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

Olivier Cots, Rémy Dutto, Sophie Jan, Serge Laporte

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

Olivier Cots* Rémy Dutto†‡§ Sophie Jan‡ Serge Laporte‡

July 9, 2024

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 the 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 of 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

Presentation of the application. Hybrid Electric Vehicles (HEVs) or Plug-in Hybrid Electric Vehicles 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 movers, respectively Internal Combustion Engine (ICE) and Electric Motor (EM). A control law provides the strategy for gear shift and torque split between these two movers 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 an embedded solution, the prediction of the speed in a connected vehicle is also a problem that is currently being studied [9, 46], but we shall not address it here.

This paper is motivated by the torque split and gear shift problem which has been widely studied in the literature [17, 28, 44] and for which a lot of approaches have been proposed. For instance, rule-based laws have been developed like thermostat strategy [24], state machine controller [33], or fuzzy logic solution [22]. Methods based on instantaneous optimization are also proposed, like the Equivalent Consumption Management Strategy (ECMS), introduced by [32], where the considered

*Institut de Recherche en Informatique de Toulouse, UMR CNRS 5505, Université de Toulouse, INP-ENSEEIHT, France.
†Vitesco Technologies, Toulouse, France.
‡Institut de Mathématiques de Toulouse, UMR CNRS 5219, Université de Toulouse, UPS, France.
§Corresponding author: remy.dutto@orange.fr.

cost is the sum of the ICE fuel consumption and an equivalent factor multiplied by the battery state of charge deviation. This method may be seen as an application of the Pontryagin maximum principle with some restrictive assumptions [38, 45]. Finally, other methods derived from the ECMS have been developed, which differ by the parameterization of the equivalent factor and its possible online adaptation [23, 30, 31, 39, 45, 46].

Optimal control framework. 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, initial and final state constraints and an objective function in integral form. A first class of numerical methods is based on the Pontryagin maximum principle [26] which provides necessary optimality conditions for optimal control problems and leads in the simplest case to a Boundary Value Problem (BVP). For instance, the indirect shooting method [40] aims to solve the BVP introducing the so-called shooting function which defines a set of nonlinear equations that can be solved by a Newton-like algorithm. This method has been applied on a similar application, see for instance [19, 25]. Another class of methods first discretizes the control and the state on a given grid of times. This leads to a large nonlinear constrained optimization problem in finite dimension that can be solved by nonlinear programming algorithms. This kind of methods are called direct methods and have already been set in this context, see [42].

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, 37, 43] or reinforcement learning [21, 41]. 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.

Motivation and objective. The torque split and gear shift optimal control problem studied here is characterized by a long cycle duration (1800 seconds for the WLTC) compared to the sampling period (classically about 1s) used to have an accurate representation of the model dynamics. This leads to a long integration time and therefore a high sensitivity of the indirect shooting function. Moreover, a good initial guess must be given to ensure that the Newton solver converges to a zero of the shooting function. The main objective of this paper is to propose a new optimal control method, called Macro–Micro, which is a first step for a new embedded solution. For that purpose, the Macro–Micro method has to be fast, robust and computationally efficient.

Contributions.

1. **Novel optimal control method.** In this paper, we present a new optimal control method, called Macro–Micro. This method is based on a bilevel formulation of the considered optimal control problem, which is strongly linked to the classical multiple shooting method as highlighted by Proposition 2 and the commutative diagram of Figure 1. This bilevel decomposition uses the Bellman's value functions, which are generally difficult to compute, and for that reason they are approximated: this is the Macro–Micro method. This method is composed of a macro optimization problem, called (Macro), which provides intermediate states at fixed times, and a set of micro independent optimal control problems, called (Micro), defined on smaller time intervals. Thanks to the link between the gradient of the Bellman's value functions and the Pontryagin's costate given by Theorem 1, we propose a natural initialization of the shooting functions associated to the micro optimal control problems. Due to the approximations of the value functions, the proposed method is sub-optimal. However, an error analysis is done in Section 3.3.4 and shows that the error of the Macro–Micro method can be controlled by the

error of the approximation of the value functions. Finally, in order to create the approximations of the value functions, an efficient method to create the ad-hoc databases is proposed. This method does not require any problem to solve compared to a naive approach. See [15] for more details about the comparison between our approach and the naive approach.

2. **Application.** The proposed Macro–Micro method is applied to the torque split and gear shift optimal control problem in Section 4. Since this problem is a well-known optimal control problem in the literature, we do not consider this problem as a contribution. However, the application of the proposed method and the comparison with classical simple shooting is a part of the contributions of this paper. For this purpose, the numerical methods used for the simple shooting and the proposed Macro–Micro method are described. Moreover, an empiric method is proposed to highlight the benefits of the (Macro) problem. A comparison in terms of time computation, number of computations and optimality is done. Moreover, we compare the convergence of the Newton solver on two cases: first we provide a fixed but well-chosen initialization and second we use our proposed natural initialization.

Organization of the paper. 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. The main contributions of this paper are postponed to the next sections. The considered optimal control problem is introduced in Section 2.1. Then, necessary optimality conditions provided by the Pontryagin maximum principle are given in Section 2.2 and some standard 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, denoted (OCP), in Lagrange form:

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

where the Lagrange cost $f^0 \colon \mathbb{R} \times \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}$ and the state dynamics $f \colon \mathbb{R} \times \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^n$ are two functions of class $\mathscr{C}^1$. The initial and final times $t_0 < t_f$ are fixed. The control domain $\mathrm{U}(t) \subset \mathbb{R}^m$ is a non-empty set for every $t \in [t_0,t_f]$ with additional standard regularity assumptions (cf. [12, Chapter 4.2, Remark 5] for more information). The mixed initial and final constraints $c \colon \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}^p$ is a function of class $\mathscr{C}^1$, with $p \leq 2n$. Moreover, c is a submersion on $c^{-1}(\{0\})$, *i.e.* $c'(a,b)$ is surjective for any pair (a,b) such that $c(a,b) = 0$. Solving (OCP) amounts to find a pair $(x,u) \in \mathcal{D} = \mathrm{AC}([t_0,t_f],\mathbb{R}^n) \times \mathrm{L}^\infty([t_0,t_f],\mathbb{R}^m)$ of an absolutely continuous[1] *state* x and an essentially bounded[2] *control* u which minimizes (1) and satisfies (2), (3) and (4).

[1] $\mathrm{AC}([t_0,t_f],\mathbb{R}^n)$ is the set of absolutely continuous functions on $[t_0,t_f]$ valued in $\mathbb{R}^n$.
[2] $\mathrm{L}^\infty([t_0,t_f],\mathbb{R}^m)$ is the set of essentially bounded functions on $[t_0,t_f]$ valued in $\mathbb{R}^m$.

