A tool for generating optimal control laws for hybrid electric powertrains ⋆

Nikolce Murgovski, Jonas Sjöberg, Jonas Fredriksson

Department of Signals and Systems, Control, Automation and Mechatronics Laboratory, Chalmers University of Technology, Göteborg, Sweden (e-mail: nikolce.murgovski@chalmers.se, jonas.sjoberg@chalmers.se, jonas.fredriksson@chalmers.se).

Abstract: This paper describes the development of a tool for automatic generation of optimal power management strategies. Given the user inputs, which are dynamic vehicle model, driving cycle and optimization criterion, the tool first produces a simplified powertrain model in the form of static maps, before dynamic programming is used to find an optimal torque split which minimizes the chosen criterion. The tool does not require a transparent vehicle model, which makes it possible to work on models hidden for intellectual property reasons.

Keywords: optimal control, hybrid vehicles, parallel powertrain, non-transparent models

1. INTRODUCTION

The interest of alternative powertrain solutions has increased during the last decade due to fuel economy and environmental reasons. Hybrid electric vehicles (HEVs) are one group of such alternatives where the traditional combustion engine is complemented with one, or several electric motors, and an energy buffer, typically a battery. This gives the vehicle two power sources, where the electric one can also be used in reversed mode, ie, as generator. The potential fuel savings mainly depend on 1) the possibility to re-generate brake energy by using the electric motors as generators and storing the energy in the buffer, and 2) the possibility to run the engine at more efficient load conditions while storing the excess energy in the buffer. See, eg, Guzzela and Sciarretta (2007) for an overview on hybrid vehicles.

Given the torque command originating from the driver's gas pedal, the control algorithm has to decide how the demanded torque should be divided into contributions from the two power sources. This is a delicate control problem where the optimal solution depends not only on vehicle design parameters. The dominating external information which influences the best power split is 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 Johannesson (2009) for more insight on this.

At the end, the vehicle performance, eg fuel consumption, depends on the configuration including the design and sizing of the components of the HEV and the torque split control algorithm. This work describes the development of a tool to assess the first of these issues, ie to estimate the potential of the parallel HEV configuration without developing a control algorithm. Such a tool is useful for gaining information of the potential of a certain parallel powertrain, and to evaluate 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, Bellman (1957), Naidu (2002). A weakness of this algorithm is that the computational time increases exponentially with the number of state variables (curse of dimensionality, Bertsekas (2000)). For this reason, in place of dynamic vehicle model, a simplified powertrain model is produced where fast states are removed. This can be done since transients typically do not influence fuel consumption and emissions significantly, Kolmanovsky et al. (1999), Pisu and Rizzoni (2007), Koot et al. (2005).

The simplified powertrain model is represented with a static map in the form of look-up tables. The tool produces these maps automatically by performing 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, Murgovski et al. (2010).

The methodology, to simplify a dynamic model and then apply dynamic programming, is not new in connection to evaluating HEV configurations. See for example Kang et al. (1999), Kolmanovsky et al. (2005), Guzzella and Amstutz (1999), Lin et al. (2003), Sundström et al. (2008). There are also tools for modeling and simulation of HEVs which support the evaluation of the size of the design

⋆ This work was supported by Volvo Trucks, Volvo Cars, Saab Automobile, BAE Hägglunds, EMFO, Vinnova.

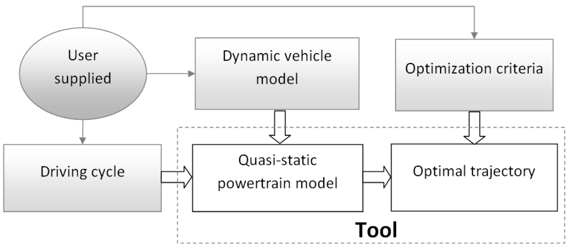

Fig. 1. Block diagram of the tool. Vehicle model, driving cycle and optimization criteria are given by the user.

parameters, JANUS by Bumby et al. (1985), SIMPLEV by Cole (1991), ADVISOR by Wipke et al. (1999), Markel et al. (2002), QSS-TB by Guzzella and Amstutz (1999), HYSDEL by Torrisi and Bemporad (2004), CAPSim by Fredriksson et al. (2006), ADAMS/Car, CARSim and others, Kolmanovsky et al. (2005), Butler et al. (1999), Mierlo and Maggetto (2001), Hayat et al. (2003), Liu and Peng (2008). Instead, the contribution of this paper is the automation of the configuration evaluation. This is done by concatenation of the two main steps into one tool. No interaction from the user is needed in the intermediate step to set values for the algorithms, which they might not know.

