



CHALMERS

Chalmers Publication Library

A methodology and a tool for evaluating hybrid electric powertrain configurations

This document has been downloaded from Chalmers Publication Library (CPL). It is the author's version of a work that was accepted for publication in:

International Journal of Electric and Hybrid Vehicles (ISSN: 1751-4088)

Citation for the published paper:

Murgovski, N. ; Sjöberg, J. ; Fredriksson, J. (2011) "A methodology and a tool for evaluating hybrid electric powertrain configurations". International Journal of Electric and Hybrid Vehicles, vol. 3(3), pp. 219-245.

<http://dx.doi.org/10.1504/IJEHV.2011.043870>

Downloaded from: <http://publications.lib.chalmers.se/publication/148748>

Notice: Changes introduced as a result of publishing processes such as copy-editing and formatting may not be reflected in this document. For a definitive version of this work, please refer to the published source. Please note that access to the published version might require a subscription.

Chalmers Publication Library (CPL) offers the possibility of retrieving research publications produced at Chalmers University of Technology. It covers all types of publications: articles, dissertations, licentiate theses, masters theses, conference papers, reports etc. Since 2006 it is the official tool for Chalmers official publication statistics. To ensure that Chalmers research results are disseminated as widely as possible, an Open Access Policy has been adopted. The CPL service is administrated and maintained by Chalmers Library.

(article starts on next page)

A methodology and a tool for evaluating hybrid electric powertrain configurations

Nikolce Murgovski*, Jonas Sjöberg and Jonas Fredriksson

Department of Signals and Systems,
Chalmers University of Technology,
Gothenburg, Sweden,
E-mail: nikolce.murgovski@chalmers.se
E-mail: jonas.sjoberg@chalmers.se
E-mail: jonas.fredriksson@chalmers.se
*Corresponding author

Abstract: This paper describes a methodology for automatic optimization of hybrid electric powertrains. This methodology is developed and implemented in a tool, CAPSimO, and the paper is written in the form of describing the tool. Given the user inputs, which are dynamic vehicle model, driving cycle and optimization criterion, the tool first produces a simplified powertrain model in a form of static maps, before dynamic programming is used to find an optimal power split which minimizes the chosen criterion. The tool does not require that the vehicle model is transparent, which makes it possible to work on models hidden for intellectual property reasons. The paper presents two examples of powertrain evaluation, in terms of fuel consumption, for a parallel and a parallel-series powertrain.

Keywords: HEV; Hybrid electric vehicle; Power management; Optimal control; Non-transparent model; Black-box model; Powertrain sizing; Model simplification; Static map; Steady-state detection.

Reference to this paper should be made as follows: Murgovski, N., Sjöberg, J. and Fredriksson, J. (2011) 'A methodology and a tool for evaluating hybrid electric powertrain configurations', *Int. J. Electric and Hybrid Vehicles*, Vol. x, No. x, pp.xxxx.

Biographical notes: Nikolce Murgovski received his MSc degree in Applied Physics 2007 at Chalmers University of Technology. Currently he is a PhD student at the department of Signals and Systems at Chalmers. His research activities include power management control of automotive powertrains.

Jonas Sjöberg received his MSc degree in Applied Physics 1989 from Uppsala University, Sweden and his PhD degree in Automatic Control in 1995 from Linköping University, Sweden. Since July 2000 he is professor in Mechatronics at the department of Signals and Systems at Chalmers University of Technology. His research activities include modeling and simulation of dynamic systems, based on physical insights, as well as black-box methods based on statistics and measured data.

Jonas Fredriksson received his MSc degree in computer science engineering at Luleå University of Technology, Sweden, in 1997 and his PhD in automatic control at Chalmers University of Technology, Sweden, in 2002. Currently he is associate professor at the Department of Signals and Systems at Chalmers. His research activities include modeling, control and simulation, with a special interest in automotive powertrains.

1 Introduction

The interest for 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, i.e. as a generator. The potential fuel savings mainly depend on 1) the possibility to re-generate braking 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, e.g. Guzzella and Sciarretta (2007) for an overview on hybrid vehicles.

Given the power command originating from the driver's gas pedal, the control algorithm has to decide how the demanded power 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, i.e. speed and altitude as a function of time (Johannesson et al., 2005). This information describes the need of energy in the near future, and with help of this information it can be decided, e.g. 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, e.g. the fuel consumption, depends on the configuration including the design and sizing of the components of the HEV and the power split control algorithm. This work describes the development of a tool to assess the first of these issues, i.e. to estimate the potential of the HEV powertrain configuration without developing a realtime control algorithm. Such a tool is useful for gaining information of the potential of a certain 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 power 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). A weakness of this algorithm is that the computational time increases exponentially with the number of state variables. For this reason, in place of dynamic vehicle model, a simplified powertrain model is produced where fast states are removed.

The simplified powertrain model is represented with static maps. The tool produces these maps automatically by performing a series of simulations of the dynamic 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 to not exaggerate this time.

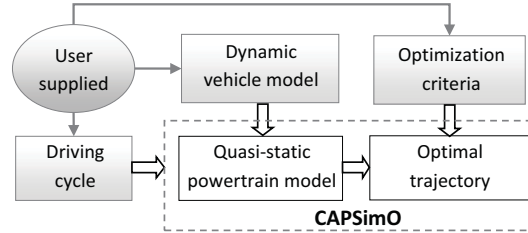
The methodology to simplify a dynamic model (typically in quasi-static relations) and then apply dynamic programming, is not new in connection to evaluating HEV configurations. See for example Kang et al. (1999); Kolmanovsky et al. (2005); Guzzella and Amstutz (1999); Lin et al. (2003); Sundström et al. (2008). There are also tools for modeling and simulation of HEVs which support the evaluation of the size of design parameters, VEHLIB (Trigui et al., 2004), AMESim, Dymola, JANUS (Bumby et al., 1985), SIMPLEV (Cole, 1991), ADVISOR (Wipke et al., 1999), Markel et al. (2002), QSS-TB (Guzzella and Amstutz, 1999), HYSDEL (Torrise and Bemporad, 2004), CAPSim (Fredriksson et al., 2006), ADAMS/Car, CARSim and others (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 automation of this process relies on the solutions of a number of problems including, deciding the grid points in the generation of the maps, deciding necessary simulation time to obtain the map values, assessing the quality of the approximate map, a strategy to make it possible to work with non-transparent models. These problems are further described in Section 2.

To sum up, the tool has the following features, which are new, to the best knowledge of the authors:

- The only requirement on the model is to provide access to some general variables and that it has a power split that can be fully controlled. This makes it possible to work with non-transparent models, e.g. models which are compiled, or hidden of intellectual property reasons.
- The tool offers a number of optimization criteria to choose between, but the user can also specify criterion and constraints themselves.
- The tool integrates the model simplification and the optimization with minimized need of interaction from the user. The user does not need to understand details of the model. Instead, parameter values for the algorithms are set automatically, although an experienced user can change some of the parameters.

The paper is outlined as follows: tool overview and problem formulation is discussed in Section 2; detailed description of the tool, requirements on the dynamic vehicle model with a parallel powertrain and solutions to technical issues

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

are given in Section 3; extension of the tool operation on a model with a parallel-series powertrain is presented in Section 4; custom optimization criteria and user interface aspects are discussed in Section 5; two examples of fuel consumption minimization for a parallel and parallel-series HEV powertrain are given in Section 6 and 7; the paper is ended with conclusions in Section 8.

2 Tool overview and problem formulation

The tool is implemented in Matlab[®] environment. It is composed of two modules, one for generation of a quasi-static powertrain model and the other for power split optimization, see Figure 1. Input data to the tool is supplied by the user and these are dynamic vehicle model, driving cycle $r(t) = [v_r(t) \ \alpha_r(t)]^T$ (a vector of demanded speed and slope of the driving profile) and optimization criterion. Depending on the chosen criterion and the powertrain configuration (series, parallel, parallel-series), the requirements on the model change.

In this presentation, for didactic reasons, we start by a model with a parallel powertrain where the criterion is minimization of fuel consumption, under the assumption that the only state in the simplified model is the battery state of charge (SOC). Later, in Section 4, a model with series-parallel powertrain is considered where fuel consumption is minimized for a simplified model with four states: gear, SOC, engine coolant temperature and engine speed. It is explained afterward, in Section 5, what changes for other optimization criteria as well as the additional, possible, but not necessary entries that could be supplied by an advanced user.

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 a list of non-trivial technical issues that needed 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.
- Simulate as little as possible by 1) deciding the grid points for at which the map is generated so that unnecessary grid points are avoided, 2)

deciding upon if the grid is dense enough by validating the simplified, grid based model and 3) deciding upon the stop time for the simulation in the generation of the map. This is further explained in Section 3.5, 3.6 and 3.7.