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.*

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 introduce:

- for any control law u, the mapping $t \mapsto x(t,t_1,x_1,u)$ as the solution of the Cauchy problem $\dot{x}(t) = f(t,x(t),u(t))$, $x(t_1) = x_1$,
- the set of admissible controls

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

- the *extremity mapping* $E_{t_2,t_1,x_1}\colon \mathcal{U}_{t_2,t_1,x_1} \to \mathbb{R}^n$, $u \mapsto E_{t_2,t_1,x_1}(u) = x(t_2,t_1,x_1,u)$,
- 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 Lagrange form with fixed final time. However, by standard techniques, see [12], 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 [12, 13].*

2.2 Pontryagin maximum principle

According to the Pontryagin maximum principle [12, 26], if (x,u) is a solution of (OCP), then there exist a *costate* trajectory $p \in \mathrm{AC}([t_0,t_f],\mathbb{R}^n)$, a scalar $p^0 \in \{-1,0\}$ and a Lagrange multiplier $\lambda \in \mathbb{R}^p$ such that (p,p^0) is non trivial, the following Hamiltonian dynamics is satisfied: for almost every $t \in [t_0,t_f]$

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

as well as the maximization condition: for almost every $t \in [t_0,t_f]$

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

where $h(t,x,p,u) = p^0\ f^0(t,x,u) + \big(p \,\big|\, f(t,x,u)\big)$ is the *pseudo-Hamiltonian* and $\big(a \,\big|\, b\big)$ is the usual scalar product on $\mathbb{R}^n$. Moreover, we have the following 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* $\operatorname{Im} c'(x(t_0),x(t_f))^\top = (\operatorname{Ker} c'(x(t_0),x(t_f)))^\perp$. *Since* c *is a submersion on* $c^{-1}(\{0\})$, *then* $c'(x(t_0),x(t_f))\colon \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}^p$ *is a surjective linear map which means that* $\operatorname{Im} c'(x(t_0),x(t_f)) = \mathbb{R}^p$. *Hence, we can construct a* $2n \times (2n-p)$ *matrix* $B_c(x(t_0),x(t_f))$ *whose columns form a basis of* $\operatorname{Ker} c'(x(t_0),x(t_f))$. *At the end, the transversality condition can be written*

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

The advantage of this formulation is that there is not λ.

Definition 2. *An* extremal *is a pair* $z := (x,p) \in \mathrm{AC}([t_0,t_f],\mathbb{R}^n) \times \mathrm{AC}([t_0,t_f],\mathbb{R}^n)$ *associated with a control* $u \in \mathrm{L}^\infty([t_0,t_f],\mathbb{R}^m)$ *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*,

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

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

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

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

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 $\dot{x}(t) = \vec{F}\left(t, x(t)\right),\ x(t_0) = x_0$.

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 applied. These assumptions are used to simplify the notation and to prove that the diagram in Figure 1 is commutative. However, the proposed method detailed in Section 3.3 can be applied under weaker assumptions as soon as $\exp_{\vec{H}}$ is well-defined. We refer to [18, 29] for examples where $\exp_{\vec{H}}$ is well-defined under weaker assumptions.*

2.4 Indirect simple shooting

Under our general assumptions, the application of the Pontryagin maximum principle to Problem (OCP) leads to the following two-points boundary value problem:

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

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

$$g(z_0, z_f) = \begin{pmatrix} c(x_0, x_f) \\ c^\star(z_0, z_f) \end{pmatrix}, \tag{9}$$

where, according to Remark 2, the function $c^\star\colon \mathbb{R}^{2n} \times \mathbb{R}^{2n} \to \mathbb{R}^{2n-p}$ is defined by

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

and where $z_0 = (x_0, p_0)$ and $z_f = (x_f, p_f)$. A well-known method to solve Problem (TPBVP) is the so-called indirect simple shooting method [40] (named also single shooting [10]) which consists in finding a zero of the simple shooting function $S_s\colon \mathbb{R}^{2n} \to \mathbb{R}^{2n}$ defined by

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

However, the simple shooting method suffers from numerical issues. Indeed, as shown in [5, 34, 35], 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 is made of 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 continuity of the state and the costate at each interface. Considering intermediate times $t_0 < t_1 < ... < t_N < t_{N+1} = t_f$ which decompose $[t_0, t_f]$ into $N+1$ sub-intervals $\Delta_i = [t_i, t_{i+1}]$, the two-points boundary value problem (TPBVP) is transformed into a multi-points boundary value problem named (MPBVP) and defined by

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

with the notation $\mathbb{N}_k = [\![0, k]\!]$. The corresponding multiple shooting function $S_m \colon (\mathbb{R}^{2n})^{N+1} \to (\mathbb{R}^{2n})^{N+1}$ is therefore

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

Thanks to the integration on smaller sub-intervals, the sensitivity with respect to the initial guess is reduced [4, 40]. Nevertheless, even multiple shooting methods need a good initialization to converge.

3 Bilevel optimal control method

The first main contribution of this article is described in this section. A new transcription path from (OCP) to (MPBVP) is introduced, which leads to the development of a novel method to solve optimal control problems. The new path is based on a 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 permits the development of a novel approach described in Section 3.3, called Macro–Micro, which promises to be robust and fast enough for embedded solutions.

3.1 Bilevel formulation

Considering the time intervals defined in Section 2.5, let us first introduce the following intermediate optimal control problems for all $i \in \mathbb{N}_N$:

$$\text{(OCP}_{i,a,b}\text{)} \qquad \begin{cases} V_i(a,b) = \min\limits_{x,u} \displaystyle\int_{t_i}^{t_{i+1}} f^0(t, x(t), u(t)) \ \mathrm{d}t, \\ \qquad \text{s.t. } \dot{x}(t) = f(t, x(t), u(t)), & t \in \Delta_i \text{ a.e.}, \\ \qquad\quad u(t) \in \mathrm{U}(t), & t \in \Delta_i, \\ \qquad\quad x(t_i) = a, \quad x(t_{i+1}) = b. \end{cases}$$

Let $i \in \mathbb{N}_N$. We only consider initial and final states for (OCP$_{i,\cdot,\cdot}$) belonging to the open set

$$\Omega_i = \left\{ (a,b) \in \mathbb{R}^n \times \mathbb{R}^n \;\middle|\; b \in \mathrm{Int}\left(A(t_{i+1}, t_i, a)\right) \right\}, \tag{10}$$

where $\mathrm{Int}(A)$ stands for the interior of the set A. The function $V_i \colon \Omega_i \to \mathbb{R}$ corresponds to the optimal value function of (OCP$_{i,\cdot,\cdot}$). Denoting $\mathcal{D}_i = \mathrm{AC}(\Delta_i, \mathbb{R}^n) \times \mathrm{L}^\infty(\Delta_i, \mathbb{R}^m)$, the cost function $J_i \colon \mathcal{D}_i \to \mathbb{R}$ is defined by

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

Let $(a,b) \in \Omega_i$. The set of admissible points of Problem ($\mathrm{OCP}_{i,a,b}$) is

