

A bi-level energy management strategy for HEVs under probabilistic traffic conditions

Arthur Le Rhun, Frédéric Bonnans, Giovanni de Nunzio, Thomas Leroy, Pierre Martinon

▶ **To cite this version:**

Arthur Le Rhun, Frédéric Bonnans, Giovanni de Nunzio, Thomas Leroy, Pierre Martinon. A bi-level energy management strategy for HEVs under probabilistic traffic conditions. 2020. hal-02278359v2

HAL Id: hal-02278359

https://inria.hal.science/hal-02278359v2

Preprint submitted on 10 Sep 2020

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

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

A bi-level energy management strategy for HEVs under probabilistic traffic conditions

Arthur Le Rhun, Frédéric Bonnans, Giovanni De Nunzio, Thomas Leroy, Pierre Martinon

***Abstract*—This work proposes a new approach for the optimal energy management of a hybrid electric vehicle taking into account traffic conditions. The method is based on a bi-level decomposition in order to make the implementation suitable for online use. The offline lower level computes cost maps thanks to a stochastic optimization that considers the influence of traffic, in terms of speed/acceleration probability distributions. At the online upper level, a deterministic optimization computes the ideal state of charge at the end of each road segment, using the computed cost maps. Since the high computational cost due to the uncertainty of traffic conditions has been managed at the lower level, the upper level is fast enough to be used online in the vehicle. Errors due to discretization and computation in the proposed algorithm have been studied. Finally, we present numerical simulations using actual traffic data, and compare the proposed bi-level method to a deterministic optimization with perfect information about traffic conditions, as well as to an established solution for the energy management of a hybrid electric vehicle. The solutions show a reasonable over-consumption compared with deterministic optimization, and manageable computational times for both the offline and online part.**

***Index Terms*—Hybrid electric vehicles, energy management, bi-level optimization, stochastic dynamic programming, traffic data clustering.**

I. Introduction

THE future of road transportation is bound to undergo major transformations in the coming years. While energy-related air pollution is considered today as one of the primary premature death causes [1], the global carbon dioxide (CO2) emissions are on a rising trend destined to grow well above the levels imposed by the international climate goals [2].

New EU fleet-wide CO2 emission targets are set for the years 2025 and 2030 for newly registered passenger cars. These targets are defined as a percentage reduction from the 2021 starting points and impose 15% reduction starting from 2025 and 37.5% reduction starting from 2030 [3].

In an effort to comply with these binding measures, automakers are planning to significantly reduce the sales of vehicles solely powered by internal combustion engines (ICEs), and promote vehicle electrification. Market share projections seem to agree on the fact that purely electric vehicles (EVs) will reach at least 8% of all vehicle sales by 2025, while hybrid electric vehicles (HEVs) will rise to 23% of market share [4], [5]. The larger market penetration of HEVs, combining conventional combustion engines and electric motors, is likely due to a more appealing trade-off between energy efficiency and driving range.

Increased efficiency of HEVs derives from the on-board energy management system (EMS) which optimizes at each time instant the power split ratio between the two propulsion systems. A detailed review of the existing power management control algorithms for HEVs is offered by [6]. However, most of the current EMS strategies are somewhat conservative and sub-optimal due to their lack of prediction capabilities of the actual driving conditions of the vehicle. In fact, driving behavior and traffic conditions have a major impact on the traction power demand and consequently on the EMS. Several recent studies attempt to precisely establish such a relationship between driving conditions and energy consumption for different types of vehicle powertrains [7], [8].

The advent of connectivity and the availability of large amounts of driving data is favoring the transition towards predictive EMS strategies, which can further improve energy efficiency of HEVs by more effectively taking into account road traffic externalities [9], [10]. Such predictive strategies need to have an estimate of the required power for traction along the vehicle trajectory, based on information about traffic conditions, road signalization and road grade. Data-based driving behavior and traffic models are typically based on historical information about traffic conditions on the different portions of a road network. Speed and acceleration probability distributions and their statistical properties are generally used to represent driving behavior [11], [12] and to establish speed predictors. Those predictors either combine deterministic and stochastic approaches [13], [14], or are fully based on stochastic processes such as Markov chains and artificial neural networks [15], [16], [17], or are determined through independent and identically distributed (i.i.d.) sequences [18]. These probability distributions are often obtained from standard driving cycles [19], [14] or real driving data [20], [16], [21], [18]. Their ability to reproduce real driving conditions strongly affects the performance of predictive EMS.

