



**HAL**  
open science

# Generation of value function data for bilevel optimal control and application to hybrid electric vehicle

Olivier Cots, Rémy Dutto, Sophie Jan, Serge Laporte

► **To cite this version:**

Olivier Cots, Rémy Dutto, Sophie Jan, Serge Laporte. Generation of value function data for bilevel optimal control and application to hybrid electric vehicle. 2023. hal-04359905v2

**HAL Id: hal-04359905**

**<https://hal.science/hal-04359905v2>**

Preprint submitted on 9 Jul 2024

**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.

# Generation of value function data for bilevel optimal control and application to hybrid electric vehicle

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

July 9, 2024

## Abstract

In this article, we present two numerical methods to create a database for the approximation of the value function of a bilevel optimal control problem. The first method is based on the computation of the value function via indirect simple shooting, which implies to find the zeros of functions. The second one amounts to solve Cauchy problems. These two techniques are compared, in terms of prior information, computation cost and data distribution, on an industrial application: the torque split and gear shift optimal control problem on hybrid electric vehicles.

**Keywords.** Database generation, Bilevel optimal control, Pontryagin maximum principle, Indirect method, Value function, Hybrid electric vehicle

## 1 Introduction

The value function is a central object in optimal control theory, that describes how the optimal cost depends on the initial conditions. This function was studied in the 1950's by Richard Bellman and leads to the Hamilton-Jacobi-Bellman (HJB) partial differential equations. These equations provide necessary and sufficient conditions for an optimal control problem, as well as the optimal control in a feedback form. They are the baseline of dynamic programming [3] and reinforcement learning [14], which are two of the main optimal control methods used in industrial applications. However, these methods are subject to the curse of dimensionality, which is a key numerical issue for embedded solutions.

The Pontryagin maximum principle [11] also introduced in the 1950's gives necessary optimality conditions for optimal control problems, and leads to the indirect numerical methods, which promise to be accurate and fast enough to be used for embedded solutions. For a description of classical numerical methods for optimal control problems, see [5].

A new method based on a bilevel decomposition of the optimal control problem, presented in [6], promises to reduce the number of computations and to be fast enough for embedded solution. This method uses some value functions between fixed times, which can rarely be calculated explicitly, and the strategy is to approximate them from a set of precomputed data. The main objective of this paper is to propose and compare two methods to create the associated database. The first method is based on the computation of the value function on a given grid thanks to the resolution of the shooting equations obtained from the Pontryagin maximum principle, while the second method amounts to evaluate the value function on a computed grid thanks to the integration of the underlying Hamiltonian flow.

---

\*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).

The article is organized as follows. We first introduce the motivations in Section 2 based on the bilevel optimal control method. Then we present the two methods to create the value function database in Section 3. We compare in Section 4 the numerical results we have obtained when applying these two methods to an industrial application: the torque split and gear shift optimal control of a hybrid electric vehicle. Finally, Section 5 concludes the article.

## 2 Motivations

This paper is motivated by the torque split and gear shift optimal control problem described in [6] and used for the numerical results in Section 4. Another motivation is the resolution of this problem by the bilevel optimal control method, fully detailed in [6] and briefly presented in Section 2.2, which is robust, fast and that can be embedded. One crucial feature of this method is the computation of approximations, denoted  $C_i(a, b)$  hereafter, of value functions, denoted  $V_i(a, b)$  and representing the optimal cost to transfer the state from the initial condition  $a$  to the target  $b$ . In the same reference [6], the cost approximations  $C_i$  are modeled by neural networks, trained on specific databases. The goal of this paper is to explain the method (referred as Method 2 in the following) used to create these databases in [6], but also to compare it to a more intuitive but naive approach (referred as Method 1 in the following).