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

and the set of solutions is

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

Given a vector of *intermediate states* $X = (X_0, \dots, X_{N+1}) \in (\mathbb{R}^n)^{N+2}$ we are led to solve $N+1$ elementary optimal control problems ($\mathrm{OCP}_{i,X_i,X_{i+1}}$). We define the cost of the vector X as the sum of the optimal values of the ($\mathrm{OCP}_{i,X_i,X_{i+1}}$) sub-problems. An equivalent manner to formulate (OCP) is then to find X that minimizes this cost. This leads to formulate (OCP) as the following bilevel optimal control problem:

$$\text{(BOCP)} \qquad \left\{ \begin{array}{l} \min\limits_{X} \; V(X) \coloneqq \sum\limits_{i=0}^{N} V_i\left(X_i, X_{i+1}\right), \\ \text{s.t. } X \in \mathcal{X}, \quad c(X_0, X_{N+1}) = 0, \end{array} \right.$$

where $\mathcal{X}$ is the open set of admissible intermediate states:

$$\mathcal{X} = \left\{ (X_0, \dots, X_{N+1}) \in (\mathbb{R}^n)^{N+2} \;\middle|\; \forall i \in \mathbb{N}_N, \ (X_i, X_{i+1}) \in \Omega_i \right\}. \tag{12}$$

Let $i \in \mathbb{N}_N$ and $(X_i, X_{i+1}) \in \Omega_i$. Since all the solutions (x_i, u_i) of Problem ($\mathrm{OCP}_{i,X_i,X_{i+1}}$) have 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:

$$\left\{ \begin{array}{l} \min\limits_{X} \max\limits_{x,u} \sum\limits_{i=0}^{N} J_i(x_i, u_i) \\ \text{s.t.} \quad X \in \mathcal{X}, \quad c(X_0, X_{N+1}) = 0, \\ \qquad \forall i \in \mathbb{N}_N, \quad (x_i, u_i) \in \mathcal{S}_i(X_i, X_{i+1}) \end{array} \right. \tag{13}$$

or equivalently

$$\left\{ \begin{array}{l} \min\limits_{X} \min\limits_{x,u} \sum\limits_{i=0}^{N} J_i(x_i, u_i) \\ \text{s.t.} \quad X \in \mathcal{X}, \quad c(X_0, X_{N+1}) = 0, \\ \qquad \forall i \in \mathbb{N}_N, \quad (x_i, u_i) \in \mathcal{S}_i(X_i, X_{i+1}) \end{array} \right. \tag{14}$$

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 ($\mathrm{OCP}_{i,X_i,X_{i+1}}$).

3.2 Link with multiple shooting

From now, (OCP) has been reformulated as (BOCP). Next, we want to show that the first-order necessary optimality conditions applied to (BOCP) lead to (MPBVP) under some assumptions. For this purpose, we introduce the following assumption which stands from now.

Hypothesis 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.*

The first-order necessary conditions of optimality for Problem (BOCP) are given by the Karush-Kuhn-Tucker conditions. Recalling that $\mathcal{X}$ is an open set, if the vector of intermediate states $X = (X_0, \dots, X_{N+1}) \in (\mathbb{R}^n)^{N+2}$ is a solution of (BOCP), then there exists $\lambda \in \mathbb{R}^p$ such that

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

where the Lagrangian $L\colon (\mathbb{R}^n)^{N+2} \times \mathbb{R}^p \to \mathbb{R}$ associated to (BOCP) is defined by

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

Using the expressions of L and V, (NCBOCP) rewrites

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

In order to make the costate appear in the above optimality conditions, we introduce the following theorem.

Theorem 1. *Given $(a, b) \in \Omega := \big\{(a, b) \in \mathbb{R}^n \times \mathbb{R}^n \;\big|\; b \in \mathrm{Int}\,(A(t_f, t_0, a))\big\}$, we consider a particular case of (OCP) in which $c(x(t_0), x(t_f)) = (x(t_0) - a, x(t_f) - b)$, that is the initial and final conditions are imposed:*

$$\text{(OCP}_*\text{)} \qquad \begin{cases} V_*(a, b) = \min\limits_{x,u} \displaystyle\int_{t_0}^{t_f} f^0\,(t, x(t), u(t))\;\mathrm{d}t, \\ \qquad \text{s.t. } \dot{x}(t) = f\,(t, x(t), u(t)) \qquad t \in [t_0, t_f] \text{ a.e.}, \\ \qquad\quad u(t) \in \mathrm{U}(t), \qquad\qquad t \in [t_0, t_f], \\ \qquad\quad x(t_0) = a, \quad x(t_f) = b. \end{cases}$$

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

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

Proof. See Appendix A. □

Let now $X = (X_0, \dots, X_{N+1}) \in (\mathbb{R}^n)^{N+1}$ be a solution of (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 (OCP$_{i,X_i,X_{i+1}}$) comes with a solution (x_i, u_i) and an associated normal BC-extremal (x_i, p_i). Since (OCP$_{i,X_i,X_{i+1}}$) has the same form as (OCP$_*$), we can apply Theorem 1 and we get

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

Therefore (NCBOCP) can be rewritten:

$$\left\{\begin{array}{l} \forall i \in \mathbb{N}_N,\ \exists\, z_i = (x_i, p_i) \text{ a BC-extremal} \\ \text{associated to a solution } (x_i, u_i) \text{ of } \left(\mathrm{OCP}_{i,X_i,X_{i+1}}\right), \\ \begin{pmatrix} -p_0(t_0) \\ p_N(t_{N+1}) \end{pmatrix} - c'(X_0, X_{N+1})^\top \lambda = 0, \\ \forall i \in \{1, ..., N\}, p_{i-1}(t_i) - p_i(t_i) = 0, \\ c(X_0, X_{N+1}) = 0, \quad X \in \mathcal{X}, \end{array}\right. \tag{15}$$

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

$$\text{(NCBOCP)} \implies \left\{\begin{array}{l} \forall i \in \mathbb{N}_N,\ \exp_{\overrightarrow{H}}\left(t_{i+1}, t_i, z_i(t_i)\right) = z_i(t_{i+1}), \\ \forall i \in \mathbb{N}_N,\ x_i(t_i) - X_i = 0, \\ \forall i \in \mathbb{N}_N,\ x_i(t_{i+1}) - X_{i+1} = 0, \\ \begin{pmatrix} -p_0(t_0) \\ p_N(t_{N+1}) \end{pmatrix} - c'(X_0, X_{N+1})^\top \lambda = 0, \\ \forall i \in \{1, ..., N\}, p_{i-1}(t_i) - p_i(t_i) = 0, \\ c(X_0, X_{N+1}) = 0, \quad X \in \mathcal{X}, \end{array}\right\} \iff \text{(MPBVP)}. \tag{16}$$

The above leads us to the following proposition.

Proposition 2. *Under our general assumptions, if* (X, λ) *is a solution of* (NCBOCP) *then*

$$z_i = \left(X_i, -\nabla_a V_i(X_i, X_{i+1})\right), \quad \forall i \in \mathbb{N}_{N-1},$$

is a solution of (MPBVP). *We denote this relation by* (NCBOCP) $\implies$ (MPBVP).

