

Automatic Simplification of Hybrid Powertrain Models for Use in Optimization

Nikolce Murgovski, Jonas Sjöberg and Jonas Fredriksson

Department of Signals and Systems
Automatic Control, Automation and Mechatronics
Chalmers University of Technology, Gothenburg, 41296, SWEDEN
E-mail: nikolce.murgovski@chalmers.se

This paper describes solutions to a number of problems related to the automatic optimization of hybrid electric powertrains. A tool has been developed which for a given dynamic vehicle model, driving cycle and optimization criterion, produces a simplified powertrain model in the form of static maps, before dynamic programming is used to find an optimal torque split that minimizes the chosen optimization criterion. The tool decides automatically upon the operating range of the vehicle and can also operate on non-transparent vehicle models.

Topics / Powertrain & Drivetrain Control, Modeling and Simulation Technology, Energy Efficient Vehicles

1. INTRODUCTION

Hybrid electric vehicles (HEVs) belong to the group of alternative powertrain solutions, potentially beneficial in aspect of fuel economy and environmental reasons. The traditional internal combustion engine (ICE) in HEVs is complemented with one, or several electric motors (EMs), and an energy storage unit, typically a battery. This gives the vehicle two or more power sources, where the electric source can also be used as a generator. See, eg, [1] for an overview on hybrid vehicles.

In comparison to conventional vehicles, HEVs require an additional control algorithm to decide how the demanded torque, originating from the driver gas pedal, should be divided into contributions from the two power sources. This is a delicate control problem where the optimal solution depends both, on the vehicle design parameters, and the dominating external information which influences the best power split in the future driving scenario, ie, speed and altitude as a function of time. This information describes the need of energy in the near future, and with help of this information it can be decided, eg, if the battery should be emptied because of an expected inflow of re-generated energy. See [2] for more insight on this.

The tool developed in this work assesses the potential of the parallel HEV configuration by evaluating reasonable sizing of different subsystems, such as the battery. Hence, given a vehicle model and a driving cycle, the tool generates the optimal power-split strategy. The approach is deterministic since the demanded torque and vehicle speed trajectories are perfectly known and the strategy is optimal only for the given driving cycle. This control is not to be implemented in real time, but used for assessment of powertrain capabilities to meet the targets and

constraints early in the powertrain design process.

The optimization is based on dynamic programming, [3]. A weakness of this algorithm is that the computational time increases exponentially with the number of state variables. For this reason, in place of a dynamic vehicle model, a simplified powertrain model is produced where fast states are removed. The simplified powertrain model is represented with a static map in the form of lookup tables. The tool produces these maps automatically by performing a series of simulations of the dynamic vehicle model at gridded values of the input signals until steady state has been reached. Hence, the production of the maps includes long simulation and special measures are taken not to exaggerate this time. The automation of this process relies on the solutions of a number of problems which is a contribution of this paper. These include decision of the range of grid points in the generation of the maps, decision of necessary simulation time to obtain the map values, assess the quality of the approximate map and a strategy to make it possible to work with non-transparent models.

2. TOOL OVERVIEW AND PROBLEM FORMULATION

The tool is implemented in Matlab/Simulink environment. It is composed of two modules, one for generation of a quasi-static powertrain model and the other for power split optimization, see Fig. 1. Input data to the tool is supplied by the user and these are dynamic vehicle model, driving cycle $r(t) = [v_{dem}(t) \ h(t)]^T$ (a vector of demanded speed $v_{dem}(t)$ and altitude $h(t)$ of the driving profile) and optimization criterion. Depending on the chosen criterion, the requirements on the model change. In this pre-

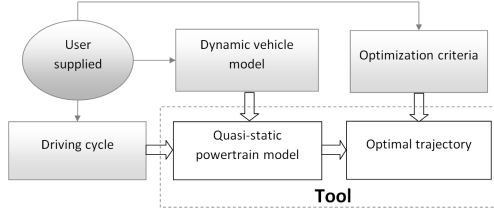


Fig. 1: Block diagram of the tool.

sensation, for didactic reasons, we choose the criterion to be fuel consumption. See [4] for more details on what changes for other optimization criteria.

