

Automated Parameter Determination for IC Engine Simulation Models

Gerhard Pirker, Franz Chmela and Andreas Wimmer
Large Engines Competence Center

Copyright © 2009 SAE International

ABSTRACT

When developing and later using simulation models for combustion prediction in internal combustion engines, it is first of all necessary to determine the model constants. This paper describes the development of a method for the automated determination of model parameters which can be applied to any internal combustion simulation model. The work is not aimed at developing a new optimizing algorithm but at adjusting and adapting an existing optimizer to the special needs and convergence problems, which occur when applied to combustion models. Consequently, the paper describes the set-up of the objective function and several methods for improving the convergence. Finally, an outline for a strategy which uses the optimizing tool for model development is presented.

INTRODUCTION

The ever increasing demands regarding power output, fuel consumption and emissions when developing new engines call for sophisticated development methods. These new approaches are particularly required due to the increasing pressure on development time and cost. Therefore, pre-optimization of mixture formation, combustion and emission formation is inevitable already as early as at the design stage of the engine. As a consequence, better development results can be achieved more quickly and at lower cost. When applying zero-dimensional simulation models for the optimization through parameter variations, a number of model constants have to be determined. Hence the desired

predictions for a certain engine configuration or certain operating conditions can be made.

This contribution describes the development of an automated method, where the values of the relevant parameters of simulation models can be determined in order to match certain measured values as accurately as possible.

The focus in this paper is not on the development of a new optimizing routine but on the design of an easily usable tool for parameter optimization, based on an existing optimizing routine. A number of measures have been developed to ensure that the optimizing routine yields useful results.

OPTIMIZATION METHODOLOGY

The design of an optimization method starts with setting up an objective function. The task of the optimizer is then simply to minimize the value of this function. A number of different optimizing strategies for local [1] and global optimization including genetic algorithms [2] were investigated for their suitability. As genetic algorithms tend to consume rather lengthy computation times, a local optimization method was selected.

As in most cases the simulation models in question are not algebraic functions, but commonly consist of a number of sub-models given as sets of differential equations. The derivative of these functions can not, or not easily, be determined.

This fact considerably reduces the availability of suitable optimizing algorithms that can be found in the literature, because most of these algorithms use the derivative of the objective function for convergence improvement. Under these limiting conditions the most appropriate basis for the development of a suitable algorithm has proven to be Powell's method [3]. Among other things, this method is characterized by its potential to be adapted without major effort for the simultaneous optimization of a theoretically unlimited number of model parameters. This feature is of significant advantage when a large number of parameters have to be determined.

BASIC DEFINITION OF THE OBJECTIVE FUNCTION - Basically, the deviance F_1 , i.e. the sum of the quadratic differences between the simulated (dQ_{Sim}) and the measured burn rate curve (dQ_{Meas}) from start of combustion (SOC) until end of combustion (EOC) can be used as the objective function, as shown in Equation (1).

$$F_1 = \sum_{i=i_{SOC}}^{i_{EOC}} (dQ_{Sim,i} - dQ_{Meas,i})^2 \quad (1)$$

This function is used as a basis and gets improved by different modifications discussed later in this paper.

"MANUAL" MODIFICATION OF THE OBJECTIVE FUNCTION - A difficulty can occur during parameter optimization when the original burn rate curve contains structures with distinctly different radii of curvature. A simple example for such a case is shown in Figure 1, where the original burn rate curve is approximated by a combination of two Wiebe [4] burn rate curves.

In the upper part of this figure, the result of the model parameter optimization is displayed, the objective function being only the sum of quadratic differences. In this case the optimizer is not able to detect the finer structure of the premixed part and uses only one of the two Wiebe curves to approximate the measured curve.

In the lower part of Figure 1, the objective function is modified by applying an increased weighting during the phase of premixed combustion. This means that a deviation in this part will increase the value of the objective function to a larger extent than during the burn-out phase. This ensures that the optimizer will give priority to the shape of the premixed burn rate curve. The remaining deviations in the burn-out phase are due to the oversimplified structure of the Wiebe model.

As an example, the Equations (2) to (5) show the design of the objective function especially suitable for the problem. F_1 from Equation (1) is now modified by multiplying with a triangular function ($i_{EOC}-i$) decreasing towards the end of combustion (EOC) in order to increase the accuracy in the initial phase, see Equation (2). In Equation (3) this is repeated by means of function F_2 for the part shortly after start of combustion (SOC), but this time with tenfold weighting.