The paper is outlined as follows: tool overview and problem formulation is discussed in Section 2, description of the tool is given in Section 3, an example of minimization of fuel consumption is given in Section 4 and the process of optimizing other optimization criteria is explained in Section 5.

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 $r(t) = [v_{dem}(t) \quad h(t)]^T$ (a vector of demanded speed $v_{dem}(t)$ and altitude of the driving profile $h(t)$) and optimization criterion. Depending on the chosen criterion, the requirements on the model change. In this presentation, for didactic reasons, we choose the criterion to be fuel consumption. Later, in Section 5, we will explain what changes for other criteria.

After user info is supplied, a quasi-static model is automatically obtained from the dynamic model where most of the dynamics are removed. This is done by simulating at gridded constant input values. The purpose of using a quasi-static model in the second step, where the actual optimization takes place, is to obtain faster simulation without losing much of the model accuracy. This led to non-trivial technical issues that need to be solved, which is a contribution of this paper:

- Generation of a map of the vehicle powertrain without taking components apart such as engine, el. machine, battery, gearbox... This also makes it possible to work with hidden models, further explained in Section 3.2.
- Obtain map values at non-stationary points. For example, the derivative of battery state of charge

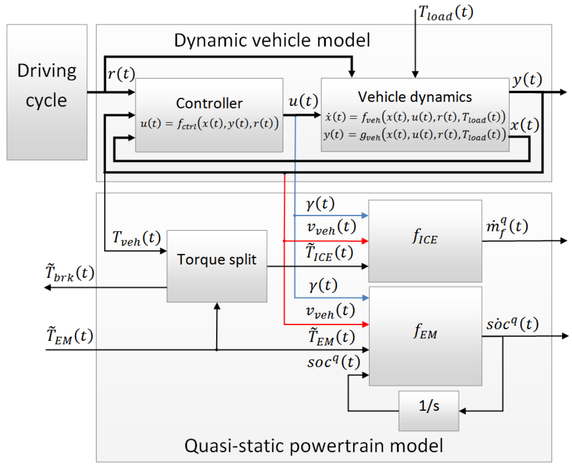

Fig. 2. Structure of the dynamic model, the quasi-static model and how signals relate to each other. The dynamic model comprises a controller f_{ctrl}, vehicle dynamics f_{veh} and output g_{veh}. The quasi-static model consists of a torque split and two lookup tables f_{ICE} and f_{EM}.

$\dot{soc}(t)$ is a function of, among other signals, $soc(t)$. The state $soc(t)$ is constant only for $\dot{soc}(t) = 0$, but evaluation is needed for other values of $\dot{soc}(t)$, which then, by definition, gives a non-constant $soc(t)$. A work around this contradiction is further explained in Section 3.2.

3. TOOL

3.1 Dynamic Vehicle Model

This subsection describes requirements on the dynamic vehicle model for it to be used by the optimization tool.

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, 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 this is a nonlinear model and can be expressed as

$$\begin{aligned} \dot{x}(t) &= f_{veh}(x(t), u(t), r(t), T_{load}(t)) \\ y(t) &= g_{veh}(x(t), u(t), r(t), T_{load}(t)) \\ u(t) &= f_{ctrl}(x(t), y(t), r(t)) \end{aligned} \tag{1}$$

where $y(t) = [v_{veh}(t) \quad T_{veh}(t) \quad \dot{m}_f(t)]^T$ 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; $x(t) = [soc(t) \quad x_r(t)^T]^T$ is a vector of continuous states in the dynamic model consisting of a battery state of charge $soc(t)$ kept in the quasi-static model and states $x_r(t)$ that will be removed in the quasi-static model; $u(t) = [\gamma(t) \quad T_{ICE}(t) \quad T_{EM}(t) \quad T_{brk}(t)]^T$ is a control signal consisting of gear $\gamma(t)$, combustion engine torque $T_{ICE}(t)$, electric machine torque $T_{EM}(t)$ and braking torque $T_{brk}(t)$; $r(t)$ is the driving cycle; and $T_{load}(t)$, which can be considered as 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.