2.1 Dynamic Vehicle Model

The dynamic vehicle model, given in the upper part of Fig. 2, comprises a controller f_{ctrl} , consisting of a driver model and torque split controller, vehicle dynamics f_{veh} and available outputs g_{veh} . The vehicle dynamics include a model of a parallel powertrain, see Fig. 3, and possibly other vehicle components. In general, the vehicle model is nonlinear and can be expressed as

$$\begin{aligned} \dot{x}(t) &= [\dot{soc}(t) \ \dot{x}_{rem}(t)]^T \\ &= f_{veh}(x(t), u(t), r(t), T_{load}(t)) \\ y(t) &= [v_{veh}(t) \ T_{veh}(t) \ \dot{m}_f(t)]^T \\ &= g_{veh}(x(t), u(t), r(t), T_{load}(t)) \\ u(t) &= [\gamma(t) \ u_{ICE}(t) \ u_{EM}(t) \ u_{brk}(t) \ u_{rest}(t)]^T \\ &= f_{ctrl}(x(t), y(t), r(t)) \end{aligned} \quad (1)$$

where $y(t)$ is model output consisting of the vehicle velocity $v_{veh}(t)$, traction torque $T_{veh}(t)$ and fuelflow $\dot{m}_f(t)$ needed in the optimization criterion; $x(t)$ is a vector of continuous states in the dynamic model consisting of battery state of charge $soc(t)$ kept in the quasi-static model and states $x_{rem}(t)$ that will be removed in the quasi-static model; $u(t)$ is a control signal consisting of boolean variables $u_{ICE}(t)$ and $u_{EM}(t)$ that control the availability of the power sources, boolean variable $u_{brk}(t)$ that is used to disable the friction brakes, gear $\gamma(t) \in \{0, 1, \dots, n_\gamma\}$, where n_γ is the number of gears, and all other inaccessible control signals $u_{rest}(t)$; $r(t)$ is the driving cycle; and $T_{load}(t)$, which stands for disturbance in the longitudinal vehicle dynamics in $g_{veh}(t)$, is an external torque load which is zero normally, but is used by the tool to add an extra load in simulations.

The states $x_{rem}(t)$, the control signals $u_{rest}(t)$ and the functions f_{veh} , g_{veh} and f_{ctrl} in (1) can be hidden from the user, as long as the dynamic model provides 1) access to the derivative of the battery state $\dot{soc}(t)$ and the output $y(t)$ during simulation; 2) possibility to manually decide gear $\gamma(t)$, selection of pure electrical, $u_{ICE}(t) = 0$, or pure combustion, $u_{EM}(t) = 0$, operation and ability to deactivate the brakes, $u_{brk}(t) = 0$; 3) ability to add additional value to the battery state $soc(t)$ (this corresponds to charging/discharging the battery and is used to keep the $soc(t)$ value constant although energy is taken out of the battery); and 4) possibility to add external

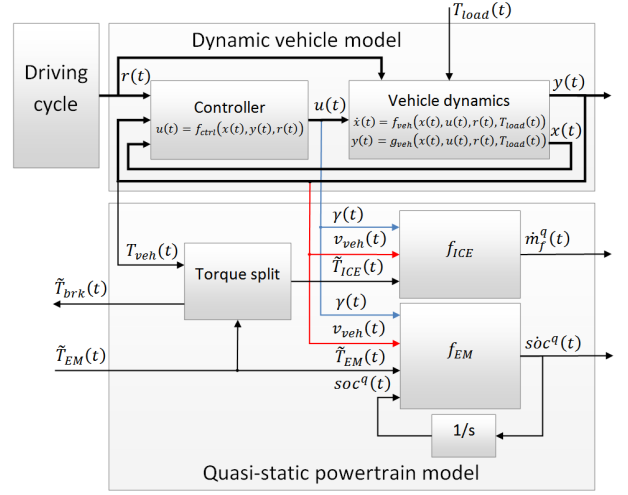


Fig. 2: Structure of the dynamic model, the quasi-static model and how signals relate to each other.

