

Online Implementation of an Optimal Supervisory Control for a Parallel Hybrid Powertrain

Alexandre Chasse, Philippe Pognant-Gros and Antonio Sciarretta
IFP

Copyright © 2009 SAE International

Abstract

The authors present the supervisory control of a parallel hybrid powertrain, focusing on several issues related to the real-time implementation of optimal control based techniques, such as the Equivalent Consumption Minimization Strategies (ECMS). Real-time implementation is introduced as an intermediate step of a complete chain of tools aimed at investigating the supervisory control problem. These tools comprise an offline optimizer based on Pontryagin Minimum Principle (PMP), a two-layer real-time control structure, and a modular engine-in-the-loop test bench. Control results are presented for a regulatory drive cycle with the aim of illustrating the benefits of optimal control in terms of fuel economy, the role of the optimization constraints dictated by drivability requirements, and the effectiveness of the feedback rule proposed for the adaptation of the equivalence factor (Lagrange multiplier).

INTRODUCTION

Nowadays hybrid propulsion systems are increasingly recognized as one of the few possibilities of combining low carbon dioxide emissions, acceptable range, and good performance in road vehicles. In spite of their complexity with respect to conventional powertrains, hybrids offer additional degrees of freedom that can be optimized.

Optimization of hybrid energy management (supervisory control) ensures that the hybrid operation along, e.g., a drive cycle is optimal with respect to some dynamic criterion, typically related to energy consumption and subjected to several constraints. This optimization is increasingly based on system modeling in contrast to heuristic strategies dictated by experience only [1, 2].

Offline optimization based on techniques such as Pontryagin's Minimum Principle (PMP) [3, 4, 5] or Dynamic Programming [6, 7, 10, 8, 9] allow calculating the optimal

power flows trajectories of a given powertrain on a given driving schedule. As a function of the speed and torque desired at the wheels the supervisory control calculates the set-points to the controlled components (engine, electric machines, transmission, etc.), i.e., their torque set-points typically. The results of such offline techniques are very useful, e.g., to assess the potential performance of a concept under testing. However, the online implementation of the same algorithms poses several issues [11, 12, 13].

First of all, the power demand at the wheels is not known from a given driving schedule but it has to be interpreted from the driver's action (acceleration and brake pedals). Secondly, the controlled machines do not respond immediately to the set-point imposed, but with a certain dynamics dictated by the machine self-response and the behavior of the respective low-level controllers. Moreover, the set-points to the various components have to be coordinated to each other.

Apart from the points above, the difference between the offline-optimized trajectories and their respective online counterparts can be different due to many other reasons. Several constraints due to drivability and driver acceptance are often neglected in online optimization to force the calculation to yield the ideal optimum. However, these constraints must be taken into account in the online control. Also, the typical optimum search procedures dictated by the PMP require some care to be implemented in a causal modeling platform such as Simulink. Finally, the knowledge of the entire driving schedule that is assumed for offline optimization is not true for online optimization. Instead, the influence of the driving conditions is lumped in one control parameter, playing the role of a Lagrange multiplier in the equations of optimal control, which has to be estimated in real time.

All these issues are discussed in the paper, which is organized as follows. A first section describes the offline optimization tool based on the PMP. The transition between this technique and the online supervisory controller is the object of the second section. Then, a third section describes the modular engine-in-the-loop test bench that has been used for the investigation. Finally, control results are presented and discussed.

OFFLINE DYNAMIC OPTIMIZATION

The dynamic optimization of a hybrid powertrain over a given drive cycle consists of calculating the trajectories of power factors for the various components in order to minimize a certain global criterion under certain local and global constraints. In the formulation of this paper, the global criterion is the overall fuel consumption and the global constraint is on the final state of charge of the battery, which has to match the value at the beginning of the cycle. Mathematically speaking, find a control policy $u(t)$ such to

$$\min \int_0^T P_f(u, t) dt \quad (1)$$

submitted to

$$\dot{\xi} = -I_b(u, t, \xi)/Q_0, \quad \xi(T) = \xi(0) \quad (2)$$

where T is the duration of the cycle, $P_f = \dot{m}_f H_f$ is the power consumed with the fuel, $\dot{m}_f$ is the fuel consumption rate, H_f is the fuel lower heating value, ξ is the battery state of charge (SOC), Q_0 is the nominal charge capacity of the battery, I_b is the battery current (positive if discharging). The fuel consumption depends on the vehicle speed (and consequently torque demand at wheels) imposed by the drive cycle at time t , as well as on the control u . The SOC variation depends also on the SOC itself, at least in principle. This problem is solved using PMP in the following way:

