# Model-based Design of Hybrid Electrical Vehicle Devices using automated Optimization Strategies

Johannes Friebe / Christine Schwarz / Thanh-Son Dao

## Abstract

Only the simulation of complete models allows the user to gain detailed insight into the often highly non-linear model behavior. The interactions between driver, combustion engine, electric motors, battery, powertrain, chassis, and road conditions are important factors in finding the optimal parameter set.

The model based design of a complete HEV requires extremely fast models when optimization strategies are applied, as they often require several hundred calls of the model to determine an optimal parameter set. This demand for high model speed causes many simulation tools to fail.

In this paper, we demonstrate the integration of an extremely fast high-fidelity series-HEV model with advanced optimization strategies. Besides determining an optimal parameter set, the analysis of sensitivities and the robustness while satisfying the model constraints are considered.

## Kurzfassung

Nur die Simulation von vollständigen Modellen erlaubt es dem Anwender detaillierte Erkenntnisse über das häufig hoch nichtlineare Modell-Verhalten zu gewinnen. Die Interaktionen von Fahrer, Verbrennungsmotor, E-Motor, Batterie, Antriebstrag, Fahrwerk und Straße sind wichtige Faktoren bei der Bestimmung des besten Parametersatzes.

Die Modell-basierte Planung eines kompletten Hybridelektrofahrzeugs (HEV) erfordert besonders schnelle Modelle, wenn Optimierungsstrategien zur Auslegung verwendet werden, da diese häufig mehrere Hundert Auswertungen des Modells benötigen, um eine optimale Parameterkombination zu finden. Durch die Anforderung an die hohe Modellgeschwindigkeit müssen viele Simulationstools bereits bei der Auswahl ausgeschlossen werden.

In diesem Vortrag wird ein Verfahren zur Kopplung von schnellen HEV Modellen mit effizienten Optimierungsalgorithmen vorgestellt. Dabei steht neben der Ermittlung einer geeigneten Parameterkombination auch die Untersuchung von Sensitivitäten und der Robustheit gegenüber der Erfüllung von Nebenbedingungen im Vordergrund.

# 1. Virtual Design of Hybrid Electrical Vehicle Devices

In the past, automotive engineers could develop cars step by step by constantly improving their knowledge. Today, vehicle design process usually requires an engineer to consider hundreds or even thousands of parameters to achieve an optimal design and ensure vehicle reliability and passenger comfort. It is also unavoidable to consider the complete system in the very beginning since many parameters are related to one another.

Only virtual systems allow comparing many different variants in a timely manner and with acceptable costs. However completing a high-fidelity vehicle model quickly reaches the performance limitation of most simulation tools.

To be successful, fast models and an easy to access optimization solutions is a precondition. An optimization process often requires a large number of experiments to determine the most relevant parameters. In this paper, we demonstrate the steps to generate the fastest virtual models and how to apply different simulation strategies. We also show how to use meta-models based on experimental data in case that a physical model is not available or FEA models are too slow for optimization.

In this paper, an HEV model based on the series hybrid configuration will be presented. Some of the key components in the model include:

- Combustion engine
- Electric motors/generators
- Electrochemical battery pack that includes temperature effects
- Driver model
- 3D chassis
- Powertrain
- Cooling system

Comparing this model with older simulation models confirms that the computational complexity increases dramatically as soon as complete HEV models are required.

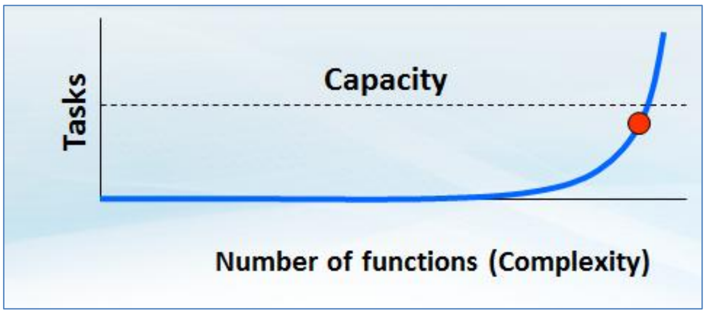



**Fig. 1** Emerging challenges demand for functions in simulation tools

## 1.1 MapleSim Physical Modeling Tool

One of the most promising approaches is to use the most efficient methods for symbolic simplification. This approach is supported by MapleSim based on the Maple symbolic engine. The sophisticated simplification and model reduction techniques in MapleSim allow very complex yet fast model of a complete HEV to be built.

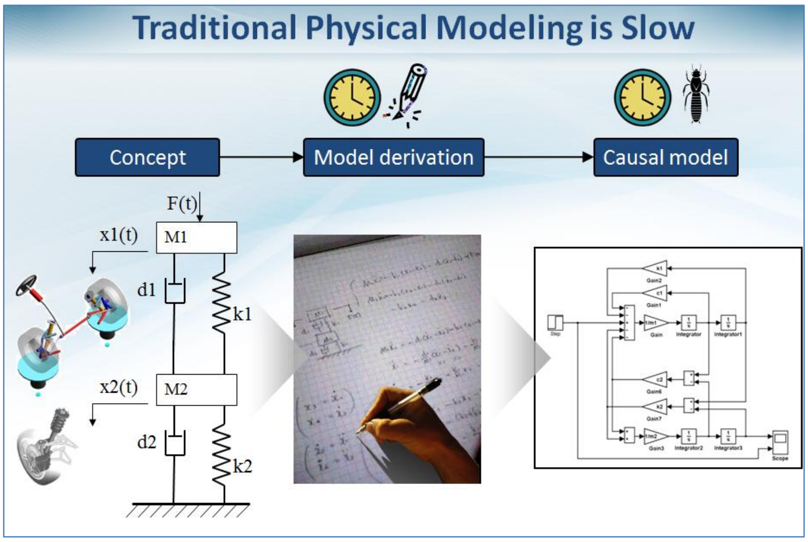



**Fig. 2** Traditional physical modeling requires time consuming and error prone manual creation of equations.

Another advantage for many engineers and researchers is the ability to create highly complex physical models and then generate their symbolic equations.