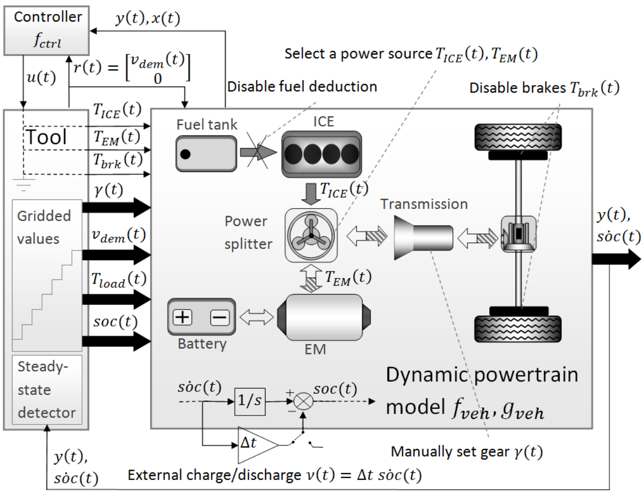

Fig. 3. Illustration of generation of a quasi-static model, given a dynamic model with parallel hybrid electric powertrain. The tool gives gridded values for $\gamma(t)$, $v_{dem}(t)$, $T_{load}(t)$ and $soc(t)$, and decides when to ground the control signals $T_{ICE}(t)$, $T_{EM}(t)$ and $T_{brk}(t)$.

The states $x_r(t)$, the functions f_{veh}, g_{veh} and f_{ctrl} in (1) can be hidden from the user, as long as the following requirements, further explained in Section 3.2, are satisfied:

(1) Access to the derivative of the battery state $\dot{soc}(t)$ and the output $y(t)$ during simulation.
(2) Possibility to disconnect the controller. This involves manual selection of pure electrical or pure combustion operation by $T_{ICE}(t), T_{EM}(t)$, manual gear selection $\gamma(t)$ and ability to deactivate the brakes $T_{brk}(t)$.
(3) Possibility to add fuel. This signal is not of importance later in the paper and notation is omitted.
(4) 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.
(5) Possibility to add external torque load $T_{load}(t)$.

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

3.2 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 the internal combustion engine (ICE), or only by the electric machine (EM)

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

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, $\dot{m}_f^q(t)$ and $\dot{soc}^q(t)$ are outputs of the quasi-static model that resemble the outputs $\dot{m}_f(t)$ and $\dot{soc}(t)$ 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 *interpn*.

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

$$
\begin{aligned}
u(t) &= \begin{bmatrix} \gamma(t) \\ T_{ICE}(t) \\ 0 \\ T_{brk}(t) \end{bmatrix}, \quad r(t) = \begin{bmatrix} v_{dem}(t) \\ 0 \end{bmatrix} \\
\begin{bmatrix} T_{ICE}(t) \\ T_{brk}(t) \end{bmatrix} &= f_{ctrl}(x(t), y(t), r(t))
\end{aligned} \tag{3}
$$

where we choose to rely on the controller and let the rest of the control signals, $T_{ICE}(t)$ and $T_{brk}(t)$, keep the values set by the controller f_{ctrl}. The driving cycle $r(t)$, generated by the tool, has zero altitude throughout the whole simulation, since instead of the longitudinal slope, the tool uses $T_{load}(t)$ to give extra load to the vehicle. 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. More information on the steady-state detection can be found in Murgovski et al. (2010).

Values saved as map inputs are $\gamma(t)$, $v_{veh}(t)$ and $T_{veh}(t)$, as in (2).

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

$$
\begin{aligned}
u(t) &= \begin{cases} [\gamma(t) \quad 0 \quad T_{EM}(t) \quad T_{brk}(t)]^T & \text{if } T_{load}(t) \geq 0 \\ [\gamma(t) \quad 0 \quad T_{EM}(t) \quad 0]^T & \text{otherwise} \end{cases} \\
r(t) &= \begin{bmatrix} v_{dem}(t) \\ 0 \end{bmatrix}, \quad \begin{bmatrix} T_{EM}(t) \\ T_{brk}(t) \end{bmatrix} = f_{ctrl}(x(t), y(t), r(t))
\end{aligned} \tag{4}
$$

where values for $soc(t)$ are set through an external signal for charging/discharging the battery, assuming that the vehicle model posses such a signal as was stated in Requirement 4) in Section 3.1. For negative torque load the tool deactivates the friction brakes, ie $T_{brk}(t) = 0$, to obtain the maximum braking torque of the electric machine. Values saved 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 this problem is to charge/disharge the battery with an external amount $\nu(t)$, such that