Since stochastic processes are considered as an effective way of predicting driving behavior, the research on EMS for HEVs has put much effort in designing predictive and stochastic optimization strategies. Such strategies can be essentially grouped into offline and online optimization methods. The offline methods such as the Stochastic Dynamic Programming (SDP) are mostly used as a benchmark for the online

A. Le Rhun, G. De Nunzio and T. Leroy are with IFP Energies nouvelles, 1 et 4 avenue de Bois-Préau, 92852 Rueil-Malmaison, France, [arthur.le-rhun, giovanni.de-nunzio, thomas.leroy]@ifpen.fr

F. Bonnans is with Inria Saclay and CMAP Ecole Polytechnique, route de Saclay, 91128 Palaiseau, France, frederic.bonnans@inria.fr

P. Martinon is with Inria Paris and Sorbonne-Université, CNRS, Université de Paris, Laboratoire Jacques-Louis Lions (LJLL), F-75005 Paris, France, pierre.martinon@inria.fr

methods, but seldom used in practical implementations due to the high computational cost [22]. The online methods, such as the stochastic Model Predictive Control (MPC), in turn, offer more practical computation times without significantly compromising accuracy and performance. Stochastic MPC methods offer a favorable framework for the online prediction of the driving behavior on a future receding horizon and are employed for the design of EMS for HEVs [23]. However, accuracy and optimality of the MPC methods strongly depend on the accuracy of the driving behavior model, namely the probability distribution generating the stochastic process, as well as the size of the prediction horizon, typically chosen in a trade-off between performance and computational burden.

In an effort to reduce the impact of the prediction and optimization horizon on the EMS performance, bi-level optimization strategies seem promising in improving performance from a global perspective thanks to their hierarchical structure [24]. The intuition behind this type of optimization strategy for the EMS of HEVs is that in the system there are slowly changing variables, such as the battery state of charge (SoC), and rapidly changing variables, such as torque and regime. Adaptive hierarchical schemes [25], [26] and hierarchical MPC [27], [28] have been applied to the EMS problem for HEVs and proved compatible with real-time implementation. However, especially hierarchical MPC still solves an online optimization problem thus requiring modeling simplifications and limited prediction horizon. In this work, the proposed EMS strategy aims to combine the advantages of the infinite horizon optimization of the SDP and the real-time capability of the MPC, as follows. The statistical traffic model based on joint speed-acceleration probability distributions and proposed in [18] is used for stochastic speed prediction. A SDP is solved offline on each segment of a road network for different traffic conditions and different SoC variation constraints. This allows us to obtain a collection of pre-calculated optimal solutions (i.e. optimal torque split ratio) satisfying several combinations of problem constraints for all the segments of a road network, without compromising neither on the precision of the traffic model nor on the optimization horizon. Then, the output of this offline stage of the approach is used by the online EMS strategy which is designed as a bi-level optimization. The higher level is entitled to compute the optimal battery SoC profile for the desired trip in accordance with the constraints (e.g. desired battery depletion at the end of the trip). This supervisory SoC trajectory planning providing the intermediate SoC levels along the trip is solved by means of a fast Deterministic Dynamic Programming (DDP). The optimal SoC targets are then used by the lower level unit which does not need to solve any online optimization, and recovers the optimal torque split from the pre-calculated optimal solutions. The SoC constraint is thus verified according to the traffic conditions pertaining to each road segment along the trip. In the online phase of the proposed approach the desired trip is provided by the user and the actual speed is provided by a GPS device. Such information is used to determine the sequence of road segments composing the trip and the actual traffic conditions on each road segment. Finally, the optimal solution provided by the proposed stochastic energy management strategy is compared to the optimal solution, serving as a benchmark, calculated on a measured speed trajectory subject to actual traffic conditions.

The main contributions of the paper are twofold:

- The statistical traffic model introduced in [18] is here adapted to represent traffic conditions over a sequence of road segments. It is then used for stochastic driving behavior generation and in the EMS optimization.
- The energy management strategy is formulated as a bi-level hierarchical optimization problem. The novelty lies in the fact that the stochastic traffic behavior is entirely embedded in the lower level, which is optimized offline by means of stochastic dynamic programming. This leads to a very effective formulation for the upper level problem that can be solved online as a global optimization by means of a low-dimension deterministic dynamic programming.

The paper is organized as follows. Section II introduces the models for the hybrid vehicle, traffic conditions, and states the general problem of optimizing the fuel consumption. Section III presents an offline/online implementation of the EMS resulting from a bi-level decomposition of the optimization. Section IV discusses the numerical errors as well as the monotonicity of the value functions. Section V contains the validation of the proposed optimization strategy, the sensitivity analysis with respect to an incorrect traffic conditions identification, and comparisons with state-of-the-art approaches using real-world driving data.

II. Problem formulation

A. Hybrid vehicle model

Automobile companies have presented different architectures for hybrid electric vehicles. HEVs can be classified in four main types (Serial, Parallel, Combined, Complex) [29]. In the following we focus on HEVs with parallel design, where both the thermal engine and the electric motor can power the vehicle[1] (see Fig. 1). This type of design can use the engine to recharge the battery, at the cost of an increased fuel consumption.

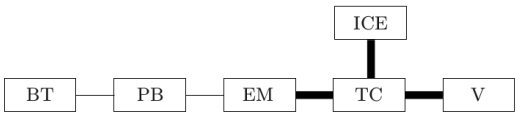


Fig. 1: Parallel hybrid configuration where ICE = internal combustion engine, BT = battery, PB = Power link, EM = electric motor, TC = torque coupler, V = vehicle. Bold lines indicate mechanical links, solid lines indicate electrical links.