- Obtain map values at non-stationary points. For example, the derivative of the battery state of charge $\dot{soc}(t)$ 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. A work around this contradiction is further explained in Section 3.2.

3 Parallel powertrain

This section describes the requirements on the dynamic vehicle model with a parallel powertrain for it to be used by the tool. The automatic model simplification into quasi-static relations and the optimization of the simplified model are also described.

3.1 Dynamic Vehicle Model

The dynamic vehicle model comprises a controller f_{ctrl} , consisting of a driver model and a power split controller, continuous and discrete vehicle dynamics f_c and f_d and continuous and discrete output functions g_c and g_d . The vehicle includes a model of a parallel powertrain, see Figure 2, and possibly other vehicle components. In general, the vehicle model is a nonlinear hybrid system (Witsenhausen, 1996) and can be expressed as

$$\begin{aligned} \dot{x}_c(t) &= f_c(x(t), u_c(t), \nu_c(t)), & x_d^+(t) &= f_d(x(t), u(t)) \\ y_c(t) &= g_c(x(t), u_c(t)), & y_d^+(t) &= g_d(x(t), u(t)) \\ u(t) &= f_{ctrl}(x(t), y(t), r(t)) \\ x(t) &= \begin{bmatrix} x_c(t) \\ x_d(t) \end{bmatrix}, & y(t) &= \begin{bmatrix} y_c(t) \\ y_d(t) \end{bmatrix}, & u(t) &= \begin{bmatrix} u_c(t) \\ u_d(t) \end{bmatrix} \end{aligned} \quad (1)$$

where $r(t)$ are the reference signals, $x_c(t)$, $u_c(t)$ and $y_c(t)$ are continuous states, inputs and outputs respectively, and $\nu_c(t)$ are disturbances to the continuous states. The signals with index d take discrete values and they change at specific times. That is, e.g. $f_d(x(t), u(t))$ is constant ($= x_d(t)$) up to some time \tilde{t} at which it jumps to a new value, i.e. $x_d^+(\tilde{t}) = f_d(x(\tilde{t}), u(\tilde{t}))$. After that f_d remains constant to the next jump in value.

The signals indexed c and d are split so that only the signals kept in the simplified model become explicit, that is

$$x(t) = \begin{bmatrix} x_{ck}(t) \\ x_{dk}(t) \\ x_r(t) \end{bmatrix}, \quad u(t) = \begin{bmatrix} u_{ck}(t) \\ u_{dk}(t) \\ u_r(t) \end{bmatrix}, \quad \nu_c(t) = \begin{bmatrix} \nu_{ck}(t) \\ \nu_{cr}(t) \end{bmatrix}, \quad y(t) = \begin{bmatrix} y_k(t) \\ y_r(t) \end{bmatrix}. \quad (2)$$

The rest of the signals are put into the signals indexed r .

For the optimization criterion of minimizing fuel consumption, described later in Section 3.8, the kept and accessible signals are

$$\begin{aligned}
 x_{ck}(t) &= soc(t) \\
 x_{dk}(t) &= [\gamma(t) x_{ICE}(t) x_{EM}(t) x_{brk}(t)]^T \\
 u_{ck}(t) &= F_{load}(t) \\
 u_{dk}(t) &= [u_\gamma(t) u_{ICE}(t) u_{EM}(t) u_{brk}(t)]^T \\
 y_k(t) &= [\dot{m}_f(t) v_w(t) F_w(t)]^T
 \end{aligned} \tag{3}$$

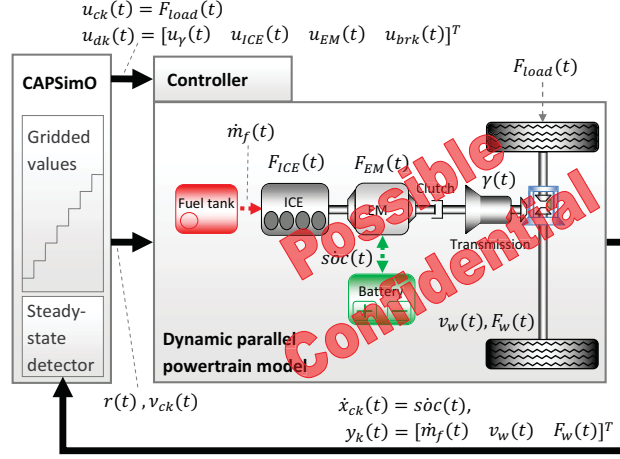
where $\gamma(t)$ is the gear, $x_{ICE}(t)$, $x_{EM}(t)$ and $x_{brk}(t)$ are binary states (0 or 1) that represent the on/off state of the internal combustion engine (ICE), on/off state of the electric machine (EM) and availability of the friction brakes, respectively, and $soc(t)$ is the battery state of charge. The outputs that are to be accessed are the vehicle velocity at the wheels $v_w(t)$, the traction force at the wheels $F_w(t)$ and the fuel flow $\dot{m}_f(t)$ needed in the optimization criterion. The discrete control signals that are to be accessed are $u_\gamma(t)$, $u_{ICE}(t)$, $u_{EM}(t)$ and $u_{brk}(t)$. These signals are used to shift gear and decide the availability of the ICE, the EM and the friction brakes, respectively. The input force $F_{load}(t)$, which is an external disturbance in the longitudinal vehicle dynamics, is zero normally, but is used by the tool to set the load in simulations.

The signals $x_r(t)$, $u_r(t)$, $y_r(t)$ and $\nu_{cr}(t)$ in (2), and the functions f_c , f_d , g_c , g_d and f_{ctrl} in (1) can be hidden from the user, as long as the model provides:

1. Access to the input events $u_{dk}(t)$. The signals $u_\gamma(t)$, $u_{ICE}(t)$ and $u_{EM}(t)$ are used by the tool to select gear and to select pure combustion or pure electrical operation. The tool uses $u_{brk}(t)$ to deactivate the friction brakes in order to find the maximum braking force of the EM.
2. Access to the discrete states $x_{dk}(t)$ and to the derivative of the continuous states $\dot{x}_{ck}(t)$, in our case $\dot{soc}(t)$, which are to be kept in the simplified model.
3. Access to the outputs $y_k(t)$.
4. Possibility to add an additional value $\nu_{ck}(t)$ to the continuous states that are to be kept in the simplified model. For the accessible continuous states in (3) this corresponds to charging/discharging the battery and is used to keep $soc(t)$ constant although energy is taken out of the battery.
5. Possibility to add an external load $F_{load}(t)$ acting at the wheels. Alternatively, the tool can use the slope of the road to simulate a force load.
6. Possibility to add fuel. This signal is not of importance later in the paper and notation is omitted.

The process of automatic simplification of the dynamic vehicle model is illustrated in Figure 2.

Figure 2 Dynamic powertrain model and the simplification process. The dynamic model, which may not be transparent, includes a parallel HEV powertrain. The tool decides the availability of the power sources and the friction brakes and simulates the dynamic model with constant gridded values for the inputs and the disturbances of the continuous states that will be kept in the simplified model.



3.2 Quasi-static powertrain model

The simplified, quasi-static model is a backward simulation model that satisfies the power balance equation

$$(\tilde{F}_{EM}(t) + \tilde{F}_{ICE}(t) + \tilde{F}_{brk}(t))v_w(t) = F_w(t)v_w(t) \quad (4)$$

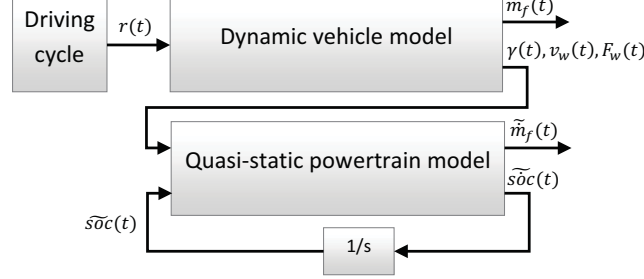
where $\tilde{F}_{ICE}(t)$ is the force generated by the ICE, $\tilde{F}_{EM}(t)$ is the force of the EM, $\tilde{F}_{brk}(t)$ is the force dissipated at the friction brakes and \sim is used to distinguish the signals of the quasi-static model from the corresponding signals of the dynamic model. The power demanded from the quasi-static model over a given driving cycle is the true output power of the dynamic model, $F_w(t)v_w(t)$, see Figure 3, simulated over the same driving cycle, where it is assumed that the controller in the dynamic model follows closely the reference velocity.