### 2.1 Optimal control problem

We consider the following optimal control problem in Bolza form:

$$\begin{aligned}
 \text{(OCP)} \quad & \begin{cases} \min_{x,u} \int_{t_0}^{t_f} f^0(t, x(t), u(t)) \, dt + g(x(t_f)), & (1) \\ \text{s.t. } \dot{x}(t) = f(t, x(t), u(t)), & t \in [t_0, t_f] \text{ a.e.}, & (2) \\ u(t) \in U, & t \in [t_0, t_f], & (3) \\ x(t_0) = x_0, & & (4) \end{cases}
 \end{aligned}$$

where the function  $f^0: \mathbb{R} \times \mathbb{R}^n \times \mathbb{R}^m \rightarrow \mathbb{R}$ , the terminal cost  $g: \mathbb{R}^n \rightarrow \mathbb{R}$ , and the state dynamic function  $f: \mathbb{R} \times \mathbb{R}^n \times \mathbb{R}^m \rightarrow \mathbb{R}^n$  are of class  $\mathcal{C}^1$ . The initial and final times  $t_0 < t_f$  are fixed, as well as the initial state  $x_0 \in \mathbb{R}^n$ . The control domain  $U \subset \mathbb{R}^m$  is a nonempty set. Solving (OCP) consists in finding an absolutely continuous<sup>1</sup> state  $x \in \text{AC}([t_0, t_f], \mathbb{R}^n)$  and an essentially bounded<sup>2</sup> control  $u \in L^\infty([t_0, t_f], \mathbb{R}^m)$ , which minimize the cost (1) and satisfy the constraints (2), (3) and (4).

**Remark 1.** *For conciseness and clarity, we consider an optimal control problem in this Bolza form but our approach can be easily extended to a more general problem with fixed final state or mixed limit conditions for instance.*

### 2.2 Bilevel optimal control method

Considering intermediate times  $t_0 < t_1 < \dots < t_N < t_{N+1} = t_f$ , the optimal control problem (OCP) can be decomposed into the following bilevel optimal control problem

$$\text{(BOCP)} \quad \begin{cases} \min_X \sum_{i=0}^N V_i(X_i, X_{i+1}) + g(X_{N+1}), \\ \text{s.t. } X = (X_0, \dots, X_{N+1}) \in \mathcal{X}, \quad X_0 = x_0, \end{cases}$$

<sup>1</sup>AC( $[t_0, t_f], \mathbb{R}^n$ ) is the set of absolutely continuous functions on  $[t_0, t_f]$  valued in  $\mathbb{R}^n$ .

<sup>2</sup>L $^\infty$ ( $[t_0, t_f], \mathbb{R}^m$ ) is the set of essentially bounded functions on  $[t_0, t_f]$  valued in  $\mathbb{R}^m$ .

where for all  $i \in \mathbb{N}_N := \{0, \dots, N\}$ , the functions  $V_i$  are the *intermediate value functions*: for any admissible<sup>3</sup> pair  $(a, b)$ ,  $V_i(a, b)$  is the optimal cost of

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

The set  $\mathcal{X}$  appearing in (BOCP) is defined as follows:  $X \in \mathcal{X} \subset (\mathbb{R}^n)^{N+2}$  if, for all  $i \in \mathbb{N}_N$ , the pair  $(X_i, X_{i+1})$  is admissible for (OCP <sub>$i, X_i, X_{i+1}$</sub> ). As detailed in [6], Problem (BOCP) can be seen as a bilevel optimal control problem [2].

In many applications, the intermediate value functions  $V_i$  cannot be computed explicitly. Thus, it has been proposed in [6] to replace them by approximations denoted  $C_i$ . This leads to a hierarchical method in two steps, called Macro-Micro. The first step consists in solving the finite dimensional optimization problem

$$(Macro) \quad \left\{ \begin{array}{l} \min_X \sum_{i=0}^N C_i(X_i, X_{i+1}) + g(X_{N+1}), \\ \text{s.t. } X \in \mathcal{X}, \quad X_0 = x_0, \end{array} \right.$$

to get the intermediate state  $\hat{X} = (\hat{X}_0, \dots, \hat{X}_{N+1})$ . The second step is to solve  $N + 1$  independent optimal control problems of the following Lagrange form

$$(Micro) \quad \left\{ \begin{array}{l} \min_{x,u} \int_{t_i}^{t_{i+1}} f^0(t, x(t), u(t)) dt, \\ \text{s.t. } \dot{x}(t) = f(t, x(t), u(t)), \quad t \in [t_i, t_{i+1}] \text{ a.e.}, \\ u(t) \in U, \quad t \in [t_i, t_{i+1}], \\ x(t_i) = \hat{X}_i, \quad x(t_{i+1}) = \hat{X}_{i+1}, \end{array} \right.$$

to get the state and command trajectory on each sub-interval  $[t_i, t_{i+1}]$ . This method is sub-optimal, due to the approximation error between  $C_i$  and  $V_i$ , but it promises to be faster and to require less computations than classical methods, which is particularly interesting for embedded applications.