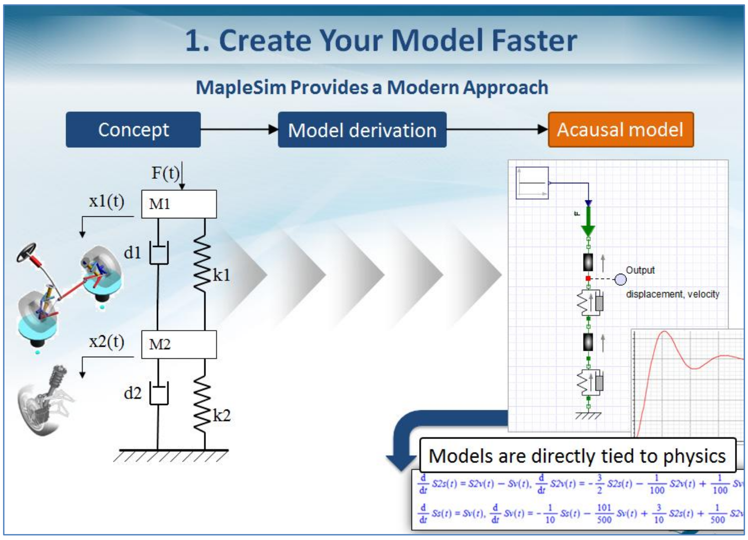



**Fig. 3** Automatic generation of the euqations saves time.

## 1.2 Setup and HEV Model

The modeling of a series-HEV in MapleSim was straightforward since all the components were pre-created and available in MapleSim. The modeling process involves mainly connecting the existing components together and tunes the parameters to get preliminary results.

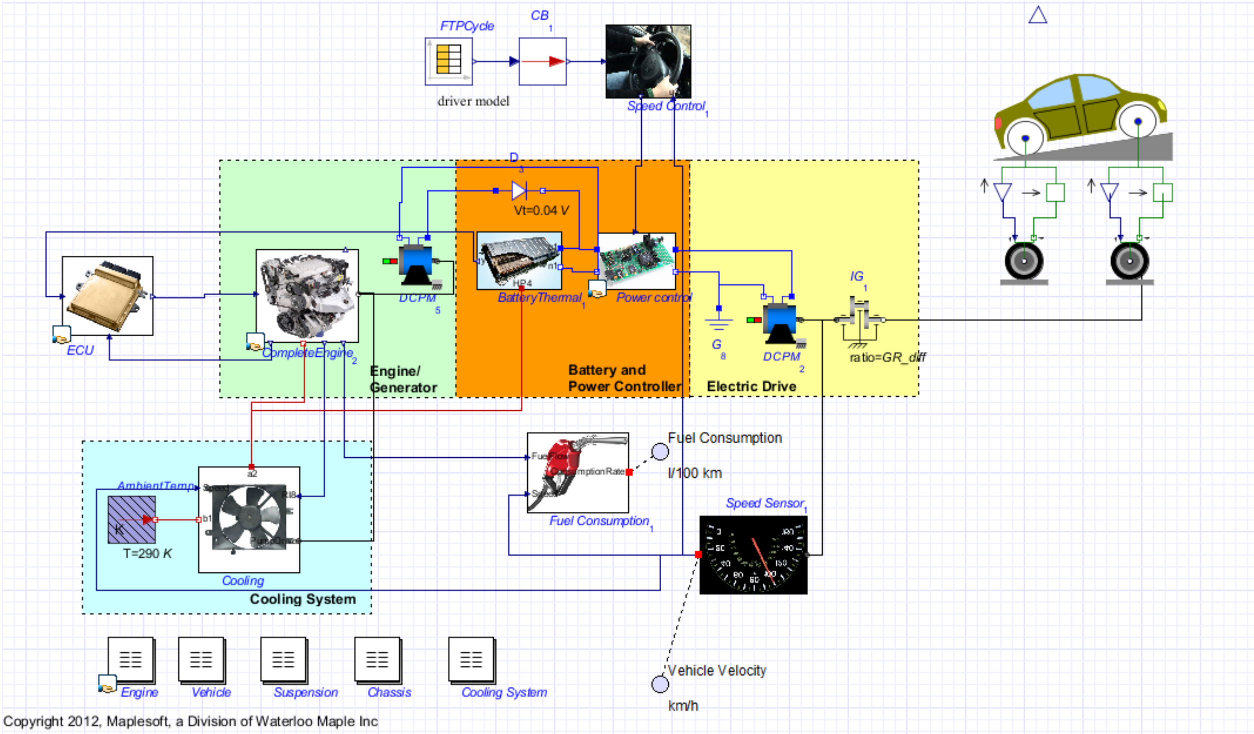



**Fig. 4** Overview HEV models as shown in MapleSim

Unlike other causal modeling software that use complex signal flow block diagrams as shown in the comparison in Figure 2, MapleSim has an intuitive graphical user interface (GUI) that allows engineers to quickly create models with different configurations. The C code generated from a MapleSim model is efficiently optimized using symbolic techniques. This allows the code to have the fast execution time, making it suitable for real-time applications and optimization solutions. In the following sections we will show the details for some key components of the series-HEV model.