Neglecting the slope effect and knowing the speed v and the acceleration a, the torque T_w and the rotation speed ω_w of

[1]Note that the proposed strategy could be applied to every HEV architecture.

the wheel at each time t can be computed using the following formula (1):

$$T_w(v(t), a(t)) = (ma(t) + \alpha_2 v^2(t) + \alpha_1 v(t) + \alpha_0) r_w \quad (1a)$$

$$\omega_w(v(t), a(t)) = \frac{60 v(t)}{2\pi r_w} \quad (1b)$$

with m the vehicle mass, r_w the wheel radius, and $\alpha_0, \alpha_1, \alpha_2$ coefficients of a quadratic approximation of the road-load force.

Then the torque required at the primary shaft T_{prim} and the rotation speeds of the engine ω_e and motor ω_m follow (2):

$$T_{prim}(v(t), a(t)) = \max\left(\frac{T_w(v(t), a(t))}{Pa_{ratio} Pa_{eff} G_R^i G_{Eff}^i}, T_{min} \right) \quad (2a)$$

$$\omega_e(v(t), a(t)) = \omega_w(v(t), a(t)) Pa_{ratio} G_R^i G_{Eff}^i \quad (2b)$$

$$\omega_m(v(t), a(t)) = \omega_w(v(t), a(t)) Pa_{ratio} G_R^i G_{Eff}^i R \quad (2c)$$

with G_R^i, G_{Eff}^i the gear ratio and efficiency, and Pa_{ratio}, Pa_{eff} the characteristics of the powertrain. Finally, R is the reduction ratio between the electric motor and the engine.

Neglecting losses due to the mechanical links, the torque of the engine and motor are linked through equation (3):

$$T_{prim}(v(t), a(t)) = T_e(t) + T_m(t) R \quad (3)$$

The consumption of the engine is modeled by a map $\hat{C}(\omega_e, T_e)$ obtained through experimental characterization, see Fig. 2. This map uses as inputs the torque request T_e and the rotation speed ω_e for the engine. Thanks to (1,2,3), we can express the consumption as a function of the electric motor torque T_m rather than T_e:

$$C(v(t), a(t), T_m(t)) = \hat{C}(\omega_e(v(t), a(t)), T_{prim}(v(t), a(t)) - T_m(t) R). \quad (4)$$

This reformulation allows us to consider the motor torque T_m as the control of our system. Similarly to the engine map, we also have a motor map that gives the electrical power $\hat{P}_m$ required by the motor:

$$P_m(v(t), a(t), T_m(t)) := \hat{P}_m(\omega_m(v(t), a(t)), T_m(t)) \quad (5)$$

This power can be positive or negative, corresponding to a discharge (resp. charge) of the battery. We note C_{max} the maximum capacity of the battery and $SoC(t) \in [0, 1]$ its state of charge at time t. The dynamics of the state of charge can be written as follows:

$$\dot{SoC}(t) = \frac{1}{C_{max}} P_m(v(t), a(t), T_m(t)) \quad (6)$$

B. Probabilistic traffic model

We consider a subdivision of the roads into small *segments*, typically delimited by topological characteristics. Let S denote the numbers of segments and let $h_0 > 0$ denote the time step. For each segment s and a given vehicle we have an entry time t_s and a time grid $t_{s,k} := t_s + k h_0$, for $k \in \mathbb{N}$. We assume that the traffic variables (speed and acceleration) are random variables $(\mathbf{V}(t), \mathbf{A}(t))$, constant over each *time-frame* $(t_{s,k}, t_{s,k+1})$, $k \in \mathbb{N}$, with discrete i.i.d. distributions μ^s, called the *traffic distribution*, whose (finite) support $\mathrm{supp}(\mu^s)$ is included in $\mathbb{R}_+ \times \mathbb{R}$. Fig. 3 shows an example of such a distribution.

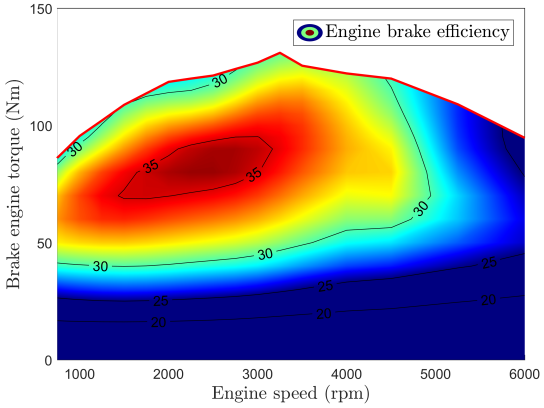


Fig. 2: Efficiency map of the engine

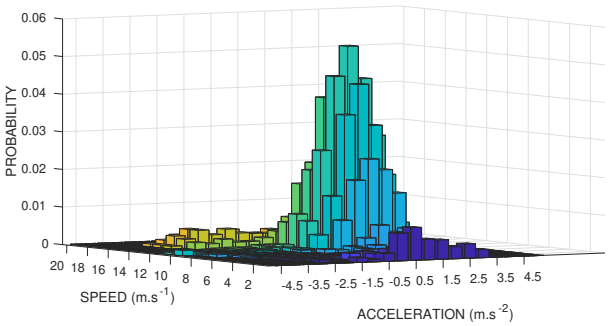


Fig. 3: Illustrative example of (speed,acceleration) distribution obtained from the traffic simulator SUMO. Source: `https://sumo.dlr.de`

