Parametric optimization of hybrid car engines

J. Frédéric Bonnans (frederic.bonnans@inria.fr)
Projet Sydoco,INRIA, B.P. 105, 78153 Rocquencourt, France

Thérèse Guilbaud (therese.guilbaud@cermics.enpc.fr)
Université Paris VIII and Cermics, ENPC, 6 and 8 avenue Blaise Pascal, Cité Descartes - Champs sur Marne, 77455 Marne la Vallée Cx 2, France.

Ahmed Ketfi-Cherif (ahmed.ketfi-cherif@renault.com) and Dirk von Wissel (dirk.von-wissel@renault.fr)
Direction de la Recherche, Technocentre Renault, 1 avenue du Golf, 78288 Guyancourt, France.

Claudia Sagastizábal (sagastiz@impa.br)
IMPA, Estrada Dona Castorina, 110, Jardim Botânico, 22460-320, Rio de Janeiro RJ, Brazil. On leave from INRIA, France.

Housnaa Zidani (zidani@ensta.fr)
ENSTA, 32 boulevard Victor, 75739 Paris Cx 15, and Projet Sydoco, INRIA, France

December 11, 2002

Abstract. We consider the problem of optimal design of hybrid car engines which combine thermic and electric power. The optimal configuration of the different motors composing the hybrid system involves the choice of certain design parameters. For a given configuration, the goal is to minimize the fuel consumption along a trajectory. This is an optimal control problem with one state variable.

The simultaneous optimization of design parameters and trajectories can be formulated as a bilevel optimization problem. The lower level computes the optimal control for a given architecture. The higher level seeks for the optimal design parameters by solving a nonconvex nonsmooth optimization problem with a bundle method.

Keywords: Hybrid car engines, bilevel optimization, bundle method, optimal control problem, Hamilton-Jacobi-Bellman equation

1. Introduction

Optimal control theory has only recently made a breakthrough in automotive industry. An important reason for this change is the increasing number of electronic devices in modern cars. As a result, there is a big bulk of relevant data that needs to be processed efficiently. For example, the air-fuel mixture for the combustion is automatically controlled. Sophisticated methods of automatic control are currently used in a variety of applications. However, most of such problems typically model very short times and do not require optimal control techniques.

In this paper we address a different type of application, with horizon times going up to a fraction of an hour. Since time intervals are longer, various nonlinear phenomena occur and optimal control methods are a must. Specifically, we consider the optimal design of the transmission system of a car combining thermical and electrical energy. Different design parameters lead to a different configuration or *architecture*, determining the connexions of the different motors and batteries. For a given architecture, the goal is to minimize the fuel consumption along a dynamic trajectory given by an optimal control problem with one state variable. The simultaneous optimization of design parameters and trajectories can be formulated as a bilevel optimization problem. The lower level computes the optimal control for a given architecture by discretizing the associated Hamilton-Jacobi-Bellman (HJB) equations. The higher level seeks for the optimal design parameters by solving a nonconvex nonsmooth optimization problem with a bundle method.

At first sight, the use of optimal control methods may be questionable in such a context, due to the poor knowledge of the future. In our particular problem such argument does not apply, for the following reasons:

- At the design stage of new car engines, it is common to suppose that the future is perfectly known. This assumption allows to evaluate the impact of the new technology, for example, in relation with dimishing fuel consumption. Also, it gives an ideal standard for comparisons when defining real time control laws.

- In some cases, short term future is indeed well known. For example, in *boulevard périphérique*, Paris ring, drivers are given estimated times to reach different exits. These times are predicted with high accuracy, typically with errors below 5%. Another example is given by navigation systems that get weather and traffic news to compute the shortest way to a chosen destination. Although such sophisticated systems are today available only in luxury cars, it is very likely that navigation systems will be widely used in, say, ten years.

There is still a concern about robustness and stability of optimal control methods based on discretization techniques, due to the absence of convergence results. A major exception to this weakness is the discretization of HJB equations in finite horizons, which is precisely the technique employed in our application. In order to guarantee convergence, a stability condition for the discrete scheme must hold. This condition forces the time step to be of the order of space steps. Moreover, it yields convergence after a certain number of operations, of order $O(NN_xN_u)$, where N and N_x are the number of time and space steps, respectively, and N_u is the number of possible values for the control (once discretized). In view of such estimate, the methodology only makes sense if neither N_x nor N_u is too big. When these values are small enough, as in our application, we get a fast and reliable method, that can be used even in a real time environment.

The paper is organized as follows. We start in Section 2 by modeling the problem. We give the main elements of the mechanical configuration and the dynamic evolution of the hybrid system. In Section 3 we consider the resulting bilevel optimization problem and study some relevant properties of Nonsmooth Analysis. We describe the bundle method used to solve the higher level nonsmooth optimization problem in Section 4. In Section 5 we explain how to solve the lower level problem, by discretizing the associated HJB equations. Finally in Section 6 we assess our approach with some preliminary numerical experience.

2. Setting the optimization problem

Suppose the hybrid system is composed by one thermic and two electric engines denoted, respectively, by th, E_1 and E_2. In addition, to increase the efficiency of the system, assume there is a battery that stores unused power, typically at low speeds. Power is produced by the motors, but also there can be some energy recovered from the wheels, through the brakes.

We start by explaining the mechanical configuration of such a system.

2.1. Overview of architecture

The transmission linking the different motors is an planetary gear train, see Fig. 1. It has three principal components, namely, a sun s, a ring r and a planetary carrier pc.

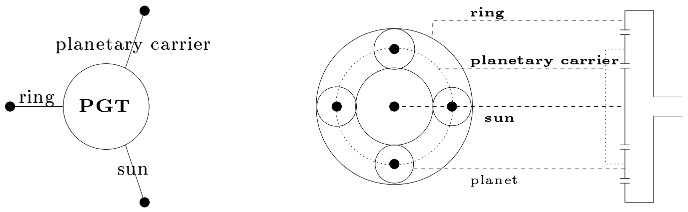


Figure 1. planetary gear train

For $i \in \{s, r, pc\}$, we denote by R_i, ω_i and C_i the corresponding radii, angular speeds and torques, with sign conventions given in Fig.2.