## Combustion engine

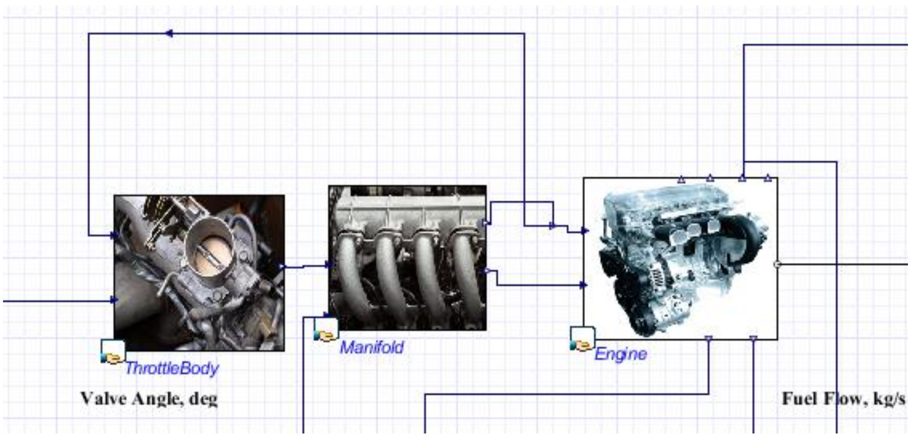



**Fig. 5** Mean-value model for a combustion engine

In this work, the mean-value engine model developed by Saeedi [9] has been used since this model is computationally efficient while capturing enough information about the physics of the engine system.

The mathematic equations describing the engine model have been given by Moskwa [6] and Heywood [4]. The engine model is composed of three main subsystems: the throttle, intake manifold, and engine power generation from the fuel combustion. The rate of air mass flowing into the engine is determined based on the geometry and position of the throttle valve set by a simple PID controller which closes the loop between the actual and desired engine speeds. The intake manifold has a significant effect on the gas flow and pressure to the engine cylinders. The pressure of the air/fuel mixture in the intake manifold can be calculated based on the ideal gas equation:

$$\dot{P}_m = \frac{RT_m}{V_m}(\dot{m}_{thr} - \dot{m}_e) \tag{1}$$

where $R$ is the gas constant, $T_m$ and $V_m$ are the temperature and volume of the intake manifold, $\dot{m}_{thr}$ is the throttle mass flow rate, and $\dot{m}_e$ is the throttle mass outflow.

The calculated air/fuel pressure and mass flow rate in the manifold are used to compute the power generated from the engine through the combustion of the fuel in the gas mixture delivered to the cylinders, accounting for thermal efficiency, friction, and inertial losses in the engine and the inertial load at the drive shaft. The engine power is calculated based on the engine equations proposed by Hendricks et al. [3]:

$$P_{net} = P_{ind} - P_{loss} - P_{load} \tag{2}$$

where $P_{ind}$, $P_{loss}$, and $P_{load}$ are the indicated power, lost power, and load power, respectively.

The engine speed can be obtained from crank shaft speed equation as:

$$\dot{n} = \frac{1}{J_e n} P_{net} \tag{3}$$

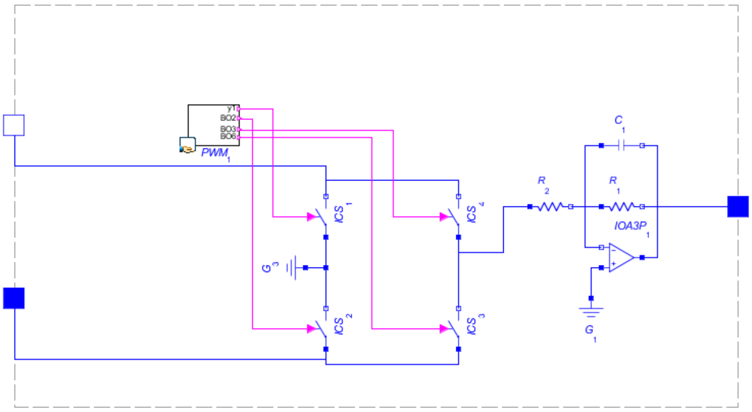
where $J_e$ is engine inertia and n is the engine rotational speed.

A complete description of the model can be found from the URL [5].

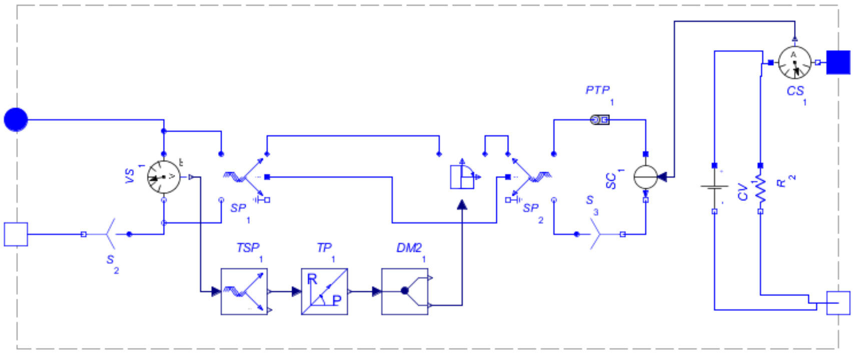
### Power electronics

In order to control the amount of power going from the battery to the motor during driving, or vice-versa during regenerative braking, a power controller is necessary. If the battery's voltage needs to be stepped-up to operate the motor, a boost converter is used. If the voltage needs to be stepped-down, then a buck converter is used [2]. Both types of converters use non-linear electrical circuits with variable-duty high frequency switching transistors to convert one DC voltage level to another.

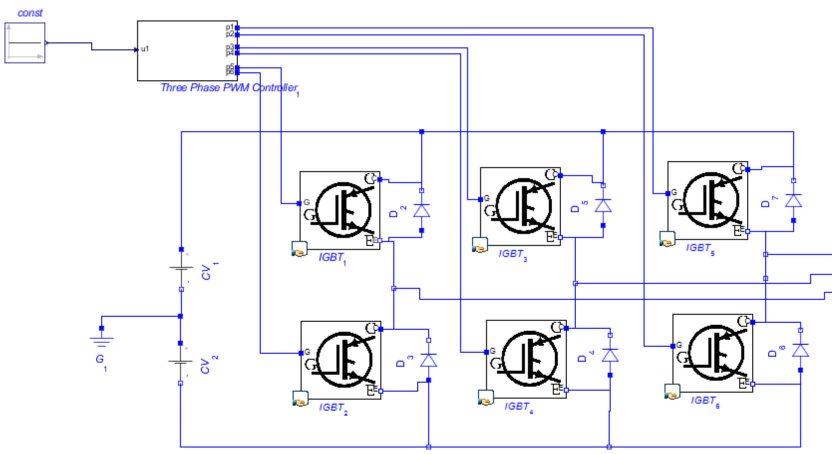
Developing power electronic components in MapleSim is easy since most of the frequently used elements are readily available in MapleSim's component libraries. This allows for both switched and average networks to be easily modeled as shown in Figure 6. This gives modeling engineers the freedom to create models with different level of complexity and fidelity to suit their needs.