Which signals that are to be kept in the quasi-static model depends on the optimization criterion and the accessible signals in the dynamic model. For the criterion described later in Section 3.8, the gear shifting sequence is assumed identical to the sequence the dynamic model chooses on the given driving cycle and no dynamics are included for changing speed and switching on/off the ICE and EM. Hence, the quasi-static model consists of only one state and three control signals

$$\begin{aligned} \tilde{x}(t) &= \tilde{soc}(t) \\ \tilde{u}(t) &= [\tilde{F}_{ICE}(t) \tilde{F}_{EM}(t) \tilde{F}_{brk}(t)]^T \end{aligned} \quad (5)$$

where $\tilde{soc}(t)$ resembles $soc(t)$ of the dynamic model. The states derivatives and the fuel-flow $\tilde{m}_f(t)$ needed in the optimization criterion are obtained from two lookup

Figure 3 Simulation of the dynamic and the quasi-static model. Input to the quasi-static model are the true vehicle velocity $v_w(t)$ and force at the wheels $F_w(t)$ obtained from the dynamic vehicle model.



tables, f_{ICE} and f_{EM} , describing the two cases where the dynamic model (1) is powered only by the ICE, or the EM

$$\begin{aligned}\tilde{m}_f(t) &= f_{ICE}(\gamma(t), v_w(t), \tilde{F}_{ICE}(t)) \\ \tilde{s}_{oc}(t) &= f_{EM}(\gamma(t), v_w(t), \tilde{F}_{EM}(t), \tilde{s}_{oc}(t))\end{aligned}\quad (6)$$

where $\tilde{m}_f(t)$ resembles $\dot{m}_f(t)$ of the dynamic model.

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*.

3.3 Generation of lookup tables

The tool f_{tool} simplifies the dynamic model automatically by simulating the model at constant gridded values for the reference signals $r(t)$, the inputs $u_{ck}(t)$ and $u_{dk}(t)$, and the continuous state disturbances $\nu_{ck}(t)$

$$\begin{aligned}\begin{bmatrix} r(t) & u_{ck}(t) & u_{dk}(t) & \nu_{ck}(t) \end{bmatrix}^T &= f_{tool}(\dot{x}_{ck}(t), y_k(t)) \\ r(t) &= [v_r(t) \ 0]^T\end{aligned}\quad (7)$$

where the driving cycle $r(t)$ has zero altitude throughout the whole simulation, since instead of the longitudinal slope the tool uses $F_{load}(t) = u_{ck}(t)$, see (3), to give an extra load to the model. For the rest of the control signals $u_r(t)$ we choose to rely on the controller f_{ctrl} and let them keep the values set by the controller.

The tool keeps the input values constant until equilibrium is reached, after which a new gridded combination is being generated. The equilibrium condition, further explained in Section 3.5, is automatically detected by the tool from the derivative of the accessible continuous states $\dot{x}_{ck}(t)$ and the outputs $y_k(t)$ of the dynamic model. If the controller f_{ctrl} , which may not be known, is good, equilibrium may be reached fast.

For the generation of f_{ICE} in (6) the dynamic vehicle model is powered only by the ICE, yielding

$$\begin{aligned}x_{dk}(t) &= [\gamma(t) \ 1 \ 0 \ 1]^T \\ \begin{bmatrix} v_r(t) & F_{load}(t) & u_{dk}(t) \end{bmatrix}^T &= f_{tool}(\dot{m}_f(t), v_w(t), F_w(t))\end{aligned}\quad (8)$$

where the steady-state detection is based on $\dot{m}_f(t)$, $v_w(t)$ and $F_w(t)$. Values saved as map inputs are $\gamma(t)$, $v_w(t)$ and $F_w(t)$ as stated in (6), where from (4) it follows that $\tilde{F}_{ICE}(t) = F_w(t)$.

For the generation of f_{EM} the vehicle is propelled only by the EM, yielding

$$x_{dk}(t) = \begin{cases} \begin{bmatrix} \gamma(t) & 0 & 1 & 1 \end{bmatrix}^T, & F_{load}(t) \geq 0 \\ \begin{bmatrix} \gamma(t) & 0 & 1 & 0 \end{bmatrix}^T, & \text{otherwise} \end{cases} \quad (9)$$

$$\begin{bmatrix} v_r(t) & F_{load}(t) & u_{dk}(t) & \nu_{ck}(t) \end{bmatrix}^T = f_{tool}(s\dot{c}(t), v_w(t), F_w(t))$$

where gridded values for $soc(t)$, according the requirement 4) in Section 3.1, are set through the state disturbance $\nu_{ck}(t)$. For negative force load the tool deactivates the friction brakes, i.e. $u_{brk}(t) = 0$, to obtain the maximum braking force of the EM. The detection of stationarity is based on $s\dot{c}(t)$, $v_w(t)$ and $F_w(t)$ which together with $\gamma(t)$ are saved as map inputs, as in (6), where from (4) it follows that $\tilde{F}_{EM}(t) = F_w(t)$.

3.4 Non-stationary points

The process of model simplification requires simulations with constant state values. The tool keeps the discrete states $x_{dk}(t)$ constant by controlling the input events $u_{dk}(t)$. However, constant values for the continuous states $x_{ck}(t)$ are also needed although the system may not be in equilibrium with regard to these states, i.e. $\dot{x}_{ck}(t) \neq 0$. A work around this problem, enclosed in requirement 4) in Section 3.1, is the possibility to add an external disturbance $\nu_{ck}(t)$ to $x_{ck}(t)$.

Consider a solver for differential equations with a sampling time that may not be constant and let $x_{ck}(t_k)$ be the continuous states at time instant t_k . Then, the successive states obtained by the solver can be expressed as

$$x_{ck}(t_{k+1}) = x_{ck}(t_k) + \int_{t_k}^{t_{k+1}} (\dot{x}_{ck}(t) + \nu_{ck}(t)) dt. \quad (10)$$

Then the tool can keep the states constant, $x_{ck}(t_{k+1}) = x_{ck}(t_k)$, by controlling the state disturbance, such that

$$\nu_{ck}(t) = -\dot{x}_{ck}(t) \quad (11)$$

assuming that the model provides access to $\dot{x}_{ck}(t)$ as was stated in requirement 2) in Section 3.1.

3.5 Simulation stop time

In this section, the problem of detecting stationarity is investigated for dynamic models that use fixed step-size solver. The section gives brief overview on the existing methods and shows the implementation of a variant of the T-test method (Montgomery and Runger, 2003). The proposed T-test variant reduces simulation time and memory storage and improves accuracy by weighting the stationarity of several signals simultaneously.

When the dynamic model is simulated for the different constant input signals, it must be decided upon when the model has reached equilibrium, $\dot{x}_c(t) = 0$, $x_d^+(t) = x_d(t)$. In a non-transparent system however, access to all states may not be possible and in fact only few signals may be available. For example, only three continuous signals, $z(t) = [\dot{m}_f(t) v_w(t) F_w(t)]^T$, are accessible for the generation of f_{ICE} in (8) and also three signals, $z(t) = [s\dot{o}c(t) v_w(t) F_w(t)]^T$, are accessible for the generation of f_{EM} in (9). Hence, stationarity is investigated over a predefined period of time (a time window), since there is a higher expectation that the changes of all states will show up as changes in the available signals within the time window.

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 (Crow et al., 1960) 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 et al. (1986) is based on a two-stage composite statistical test to detect departures from steady-state. The method first tests the equality of the covariance matrices of consecutive time windows and then a second test establishes whether the means of the two periods are equal using the Hotelling's T^2 -test (Hotelling, 1931).

Alternatively, methods have been developed that do not require a time window, see Flehmig et al. (1998); Cao and Rhinehart (1995); Rhinehart (1995); Brown and Rhinehart (2000). According the authors, these methods are faster, but typically also applicable only to noisy measurements.

A direct approach to steady-state detection, known as T-test (Montgomery and Runger, 2003), 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 the consecutive derivatives $\dot{z}_j(t)$ in the time window.

For each component $z_j(t)$ in $z(t)$, a steady-state index $\alpha_j(t)$ is defined as

$$\alpha_j(t) = \frac{1}{K} \int_t^{t+N_w h} |\dot{z}_j(\tau)| d\tau, \quad j = 1, 2, 3 \quad (12)$$

where h is the sampling interval and N_w is the number of samples in the time window (window size). The constant K is used for normalization

$$K = N_w z_{j,max}, \quad z_{j,max} = \max\{|z_j|\} \quad (13)$$

where $z_{j,max}$ is the signal magnitude automatically found by the tool.