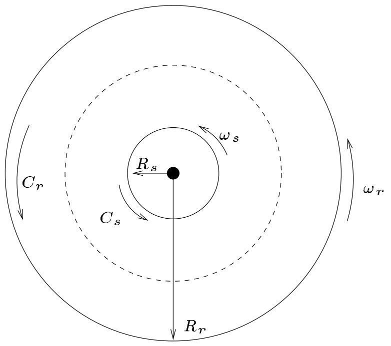


Figure 2. Sign convention for angular speeds and torques

By Willis formula[1] it holds that

$$\frac{\omega_s - \omega_{pc}}{\omega_r - \omega_{pc}} = -\frac{R_r}{R_s} = -\alpha, \tag{1}$$

where $\alpha > 0$ is called the planetary gear train ratio. Two other basic relations, expressing both the conservation of torques and mechanical power are

$$C_r + C_{pc} + C_s = 0, \tag{2}$$

$$C_r\omega_r + C_{pc}\omega_{pc} + C_s\omega_s = 0. \tag{3}$$

Add to (3) the quantity $\pm(C_s + C_r)\omega_{pc}$, and use (2) and (1), to obtain $C_r - \alpha C_s = 0$. Similar algebraic manipulations lead to the identity $C_r(1+\alpha) + \alpha C_{pc} = 0$. Altogether we obtain the relations

$$C_s = \frac{1}{\alpha} C_r; \quad C_{pc} = -\frac{1+\alpha}{\alpha} C_r.$$

As shown in Fig. 3, there are four gear pairs transmitting power from the different engines to the planetary gear train.

Each pair has associated design parameters z_j, for $j \in \{1, 2, th, w\}$, standing for each electrical motor, the thermic motor and the wheels, respectively; the notation refers to the connections to the electrical and thermic motors, and to the wheels. Denote by ω_{th}, ω_1, ω_2, and ω_w the angular speeds at these connections, and by C_{th}, C_1, C_2, and C_w the associated torques (the latter is called transmitted torque). The gear pairs relate input and output data according to the following relations:

$$C_s = z_1 C_1, \quad C_{pc} = z_2 C_2 + z_w C_w, \quad C_r = z_{th} C_{th},$$

[1] This classical formula was already quoted in [18], in its article "Trains épicicloïdaux", on page 1354, that refers to the book "Principles of mechanism" by Willis.

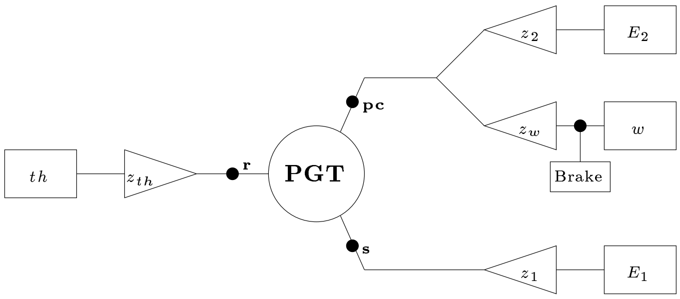


Figure 3. Coupling.

and similarly for the angular speeds. Note that C_w contains the torques of both the wheels and the brakes. Combining these identities with (1)-(3), we obtain that

$$\begin{pmatrix} C_1 \\ C_2 \end{pmatrix} = \begin{pmatrix} \alpha^{-1} z_1/z_{th} & 0 \\ -\alpha(1+\alpha)^{-1} z_2/z_r & -z_2/z_r \end{pmatrix} \begin{pmatrix} C_{th} \\ C_w \end{pmatrix}, \tag{4}$$

$$\begin{pmatrix} \omega_1 \\ \omega_2 \end{pmatrix} = \begin{pmatrix} -\alpha z_{th}/z_1 & (1+\alpha) z_w/z_1 \\ 0 & z_w/z_2 \end{pmatrix} \begin{pmatrix} \omega_{th} \\ \omega_w \end{pmatrix}. \tag{5}$$

In addition,

$$J_{vh}\dot{\omega}_w = C_w + C_{brk} + C_{res}(\omega_w) \tag{6}$$

where J_{vh}, C_w, C_{brk}, and $C_{res}(\omega_w)$ are the the vehicle inertia, transmitted torque, breakin torque, and resistive torque, respectively. The breakin and resistive torque have nonpositive values. The resistive torque is a given function of angular speed. The functions of time $\omega_w(t)$ and $\dot{\omega}_w(t)$ are given. It follows that ω_{th}, C_{th} and C_{brk} can be chosen independently, and determine the values of every angular speed and torque. More precisely, C_{brk} determines C_w through (6), and then (4)-(5) determine the other angular speeds and torques. The variables ω_{th}, C_{th} and C_{brk} will therefore be the control variables of our dynamic system.

2.2. Storage of energy

The role of the storage battery is to accumulate and store unused electrical energy. Therefore, as time passes and the hybrid compound works, the battery level varies and determines the state of the system, denoted by $y(t)$.

The evolution of the battery level depends on its specifications. In particular, at each time t, the electrical power of the battery, P_{bat}, varies with the energy produced by the electrical motors and also with the amount of electrical power used by other devices of the car. The simplest relation is given by the addition of the energy produced by the motors and a constant term, i.e.,

$$P_{bat}(t) = P_1(C_1(t), \omega_1(t)) + P_2(C_2(t), \omega_2(t)) + P_0. \tag{7}$$

We have denoted by $P_0 > 0$, assumed to be constant, the electrical power used by various devices other than the electrical motors, and by $P_i(C_i(t), \omega_i(t))$, $i = 1, 2$, the power sent from the battery to motor i. If we assume that there are no losses of energy in these motors, then

$$P_i(C_i(t), \omega_i(t)) = C_i(t)\,\omega_i(t), \quad i = 1, 2.$$

In the more realistic case when we take into account these losses of power, P_i is larger (resp. less) than $C_i(t)\,\omega_i(t)$ if it is positive (resp. negative). In our model, the electrical power is estimated for a finite number of values of (torque,angular speed), and interpolated by a standard two dimensional splines procedure.