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

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

3.3 Optimization

The second step in the tool, after model simplification, is optimization of the energy management problem. The optimization problem is formulated by the optimization criterion, eg fuel consumption, and a driving cycle chosen by the user. Then, the tool generates control $\tilde{T}^*_{EM}$, optimal for that cycle, that minimizes the cost defined by the criterion

$$
\begin{aligned}
&\tilde{T}^*_{EM}(t) = \min_{\tilde{T}_{EM}(t)} \int_0^{t_f} \dot{m}^q_f(t)dt \\
&\quad \text{subject to} \\
&T_{veh}(t) = \tilde{T}_{ICE}(t) + \tilde{T}_{EM}(t) + T_{brk}(t) \\
&soc^q(0) = soc^q(t_f) \\
&\tilde{T}_{ICE,min}(t) \leq \tilde{T}_{ICE}(t) \leq \tilde{T}_{ICE,max}(t) \\
&\tilde{T}_{EM,min}(t) \leq \tilde{T}_{EM}(t) \leq \tilde{T}_{EM,max}(t) \\
&soc_{min} \leq soc^q(t) \leq soc_{max} \\
&soc_l \leq soc^q(t) \leq soc_h
\end{aligned}
\tag{6}
$$

where t_f is the final time, $T_{ICE,min}(t)$, $T_{ICE,max}(t)$, $T_{EM,min}(t)$, $T_{EM,max}(t)$, soc_{min} and soc_{max} are torque and battery state boundaries obtained automatically from the quasi-static model and soc_l and soc_h define the desired range of the battery state that could be given by the user and is used to avoid excessive wear of the battery. The tool approaches (6) as a constraint satisfaction problem, Bertsekas (1996), and dynamic programming, Bellman (1957), is used to find a numerical solution.

4. 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" and constant road altitude. Assuming, for simplicity, that the gear shifting strategy is given in advance as the highest gear capable of delivering the demanded vehicle speed, the only control signal in the vehicle model is the torque split.

The first step in the tool operation is model simplification. As was described in Section 3.2, the dynamic model is to be simulated over a set of gridded values. 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, the tool automatically detects the operational regions and speeds-up the map generation process, see Murgovski et al. (2010). 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).

The second step in the tool operation is optimization. The tool first simulates the dynamic model on the given driving cycle to obtain the true velocity $v_{veh}(t)$, torque at the wheels $T_{veh}(t)$ and gear $\gamma(t)$. Then, these signals are used as inputs to the quasi-static model during the optimization. The energy management problem is defined as in (6), where the desired range of the battery state is chosen $[soc_l \quad soc_h] = [0.4 \quad 0.6]$. The engine is switched off whenever the torque demanded by the engine is zero.

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	

4.1 Results and discussion

The maps f_{ICE} and f_{EM} in the quasi-static model are multidimensional, and studying them can be a cumbersome process, since they are not easy for illustration. Given gearing and wheel radius of the dynamic vehicle model, the tool can present these maps as two-dimensional, see Fig. 4 - 6. Map inputs are the ICE torque, T^q_{ICE} and the ICE speed ω^q_{ICE} (T^q_{EM} and ω^q_{EM} for EM respectively), acting on the drive-shaft between the gearbox and the power-split unit. These are not maps of ICE and EM, but could be considered as such if there were no powertrain losses. Indeed, it could be noticed in Fig. 4 - 6 that because of the powertrain losses the maximum torque per vehicle speed from the quasi-static model is lower than the actual torque limits from the dynamic vehicle model. Similarly, the minimum EM torque from the quasi-static model in Fig. 6 is higher than the minimum torque limit in the dynamic model. Another difference with an actual ICE map is that the map in Fig. 4 has operating points below the idle engine speed (about $750rpm$). The reason is that when the map f_{ICE} was generated, low vehicle velocities could be achieved because of the slipping clutch. The ICE and EM torque limits of the dynamic vehicle model are used here only for comparison and are not needed by the tool otherwise.

In Fig. 4 and Fig. 5 the brake specific fuel consumption (BSFC) of the engine is given and Fig. 6 illustrates the normalized efficiency of the electric machine. The operating points in Fig. 4 are obtained when the vehicle is run without using the electric machine. In Fig. 5 and 6 the optimal operating points of ICE and EM are given when both ICE and EM can propel the vehicle. Five operational modes are considered, which are 1) pure combustion, the vehicle is run by ICE only and EM is turned off, 2) boosting, ICE is assisted by EM, 3) discharging, EM propels the vehicle with ICE turned off, 4) recharging, ICE generates more torque than needed and the surplus is used to recharge the battery and 5) regenerative braking, EM is used to charge the battery with ICE turned off.