Remark 5. *We recall our general assumptions. To apply Theorem 1 we have assumed that the value functions are differentiable at the points of interest and that the reference BC-extremals are normal. Additionally to these technical assumptions we have considered the Hamiltonian frame where the maximized Hamiltonian is well-defined and smooth, see Section 2.3. The Hamiltonian frame permits to define simply the multiple shooting method introduced in Section 2.5 and leads to* (MPBVP).

As already seen in Sections 2.4 and 2.5, the initial problem (OCP) is linked to (MPBVP) through (TPBVP). A new path is now highlighted by Proposition 2 via (BOCP) and (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: the Pontryagin maximum principle for (OCP) and the Karush-Kuhn-Tucker 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) as 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.

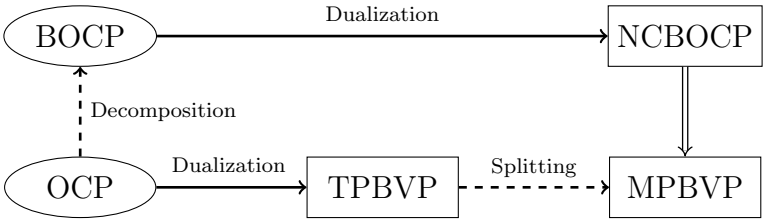


Figure 1: Commutative diagram between (OCP) and (MPBVP). The solid arrows correspond to Dualization operator and the dashed arrows correspond to Decomposition or Splitting. Finally, the double arrow is the implication operator from Proposition 2.

3.3 A novel optimal control method

As indicated in Figure 1, a new path is proposed to transform (OCP) into (MPBVP) using the bilevel decomposition. This path leads to a novel approach and a new method presented in this section.

3.3.1 Starting point

Let us suppose that the value functions V_i are known for every $i \in \mathbb{N}_N$. 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) to get optimal intermediate states $X^* = (X_0^*, \dots, X_{N+1}^*) \in (\mathbb{R}^n)^{N+1}$,
- the second one is to solve the $N+1$ independent optimal control problems $(\mathrm{OCP}_{i,X_i^*,X_{i+1}^*})$. Each of these problems is defined on a smaller time interval than (OCP) 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 $\left(X_i^*, -\nabla_a V_i(X_i^*, X_{i+1}^*)\right)$ provided by the first step is a zero of the associated simple shooting function, and so it is no more necessary to use the shooting method. Hence, 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 Main idea: 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 denoted C_i. Problem (BOCP) is then transformed into

$$\text{(Macro)} \qquad \begin{cases} \min\limits_{X} C(X) \coloneqq \sum\limits_{i=0}^{N} C_i\left(X_i, X_{i+1}\right), \\ \text{s.t. } X \in \mathcal{X}, \quad c(X_0, X_{N+1}) = 0, \end{cases}$$

to get the intermediate state $\hat{X} = (\hat{X}_0, \dots \hat{X}_{N+1}) \in (\mathbb{R}^n)^{N+2}$ solution of (Macro), and then to solve the $N+1$ following optimal control problems

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

The pair

$$\left(\hat{X}_i, -\nabla_a C_i(\hat{X}_i, \hat{X}_{i+1})\right) \tag{17}$$

is no longer necessary 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 (17) provides a natural initial guess. A similar approach has been developed in [16], where a rough approximation of the value function, obtained by a HJB based method, is used to provide an initial guess for the indirect shooting method.

Numerically, the resolution of Problems (Macro) and (Micro) 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 block diagram of the proposed Macro–Micro method is given in Figure 2.

A similar Macro–Micro approach has been developed in [27] for stochastic optimal control problems with dynamic programming methods on predefined traffic conditions.

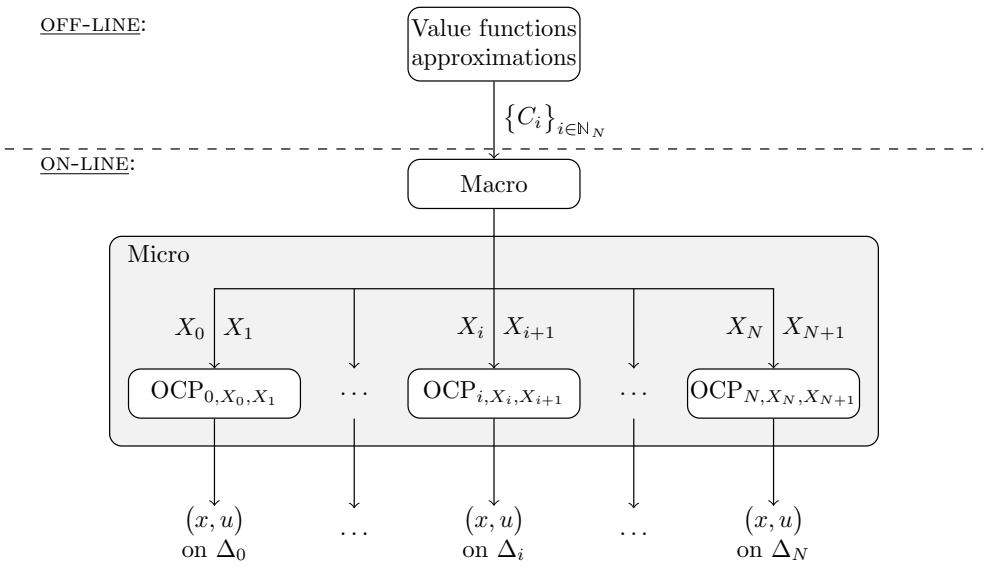


Figure 2: Block diagram of the Macro–Micro method. Since the approximations of the value functions are considered to be known, it is an "off-line" task, compared to the Macro–Micro resolution which is an "on-line" task. This schema highlights that Problem (Micro) is composed of $N+1$ independent optimal control problems defined on smaller time intervals.

3.3.3 Advantages

Most of the computation time of the Macro–Micro method is due to the resolution of Problem (Micro). In comparison with the simple shooting method, $N+1$ optimal control problems have to be solved but on $N+1$ shorter time intervals. Thanks to parallel computing, the computation time for (Micro) resolution is divided by a factor $N+1$. Moreover, the proposed initialization given by Equation (17) could reduce the number of iterations of the Newton-like solver (see Figure 7 on the application), which further decreases the computation time.