In addition, the battery is specified by its capacity, cap_{bat}, and certain charge/discharge efficiencies, $\eta^\sharp$ and $\eta^\flat$. With this data, the battery level evolves according to the following dynamic equation:

$$\begin{cases} \dot{y}(t) & = \begin{cases} -\dfrac{P_{bat}(t)}{cap_{bat}}\eta^\sharp, & \text{si } P_{bat}(t) \le 0, \\ -\dfrac{P_{bat}(t)}{cap_{bat}}\dfrac{1}{\eta^\flat}, & \text{si } P_{bat}(t) \ge 0, \end{cases} \\ y(0) & = y_{init}. \end{cases} \tag{8}$$

Note that, because of (4)-(5) and through P_{bat}, the state $y(t)$ depends on both $u = (\omega_{th}, C_{th}, C_{brk})$ and $z = (z_1, z_2, z_{th}, z_w)$. We now focus on various constraints. First, we have final and distributed constraints over the state:

$$y(T) \ \ge \ y_{fin}, \tag{9}$$

$$y(t) \ \in \ [y_{low}, y_{up}], \quad t \in [0, T]. \tag{10}$$

The final constraint is set in order to make the average fuel consumption significant of what it would be over a long time. Distributed constraints are classical requirements in order to avoid shortening the lifetime of the batteries. In addition, there are various bound constraints:

$$\begin{cases} 0 & \le & C_{brk} & \le & C_{brk}^{up}, & \\ C_{th}^{low} & \le & C_{th}(t) & \le & C_{th}^{up}, & \\ \omega_{th}^{low} & \le & \omega_{th}(t) & \le & \omega_{th}^{up} & \\ C_i^{low} & \le & C_i(t) & \le & C_i^{up}, & i = 1, 2, \\ \omega_i^{low} & \le & \omega_i(t) & \le & \omega_i^{up}, & i = 1, 2, \\ P_i^{low} & \le & P_i(t) & \le & P_i^{up}, & i = 1, 2, \\ P_{bat}^{low} & \le & P_{bat}(t) & \le & P_{bat}^{up} & \end{cases} \tag{11}$$

$$(z_1, z_2, z_w, z_{th}) \in \mathcal{K} = [z_{low}, z_{up}]. \tag{12}$$

Relation (12) is a compact notation for the bound constraints on the parameters of the gear pairs. The final goal is to reduce the amount of fuel used by the thermic motor, that only depends on thermic controls, $Q_{gas}(u(t)) = Q_{gas}(C_{th}(t), w_{th}(t))$. the optimization problem is

$$\operatorname*{Min}_{u,z} \int_0^T Q_{gas}(u(t))\mathrm{d}t, \qquad \text{subject to (4)} - (12). \tag{13}$$

The function $Q_{gas}(u(\cdot))$, is given at a finite number of points, and interpolated by a spline procedure.

3. A bilevel optimization formulation

In order to express (13) as a bilevel optimization problem, we split the design and control constraints in two levels and choose z as the variable in the higher level. Recall that $L^\infty(0,T;\mathbb{R}^3)$ is the set of essentially bounded functions from $(0,T)$ into $\mathbb{R}^3$, and "a.e." means that the property holds almost everywhere, that is, except on a set of Lebesgue measure 0. Accordingly, let $Y := [y_{low}, y_{up}]$ be the interval of $\mathbb{R}$ defined by the bounds over the state, and define the sets of feasible states and controls, respectively, as

$$\mathcal{Y} := \{y \in C([0,T]);\ y(t) \in Y,\ t \in [0,T]\}, \tag{14}$$

where $C([0,T])$ is the space of continuous functions over $[0,T]$, and

$$\mathcal{U}(z) := \Big\{(C_{th}, \omega_{th}, C_{brk}) \in L^\infty(0,T;\mathbb{R}^3);\ \text{(9)-(12) holds a.e.};\ y \in \mathcal{Y}\Big\}. \tag{15}$$

The upper level problem is

$$\min_{z\in\mathcal{K}} \mathcal{V}(z), \tag{16}$$

where

$$\mathcal{V}(z) := \min_{u\in\mathcal{U}(z)} \int_0^T Q_{gas}(u(t))\mathrm{d}t. \tag{17}$$

The objective function of the higher level problem, $\mathcal{V}(\cdot)$, is a nonsmooth and nonconvex function. We now recall a few notions relevant in this setting, see [7] and [14].

3.1. Some elements of Nonsmooth Analysis

A nondifferentiable Lipschitz continuous function $\Phi : \mathbb{R}^{N_z} \to \mathbb{R}$ has at any $z \in \mathbb{R}^{N_z}$ a *Clarke directional derivative* in any direction $d \in \mathbb{R}^{N_z}$, defined by

$$\Phi'(z;d) := \limsup_{h\to 0, t\downarrow 0} \frac{\Phi(z+td+h) - \Phi(z+h)}{t}.$$

The associated *Clarke subdifferential* is the (nonempty, closed and convex) set

$$\partial\Phi(z) := \left\{ s \in \mathbb{R}^{N_z}; \langle s, d\rangle \le \Phi'(z;d), \text{ for all } d \in \mathbb{R}^{N_z} \right\}.$$

We say that Φ is *weakly semismooth* at z when for each $d \in \mathbb{R}^{N_z}$ and for sequences $\{t_k\} \downarrow 0^+$, $\{s_k\} \subset \partial\Phi(z + t_k d) \in \mathbb{R}^{N_z}$ the sequence $\{\langle s_k, d\rangle\}$ converges. In this case, it holds that

$$\lim_k \langle s_k, d\rangle = \Phi'(z;d) = \lim_{t\downarrow 0} \frac{\Phi(z+td) - \Phi(z)}{t},$$

i.e., the limit is equal to the classical directional derivative.

A *stationary point* of the bound constrained problem

$$\min_{z\in[z_{low}, z_{up}]} \Phi(z), \tag{18}$$

is a point $\bar z$ satisfying $0 \in \partial\Phi_r(\bar z)$, where for $r > 0$ big enough we define the penalized function $\Phi_r(z) := \Phi(z) + r \inf\{\|z' - z\| : z' \in \mathcal{K}\}$.

3.2. Function and subgradient computation

Numerical methods for solving (18) need at each iterate z_k the value of the function, $\Phi_k := \Phi(z_k)$, and of one subgradient, $s_k \in \partial\Phi(z_k)$. In the case of the value function $\mathcal{V}(z)$ of the upper level optimal control problem, these computations are very expensive and it is crucial to organize them in order to obtain both Φ_k and s_k simultaneously. We first address this point here in a more general setting. In Section 5 below we show that once a suitable discretization scheme is applied, the upper level objective function can be interpreted as the value function of a nonlinear program parameterized by z.