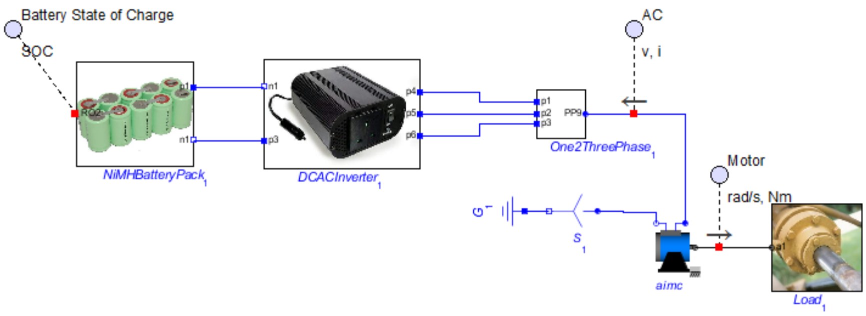

(a)




(b)




(c)



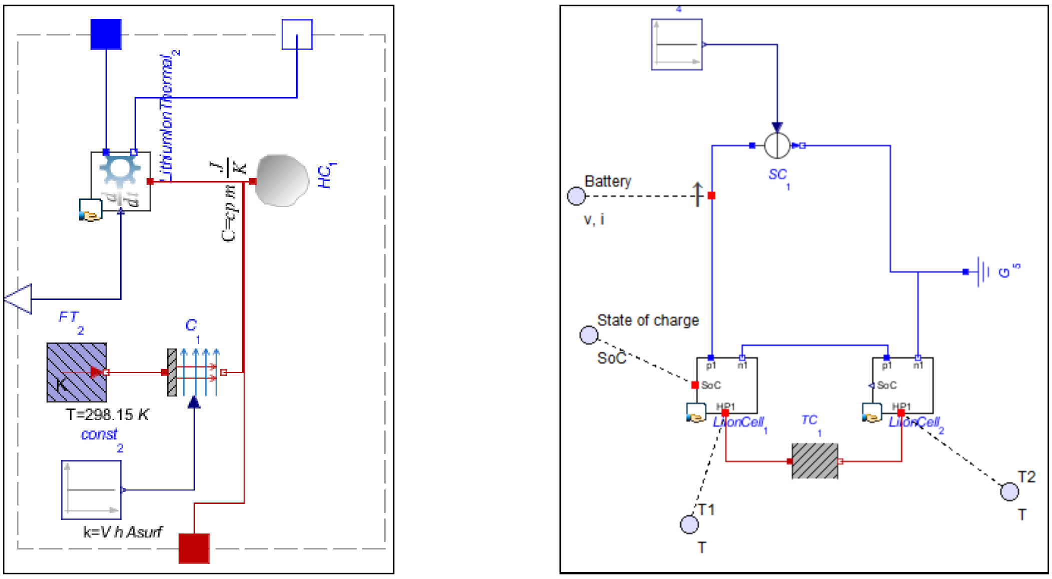

(d)

**Fig. 6** MapleSim power electronic models: (a) Switched network (b) Average inverter (c) IGBT Inverter (d) Dynamo with power electronic components

### Electrochemical Battery Model With Thermal Effects

Typically, the manufacturer will provide a variety of charge/discharge curves and parameters that give information on the dynamic behavior of a battery. The challenge for engineers is to model the dynamic behavior of a battery and fit the manufacturer data to the chosen model. Many of the available models are circuit-based and rely on dynamic components like resistors and capacitors whose values change in response to the operating conditions. However, purely circuit-based implementations do not implement the thermal characteristics of the battery, and very few circuit simulators allow for dynamic components with such complicated governing equations. Consequently, existing models are inadequate predictors of battery behavior, and are therefore ill-suited for use in engine models where heat loss must be taken into account or in studies of overall energy efficiency.

The multiphysics and parametric nature of MapleSim makes it especially well-suited to implement a realistic battery model. Sources are also used to facilitate charging and discharging of the battery model. Since electrochemical battery models are developed based on the fundamentals of physics and chemistry, phenomina such as temperature effects, side reactions, or battery aging can be incorporated easily. In MapleSim, battery thermal effects can be modeled using thermal components as shown in Figure 7. This model also accounts for the temperature exchange with the air through a convection block. The model can also be parameterized easily by giving symbolic names for the parameters. These parameters will remain symbolic in the model equations until a numerical solver is called to solve the equations.




(a) (b)

**Fig. 7** Battery thermal modeling: (a) Thermal within a cell, (b) Temperature exchange between two cells

### Chassis dynamics

The vehicle is represented as a simple 2D vehicle model with longitudinal dynamics. This model is sufficient to predict the handling and braking behaviors of an automobile without the effort of modeling all of the small details such as suspensions, bushings, linkage properties, etc.