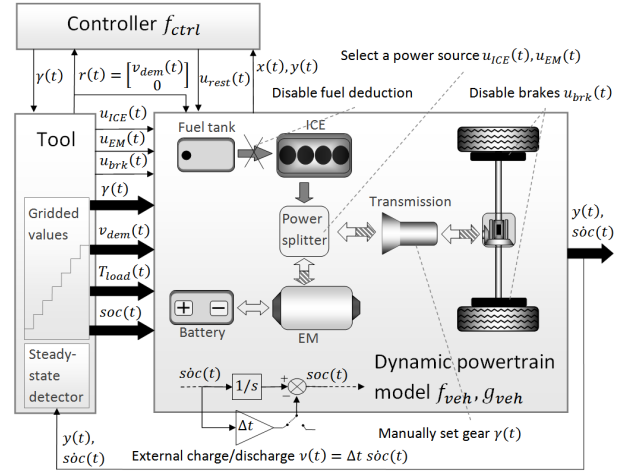


Fig. 3: Illustration of generation of a quasi-static model, given a dynamic model with parallel hybrid electric powertrain.

torque load $T_{load}(t)$ in simulation.

The process of generation of quasi-static powertrain model including requirements on the dynamic vehicle model is illustrated in Fig. 3.

2.2 Quasi-Static Powertrain Model

The quasi-static model, see Fig. 2, comprises a torque split and two lookup tables, f_{ICE} and f_{EM} , describing the two cases where the dynamic model (1) is powered only by ICE, or only by EM

$$\begin{aligned} \tilde{T}_{ICE}(t) &= \begin{cases} T_{veh}(t) - \tilde{T}_{EM}(t) & \text{if } T_{veh}(t) \geq 0 \\ 0 & \text{otherwise} \end{cases} \\ \tilde{T}_{brk}(t) &= \begin{cases} T_{veh}(t) - \tilde{T}_{EM}(t) & \text{if } T_{veh}(t) < 0 \\ 0 & \text{otherwise} \end{cases} \\ \dot{m}_f^q(t) &= f_{ICE}(\gamma(t), v_{veh}(t), \tilde{T}_{ICE}(t)) \\ \dot{soc}^q(t) &= f_{EM}(\gamma(t), v_{veh}(t), \tilde{T}_{EM}(t), soc^q(t)) \end{aligned} \quad (2)$$

where $\tilde{T}_{ICE}(t)$ is the torque at the wheels produced

by ICE, $\tilde{T}_{EM}(t)$ is the torque at the wheels produced by EM, $\tilde{T}_{brk}(t)$ is the braking torque at the wheels, $\dot{m}_f^q(t)$ and $\dot{s}oc^q(t)$ are outputs of the quasi-static model that resemble the outputs $\dot{m}_f(t)$ and $\dot{s}oc(t)$ of the dynamic model and $\gamma(t)$, $v_{veh}(t)$ and $T_{veh}(t)$ are input signals. The only states retained in the model are those needed in the optimization criterion, ie, in our case only $soc^q(t)$. The quasi-static model is open and all variables within are accessible. Its simulation entails linear interpolation of the underlying multidimensional maps f_{ICE} and f_{EM} , accomplished by the Matlab function *interp*.

Generation of lookup tables: The map f_{ICE} is generated by simulating the dynamic model over a set of gridded values produced by the tool f_{tool} , as illustrated in Fig. 3, of the input variables $\gamma(t)$, $v_{dem}(t)$ and $T_{load}(t)$, having only ICE to power the vehicle, ie $u_{EM}(t) = 0$. Equation (1) then gives us

$$\begin{aligned} u(t) &= [\gamma(t) \ 1 \ 0 \ 1 \ u_{rest}(t)]^T \\ u_{rest}(t) &= f_{ctrl}(x(t), y(t), r(t)) \\ r(t) &= [v_{dem}(t) \ 0]^T \\ \begin{bmatrix} \gamma(t) \\ v_{dem}(t) \\ T_{load}(t) \end{bmatrix} &= f_{tool}(\dot{m}_f(t), v_{veh}(t), T_{veh}(t)) \end{aligned} \quad (3)$$

where we choose to rely on the controller and let the control signals $u_{rest}(t)$ keep the values set by the controller. If this controller, which may not be known, is good, equilibrium will be reached fast. The driving cycle $r(t)$, generated by the tool, has zero altitude throughout the whole simulation, since instead of the longitudinal slope, $T_{load}(t)$ is used to give extra load to the model.