Accordingly, for $z \in \mathcal{K}$ consider the following framework, to which our optimal control problem boils down, once a time and space discretization is performed:

$$\operatorname*{Min}_u F(u,z); \quad u \in \mathcal{F}(z), \tag{19}$$

where

$$\mathcal{F}(z) := \{u \in \mathbb{R}^m;\ g_j(u,z) \le 0, \quad j = 1, \cdots, N_g\}, \tag{20}$$

and the functions F and g_j, $j = 1, \cdots, N_g$, are assumed to be continuously differentiable. Denote the value function, solution set and Lagrangian function of the nonlinear programming problem (19) as

$$\begin{aligned} \Phi(z) &:= \inf_u \{F(u,z);\ u \in \mathcal{F}(z)\}. \\ S(z) &:= \{u \in \mathcal{F}(z);\ F(u,z) = \Phi(z)\}. \\ \mathcal{L}(u,z,\lambda) &= F(u,z) + \lambda^\top g(u,z), \end{aligned}$$

where $\top$ denotes the transposition operator. If a solution u of (19) is such that the gradients of active constraints are linearly independent, then there exists a unique Lagrange multiplier denoted $\lambda(u)$, characterized by the relations

$$\lambda(u) \in \mathbb{R}_+^{N_g};\ D_u\mathcal{L}(u,z,\lambda(u)) = 0;\ \lambda_j(u) g_j(u,z) = 0, \quad j = 1, \cdots, N_g$$

where the symbol D_u stands for the partial derivative with respect to u. By conv we mean the convex closure of the set E (smallest closed convex set containing E).

THEOREM 1. *With the notation above, suppose that* $\bigcup\{\mathcal{F}(z); z \in \mathcal{K}\}$ *are locally uniformly compact, and that, for each* $u \in S(z)$, *the gradients of the active constraints are linearly independent. Then the following hold:*

(i) *The value function* $\Phi(\cdot)$ *is locally Lipschitz continuous with Clarke subdifferential* $\partial\Phi(z) = \operatorname{conv}\{D_z\mathcal{L}(u,z,\lambda(u));\ \ u \in S(z)\}$.

(ii) *The value function* $\Phi(\cdot)$ *is weakly semismooth.*

Proof. The expression of Clarke gradients, using the Mangasarian-Fromovitz qualification condition, can be found in [8]. The first item is an immediate consequence of Gauvin's result. As for the second item, see the companion paper [6]. A similar result, for unconstrained problems, was first given in [14]. ∎

A consequence of Theorem 1 is that, when computing $\Phi(z_k)$, both a minimizer $\bar{u}_k$ and multipliers $\lambda(\bar{u}_k)$ are known. So, with negligible extra calculations, the subgradient $D_z\mathcal{L}(\bar{u}_k, z_k, \lambda(\bar{u}_k))$ is readily available.

We now explain the numerical method used for solving (18).

4. Solving the higher level problem: bundle methods

The constrained minimization of a semismooth function is a difficult problem that calls for specialized methods. Bundle methods are at the moment the most efficient and promising methods for nonsmooth optimization, see for instance [16].

Under the assumption of semismoothness, bundle methods ensure convergence to a stationary point, see [17]. Since stationarity is a local notion, the efficiency of the method will strongly rely on the chosen starting point. A usual measure for the performance of a nonsmooth optimization method is the number of function/subgradient evaluations required to converge.

For our implementation, we apply a penalized variant of bundle methods, based on the (unconstrained and convex) variant of [11]. We refer to this work for a complete description of each step of the algorithm, and sketch here just the basic ideas.

In order to avoid oscillations, it is important that the algorithm "remembers" the best point obtained so far. Bundle methods collect relevant data along iterations in the following *bundle* of information:

$$\{\Phi(z_j), s(z_j)\}_{j \leq \min(k, maxbun)} \quad \text{and} \quad \hat{z},$$

the point with smallest objective value until the current iteration k. Here, $maxbun$ is a positive integer limiting the maximum size of the bundle. A fundamental concept in bundle methods is the notion of *model* of Φ. This piecewise-linear approximation of Φ has the form

$$\hat{\Phi}_k(z) := \max_{j \leq \min(k, maxbun)} \{\Phi_j + \langle s(z_j), z - z_j \rangle + \gamma_j \|z_j - \hat{z}\|^2 \rangle\}.$$

The third righthand side term above allows, via γ_j, to have a model satisfying $\hat{\Phi}_k(z) \leq \Phi(z)$ for all z near $\hat{z}$. When Φ is convex, $\gamma_j \equiv 0$ and $\hat{\Phi}_k(z) \leq \Phi(z)$ for all z, so the model always stays below Φ. When Φ is nonconvex, it may happen that $\Phi(\hat{z}) - \Phi_j - \langle s(z_j), \hat{z} - z_j \rangle < 0$; in this case γ_j is a convexification factor that should be taken "big enough" to ensure that at least locally the model stays below Φ, see also [12], [15].

With the bundle information, the algorithm generates two sequences of points. One is the sequence of sample points used to define the model. We call these points *candidate* points and denote them by z_k. A second sequence consists of those sample points where the objective function decreases. We call these points *stability* or *prox-centers* and denote them by $\hat{z}$. Note that $\{\hat{z}\}$ is a subsequence of $\{z_k\}$. Candidate points are computed by solving a quadratic program:

$$\min_{z \in \mathcal{K}} \{\hat{\Phi}_k(z) + \frac{1}{2}\mu\|z - \hat{z}\|^2\}, \tag{21}$$

where $\mu > 0$ is a parameter playing the role of a line-search stepsize in this nonlinear context. It should be noted that (21) is a convex quadratic programming problem (QP). Contrary to a widespread belief, convex QPs are not much more difficult to solve than linear programs. The underlying resolution technique is essentially the same (active sets strategies, interior points, etc). In actual implementations a QP such as (21) is solved through its dual. This dual QP has a very special structure, namely, a simplicial feasible set. Good QP solvers exploit such structure, as well as the fact of knowing how QP-data changes along iterations in order to define warm starts. As a result, solving (21) does not require much more computational work than solving a linear program. Finally, a pay-off compensating the extra computational burden is that having QP problems allows to compress the bundle of information by using the so-called aggregation techniques. Aggregation is the synthesis mechanism that condenses essential information of the bundle into one single data, see e.g. [5, Part II]. With this technique, the size of (21) is kept controlled along iterations and convergence is preserved.