The assumption that traffic distributions do not depend on time is an important simplification, reducing considerably the burden of computation. In our study, we consider travels of limited length, which justifies this hypothesis of stationary traffic distributions, and we choose the time step $h_0 = 1s$, which is consistent with the typical drivers reaction time.

We make the central assumption that drivers 'follow the traffic', meaning that their speed and acceleration coincide with the random variables $(\mathbf{V}(t), \mathbf{A}(t))$. Such a statistical traffic model that performs a clustering of the traffic distributions was presented in [18]. A derived consumption estimate is also provided, based on a memoryless sampling method for the (speed,acceleration) of the vehicle following the traffic conditions. Numerical simulations using actual traffic data (cf V-A) indicate that a small number of clusters (i.e. 4) can be sufficient to obtain a rather accurate prediction of the energy consumption, as shown in Fig. 4. As for data occupancy, using 4 clusters leads to a 99% reduction in size compared to the raw non-clustered (v,a) distributions.

Furthermore, clusters obtained from the (speed,acceleration) distributions can be interpreted in terms of traffic conditions.

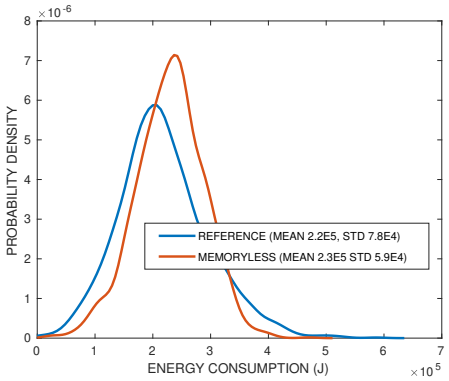


Fig. 4: Probability distributions of energy consumption. Reference (blue) vs Memoryless sampling method (red). The mean and std are indicated for the 2 discrete distributions.

Fig. 5 shows the clusters associated with each time-frame. Cluster 3 (dense traffic) appears in the morning between 7am and 9:30am, and cluster 4 (traffic jam) is present essentially between 4:30pm and 8pm. Cluster 1 (normal traffic) appears during the day and cluster 2 (fluid conditions) during the night. Overall, this traffic pattern is consistent with a typical commuting behavior, exhibiting two peak hours in the morning and evening.

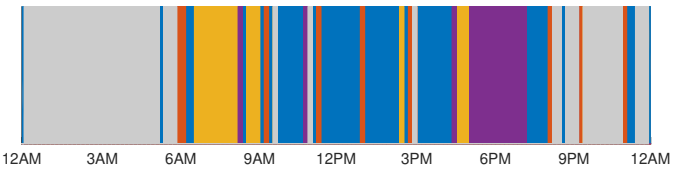


Fig. 5: Traffic clusters (Cluster 1: blue, Cluster 2: red, Cluster 3: yellow, Cluster 4: purple) according to the time of day (grey for time-frames without sufficient data for statistical analysis).

C. Global minimal expected consumption problem

Consider the problem of minimizing the total fuel consumption over a given route, defined by the sequence of segments 1 to S, in this order. Denote by L_F the total trip length, and by t_F the corresponding final time, which is a random time in view of the previous model of the traffic on each segment. The consumption is a random variable as well. So, the problem of *global minimum expected consumption* reads as $(\mathcal{GP})$:

$$\min_{T_m} \quad \mathbb{E}\left[\int_{t=0}^{t_F} C(\mathbf{V}(t),\mathbf{A}(t),T_m(t))\mathrm{d}t + P_F(SoC(t_F))\right] \tag{7}$$

$$s.t\ \forall t, \quad \dot{SoC}(t) = \frac{1}{C_{max}} P_m(\mathbf{V}(t),\mathbf{A}(t),T_m(t)) \tag{8}$$

$$\dot{D}(t) = \mathbf{V}(t) \tag{9}$$

$$t_F = \min\{t, D(t) > L_F\} \tag{10}$$

$$SoC(0) = SoC_0,\ D(0) = 0 \tag{11}$$

$$s.t\ \forall t, \quad T_m(t) \in [T_{min}, T_{max}] \tag{12}$$

$$SoC(t) \in [0,1] \tag{13}$$

The state variables are the state of charge SoC and distance D, and the control is the motor torque T_m, assumed constant over each time-frame (t_k, t_{k+1}). In this formulation the random entry times t_s over each segment s are implicitly defined. Assuming $t_1 = 0$, t_{s+1} is the minimum time greater than t_s such that $D(t_{s+1}) = D(t_s) + L_s$, where L_s is the length of segment s. In this Hazard-Decision framework, the control variable depends on the present value of the random variables for speed and acceleration $(\mathbf{V}(t), \mathbf{A}(t))$. Finally, the final cost $P_F : \mathbb{R} \to \mathbb{R}$ expresses the preference for given final SoC values, typically in order to avoid solutions that would systematically discharge the battery. P_F should be non increasing and we assume it to be continuous.