Replacing the derivative with the difference

$$\dot{z}_j(t) \approx \frac{\Delta z_j(t)}{h}, \quad \Delta z_j(t) = z_j(t+h) - z_j(t) \quad (14)$$

valid for small sampling time h , and with some manipulation, (12) can be rewritten as

$$\alpha_j(t+h) \approx \alpha_j(t) + \frac{1}{hK} (|\Delta z_j(t+N_w h)| - |\Delta z_j(t)|). \quad (15)$$

Equation (15) requires lower storage and fewer operations than (12) and thus saves simulation time. For each observed signal and at each time instance, the method requires one variable to be stored and one product, four additions and two absolute values to be performed. Only at the start of the simulation, $2N_w$ additions and N_w absolute values are needed.

The stationarity of the whole system is computed following the Dempster's rule of combination (Shafer, 1976) over all the signals $z(t)$. The system is considered in equilibrium when the variable

$$\beta(t) = 1 - \prod_{j=1}^3 (1 - \alpha_j(t))^{\frac{\theta_j}{\sum \theta_j}} \quad (16)$$

is within an acceptable tolerance ϵ . The parameter 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 variant described above is compared to a simpler method that indicates stationarity when the derivatives of all the three components of $z(t)$ are close to zero

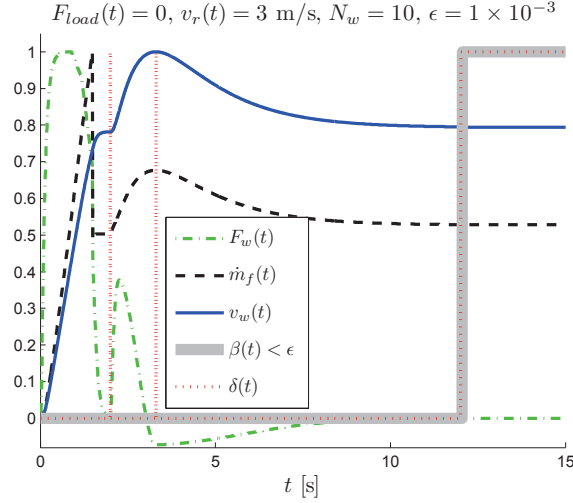
$$\delta(t) = \prod_{j=1}^3 \left(\frac{1}{z_{j,max}} |\dot{z}_j(t)| \leq \epsilon \right). \quad (17)$$

This comparison is shown in Figure 4 and the difference is visible for a step in reference velocity. The modified T-test method, used by the tool, avoids faulty detection of stationary condition at $t \approx 2$ s and $t \approx 3$ s, where the derivative of all the three signals is close to zero. Equilibrium is correctly found after $t \gtrsim 12$ s.

3.6 Gridded values and simulation speedup

The model simplification process, as mentioned in Section 3.3, requires the values for $r(t)$, $u_{ck}(t)$, $u_{dk}(t)$ and $\nu_{ck}(t)$ to be set by the tool for at which the dynamic model is to be simulated. The tool uses these signals to excite the system in order to obtain a quasi-static model that can accurately replace the dynamic model under steady-state conditions. This procedure requires a grid on the continuous signals $r(t)$, $u_{ck}(t)$ and $\nu_{ck}(t)$ ($u_{dk}(t)$ already belong to a discrete set), and obtaining this grid may not be trivial since a decision need to be made on the grid size and density.

Figure 4 Steady-state detection for a step in reference velocity. Investigated variables are fuelflow, velocity and force at the wheels of the dynamic model. The result of two methods is presented, 1) the T-test variant used by the tool and 2) test if the derivatives of the investigated signals are close to zero. The second method had false indications at $t \approx 2$ s and $t \approx 3$ s, while the T-test variant method performed better by correctly detecting equilibrium. All variables are normalized to 1.



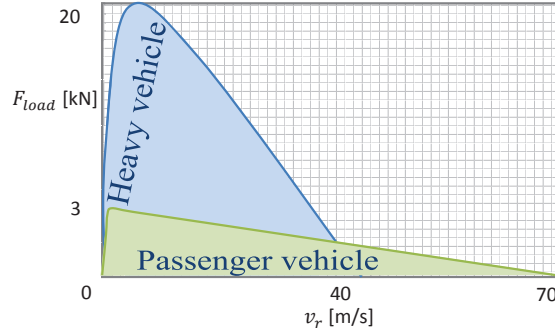
3.6.1 Grid size

Since no previous knowledge on 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. For example, the speed may range from 0 to 70 m/s and the force load up to 20 kN, see Figure 5. A dense grid, discussed later in Section 3.6.2, may give a large set of grid points, most of them outside the vehicle's operating range, and to simulate for all these may take a long time. Hence, some special measures are taken to shorten this time.

First, the grid points demanding positive vehicle power are sorted in ascending order so that the power gradually increases (the grid points demanding negative power are sorted in descending order so that the absolute value of the power increases). Input combinations are changed from one stationary value to another without restarting the simulation so that the kinetic energy of the system is preserved. For a new grid point the tool starts from the neighboring one so that only the transient dynamics between two close stationary points needs to be simulated.

Second, assume that the grid is uniform and n_γ and n_v are the number of grid points for the map inputs, gear and velocity respectively, as in (8). Then the tool can automatically remove the points outside the operating vehicle range by identifying that reference values cannot be met. Consider the case, for example, where for a fixed gear and velocity a force load is demanded that the vehicle cannot satisfy. Then the vehicle may end up in 1) never getting steady, but starts moving backwards (the velocity gets negative) or 2) equilibrium, but either its velocity

Figure 5 Default set of gridded values. The set is chosen wide to cover the operation of various vehicle models. The non-operating region, shaded in the figure for the two examples of a passenger and heavy vehicle, is automatically removed by the tool.



or traction force do not meet the reference, i.e. they differ from the reference by more than half of their step size in the grid. In either case, the tool will identify the maximum force load the vehicle can cope and all grid points with higher force demands at the same gear and velocity will be automatically removed from the grid. The number of performed tests to obtain the force limits per needed gear and velocity is $n_\gamma n_v$.

3.6.2 Grid density

The grid density is characterized by the number of grid points for the continuous map inputs, such as n_v for velocity. The grid points use default step sizes chosen by the authors' experience and are trade-off between small interpolation error and simulation time. For example, the velocity grid is uniformly spaced with step size of 1 m/s and the force values use increment of 50 N. The simplified model is validated and if there is a significant miss-fit between the dynamic and the quasi-static model, see Section 3.7, then new grid is created with half the step sizes and the simplification process is repeated only at the new grid points.

The procedure to describe 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, see Section 3.7 for details.

3.7 Validation of the quasi-static model

Validation of the quasi-static model (6) is the process of determining the degree to which its simulation is an accurate representation of the dynamic vehicle model (1). The tool gives the user possibility to perform validation by simulating both models over a chosen driving cycle that will be used later in optimization. The signals $\dot{m}_f(t)$ and $\dot{s}oc(t)$ of the dynamic model are compared to the outputs of the quasi-static model, $\hat{\dot{m}}_f(t)$ and $\hat{\dot{s}oc}(t)$, obtained by simulating the quasi-static model with the true output velocity $v_w(t)$ and force $F_w(t)$ of the dynamic model. The percentage of the output variation is expressed as the root-mean-square of

the difference between the outputs, normalized by the standard deviation of the dynamic model output

$$fit = 100 \left(1 - \frac{\sqrt{\int_{t_0}^{t_f} (\tilde{z}(t) - z(t))^2 dt}}{\sqrt{\int_{t_0}^{t_f} (z(t) - \bar{z})^2 dt}} \right), \quad \bar{z} = \frac{1}{t_f - t_0} \int_{t_0}^{t_f} z(t) dt \quad (18)$$

where t_0 and t_f are the initial and the final time of the drive cycle, $\tilde{z}(t)$ is the output of the quasi-static model, either $\tilde{m}_f(t)$ or $\tilde{soc}(t)$, and $z(t)$ is the respective signal in the dynamic model.

The quasi-static model is considered accurate if for all the outputs of its lookup tables the fit is at least $fit_{lim} = 80\%$.

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, then the user may need to include additional input signals to the maps in the quasi-static model. This is further explained in Section 5.

3.8 Optimization criterion

The tool offers a number of optimization criteria to choose from. In this section, for didactic reasons, a simple criterion of minimizing fuel consumption is presented, where the gear shifting strategy is assumed known and the ICE and EM can instantaneously change speed and on/off state. Later in Section 4.3 a more complex criterion is considered.