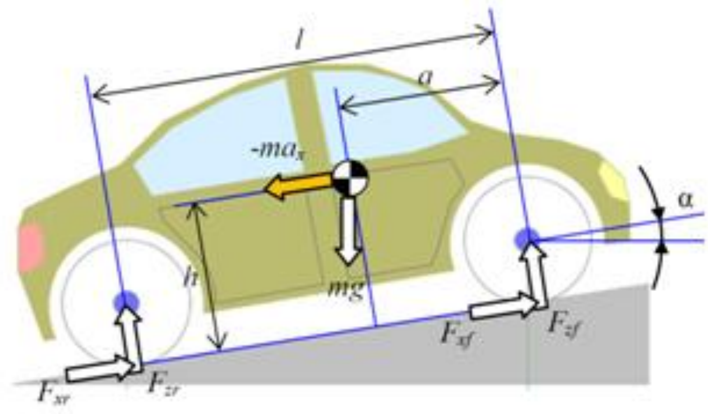



**Fig. 8** Vehicle chassis

The vehicle equations are given by:

$$s_{xr} = s - l - a \tag{4}$$

$$s_{xf} = s + a \tag{5}$$

$$v_x = \frac{ds}{dt} \tag{6}$$

$$a_x = \frac{dv}{dt} \tag{7}$$

where *s* is the longitudinal DOF of the vehicle, *l* is the vehicle wheel base, and *a* is the distance from the front axle to the projection point of the vehicle CG onto the line connecting the front and rear axles.

The equation for the longitudinal motion is:

$$m\,a_x = F_x - F_d - m\,g\sin(\alpha) - F_l \tag{8}$$

where $F_l$ is the external force on the vehicle and $F_x$ is the traction force and is defined by $F_x = f_{xr} + f_{xf}$.

The air drag $F_d$ is calculated as:

$$F_d = \frac{1}{2} C_d \rho A\,|v_x - v_w|(v_x - v_w) \tag{9}$$

The weight distribution for front and rear axles is calculated using the following equations:

$$l\,f_{zf} = -h(m\,a_x + F_d + m\,g\sin(\alpha) - F_l) + m\,g\cos(\alpha)(l - a) \tag{10}$$

$$l\,f_{zf} = \ -h(m\,a_x + F_d + m\,g\sin(\alpha) + m\,g\cos(\alpha)a \tag{11}$$

The vehicle model uses the Magic Formula developed by Pacejka et al. [1][8] for the longitudinal tire forces and relaxation lengths to model the tire transients.

### Powertrain

In this model we use a simple reduction gear, but this could easily be replaced by several more complex gears from the power train library.

### Cooling system to see thermal effects

The cooling system consist of and airflow a radiator and a circuit that connects the fan to the engine and the battery.

### 1.3 Selected output parameters and their contradicting objectives

| **Parameter** | **Objective** | **Range** |
|---|---|---|
| Fuel consumption | Minimize | 2-8 l/100 km |
| Power loss | Minimize | 100 – 200 |
| Battery temperature | Below 315 K | < 315 K |
| Battery current | Keep below a threshold that can harm the battery life | < X A (4.0e+007) |

## 2. Automation and Process Integration

Process automation is an important step in increasing the efficiency of a product development process. Especially performing recurring simulations manually is very time consuming and can be automated very efficiently. Therefore the software tool

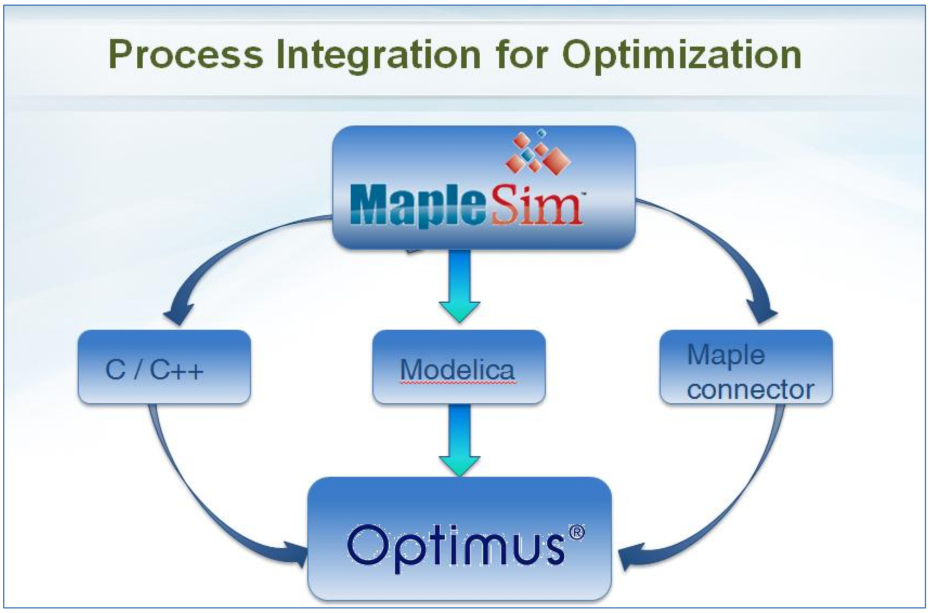



**Fig. 9** Three options to connect OPTIMUS to MapleSim. The most efficient connection is the direct connection. OPTIMUS will see the input and output parameters defined in the model.

OPTIMUS is introduced to automatically start the MapleSim calculations and analyze the results. How the process integration was realized will be discussed in the following sections.

**Integration in the Optimization Process**

OPTIMUS can exchange information with other simulation tool either with direct interfaces or using the open ASCII interface. MapleSim comes with an optional exporter to OPTIMUS. It is very simple to create an export file that can be used for the connection in OPTIMUS. The input variables shown on the left side in Figure 10 are substituted in the MapleSim input file that contains MapleSim commands to call the pre-compiled MapleSim procedure. In the output interface the MapleSim procedure is called and the produced responses on the right are extracted automatically from the result file.