In the sequel we assume that no matter what the initial state of charge $SoC_0 \in [0,1]$ is, there exists a sequence of control variables such that the state of charge remains in $[0,1]$. In particular the set below of 'one step' feasible controls

$$\mathcal{T}(SoC, D, \mathbf{V}, \mathbf{A}) := \{T_m \in [T_{min}, T_{max}]; \quad SoC + \frac{h(D,\mathbf{V})}{C_{max}} P_m(\mathbf{V},\mathbf{A},T_m) \in [0,1]\} \tag{14}$$

is nonempty, for all $SoC \in [0,1]$ and $(\mathbf{V}, \mathbf{A})$ in the support of the traffic distribution μ^s.

III. Bi-level decomposition ('micro'/'macro')

The above global problem $(\mathcal{GP})$ can be solved by dynamic programming techniques. However, this is not adapted to a real-time setting where the traffic could be perturbed on a particular segment by, say, an accident. Therefore we propose a bi-level decomposition of the problem that can be applied in real time. The core idea is to consider at the upper level the trip as a sequence of road segments. This introduces a dynamic structure that allows us to compute the value function by backward induction (dynamic programming).

At this so-called 'macro' level, the upper-level decision considered is a deterministic approximation of the state of charge of the battery at the exit point of segment s, called the *reference state of charge* and denoted by SoC_s^r. The macro problem consists in solving a dynamic programming problem where each decision consists in choosing the best reference value, and where each step $s \in \{1, \dots, S\}$ corresponds to a segment.

The 'micro' problem minimizes the expected consumption, for a given segment s and a given SoC variation.

Energy management architecture Following this mathematical model, we design an energy management system that can operate on-board of a hybrid electrical vehicle, see Fig. 6. The micro level is offline and computes value functions and associated cost maps for all road segments, traffic conditions and SoC conditions, using stochastic optimization, see III-A. The macro level is online and performs a deterministic optimization for the reference state of charge SoC^r at the end of each road segment, see III-B A third part, also online, is the recovery of the optimal torque split, whose inputs are the current SoC, speed and acceleration, and the value functions from the micro level, see III-C.

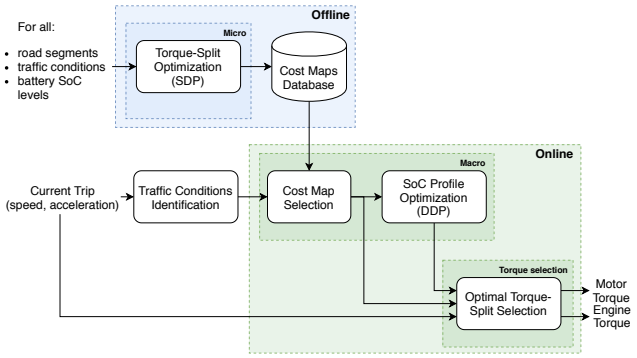


Fig. 6: Overview of the bi-level optimization framework

A. Micro problem

The framework is similar to the one of the global consumption problem, but restricted to a single segment s. By a translation argument we can assume that the entry time on segment s is 0. We denote by $SoC_s(t)$ and $D_s(t)$ the state of charge and distance, which will be the state variables, and by $T_m(t)$ the motor torque, which will be the control variable, constant over each time step. A Hazard-Decision framework is used: the control variable depends on the present value of the random variables $(\mathbf{V}(t), \mathbf{A}(t))$ (speed and acceleration). The final cost penalizes any final SoC different from the reference value. The resulting problem $(\mathcal{P}_{micro})$ reads as

$$\min_{T_m} \ \mathbb{E}\left[\int_{t=0}^{t_f} C(\mathbf{V}(t), \mathbf{A}(t), T_m(t))\mathrm{d}t + P_s(SoC_s(t_f), SoC_s^r)\right] \tag{15}$$

$$s.c\ \forall t,\ \dot{SoC}_s(t) = \frac{1}{C_{max}} P_m(\mathbf{V}(t), \mathbf{A}(t), T_m(t)) \tag{16}$$

$$\dot{D}_s(t) = \mathbf{V}(t) \tag{17}$$

$$T_m(t) \in [T_{min}, T_{max}] \tag{18}$$

$$SoC_s(t) \in [0,1] \tag{19}$$

$$SoC_s(0) = SoC_0,\ D_s(0) = 0 \tag{20}$$

$$t_f = \min\{t, D_s(t) > L_s\} \tag{21}$$

Alternatively we can reformulate the problem in a discrete time setting, dropping indexes s for the state variables:

$$\min_{T_m} \quad \mathbb{E}\left[\sum_{k=0}^{k_f} h^k C(\mathbf{V}^k, \mathbf{A}^k, T_m^k,) + P_s(SoC^{k_f}, SoC^r)\right] \tag{22}$$

$$s.c\ \forall k,\quad SoC^{k+1} = SoC^k + \frac{h^k}{C_{max}} P_m(\mathbf{V}^k, \mathbf{A}^k, T_m^k) \tag{23}$$

$$D^{k+1} = D^k + h^k \mathbf{V}^k \tag{24}$$

$$T_m^k \in [T_{min}, T_{max}] \tag{25}$$

$$SoC^k \in [0,1] \tag{26}$$

$$h^k := \min(h_0, (L_s - D^k)/\mathbf{V}^k) \tag{27}$$

$$k_f := 1 + \max\{k, D^k + h_0 \mathbf{V}^k < L_s\} \tag{28}$$

$$SoC^0 = SoC_0;\ D^0 = 0. \tag{29}$$

For the last segment we have a similar formulation, with final cost $P_F(SoC(t_f))$ introduced in section II-C. The dynamic programming principle used to solve these micro problems is detailed in section IV-A.

B. Macro problem

We denote by $\nu_s(SoC_0, SoC^r)$ the value of the micro problem with initial state of charge SoC_0 and reference state of charge SoC^r, and by $\nu_F(SoC_0)$ the value for the micro problem associated with the last segment. The (deterministic) macro problem is, when starting from segment $s_0 \in \{1, \dots, S-1\}$:

$$\min_{Soc^r} \quad \sum_{s=s_0}^{S-1} \nu_s(SoC_s^r, SoC_{s+1}^r) + \nu_F(SoC_S^r) \tag{30}$$

$$SoC^r \in [0,1]^S \tag{31}$$

$$SoC_{s_0}^r = SoC_0 \tag{32}$$

Denote by $\mathcal{V}_{s_0}$ the value function of the above problem. The corresponding dynamic programming principle is, for $1 \leq s \leq S-1$:

$$\mathcal{V}_s(SoC_s^r) = \min_{SoC_{s+1}^r} \nu_s(SoC_s^r, SoC_{s+1}^r) + \mathcal{V}_{s+1}(SoC_{s+1}^r), \tag{33}$$

and final condition

$$\mathcal{V}_S(SoC_S^r) = \nu_F(SoC_S^r). \tag{34}$$

This is a standard discrete time setting. If the functions involved in the model are Lipschitz, so is the value function $\mathcal{V}_s$. It is then possible to give estimates of the discretization error, following the analysis in [30]. We do not detail this part but rather put emphasis on the more complex case of the micro problem.

C. Torque selection

Once the values of SoC^r have been computed by solving the macro optimization, the travel starts and the vehicle follows the traffic. The optimal torque can then be determined, using the approximation $\tilde{U^c}$ of the value functions of the micro problem (we detail in IV-A the computation of $\tilde{U^c}$). Indeed, knowing the actual speed v and reconstructed acceleration a (typically thanks to the vehicle on-board sensors), the optimal torque T_m^* is chosen such that

$$T_m^* \in \underset{T_m \in \mathcal{T}(SoC, D, v, a)}{\operatorname{argmin}} \ h(D, v) C(v, a, T_m) + \tilde{U^c}(x_+) \tag{35}$$

This minimization, performed over a moderate-size discrete set, is fast enough for online use.

IV. Mathematical and numerical analysis

Here, we present the dynamic programming principle for the micro problems. Then, we give an analysis of the errors between the original problem and the problem actually solved, due to discretization and numerical methods. We also establish some monotonicity properties of the value functions.

A. Dynamic programming for 'micro' problem

A dynamic programming principle in discrete time is as follows. Clearly the value of the problem does not depend on time, so we look for $U(D, SoC) : [0, L_s] \times [0, 1] \to \mathbb{R}$, value of same problem as the 'micro' one but starting with state (D, SoC). We have an exit time problem since the process stops when $D = L_s$. For lighter notations, write $x := (D, SoC)$, and

$$x_+ := (D+h(D,\mathbf{V})\mathbf{V}, SoC+\frac{h(D,\mathbf{V})}{C_{max}}P_m(\mathbf{V},\mathbf{A},T_m)), \quad (36)$$

where $h(D, v) := \min(h_0, (L_s - D)/v)s$. Then

$$U(x) = \mathbb{E}\min_{T_m\in\mathcal{T}(SoC,D,\mathbf{V},\mathbf{A})}[h(D,\mathbf{V})C(\mathbf{V},\mathbf{A},T_m) + U(x_+)] \quad (37)$$

The expectation is over $(\mathbf{V}, \mathbf{A})$ with law ρ_s, and exit cost

$$U(L_s, SoC) = P_s(SoC, SoC^r). \quad (38)$$

Practical computations need to discretize the state space. Then an SDP algorithm can compute an estimate of the value function over the whole state grid. This allows to compute an approximation of the cost map $\nu_s(SoC_0, SoC^r)$ for given SoC_0 and SoC^r on a grid. The discretization analysis is given below.