Candidate points are declared new stability centers only when they provide a sufficient decrease for Φ:

$$\Phi(z_k) \leq \Phi(\hat{z}) - m_1\delta_k,$$

where $m_1 \in]0,1[$ is a parameter and $\delta_k = \Phi(\hat{z}) - \hat{\Phi}(z_k)$ measures the decrease predicted by the model.

The algorithm stops when δ_k is small enough:

$$\text{If } \delta_k \leq tol_\delta, \text{ stop.}$$

Writing the dual of (21) it can be seen [5, Section 9.3.1], that this stopping test ensures the existence of a tolerance ε and a regularized subgradient G, satisfying

$$\Phi(z) \geq \Phi(\hat{z}) + \langle G, z - \hat{z}\rangle - \varepsilon, \text{ for all } z \text{ close enough to } \hat{z}.$$

In other words, the last stability center $\hat{z}$, is an ε-stationary point.

We now address in detail the question of computing the value function and one subgradient for a given z.

5. Solving the lower level optimal control problem

For each design parameter z, the objective function of the higher level problem, defined in (17) is the optimal value function of an optimal control problem, with state equation (8). There is only one state variable, and hence, this optimal control problem can be solved efficiently by discretization of the state space.

5.1. The Hamilton-Jacobi-Bellman equation

In this section we briefly recall the standard ways of solving optimal control problems through a discretization of the associated HJB equation, in the case of a single state variable. Consider the dynamic system

$$\dot{y}(t) = f(t, y(t), u(t), z), \quad t \in [\tau, T]; \quad y(\tau) = x,$$

whose solution is denoted $y^{x,\tau}_{u,z}$. The control $u(t)$ belongs to some set $U(t,z)$, defined by

$$U(t,z) := \{u \in \mathbb{R}^m;\ g_j(t,u,z) \leq 0,\ j = 1, \cdots, N_g\}. \tag{22}$$

Again, let $Y := [y_{low}, y_{up}]$ and define $\mathcal{Y}$ by (14). Given $(\tau, x) \in [0,T] \times Y$, consider the function

$$v(\tau, x, z) = \inf_{u \in \mathcal{U}(z)} \int_\tau^T \ell(t, y^{x,\tau}_{u,z}(t), u(t))\mathrm{d}t + \phi(y^{x,\tau}_{u,z}(T))$$

where

$$\mathcal{U}(z) := \left\{ u \in L^\infty(0, T, U(t,z));\ y^{x,\tau}_{u,z} \in \mathcal{Y} \right\}.$$

In our application the state is the state of charge of the battery, and $U(t,z)$ is the set of controls for which the bounds (11) hold; the dependance on time is due to (6). The running cost ℓ is the amount of fuel used by the thermic motor $Q_{gas}(u(t))$, and the final cost ϕ is just a penalization term, proportional to the distance to the set of states satisfying the final state constraint.

We have that v is the value function of the family of optimal control problems starting at an arbitrary time $s \in [0,T]$, and initial state $x \in Y$. Under standard assumptions (all functions Lipschitz continuous and bounded, uniform compactness of $U(t,z)$, and instantaneous local controllability), it is known [1, 2, 13] that v is the unique solution, in the so-called viscosity sense, of the Hamilton-Jacobi-Bellman (HJB) equation

$$\begin{cases} -\partial_\tau v(\tau, x, z) + \mathcal{H}(x, \partial_x v(\tau, x, z)) = 0 & \text{for } (\tau, x, z) \in [0, T[\times Y \times \mathcal{K} \\ v(T, x, z) = \phi(x) & \text{for } (x, z) \in Y \times \mathcal{K}, \end{cases}$$

where the *Hamiltonian function* $\mathcal{H}$, defined for all $(t, x, p, z) \in [0,T] \times Y \times \mathbb{R} \times \mathcal{K}$ by

$$\mathcal{H}(t, x, p, z) := \sup_{u \in U(t,z)} \{-\ell(t, x, u) - pf(t, x, u, z)\},$$

and appropriate boundary conditions taking into account the state constraint, see [9]. In our application the following hypothesis is satisfied:

$$\inf_{u\in U(t,z)} f(t,x,u,z) < 0 < \sup_{u\in U(t,z)} f(t,x,u,z), \tag{23}$$

for each $(t,x,z) \in [0,T] \times Y \times \mathcal{K}$. This implies that the value function is everywhere finite and continuous.

Therefore, the optimal cost function v can be approximated by discretizing the HJB equation. We now show how a classical finite differences discretization scheme can be used in our framework.

5.2. Discretization method

The discretization of the HJB equation yields approximate values of the value function v, evaluated on a mesh of points, representing time and space. More precisely, let $\Delta x = (y_{up} - y_{low})N_x$ be the space step (keeping into account both the state and control variables), and for a positive integer N, let $\Delta t := T/N$ be the time step. We define the mesh of points

$$x_i := i\Delta x,\ i = 0,\dots,N_x; \quad t_n := n\Delta t,\ n = 0,\dots,N,$$

and denote by $V_i^n(z)$ (or V_i^n when there is no ambiguity) the corresponding approximate values of $v(t_n, x_i, z)$.