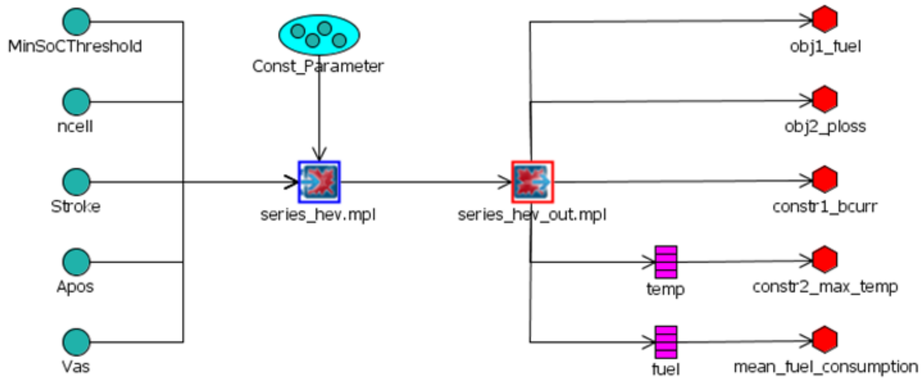



**Fig. 10** Visualization of HEV workflow in OPTIMUS with used input and output variables

## 3. Selection of Parameters

At any stage of the design process the challenge is to choose suitable parameters for the optimization task from an often large design space. One possibility to automatically support the parameter selection is to use knowledge from sensitivity analysis based on intelligent design tables. The aim of this process is to determine the most significant design variables with the greatest influence on the responses. The software OPTIMUS supplies the user with different statistical measures to compute the correlation values between the variables of the system. As a result the most significant factors can be determined and less important parameters can be excluded from the further process, which leads to a reduction of dimensionality of the design space and therefore also to the reduction of the problem complexity.

### 3.1 Design of Experiments

Intelligent design tables or Design of Experiments (DOE) are a statistical method to systematically plan experiments and analyze technical systems. The purpose is to maximize the gain of information while keeping the number of evaluations at a minimum. In the context of simulation processes different DOE methods are used for screening as there are often many design variables that do not have to be considered in a first step.

In many use cases a Latin-Hypercube sampling is chosen to achieve a complete and equally distributed covering of the design range for each variable. An additional advantage is that the number *N* of requested experiments can be chosen independently from the number of variables. The method consists of separating the definition range of each variable in *N* intervals and drawing a probe in each interval as shown for a 2D example in Figure 11.

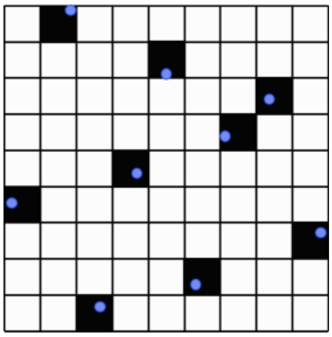


**Fig. 11** Latin-Hypercube Sampling

### 3.2 Sensitivity Analysis for Initial Parameters

The initial selection of parameters was determined by engineering experience and possible variations in the model. Starting with a set of 13 design variables a Latin-Hypercube sampling is computed with 500 experiments. The ranges for each parameter are chosen in physically feasible ranges while not being too restrictive on the limits beforehand.

| **Parameter** | **Description [unit]** | **Range** |
|---|---|---|
| MinSoCThreshold | Minimum State of Charge Threshold | 0.3 – 0.6 |
| ncell | Number of battery cells | 100 – 200 |
| Vmanifold | Engine manifold volume [$m^3$] | 0.003 – 0.005 |
| Bore | Engine bore [m] | 0.0855 – 0.1 |
| Stroke | Engine stroke [m] | 0.0814 – 0.19 |
| Apos | Area of battery positive electrode [$cm^2$] | 100 – 500 |
| Aneg | Area of battery negative electrode [$cm^2$] | 100 – 500 |
| apos | Specific surface area of battery positive electrode [$cm^2/cm^3$] | 3000 – 5000 |
| aneg | Specific surface area of battery negative electrode [$cm^2/cm^3$] | 2000 – 4000 |
| Va | Nominal voltage of electric motor [V] | 400 – 1000 |
| Ia | Nominal current of electric motor [V] | 50 – 100 |
| Vas | Nominal voltage of electric motor [V] | 50 – 150 |
| Ias | Nominal current of electric motor [V] | 50 – 150 |

To determine the main factors with the greatest impact on the responses linear correlation factors are considered as the main measurement factors. This value varies in a range of -1 to 1, while a correlation factor of 1 represents a direct linear influence on the considered response. Analyzing the correlation matrix reveals some design variables without any or only with minor influence on any of the responses. These parameters are consequently excluded from further computations and set to a constant value which was predefined as nominal value or can be chosen arbitrarily.

| Input / Output | MinSoCThreshold | ncell | Vmanifold | Bore | Stroke | Apos | Aneg | apos | aneg | Va | Ia | Vas | Ias |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| obj1_fuel | 0.48 | -0.04 | 0.01 | 0.02 | 0.03 | -0.51 | 0.01 | 0.04 | -0.07 | -0.01 | -0.00 | -0.33 | 0.06 |
| obj2_ploss | 0.16 | 0.34 | 0.06 | -0.05 | -0.03 | -0.08 | -0.02 | 0.02 | -0.04 | -0.01 | 0.06 | -0.57 | 0.05 |
| constr1_bcurr | -0.10 | -0.54 | 0.01 | 0.10 | 0.24 | 0.32 | -0.01 | 0.05 | -0.05 | -0.02 | 0.04 | -0.12 | -0.... |
| cpnstr2_max_temp | 0.13 | 0.41 | 0.05 | -0.05 | -0.05 | 0.02 | -0.03 | 0.02 | -0.03 | -0.01 | 0.04 | -0.64 | 0.03 |
| mean_FC | 0.53 | -0.15 | 0.00 | 0.04 | 0.03 | -0.63 | 0.02 | 0.03 | -0.06 | -0.02 | 0.01 | -0.25 | 0.06 |

**Fig. 12** Linear correlation coefficients with significant input parameters highlighted

Figure 12 shows the correlation matrix and highlights the parameters with significant correlation values with any of the chosen responses. Using this approach the dimensions of the design space can be reduced to 5. Therefore only 5 parameters are chosen to be considered during the optimization: *MinSoCThreshold, ncells, Stroke, Apos, Vas*.