As proved in Section 3.1, the Macro–Micro method is strongly linked to multiple shooting. Both methods take advantage of shorter time intervals and therefore show a reduced sensitivity to the initial guess. They also can benefit from parallel computing but with a major difference: optimal control problems in Problem (Micro) 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 another remarkable advantage: only the first optimal control problem defined on the first sub-interval of Problem (Micro) 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 $V(X^*)$, where we recall that $\hat{X}$ is a solution of Problem (Macro), and X^* is an optimal intermediate state, that is a solution of (BOCP). 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 ε_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 $\alpha \coloneqq (N+1) \max_{i \in \mathbb{N}_N} \varepsilon_i$, we obtain equivalently

$$V(X) - \alpha \leq C(X) \leq V(X) + \alpha. \tag{18}$$

Since X^* is a global solution of (BOCP) and using (18) 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 (18) 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

$$\left| V(\hat{X}) - V(X^*) \right| \leq 2\alpha. \tag{19}$$

The maximum gap e between $V(X^*)$ and $V(\hat{X})$ being given, (19) 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 $(a,b) \in \Omega_i$

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

ensures that the cost difference between the optimal solution and the Macro–Micro one is less than e.

3.3.5 Extremal database construction

The Macro–Micro method requires approximations of value functions. 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

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

which needs to be created. For that purpose, the natural method consists in evaluating V_i on various admissible pairs (a,b). This implies the resolution of optimal control problems. We propose here an alternative method, where there is no problem to solve. Let us define

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

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

Hypothesis 2. *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 2, for all (x_0,p_0) and $(x_f,p_f) = \exp_{\vec{H}}\big(t_{i+1},t_i,(x_0,p_0)\big)$, we have

$$V_i(x_0,x_f) = c_i(x_0,p_0), \tag{20}$$

where $c_i(x_0,p_0)$ is the cost of the trajectory on the time interval Δ_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 (11) along the pair of state and control trajectories, the control being given by the maximization condition.

Remark 6. *Hypothesis 2 is motivated by the application and it is shown in Figure 5 that this hypothesis can be considered to be numerically valid. Equation (20) 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

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

For further information, the pros and cons of this method compared to those of the natural one are detailed in [15].

4 Case study

The second main contribution of this paper is presented in this section. The objective is to compare the Macro–Micro method to simple shooting and an empiric method on an industrial application. First, the application we consider 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 B.

4.1 Hybrid electric vehicle torque split and gear shift problem

The selected vehicle is a light HEV with a 400V asynchronous EM, a 1.5L 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 HEV is a parallel P3 with a double drive shaft, as presented in Figure 3. Other HEV's architectures exist, and are listed in [17]. 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.

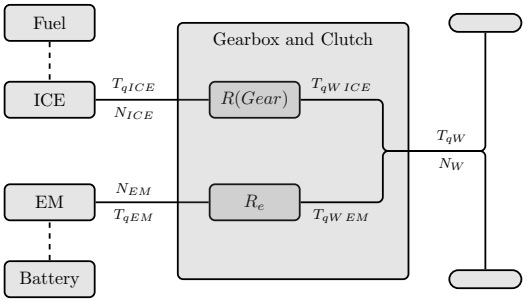


Figure 3: Scheme of the selected HEV. The solid lines represent mechanical connections and the dashed lines represent energy connections, using fuel for ICE and electricity for EM.

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:

$$
\text{(P)} \qquad \begin{cases} \min\limits_{SOC, T_{qICE}, Gear} \displaystyle\int_{t_0}^{t_f} f_{m_F}\left(t, T_{qICE}(t), Gear(t)\right)\ \mathrm{d}t, \\ \text{s.t.}\ \ \dot{SOC}(t) = f_{SOC}\left(t, SOC(t), T_{qICE}(t), Gear(t)\right), \quad t \in [t_0, t_f] \text{ a.e.}, \\ \qquad \left(T_{qICE}(t), Gear(t)\right) \in \mathrm{U}(t), \qquad\qquad t \in [t_0, t_f], \\ \qquad SOC(t_0) = SOC_0, \quad SOC(t_f) = SOC_f. \end{cases}
$$

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 in Figure 4. 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.

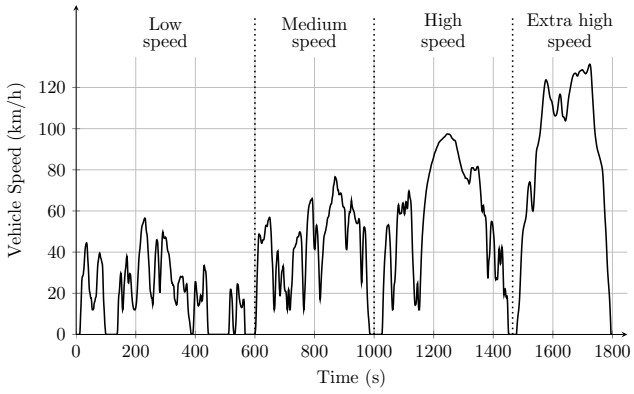


Figure 4: Speed evolution for the WLTC. This cycle is mainly used for emission and fuel consumption test.

4.1.2 System modeling

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

$$\begin{bmatrix} N_{ICE} \\ N_{EM} \end{bmatrix} = \begin{bmatrix} R(Gear) \\ R_e \end{bmatrix} N_W,$$

where N_{ICE} and N_{EM} represent respectively the rotation speed of the ICE and the EM. The function $R\colon \mathbb{R} \to \mathbb{R}$ is smooth and corresponds to the ratio of the selected $Gear$ between the wheel and the ICE drive shaft, and the parameter R_e corresponds to the ratio between the wheel and the EM drive shaft. The torque of the wheel T_{qW} is the sum of the torque of the wheel produced by the ICE $T_{qW\,ICE}$ and the one produced by the EM $T_{qW\,EM}$

$$T_{qW} = T_{qW\,ICE} + T_{qW\,EM}.$$

These two torques are calculated by

$$T_{qW\,ICE} = R(Gear)T_{qICE} - T_{qL\,ICE}, \qquad \text{and} \qquad T_{qW\,EM} = R_e T_{qEM} - T_{qL\,EM},$$

where T_{qICE} (respectively T_{qEM}) is the torque of the ICE (respectively EM) and $T_{qL\,ICE}$ (respectively $T_{qL\,EM}$) corresponds to the transmission losses. These losses can be calculated by two different ways:

- a map depending on the input of the gearbox, which is (N_{ICE},T_{qICE}) for $T_{qL\,ICE}$ and (N_{EM}, T_{qEM}) for $T_{qL\,EM}$,
- a map depending on the output of the gearbox, which is (N_W, $T_{qW\,ICE}$) for $T_{qL\,ICE}$ and (N_W, $T_{qW\,EM}$) for $T_{qL\,EM}$.

Then, the state of charge dynamics is calculated by the equivalent Thevenin RC circuit [23, 31] 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 $T = 90°C$. The other constraints of (P) come from physical limitations. The ICE and EM rotation speeds and torques have min-max constraints and $Gear \in [\![0, 4]\!]$. These constraints are seen as a control admissible domain $\big(T_{qICE}, Gear\big) \in \mathrm{U}$. The state of charge usually has 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 modeling can be found in the literature [20, 23, 32]. The dynamics of our vehicle is given by an industrial code, developed in Matlab Simulink by Vitesco Technologies with a 100ms sampling time, 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 the Macro–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 WLTC lasts 1800 seconds. We arbitrarily choose to decompose the 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 $\mathrm{U}(t)$ is thus discretized and the maximizing control $u^*(t, z)$ is extracted from the resulting finite set $\tilde{\mathrm{U}}(t)$, *i.e.*

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

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

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

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 are needed to use the Macro–Micro method.