When the torque demand is low and the battery is not fully charged, the vehicle is run in recharging mode. An extra torque is delivered from ICE to shift the operating points toward the higher efficiency regions. The efficiency of EM decides the quantity of surplus torque, and this is the reason why ICE is not run with higher torque, where it is most efficient, see Fig. 5. Choosing electric machine with higher efficiency for lower torque load or downsizing

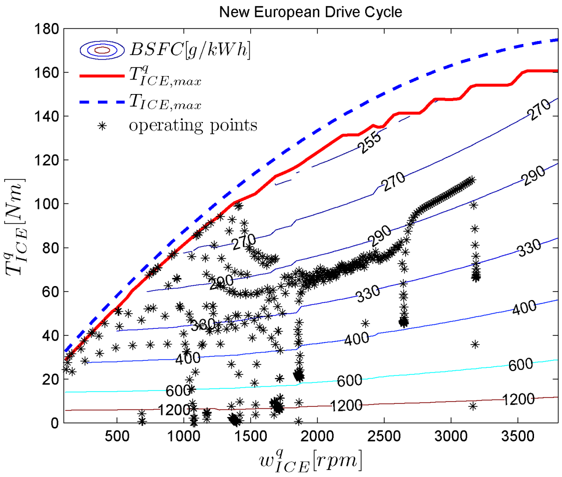

Fig. 4. Non-optimal operating points when only ICE propels the vehicle. The maximum torque limit, obtained by the map f_{ICE}, is lower than the actual torque limit of the dynamic engine model, which is due to the powertrain losses included in f_{ICE}.

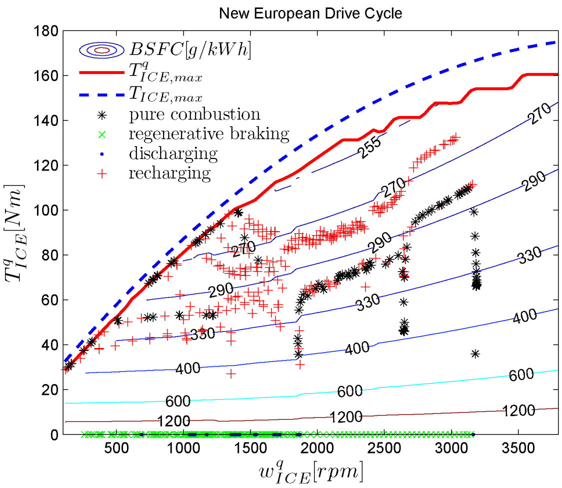

Fig. 5. ICE operating points of the optimal power management strategy. Most of the operating points are shifted toward higher torque operation where the engine is most efficient and the excess torque is used to charge the battery. Lower torque demands are achieved by the electric machine, while the engine is turned off.

the engine could lead to increased fuel economy. It is interesting to note that no operating points belong to boosting mode. This perhaps justifies the idea of downsizing the engine and use EM assistance when needed.

The optimal control, resulted in approximately 6.1 l/100km fuel consumption compared to about 8 l/100km when the vehicle is operated by ICE only. The improvement in fuel consumption will be lower if the problem of cold start of the engine is considered, or if transients in engine torque or the frequent switch between fuel and electric operation mode are penalized, see Sciarretta et al. (2004). The result nevertheless shows the advantage of the hybrid electric vehicle over the conventional fuel-cell vehicle.

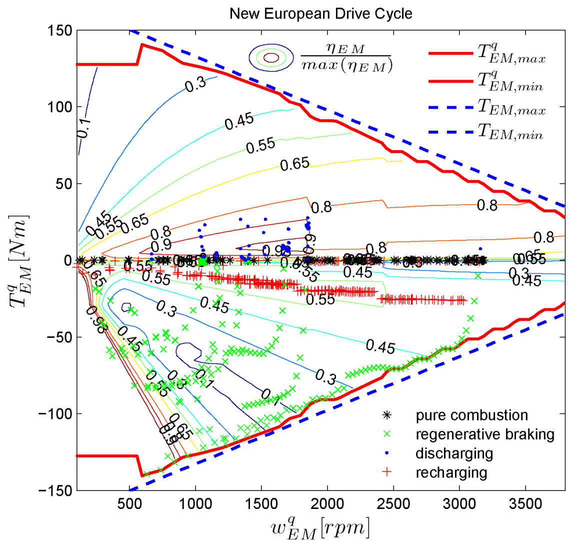

Fig. 6. EM operating points of the optimal power management strategy. For negative torque demands, the electric machine is used for regenerate braking as much as possible. EM propels the vehicle only for small torque demands and is used in recharging mode only in its most efficient region.