However, before applying the Macro-Micro method, we need to build the approximation functions  $C_i$ . We postpone to Section 3 the creation of the optimal values database which will be used for that purpose and which is the main contribution of this article. From now on, we consider one optimal control problem of the Problem (Micro), for a given fixed  $i \in \mathbb{N}_N$ . The next section gives some classical details about the so-called simple indirect shooting method to solve it.

## 2.3 Indirect method

The considered optimal control problem of Problem (Micro) may be solved by the classical indirect simple shooting method which is based on the Pontryagin maximum principle [11]. The core function of this principle is the *pseudo-Hamiltonian*

$$h(t, x, p, u) = -f^0(t, x, u) + (p \mid f(t, x, u)),$$

<sup>3</sup>The pair  $(a, b)$  is admissible for (OCP <sub>$i,a,b$</sub> ) if  $b$  is reachable at time  $t_{i+1}$  from  $a$  at  $t_i$ .

where  $(\cdot | \cdot)$  stands for the usual scalar product in  $\mathbb{R}^n$ . For a given initial state and costate couple  $z_i = (x_i, p_i) \in \mathbb{R}^n \times \mathbb{R}^n$ , let us define the problem of finding the state and costate trajectory  $z(\cdot)$  solution of

$$\begin{cases} \dot{z}(t) = \vec{h}(t, z(t), u(t)), & t \in [t_i, t_{i+1}] \text{ a.e.}, \\ h(t, z(t), u(t)) = \max_{w \in U} h(t, z(t), w) & t \in [t_i, t_{i+1}] \text{ a.e.}, \\ z(t_i) = z_i, \end{cases} \quad (5)$$

where  $\vec{h}$  is the *pseudo-Hamiltonian vector field* defined by

$$\vec{h}(t, (x, p), u) = \left( \nabla_p h(t, (x, p), u), -\nabla_x h(t, (x, p), u) \right).$$

If we assume that the *exponential map*  $\exp_{\vec{h}}: (t_{i+1}, t_i, z_i) \mapsto z(t_{i+1})$  is a well-defined application, where  $z(\cdot)$  is a solution of Problem (5), then the maximum principle leads to the resolution of the two-points boundary value problem

$$\text{(TPBVP)} \quad \begin{cases} (x_{i+1}, p_{i+1}) = \exp_{\vec{h}}(t_{i+1}, t_i, (x_i, p_i)), \\ x_i = \hat{X}_i, \quad x_{i+1} = \hat{X}_{i+1}. \end{cases}$$

To use simple notation, let us define  $\bar{x}_{i+1}(z_i)$  by

$$\bar{x}_{i+1}(z_i) = \pi_x \left( \exp_{\vec{h}}(t_{i+1}, t_i, z_i) \right), \quad (6)$$

where  $\pi_x(x, p) = x$  is the projection on the state space.

In the following, we assume that for all initial and final admissible state  $(a, b)$ , if  $(x, u)$  is a solution of  $(\text{OCP}_{i,a,b})$ , then there exists  $p \in \text{AC}([t_i, t_{i+1}], \mathbb{R}^n)$  such that the pair  $(x, p)$  is a solution of  $(\text{TPBVP})$ . Such a pair solution of Problem (5) is called an *extremal*. A solution of  $(\text{TPBVP})$  is an extremal that satisfies the boundary conditions, and it is called a *BC-extremal*. This assumption is linked to the normality of the BC-extremals associated to the solution (cf. [1] for more details). For the sake of simplicity, we denote by  $\bar{c}_i(z_i)$  the cost of the state and command trajectory, given by the solution of Problem (5), with the initial state and costate conditions  $z_i = (x_i, p_i)$ . We also consider the following hypothesis, that will be numerically verified in our application.