$$u^o(t) = \operatorname{argmin} H(u, t, \xi) \quad (3)$$

where the Hamiltonian function is defined as

$$H(u, t, \xi) = P_f(u, t) + s^o(t) P_{ech}(u, t, \xi) \quad (4)$$

In the latter equation, the superscript ^{o} denotes optimal trajectories, P_{ech} is the inner (electrochemical) battery power, given by the product of the battery current and the open-circuit voltage, at least in a simple electric-circuit approximation of the battery dynamics. The equivalence factor s that weights the electrochemical power before summing it to the fuel power results from the costate adjoint to ξ in the Hamilton-Jacobi formulation of Optimal Control [14]. It can be proven that, if the dependency of H on ξ is negligible, s^o is a constant during the drive cycle.

The components of the control vector u are the degrees of freedom available at each time to satisfy the torque imposed by the drive cycle $T_{wh}(t)$. The nature of these degrees of freedom depends on the particular hybrid architecture. For example, in a parallel hybrid the control variable u is one between the torque coming from the electric

machine and that coming from the engine, their sum being $T_{wh}(t)$. Other hybrid architectures have different control vectors.

The offline Hybrid Optimization Tool (HOT) presented in this section deals with generic hybrid architectures. Its structure is based on PMP and on the aforementioned approximation of a constant optimal equivalent factor s^o . This particular value is sought using a shooting algorithm according to the sign of the deviation $\xi(T) - \xi(0)$, as shown by the pseudocode :

```

initial guess for s
begin
for each time t=1:1:T
    drive cycle at time t
    for each control u
        invoke system model(t,u,SOC)
        calculate tentative H (Eq. 4)
    end for
    find optimal u(t) (Eq.3)
end for
evaluate SOC deviation
adjust s, goto begin

```

Several features of HOT emerge from the following pseudocode:

- The shooting algorithm is based on the fact that a too high estimation of s^o tends to overpenalize the use of the battery, so that the final SOC will be higher than the target, while a too low estimation tends to overfavor the battery use, so that the final SOC will be lower than the target.
- The tentative values of u are necessarily discretized. HOT selects the appropriate degrees of freedom represented by u by calculating the rank of a kinematic matrix that links the speed and torque values of each component with those imposed by the drive cycle. If a certain variable (i.e., the engine torque) is a part of u , a finite number of tentative values is selected while respecting the corresponding physical limits.
- The system model necessary to compute the tentative Hamiltonian functions is a collection of quasi-static backward models [1] for the various components, such as engine fuel map, motor efficiency map, battery equivalent circuit, etc. HOT uses a generic kinematic matrix and consequently a generic model structure that is able to represent several hybrid architectures (Fig. 1). The generic model structure includes an engine E, two electric machines M and G, a clutch U, a mechanical node N2. The element N1 can be either a node (where the concurrent speed values are equal and the concurrent torque values sum up) or its dual, a splitter (speeds sum up, torques are equal). Common hybrid architectures are obtained by selecting the four Booleans i_n (N1 is a

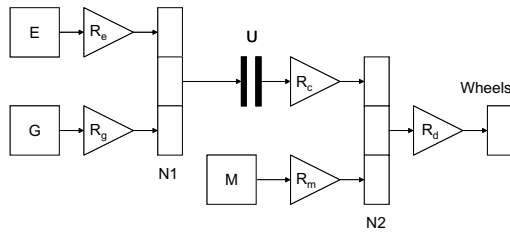


Figure 1: HOT generic hybrid structure.

Table 1: Hot parameterization to obtain common power-train structures.

Architecture	i_n	i_m	i_g	i_u
Conventional	0	0	0	1
Parallel	0	1	0	1
Stop-start	0	0	1	1
Series	0	1	1	0
Power-split	1	1	1	1

splitter rather than a node), i_m (machine M is physically present), i_g (machine G is physically present), i_u (clutch U is physically disconnected) as shown in Table 1. The transmission ratios shown in Fig. 1. can be fixed or time varying. In the latter case, they can be prescribed or optimized alongside the components of the vector u .