The optimization criterion is defined by an objective function and a list of constraints

$$\tilde{u}^*(t) = \arg \min_{\tilde{u}(t)} \int_{t_0}^{t_f} \Psi(t) dt$$

$$\Psi(t) = \tilde{m}_f^q(t)$$

subject to (4), (6) and

$$\tilde{F}_{ICE}(t) \in [\tilde{F}_{ICE,min}(t), \tilde{F}_{ICE,max}(t)] \quad (19a)$$

$$\tilde{F}_{EM}(t) \in [\tilde{F}_{EM,min}(t), \tilde{F}_{EM,max}(t)] \quad (19b)$$

$$\tilde{F}_{brk}(t) \leq 0 \quad (19c)$$

$$\tilde{soc}(t_0) = soc_0 \quad (19d)$$

$$\tilde{soc}(t_f) = soc_f \quad (19e)$$

$$\tilde{soc}(t) \in [soc_{min}, soc_{max}] \quad (19f)$$

where $\tilde{u}^*(t)$ are the optimal control signals, $\Psi(t)$ is the instantaneous cost, soc_0 and soc_f are the initial and final SOC value and (19a-19c) impose physical limits to the powertrain components. The limits (19a) and (19b) are found automatically by the tool from the maps f_{ICE} and f_{EM} in (6) and are function of the corresponding map inputs. The presence of (19d-19f) can be decided by the user, where (19f) is imposed to avoid excessive wear of the battery.

3.9 Optimal trajectory

The next step in the tool, after the model simplification, is the optimization of the energy management (19). This kind of problem is conveniently solved using dynamic programming, Bertsekas (2000), since it handles nonlinearities and constraints in a straightforward way. Dynamic programming uses the Bellman's principle of optimality, Bellman (1957), and solves the problem via backwards recursion

$$\begin{aligned} J^*(\tilde{x}(t_f), t_f) &= c(\tilde{x}(t_f) - x_f)^2 \\ J^*(\tilde{x}(t_k), t_k) &= \min_{\tilde{u}(t_k)} \left(\int_{t_k}^{t_{k+1}} \Psi(t) dt + J^*(\tilde{x}(t_{k+1}), t_{k+1}) \right) \\ t \in \mathcal{T}, \quad \tilde{u}(t) \in \mathcal{U}, \quad \tilde{x}(t) \in \mathcal{X} \end{aligned} \quad (20)$$

where J^* is the optimal cost-to-go function from time t_k to the final time t_f , c is a penalty coefficient, usually a large number which relaxes (19e), $\tilde{x}(t)$ and $\tilde{u}(t)$ are as in (5) and $x_f = soc_f$. The cost function is calculated over a grid of the time, the state and the control signal, i.e. the sets \mathcal{T} , \mathcal{U} and \mathcal{X} are discrete. The grid resolution determines the accuracy of the solution. For state values that do not belong to \mathcal{X} , the cost is obtained by linear interpolation in J^* .

4 Parallel-Series (Combined) powertrain

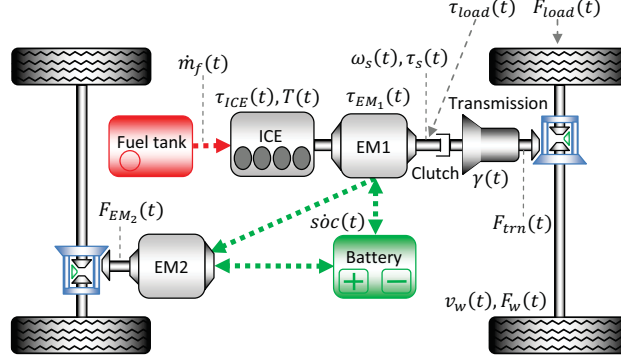
This section describes the requirements the dynamic model with a combined powertrain needs to comply in order to be used by the tool. A more detailed quasi-static model is described in which the fuel consumption depends also on the ICE coolant temperature. The quasi-static model will be used in an optimization criterion where frequent gear and ICE on/off switches are penalized.

4.1 Dynamic vehicle model

The dynamic vehicle model, that follows the same state space description as in (1), includes a combined powertrain, configured as a mild parallel hybrid powertrain propelling the front wheels, in addition to an electric machine propelling the rear wheels. The electric machine on the rear axle, EM2, is supplied from the same energy buffer as the electric machine on the front axle, EM1, see Figure 6. The transmission includes a gearbox and a torque converter, see Heisler (2002) for background.

The states that are to be kept in the simplified model and the signals that need to be accessed by the tool are

$$\begin{aligned} x_{ck}(t) &= [soc(t) T(t) \omega_s(t)]^T \\ x_{dk}(t) &= [\gamma(t) x_{ICE}(t) x_{EM_1}(t) x_{EM_2}(t) x_{brk}(t)]^T \\ u_{ck}(t) &= [F_{load}(t) \tau_{load}(t)]^T \\ u_{dk}(t) &= [u_\gamma(t) u_{ICE}(t) u_{EM_1}(t) u_{EM_2}(t) u_{brk}(t)]^T \\ y_k(t) &= [v_w(t) F_w(t) \dot{m}_f(t) \tau_s(t)]^T \end{aligned} \quad (21)$$

Figure 6 Combined powertrain model.

where $T(t)$ is the ICE coolant temperature, $\omega_s(t)$, $\tau_s(t)$ and $\tau_{load}(t)$ are the shaft speed, torque and torque disturbance respectively between the ICE and the clutch, $x_{EM_1}(t)$ and $x_{EM_2}(t)$ are the on/off states of the EM1 and EM2 and $u_{EM_1}(t)$ and $u_{EM_2}(t)$ are the input events (0 or 1) that control these states. The rest of the signals are as in (3).

The requirements on the model are the same as in Section 3.1, including the possibility to externally add the torque load $\tau_{load}(t)$, see Figure 6. The necessity of this signal, compared to the parallel powertrain where only the force $F_{load}(t)$ acting at the wheels was used to simulate a longitudinal load, is in the additional possibility of the combined powertrain to operate in series mode, when the clutch is open.

4.2 Quasi-static powertrain model

The simplified, quasi-static model is a backward simulation model that satisfies the power balance equation

$$\begin{aligned} (\tilde{\tau}_{ICE}(t) + \tilde{\tau}_{EM_1}(t))\tilde{\omega}_s(t) &= \tilde{\tau}_s(t)\tilde{\omega}_s(t) \\ (\tilde{F}_{EM_2}(t) + \tilde{F}_{trn}(t) + \tilde{F}_{brk}(t))v_w(t) &= F_w(t)v_w(t) \end{aligned} \quad (22)$$

where $\tilde{\omega}_s(t)$ resembles $\omega_s(t)$ in (21), $\tilde{\tau}_{ICE}(t)$ is the torque of the ICE, $\tilde{\tau}_{EM_1}(t)$ is the torque of the EM1, $\tilde{F}_{EM_2}(t)$ is the force of the EM2 and $\tilde{F}_{trn}(t)$ is the force at the transmission output.

In accordance to the optimization criterion described later in Section 4.3, the states and inputs in the quasi-static model are

$$\begin{aligned} \tilde{x}(t) &= [\tilde{\gamma}(t) \tilde{x}_{ICE}(t) \tilde{s}o\tilde{c}(t) \tilde{T}(t)]^T \\ \tilde{u}(t) &= [\tilde{u}_\gamma(t) \tilde{u}_{ICE}(t) \tilde{\omega}_s(t) \tilde{\tau}_{ICE}(t) \tilde{\tau}_{EM_1}(t) \tilde{F}_{EM_2}(t) \tilde{F}_{brk}(t)]^T \end{aligned} \quad (23)$$

where $\tilde{\gamma}(t)$, $\tilde{x}_{ICE}(t)$, $\tilde{s}o\tilde{c}(t)$, $\tilde{T}(t)$ and $\tilde{u}_\gamma(t)$ resemble the respective states and inputs of the dynamic model (21). The shaft speed $\tilde{\omega}_s(t)$ which resembles the state $\omega_s(t)$ in the dynamic model is rather used as an input to the quasi-static model. This is because 1) in the optimization criterion the ICE and EM1 speed dynamics are neglected and there is no need of including $\tilde{\omega}_s(t)$ as a state and 2)

the computational time of the optimization algorithm increases exponentially with the number of states. This signal, however, cannot be completely removed from the quasi-static model since it is needed as an input to some of the lookup tables, see (24), and thus it is included in $\tilde{u}(t)$.

