



**HAL**  
open science

# A bilevel optimal control method and application to the hybrid electric vehicle

Olivier Cots, Rémy Dutto, Sophie Jan, Serge Laporte

► **To cite this version:**

Olivier Cots, Rémy Dutto, Sophie Jan, Serge Laporte. A bilevel optimal control method and application to the hybrid electric vehicle. 2023. hal-04359870

**HAL Id: hal-04359870**

**<https://hal.science/hal-04359870>**

Preprint submitted on 21 Dec 2023

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# A bilevel optimal control method and application to the hybrid electric vehicle

Olivier Cots\*      Rémy Dutto<sup>†‡§</sup>      Sophie Jan<sup>‡</sup>      Serge Laporte<sup>‡</sup>

December 21, 2023

## Abstract

In this article we present a new numerical method based on a bilevel decomposition of optimal control problems. A strong connection between the proposed method and the classical indirect multiple shooting method is shown in the regular case, thanks to a link between the Bellman's value function and the costate from Pontryagin Maximum Principle. The value functions are needed in our bilevel decomposition but they are generally difficult to compute. We approximate them by neural networks that have a high potential of generalization and that provide an efficient computation of the gradient of the cost function. We apply the proposed method to an industrial problem, consisting in the determination of torque split and gear shift of a hybrid electric vehicle, the objective being the minimization of the fuel consumption on a given representative cycle. Numerical methods and results are discussed, as well as the possible improvements of the proposed approach.

**Keywords.** Optimal Control, Bilevel Optimization, Hybrid Electric Vehicle, Indirect Shooting, Value Function, Neural Network.

## 1 Introduction

Hybrid Electric Vehicles (HEVs) or Plug-in Hybrid Electric Vehicles (PHEVs) are seen as a solution for fuel saving and/or reduction of polluting emissions. These kinds of vehicles use two sources of energy, respectively fuel and electricity, and at least two motors, respectively Internal Combustion Engine (ICE) and Electric Motor (EM). A control law provides the strategy for gear shift and torque split between these two motors in order to minimize the fuel consumption with respect to the needed torque and wheel speed demand on a given cycle. The cycle, that is the vehicle speed and acceleration on a given path, is assumed to be known. In this article we consider for the numerical experiments the *Worldwide harmonized Light vehicles Test Cycle* (WLTC) which is commonly used by industry for evaluation of fuel consumption and pollutant emission. In the case of embedded solution, the prediction of speed on connected vehicle is also a problem that is currently being studied [9, 43], but we shall not address it here.

The torque split and gear shift problem has been widely studied in the literature [16, 26, 41] and a lot of approaches have been proposed. For example, based-rules laws have been developed like thermostat strategy [22], state machine controller [30], or fuzzy logic solution [20]. Methods based on instantaneous optimization are also proposed, like the Equivalent Consumption Management

---

\*Institut de Recherche en Informatique de Toulouse, UMR CNRS 5505, Université de Toulouse, INP-ENSEEIH, France.

<sup>†</sup>Vitesco Technologies, Toulouse, France.

<sup>‡</sup>Institut de Mathématiques de Toulouse, UMR CNRS 5219, Université de Toulouse, UPS, France.

<sup>§</sup>Corresponding author: [remy.dutto@orange.fr](mailto:remy.dutto@orange.fr).

Strategy (ECMS) introduced by [29] which considers that the cost is the sum of the ICE fuel consumption and an equivalent factor times the battery state of charge deviation. This method can be seen as an application of the Pontryagin Maximum Principle (PMP) with some restrictive assumptions [35, 42]. Finally, other methods derived from the ECMS have been developed, which differ by the parameterization of the equivalent factor and its possible online adaptation [21, 27, 28, 36, 42, 43].

Optimal control theory provides a natural frame for the development of methods for control laws determination in this context. Indeed, the problem studied here can be seen as a classical non-autonomous optimal control problem with a Lagrange cost. This kind of problem is composed of a set of controlled ordinary differential equations, command bounds and initial and final state constraints. A first class of numerical methods is based on the Pontryagin Maximum Principle [24] which provides necessary optimality conditions for optimal control problems and leads to a Boundary Value Problem (BVP). For instance, the indirect shooting method [37] consists in solving the BVP introducing the so-called shooting function and then solving the associated set of nonlinear equations by a Newton-like algorithm. This method has been used on a similar application, see [17, 23]. Another class of methods consists in discretizing the control and the state on a given grid of times. This leads to a large nonlinear constrained optimization problem (Mixed-Integer NonLinear Programming (MINLP) in our application) in finite dimension that can be solved by nonlinear programming algorithms. This method is called a direct method and has been used for instance in [39].

Direct and indirect methods are considered as local methods. Other methods that provide global minima exist, based on the Hamilton Jacobi Bellman (HJB) equation, such as dynamic programming [3, 34, 40] or reinforcement learning [19, 38]. However, the dynamic programming method is known to be time consuming and subject to the curse of dimensionality. For its part, the deep reinforcement learning uses a neural network as controller, that may not be trustable for some critical applications. Compared to HJB based and direct methods, indirect methods promise to be accurate and fast enough to provide online optimal control solutions. Unfortunately, they are known to be sensitive to the initial guess, due to the underlying Newton solver.

The torque split and gear shift problem studied here is characterized by a long cycle duration (1800s for the WLTC) compared to the maximum 100ms sampling period required for an accurate representation of the model dynamic. This leads to a long integration time and therefore a high sensitivity of the indirect shooting function, the latter being exacerbated by the possible high frequency of the command changes induced by the high torque request irregularity on this cycle. Motivated by the industrial application, the main objective of this paper is to present a new numerical method, more accurate than ECMS, more robust than simple shooting and fast enough to be embedded. This method can be sub-optimal but the computed solution needs to be close to the optimal one in terms of fuel consumption. The proposed method satisfies these requirements; it is based on a bilevel decomposition of the optimal control problem, and leads to a hierarchical resolution which is strongly linked to the indirect multiple shooting method as it will be shown. This decomposition uses the Bellman's value functions, which are generally difficult to compute, and for that reason they will be approximated. Neural networks have been chosen for their simplicity and their generalization capability. Moreover, the efficient computation of their gradients allows not only to use first order optimization methods but also to get good initial guesses for the shooting method.

The paper is organized as follows. The optimal control problem and the classical indirect methods are recalled in Section 2. In Section 3, the bilevel formulation is presented, the link with indirect multiple shooting is given and a novel approach is proposed. Finally, in Section 4, the application to the torque split and gear shift problem is described, as well as the numerical methods and the results. Section 5 concludes the article.

## 2 Optimal control framework

The main objective of this section is to present classical indirect methods used to solve optimal control problems. For that purpose, the general formulation of Optimal Control Problem (OCP) is introduced in Section 2.1. Then, necessary optimality conditions provided by the Pontryagin Maximum Principle are given in Section 2.2 and some classical assumptions are considered in Section 2.3. Finally, indirect simple and multiple shooting methods are presented in Sections 2.4 and 2.5.

### 2.1 Optimal control problem

We consider the following Optimal Control Problem in a general Lagrange form:

$$(OCP) \quad \begin{cases} \min_{x,u} \int_{t_0}^{t_f} f^0(t, x(t), u(t)) dt, & (1) \\ \text{s.t. } \dot{x}(t) = f(t, x(t), u(t)), & t \in [t_0, t_f] \text{ a.e.}, & (2) \\ u(t) \in U(t), & t \in [t_0, t_f], & (3) \\ c(x(t_0), x(t_f)) = 0, & & (4) \end{cases}$$

where the Lagrange cost  $f^0: \mathbb{R} \times \mathbb{R}^n \times \mathbb{R}^m \rightarrow \mathbb{R}$  and the state dynamics  $f: \mathbb{R} \times \mathbb{R}^n \times \mathbb{R}^m \rightarrow \mathbb{R}^n$  are two functions of class  $C^1$ . The initial and final times  $t_0 < t_f$  are fixed. The control domain  $U(t) \subset \mathbb{R}^m$  is a non-empty set for every  $t \in [t_0, t_f]$  with additional standard regularity assumptions (cf. [11, Chapter 4.2, Remark 5] for more information), and the mixed initial and final constraints  $c: \mathbb{R}^n \times \mathbb{R}^n \rightarrow \mathbb{R}^p$  is a function of class  $C^1$ , with  $p \leq 2n$ . Moreover,  $c$  is a submersion on  $c^{-1}(\{0\})$ , i.e.  $c'(a, b)$  is surjective for all pair  $(a, b)$  such that  $c(a, b) = 0$ . Solving (OCP) consists in finding a pair  $(x, u) \in \mathcal{D} = AC([t_0, t_f], \mathbb{R}^n) \times L^\infty([t_0, t_f], \mathbb{R}^m)$  of an absolutely continuous *state*  $x$  and an essentially bounded *control*  $u$  which minimizes the cost (1) and satisfies the constraints (2), (3) and (4).

**Definition 1.** An admissible point of (OCP) is a pair  $(x, u) \in \mathcal{D}$  which satisfies (2)-(4). A solution of (OCP) is an admissible point which minimizes the cost (1) among all other admissible points.

We denote by  $\mathcal{A} \subset \mathcal{D}$  the set of admissible points and  $\mathcal{S} \subset \mathcal{A}$  the set of solutions of (OCP). Moreover, considering two times  $t_1, t_2$  such that  $t_0 \leq t_1 < t_2 \leq t_f$ , and a state  $x_1 \in \mathbb{R}^n$ , we define:

- for any control law  $u$ , the mapping  $t \mapsto x(t, t_1, x_1, u)$  as the solution of the Cauchy problem

$$\dot{x}(t) = f(t, x(t), u(t)), \quad x(t_1) = x_1,$$

- the set of admissible controls

$$\mathcal{U}_{t_2, t_1, x_1} = \{u \in L^\infty([t_1, t_2], \mathbb{R}^m) \mid \forall t \in [t_1, t_2]: u(t) \in U(t) \text{ and } x(t, t_1, x_1, u) \text{ is well defined}\},$$

- the *extremity mapping*  $E_{t_2, t_1, x_1}: \mathcal{U}_{t_2, t_1, x_1} \rightarrow \mathbb{R}^n$ ,  $u \mapsto E_{t_2, t_1, x_1}(u) = x(t_2, t_1, x_1, u)$ ,
- the *accessibility set* at  $t_2$  from  $x_1$  at  $t_1$  by  $A(t_2, t_1, x_1) = E_{t_2, t_1, x_1}(\mathcal{U}_{t_2, t_1, x_1})$ .

