) \quad (7)$$

Where k_1 and k_2 are carefully chosen factors, $\hat{y}$ is the output from the training data, and nd is the number of experiments used for training.

Clearly the first part of the equation describes the sum of squared error between each neural network output and the actual one and the second part describes the sum of squared weights and biases. The choice of the values for the penalties k_1 and k_2 reflects the balance between accuracy in the training data and accuracy in new data. More precisely, large values of the ratio k_1/k_2 produce large values of the weights and biases and thus the

resulting neural networks are very accurate in the training data, but have poor generalization capability i.e. produce poor accuracy when introduced to fresh inputs. On the other hand, small values of the ratio k_1/k_2 produce small weight and bias values and thus very smooth curves that are not capable of capturing hard nonlinearities. Therefore, the choice of k_1 and k_2 is very important. One of the methods for choosing these values is by using the Bayesian regularization framework. In this case k_1 and k_2 are considered random variables and they are updated iteratively until optimality is reached [2].

Finally, for the solution of the entire non-linear optimization problem the Levenberg-Marquardt algorithm is proposed. This algorithm, which is a combination of the Steepest Decent and the Gauss-Newton algorithms, provides stability and satisfactory convergence speed [3].

The 3-layer neural network architecture has been chosen because it has proven to be the most accurate and reliable compared to others that have been tried. Indeed, by looking at typical regression plots of IMEP for 4 different neural network architectures using 25 neurons each, namely Radial Basis Function (RBF), Self-Organizing Local Linear (SLL), Self-Organizing Local Quadratic (SLQ) and 3-Layer Feedforward (FF) neural network, it is clear that the feedforward neural network achieves much higher accuracy. Similar conclusions have been drawn for volumetric efficiency and exhaust gas temperature regression plots.

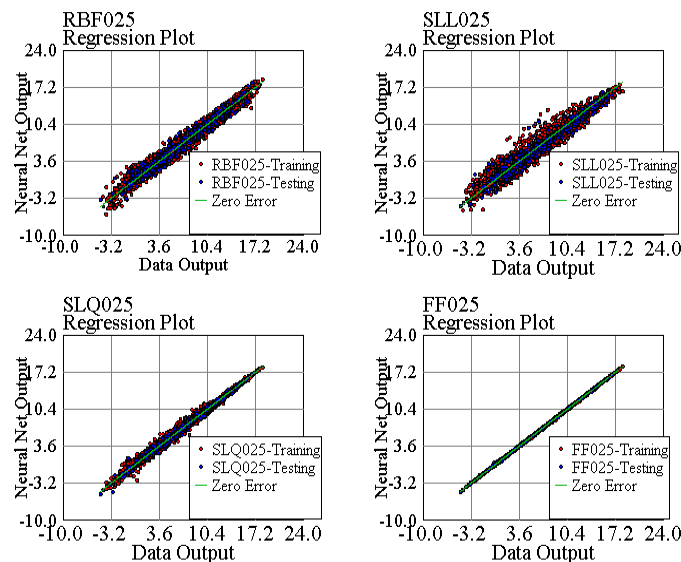


Figure 3. IMEP plots for various neural network architectures.

It has to be noted at this point that the described solution provides an option to predict crank angle resolved cylinder pressure; however this decreases execution speed and thus the modeler may choose not utilize this feature. This means that peak cylinder pressure per cycle is not captured by the model and therefore any quantity that depends on peak cylinder pressure has to be calculated using another way. Typical example is the

engine friction model which normally relies on mean piston speed and peak cylinder pressure. In this case, one more 3-layer feedforward neural network is suggested that will provide friction torque T_f as a function of the operating conditions of the engine, i.e.

$$T_f = \frac{V_d}{4 \cdot \pi} \cdot f_{mep}(\mathbf{x}) \quad (8)$$

Where $f_{mep}(\mathbf{x})$ is the friction mean effective pressure as a function of the same input vector as in equations (1)-(3).

As noted above, the model does possess the ability to predict crank angle resolved cylinder pressure, via a computationally efficient yet quite accurate cylinder combustion model implemented within the mean value based cylinder model. The cylinder pressure model runs at smaller, user-definable timesteps than the rest of the simulation, usually around one crank degree. Since the other models within the simulation are running at 5-10 degree timesteps, the cylinder pressure model substeps within the main model timestep, taking as many smaller steps as needed to synchronize with the main model. The combustion model, given a burn rate profile and accurate estimations of pressure and residual fraction at IVC (Intake Valve Closing), utilizes equilibrium of chemical species to determine the heat release rate from the charge mixture. Pressure and residual fraction estimation at IVC is achieved with the same neural network architecture mentioned before. The entire algorithm provides a computationally efficient and accurate in-cylinder pressure model which can be incorporated into a real time model for HiL applications, which is becoming a requirement to meet today's OBD (On-Board Diagnostics) requirements. Typical model accuracy is shown in the figure below, which shows the real-time cylinder pressure model captures peak cylinder pressure, location of peak cylinder pressure, and the rest of the cylinder pressure characteristics, quite accurately.

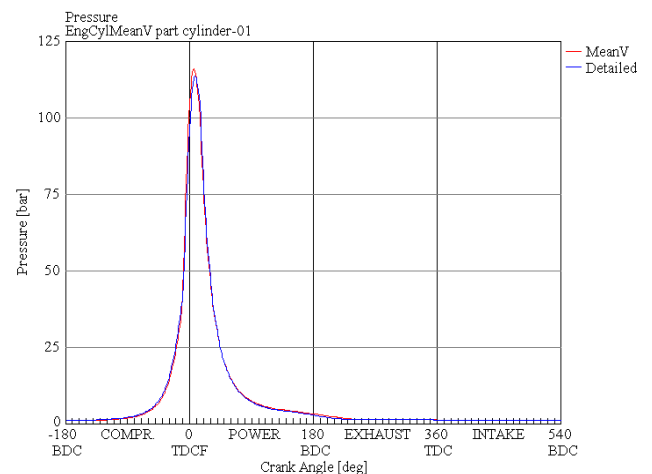


Figure 4. Detailed and Mean Value Cylinder In-cycle Pressure