**Hypothesis 1.** *Let  $(x_i, x_{i+1})$  be a given initial and final admissible state. Then, all the zeros of*

$$p_i \mapsto \bar{x}_{i+1}(x_i, p_i) - x_{i+1} \quad (7)$$

*are associated to the same state and command trajectory  $(x, u)$ , depending on the pair  $(x_i, x_{i+1})$ .*

**Remark 2.** *Under the previous assumptions and Hypothesis 1, for all initial and final admissible state  $(a, b)$ , there exists at most one solution of  $(\text{OCP}_{i,a,b})$ .*

**Remark 3.** *Hypothesis 1 comes from our application but is not necessary to apply the two proposed methods described in Section 3. However, it simplifies their descriptions and helps to understand their main characteristics. In fact, consider another application where it does not hold, namely where there are several zeros of (7) associated to different state-control pairs. If we still have a unique solution of  $(\text{OCP}_{i,a,b})$ , we have to add another step to the two proposed methods which consists simply of keeping the zero with the lowest cost. On the other hand, if there are several solutions, we can either pick one or keep all since it is not an issue to have redundant points in the database.*

The indirect simple shooting method aims to solve  $(\text{TPBVP})$  by finding a zero of the *shooting function*

$$S_{i, \hat{X}_i, \hat{X}_{i+1}}(p_i) = \bar{x}_{i+1}(\hat{X}_i, p_i) - \hat{X}_{i+1}. \quad (8)$$

Thanks to Hypothesis 1, all the zeros of the shooting function lead to the same optimal cost, and therefore only one of them need to be found. On the opposite, without this hypothesis, it would be necessary to find all the zeros and to compare them in terms of cost to get the optimal one.

Equation (8) is solved using a classical Newton-like solver, and it is well known that finding a good initial guess for the costate is a critical issue. However, given a solution  $X^*$  of (BOCP), if  $V_i$  is differentiable at  $(X_i^*, X_{i+1}^*)$ , it is established [4] that the vector  $-\nabla_a V_i(X_i^*, X_{i+1}^*)$  is a zero of  $S_{i, X_i^*, X_{i+1}^*}$ . Unfortunately,  $V_i$  is not known, but since  $C_i$  is an approximation of  $V_i$ , the vector  $-\nabla_a C_i(\hat{X}_i, \hat{X}_{i+1})$  could be a good initial guess for the associated optimal control problem of Problem (Micro) shooting function [8]. This initial guess is used with a geometric preconditioning method of the shooting function on the proposed application in [7].

Now we have described how to solve the optimal control problems of Problem (Micro), we will focus on the creation of a database for the computation of  $C_i$ , which is the main contribution of this paper.

### 3 Data generation

In this section, we shall describe the creation of a database  $\mathbb{D}_i$  containing optimal values of  $(\text{OCP}_{i, x_i, x_{i+1}})$  for various pairs  $(x_i, x_{i+1})$ :

$$\mathbb{D}_i \subset \left\{ (x_i, x_{i+1}, c) \mid (x_i, x_{i+1}) \text{ admissible and } V_i(x_i, x_{i+1}) = c \right\}.$$

Two methods are presented to achieve this goal. They are compared in terms of prior information, computation cost and data distribution.

#### 3.1 Method 1

The first method consists in computing the value function  $V_i$  on a given grid by solving optimal control problems thanks to the simple shooting method. This is the intuitive approach compared to the second method presented in the next section. Let  $\mathbb{X}_i$  be a discretization of admissible initial and final states. For all  $(x_i, x_{i+1}) \in \mathbb{X}_i$ , we solve the optimal control problem  $(\text{OCP}_{i, x_i, x_{i+1}})$  by an indirect simple shooting method and store the corresponding optimal value  $V_i(x_i, x_{i+1})$  in  $\mathbb{D}_i$ . The main complexity comes from the fact we have to solve equations which essentially amounts to invert the exponential map given in (6). Algorithm 1 provides an overview of these steps.

---

#### Algorithm 1 Method 1

---

**Require:**  $\mathbb{X}_i$

**Ensure:**  $\mathbb{D}_i$

**for all**  $(x_i, x_{i+1}) \in \mathbb{X}_i$  **do**

$p_i \leftarrow \text{solve}(S_{i, x_i, x_{i+1}}(p_i) = 0)$  ▷ Newton solver

$c \leftarrow \bar{c}_i(x_i, p_i)$  ▷ Optimal cost

$\mathbb{D}_i.\text{append}(x_i, x_{i+1}, c)$  ▷ Storage

**end for**

---

**Prior information** First of all, the set  $\mathbb{X}_i$  needs to be created as follows: the initial state is first discretized, then for each of these values, the set of associated reachable final states has to be determined and discretized. This ensures that the space of initial and final state is fully explored and that all pairs  $(x_i, x_{i+1}) \in \mathbb{X}_i$  are admissible, which prevents the Newton solver to fail. The determination of the reachable final states is a difficulty in this method. See the remark page 7 for details about how we compute this set of reachable states in our application.