**Remark 1.** In this paper, a simple and general framework compliant with the Pontryagin Maximum Principle and the industrial application is considered. More precisely, we deal with optimal control problems in a Lagrange form with fixed final time. However, thanks to [11], our results can be extended to Mayer, Lagrange and Bolza formulations, with fixed or free final time. Moreover, the regularity assumptions on the functions  $f^0$  and  $f$  can be weakened [11, 12].

## 2.2 Pontryagin Maximum Principle

According to the Pontryagin Maximum Principle [11, 24], if  $(x, u)$  is a solution of (OCP), then there exist a *costate* trajectory  $p \in AC([t_0, t_f], \mathbb{R}^n)$ , a scalar  $p^0 \in \{-1, 0\}$  and Lagrange multipliers  $\lambda \in \mathbb{R}^p$  such that  $(p, p^0)$  is non trivial, the pair  $(x, p)$  follows the Hamiltonian dynamics: for almost every  $t \in [t_0, t_f]$

$$\begin{aligned}\dot{x}(t) &= \nabla_p h(t, x(t), p(t), u(t)), \\ \dot{p}(t) &= -\nabla_x h(t, x(t), p(t), u(t)),\end{aligned}\tag{5}$$

and the control  $u$  satisfies the maximization condition: for almost every  $t \in [t_0, t_f]$

$$h(t, x(t), p(t), u(t)) = \max_{w \in U(t)} h(t, x(t), p(t), w),\tag{6}$$

where  $h(t, x, p, u) = p^0 f^0(t, x, u) + (p \mid f(t, x, u))$  is the *pseudo-Hamiltonian* and  $(a \mid b)$  is the usual scalar product on  $\mathbb{R}^n$ . Moreover, the costate  $p$  fulfills the transversality condition:

$$\begin{pmatrix} -p(t_0) \\ p(t_f) \end{pmatrix} - c'(x(t_0), x(t_f))^\top \lambda = 0.\tag{7}$$

**Remark 2.** *The transversality condition can be written in another form. Indeed, Equation (7) implies that the pair  $(-p(t_0), p(t_f))$  belongs to  $\text{Im } c'(x(t_0), x(t_f))^\top = (\text{Ker } c'(x(t_0), x(t_f)))^\perp$ . Since  $c$  is a submersion on  $c^{-1}(\{0\})$ , then  $c'(x(t_0), x(t_f)) : \mathbb{R}^n \times \mathbb{R}^n \rightarrow \mathbb{R}^p$  is a surjective linear map which means that  $\text{Im } c'(x(t_0), x(t_f)) = \mathbb{R}^p$ . We can hence construct a  $2n \times (2n - p)$  matrix  $B_c(x(t_0), x(t_f))$  whose columns form a basis of  $\text{Ker } c'(x(t_0), x(t_f))$ . Then the transversality condition is equivalent to*

$$B_c(x(t_0), x(t_f))^\top \begin{pmatrix} -p(t_0) \\ p(t_f) \end{pmatrix} = 0.$$

*The advantage of this formulation is that  $\lambda$  disappears.*

**Definition 2.** *An extremal is a pair  $z := (x, p) \in AC([t_0, t_f], \mathbb{R}^n) \times AC([t_0, t_f], \mathbb{R}^n)$  associated with a control  $u \in L^\infty([t_0, t_f], \mathbb{R}^m)$  that satisfy (5)-(6). A BC-extremal is an extremal that satisfies the boundary conditions (4) and the transversality conditions (7). An extremal is normal if  $p^0 = -1$  and abnormal if  $p^0 = 0$ .*

## 2.3 General assumptions

Motivated by the application, throughout the article, we consider that all the extremals we encounter are normal ( $p^0 = -1$ ). Moreover, we are only interested in optimal control problems that can be solved by the so-called indirect simple shooting method. Hence, for any optimal control problem, denoting  $z = (x, p)$ , we consider that the maximized *Hamiltonian*,

$$H(t, z) = \max_{u \in U(t)} h(t, z, u),\tag{8}$$

is well defined and smooth (at least of class  $C^1$ ) in the neighborhood of any given extremal. Under these assumptions, we can define the following Hamiltonian vector field:

$$\vec{H}(t, z) = (\nabla_p H(t, z), -\nabla_x H(t, z)),$$

and we get the following proposition.

**Proposition 1** ([1], Proposition 12.1). *A pair  $z = (x, p)$  is an extremal of (OCP) if and only if*

$$\dot{z}(t) = \vec{H}(t, z(t)).$$

**Remark 3.** *As in Remark 1, we do not consider the most general framework, but only a simple set-up in which the optimal control problem is well posed and indirect methods can be used. These assumptions are used to simplify the notations and to prove that the diagram on the Figure 1 is commutative. However, the proposed method detailed in Section 3.3 can be applied even if these assumptions are relaxed.*

## 2.4 Indirect simple shooting

Let us introduce the exponential map  $\exp_{\vec{F}}(t_1, t_0, x_0)$  of a vector field  $\vec{F}$  as the solution at time  $t_1$  of the Cauchy problem  $\dot{x}(t) = \vec{F}(t, x(t))$ ,  $x(t_0) = x_0$ . Applying the Pontryagin Maximum Principle to the Optimal Control Problem (OCP) leads to the following Two-Point Boundary Value Problem

$$(TPBVP) \quad \begin{cases} \exp_{\vec{H}}(t_f, t_0, z_0) = z_f, \\ g(z_0, z_f) = 0, \end{cases}$$

where  $g: \mathbb{R}^{2n} \times \mathbb{R}^{2n} \rightarrow \mathbb{R}^{2n}$  gathers the initial and final state and costate constraints, given by Equations (4) and (7), and can be written as follows

$$g(z_0, z_f) = \begin{pmatrix} c(x_0, x_f) \\ c^*(z_0, z_f) \end{pmatrix}. \quad (9)$$

According to Remark 2, the function  $c^*: \mathbb{R}^{2n} \times \mathbb{R}^{2n} \rightarrow \mathbb{R}^{2n-p}$  is defined by

$$c^*(z_0, z_f) = B_c(x_0, x_f)^\top \begin{pmatrix} -p_0 \\ p_f \end{pmatrix},$$

where  $z_0 = (x_0, p_0)$  and  $z_f = (x_f, p_f)$ . A well known method to solve the Two-Point Boundary Value Problem (TPBVP) is the so-called indirect simple shooting method which consists in finding a zero of the simple shooting function  $S_s: \mathbb{R}^{2n} \rightarrow \mathbb{R}^{2n}$  defined by

$$S_s(z) = g(z, \exp_{\vec{H}}(t_f, t_0, z)).$$

However, the simple shooting method suffers from numerical issues. Indeed, as shown in [5, 31, 32], the Hamiltonian dynamics is ill-conditioned because the divergence of the Hamiltonian is constant. This implies that  $S_s$  is highly sensitive with respect to the costate initialization, and even more when the control problem shows a long horizon time and/or is highly nonlinear. The multiple shooting method [4] has been developed to overcome these numerical difficulties.

## 2.5 Indirect multiple shooting

The main idea of the multiple shooting method is to integrate the Hamiltonian vector fields on  $N+1$  smaller sub-intervals and to force the state and costate continuity at each interface. Considering intermediate times  $t_0 < t_1 < \dots < t_N < t_{N+1} = t_f$  which decompose  $[t_0, t_f]$  into  $N+1$  sub-intervals  $\Delta_i = [t_i, t_{i+1}]$ , the Two-Point Boundary Value Problem (TPBVP) is transformed into a Multi-Point Boundary Value Problem

$$(MPBVP) \quad \begin{cases} \exp_{\vec{H}}(t_{i+1}, t_i, z_i) = z_{i+1}, & \forall i \in \mathbb{N}_{N-1}, \\ g(z_0, \exp_{\vec{H}}(t_{N+1}, t_N, z_N)) = 0, \end{cases}$$

with the notation  $\mathbb{N}_k = \llbracket 0, k \rrbracket$ . The corresponding multiple shooting function  $S_m: (\mathbb{R}^{2n})^{N+1} \rightarrow (\mathbb{R}^{2n})^{N+1}$  is therefore

$$S_m(z_0, \dots, z_N) = \begin{pmatrix} \exp_{\vec{H}}(t_1, t_0, z_0) - z_1 \\ \vdots \\ \exp_{\vec{H}}(t_N, t_{N-1}, z_{N-1}) - z_N \\ g(z_0, \exp_{\vec{H}}(t_{N+1}, t_N, z_N)) \end{pmatrix}.$$

Thanks to the integration on smaller sub-intervals, the sensitivity with respect to the initial guess is reduced [4, 37]. Nevertheless, even multiple shooting methods need a good initialization to converge to an admissible solution of (OCP).

### 3 Bilevel optimal control method

The main contributions of this article are first the introduction of a new transcription path from (OCP) to (MPBVP), and then the development and the application of a novel method to solve optimal control problems. The new path is based on a classical bilevel formulation of (OCP) described in Section 3.1. The link between the necessary optimality conditions of this bilevel decomposition and the multiple shooting is presented in Section 3.2. Finally, this new point of view enables the development of a novel approach described in Section 3.3, which promises to be fast enough for embedded solutions.