The quasi-static model comprises four lookup tables from which f_{ICE} , f_{EM_1} and f_{EM_2} describe the three cases where the dynamic model is powered only by the ICE, the EM1 or the EM2, and efficiency map of the transmission, f_{trn} , obtained when both the ICE and EM1 power the vehicle. It is expressed as

$$\begin{aligned} \begin{bmatrix} \tilde{m}_f(t) \tilde{T}(t) \tilde{s}\tilde{c}_a(t) \\ \tilde{s}\tilde{c}_{EM_1}(t) \\ \tilde{s}\tilde{c}_{EM_2}(t) \\ \tilde{\omega}_s(t) \tilde{\tau}_s(t) \end{bmatrix}^T &= f_{ICE}(\tilde{\omega}_s(t), \tilde{\tau}_{ICE}(t), \tilde{T}(t)) \\ \tilde{s}\tilde{c}_{EM_1}(t) &= f_{EM_1}(\tilde{\omega}_s(t), \tilde{\tau}_{EM_1}(t), \tilde{s}\tilde{c}(t)) \\ \tilde{s}\tilde{c}_{EM_2}(t) &= f_{EM_2}(v_w(t), \tilde{F}_{EM_2}(t), \tilde{s}\tilde{c}(t)) \\ \begin{bmatrix} \tilde{\omega}_s(t) \tilde{\tau}_s(t) \end{bmatrix}^T &= f_{trn}(\tilde{\gamma}(t), v_w(t), \tilde{F}_{trn}(t)) \end{aligned} \quad (24)$$

where $\tilde{s}\tilde{c}_a(t)$, $\tilde{s}\tilde{c}_{EM_1}(t)$ and $\tilde{s}\tilde{c}_{EM_2}(t)$ are derivatives of the SOC that correspond to the power used by the auxiliary devices, EM1 and EM2 respectively and satisfy

$$\tilde{s}\tilde{c}(t) = \tilde{s}\tilde{c}_a(t) + \tilde{s}\tilde{c}_{EM_1}(t) + \tilde{s}\tilde{c}_{EM_2}(t). \quad (25)$$

The lookup tables are generated as in (7).

4.3 Optimization criterion

The optimization criterion includes minimization of fuel consumption plus additional terms that shape the states trajectories. It is given as

$$\tilde{u}^*(t) = \arg \min_{\tilde{u}(t)} \int_{t_0}^{t_f} \Psi(t) dt$$

$$\Psi(t) = \tilde{m}_f^q(t) + c_1 |\tilde{x}_{ICE}(t_{k+1}) - \tilde{x}_{ICE}(t_k)| + c_2 \min\{|\tilde{\gamma}(t_{k+1}) - \tilde{\gamma}(t_k)|, 1\}$$

subject to (19c-19f), (22), (24), (25) and

$$\tilde{\tau}_{ICE}(t) \in [\tilde{\tau}_{ICE,min}(t), \tilde{\tau}_{ICE,max}(t)] \quad (26a)$$

$$\tilde{\tau}_{EM_1}(t) \in [\tilde{\tau}_{EM_1,min}(t), \tilde{\tau}_{EM_1,max}(t)] \quad (26b)$$

$$\tilde{F}_{EM_2}(t) \in [\tilde{F}_{EM_2,min}(t), \tilde{F}_{EM_2,max}(t)] \quad (26c)$$

$$\tilde{F}_{trn}(t) \in [\tilde{F}_{trn,min}(t), \tilde{F}_{trn,max}(t)] \quad (26d)$$

$$\tilde{\omega}_s(t) \in [\omega_{s,min}, \omega_{s,max}] \quad (26e)$$

$$\tilde{T}(t) \in [T_{min}, T_{max}] \quad (26f)$$

$$\tilde{\gamma}(t) \in \{0, 1, \dots, \gamma_{max}\}$$

$$\tilde{u}_\gamma(t) \in \{-\gamma_{max}, -\gamma_{max} + 1, \dots, 2\} \quad (26g)$$

$$\tilde{x}_{ICE}(t) \in \{0, 1\}$$

$$\tilde{u}_{ICE}(t) \in \{0, 1\}$$

$$\tilde{x}_{ICE}(t_0) = 0$$

$$\tilde{\gamma}(t_0) = 0$$

where the instantaneous cost $\Psi(t)$ includes the fuel-flow and penalties for frequent ICE on/off switching and frequent gear shifting. The penalty coefficients c_1 , c_2 are user adjustable. The physical limits of the powertrain components in (26a-26f) and the highest gear γ_{max} are found automatically by the tool. The ambient temperature T_{min} can be given by the user and (26g) prevents upshifts of more than 2 gears. The vehicle starts the simulation in neutral gear and with the ICE turned off.

The optimization process is carried on as in (20), where the optimal cost matrix J^* is a five dimensional matrix, four states and time, and its computation may require a significantly long time.

5 Custom optimization criteria and user interface aspects

There is a possibility to extend the tool library with custom optimization criteria, e.g. minimization of pollutants and sound emissions. This can be done by assigning the signals that will be used in the criterion and to which the dynamic model provides access, as in (3) and (21). Given the transparency of the dynamic model, it is up to the user to decide which vehicle dynamics will be kept in the simplified model. If for example NOx emissions are to be minimized, the states describing the ICE transients should be kept. This may require relaxation of the power balance constraints (4) and (22) in order of avoiding infeasible solutions.

Except a custom optimization criterion, an experienced user may also provide:

- Desired vehicle performance by deciding the driving cycle $r(t)$.
- A custom set of gridded values needed in the simplification process, for $r(t)$, $u_{ck}(t)$, $u_{dk}(t)$ and $\nu_{ck}(t)$, at which the dynamic model is simulated, see Section 3.2.
- The tolerance ϵ , the windows size N_w and the level of significance θ_j of the signals used for detecting equilibrium, see Section 3.5.
- The required fit fit_{lim} for deciding the accuracy of the simplified model, see Section 3.7.
- The penalty coefficients, the initial and final state values and other constraints in an existing optimization criterion.
- A custom discrete sets \mathcal{X} , \mathcal{U} and \mathcal{T} at which the optimal cost-to-go matrix J^* is to be computed, see Section 3.9.

6 Example 1: Evaluation of a parallel powertrain

This section demonstrates the tool operation on the problem of minimizing fuel consumption. The main focus is on the simplification process, while in the following example in Section 7 the focus is shifted to the optimization process.

Two non-transparent dynamic models are considered, both modeled in Matlab/Simulink, and both using a fixed step-size solver. The first model is a

conventional passenger vehicle and the second model is a parallel HEV constructed by adding a battery and EM to the first model. The problem of this study is to investigate how big improvement in fuel consumption the hybridization may give, without considering the economic cost for the additional components. Both models fulfill the requirements listed in Section 3.1.

The chosen driving cycle is the "New European Driving Cycle" (NEDC) with constant road altitude, see Figure 7, and the optimization criterion is as in (19).

6.1 Results

The first step in the tool operation is the model simplification. The quasi-static model (6) is generated by simulating the dynamic model over the default set of gridded values given by the tool. Assuming that the maximum gear, velocity and traction force the dynamic model provides are known, i.e. $\gamma_{max} = 5$, $\max(v_r(t)) = 180$ km/h and $\max(F_{load}(t)) = 2.5$ kN, then for the generation of f_{EM} there would be more than 220000 grid points. The tool's automatic detection of the operating region found that only 53920 points are operational. More precisely, there are 54180 points, including the number of performed tests for obtaining the velocity and torque boundaries, for which the dynamic model is simulated.

With the steady-state detector in use, the total simulation time for the generation of f_{EM} was about 35 minutes (by running Simulink in *Normal* mode; simulating in *Accelerator* mode would require smaller simulation time). 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 7 times longer.

After the quasi-static model is obtained, it is validated as was described in Section 3.7 and the results are given in Figure 7. The 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.

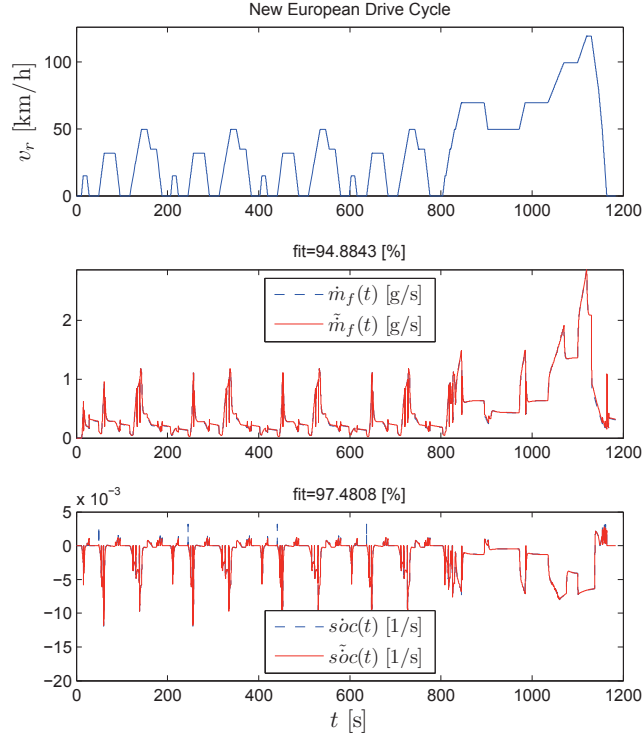
The second step after the simplification is optimization. The gear sequence $\gamma(t)$, the true vehicle velocity $v_w(t)$ and the traction force $F_w(t)$ used as inputs in the simplified model are obtained from the dynamic model simulated on the same driving cycle, see Figure 3. The allowed battery SOC deviation is 20% placed symmetrically around the initial charge of 50%. Charge sustain operation is preserved.