The *centered finite differences scheme* for the HJB equation is

$$\begin{cases} V_i^n = \frac{1}{2}\left(V_{i+1}^{n+1} + V_{i-1}^{n+1}\right) \\ \qquad +\Delta t \min\limits_{u\in U(t_n,z)} \left\{ \ell(t_n,x_i,u) + \dfrac{V_{i+1}^{n+1} - V_{i-1}^{n+1}}{2\Delta x} f(t_n,x_i,u,z) \right\} \\ \qquad\qquad \text{for } i = 1,\dots,N_x - 1,\ n = 0,\dots,N-1, \\ V_0^n = V_0^{n+1} \\ \qquad +\Delta t \min\limits_{u\in U_+(t_n,z)} \left\{ \ell(t_n,y_{low},u) + \dfrac{V_1^{n+1} - V_0^{n+1}}{\Delta x} f(t_n,y_{low},u,z) \right\} \\ \qquad\qquad \text{for } n = 0,\dots,N-1, \\ V_{N_x}^n = V_{N_x}^{n+1} \\ \qquad +\Delta t \min\limits_{u\in U_-(t_n,z)} \left\{ \ell(t_n,y_{up},u) + \dfrac{V_1^{n+1} - V_0^{n+1}}{\Delta x} f(t_n,y_{up},u,z) \right\} \\ \qquad\qquad \text{for } n = 0,\dots,N-1, \\ V_i^N = \phi(x_i), i = 0,\dots,N_x, \end{cases} \tag{24}$$

where we denote by $U_+(t,z)$ (resp. $U_-(t,z)$) the set of controls in $U_+(t,z)$ having a nonnegative (resp. positive) dynamics, when the state is y_{low} (resp. y_{up}); in view of (23), these sets are non empty. The scheme is said to be *monotone* when, at each time step n, V_i^n is a nondecreasing function of $\{V_j^{n+1}\}$, for all $j = 0, \dots, N_x$. It is easily checked that a sufficient condition for monotonicity to hold is that, for all $z \in \mathcal{K}$,

$$\Delta t \sup\{|f(t,x,u,z)|;\ t \in [0,T], x \in Y, u \in U(t,z), z \in \mathcal{K}\} \le \Delta x. \quad (25)$$

This is a natural extension of the well known Courant Friedrich Levy condition for hyperbolic equations. It implies that during a time step Δt the physical distance of the propagation $\Delta t|f(t,x,u,z)|$ is bounded from above by the space step. In other words, the speed of propagation in the numerical scheme is always greater or equal than the physical speed of the process.

Whenever (25) holds the discretization scheme is *convergent* (cf. [3, 1, 13]), i.e.,

$$v(\tau,x,z) = \lim_{\substack{n\Delta t \to \tau \\ i\Delta x \to x}} V_i^n(z)$$

for any $\tau \in [0,T]$, $x \in \mathbb{R}$ and z. Moreover, convergence is uniform in x, over any bounded interval of $\mathbb{R}$. Furthermore, this limit is also uniform over the parameter z, when the latter is restricted to a bounded set, like $\mathcal{K}$.

5.3. Sensitivity analysis

It remains to show how to compute a subgradient of the value function v. To this aim, we mold the discrete HJB equation into the structure assumed in Theorem 1. This is possible by interpreting the discretization scheme as the application of the Dynamic Programming Principle to a certain stochastic optimal control problem.

More precisely, denote $f^\Delta(t,x,u,z) := \Delta t(\Delta x)^{-1} f(t,x,u,z)$, and for any $u \in \mathcal{U}(z)$ and $n \le N$, consider the following square matrix of order $N_x + 1$:

$$M_{ij}^n(u,z) := \begin{cases} \frac{1}{2} + \frac{1}{2} f^\Delta(t_n, x_i, u, z), & \text{if } j = i-1, \\ \frac{1}{2} - \frac{1}{2} f^\Delta(t_n, x_i, u, z), & \text{if } j = i+1, \\ 0 & \text{otherwise} \end{cases}$$

for $i = 1, \dots, N_x - 1$, and

$$\begin{array}{lcl} M_{0,0}^n(u,z) & := & 1 - f^\Delta(t_n, y_{low}, u, z), \\ M_{0,1}^n(u,z) & := & f^\Delta(t_n, y_{low}, u, z), \\ M_{N_x,N_x-1}^n(u,z) & := & -f^\Delta(t_n, y_{low}, u, z), \\ M_{N_x,N_x}^n(u,z) & := & 1 + f^\Delta(t_n, y_{low}, u, z), \end{array}$$

the other terms being equal to 0.

Condition (25) implies that the matrix components are all nonnegative and the sum by columns equals 1 for any row. Therefore, $M^n(u,z)$ is stochastic transition matrix. Let $c_i^n(u) := \Delta t \ell(t_n, x_i, u)$, and denote by $V^{n+1}(z)$ the vector with components $V_j^{n+1}(z)$, $j = 0, \dots, N_x$, defined in (24). It holds that

$$\begin{cases} V_i^n(z) &= \min\limits_{u \in U_i^n(z)} \left\{ c_i^n(u) + \sum\limits_j M_{ij}^n(u,z) V_j^{n+1}(z) \right\} \\ V_i^N(z) &= \phi(x_i). \end{cases} \tag{26}$$

where $U_i^n(z) := U(t_n, z)$, $i = 1, \dots, N_x - 1$, $U_0^n(z) := U_+(t_n, z)$, and $U_{N_x}^n(z) := U_-(t_n, z)$. With this new notation, see [10], the vector $V^n(z)$ given by this recurrence can be interpreted as the application of the Dynamic Programming Principle to the stochastic problem

$$V_i^n(z) = \min_{u \in \mathcal{U}^\Delta(t_n, z)} \mathbb{E} \left(\sum_{k=n}^{N-1} c_{x(k)}^k(u_{x(k)}^k) + \phi(x(N)) \,|\, x_n = i \right).$$

Here $x(k)$ is the realization at time k of a Markovian stochastic process which takes integer values in $0, \dots, N_x$ and whose probability of transition from state i to state j is given by $M_{ij}^n(u,z)$, and $\mathcal{U}^\Delta(t_n, z)$ denotes the set of feedback controls such that $u_i^k \in U_i^k(z)$ for all i and $k = n, \dots, N$.

Reformulation (26) allows to write the i^{th}-component of the approximate value function $V^n(z)$ in the format (19). The following result is a consequence of Theorem 1.