#### 3.1 Bilevel formulation

In order to describe (OCP) in another manner, given the time intervals introduced in section (2.5), let us consider the following intermediate optimal control problems for all  $i \in \mathbb{N}_N$

$$(OCP_{i,a,b}) \quad \left\{ \begin{array}{l} V_i(a, b) = \min_{x,u} \int_{t_i}^{t_{i+1}} f^0(t, x(t), u(t)) dt, \\ \text{s.t. } \dot{x}(t) = f(t, x(t), u(t)), \quad t \in \Delta_i \text{ a.e.}, \\ u(t) \in U(t), \quad t \in \Delta_i, \\ x(t_i) = a, \quad x(t_{i+1}) = b. \end{array} \right.$$

The set of admissible initial and final states of (OCP<sub>i,·,·</sub>) is

$$\Omega_i = \{(a, b) \mid b \in \text{int}(A(t_{i+1}, t_i, a))\}, \quad (10)$$

where  $\text{int}(A)$  stands for the interior of the set  $A$ . The function  $V_i: \Omega_i \rightarrow \mathbb{R}$  corresponds to the optimal value functions of (OCP<sub>i,·,·</sub>). Denoting  $\mathcal{D}_i = \text{AC}(\Delta_i, \mathbb{R}^n) \times L^\infty(\Delta_i, \mathbb{R}^m)$ , the cost  $J_i: \mathcal{D}_i \rightarrow \mathbb{R}$  is defined by

$$J_i(x, u) = \int_{t_i}^{t_{i+1}} f^0(t, x(t), u(t)) dt. \quad (11)$$

For all  $(a, b) \in \Omega_i$ , the set of admissible points is

$$\mathcal{A}_i(a, b) = \left\{ (x, u) \in \mathcal{D}_i \left| \begin{array}{l} \dot{x}(t) = f(t, x(t), u(t)), \quad t \in \Delta_i \text{ a.e.}, \\ u(t) \in U(t), \quad t \in \Delta_i, \\ x(t_i) = a, \quad x(t_{i+1}) = b, \end{array} \right. \right\}$$

and the set of solutions is

$$\mathcal{S}_i(a, b) = \arg \min_{(x,u) \in \mathcal{A}_i(a,b)} J_i(x, u).$$

Given a vector  $X$  of admissible states at the intermediate times, we are lead to solve  $N+1$  elementary optimal control problems (OCP<sub>i, X<sub>i</sub>, X<sub>i+1</sub></sub>). The cost corresponding to this  $X$  is the sum of the optimal values of each (OCP<sub>i, X<sub>i</sub>, X<sub>i+1</sub></sub>) sub-problems. An equivalent manner to formulate (OCP) is then to find  $X$  that minimizes this cost. More precisely, we now have to solve

$$(BOCP) \quad \left\{ \begin{array}{l} \min_X V(X) := \sum_{i=0}^N V_i(X_i, X_{i+1}), \\ \text{s.t. } X \in \mathcal{X}, \quad c(X_0, X_{N+1}) = 0, \end{array} \right.$$

where the admissible set is given by

$$\mathcal{X} = \{(X_0, \dots, X_{N+1}) \mid \forall i \in \mathbb{N}_N, (X_i, X_{i+1}) \in \Omega_i\}. \quad (12)$$

Since for all  $i \in \mathbb{N}_N$  and for all  $(X_i, X_{i+1}) \in \Omega_i$ , all the solutions  $(x_i, u_i)$  lead to the same cost:

$$V_i(X_i, X_{i+1}) = J_i(x_i, u_i), \quad \forall (x_i, u_i) \in \mathcal{S}_i(X_i, X_{i+1}),$$

Problem (BOCP) can be rewritten as follows:

$$\begin{cases} \min_X \max_{x,u} \sum_{i=0}^N J_i(x_i, u_i) \\ \text{s.t. } X \in \mathcal{X}, \quad c(X_0, X_{N+1}) = 0, \\ \forall i \in \mathbb{N}_N, \quad (x_i, u_i) \in \mathcal{S}_i(X_i, X_{i+1}) \end{cases} \quad (13)$$

or equivalently

$$\begin{cases} \min_X \min_{x,u} \sum_{i=0}^N J_i(x_i, u_i) \\ \text{s.t. } X \in \mathcal{X}, \quad c(X_0, X_{N+1}) = 0, \\ \forall i \in \mathbb{N}_N, \quad (x_i, u_i) \in \mathcal{S}_i(X_i, X_{i+1}) \end{cases} \quad (14)$$

which are standard bilevel problems, in the pessimistic (13) and optimistic (14) forms. More precisely, these formulations can be seen as Single-Leader-Multi-Follower games [2] where the leader (or upper level) wants to find the best *intermediate state*  $X$  and the  $N + 1$  followers (or lower level) want to find a solution  $(x_i, u_i)$  of the associated optimal control problem (OCP $_{i,X_i,X_{i+1}}$ ).

### 3.2 Link with the multiple shooting

Now (OCP) has been reformulated as (BOCP), we show that the necessary optimality conditions of (BOCP) lead to (MPBVP). For this purpose, we introduce the following assumption which will stand from now.

**Assumption 1.** *The function  $V$  is differentiable at  $X$  solution of (BOCP).*

**Remark 4.** *Despite value functions are not necessarily differentiable in a general case, this assumption, which could appear quite strong, is numerically verified in our application. Moreover, it allows to remain in a simple framework.*

Thanks to the Karush-Kuhn-Tucker (KKT) necessary optimality conditions, if  $X$  is solution of (BOCP), then there exists  $\lambda \in \mathbb{R}^p$  such that

$$\text{(NCBOCP)} \quad \begin{cases} \nabla_X L(X, \lambda) = 0, \\ X \in \mathcal{X}, \quad c(X_0, X_{N+1}) = 0, \end{cases}$$

where the Lagrangian  $L: (\mathbb{R}^n)^{N+1} \times \mathbb{R}^p \rightarrow \mathbb{R}$  associated to (BOCP) is defined by

$$L(X, \lambda) = V(X) - (\lambda \mid c(X_0, X_{N+1})).$$

Using the expression of  $L$  and  $V$ , (NCBOCP) becomes

$$\begin{cases} \left( \begin{array}{c} \nabla_a V_0(X_0, X_1) \\ \nabla_b V_N(X_N, X_{N+1}) \end{array} \right) - c'(X_0, X_{N+1})^\top \lambda = 0, \\ \nabla_b V_{i-1}(X_{i-1}, X_i) + \nabla_a V_i(X_i, X_{i+1}) = 0, \quad \forall i \in \{1, \dots, N\}, \\ c(X_0, X_{N+1}) = 0, \quad X \in \mathcal{X}. \end{cases}$$

In order to make the costate appears in the above optimality conditions, we introduce the following theorem, the proof of which is given in appendix.



**Theorem 1.** Given  $(a, b) \in \Omega := \{(a, b) \mid b \in \text{int } A(t_f, t_0, a)\}$ , we consider a particular case  $(\text{OCP}_*)$  of  $(\text{OCP})$  in which  $c(x(t_0), x(t_f)) = (x(t_0) - a, x(t_f) - b)$  (initial and final conditions imposed):

$$(\text{OCP}_*) \quad \left\{ \begin{array}{l} V_*(a, b) = \min_{x, u} \int_{t_0}^{t_f} f^0(t, x(t), u(t)) \, dt, \\ \text{s.t. } \dot{x}(t) = f(t, x(t), u(t)) \quad t \in [t_0, t_f] \text{ a.e.}, \\ u(t) \in U(t), \quad t \in [t_0, t_f], \\ x(t_0) = a, \quad x(t_f) = b. \end{array} \right.$$

The value function  $V_*(a, b)$  of  $(\text{OCP}_*)$  is assumed to be differentiable at  $(a, b)$ . Then, if  $(x, u)$  is a solution of  $(\text{OCP}_*)$  with  $(x, p)$  the associated normal BC-extremal, we have:

$$\nabla V_*(x(t_0), x(t_f)) = (-p(t_0), p(t_f)).$$

*Proof.* See Appendix. □

Let now  $X$  be a solution of  $(\text{BOCP})$ . For each  $i \in \mathbb{N}_N$ , assuming  $\mathcal{S}_i(X_i, X_{i+1})$  non-empty, the optimal value  $V_i(X_i, X_{i+1})$  of  $(\text{OCP}_{i, X_i, X_{i+1}})$  comes with a solution  $(x_i, u_i)$  and its associated normal BC-extremal  $(x_i, p_i)$ . Since  $(\text{OCP}_{i, X_i, X_{i+1}})$  has the same form as  $(\text{OCP}_*)$ , we can apply Theorem 1 and we get

$$\nabla_a V_i(x_i(t_i), x_i(t_{i+1})) = -p_i(t_i), \quad \nabla_b V_i(x_i(t_i), x_i(t_{i+1})) = p_i(t_{i+1}).$$

Therefore  $(\text{NCBOCP})$  can be rewritten:

$$\left\{ \begin{array}{l} \forall i \in \mathbb{N}_N, \exists z_i = (x_i, p_i) \text{ a BC-extremal associated to a solution } (x_i, u_i) \text{ of } (\text{OCP}_{i, X_i, X_{i+1}}), \\ \left( \begin{array}{c} -p_0(t_0) \\ p_N(t_{N+1}) \end{array} \right) - c'(X_0, X_{N+1})^\top \lambda = 0, \\ \forall i \in \{1, \dots, N\}, p_{i-1}(t_i) - p_i(t_i) = 0, \\ c(X_0, X_{N+1}) = 0, \quad X \in \mathcal{X}, \end{array} \right.$$

and replacing  $(x_i, u_i)$  solution of  $(\text{OCP}_{i, X_i, X_{i+1}})$  by the associated necessary optimality conditions, we get

$$(\text{NCBOCP}) \implies \left\{ \begin{array}{l} \forall i \in \mathbb{N}_N, \exp_{\overline{H}}(t_{i+1}, t_i, z_i(t_i)) = z_i(t_{i+1}), \\ \forall i \in \mathbb{N}_N, x_i(t_i) - X_i = 0, \\ \forall i \in \mathbb{N}_N, x_i(t_{i+1}) - X_{i+1} = 0, \\ \left( \begin{array}{c} -p_0(t_0) \\ p_N(t_{N+1}) \end{array} \right) - c'(X_0, X_{N+1})^\top \lambda = 0, \\ \forall i \in \{1, \dots, N\}, p_{i-1}(t_i) - p_i(t_i) = 0, \\ c(X_0, X_{N+1}) = 0, \quad X \in \mathcal{X}, \end{array} \right\} \iff (\text{MPBVP}).$$

The above leads us to the following proposition.