**Computation cost** For each iteration of the Newton solver used to find a zero of the shooting function, at least one integration of the Hamiltonian vector field is required.

**Data distribution** By construction, the distribution of the pairs  $(x_i, x_{i+1})$  is fully controlled inside the admissibility set. For example a uniform distribution may be chosen for an easy interpolation.

## 3.2 Method 2

The second method, described in Algorithm 2, is based on a discretization  $\mathbb{Z}_i$  of the initial state and costate space. For all  $z_i = (x_i, p_i) \in \mathbb{Z}_i$ , the exponential map  $z_{i+1} = (x_{i+1}, p_{i+1}) = \exp_{\vec{h}}(t_{i+1}, t_i, z_i)$  is computed, and Hypothesis 1 ensures that  $V_i(x_i, x_{i+1}) = \bar{c}_i(x_i, p_i)$ . Compared to Method 1, here we only perform a direct computation of the exponential map  $\exp_{\vec{h}}$  instead of inverting it.

---

### Algorithm 2 Method 2

---

**Require:**  $\mathbb{Z}_i$

**Ensure:**  $\mathbb{D}_i$

**for all**  $z_i = (x_i, p_i) \in \mathbb{Z}_i$  **do**

$x_{i+1} \leftarrow \bar{x}_{i+1}(z_i)$

    ▷ Final state

$c \leftarrow \bar{c}_i(z_i)$

    ▷ Optimal cost

$\mathbb{D}_i.append(x_i, x_{i+1}, c)$

    ▷ Storage

**end for**

---

**Prior information** Since there are no admissibility constraints on  $(x_i, p_i)$ , the set  $\mathbb{Z}_i$  can be the cartesian product between discretization of initial state and costate. Nevertheless, the range of the costate discretization must be well chosen in order to ensure a full exploration of the reachable states  $x_{i+1}$  while limiting the number of useless calls of the exponential map.

**Computation cost** In contrast to Method 1, no (Newton) solver is involved here. Hence, only one evaluation of  $\exp_{\vec{h}}$  is needed to compute a point of  $\mathbb{D}_i$ .

**Data distribution** In this method, the distribution of the reached final states  $x_{i+1}$  is no more controlled. The main consequence is the risk of empty zones. Fortunately, we shall see in Section 4 that, in our application, a sufficient density in the set  $\mathbb{Z}_i$  ensures a satisfactory distribution of reached states  $x_{i+1}$ .

## 4 Results

**Presentation of the numerical experiments** The considered application is the optimal control of the torque split and the gear shift of a Hybrid Electric Vehicle (HEV) on the *Worldwide harmonized Light vehicles Test Cycle* (WLTC). The goal is to minimize the fuel consumption of the Internal Combustion Engine (ICE) for fixed initial condition and fixed or free final state of charge of the battery by acting on the gear and on the torque split between the ICE and the Electric Motor (EM). We refer to [9, 10, 12, 13, 15, 16] for a general presentation of this kind of applications.

**Remark 4.** *The model of the considered application is given by an industrial code developed by the company Vitesco Technologies, which uses tabulated data to model complex functions as transmission torque losses or fuel consumption of the ICE. We refer to [6] for a more detailed presentation of this specific nonlinear model.*

This problem can be formulated as (OCP) and we consider that we stand in the frame of the assumptions stated in Section 2.3. The bilevel method can be applied in this context, and the time interval (1800 seconds of the WLTC) is arbitrary decomposed into 18 time sub-intervals of 100s.

The goal of this section is to compare the two methods presented in Sections 3.1 and 3.2 on our HEV application. For this purpose and to clarify the presentation of the results, we only focus on the first sub-interval  $[0, 100]$  and we fix the initial state, denoted  $x_0$ , to 0.5. So next, we apply the two methods to create the database  $\mathbb{D}_0$ .

**Validation of Hypothesis 1** The evolution of the function  $p_0 \mapsto \bar{x}_1(x_0, p_0)$  is shown in Figure 1. It can be noticed that for all  $p_0 \in [p_0^-, p_0^+]$ , the function (6) is injective. Moreover, for all  $p_0 \leq p_0^-$  (respectively  $p_0 \geq p_0^+$ ), we numerically verified that the extremal resulting from the exponential map is associated to a unique  $(x, u)$ . Hence, the Hypothesis 1 can be considered numerically valid.