THEOREM 2. *With the notation above, suppose that the sets $U(t_n, z)$ from (22) are such that $\cup_{z \in \mathcal{K}} U(t_n, z)$ is bounded, and let the function and mappings: c_i^n, M_{ij}^n, and g be continuously differentiable. Suppose in addition that at each solution $\bar{u}$ to (26), the gradients of the active constraints are linearly independent. Then the following hold for all $i = 0, \dots, N_x$ and $n \le N$:*

(i) *The value function $V_i^n(\cdot)$ is locally Lipschitz continuous, and one of its Clarke subgradients $s^{n,i} \in \partial V_i^n(z)$, for $i = 0, \dots, N_x$, and $n = 0, \dots, N$, satisfies the following backward recurrence relation, where $M_i^n(u,z)$ denotes the ith row of $M^n(u,z)$:*

$$\begin{cases} s_j^{n,i} &= D_{z_j} c_i^n(u_i^n) + D_{z_j}^\top g(t_n, u_i^n, z) \lambda_i^n \\ &\quad + D_{z_j} M_i^n(u_i^n, z) V^{n+1}(z) + M_i^n(u_i^n, z) s_j^{n+1,i} \\ s_j^{N,i} &= 0, \end{cases}$$

where $u_i^n \in U_i^n(z)$ attains the minimum in (26), and λ_i^n is an associated Lagrange multiplier.

(ii) *The value function $V_i^n(\cdot)$ is weakly semismooth.*

6. Numerical results

We now give some results obtained with our approach, for four test problems.

In Section 4 we mentioned that for the higher level problem (18), the total number of function/subgradient evaluations, i.e., of v_k/s_k, is a good measure of efficiency of the bundle method. As shown in Theorem 2, each discrete approximation of v_k (and of s_k), many nonlinear programs of the form (19) need to be solved. Furthermore, for the approximate values to be accurate, $V_i^n(z_k)$ needs to "converge" to $v(t_i, x_n, z_k)$, and this, for each meshpoint (t_i, x_n). We solve those many nonlinear programs by applying a warm start to the sequential quadratic programming algorithm (SQP) from [4]. This is a globalized Newton type method see [5, Part III], which requires the evaluation of the corresponding objective function and constraints, as well as of the respective first and second order derivatives. These computations are done "exactly" using one and two dimensional spline interpolation. In order to provide a good starting point to this SQP algorithm, we evaluate the cost function and constraints on a coarse mesh. Only when the discrete scheme fails to converge, we refine the mesh in a neighbourhood of the best point obtained with the coarse mesh and restart the SQP algorithm.

We represent in figure 4 the vehicle speed as a function of time; the final time is 400 seconds. For all our test problems we take $C_{th} \in [-10, 140]$ in N.m, $\omega_{th} \in [120, 680]$ (hundreds of turns by minute), $y \in [0.4, 0.6]$ (without dimension); $\Delta x = 0.01$, $\Delta t = 0.5$, $N_x = 20$, and $y_{fin} = y_{low}$.

For the first three test problems only two design variables are considered, namely $z = (z_{th}, z_w)$. In the fourth example we include the design variables corresponding to the electrical motors, z_1 and z_2.

In our two first cases, we neglect losses in the electrical motors and take $P_{bat}(t) = C_1(t)\omega_1(t) + C_2(t)\omega_2(t) + P_0$.

6.1. Case 1

Here we take $\mathcal{K} := [1, 1.5] \times [3, 3.5]$, $t \in [325, 330]$ and $N = 10$. Table I shows, for different starting points z_0 (column 1) the solution obtained

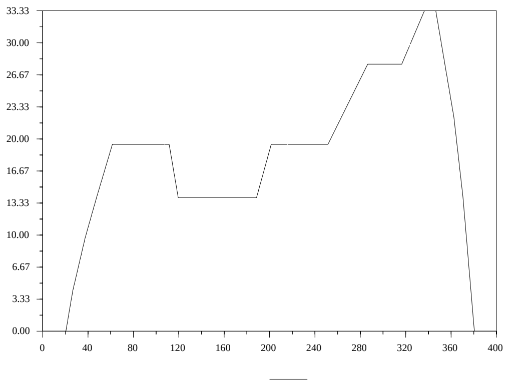


Figure 4. Vehicle speed function of time.

by the bundle method (column 2), the optimal value function (column 3), and the number of function/subgradient evaluations required to converge (column 4).

Table I. Results for Case 1

z_0	z_{optim}	v_{optim}	# v/s
(1.5,3.5)	(1.5, 3.0)	$0.1264\ 10^{-2}$	4
(1.0,3.0)	(1.5, 3.0)	$0.1264\ 10^{-2}$	4
(1.0,3.5)	(1.0, 3.5)	$0.2197\ 10^{-2}$	1
(1.1,3.1)	(1.5, 3.0)	$0.1264\ 10^{-2}$	7

As shown in Figs. 5 and 6, for this example (18) has two stationary points. In our runs, we found one of such local minima.

6.2. Case 2

In Table II, we give some results for the entire time interval $t \in [0, 400]$, so $N = 800$. Also, $\mathcal{K} := [0.5, 1.5] \times [3, 6]$.

For this case, each evaluation v/s for the bundle method required the resolution of $N_x N = 16\,000$ nonlinear programs, which on a Pentium III computer took 4 minutes in total. The first problem needed a total of

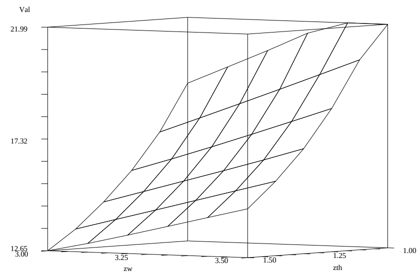
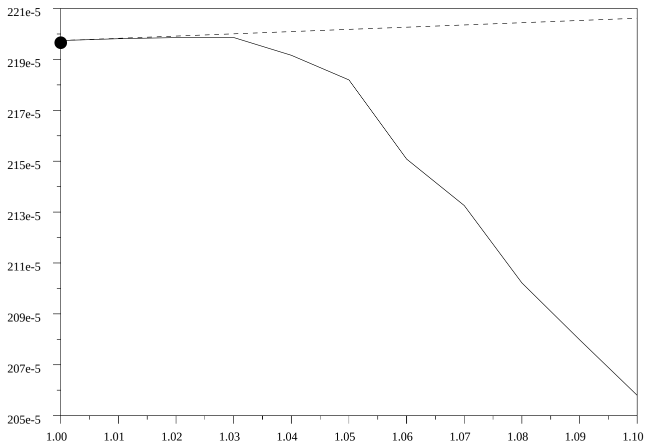


Figure 5. Dependence of value function with respect to z_w and z_{th}.



$z_{th} = 3.5$

Figure 6. Focus around $z_{th} = 1$.