**Proposition 2.** Under our general assumptions, we have  $(\text{NCBOCP}) \implies (\text{MPBVP})$ .

As already seen in Sections 2.4 and 2.5, the initial problem  $(\text{OCP})$  is linked to  $(\text{MPBVP})$  through  $(\text{TPBVP})$ . A new path is now highlighted by Proposition 2 via  $(\text{BOCP})$  and  $(\text{NCBOCP})$ . In order to explicit the various connections between all the above formulations, we define the following operators:

$$\begin{array}{ccc} (\text{OCP}) & \xrightarrow{\text{Dualization}} & (\text{TPBVP}), \\ (\text{TPBVP}) & \xrightarrow{\text{Splitting}} & (\text{MPBVP}), \\ (\text{OCP}) & \xrightarrow{\text{Decomposition}} & (\text{BOCP}), \\ (\text{BOCP}) & \xrightarrow{\text{Dualization}} & (\text{NCBOCP}). \end{array}$$

More precisely,

- The *Dualization* operator consists in applying necessary optimality conditions: PMP for (OCP) and KKT conditions for (BOCP).
- The *Splitting* operator transforms (TPBVP) into (MPBVP) via the standard point of view described in Section 2.5.
- The *Decomposition* operator transforms (OCP) into (BOCP) and is described in Section 3.1.

The connections defined above are summarized in Figure 1 which is a commutative diagram. It can be interpreted in terms of permutation of Dualization and Splitting/Decomposition operators.

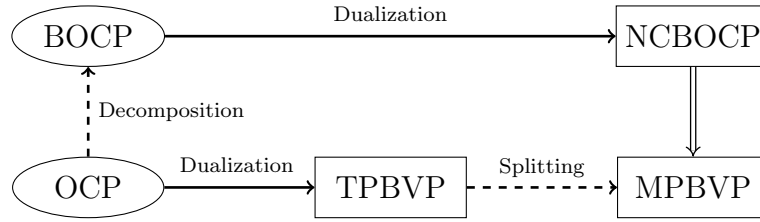


Figure 1: Commutative diagram between (OCP) and (MPBVP). The solid arrows correspond to the Dualization operator and the dashed arrows correspond to the Decomposition or the Splitting one. Finally, the double arrow is the classical implication operator.

### 3.3 A novel approach based on value functions approximations

As indicated on Figure 1, a new path is proposed to transform the Optimal Control Problem (OCP) into the Multi-Point Boundary Value Problem (MPBVP) using the bilevel decomposition. This path leads to a novel approach and a new method presented in this Section.

#### 3.3.1 Main idea

Let us suppose that the value functions  $V_i$  are known for all  $i \in \mathbb{N}_N$ . The resolution of the optimal control problem (OCP) can be decomposed into two main steps:

- the first one is to solve the low-dimensional optimization problem (BOCP) using classical methods to get the optimal intermediate states  $X^*$ ,
- the second one is to solve the  $N + 1$  independent optimal control problems  $(\text{OCP}_{i, X_i^*, X_{i+1}^*})$  for all  $i \in \mathbb{N}_N$ . Each of these problems is defined on a smaller time interval and could be solved by the indirect simple shooting method described in Section 2.4. Nevertheless, thanks to Theorem 1, for all  $i \in \mathbb{N}_N$ , the pair  $(X_i^*, -\nabla_a V_i(X_i^*, X_{i+1}^*))$  is a zero of the associated simple shooting function, and so it is no more necessary to use the shooting method. Finally, we have only to integrate  $N + 1$  Hamiltonian vector fields.

As a conclusion, the numerical resolution of (OCP) is reduced to the resolution of a low-dimensional optimization problem, and the integration of  $N + 1$  Hamiltonian vector fields, which are independent and can therefore be processed in parallel.

### 3.3.2 Macro and Micro problems

The crucial assumption above is the *a priori* knowledge of the value functions  $V_i$  for all  $i \in \mathbb{N}_N$ . Since it is rarely satisfied, we propose to replace the  $V_i$  by approximations  $C_i$ . Problem (BOCP) is then transformed into

$$\text{(Macro)} \quad \begin{cases} \min_X C(X) := \sum_{i=0}^N C_i(X_i, X_{i+1}), \\ \text{s.t. } X \in \mathcal{X}, \quad c(X_0, X_{N+1}) = 0, \end{cases}$$

to get the intermediate state  $\hat{X}$ , and then to solve the  $N + 1$  following optimal control problems

$$\text{(Micro)} \quad \begin{cases} V_i(\hat{X}_i, \hat{X}_{i+1}) = \min_{x,u} \int_{t_i}^{t_{i+1}} f^0(t, x(t), u(t)) \, dt, \\ \text{s.t. } \dot{x}(t) = f(t, x(t), u(t)), & t \in \Delta_i \text{ a.e.}, \\ u(t) \in U(t), & t \in \Delta_i, \\ x(t_i) = \hat{X}_i, \quad x(t_{i+1}) = \hat{X}_{i+1}. \end{cases}$$

The pair

$$(\hat{X}_i, -\nabla_a C_i(\hat{X}_i, \hat{X}_{i+1})) \tag{15}$$

is no more necessarily a zero of the simple shooting function associated to the corresponding (Micro) problem, and therefore a classical Newton-like solver has to be used. However, the pair (15) provides a natural initial guess. A similar approach has been developed in [15], where a rough approximation of the value function, obtained by a Hamilton-Jacobi-Bellman method, is used to provide an initial guess for the indirect shooting method.

Numerically, the (Macro) and (Micro) resolution can be faster and computationally more efficient than the simple or multiple shooting methods. The counterpart of this benefit is a large computation time induced by the construction of the value functions approximations.

A similar (Macro)–(Micro) approach has been developed in [25] for stochastic optimal control problem with dynamic programming method on predefined traffic conditions.

### 3.3.3 Advantages

Most of the time computation for the resolution of the (Macro)–(Micro) problem is due to the (Micro) part. In comparison with the simple shooting method,  $N + 1$  optimal control problems have to be solved but on  $N + 1$  times shorter intervals. Thanks to parallel computing, the time computation for (Micro) resolution is divided by a factor  $N + 1$ . Moreover, the proposed initialization (cf. Equation (15)) reduces the number of iterations of the Newton-like solver (see Figure 6), which further decreases the computation time.

As proved in Section 3.1, the (Macro)–(Micro) method is strongly linked to the multiple shooting one. Both methods take advantage of shorter time intervals and therefore show a reduced sensitivity to initial guess for the shooting function. They also can benefit from parallel computing but with a major difference: (Micro) problems are independent from each other and so can be solved in parallel whereas in multiple shooting method, although the Hamiltonian flows on each interval can be parallelized, they are coupled through the matching conditions of the shooting function  $S_m$ . For an embedded solution, where the goal is to get the current control in real time, the (Macro)–(Micro) has a remarkable advantage: only the (Micro) problem on the first sub-interval has to be solved. Thus, the number of computations for the (Macro)–(Micro) method is reduced by a factor  $N + 1$  compared to indirect shooting.

### 3.3.4 Error analysis

Due to the approximation of the value function, the (Macro)–(Micro) method is sub-optimal and its cost  $V(\hat{X})$  is likely to be greater than the optimal one  $V(X^*)$ . Given a prescribed error  $e$  between  $V(X^*)$  and  $V(\hat{X})$ , our purpose here is to provide a bound on the error between  $C_i$  and  $V_i$ , for all  $i \in \mathbb{N}_N$ . We consider that for all  $i \in \mathbb{N}_N$ , there exists  $\varepsilon_i$  such that for all  $(a, b) \in \Omega_i$ ,  $|V_i(a, b) - C_i(a, b)| \leq \varepsilon_i$ . Therefore, for all  $X \in \mathcal{X}$

$$|V(X) - C(X)| \leq \sum_{i=0}^N \varepsilon_i \leq (N+1) \max_{i \in \mathbb{N}_N} \varepsilon_i.$$

Denoting  $\alpha := (N+1) \max_{i \in \mathbb{N}_N} \varepsilon_i$ , we obtain equivalently

$$V(X) - \alpha \leq C(X) \leq V(X) + \alpha. \quad (16)$$

Since  $X^*$  is a global solution of (BOCP) and using (16) with  $X = \hat{X}$ , we have  $V(X^*) - \alpha \leq V(\hat{X}) - \alpha \leq C(\hat{X}) \leq V(\hat{X}) + \alpha$ , and therefore  $V(X^*) - V(\hat{X}) \leq 2\alpha$ . Since  $\hat{X}$  is a global solution of (Macro), using left part of (16) with  $X = \hat{X}$  and right part with  $X = X^*$ , we have  $V(\hat{X}) - \alpha \leq C(\hat{X}) \leq C(X^*) \leq V(X^*) + \alpha$ , and therefore  $-2\alpha \leq V(X^*) - V(\hat{X})$ . Finally we have proven that

$$\left| V(\hat{X}) - V(X^*) \right| \leq 2\alpha. \quad (17)$$

The maximum gap  $e$  between  $V(X^*)$  and  $V(\hat{X})$  being given, (17) provides a bound on the models error :  $\max_{i \in \mathbb{N}_N} \varepsilon_i \leq \frac{e}{2(N+1)}$ . From a practical point of view, building  $C_i$  models of  $V_i$  such that for all  $(a, b) \in \Omega_i$

$$|V_i(a, b) - C_i(a, b)| \leq \frac{e}{2(N+1)}$$

will ensure that the cost difference between the (BOCP) solution and the (Macro)–(Micro) one is less than  $e$ .

### 3.3.5 Extremal database construction

The (Macro)–(Micro) method requires the approximation of the value function. These approximations  $C_i$  are fitted among a given class of parameterized models on a database and the objective of this section is to propose a numerical method to generate such a database.

Let  $i \in \mathbb{N}_N$ . Our approximation of the value function  $V_i$  relies on a database of optimal values:

$$\mathbb{D}_i \subset \{(a, b, c) \mid (a, b) \in \Omega_i, V_i(a, b) = c\}$$

which needs to be created. For that purpose, the natural method consists in evaluating  $V_i$  on various admissible pairs  $(a, b)$ . We propose here an alternative method. Let us define

$$\bar{x}_{i+1}(x_0, p_0) = \pi_x(\exp_{\bar{H}}(t_{i+1}, t_i, (x_0, p_0))) \quad \text{and} \quad \Psi_i = \left\{ (x_0, p_0) \in \mathbb{R}^{2n} \mid (x_0, \bar{x}_{i+1}(x_0, p_0)) \in \Omega_i \right\},$$

and introduce the following hypothesis that ensures the existence and uniqueness of a (normal) BC-extremal.