B. State variables discretization error

We discretize the state variables $x := (D, SoC)$ with steps $h_D := L_s/N_D$ and $h_{SoC} := 1/N_{SoC}$, where N_D and N_{SoC} are positive integers. The value function $U^h = U^h(x)$ is defined at gridpoints $x = (k_D h_D, k_{SoC} h_{SoC})$ with $0 \le k_D \le N_D$, $0 \le k_{SoC} \le N_{SoC}$ and extended to $[0, L_s] \times [0, 1]$ by the usual multidimensional linear interpolation formula, see e.g. [30, Ch.3], denoted by $\tilde{U}^h$. By $\mathcal{G}$ (resp. $\mathcal{G}_-$) we denote the set of gridpoints (resp. of gridpoints with $D < L_s$). We consider the 'approximate' dynamic programming principle

$$U^h(D, SoC) = \mathbb{E}\min_{T_m\in\mathcal{T}(SoC,D,\mathbf{V},\mathbf{A})} \quad (39)$$
$$\Big[h(D,\mathbf{V})C(\mathbf{V},\mathbf{A},T_m) + \tilde{U}^h(x_+)\Big]$$

for all $x = (D, SoC) \in \mathcal{G}_-$, with exit cost as in (38). Since the velocities are non negative and have a positive expectation, the above formula implicitly expresses $U^h(D, SoC)$ as function of $U^h(D', \cdot)$ for $D' \in [D + h_D, L_s]$. So, (39) can be solved by backward induction over distances. For $D_k := kh_D$ and (D_k, SoC) in $\mathcal{G}_-$, set

$$U_k(SoC) := U(D_k, SoC); \;\; U_k^h(SoC) := U^h(D_k, SoC); \quad (40)$$

The corresponding value error is $W_k(SoC) := U_k^h(SoC) - U_k(SoC)$. Set

$$\hat{U}_k(SoC,\mathbf{V},\mathbf{A},T_m) := \tilde{U}(D_k + h(D_k,\mathbf{V})\mathbf{V}, \quad (41)$$
$$SoC + \frac{h(D_k,\mathbf{V})}{C_{max}}P_m(\mathbf{V}(t),\mathbf{A}(t),T_m(t))$$

with a similar definition for $\hat{U}_k^h$, and set $\hat{W}_k := \hat{U}_k^h - \hat{U}_k$. From the above dynamic programming principles (37) and (39) we deduce that

$$\|\hat{W}_k\|_\infty \le \mathbb{E}\sup_{SoC,T_m}\{|\hat{W}_k(SoC,\mathbf{V},\mathbf{A},T_m)|\}. \quad (42)$$

In the above supremum we take as always the expectation over $(\mathbf{V}, \mathbf{A})$ in the support of ρ_s, and the supremum over those $SoC \in [0, 1]$ such that $(D_k, SoC) \in \mathcal{G}_-$, and over $T_m \in \mathcal{T}(SoC, D, \mathbf{V}, \mathbf{A})$.

Let the supremum be attained at $(\overline{SoC}, \bar{T}_m)$, which are functions of $(\mathbf{V}, \mathbf{A})$. Denote by $(\bar{\alpha}_i, \bar{x}_i)$ the coefficients and gridpoints of the corresponding linear interpolation (also function of $(\mathbf{V}, \mathbf{A})$ but we skip these arguments). Then

$$\|\hat{W}_k\|_\infty \le \mathbb{E}\left(|\Delta_1(\mathbf{V},\mathbf{A})| + |\Delta_2(\mathbf{V},\mathbf{A})|\right), \quad (43)$$

with

$$\Delta_1(\mathbf{V},\mathbf{A}) := \sum_i \bar{\alpha}_i(U^h(\bar{x}_i) - U(\bar{x}_i)); \quad (44)$$

$$\Delta_2(\mathbf{V},\mathbf{A}) := \sum_i \bar{\alpha}_i(U(\bar{x}_i) - U(\bar{x}_+)). \quad (45)$$

We can interpret Δ_1 as a combination of previous errors at grid points, and Δ_2 as an interpolation error for U at x_+. Let $e_I(h_D, h_{SoC})$ denote a majorant of the interpolation error $|\Delta_2|$, so that we have

$$|\Delta_2| \le e_I(h_D, h_{SoC}). \quad (46)$$

In the case when no interpolation in D is necessary, we denote the corresponding interpolation error by $e'_I(h_{SoC})$. We now estimate $|\Delta_1|$. Setting I (resp. J) for elements of gridpoints with distance index equal to (resp. greater than) k we get

$$\Delta_1 := \sum_{i\in I}\bar{\alpha}_i(U^h(x_i) - U(x_i)) + \sum_{i\in J}\bar{\alpha}_i(U^h(x_i) - U(x_i)) \quad (47)$$

so that setting $\beta := \mathbb{E}\sum_{i\in I}\bar{\alpha}_i$:

$$|\Delta_1| \le \beta\|W_k\|_\infty + (1-\beta)\max_{k'>k}\|W_{k'}\|_\infty. \quad (48)$$

Observe that β represents the probability of having zero speed, and therefore is a given constant in $[0, 1)$.