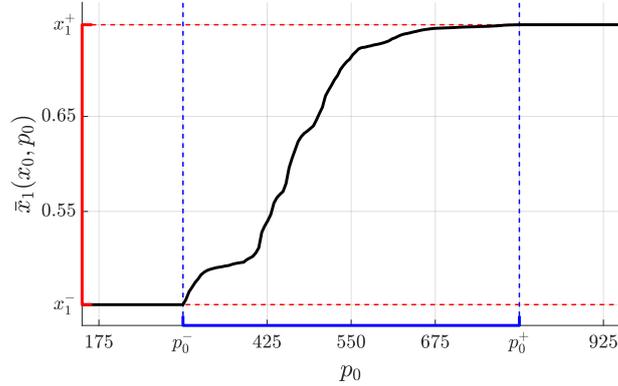


Figure 1: Evolution of the function (6), with a fixed initial state  $x_0 = 0.5$ . The red interval on the y-axis corresponds to the admissible final state set at time  $t_1$ , and the blue interval on the x-axis corresponds to the useful costate initialization set, mentioned in the Prior Information paragraph of Method 2.

**Remark 5.** *As discussed in Section 3, the final states must be chosen in  $[x_1^-, x_1^+]$  for Method 1. In contrast, for Method 2 they naturally belong to this interval, but although the initial costate can be taken in  $\mathbb{R}^n$ , its useful range is limited to the interval  $[p_0^-, p_0^+]$ , cf. Figure 1. In our application, the bounds  $x_1^-$  (respectively  $x_1^+$ ) can be evaluated by simulation, fixing the command  $u = 0$  (respectively the command that maximizes  $\dot{x}(t)$  at each time step).*

**Results analysis** In order to compare both methods in terms of number of calls to the exponential map, 475 points  $(x_i, x_{i+1}, c)$  have been generated for various initial and final admissible states with Method 1. We observe that the Newton solver needs in average 11.2 iterations to find a zero, which means that this method requires at least 11.2 times more calls to  $\exp_{\vec{h}}$  than Method 2. Hence, the computation time needed to obtain a point with Method 1 is at least 11.2 times longer than with Method 2.

The two methods are now compared in terms of distribution of the final states. First, 10 points of  $\mathbb{D}_0$  are generated by each method. As expected, we can observe in Figures 2a and 2b that the final states are obviously uniformly distributed with Method 1, while there are not with Method 2. Second, for at least the same computational cost as Method 1, 112 points of  $\mathbb{D}_0$  are generated by Method 2. The results are shown in Figure 2c. It can be observed that if the distribution remains non-uniform, the coverage of the reachable interval  $[x_1^-, x_1^+]$  is much better than the one obtained by Method 1.

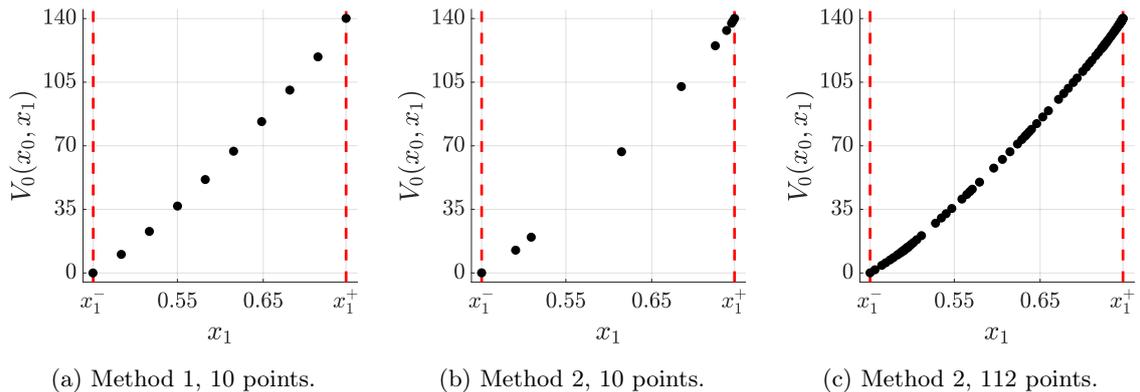


Figure 2: Value function data created by the two proposed methods. The black points correspond to the computed points by the methods.