The tool keeps the gridded values constant until equilibrium is reached, followed by reading the dynamic model outputs $y(t)$, after which a new gridded combination is being generated. Values saved as map inputs are $\gamma(t)$, $v_{veh}(t)$ and $T_{veh}(t)$, as stated in (2).

For the generation of f_{EM} the vehicle is propelled only by EM, ie $u_{ICE}(t) = 0$, over gridded values of $\gamma(t)$, $v_{dem}(t)$, $T_{load}(t)$ and $soc(t)$

$$\begin{aligned} u(t) &= \begin{cases} [\gamma(t) \ 0 \ 1 \ 1 \ u_{rest}(t)]^T, & T_{load}(t) \geq 0 \\ [\gamma(t) \ 0 \ 1 \ 0 \ u_{rest}(t)]^T, & \text{otherwise} \end{cases} \\ u_{rest}(t) &= f_{ctrl}(x(t), y(t), r(t)) \\ r(t) &= [v_{dem}(t) \ 0]^T \\ \begin{bmatrix} \gamma(t) \\ v_{dem}(t) \\ T_{load}(t) \\ soc(t) \end{bmatrix} &= f_{tool}(\dot{s}oc(t), v_{veh}(t), T_{veh}(t)). \end{aligned} \quad (4)$$

For negative torque load the tool deactivates the friction brakes, ie $u_{brk}(t) = 0$, to obtain the maximum braking torque of the electric machine. Values used as map inputs are $\gamma(t)$, $v_{veh}(t)$, $T_{veh}(t)$ and $soc(t)$, as in (2).

Non-stationary points: For the generation of f_{EM} , simulations are needed with constant $soc(t)$, while energy is taken out of the battery. A work around

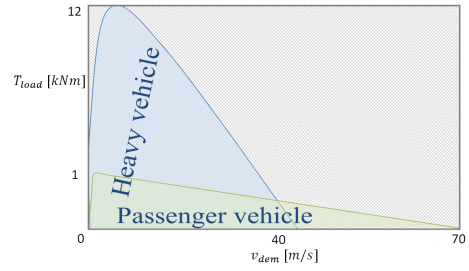


Fig. 4: Default set of gridded values.

this problem is to charge/discharge the battery with an external amount $\nu(t)$, such that

$$soc(t+1) - soc(t) = \dot{s}oc(t)\Delta t - \nu(t) = 0 \quad (5)$$

satisfied for $\nu(t) = \dot{s}oc(t)\Delta t$, as illustrated in Fig. 3.

2.3 Gridded Values and Simulation Speedup

Since no knowledge about the operating regions of the power sources is assumed, the default set of gridded values used by the tool is chosen wide, to cover the operation of both, light and heavy vehicle models, see Fig. 4. The grid density is characterized by the number of grid points n_γ , n_v , n_T and n_{soc} , for the map inputs, which are gear, demanded velocity, torque load and battery state, as in (2). The default values for n_v , n_T and n_{soc} , chosen by the authors' experience, are trade-off between small interpolation error and simulation time. This may give a large set of different input values, and to simulate for all these would take a long time. Hence, some special measures are taken to shorten this time.

First, operational velocity is automatically detected prior the map generation. The dynamic model is simulated with high reference velocity to obtain maximum velocity per gear. The number of performed tests is equal to the number of gears n_γ . Input values with torque request outside the operational region are automatically detected during the map generation. This is done by identifying that reference velocity cannot be met, which is recognized in the scenario where 1) the vehicle velocity in steady-state condition has dropped more than half of the velocity step size in the grid, when no change in velocity is demanded, or 2) the vehicle starts moving backwards. The number of performed tests needed to obtain the torque boundaries is $n_\gamma n_v$. Torque requests outside the operational region, shaded in Fig. 4, are removed automatically.