# 4. Optimization Strategy

To achieve optimal model characteristics the objective functions have to be defined, i.e. the responses that have to be minimized or maximized. Additionally the possibility is given to respect certain output variables as constraints during the optimization process. The selection of a suitable optimization strategy and algorithm is influenced by many factors. Long simulation times of a single experiment, parallelization possibilities and available solver licenses, number of design variables and behavior of the system have great influence on which strategy should be chosen. In this case simulation time is not a limiting factor as a single analysis runs very fast.

## 4.1 Response Surface Modeling

Response Surface Models or mathematical models are computed based on simulation data from DOEs or testing data. To achieve a good global quality for the model the sampling points have to be spread uniformly in the design space. This is why a Latin-Hypercube sampling is used in many cases with an arbitrary number of evaluations.

The software OPTIMUS provides two different techniques to create meta-models. The user can either choose manually from a list of different methods or make use of the automated computation of the best fitting model on the data for each response. Dependent on the systems behavior and the purpose of the model a certain method can be chosen directly. In case of high nonlinearities a Kriging model is often a good choice to create a continuous representation of the functions behavior. Once a model is created it can be used for running algorithms on the analytic model, given a sufficient model quality.

The quality of the analytical models can be assessed with different available criteria that are computed automatically during model creation. One method to determine the quality of the response surface is based on cross validation. The procedure is to exclude subsets of the results from creating the meta-model and compare the results in the left out set with the predicted values on the model. The cumulated and normalized error R2Press can then be used as criteria for the model quality. In this case only an R2Press value of 0.8 – 0.9 can be achieved on the best model for the objective functions based on 1000 evaluations. In Figure 13 three cuts through the created models are displayed which shows the differences between the models on the same data set.

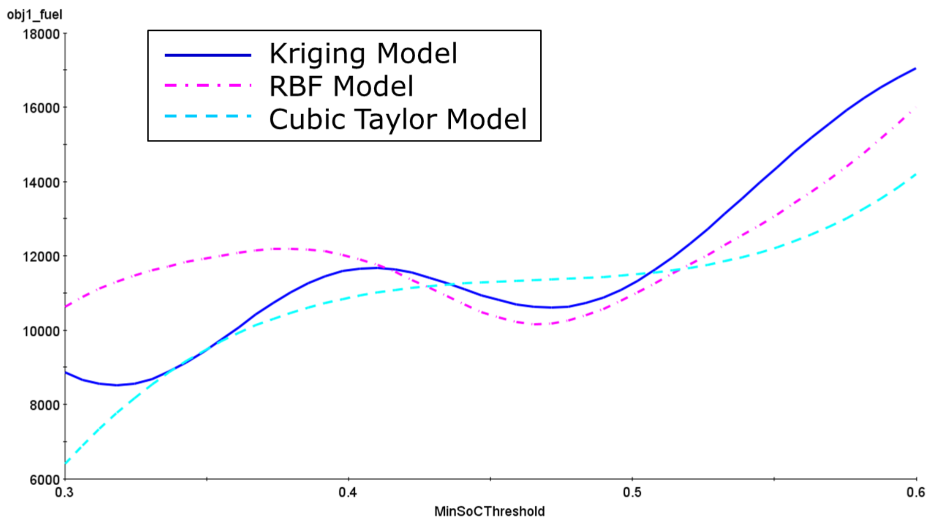



**Fig. 13** Comparing different response surface models

The biggest advantage of this approach is that no additional simulations have to be evaluated. Due to the not sufficient model quality (R2Press < 0.95) and fast simulation times the optimization for the discussed application example was not performed on the model. Instead the optimization algorithm was run directly on the simulation sequence calling MapleSim.

## 4.2 Multi-Objective Optimization

In this application example more than one objective function has to be fulfilled simultaneously. When considering two conflicting objectives one optimum for both functions can not be found. Instead the goal is to determine the complete set of compromise solutions, also known as Pareto front. To detect the Pareto points the evolutionary algorithm NSEA+ is used. Figure 14 displays the optimal points lying on the Pareto front.

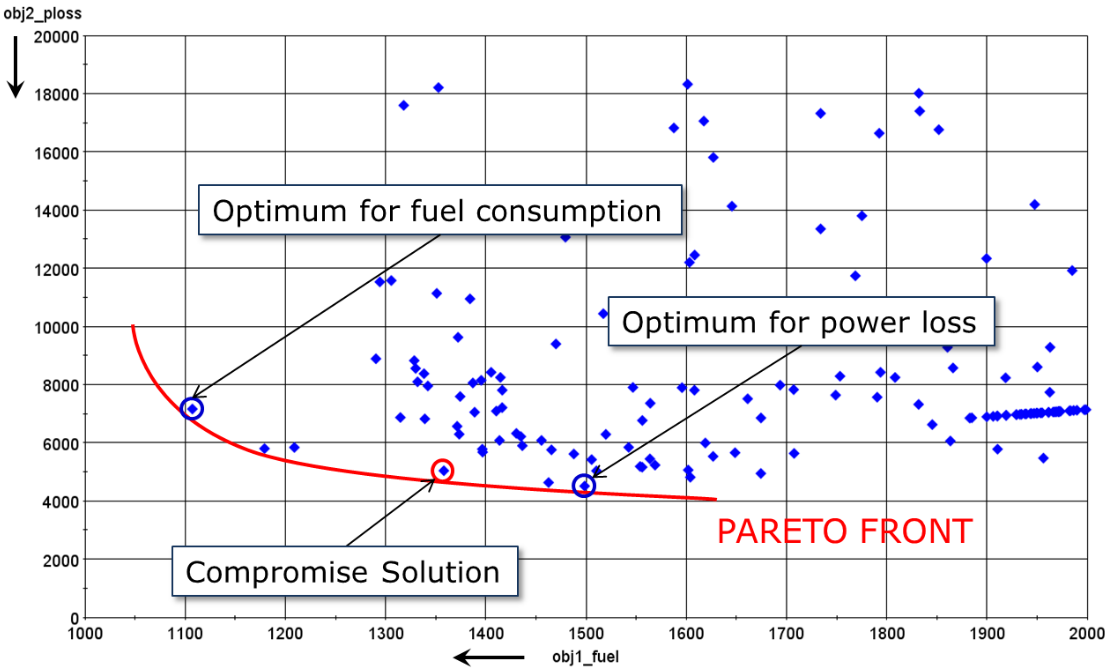



**Fig. 14** Visualization of the optimization results

### 4.3 Robustness of the found Optimum

To reach a decision, which of the found compromise solutions should be defined as solution for the HEV model, the sensitivity in each of the Pareto points is evaluated. This means taking into account small perturbations on the input side and measuring the Sigma values for the objectives on the output side.