**Hypothesis 1.** *For all  $i \in \mathbb{N}_N$  and for all  $x_0 \in \mathbb{R}^n$ , the function  $p_0 \mapsto \bar{x}_{i+1}(x_0, p_0)$  is injective on the set of all  $p_0$  such that  $(x_0, p_0) \in \Psi_i$ .*

Under the general assumptions (see Section 2.3) and Hypothesis 1, for all initial  $(x_0, p_0)$  and final state and costate pair  $(x_f, p_f) = \exp_{\bar{H}}(t_{i+1}, t_i, (x_0, p_0))$ , we have

$$V_i(x_0, x_f) = c_i(x_0, p_0), \quad (18)$$

where  $c_i(x_0, p_0)$  is the cost of the trajectory on the time interval  $\Delta_i$  starting from  $(x_0, p_0)$  at time  $t_i$ . More precisely, the computation of  $c_i(x_0, p_0)$  consists simply in evaluating the integral  $J_i$  defined by equation (11) along the pair of state and control trajectories, the control being given by the maximization condition.

**Remark 5.** *Hypothesis 1 is motivated by the application, Equation (18) allows to build the database  $\mathbb{D}_i$  without solving optimal control problems, but simply by integrating the Hamiltonian flow.*

Finally, considering an initial state and costate discretization  $\mathbb{Z}_i$ , we propose to create the database  $\mathbb{D}_i$  by

$$\mathbb{D}_i = \{(x_0, x_f, c) \mid (x_0, p_0) \in \mathbb{Z}_i, (x_f, p_f) = \exp_{\vec{H}}(t_{i+1}, t_i, (x_0, p_0)), c = c_i(x_0, p_0)\}.$$

For further information, the pros and cons of this method compared to those of the natural one are detailed in [14].

## 4 Case study

The main objective of this section is to compare our bilevel approach to the simple shooting method and an empiric one on an industrial application. First, the application to be considered is presented in Section 4.1, then the numerical methods are precised in Section 4.2 and some numerical results are shown in Section 4.3. All the variables and parameters mentioned in Section 4.1 are described in Table 3 in Appendix.

### 4.1 Hybrid electric vehicle torque split and gear shift problem

The selected vehicle is a light Hybrid Electric Vehicle (HEV) with a 400V asynchronous Electrical Machine (EM), a 1.5L Internal Combustion Engine (ICE) and a battery of 1.5 kWh. The considered optimal control problem is the torque split and gear shift determination, with a fixed initial and final state of charge of the battery. The strategy for choosing the initial  $SOC_0$  and the final  $SOC_f$  states of charge of the battery is out of the topic of this paper. The architecture of our vehicle is a parallel P3 with a double drive shaft, as presented on Figure 2. Other HEV's architectures exist, and are listed in [16]. This vehicle has a 4 gears transmission with a claw clutch. Note that in our case, clutch and gear are combined in a unique control called *Gear*: value 0 corresponds to open clutch and ICE off, while values 1 to 4 are associated to the corresponding gears.

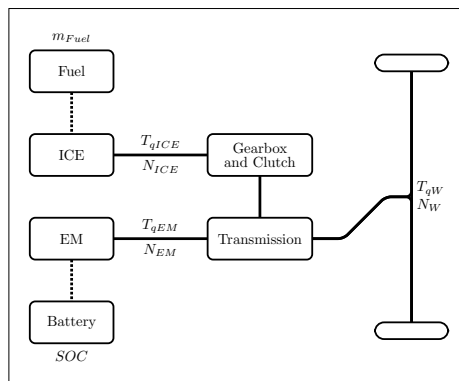


Figure 2: Scheme of the selected HEV. The solid lines represent mechanical connections and the dashed lines represent energy connections, using fuel for ICE and electricity for EM.

#### 4.1.1 General formulation

As stated before, the goal in the considered industrial application is to minimize the fuel consumption of a HEV while taking into account some physical constraints from ICE, EM and battery. The problem can be written as follows:

$$(P) \quad \begin{cases} \min_{SOC, T_{qICE}, Gear} \int_{t_0}^{t_f} f_{m_F}(t, T_{qICE}(t), Gear(t)) dt, \\ \text{s.t. } \dot{SOC}(t) = f_{SOC}(t, SOC(t), T_{qICE}(t), Gear(t)), & t \in [t_0, t_f] \text{ a.e.}, \\ (T_{qICE}(t), Gear(t)) \in U(t), & t \in [t_0, t_f], \\ SOC(t_0) = SOC_0, \quad SOC(t_f) = SOC_f. \end{cases}$$

This problem is solved on a predefined cycle, and we choose the WLTC, for which the slope is null and the speed evolution is presented on Figure 3. Using the vehicle model (mass, wheel diameter, aerodynamic coefficient...), the cycle information is decomposed into requested torque  $T_{qW}$  and wheel speed rotation  $N_W$ . The two functions  $f_{m_F}$  and  $f_{SOC}$  result from the static model described in the following section.

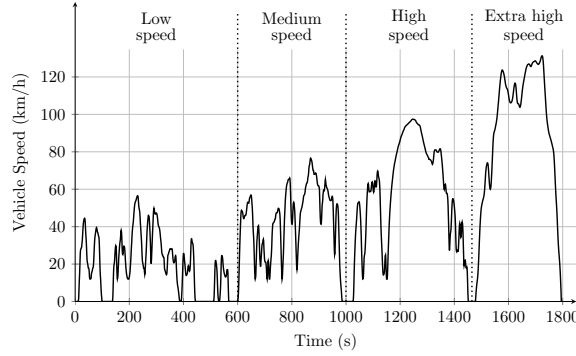


Figure 3: Speed evolution for the Worldwide harmonized Light vehicles Test Cycle (WLTC). This cycle is mainly used for emission and fuel consumption test.

#### 4.1.2 System modelling

The state of charge and fuel dynamics are given by the models of vehicle equipment described below. First, the speed and torque relations are given by the following transmission relations:

$$\begin{bmatrix} N_{ICE} \\ N_{EM} \end{bmatrix} = \begin{bmatrix} P & R(Gear) \\ & P_e \end{bmatrix} N_W, \quad T_{qW} = P_e (T_{qEM} - T_{qEM_L}) + P (R(Gear) T_{qICE} - T_{qICE_L}),$$

where  $T_{qICE_L}$  (respectively  $T_{qEM_L}$ ) represents the ICE (respectively EM) torque losses, given by a map depending on ICE torque  $T_{qICE}$  and ICE rotation speed  $N_{ICE}$  (respectively  $T_{qEM}$  and  $N_{EM}$ ). The function  $R: \mathbb{R} \rightarrow \mathbb{R}$  is smooth and corresponds to the ratio of the gearbox. The parameter  $P$  (respectively  $P_e$ ) is the transmission ratio between the main axis and the ICE axis (respectively the EM one).

Then, the state of charge dynamics is calculated by the equivalent RC circuit where the battery terminal voltage is considered constant:

$$\dot{SOC} = \frac{I}{Q},$$

with  $Q$  the battery charge and  $I$  the current, computable using a map depending on the state of charge  $SOC$  and the power demand  $P_{elec}$ . This requested power is the sum of the average additional power of other vehicle equipment and the requested power of the EM. The latter is known through a map depending on the EM rotation speed  $N_{EM}$  and torque  $T_{qEM}$ .

Finally, the fuel consumption of the ICE  $\dot{m}_F$  is given through a map depending on ICE torque  $T_{qICE}$ , rotation speed  $N_{ICE}$  and temperature of the coolant  $T$ . In this study, we assume that the ICE is already warm, and thus that the temperature is constant and equal to  $T = 90^\circ C$ . The other constraints of (P) come from physical limitations. The ICE and EM rotation speeds and torques have min-max constraints and  $Gear \in \llbracket 0, 4 \rrbracket$ . These constraints are seen as a control admissible domain  $(T_{qICE}, Gear) \in U$ . The state of charge has usually box constraints  $SOC \in [SOC_{\min}, SOC_{\max}]$ . Taking into account state constraints with the Pontryagin Maximum Principle is complex [7] but could be handled, by penalization for instance. However, this is not the purpose of this article.

This kind of HEV's modelling can be found in the literature [18, 21, 29]. The dynamics of our vehicle is given by an industrial code, developed in Matlab Simulink by Vitesco Technologies, and we consider that all maps are smooth functions. The torque split and gear shift problem (P) has the same formulation as the general optimal control problem (OCP) defined in Section 2.1. Therefore, it can be numerically solved by the methods presented above.

## 4.2 Application of Macro and Micro method to the industrial problem

The methods to be tested require a time discretization and a numerical integration of the Hamiltonian flow. In this industrial application, the selected cycle (WLTC) lasts 1800 seconds. We arbitrarily choose to decompose this time interval  $\Delta = [t_0, t_f] = [0, 1800]$  into  $N + 1 = 18$  sub-intervals of 100 seconds:

$$\forall i \in \mathbb{N}_N = \{0, \dots, 17\}, \quad \Delta_i = [t_i, t_{i+1}] = [100 i, 100 (i + 1)].$$

Due to the complexity of the model, the maximized Hamiltonian (8) cannot be computed analytically. The admissible control domain  $U(t)$  is thus discretized and the maximizing control  $u^*(t, z)$  is extracted from the resulting finite set  $\tilde{U}(t)$ , *i.e.*

$$u^*(t, z) \in \arg \max_{u \in \tilde{U}(t)} \{h(t, z, u)\},$$

where  $z = (x, p)$ . This computation is vectorized to make it faster. The Hamiltonian vector field is calculated as follows

$$\vec{H}(t, z) = \begin{pmatrix} \frac{\partial h}{\partial p}(t, z, u^*(t, z)) \\ -\frac{\partial h}{\partial x}(t, z, u^*(t, z)) \end{pmatrix} = \begin{pmatrix} f(t, x, u^*(t, z)) \\ -\frac{\partial h}{\partial x}(t, z, u^*(t, z)) \end{pmatrix}$$

and the partial derivative of the pseudo-Hamiltonian with respect to  $x$  is approximated by finite differences method. The Hamiltonian flow is integrated using an explicit Euler method from Matlab Simulink. Based on these elements, we first construct the value function approximations  $C_i$ , which we then use to solve (Macro)-(Micro).