4.2.1 Construction of the 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 in Figure 5 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 2 is considered numerically valid.

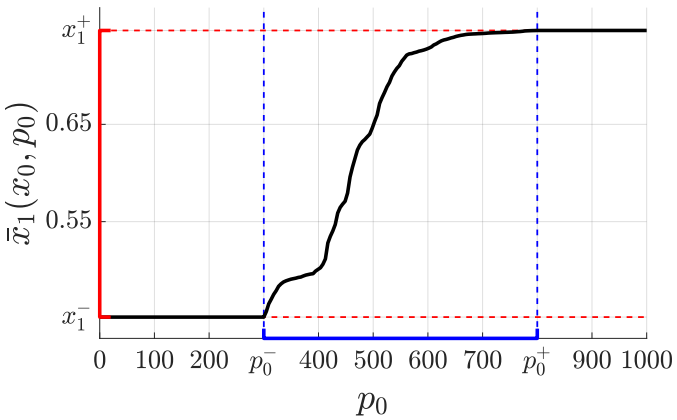


Figure 5: Evolution of the function $\bar{x}_1(x_0, .)$, with $x_0 = 0.5$. The red interval on the y-axis corresponds to the admissible state domain $A(t_1, t_0, x_0)$ and the blue interval on the x-axis corresponds to the section at x_0 of the domain Ψ_0.

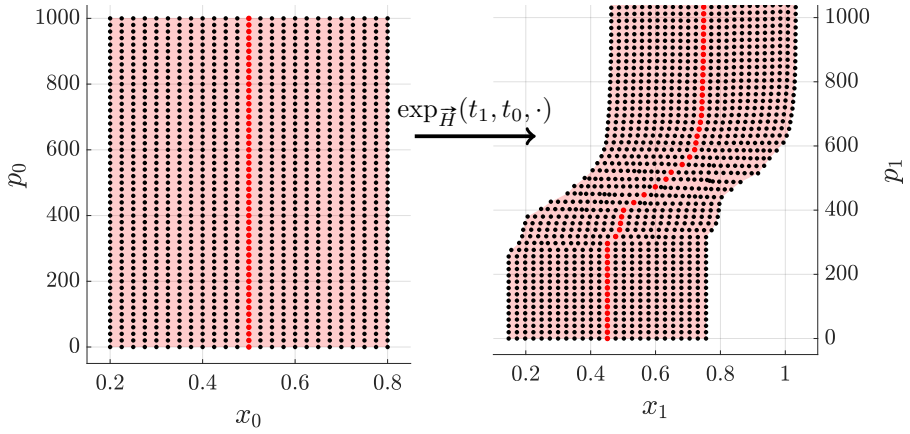


Figure 6: Transport of the initial state and costate space by the exponential map $(x_0, p_0) \mapsto \exp_{\vec{H}}(t_1, t_0, (x_0, p_0)) =: (x_1, p_1)$. The dots correspond to the points in the database $\mathbb{D}_0$. The red ones corresponds to $x_0 = 0.5$ and are highlighted to be compared with Figure 5.

For all $i \in \mathbb{N}_N$, the database of optimal values $\mathbb{D}_i$ associated to the time interval Δ_i is created 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 $SOC_{\min} = 0.2$ and $SOC_{\max} = 0.8$ (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 (and for all time intervals). For this purpose, we have chosen $p_{\min} = 0$ and $p_{\max} = 1000$. Moreover, the number of points of each discretization is chosen large enough to capture the variations of the value function. For this purpose, we have chosen a step of 2.5% for the initial state of charge, and respectively 20 for the initial costate. Each database $\mathbb{D}_i$ is composed of 1275 points. Figure 6 shows the transformation of the initial state and costate domain by the exponential map on the first time interval $\Delta_0 = [0, 100]$, for all the points in $\mathbb{D}_0$.

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 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	sigmoïd	8	136
Output (c)	id	1	8
Total			192

Table 1: Description of neural networks used to model the 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

Macro problem. Applied to the considered Problem (P), the Problem (Macro) can be written in the following form

$$\min_{(X_1,\dots,X_N)\in\mathcal{X}_P} C_P(X_1,\dots,X_N), \tag{21}$$

where $\mathcal{X}_P = \left\{ (X_1,\dots,X_N) \in \mathbb{R}^N \,\middle|\, (SOC_0, X_1,\dots,X_N, SOC_f) \in \mathcal{X} \right\}$, and the function $C_P\colon \mathbb{R}^N \to \mathbb{R}$ is defined by

$$C_P(X_1,\dots,X_N) = C(SOC_0, X_1,\dots,X_N, SOC_f).$$

Since the set $\mathcal{X}_P$ is an open set, we can solve the unconstrained problem

$$\min_{(X_1,\dots,X_N)\in\mathbb{R}^N} C_P(X_1,\dots,X_N) \tag{22}$$

and verify *a posteriori* that the solution belongs to $\mathcal{X}_p$. Problem (22) is numerically solved with the trust-ncg algorithm from 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. Due to the complexity of the industrial problem and the use of neural networks, Problem (Macro) is likely to be non-convex, and thus a parallel multistart strategy is adopted to increase the robustness of our method.

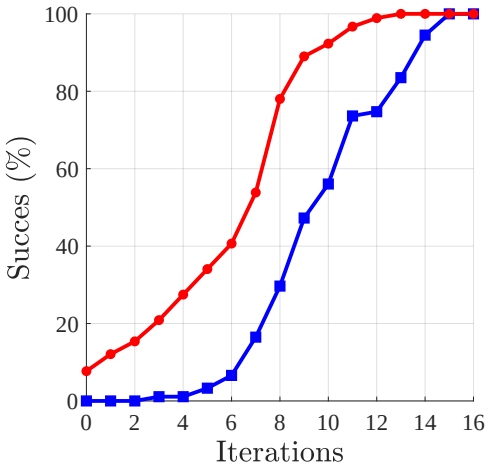
Remark 7. *In all the experiments, we have checked that the obtained numerical solutions of Problem (22) belong to the admissible set $\mathcal{X}_P$. We have to check this because the extrapolations of the neural networks $C_i(X_i, X_{i+1})$ when $X_{i+1} \notin A(t_{i+1}, t_i, X_i)$ are not controlled, and so it may exist a local (even global) minimum of Problem (22) outside of $\mathcal{X}_P$. If this case had happened, we could have handled it using a diffeomorphism $\phi\colon \mathbb{R}^n \to \mathcal{X}_P$ by replacing the minimization of C_p on $\mathcal{X}_P$ by the minimization of $C_p \circ \phi$ on $\mathbb{R}^n$.*

Micro problem. Optimal control problems of Problem (Micro) 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 $S_i(p) = \pi_x\big(\exp_{\overrightarrow{H}}(t_{i+1}, t_i, (X_i, p))\big) - X_{i+1}$. Since these optimal control problems are independent from each other, they are solved in parallel.