The fuel consumption of the parallel HEV obtained by the optimal control law is approximately 6.1l/100km which is about 18% improvement compared with the conventional vehicle. However, the quasi-static model is oversimplified and lower fuel improvement is to be expected if, for example, the cold ICE start is considered and the frequent ICE on/off and gear switches are penalized. Such scenario is investigated in the following section.

7 Example 2: Evaluation of a combined powertrain

This section investigates the improvement in fuel economy by upgrading a mild parallel to a combined powertrain. Two non-transparent models of passenger

Figure 7 Validation of the quasi-static powertrain model with the f_{ICE} output in the middle and the f_{EM} output in the bottom plot. The quasi-static model gives good fit in stationary conditions, but also in transients and being on top it covers the output of the dynamic model almost completely.



vehicles are given, both modeled in Matlab/Simulink, and both using a variable step-size solver. The first model includes a mild parallel powertrain propelling the front axel, while the second model has a combined powertrain, as described in Section 4.1, constructed by adding an additional electric machine connected to the rear axel. The goal is to investigate the potential benefit in fuel economy the full hybridization can give, without considering the economic cost of the additional components. The models fulfill the requirements listed in Section 3.1.

7.1 Results: Optimal state trajectories

The simplified quasi-static model, described as in (24), is automatically generated by the tool and is used to optimize the criterion (26) over the NEDC. The initial and the final SOC values are set to 50% charge and the allowed SOC interval is within 40% and 60%. The ambient temperature is 30 °C.

The resulting optimal state trajectories are given in Figure 8, where in the left column results are given of the combined HEV powertrain and in the right column results are given of the mild parallel HEV powertrain. These results are further grouped in three rows as follows, Figure 9(a) and Figure 9(b) show the optimal state trajectories for a cold ICE start and penalized frequent gear and ICE

on/off switching, Figure 9(c) and Figure 9(d) show the optimal state trajectories for a hot ICE start and also penalized frequent gear and ICE on/off switching and Figure 9(e) and Figure 9(f) show the optimal state trajectories for a cold ICE start but with no penalty for frequent gear and ICE on/off switching.

The results show that frequent gear shifts and the ICE on/off switching are prevented when penalized in the cost function, see Figure 9(a)-9(d). This causes oscillatory behavior in the battery state, within the allowed interval, since the optimal control tries to keep the ICE in the same state, off or on, for longer time periods. When the ICE is off the vehicle is driven by the EM1 and/or EM2 and the battery is discharged, until the ICE is turned back on to recharge the battery. This is especially evident for the combined powertrain. Without the switching penalties the battery state is steadier. In this case the battery is discharged only at the end of the driving cycle, at about 1000 s, because the big deceleration that follows soon after, see the NEDC in Figure 7, is used to recharge the battery.

The optimal control prefers shifting two gears up when frequent gear shifts are not desired. This happens because the penalty for fast gear shifting in the cost function can be understood as the energy wasted in the gear shifting transient. Hence, two gears up-shift (environment friendly driving) will increase the overall system efficiency, since the amount of wasted energy is assumed equivalent to the amount of energy wasted in only one shifting transient.

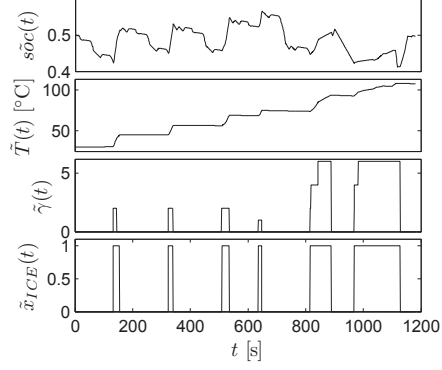
The mild parallel powertrain reaches faster the high ICE coolant temperatures. Although the ICE has better efficiency on higher temperatures, the amount of consumed fuel for reaching this temperature is high, which makes this powertrain less efficient than the combined powertrain, see Table 1. Both powertrains consume less fuel when the ICE is hot started with 100 °C, see Table 1, but difference in the optimal control for the rest of the states is not evident.

The combined powertrain almost never uses EM1 alone to power the vehicle (look for closed clutch when the ICE is off in Figure 8). This is due to the clutch placement, see Figure 6, since EM1 will also need to rotate the ICE in order to drive the vehicle, which causes energy losses. The better solution is to use the EM2 instead.

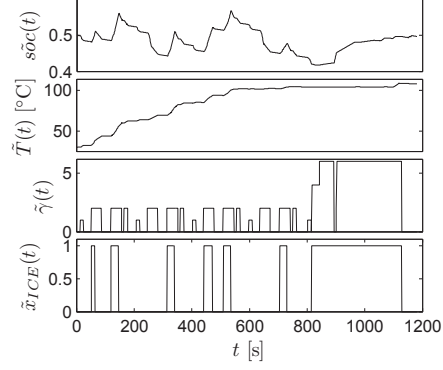
The improvement in fuel economy of the combined powertrain, compared to the mild parallel powertrain, is as high as 10 %, see Table 1.

Table 1 Normalized fuel consumption of the combined and the mild parallel HEV powertrain model.

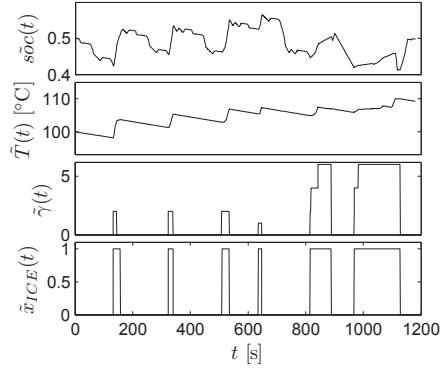
	Cold start	Hot start	cold start and frequent switches
Mild parallel	1	0.99	0.92
Combined	0.91	0.89	0.86
Improvement [%]	9.25	10.12	6.31

Figure 8 Optimal state trajectories for the combined and the mild parallel HEV.

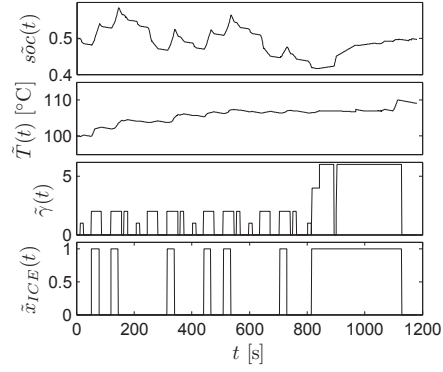
(a) Combined HEV powertrain with cold ICE start and penalized frequent gear and ICE on/off switching



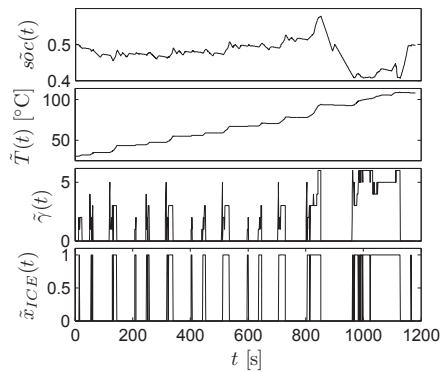
(b) Mild parallel HEV powertrain with cold ICE start and penalized frequent gear and ICE on/off switching



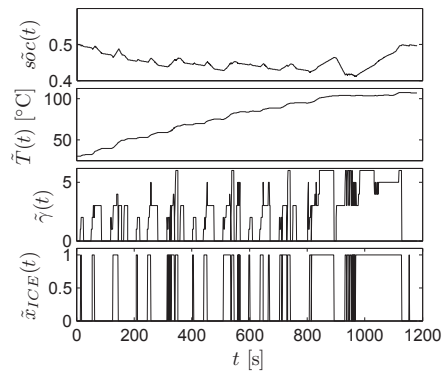
(c) Combined HEV powertrain with hot ICE start and penalized frequent gear and ICE on/off switching