### 4.2.1 Construction of value functions approximations

The value functions  $V_i$  are approximated by  $C_i$  following the process described in Section 3.3.5. We first observe on Figure 4 that for a given  $x_0$  the function  $p_0 \mapsto \bar{x}_1(t_1, t_0, (x_0, p_0))$  is injective on the set of all  $p_0$  such that  $(x_0, p_0) \in \Psi_0$ . This injectivity property remains true for all the other time intervals and initial states. Thus, in our application, Hypothesis 1 is considered numerically valid.

For all  $i \in \mathbb{N}_N$ , the database of optimal values  $\mathbb{D}_i$  associated to the time interval  $\Delta_i$  is created by computing the exponential map of the Hamiltonian flow for each  $(x_0, p_0)$  on a fixed grid. This grid

is created by a uniform discretization between  $SOC_{\min}$  and  $SOC_{\max}$  (respectively  $p_{\min}$  and  $p_{\max}$ ) for the state (respectively costate). The bounds  $p_{\min}$  and  $p_{\max}$  are chosen such that the interval of final admissible states is fully explored for all initial states. Moreover, the number of points of each discretization is chosen large enough to capture the variations of the value function. The Figure 5 shows the transformation of the initial state and costate domain by the exponential map on the first time interval  $\Delta_0$ .

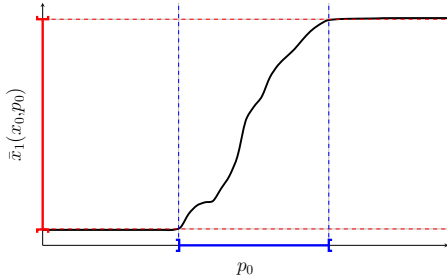


Figure 4: Evolution of the function  $\bar{x}_1(x_0, \cdot)$ . The red interval on the y-axis corresponds to the admissible state domain  $A(t_1, t_0, x_0)$  and the blue interval on the x-axis corresponds to the section at  $x_0$  of the domain  $\Psi_0$ .

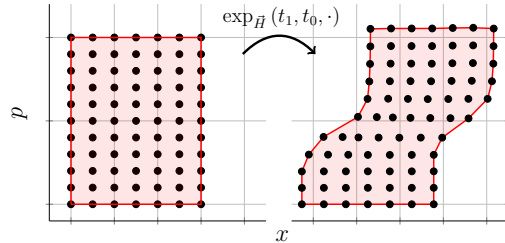


Figure 5: Transport of the initial state and costate space by the exponential map  $(x_0, p_0) \mapsto \exp_{\bar{H}}(t_1, t_0, (x_0, p_0))$ . The black dots correspond to some points in our database and red areas correspond to the state and costate space, at time  $t_0$  for the left plot and  $t_1$  for the right plot.

The cost transition functions  $C_i$  are modeled by simple smooth neural networks described in Table 1. These networks are trained on  $\mathbb{D}_i$  using the Tensorflow package in Python.

Layer Name	Activation function	Number of units	Number of parameters
Input $(x_0, x_f)$	id	2	0
Hidden Layer 1	tanh	16	48
Hidden Layer 2	sigmoid	8	136
Output $(c)$	id	1	8
Total			192

Table 1: Description of neural networks used to model functions  $C_i$ . They are trained using Adam optimizer with a learning rate of 0.01 on 4000 epochs and a 16 batch size. The database  $\mathbb{D}_i$  is randomly split into train and test databases with a (80% / 20%) partition. The selected weights minimize the loss on the test database.

#### 4.2.2 Resolution of Macro and Micro problems

The resolution of (Macro) problem requires the determination of the intermediate admissible state domain  $\mathcal{X}$  based on the accessibility sets through Equations (12) and (10). Assuming that the minimal and maximal gaps of  $x_f - x_0$  between a given initial state  $x_0$  and all the final states  $x_f$  reachable from  $x_0$  are nearly independent from  $x_0$ , the accessibility sets  $A(t_{i+1}, t_i, x_0)$  can be approximated for all  $i \in \mathbb{N}_N$  by  $[x_0 - \delta_i^-, x_0 + \delta_i^+]$ , where  $\delta_i^-$  and  $\delta_i^+$  are independent from  $x_0$ . In order to ensure that the approximation belongs to the actual bounds and so  $[x_0 - \delta_i^-, x_0 + \delta_i^+] \subset$



$A(t_{i+1}, t_i, x_0)$ , we set

$$\delta_i^- = \min_{x_0 \in \mathbb{D}_i^-} \left\{ \max_{x_f \in \mathbb{D}_i^+(x_0)} \{x_0 - x_f\} \right\}, \quad \delta_i^+ = \min_{x_0 \in \mathbb{D}_i^-} \left\{ \max_{x_f \in \mathbb{D}_i^+(x_0)} \{x_f - x_0\} \right\},$$

with  $\mathbb{D}_i^- = \{x_0 \mid (x_0, \cdot, \cdot) \in \mathbb{D}_i\}$  and  $\mathbb{D}_i^+(x_0) = \{x_f \mid (x_0, x_f, \cdot) \in \mathbb{D}_i\}$  the initial and final state data in  $\mathbb{D}_i$ .

The (Macro) problem is numerically solved with the trust-ncg algorithm from the Scipy package. The gradient of the cost transition functions  $C_i$  is provided to the solver, taking advantage of the ability of TensorFlow package to compute it. The constraint  $X \in \mathcal{X}$  is taken into account through a penalization method. Due to the complexity of the industrial problem, the (Macro) optimization problem is likely to be non-convex, and thus a parallel multistart strategy is adopted to increase the robustness of our method.

The (Micro) problems are solved by indirect simple shooting using the trust region dogleg method of the `fsolve` Matlab function to find for all  $i \in \mathbb{N}_N$  a zero of

$$S_i(p) = \pi_x(\exp_{\vec{H}}(t_{i+1}, t_i, (X_i, p))) - X_{i+1}.$$

Since the (Micro) problems are independent from each other, they are solved in parallel.

### 4.3 Numerical experiments

A first objective of the numerical experiments is to evaluate the benefits of the initialization suggested by Equation (15). We observe on Figure 6 that the convergence of the (Micro) resolution is significantly improved by this choice compared to a fixed arbitrary initialization.

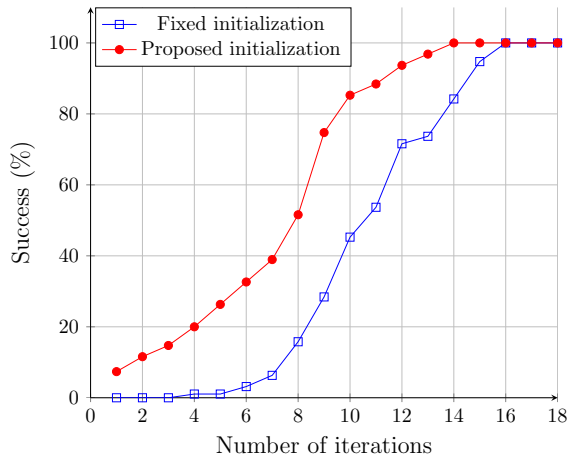


Figure 6: Evolution of the percentage of success of  $S_0 = 0$  resolution according to the number of iterations of the solver. The blue line with squares represents this evolution for a fixed initialization and the red line with disks for the initialization suggested by Equation (15). These data result from 300 shooting functions with random initial and final states. The resolution is considered successful when the norm of the shooting function is smaller than a given threshold.

The second objective is to compare the bilevel method to two other methods. The first other method is the simple shooting described in Section 2.4, used as reference. The second other method (called *empiric*) aims to evaluate the relevance of the (Macro) problem. For this purpose, an empirical rule is created to choose the vector  $X = (X_0, \dots, X_{N+1})$  of intermediate states. More precisely,  $\forall i = 1, \dots, N$ , each intermediate state  $X_i$  is chosen using a linear interpolation at the intermediate time  $t_i$  between  $X_0 = x_0$  at time  $t_0$  and  $X_{N+1} = x_f$  at time  $t_f$ . If the proposed vector is not

admissible, *i.e.*  $X \notin \mathcal{X}$ ,  $X$  is projected on  $\mathcal{X}$ . Then, (Micro) problems are solved with the same simple shooting method as for the bilevel one. Figure 7 compares the state trajectories obtained with the simple shooting, the empiric and the bilevel methods for 5 different initial and final states.