Second, the input combinations are sorted in ascending order so that for each selected gear the demanded power gradually increases. The only exceptions are the negative torque values, sorted in descending order. Input combinations are changed from one stationary value to another without restarting the simulation, so that the kinetic energy of the dynamic model is preserved. For a new grid point we start from the neighboring one, so that the dynamics between two close stationary points needs to be simulated.

The set of gridded values is fully customizable, in

the case where the user has prior knowledge of the operating regions. For example, to gain simulation time, low demanded velocities in higher gear, which are not likely to go into, can be removed manually.

2.4 Simulation Stop-Time

In this section, the problem of detecting stationarity is investigated for dynamic models that use fixed step-size solver and a method is proposed and implemented in the tool for detection of steady-state.

When the dynamic model is simulated for the different, constant, input signals, it must be decided upon when the model has reached equilibrium, ie $\dot{x}(t) = 0$. Access to all states $x(t)$ in a non-transparent system is infeasible and investigation over incomplete set of states could lead to incorrect results. To overcome the problem, the steady-state detection is based on the mean value of the available signals over a predefined period of time (a time window). There is higher expectation that the changes of all states will show up as changes in the available signals within the time window. Available signals used for the steady-state detection are $z(t) = [\dot{m}_f(t) \ v_{veh}(t) \ T_{veh}(t)]^T$ for the generation of f_{ICE} and $z(t) = [s\dot{oc}(t) \ v_{veh}(t) \ T_{veh}(t)]^T$ for the generation of f_{EM} .

Using a time window for detecting stationarity is common for many methods in literature, but most of them are based on the assumption that only noisy measurements are available and the problem then condense to detecting changes in presence of noise. Since simulations are noise-free, these methods cannot be used straight away, but could be a good starting point for the development of an accurate and efficient algorithm. For example, the F-test method of statistics [5] investigates the ratio of the mean square deviation from the mean in a time window to the mean of squared differences in successive data. A method presented by Narasimhan [6] is based on a two-stage composite statistical test to detect departures from steady-state. Consecutive time windows are considered and first the equality of the covariance matrices is tested and then a second test establishes whether the means of the two periods are equal using the Hotelling's T^2 -test [7].

Alternatively, methods have been developed that do not require a time window, see [8]-[11]. According the authors, these methods are faster, but typically also applicable only to noisy measurements.

T-test variant for steady-state detection: A direct approach to steady-state detection, known as T-test [12], investigates the linear regression slope over a time window. If the slope is close to zero, a steady-state condition is indicated. The steady-state detector used in this work is a variant of the above mentioned and is a measure based on the mean of the absolute values of consecutive derivatives $\dot{z}_j(t)$ in the time window.

A steady-state index, $\alpha_j(t)$, is defined as a mea-

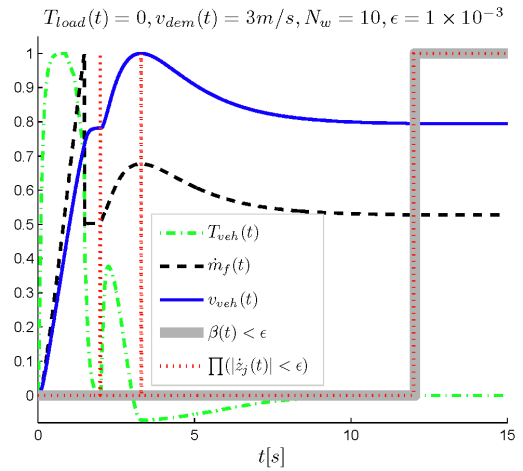


Fig. 5: Steady-state detection for a step in reference velocity. All variables are normalized to 1.

sure of the derivative of the state variable $z_j(t)$,

$$\alpha_j(t) = \frac{1}{K} \sum_{\tau=t}^{t+(N_w-1)\Delta t} |\dot{z}_j(\tau)|, \quad j = 1, 2, 3 \quad (6)$$

where N_w is the number of samples in the time window (window size), and $z(t)$ are the available signals. The constant K is used for normalization,

$$K = N_w z_{j,max}, \quad z_{j,max} = \max\{|z_j|\} \quad (7)$$

where $z_{j,max}$ is a signal magnitude automatically found by the tool. The absolute value in (6) prevents cases such as sinusoidal signal within the window being detected as steady.