240 000 nonlinear programs, and the optimization process took almost one hour.

Table II. Results for Case 2

z_0	z_{optim}	v_{optim}	# v/s
$(1.0, 5.5)$	$(1.5, 3.0)$	$0.6858\ 10^{-1}$	15
$(1.5, 3.0)$	$(1.5, 3.0)$	$0.6858\ 10^{-1}$	1

6.3. Case 3

In Table III we give results obtained when we take into account losses of the electrical motors in $P_{bat}(t)$. All other data are the same than for Case 2, except for $\mathcal{K} := [0.5, 1.5] \times [3, 4]$.

Table III. Results for Case 3

z_0	z_{optim}	v_{optim}	# v/s
$(1.0, 3.0)$	$(1.5, 3.0)$	$0.721\ 10^{-1}$	3
$(1.5, 3.0)$	$(1.5, 3.0)$	$0.721\ 10^{-1}$	1

Now, first and second order derivatives for each nonlinear program are computed using spline interpolation and each evaluation v/s for the bundle method takes 15 minutes.

6.4. Case 4

Finally, we take the same data than for Case 3, but $t \in [0, 50]$ and with four design variables, so

$$\mathcal{K} := [0.5, 1.5] \times [0.5, 1.5] \times [0.5, 1.5] \times [3, 4].$$

Table IV shows that the same final point is obtained for two widely different initial points.

Table IV. Results for Case 4

z_0	z_{optim}	v_{optim}	# v/s
$(1.0, 3.5, 0.8, 0.8)$	$(1.5, 3, 1, 0.7028)$	$0.92\ 10^{-2}$	10
$(1.5, 3.0, 1, 0.7028)$	$(1.5, 3, 1, 0.7028)$	$0.92\ 10^{-2}$	6

7. Conclusion

In this paper we have shown how to design and implement a bilevel optimization method for solving an optimal control problem with design parameters. We have dealt with the case of scalar states, but the method readily extends to multidimensional problems, although its cost may become excessive for high values of the state dimension. It is interesting to observe that most of the computing time is spent in computing the nonlinear functions over a mesh of controls, at each point of the mesh of the state space. This costly step has to be done at each iteration of the upper-level problem. In comparison, the use of the nonlinear programming solver is relatively cheap since it needs few iterations. This is not surprising since the starting point should be good, and the control dimension is small (two in our application). The difficulty with the nonlinear programming solver is that it has to work correctly many times (here, hundreds of thousands) and hence, it has to be robust.

One by-product of this study is an evaluation of the possible use dynamic programming algorithms for real-time control. In such a case the nonlinear programming solver is useless, as already observed, and a coarse mesh for controls is enough. Since predictions are reliable, and the algorithm is guaranteed to converge in a given time, it is a serious candidate for real-time control.

For the computation of optimal design parameters, we note that applying nonlinear programming solvers to a time-discretized version of this problem will certainly give shorter computation times. The advantage of the bilevel optimization is to provide a global solution to the lower-level problem.

References

1. M. Bardi and I. Capuzzo-Dolcetta. *Optimal control and viscosity solutions of Hamilton-Jacobi-Bellman equations.* Systems and Control: Foundations and Applications. Birkhäuser, Boston, 1997.
2. G. Barles. *Solutions de viscosité des équations de Hamilton-Jacobi*, volume 17 of *Mathématiques et applications.* Springer, Paris, 1994.
3. G. Barles and P. E. Souganidis. Convergence of approximation schemes for fully nonlinear second order equations. *Asymptotic Analysis*, 4:271–283, 1991.
4. G. Blanchon, J.C. Dodu, and J.F. Bonnans. Optimisation des réseaux électriques de grande taille. In A. Bensoussan and J.L. Lions, editors, *Analysis and optimization of systems*, volume 144 of *Lecture Notes in Information and Control Sciences.* Springer-Verlag, Berlin, 1990.
5. J.F. Bonnans, J. Ch. Gilbert, C. Lemaréchal, and C. Sagastizábal. *Numerical optimization.* Universitext. Springer-Verlag, Berlin, 2002.

6. J.F. Bonnans, Th. Guilbaud, and H. Zidani. Bilevel optimization for parametric optimal control problems. In preparation, 2002.
7. F.H. Clarke. *Optimization and nonsmooth analysis.* J. Wiley, New York, 1983.
8. J. Gauvin. The generalized gradient of a marginal function in mathematical programming. *Mathematics of Operations Research*, 4:458–463, 1979.
9. H. Ishii and S. Koike. A new formulation of state constraint problems for first-order PDEs. *SIAM Journal on Control and Optimization*, 34(2):554–571, 1996.
10. H.J. Kushner and P. G. Dupuis. *Numerical methods for stochastic control problems in continuous time*, volume 24 of *Applications of mathematics.* Springer, New York, 2001. Second edition.
11. C. Lemaréchal and C. Sagastizábal. Variable metric bundle methods: from conceptual to implementable forms. *Mathematical Programming*, 76:393–410, 1997.
12. C. Lemaréchal, J.-J. Strodiot, and A. Bihain. On a bundle method for nonsmooth optimization. In O.L. Mangasarian, R.R. Meyer, and S.M. Robinson, editors, *Nonlinear Programming 4*, pages 245–282. Academic Press, 1981.
13. P. L. Lions. *Generalized solutions of Hamilton-Jacobi equations*, volume 69 of *Research notes in mathematics.* Pitman, Boston, 1982.
14. R. Mifflin. Semismooth and semiconvex functions in constrained optimization. *SIAM J. Control and Optimization*, 15:959–972, 1977.
15. R. Mifflin. A modification and extension of Lemarechal's algorithm for nonsmooth minimization. *Math. Programming Stud.*, 17:77–90, 1982.
16. J. Outrata, M. Kočvara, and J. Zowe. *Nonsmooth approach to Optimization Problems with Equilibrium Constraints.* Kluwer Academic Publishers, Boston, 1998.
17. H. Schramm and J. Zowe. A version of the bundle idea for minimizing a nonsmooth function: conceptual idea, convergence analysis, numerical results. *SIAM J. on Optimization*, 2(1):121–152, 1992.
18. H. Sonnet. *Dictionnaire des mathématiques appliquées.* Librairie de L. Hachette et Cie, Paris, 1867.