5. OTHER OPTIMIZATION CRITERIA

The tool offers several optimization criteria, fuel consumption, pollutant emissions, or combinations of both, for the user to choose from. Corresponding cost function and constraints (6) are prepared in the tool and automatically loaded for each chosen criterion. The outputs and states kept in the quasi-static model will also change by the chosen criterion and the dynamic model must provide access to them. For example, minimization of pollutants, such as NOx, HC, CO2, CO, will require the model to provide the flow of the pollutants $y_k(t)$ as outputs $y(t) = [v_{veh}(t) \quad T_{veh}(t) \quad y_k(t)^T]^T$. Access to the derivative of states $x_k(t)$, such as battery state, boost pressure, engine temperature and catalyst temperature, on which the output $y_k(t)$ mainly depends on, should also be provided by the model, as well as ability to externally add value to $x_k(t)$ to keep the states constant, as was stated in Requirement 4) in Section 3.1 and further explained in Section 3.2.

The quasi-static model depends on the chosen criterion and in (2) it was given for the case when fuel consumption was to be minimized. For a general choice of criterion, instead of only one state $soc^q(t)$ and one output $\dot{m}^q_f(t)$, the quasi-static model will have $x^q_k(t)$ states and $y^q_k(t)$ outputs

$$\begin{aligned}
\begin{bmatrix} y^q_{ICE}(t) \\ \dot{x}^q_{ICE}(t) \end{bmatrix} &= f_{ICE}(\gamma(t), v_{veh}(t), \tilde{T}_{ICE}(t), x^q_{ICE}(t)) \\
\begin{bmatrix} y^q_{EM}(t) \\ \dot{x}^q_{EM}(t) \end{bmatrix} &= f_{EM}(\gamma(t), v_{veh}(t), \tilde{T}_{EM}(t), x^q_{EM}(t)) \\
x^q_k(t) &= [x^q_{ICE}(t)^T \quad x^q_{EM}(t)^T]^T \\
y^q_k(t) &= [y^q_{ICE}(t)^T \quad y^q_{EM}(t)^T]^T.
\end{aligned} \tag{7}$$

where $x^q_k(t)$ and $y^q_k(t)$ resemble the states $x_k(t)$ and outputs $y_k(t)$ of the dynamic model.

6. CONCLUSIONS AND FUTURE WORK

This paper describes a tool that can be used for generation of optimal power management strategies for a given HEV model with a parallel powertrain. The model details can be hidden from the user as long as the model satisfies the requirements stated in Section 3.1. The process is automized as much as possible, so that user insight in vehicle modeling and simulation is not necessary.

There is a potential for future extension of the tool operation on other powertrain configurations and optimization criteria. Additionally, the possibility of generating optimal controller for real-time usage, see Johannesson et al. (2005), is to be investigated.

REFERENCES

R. Bellman. *Dynamic Programming*. Princeton Univ Pr, June 1957. ISBN 069107951X.

D. P. Bertsekas. *Constrained Optimization and Lagrange Multiplier Methods*. Athena Scientific, January 1 1996. ISBN 1886529043.

D. P. Bertsekas. *Dynamic Programming and Optimal Control*. Athena Scientific, November 15 2000. ISBN 1886529094.

J. Bumby, P. Clarke, and I. Forster. Computer modelling of the automotive energy requirements for internal combustion engine and battery electric-powered vehicles. In *IEE proceedings*, volume 132, September 1985.

K. L. Butler, M. Ehsani, and P. Kamath. A matlab-based modeling and simulation package for electric and hybrid electric vehicle design. *IEEE Transactions on Vehicular Technology*, 48(6), November 1999.