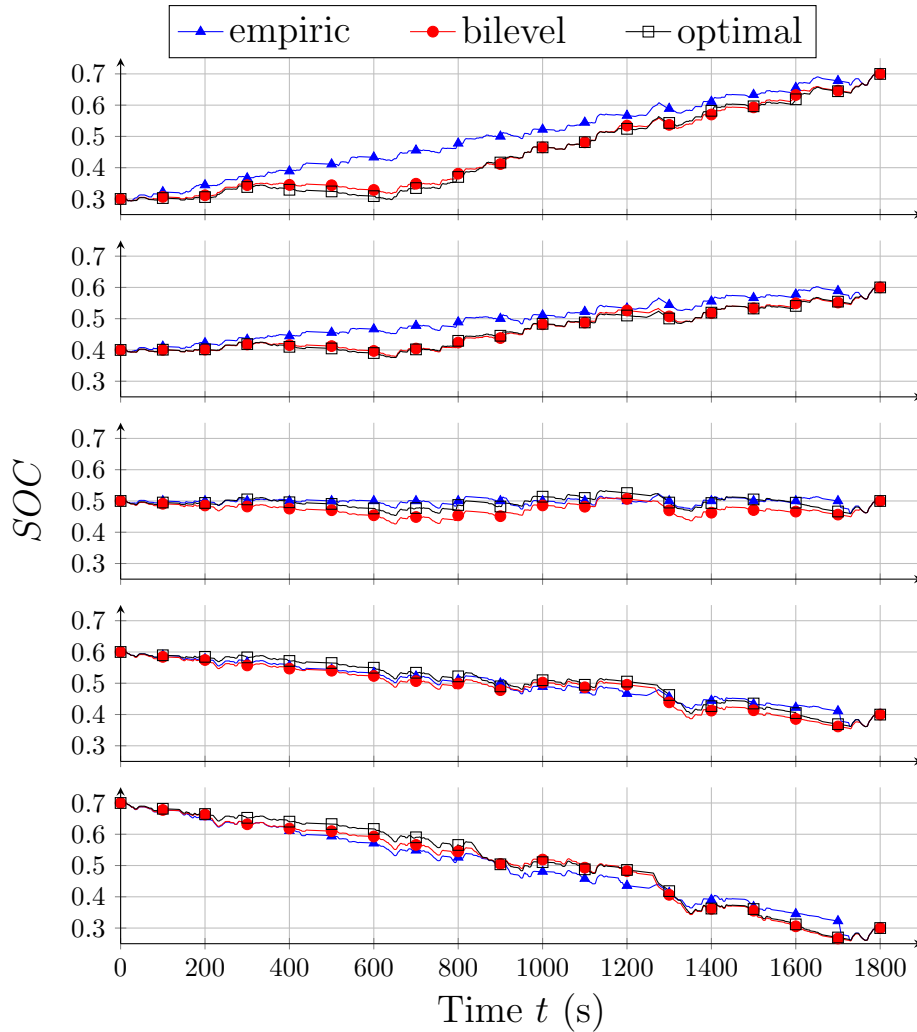


Figure 7: State trajectories for various initial  $SOC_0$  and final  $SOC_f$  states. The black line with squares corresponds to the optimal state trajectory, calculated by simple shooting method. Red line with disks corresponds to the bilevel method and blue line with triangles to the empiric one. The red and blue markers are the intermediate points, respectively for bilevel and empiric methods

Table 2 compares the bilevel and the empiric solutions to the simple shooting one in terms of difference of cost, and in terms of deviation of the state trajectories in  $L^2$  norm. It appears that the fuel consumption obtained by the bilevel approach is much closer to the simple shooting one than with the empiric method. Furthermore, the average bilevel error is more than 9 times smaller than the empiric one in terms of fuel consumption.

$SOC_0$	$SOC_f$	bilevel cost error (g / %)	empiric cost error (g / %)	bilevel state deviation $\ \cdot\ _{L^2}$ (· / %)	empiric state deviation $\ \cdot\ _{L^2}$ (· / %)
0.3	0.7	<b>0.42 / 0.045</b>	6.65/0.72	<b>0.42 / 0.05</b>	2.80 / 2.05
0.4	0.6	<b>0.34 / 0.039</b>	7.35/0.89	<b>0.22 / 0.01</b>	1.77 / 0.80
0.5	0.5	<b>1.00 / 0.129</b>	9.17/1.18	1.00 / 0.23	<b>0.61 / 0.08</b>
0.6	0.4	<b>1.71 / 0.244</b>	8.53/1.21	<b>0.75 / 0.12</b>	0.76 / 0.13
0.7	0.3	<b>0.91 / 0.143</b>	10.19/1.60	<b>0.58 / 0.07</b>	1.29 / 0.33
Average		<b>0.88 / 0.120</b>	8.37/1.12	<b>0.59 / 0.1</b>	1.45 / 0.68

Table 2: Comparison of the bilevel and empiric methods for the 5 experiments defined by the first two columns. The bilevel and empiric cost error columns give the difference between their respective costs and the simple shooting reference. Similarly, the last two columns compare the deviation of the state trajectories in numerical  $L^2$  norm.

## 5 Conclusion

A novel resolution method for optimal control problems (OCP), called (Macro)–(Micro), has been presented. This method, based on a bilevel formulation of (OCP), is strongly linked to the multiple shooting method, thanks to a new path between (OCP) and the Multi-Point Boundary Value Problem. This method has been applied to a complex industrial problem: the HEVs torque split and gear shift determination. Its main characteristics are the following:

- small state dimension,
- strong dependency on the non-autonomous part,
- high frequency command compared to state (cf. Figure 8),
- long integration time compared to integration time step.

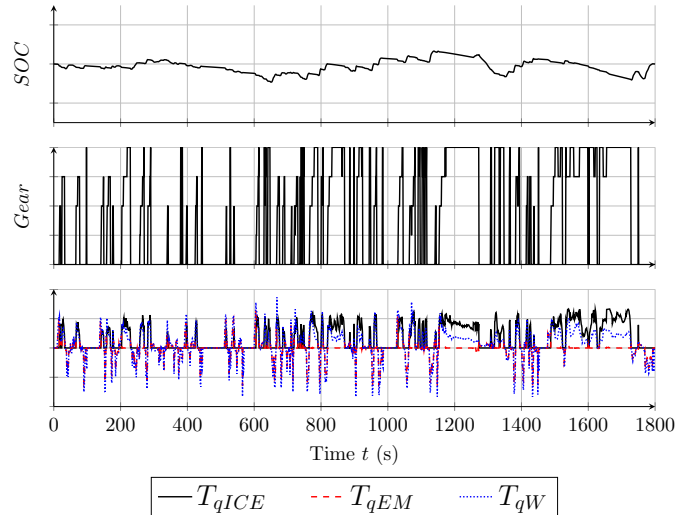


Figure 8: Simple shooting solution (state ( $SOC$ ) and command ( $Gear$ ,  $T_{qICE}$ )) with  $SOC_0 = SOC_f = 0.5$  on the WLTC.

The (Macro)–(Micro) method has been applied to solve this industrial problem. An initialization of the shooting function based on theoretical results has been used and an improvement with respect to standard initialization has been exhibited in terms of convergence rate.

As discussed in Section 3.3.3, this method is particularly interesting in terms of time computation and ability to be embedded, while keeping a satisfactory accuracy on fuel consumption compared to the optimal solution.

### Perspectives.

The industrial problem considered here has been deliberately simplified: ICE is supposed to be warm. Actually, the fuel consumption is dependent on the ICE temperature and an additional state variable, the coolant temperature  $T$ , should be considered. Problem (P) then becomes a hybrid optimal control problem, with a transient regime ( $T \leq 90^\circ C$ ) and a steady state ( $T = 90^\circ C$ ).

Our novel (Macro)–(Micro) method could be extended to consider any cycle (not only the WLTC), using a unique neural network with time series as additional inputs. Such neural networks exist (1D convolutional networks for instance). The counterpart of this generalization is obviously the need of a huge amount of data (value function samples on many sub-cycles) and a long computation time required to training such networks.

Finally, it could be really interesting to create a benchmark to compare the bilevel method with the classical ones (simple/multiple shooting, direct methods, dynamic programming...) on various optimal control problems. A classification of these resolution methods with respect to the main characteristics of the problems (horizon time, autonomous or not, command frequency...) could then be carried out. A class of optimal control problems where the bilevel method provides an advantage in terms of a compromise between computation time, accuracy and computing power compared to other classical optimal control methods could thus be revealed.

**Acknowledgments** This research was partially supported by Vitesco Technologies. In particular, we want to thank Olivier Flebus, artificial intelligence and optimization group leader at Vitesco Technologies and Mariano Sans, optimal control senior expert at Vitesco Technologies, for their supervision, valuable insights and suggestions.

## References

- [1] A. A. AGRACHEV AND Y. L. SACHKOV, *Control Theory from the Geometric Viewpoint*, Springer Berlin Heidelberg, 2004.
- [2] D. AUSSEL AND A. SVENSSON, *A Short State of the Art on Multi-Leader-Follower Games*, in *Bilevel Optimization: Advances and Next Challenges*, Springer, 2020, ch. 3, pp. 53–76.
- [3] R. BELLMAN, *On the Theory of Dynamic Programming*, Proc. Natl. Acad. Sci., 38 (1952), pp. 716–719.
- [4] H. BOCK AND K. PLITT, *A Multiple Shooting Algorithm for Direct Solution of Optimal Control Problems*, IFAC Proc. Vol., 17 (1984), pp. 1603–1608.
- [5] T. J. BÖHME AND B. FRANK, *Indirect Methods for Optimal Control*, in *Hybrid Systems, Optimal Control and Hybrid Vehicles: Theory, Methods and Applications*, Springer International Publishing, 2017, ch. 7, pp. 215–231.
- [6] O. BOKANOWSKI, A. DÉSILLES, AND H. ZIDANI, *Relationship between maximum principle and dynamic programming in presence of intermediate and final state constraints*, ESAIM - Control Optim. Calc. Var., 27 (2021), p. 91.
- [7] L. BOURDIN, *Note on Pontryagin maximum principle with running state constraints and smooth dynamics – Proof based on the Ekeland variational principle*, 2016.