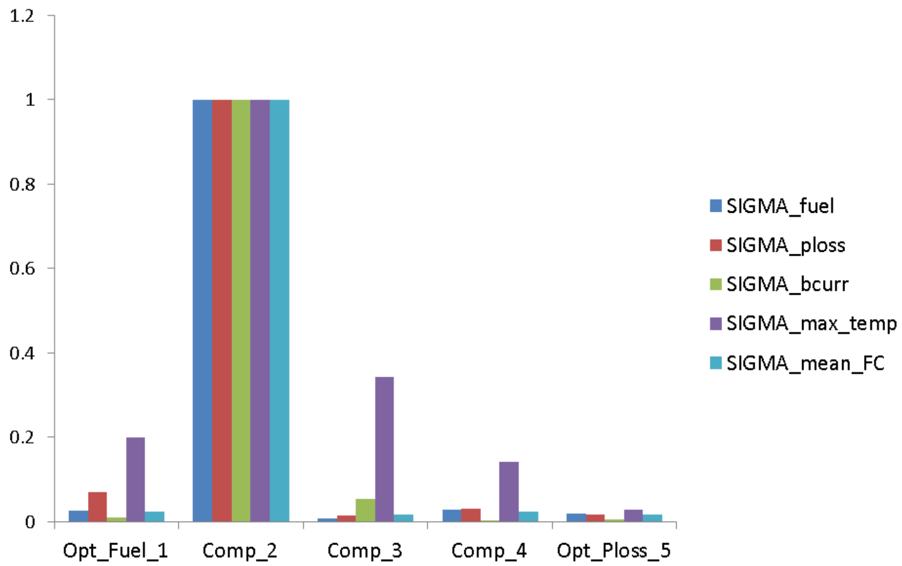



**Fig. 15** Comparison of Sigma values for 5 Pareto points

The goal is to identify one point with robust behavior when considering distributed input variables, which means determining the Pareto point with smallest Sigma values. In Figure 15 the normalized Sigma values for 5 Pareto points are shown. To achieve the best robustness and consequently the best predictive quality for the responses the optimal point for the power loss *Opt_Ploss_5* should be selected.

## 5. Conclusion and Results

In this paper a new and efficient approach for automating and optimizing the design parameters of a complete hybrid electric vehicle was shown. The goal was not only to handle a complex vehicle model in an easy and intuitive way, but also to detect coherences and perform advanced optimizations without much additional effort for the user.

The following table shows the results of the optimized design compared to the initial design. For the two objective functions fuel consumption and power loss a great improvement could be achieved. Also both constraints on battery current (< 4.0e+007) and the maximum temperature (< 315 K) could be fulfilled in the final design.

| | obj1 **fuel** | obj2 **ploss** | constr1 **bcurr** | constr2 **maxTemp** (K) | **mean_FC** (l/100km) |
|---|---|---|---|---|---|
| **Initial Design** | 12302.77 | 145218.92 | 12542666.38 | 316.37 | 2.10 |
| **Optimal Design** | 1498.60 | 4505.82 | 9990640.07 | 301.28 | 1.56 |
| **Absolute Improvement** | -10804.16 | -140713.10 | -2552026.31 | -15.09 | -0.54 |
| **Relative Improvement** | **-87.82%** | **-96.90%** | **-20.35%** | **-4.77%** | **-25.53%** |

The complete model setup including all features was realized using the software MapleSim. All described optimization algorithms and used techniques to create DOE tables, response surface models or robustness analyses are included in the software OPTIMUS [7].

# Bibliography


[1] Bakker, E., Nyborg, L., and Pacejka, H.B., Tyre modeling for use in vehicle dynamic studies, SAE technical paper 870421.

[2] Bauman, J., Advances in Fuel Cell Vehicle Design, PhD thesis, University of Waterloo, 2008.

[3] Hendricks, E., Chevalier, A., Jensen, M., Sorenson, S.C., Trumpy, D., and Asik, J., Modelling of the Intake Manifold Filling Dynamics, SAE-960037, 1996.

[4] Heywood, J.B., Internal Combustion Engine Fundamentals, McGraw Hill, 1988.

[5] MapleSoft, 2013, www.maplesoft.com/contact/webforms/whitepapers/engine model.aspx

[6] Moskwa, J.J., Automotive Engine Modeling for Real-Time Control, Ph.D. Dissertation, Massachusetts Institute of Technology, 1988.

[7] OPTIMUS – Design for real, Munich, 2013, www.isko-engineers.de

[8] Pacejka, H.B., Tire and Vehicle Dynamics, Society of Automotive Engineers, Warrendale, PA, 2002.

[9] Saeedi, M., A Mean Value Internal Combustion Engine in MapleSim, MSc thesis, University of Waterloo, 2010.

## The Authors:

Dr., Thanh-Son, Dao, Applications Engineer, MapleSoft, Waterloo, ON, Canada

Dr., Johannes, Friebe, Sales Director Europe MapleSim, Maplesoft Europe, Munich, Germany

Dipl.-Tech.Math., Christine, Schwarz, Applications Engineer, ISKO engineers, Munich, Germany