In order to accurately match the start of combustion, the weighted deviation in crank angle between the simulated ($\varphi_{SOC,Sim}$) and measured ($\varphi_{SOC,Meas}$) start of combustion is added in Equation (4) as function F_3 with an even higher weighting. The three parts are then combined in Equation (5).

$$F_1 = \sum_{i=i_{SOC}}^{i_{EOC}} (dQ_{Sim,i} - dQ_{Meas,i})^2 \cdot (i_{EOC} - i) \quad (2)$$

$$F_2 = 10 \cdot \sum_{i=i_{SOC}}^{i_{SOC}+5} (dQ_{Sim,i} - dQ_{Meas,i})^2 \quad (3)$$

$$F_3 = 100 \cdot (\varphi_{SOC,Sim} - \varphi_{SOC,Meas})^2 \quad (4)$$

$$F = \sqrt{F_1 + F_2 + F_3} \quad (5)$$

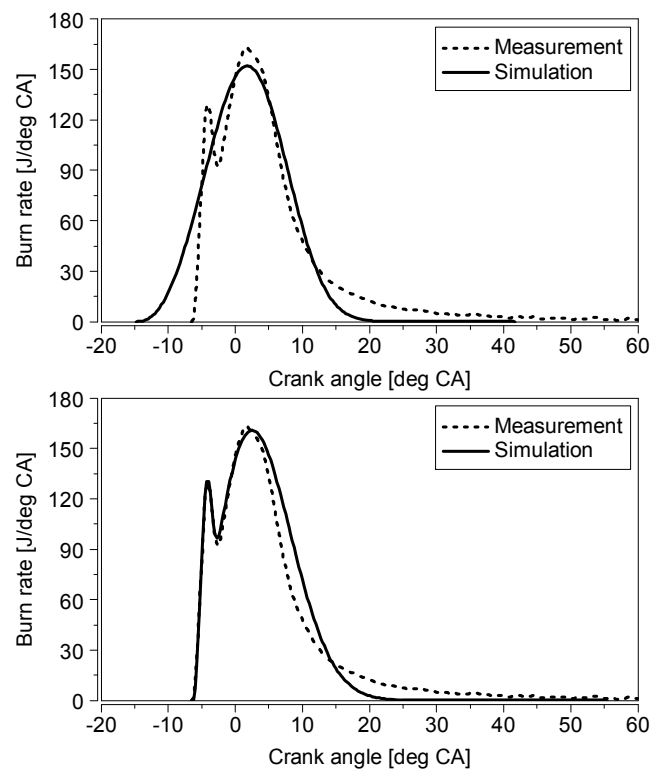


Figure 1: Optimized double Wiebe burn rate curve without (upper part) and with (lower part) weighted objective function

AUTOMATED WEIGHTING - With the "manual" weighting approach described in the previous section a suitable objective function can only be found by trial and error. An automated strategy for defining a suitable weighting function would be far more convenient. To this end, the control program should analyze the shape of the original curve and automatically return a higher weighting for the curve sections with stronger curvature. In Figure 2 a simple scheme for the assessment of the local curvature at the position i is shown for three consecutive points of the original curve. The local curvature K_i is calculated using Equation (6) as the

quotient of the distances from P_i to P_i' and from φ_{i-1} to φ_{i+1} .

$$K_i = \frac{|P_i' - P_i|}{\varphi_{i+1} - \varphi_{i-1}} \quad (6)$$

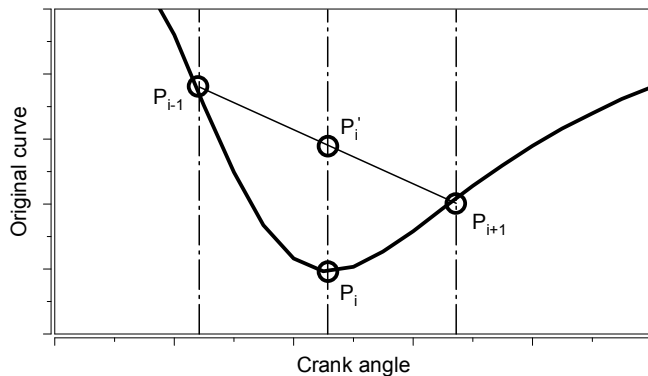


Figure 2: Scheme for assessment of the local curvature

The resulting curve shows sharp peaks at the locations with the highest curvature (see the dashed curve in Figure 3) similar to the actual evolute, i.e. the locus of all centers of curvature of the original curve.