**Remark 6.** *These results have been presented on the first sub-interval of 100 seconds of the WLTC for a unique initial state  $x_0 = 0.5$ . Note that Method 2 has been implemented in [6] on all the 18 sub-intervals of the WLTC on a fine discretization of initial states and costates. Method 2 has been preferred essentially because it is fast and there is no equation to solve, just numerical integration to perform. In [6], this torque split and gear shift optimal control problem has been solved by the previously described bilevel method with neural networks  $C_i$  to approximate the value functions  $V_i$ , trained on databases created by Method 2.*

## 5 Conclusion

The Macro-Micro method proposed in [6] requires the approximation of value functions which can be performed using a database composed of points  $(a, b, V_i(a, b))$ . We present two numerical methods to create such a database. The first one, based on the computation of the value functions  $V_i(a, b)$  for a set of given pairs  $(a, b)$ , requires to find an initial costate  $p'$  solution of the shooting equation that appears in the indirect method. The second method consists in evaluating the exponential map  $\exp_{\vec{h}}$  for a set of given pairs  $(a, p)$ , where  $p$  is the initial costate, to get a final state  $b'$  and the corresponding cost  $\bar{c}_i(a, p)$ , which under Hypothesis 1, is the optimal one:  $V_i(a, b') = \bar{c}_i(a, p)$ .

The pros and cons of both methods are gathered in Table 1. The uncontrolled distribution obtained by Method 2 can be balanced by its favorable computation cost. Indeed, in our application, for at least the same number of calls to the exponential map as in Method 1, it appears that the domain of reachable final states is well covered. This makes this Method 2 really competitive.

	Method 1	Method 2
Pros	Controlled distribution	Computation cost
Cons	Prior knowledge of admissible domain Computation cost	Prior knowledge of initial costate range Non controlled distribution

Table 1: Pros and Cons of Method 1 and Method 2.

Once the database is created, the associated value function approximation  $C_i$  has to be built. A wide range of methods can be used such as neural networks models. The latter are renowned for their generalization capacities and their ability to provide the gradient of the approximation  $C_i$  through retropropagation technique. Indeed, the availability of this gradient should ease the resolution of Problem (Macro) and provide a good initialization for the resolution of the optimal control problems of Problem (Micro) by indirect method.

Another perspective is to use the gradient of the value function to improve the fitting of the approximations  $C_i$ . Indeed, it can be shown under some assumptions that the gradient of the value function  $V_i(x_i, x_{i+1})$  is the vector  $(-p_i, p_{i+1})$ . Therefore the computation of the Hamiltonian flow  $(x_{i+1}, p_{i+1}) = \exp_{\vec{h}}(t_{i+1}, t_i, (x_i, p_i))$  provides the optimal cost transition  $V_i(x_i, x_{i+1})$  as well as its associated gradient. An extended database containing this additional information on the gradient could then be created without additional cost and used for instance to train Physics-Informed Neural Networks (PINNs).

**Acknowledgements.** 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, *Dynamic Programming*, Dover Publications, 1957.
- [4] 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.
- [5] 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.
- [6] O. COTS, R. DUTTO, S. LAPORTE, AND S. JAN, *A bilevel optimal control method and application to the hybrid electric vehicle*, 2023.
- [7] ———, *Geometric preconditioner for indirect shooting and application to hybrid vehicle*, 2024.
- [8] 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.
- [9] L. JOHANNESSON, M. ASBOGARD, AND B. EGARDT, *Assessing the potential of predictive control for hybrid vehicle powertrains using stochastic dynamic programming*, in *Proceedings. 2005 IEEE Intelligent Transportation Systems*, 2005, pp. 366–371.
- [10] 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.
- [11] L. S. PONTRYAGIN, V. G. BOLTYANSKII, R. V. GAMKRELIDZE, E. F. MISHECHENKO, *The Mathematical Theory of Optimal Processes*, New York, (1962).

- [12] A. A. MALIKOPOULOS, *Supervisory Power Management Control Algorithms for Hybrid Electric Vehicles: A Survey*, IEEE Trans. Intell. Transp. Syst., 15 (2014), pp. 1869–1885.
- [13] G. ROUSSEAU, *Véhicule hybride et commande optimale*, PhD thesis, ENMP, 2008.
- [14] R. S. SUTTON AND A. G. BARTO, *Reinforcement Learning: An Introduction*, The MIT Press, 2018.
- [15] 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.
- [16] 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.