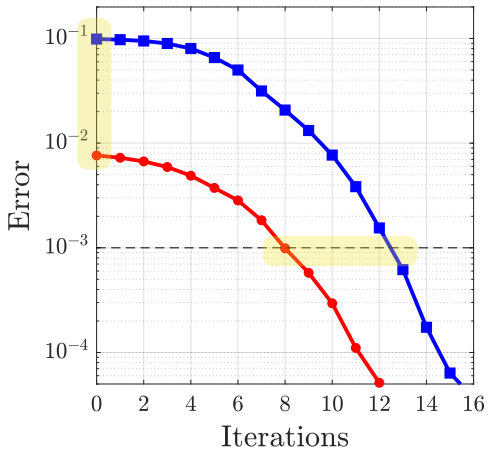
4.3 Numerical experiments

A first objective of the numerical experiments is to evaluate the benefits of the initialization suggested by Equation (17). We observe in Figure 7 that the convergence of S_0 is significantly improved by this choice compared to a fixed but well-chosen initialization, in the sense that this fixed initialization belongs to $[p_0^-, p_0^+]$. In particular, as highlighted in yellow in Figure 7b, this initial guess is in average more than 10 times better than the fixed one which improves the robustness of the Macro–Micro method. Moreover, the proposed initial guess reduces the number of iterations of the solver to reach a zero of the shooting function, which reduces the computation time and the number of computations of the proposed Macro–Micro method.

The second objective is to compare the Macro–Micro method to two other methods. The first one is the simple shooting described in Section 2.4, used as reference. The second other method is called *empiric* and aims to evaluate the relevance of Problem (Macro). 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.* $X \notin \mathcal{X}$, then X is projected on the boundary of $\mathcal{X}$. Then, Problem (Micro) is solved with the same simple shooting method as for the bilevel one. Figure 8 compares the state trajectories obtained with the simple shooting, the empiric and the Macro–Micro methods for 5 different initial and final states.



(a) Percentage of success of $S_0 = 0$ resolution w.r.t the number of iterations of the solver.



(b) Error $|S_0(\cdot)|$ w.r.t the number of iterations of the solver.

Figure 7: Evolution of percentage of success (Figure 7a) and of the error (Figure 7b) with respect to the number of iterations of the solver. The blue line with squares represents the evolution for a fixed initialization ($p = 500$) and the red line with disks for the initialization suggested by Equation (17). These data result from 100 shooting functions with random initial and final admissible states. The resolution is considered successful when the norm of the shooting function is smaller than a given threshold of 10^{-3}, which correspond to the black dashed line in Figure 7b.

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 Macro–Micro method is much closer to the simple shooting one than with the empiric method. Furthermore, the average error of Macro–Micro method is more than 9 times smaller than the empiric one in terms of fuel consumption.

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

Table 2: Comparison of the Macro–Micro and empiric methods for the 5 experiments defined by the first two columns. The Macro–Micro 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.

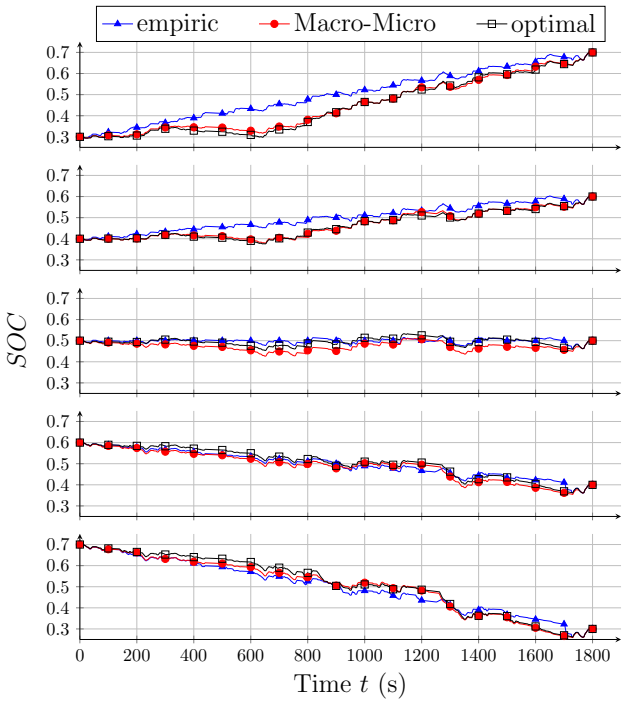


Figure 8: State trajectories for various initial SOC_0 and final SOC_f states. The black line with squares corresponds to the optimal trajectory, calculated by simple shooting. Red line with disks corresponds to the Macro–Micro method and blue line with triangles to the empiric one. The red and blue markers are the intermediate points, respectively for Macro–Micro and empiric methods.

5 Conclusion and perspectives

Conclusion. A novel method to solve optimal control problems, called Macro–Micro, has been presented. This new method has been applied to the HEVs torque split and gear shift optimal control problem, which is a complex industrial problem.

1. **Robustness.** As shown by Proposition 2 and Figure 1, this method is closely related to multiple shooting, thanks to a new path between (OCP) and (MPBVP). Multiple shooting is known to be more robust than simple shooting, due to the reduction of the time interval integration. For the same reason, the Macro–Micro method is at least as robust as multiple shooting and so more robust than simple shooting. Moreover, in the considered application, the proposed initialization of the shooting function given by Equation (17) provides in average a more than 10 times better initial guess than a fixed but well-chosen one, as shown in Figure 7, which further increase the robustness of the Macro–Micro method.

2. **Computation time.** As illustrated in Figure 2, the construction of the approximations of the value functions is an off-line task. Even if the construction of the database was done by an efficient method compared to the classical method [15], this step and the training of the neural networks is a long task compared to the resolution of (OCP) by shooting methods. However, motivated by an embedded point of view, these approximations are considered to be known, and their constructions are not a part of the Macro–Micro method. As discussed in Section 3.3.3, the computation time of the Macro–Micro can be divided by a factor $N + 1$ compared to simple shooting method. Moreover, as shown in Figure 7, the proposed initialization given by Equation (17) reduces the number of iterations of the solver, which further reduces the computation time.

3. **Computation cost.** Problem (Micro) is composed of $N+1$ independent optimal control problems. As discussed in 3.3.3, this characteristic leads to divide by a factor $N+1$ the number of computations to obtain the current control with the proposed method compared to classical indirect methods. Moreover, the proposed initial guess leads to less number of iterations of the solver and so a smaller computational cost.

4. **Optimality.** Due to the approximation of the value function, the proposed Macro–Micro method is sub-optimal. However, we have shown in Section 3.3.4 that the error of the proposed method can be controlled by the error of the approximations. Moreover, in the considered application, the error of the Macro–Micro is close to the optimal one as shown in Table 2: a difference of 0.88 grams of fuel on average on the 5 considered experiments. Furthermore, the comparison of the proposed method to the empiric one in terms of optimality shows the benefit of the (Macro) resolution.