- [8] A. E. BRYSON AND Y. C. HO, *Applied Optimal Control*, Taylor and Francis Group, 1975.
- [9] K.-H. N. BUI, J. CHO, AND H. YI, *Spatial-temporal graph neural network for traffic forecasting: An overview and open research issues*, *Appl. Intell.*, (2021), pp. 1–12.
- [10] A. CERNEA AND H. FRANKOWSKA, *A Connection Between the Maximum Principle and Dynamic Programming for Constrained Control Problems*, *SIAM J. Control Optim.*, 44 (2005), pp. 673–703.
- [11] L. CESARI, *Statement of the Necessary Condition for Mayer Problems of Optimal Control*, in *Optimization—Theory and Applications: Problems with Ordinary Differential Equations*, Springer New York, 1983, ch. 4, pp. 159–195.
- [12] F. CLARKE, *Functional Analysis, Calculus of Variations and Optimal Control*, Springer London, 2013.
- [13] F. H. CLARKE AND R. B. VINTER, *The Relationship between the Maximum Principle and Dynamic Programming*, *SIAM J. Control Optim.*, 25 (1987), pp. 1291–1311.
- [14] O. COTS, R. DUTTO, S. LAPORTE, AND S. JAN, *Generation of value function data for bilevel optimal control method*, 2023.
- [15] E. CRISTIANI AND P. MARTINON, *Initialization of the Shooting Method via the Hamilton-Jacobi-Bellman Approach*, *J. Optim. Theory Appl.*, 146 (2010), pp. 321–346.
- [16] S. DELPRAT, *Evaluation de stratégies de commande pour véhicules hybrides parallèles*, PhD thesis, Université de Valenciennes et du Hainaut-Cambresis, 2002.
- [17] I. H. A. HAMZA, *Commande Prédictive optimale temps-réel, appliquée au contrôle de véhicules automobiles hybrides connectés à leurs environnements*, PhD thesis, INPT, 2018.
- [18] M. HEDON, *Modeling and Simulation of a Hybrid Powertrain*, Master’s thesis, School of Electrical Engineering and Computer Science (EECS), 2018.
- [19] J. HOFSTETTER, H. BAUER, W. LI, AND G. WACHTMEISTER, *Energy and Emission Management of Hybrid Electric Vehicles using Reinforcement Learning*, *IFAC-Pap.*, 52 (2019), pp. 19–24.
- [20] L. HYEOUN-DONG AND S.-K. SUL, *Fuzzy-logic-based torque control strategy for parallel-type hybrid electric vehicle*, *IEEE Trans. Ind. Electron.*, 45 (1998), pp. 625–632.
- [21] H. KAZEMI, Y. P. FALLAH, A. NIX, AND S. WAYNE, *Predictive AECMS by Utilization of Intelligent Transportation Systems for Hybrid Electric Vehicle Powertrain Control*, *IEEE Trans. Intell. Veh.*, 2 (2017), pp. 75–84.
- [22] M. KIM, D. JUNG, AND K. MIN, *Hybrid Thermostat Strategy for Enhancing Fuel Economy of Series Hybrid Intracity Bus*, *IEEE Trans. Veh. Technol.*, 63 (2014), pp. 3569–3579.
- [23] N. KIM, S. CHA, AND H. PENG, *Optimal Control of Hybrid Electric Vehicles Based on Pontryagin’s Minimum Principle*, *IEEE Trans. Control Syst. Technol.*, 19 (2011), pp. 1279–1287.
- [24] L. S. PONTRYAGIN, V. G. BOLTYANSKII, R. V. GAMKRELIDZE, E. F. MISHECHENKO, *The Mathematical Theory of Optimal Processes*, New York, (1962).
- [25] A. LE RHUN, *Stochastic optimal control for the energy management of hybrid electric vehicles under traffic constraints*, PhD thesis, Université Paris Saclay (COmUE), 2019.

- [26] A. A. MALIKOPOULOS, *Supervisory Power Management Control Algorithms for Hybrid Electric Vehicles: A Survey*, IEEE Trans. Intell. Transp. Syst., 15 (2014), pp. 1869–1885.
- [27] Y. MILHAU, D. SINOQUET, AND G. ROUSSEAU, *Design optimization and optimal control for hybrid vehicles*, Optim. Eng., 12 (2011), pp. 199–213.
- [28] C. MUSARDO, G. RIZZONI, Y. GUEZENNEC, AND B. STACCIA, *A-ECMS: An Adaptive Algorithm for Hybrid Electric Vehicle Energy Management*, Eur. J. Control, 11 (2005), pp. 509–524.
- [29] G. PAGANELLI, S. DELPRAT, T.-M. GUERRA, J. RIMAU, AND J.-J. SANTIN, *Equivalent consumption minimization strategy for parallel hybrid powertrains*, in IEEE 55th Vehicular Technology Conference, vol. 4, 2002, pp. 2076–2081.
- [30] A. PHILLIPS, M. JANKOVIC, AND K. BAILEY, *Vehicle system controller design for a hybrid electric vehicle*, in Proceedings of the 2000. IEEE International Conference on Control Applications., 2000, pp. 297–302.
- [31] A. RAO, *A Survey of Numerical Methods for Optimal Control*, Adv. Astronaut. Sci., 135 (2010).
- [32] A. RAO AND K. MEASE, *Dichotomic basis approach to solving hyper-sensitive optimal control problems*, Automatica, 35 (1999), pp. 633–642.
- [33] H. SCHÄTTLER AND U. LEDZEWICZ, *Geometric optimal control: theory, methods and examples*, vol. 38, Springer, 2012.
- [34] J. SCORDIA, M. DESBOIS-RENAUDIN, R. TRIGUI, B. JEANNERET, F. BADIN, AND C. PLASSE, *Global optimisation of energy management laws in hybrid vehicles using dynamic programming*, Int. J. Veh. Des., 39 (2005).
- [35] L. SERRAO, S. ONORI, AND G. RIZZONI, *A Comparative Analysis of Energy Management Strategies for Hybrid Electric Vehicles*, J. Dyn. Syst. Meas. Control, 133 (2011).
- [36] M. SIVERTSSON AND L. ERIKSSON, *Design and Evaluation of Energy Management using Map-Based ECMS for the PHEV Benchmark*, Oil Gas Sci. Technol. - Rev. IFP Energies nouvelles, 70 (2015), pp. 195–211.
- [37] J. STOER AND R. BULIRSCH, *Introduction to Numerical Analysis*, Springer New York, 2002.
- [38] R. S. SUTTON AND A. G. BARTO, *Reinforcement Learning: An Introduction*, The MIT Press, 2018.
- [39] E. TATE AND S. BOYD, *Finding Ultimate Limits of Performance for Hybrid Electric Vehicles*, SAE Trans., 109 (2000), pp. 2437–2448.
- [40] X. WANG, H. HE, F. SUN, AND J. ZHANG, *Application Study on the Dynamic Programming Algorithm for Energy Management of Plug-in Hybrid Electric Vehicles*, Energies, 8 (2015), pp. 3225–3244.
- [41] S. G. WIRASINGHA AND A. EMADI, *Classification and Review of Control Strategies for Plug-In Hybrid Electric Vehicles*, IEEE Trans. Veh. Technol., 60 (2011), pp. 111–122.
- [42] S. XIE, X. HU, S. QI, AND K. LANG, *An Artificial Neural Network-Enhanced Energy Management Strategy for Plug-In Hybrid Electric Vehicles*, Energy, 163 (2018).
- [43] Y. ZHANG, L. CHU, Z. FU, N. XU, C. GUO, D. ZHAO, Y. OU, AND L. XU, *Energy management strategy for plug-in hybrid electric vehicle integrated with vehicle-environment cooperation control*, Energy, 197 (2020).

## A Proof of Theorem 1

We want to prove that

$$\nabla_a V_*(x(t_0), x(t_f)) = -p(t_0), \quad (19)$$

$$\nabla_b V_*(x(t_0), x(t_f)) = p(t_f). \quad (20)$$

Equation (19) is a classical result that can be found in [6, 8, 10, 13, 33]. We shall use this result to prove (20). For this purpose, we transform  $(\text{OCP}_*)$  into  $(\text{ROCP}_*)$  using the reverse time transformation  $\phi: [t_0, t_f] \rightarrow [t_0, t_f]$  defined by  $\phi(t) = t_f + t_0 - t$ :

$$(\text{ROCP}_*) \quad \left\{ \begin{array}{l} V_R(b, a) = \min_{\hat{x}, \hat{u}} \int_{t_0}^{t_f} f^0(\phi(t), \hat{x}(t), \hat{u}(t)) dt, \\ \text{s.t. } \dot{\hat{x}}(t) = -f(\phi(t), \hat{x}(t), \hat{u}(t)), \quad t \in [t_0, t_f] \text{ a.e.}, \\ \hat{u}(t) \in U(\phi(t)), \quad t \in [t_0, t_f], \\ \hat{x}(t_0) = b, \quad \hat{x}(t_f) = a. \end{array} \right.$$

We have naturally the following relation between the value functions:

$$V_R(b, a) = V_*(a, b). \quad (21)$$

Using the classical transformation  $\theta_R(x, p, u) = (x \circ \phi, -p \circ \phi, u \circ \phi)$  and denoting  $(\hat{x}, \hat{p}, \hat{u}) = \theta_R(x, p, u)$  it can easily be shown that  $(\text{OCP}_*)$  is equivalent to  $(\text{ROCP}_*)$  in the sense that

$$\begin{aligned} & (x, p) \text{ is a BC-extremal associated to } (x, u) \text{ solution of } (\text{OCP}_*) \\ \iff & (\hat{x}, \hat{p}) \text{ is a BC-extremal associated to } (\hat{x}, \hat{u}) \text{ solution of } (\text{ROCP}_*). \end{aligned}$$

Since  $(\text{ROCP}_*)$  has the same form as  $(\text{OCP}_*)$  and the value function  $V_R$  is differentiable at  $(b, a)$ , we can apply (19) to  $(\text{ROCP}_*)$ :

$$\nabla_b V_R(\hat{x}(t_0), \hat{x}(t_f)) = -\hat{p}(t_0) = -(-p \circ \phi)(t_0) = p(t_f).$$

Finally, using (21), we get  $\nabla_b V_*(a, b) = \nabla_b V_R(b, a) = p(t_f)$ .  $\square$

## B Case study notations

Name	Description	Unit
Cost		
$m_F$	Fuel consumption	g
State		
$SOC$	Battery state of charge	
Controls		
$Gear$	Gearbox selector	
$T_{qICE}$	ICE torque	N.m
Variables		
$T_{qICE_L}$	ICE torque losses	N.m
$T_{qEM}$	EM torque	N.m
$T_{qEM_L}$	EM torque losses	N.m
$T_{qW}$	Wheels requested torque	N.m
$N_{ICE}$	ICE rotation speed	RPM
$N_{EM}$	EM rotation speed	RPM
$N_W$	Wheels requested rotation speed	RPM
$I$	Electrical circuit current	A
$P_{elec}$	Power demand	W
Constants		
$R(Gear)$	ICE gearbox ratio (constant for each $Gear$ )	
$P$	Transmission ratio between main and ICE axes	
$P_e$	Transmission ratio between main and EM axes	
$Q$	Battery charge	C
$T$	Coolant temperature	°C

Table 3: List of parameters involved in the static model of HEV's torque split.