Replacing the derivative with the difference

$$\dot{z}_j(t) \approx \frac{\Delta z_j(t)}{\Delta t}, \quad \Delta z_j(t) = z_j(t + \Delta t) - z_j(t) \quad (8)$$

valid for small Δt , and with some manipulation, (6) can be rewritten as

$$\alpha_j(t + \Delta t) = \alpha_j(t) + \frac{1}{\Delta t K} (|\Delta z_j(t + N_w \Delta t)| - |\Delta z_j(t)|). \quad (9)$$

The final equation (9) requires lower storage and fewer operations than (6) and thus saves simulation time. For each observed signal, the method requires one variable to be stored and four additions and two absolute values to be performed. It is only at $t = 0$ that equation (6) is computed resulting in $2N_w - 1$ additions and N_w absolute values.

Steady-state investigation of the dynamic model (1) is characterized by combining probabilities of the signals $z_j(t)$ being steady, decided by the steady-state index $1 - \alpha_j(t)$. The dynamic model is considered steady if the boolean variable, $\beta(t)$, computed following the Dempster's rule of combination, [13], is within an acceptable tolerance ϵ

$$\beta(t) = 1 - \prod_{j=1}^3 (1 - \alpha_j(t))^{\frac{\theta_j}{\sum \theta_j}} < \epsilon \quad (10)$$

where θ_j indicates the level of significance of the signal $z_j(t)$ and can be customized by the user, together with the window size N_w . By default, all signals are considered equally significant $\theta_j = 1$, the window size is 10 samples and the tolerance is $\epsilon = 1 \times 10^{-3}$.

The suggested T-test method described above is compared to a simpler method that indicates stationarity when the derivatives of all three components of $z(t)$ are close to zero, $\prod_{j=1}^3 (|\dot{z}_j(t)| \leq \epsilon)$. This comparison is shown in Fig. 5 and the difference is visible for a step in the reference velocity. The T-test method, used by the tool, avoids faulty detection of stationary condition at $t \approx 2s$ and $t \approx 3s$, where the derivative of all three signals is close to zero. Equilibrium is correctly indicated after $t \gtrsim 12s$.

2.5 Validation of the Quasi-Static Model

The tool gives the user possibility to perform validation of the quasi-static model by simulating both models over a chosen driving cycle that will be used later in optimization. The dynamic model outputs $\dot{m}_f(t)$, $\dot{soc}(t)$, are compared to the outputs of the quasi-static model, $\dot{m}_f^q(t)$, $\dot{soc}^q(t)$, obtained by simulating the quasi-static model with the true output velocity $v_{veh}(t)$ and torque $T_{veh}(t)$ of the dynamic model, see Fig. 2. The percentage of the output variation is expressed as the Euclidean norm of the difference between the outputs, normalized by the standard deviation of the dynamic model output

$$fit = 100 \left(1 - \frac{\sqrt{\sum_{t=0}^{t_f} (\dot{m}_f^q(t) - \dot{m}_f(t))^2}}{\sqrt{\sum_{t=0}^{t_f} (\dot{m}_f(t) - \bar{\dot{m}}_f)^2}} \right) \quad (11)$$

$$\bar{\dot{m}}_f = \frac{\sum_{t=0}^{t_f} \dot{m}_f(t)}{t_f + \Delta t}.$$

The fit between the derivatives of the battery states, $\dot{soc}(t)$ and $\dot{soc}^q(t)$, is computed in the same manner.

The user can also visually inspect the quality of the signals and depending on the required accuracy of the predicted fuel consumption, the user can decide upon the required fit. In particular, high accuracy is expected in non-transient regions and if this is not the case, the input set of discrete values should be refined by 1) increasing the number of discrete values, changing their distribution within the interval, or increasing the interval in which the discrete values vary and 2) including additional input signals to the maps in the quasi-static model that replace the dynamics taken away.