One of the major advantages of PMP-based optimization over other techniques is that the core of the algorithm (the inner for loop in the previous pseudocode) is directly implementable in a real-time controller, even if the drive cycle is not known a priori. The controllers based on the minimization of an Hamiltonian function are often denoted as Equivalent Consumption Minimization Strategies (ECMS) [16, 17].

ONLINE OPTIMAL CONTROL

This section presents the implementation of an ECMS-based supervisory controller. Considering the inner loop in the pseudocode presented in the last section, what is missing in real time applications with respect to offline optimization is essentially:

- the prior knowledge of the drive cycle (torque demand), and
- the possibility of over-shooting for the value of s^o to find the optimal value.

The former issue is treated in the supervisory controller (Fig. 2) in a similar way than in torque-controlled internal combustion engines (ICE), that is, with an *interpreter* block that estimates the driver's torque demand from the position of the acceleration and brake pedals. The output of such a block is the estimated powertrain torque

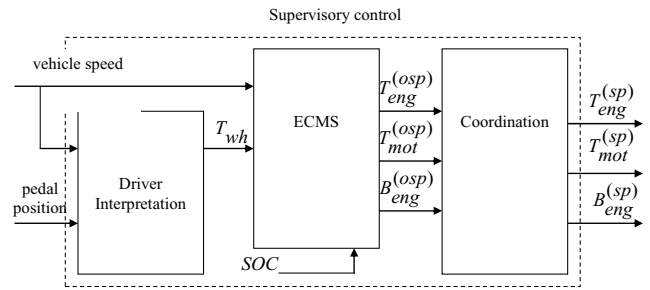


Figure 2: Outer layer of the supervisory controller.

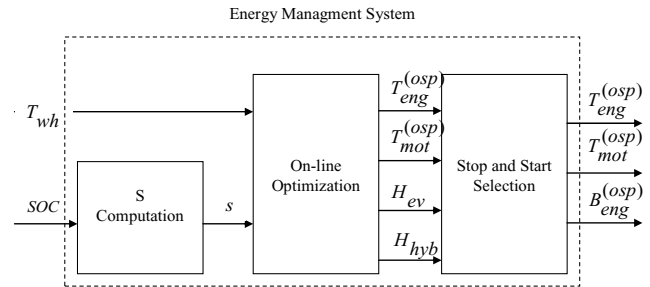


Figure 3: ECMS layer of the supervisory controller.

T_{wh} . With respect to ICE-based torque control, in hybrid systems T_{wh} could be also negative in principle, allowing for regenerative braking. However, regenerative braking needs an ad hoc controller that has full access on the braking system in order to split the resistive torque between the hydraulic brakes and the electric powertrain. For the sake of simplicity, regenerative braking is not taken into consideration in this paper. Accordingly, the interpreter block provides an only slightly negative T_{wh} when the brake pedal is depressed or when the acceleration pedal is released.

The driver interpreter is the first out of three sub-blocks constituting the supervisory controller (Fig. 2). The intermediate function is the ECMS, whose outputs are the various commands needed by the system components. In the following, the case of a simple parallel hybrid will be assumed, for which the ECMS outputs the torque commands to the engine and the motor, $T_{eng}^{(osp)}$ and $T_{mot}^{(osp)}$, the gear command to the automatic gearbox, $n_{gb}^{(osp)}$, as well as a Boolean engine stop/start command $B_{eng}^{(osp)}$. The third block, or *coordination layer*, refines these setpoints dealing with particular dynamic operations, finally yielding corrected values $T_{eng}^{(sp)}$, $T_{mot}^{(sp)}$, $n_{gb}^{(sp)}$, and $B_{eng}^{(sp)}$.

Focusing on the ECMS block, again a three-step structure has been used (Fig. 3). The upstream block is responsible for estimating the equivalent factor s that is necessary to the ECMS core, as outlined above. Several strategies can be adopted for such a task. Indeed, the estimation of s is the most critical aspect of an ECMS implementation and a key factor for obtaining a good performance from the supervisory controller. The simplest method is to use the constant value of s^o issued from the offline optimization