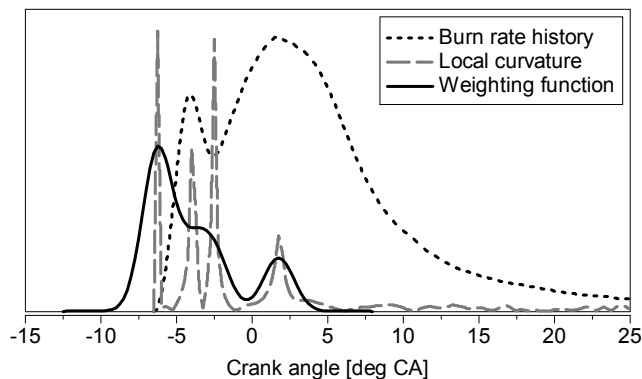


Figure 3: Original burn rate history, local curvature and weighting function

In order to get a practical weighting curve, Gaussian distribution functions with a user defined standard deviation are fitted through every peak. The summation of all distribution functions eventually yields the resulting smooth weighting function (see the solid line in Figure 3), which can be used for the weighting of the objective function. Together with this automated weighting method the optimizer will yield at least equally satisfactory results as with the rather arbitrary manual weighting method. The start of combustion should, however, be dealt with in a special way, as shown in Equation (4).

In Figure 4 the objective function according to Equation (1) is discussed for a Wiebe model. A Wiebe burn rate model contains in its simplest case four model parameters i.e. start of combustion, combustion duration, energy input and form factor. The form factor defines the character of the burn rate curve [4]. In the

following investigations the model parameters used for the optimization are form factor and combustion duration.

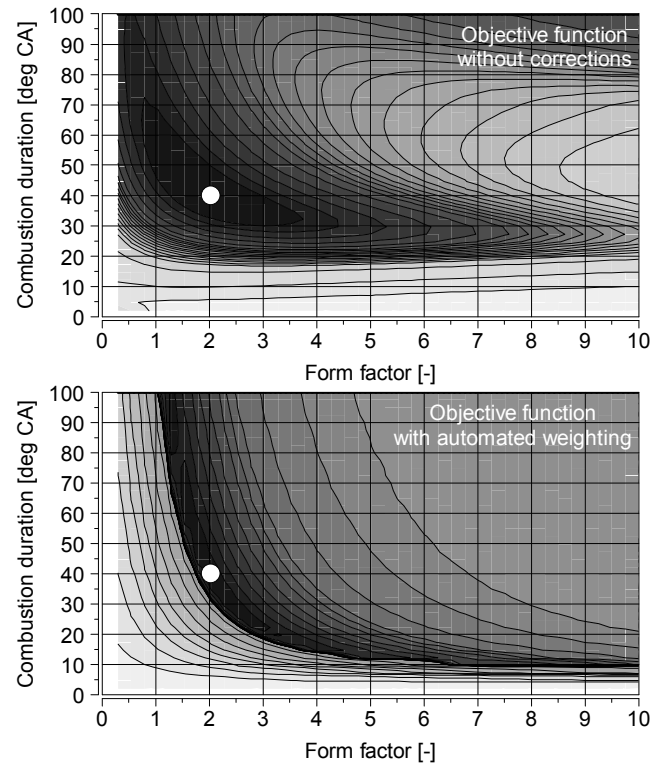


Figure 4: Effect of automated weighting on objective function for Wiebe model

In the upper part of Figure 4 the objective function shows a considerable degree of waviness with a distinct ridge between the minimum at (2, 40) and another region with low function values at the upper right corner towards (10, 100). In the lower part, the result of the objective function with automated weighting is shown. It appears that the waviness is reduced to a great extent, so that only one region with low function values exists. Therefore this region also contains the minimum. It is obvious that the optimizing routine will be able to detect this minimum without further difficulties.

BASIC OPTIMIZING ROUTINE - The input for the optimizer comprises a set of start values for the model parameters, a set of direction vectors for the initial variation of these parameters, and if required, physical limits for the parameter values.

The core process of the optimizer is the minimization of the objective function along one of the direction vectors, which corresponds to the parameter variation itself for a one-dimensional problem. In case of a multi-dimensional optimization, a sequence of one-dimensional optimizations has to be performed, each with consideration of the results of the previous minimization step. After completion of the optimization in each parameter direction considering the physical limits, a new average direction vector is construed in the parameter space. The new vector uses the connecting