The grid density works a little bit like the step size by numeric solution of differential equations. If the grid is made denser, but the fit does not improve any further, then the grid is dense enough and the remaining miss-fit can be expected to be due to transients. Measures for automatic decision of grid density for the desired interpolation error are currently not implemented in the tool. Thus, if there is a need for smaller interpolation error, the user will need to manually increase the grid density. A drawback in such case is the longer simulation time that

Table 1: Gridded values for the generation of the quasi-static model.

map f_{ICE}	values	#	unit
$\gamma(t) =$	{1, 2, 3, 4, 5}	5	
$v_{dem}(t) =$	{0, 1, ..., 50}	51	[m/s]
$T_{load}(t) =$	{0, 20, ..., 1000}	51	[Nm]
map f_{EM}	values	#	unit
$\gamma(t) =$	{1, 2, 3, 4, 5}	5	
$v_{dem}(t) =$	{0, 1, ..., 50}	51	[m/s]
$T_{load}(t) =$	{-1000, -980, ..., 1000}	101	[Nm]
$soc(t) =$	{0.2, 0.267, ..., 0.8}	10	

may arise from the choice of a denser grid.

3. EXAMPLE: OPTIMIZATION OF FUEL CONSUMPTION

The tool operation is demonstrated on a dynamic HEV model of a passenger vehicle with a torque-assist parallel powertrain, see Fig. 3. The torque assist HEV have mechanically coupled engine and electric machine, which speed is imposed by the instantaneous vehicle velocity.

The chosen optimization criterion is minimization of fuel consumption over the "New European Driving Cycle", see Fig. 6, and constant road altitude. To simplify the problem we choose the gear selection strategy from the dynamic model previously simulated over the given driving cycle and we leave the torque split as the only control variable.

The tool first simplifies the dynamic model, as was described in Section 2.2. The vehicle has 5 gears and we provide custom gridded values for the other inputs, see Table 1. Note that gridded values do not need to be provided by the user, but the default ones can be used that already exist in the tool. Moreover, minimum velocity per gear is given, [0 0 4 8 13]m/s, thus all combinations with the respective gear and smaller vehicle velocity are manually removed. We leave the other combinations outside the operational region to be automatically removed by the tool, as was described in Section 2.3. The tool then simulates the vehicle model with one power source at a time and two maps f_{ICE} and f_{EM} are obtained, as in (2).

Details on the optimization, which is the next step in the tool, are further explained in [4].

3.1 Results and Discussion

For the generation of the quasi-static model (2), the dynamic model is simulated over a set of gridded values given in Table 1. Although custom gridded values were chosen to decrease the simulation time, still there are more than 220000 input combinations for the generation of f_{EM} . The tool's automatic detection of the operating region resulted in 53920 operating points for f_{EM} , which together

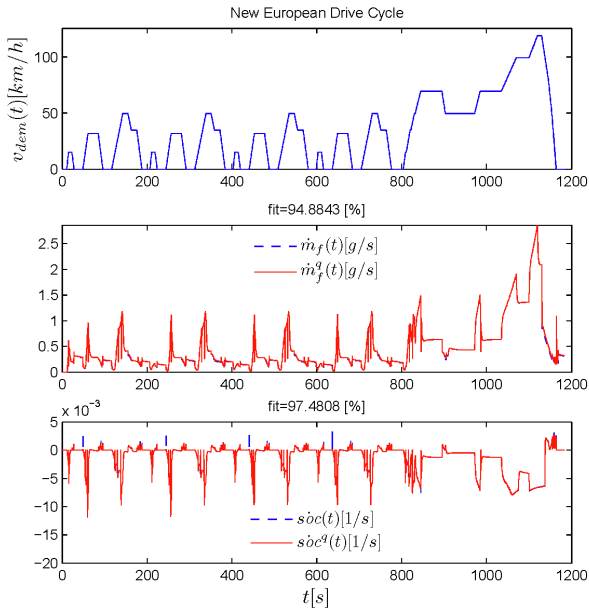


Fig. 6: Validation of the quasi-static powertrain model.

with the number of performed tests for obtaining the velocity and torque boundaries, see Section 2.3, gives $54180 = 53920 + n_\gamma + n_\gamma n_v$ input combinations for which the dynamic model was simulated. With the steady-state detector in use, the total simulation time for the generation of f_{EM} was about 35 minutes. Only for comparison, the same operating points were simulated on the same computer, but without a steady-state detector. The simulation time for each operating point was set equal to the slowest step response time for which any operating point would reach equilibrium. With this setup, the total simulation time was about 4 hours, which is about 7 times longer simulation.