(d) Mild parallel HEV powertrain with hot ICE start and penalized frequent gear and ICE on/off switching

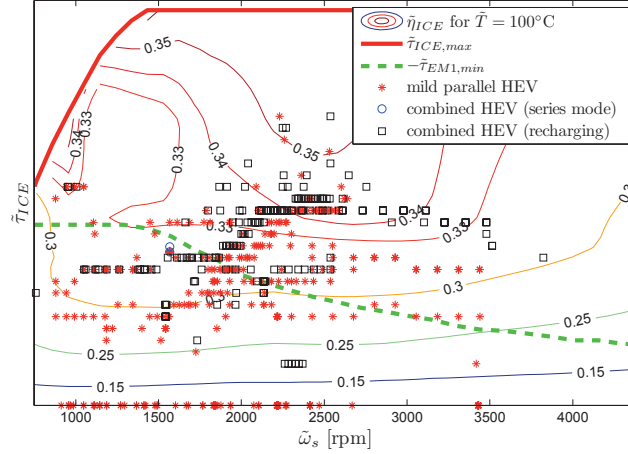


(e) Combined HEV powertrain with cold ICE start and no penalty for frequent gear and ICE on/off switching

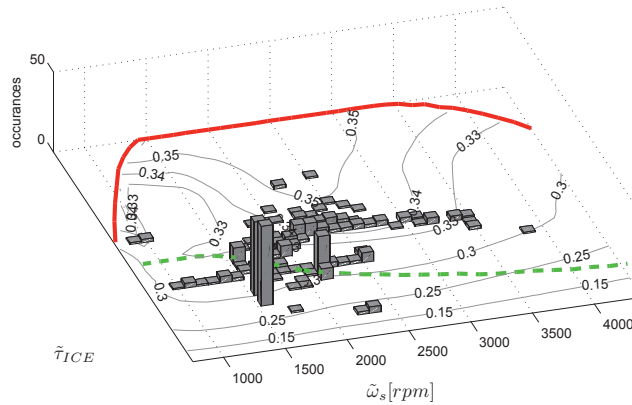


(f) Mild parallel HEV powertrain with cold ICE start and no penalty for frequent gear and ICE on/off switching

Figure 9 Operating points of the ICE for the case of cold ICE start and no frequent gear and ICE on/off switching. The contour plot shows the ICE efficiency, $\tilde{\eta}_{ICE}$, for coolant temperature of $\tilde{T} = 100^\circ\text{C}$. The bottom plot shows the number of occurrences for the operating points of the combined powertrain.



(a) ICE efficiency and operating points.



(b) Histogram of the ICE operating points.

7.2 Results: Optimal operating points

The optimal ICE operating points for the case of cold ICE start and penalized frequent gear and ICE on/off switching are given in Figure 9. It can be noticed that the combined powertrain operates either in series mode (the clutch is opened and EM1 is used as a generator), or recharging mode (the ICE provides more power than needed and the surplus is used to recharge the battery). Series mode is a reasonable choice when the battery has low energy level. Typical example would be charging the battery at stand-still or cruising. As can be seen in Figure 9 this mode occurs only at one specific operating point, which is the point of optimal efficiency of the combined engine-generator unit.

The vehicle operates in recharging mode for most of its trip. This is not surprising, since running the ICE with higher load at some instance and keeping it off at other, increases the system efficiency and therefore decreases fuel consumption. It is clearly visible in Figure 9 that the combined powertrain never runs the ICE with zero torque and non-zero speed. This confirms what was previously concluded from Figure 8 that there is no incentive in driving with EM1 alone, when EM2 is also present in the vehicle. As a comparison, the mild parallel powertrain has no other choice to empty the accumulated battery power, but to rotate the ICE as well.

The combined powertrain is never operated in boosting mode (the EM1 and/or EM2 assist the ICE in delivering the power demand). This is because the ICE is obviously designed to handle much higher power requests and could drive the cycle even without the battery. This raises curiosity of evaluating alternative powertrain in which the ICE is downsized by at least one quarter of its maximum power. This may slightly sacrifice the vehicle performance, such as safety and driveability, but it may significantly improve the fuel economy since 1) the ICE will be operated mostly with high torque, i.e. in high efficiency regions and 2) the ICE will be lighter and the vehicle will need less fuel to finish the same driving cycle.

There are also no operating points where ICE alone operates the vehicle. This is because the ICE efficiency, in general, increases with the torque, see Figure 9. Hence, it is beneficial to either always lift the operating points (this is what the optimal control does in recharging mode), or to completely turn off the ICE.

8 Conclusion

This paper described a methodology and a tool for fast and simple assessment of powertrain performance for vehicle models that may not be transparent. The assessment is useful for exploring the potential of different powertrain designs and alternatives at an early stage of the development process.

The tool first simplifies the model into quasi-static relations, after which dynamic programming is used to optimize the criterion chosen by the user. The model details can be hidden from the user as long as the model satisfies the set of requirements stated in Section 3.1.

The methodology of model simplification and powertrain evaluation was presented through two examples of power management optimization. The first example showed that upgrading the conventional powertrain to a parallel HEV powertrain may improve fuel economy up to 18%. The second example showed that the full hybridization of the mild parallel powertrain to a combined powertrain may decrease fuel consumption by additional 10%. But most significantly, this study showed that the process of powertrain evaluation can be automatized such that user insight in vehicle modeling, simulation and optimization is not necessary.

References

- R. Bellman. *Dynamic Programming*. Princeton Univ Pr, New Jersey, June 1957.
- D. P. Bertsekas. *Dynamic Programming and Optimal Control*. Athena Scientific, Belmont, Massachusetts, November 15 2000.

- P. R. Brown and R. R. Rhinehart. Demonstration of a method for automated steady-state identification in multivariable systems. *Hydrocarbon Processing*, 79(9):79–83, 2000.
- J. Bumby, P. Clarke, and I. Forster. Computer modelling of the automotive energy requirements for internal combustion engine and battery electric-powered vehicles. *IEE proceedings*, 132(5), 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.
- S. Cao and R. R. Rhinehart. An efficient method for on-line identification of steady state. *J. Proc. Cont.*, 5(6):363–374, 1995.
- G. H. Cole. SIMPLEV: A simple electric vehicle simulation program version 1.0. EG and G Idaho, Inc., Idaho Falls, United States, June 1991.
- E. L. Crow, F. A. Davis, and M. W. Maxfield. *Statistics Manual*. Dover Publications, Mineola, N.Y. 11501, 1960.
- F. Flehmig, R. v. Watzdorf, and W. Marquardt. Identification of trends in process measurements using the wavelet transform. *Computers chem. Engng*, 22(Supplement 1):491–496, 1998.
- 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. Guzzella and A. Amstutz. CAE tools for quasi-static modeling and optimization of hybrid powertrains. *IEEE Transactions on Vehicular Technology*, 48(6), November 1999.
- L. Guzzella and A. Sciarretta. *Vehicle propulsion systems, introduction to modeling and optimization*. Springer, Berlin, Heidelberg, 2nd edition, 2007.
- O. Hayat, M. Lebrun, and E. Domingues. Powertrain drivability evaluation: Analysis and simplification of dynamic models. In *SAE World Congress*, Detroit, Michigan, March 2003.
- H. Heisler. *Advanced Vehicle Technology*. A Butterworth-Heinemann Title, Linacre House, Jordan Hill, Oxford, 2nd edition, July 2002.
- H. Hotelling. The generalization of student's ratio. *Ann. Math. Statist.*, 2(3):360–378, 1931.
- L. Johannesson. *Predictive Control of 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. 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.
- 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(2):255–266, 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.
- D. C. Montgomery and G. C. Runger. *Applied Statistics and Probability for Engineers*. John Wiley and Sons, Inc., USA, 2003.
- 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*, 32(9):1409–1418, September 1986.
- R. R. Rhinehart. A novel method for automated identification of steady state. In *Proceedings of the American Control Conference*, Seattle, Washington, June 1995.
- G. Shafer. *A mathematical theory of evidence*. Princeton University Press, 1976.
- 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.
- R. Trigui, B. Jeanneret, and F. Badin. System modelling of hybrid vehicles in order to predict their energy and dynamic performance: Building VEHLIB library. *Recherche Transports Sécurité*, 83(2):129–150, 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.
- H. Witsenhausen. A class of hybrid-state continuous-time dynamic systems. *IEEE Transactions on Automatic Control*, 11(2):161–167, April 1996.