G. H. Cole. SIMPLEV: A simple electric vehicle simulation program version 1.0. June 1991.

J. Fredriksson, J. Larsson, J. Sjöberg, and P. Krus. Evaluating hybrid electric and fuel cell vehicles using the CAPSim simulation environment. In *Proceedings of the 22nd International Battery Hybrid and Fuel Cell Electric Vehicle Symposium and Exposition*, Yokohama, Japan, October 2006.

L. Guzzela and A. Sciarretta. *Vehicle propulsion systems, introduction to modeling and optimization*. Springer, 2 edition, 2007. ISBN 3540746919.

L. Guzzella and A. Amstutz. CAE tools for quasi-static modeling and optimization of hybrid powertrains. *IEEE Transactions on Vehicular Technology*, 48(6), November 1999.

O. Hayat, M. Lebrun, and E. Domingues. Powertrain drivability evaluation: Analysis and simplification of dynamic models. In *SAE World Congress*, Detroit, Michigan, March 2003.

L. Johannesson. *Predictive Control pf Hybrid Electric Vehicles on Prescribed Routes*. PhD thesis, Chalmers University of Technology, Göteborg, Sweden, 2009.

L. Johannesson, M. Åsbogård, and B. Egardt. Assessing the potential of predictive control for hybrid vehicle powertrains using stochastic dynamic programming. *IEEE Transactions on Intelligent Transportation Systems*, 2005.

J. Kang, I. Kolmanovsky, and J. Grizzle. Approximate dynamic programming solutions for lean burn engine aftertreatment. In *Proceedings of the 38th IEEE Conference on Decision and Control*, volume 2, pages 1703–1708, 1999.

I. Kolmanovsky, M. van Nieuwstadt, and J. Sun. Optimization of complex powertrain systems for fuel economy and emissions. In *Proceedings of the 1999 IEEE International Conference on Control Applications*, Kohala Coast-Island of Hawai, USA, August 22-27 1999.

I. V. Kolmanovsky, S. N. Sivashankar, and J. Sun. Optimal control-based powertrain feasibility assessment: A software implementation perspective. In *American Control Conference*, Portland, OR, USA, June 8-10 2005.

M. Koot, J. T. B. A. Kessels, B. de Jager, W. P. M. H. Heemels, P. P. J. van den Bosch, and M. Steinbuch. Energy management strategies for vehicular electric power systems. *IEEE transactions on vehicular technology*, 54 (3), May 2005.

C. Lin, H. Peng, J. W. Grizzle, and J. Kang. Power management strategy for a parallel hybrid electric truck. *IEEE transactions on control systems technology*, 11(6), July 2003.

J. Liu and H. Peng. Automated modelling of power-split hybrid vehicles. In *Proceedings of the 17th World Congress The International Federation of Automatic Control*, Seoul, Korea, July 6-11 2008.

T. Markel, A. Brooker, T. Hendricks, V. Johnson, K. Kelly, B. Kramer, M. O'Keefe, S. Sprik, and K. Wipke. ADVISOR: a systems analysis tool for advanced vehicle modeling. *Journal of Power Sources 110*, page 255, 2002.

J. V. Mierlo and G. Maggetto. Vehicle simulation program: a tool to evaluate hybrid power management strategies based on an innovative iteration algorithm. *Proc Instn Mech Engrs*, 215 Part D, 2001.

N. Murgovski, J. Sjöberg, and J. Fredriksson. Automatic simplification of hybrid powertrain models for use in optimization. *International Symposium on Advanced Vehicle Control*, 2010.

D. S. Naidu. *Optimal Control Systems*. CRC Press, July 2002. ISBN 0849308925.

P. Pisu and G. Rizzoni. A comparative study of supervisory control strategies for hybrid electric vehicles. *IEEE*, 2007.

A. Sciarretta, M. Back, and L. Guzzella. Optimal control of parallel hybrid electric vehicles. *IEEE transactions on control systems technology*, 12(3), May 2004.

O. Sundström, L. Guzzella, and P. Soltic. Optimal hybridization in two parallel hybrid electric vehicles using dynamic programming. In *Proceedings of the 17th World Congress The International Federation of Automatic Control*, Seoul, Korea, July 6-11 2008.

F. D. Torrisi and A. Bemporad. HYSDEL-a tool for generating computational hybrid models for analysis and synthesis problems. *IEEE transactions on control systems technology*, 12(2), 2004.

K. Wipke, M. Cuddy, and S. Burch. ADVISOR 2.1: A user-friendly advanced powertrain simulation using a combined backward/forward approach. *IEEE Transactions on Vehicular Technology*, August 1999.