Perspectives.

1. **ECMS.** The considered torque split and gear shift optimal control problem of HEVs is a well-studied application in the literature. As discussed in the introduction, the ECMS is a method developed for this application. A comparison of the proposed Macro–Micro method with ECMS could be an interesting future work. Moreover, the idea of the bilevel decomposition could be applied to the ECMS, and could lead to a new interesting version of this method. In particular, this version could benefit from the good initialization developed in this article to initialize the equivalent factor.

2. **Hybrid optimal control problem.** 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 ($T \leq 90°C$) and a steady state ($T = 90°C$).

3. **Convolutional Neural Network.** 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 neural 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 create the ad-hoc database and to train such a network.

4. **Others research fields.** Finally, it might be really interesting to look for other applications where the proposed method could be applied. Since the Macro–Micro was developed on a large class of optimal control problems formulated as (OCP), this method could be applied to other optimal control problems from other research fields, such as aeronautic, agronomy, or spatial for instance.

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] J.-B. Caillau, R. Ferretti, E. Trélat, and H. Zidani, *An algorithmic guide for finite-dimensional optimal control problems*, in Handbook of numerical analysis: Numerical control, Part B, North-Holland, 2022.

[11] 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.

[12] 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.

[13] F. Clarke, *Functional Analysis, Calculus of Variations and Optimal Control*, Springer London, 2013.

[14] 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.

[15] O. Cots, R. Dutto, S. Laporte, and S. Jan, *Generation of value function data for bilevel optimal control and application to hybrid electric vehicle*, 2023.

[16] 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.

[17] S. Delprat, *Evaluation de stratégies de commande pour véhicules hybrides paralleles*, PhD thesis, Université de Valenciennes et du Hainaut-Cambresis, 2002.

[18] J. Gergaud and T. Haberkorn, *Homotopy method for minimum consumption orbit transfer problem*, ESAIM - Control Optim. Calc. Var., 12 (2006).

[19] I. H. A. Hamza, *Commande Prédictive optimale temps-réel, appliquée au contrôle de véhicules automobiles hybrides connectés à leurs environnements*, PhD thesis, INPT, 2018.

[20] M. Hedon, *Modeling and Simulation of a Hybrid Powertrain*, Master's thesis, School of Electrical Engineering and Computer Science (EECS), 2018.

[21] 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.

[22] 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.

[23] 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.

[24] 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.

[25] 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.

[26] L. S. Pontryagin, V. G. Boltyanskii, R. V. Gamkrelidze, E. F. Mishechenko, *The Mathematical Theory of Optimal Processes*, New York, (1962).

[27] A. Le Rhun, *Stochastic optimal control for the energy management of hybrid electric vehicles under traffic constraints*, PhD thesis, Université Paris Saclay (COmUE), 2019.

[28] A. A. Malikopoulos, *Supervisory Power Management Control Algorithms for Hybrid Electric Vehicles: A Survey*, IEEE Trans. Intell. Transp. Syst., 15 (2014), pp. 1869–1885.

[29] P. Martinon and J. Gergaud, *Using switching detection and variational equations for the shooting method*, Optim. Control Appl. Methods, (2007).

[30] Y. Milhau, D. Sinoquet, and G. Rousseau, *Design optimization and optimal control for hybrid vehicles*, Optim. Eng., 12 (2011), pp. 199–213.

[31] 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.

[32] G. Paganelli, S. Delprat, T.-M. Guerra, J. Rimaux, and J.-J. Santin, *Equivalent consumption minimization strategy for parallel hybrid powertrains*, in IEEE 55th Vehicular Technology Conference, vol. 4, 2002, pp. 2076–2081.

[33] 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.

[34] A. Rao, *A Survey of Numerical Methods for Optimal Control*, Adv. Astronaut. Sci., 135 (2010).

[35] A. Rao and K. Mease, *Dichotomic basis approach to solving hyper-sensitive optimal control problems*, Automatica, 35 (1999), pp. 633–642.

[36] H. Schättler and U. Ledzewicz, *Geometric optimal control: theory, methods and examples*, vol. 38, Springer, 2012.

[37] 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).

[38] 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).

[39] 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.

[40] J. Stoer and R. Bulirsch, *Introduction to Numerical Analysis*, Springer New York, 2002.

[41] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*, The MIT Press, 2018.

[42] E. Tate and S. Boyd, *Finding Ultimate Limits of Performance for Hybrid Electric Vehicles*, SAE Trans., 109 (2000), pp. 2437–2448.

[43] 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.

[44] 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.

[45] 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).

[46] 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

$$\nabla_a V_*\big(x(t_0), x(t_f)\big) = -p(t_0), \tag{23}$$

$$\nabla_b V_*\big(x(t_0), x(t_f)\big) = p(t_f). \tag{24}$$

Equation (23) is a classical result that can be found in [6, 8, 11, 14, 36]. We shall use this result to prove (24). For this purpose, we transform (OCP_*) into (ROCP_*) using the reverse time transformation $\phi\colon [t_0, t_f] \to [t_0, t_f]$ defined by $\phi(t) = t_f + t_0 - t$:

$$(\mathrm{ROCP}_*) \qquad \begin{cases} V_R(b,a) = \min\limits_{\hat{x},\hat{u}} \displaystyle\int_{t_0}^{t_f} f^0(\phi(t), \hat{x}(t), \hat{u}(t))\,\mathrm{d}t, \\ \qquad\qquad \text{s.t. } \dot{\hat{x}}(t) = -f(\phi(t), \hat{x}(t), \hat{u}(t)), & t \in [t_0, t_f] \text{ a.e.}, \\ \qquad\qquad\quad \hat{u}(t) \in \mathrm{U}(\phi(t)), & t \in [t_0, t_f], \\ \qquad\qquad\quad \hat{x}(t_0) = b, \quad \hat{x}(t_f) = a. \end{cases}$$

We have naturally the following relation between the value functions:

$$V_R(b,a) = V_*(a,b). \tag{25}$$

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 (OCP_*) is equivalent to (ROCP_*) in the sense that

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

Since (ROCP_*) has the same form as (OCP_*) and the value function V_R is differentiable at (b,a), we can apply (23) to (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 (25), we get $\nabla_b V_*(a,b) = \nabla_b V_R(b,a) = p(t_f)$. □

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_{qEM}	EM torque	N.m
T_{qW}	Wheels requested torque	N.m
$T_{qW\,ICE}$	Part of the wheels torque produced by the ICE	N.m
$T_{qW\,EM}$	Part of the wheels torque produced by the EM	N.m
$T_{qL\,ICE}$	Transmission torque losses from the ICE	N.m
$T_{qL\,EM}$	Transmission torque losses from the EM	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 $Gear$)	
R_e	Ratio between main and EM shafts	
Q	Battery charge	C
T	Coolant temperature	°C

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