After the quasi-static model is obtained, it is validated as was described in Section 2.5 and the results are given in Fig. 6. The maps, f_{ICE} and f_{EM} , are validated by simulating the dynamic and the quasi-static model with one power source at a time. Hence, for the validation of f_{EM} , when EM propels the vehicle, the battery may deplete and whenever this happens, the battery level is reset to the highest $soc(t)$ value in Table 1. The maps validation shows that the quasi-static model describes the behavior of the dynamic model well and not only in steady-state conditions, but also in transients, see Fig. 6.

4. CONCLUSIONS

This paper describes a tool for automatic powertrain simplification of a given HEV model with a parallel powertrain, to be used later in optimization. The model details can be hidden from the user as long as the model satisfies a set of requirements stated in Section 2.1. The process is automatized as much as possible, so that user insight in vehicle modeling and simulation is not necessary.

The tool first simplifies the dynamic vehicle model

into a quasi-static powertrain model, after which dynamic programming is used to optimize the criterion chosen by the user. A predefined set of gridded input values that could be customized by the user, is used for the map generation and the tool decides automatically upon the operating region, thus speeding up the simulation. A time-efficient method is presented and implemented in the tool for detection of steady-state.

A possible future extension of the tool is to adapt it to other powertrain configurations and other optimization criteria. The map generation process could be furthermore improved by automatically deciding on the necessary grid resolution to obtain certain interpolation accuracy.

REFERENCES

- [1] L. Guzzela and A. Sciarretta, *Vehicle propulsion systems, introduction to modeling and optimization*, 2nd ed. Springer, 2007.
- [2] L. Johannesson, “Predictive control of hybrid electric vehicles on prescribed routes,” Ph.D. dissertation, Chalmers University of Technology, Göteborg, Sweden, 2009.
- [3] R. Bellman, *Dynamic Programming*. Princeton Univ Pr, June 1957.
- [4] N. Murgovski, J. Sjöberg, and J. Fredriksson, “A tool for generating optimal control laws for hybrid electric powertrains,” *IFAC Symposium on Advances in Automotive Control*, 2010.
- [5] E. L. Crow, F. A. Davis, and M. W. Maxfield, *Statistics Manual*. Dover Publications, 1960.
- [6] S. Narasimhan, R. S. H. Mah, A. C. Tamhane, J. W. Woodward, and J. C. Hale, “A composite statistical test for detecting changes of steady states,” *AICHE Journal*, vol. 32, no. 9, pp. 1409–1418, September 1986.
- [7] H. Hotelling, “The generalization of student’s ratio,” *Ann. Math. Statist.*, vol. 2, pp. 360–378, 1931.
- [8] F. Flehmig, R. v. Watzdorf, and W. Marquardt, “Identification of trends in process measurements using the wavelet transform,” *Computers chem. Engng*, vol. 22, pp. S491–S496, 1998.
- [9] S. Cao and R. R. Rhinehart, “An efficient method for on-line identification of steady state,” *J. Proc. Cont.*, vol. 5, no. 6, pp. 363–374, 1995.
- [10] R. R. Rhinehart, “A novel method for automated identification of steady state,” in *Proceedings of the American Control Conference*, Seattle, Washington, June 1995.
- [11] P. R. Brown and R. R. Rhinehart, “Demonstration of a method for automated steady-state identification in multivariable systems,” *Hydrocarbon Processing*, vol. 79, no. 9, pp. 79–83, 2000.
- [12] D. C. Montgomery and G. C. Runger, *Applied Statistics and Probability for Engineers*. John Wiley and Sons, Inc., USA, 2003.
- [13] G. Shafer, *A mathematical theory of evidence*. Princeton University Press, 1976.