Theorem 1. *The following error estimate holds: for all $(D, SoC) \in \mathcal{G}$, we have that*

$$\|U_k^h - U_k\|_\infty \le \frac{L_s e_I(h_D, h_{SoC})}{(1-\beta)h_D}. \quad (49)$$

If in addition, whenever v belongs to the marginal (in speed) of ρ_s:

$$hv \text{ is a multiple of } h_D, \quad (50)$$

then

$$\|U_k^h - U_k\|_\infty \le \frac{L_s e'_I(h_{SoC})}{(1-\beta)h_D}. \quad (51)$$

Proof. It follows from the previous discussion that

$$\|W_k\|_\infty \le e_I(h_D, h_{SoC}) + \beta\|W_k\|_\infty + (1-\beta)\max_{k'>k}\|W_{k'}\|_\infty. \quad (52)$$

Equivalently

$$\|W_k\|_\infty \le \max_{k'>k}\|W_{k'}\|_\infty + \frac{e_I(h_D, h_{SoC})}{1-\beta}. \quad (53)$$

Since there are L_s/h_D steps, (49) follows. Finally, if (50) holds, we deduce from (49) that (51) holds. □

Remark 2. *If U is Lipschitz w.r.t. SoC with constant L_{SoC}, then $e'_I(h_{SoC}) \leq L_{SoC}h_{SoC}$. The resulting error estimate is then of order h_{SoC}/h_D, which is similar to the standard error estimates in the case of a fixed horizon (where the 'exit variable' is replaced by time), see e.g. the appendix by Falcone in [31].*

C. Additional error due to computation

Instead of the approximate dynamic programming (39), at each step of the backward induction over distance, what we actually solve approximately is the problem

$$U_k^t(SoC) = \mathbb{E} \min_{T_m \in \mathcal{T}(SoC,D,\mathbf{V},\mathbf{A})} \Big[h(D_k,\mathbf{V})C(\mathbf{V},\mathbf{A},T_m) + \tilde{U}^c(x_+)\Big] \tag{54}$$

for all $x = (D_k, SoC) \in \mathcal{G}_-$, with exit cost as in (38); here we have replaced the discrete value U^h in the l.h.s. by the 'target value' denoted by U^t, and the 'future values' $\tilde{U}^h(x_+)$ with the 'computed values' (at gridpoints) denoted by $\tilde{U}^c(x_+)$; but note that while the computed values are given (at grid points) for distances greater than $D_k = h_D k$, their values at distance D_k is U_k^t. For $k \in \{0, \dots, N_D\}$, denote the corresponding error term estimate by e_k, so that

$$e_k \geq \max_{k' \geq k} \|U_{k'}^c - U_{k'}\|_\infty. \tag{55}$$

We may assume e_k to be non-increasing, with zero value for $k = N_D$. By arguments similar to those of the previous section we obtain that for $(D_k, SoC) \in \mathcal{G}_-$:

$$\|U_k^t - U_k\|_\infty \leq e_I + e_{k+1}. \tag{56}$$

Therefore, by the triangle inequality

$$\|U_k^c - U_k\|_\infty \leq \|U_k^c - U_k^t\|_\infty + e_I + e_{k+1}. \tag{57}$$

Next, we choose to solve (54) by value iterations, i.e., as the limit of the sequence $U_k^i(SoC)$, for $i \in \mathbb{N}$, defined by

$$U_k^i(SoC) := \mathbb{E} \min_{T_m \in \mathcal{T}(SoC,D,\mathbf{V},\mathbf{A})} \Big[h(D_k,\mathbf{V})C(\mathbf{V},\mathbf{A},T_m) + \tilde{U}_k^i(x_+)\Big]. \tag{58}$$

The infimum is of course for each grid value of SoC, and again the tilde corresponds to the interpolation operator. The contraction factor of the corresponding fixed-point operator is easily seen to be at most β. We initialize U_k^0 with U_{k+1}^c. It follows that

$$\|U_k^i - U_k^t\|_\infty \leq \beta^i \|U_{k+1}^c - U_k^t\|_\infty. \tag{59}$$

So, if we perform i_k iterations at step k we get with (57) that

$$\|U_k^c - U_k\|_\infty \leq e_I + e_{k+1} + \beta^{i_k} \|U_{k+1}^c - U_k^t\|_\infty. \tag{60}$$

While β may be computed, U_k^t is unknown so that explicit estimates can be derived only in specific examples. Nevertheless the above inequality suggests that, in the absence of additional information, it may be wise to take i_k independent on k.

D. Monotonicity of the value functions

Intuitively, we expect that the value functions appearing in the micro problems (both continuous and discretized) satisfy the following 'monotonicity property': they are nonincreasing w.r.t. SoC_0 and nondecreasing w.r.t. SoC^r. We need the following hypotheses:

1) The consumption function C is a continuous and non-increasing function of the torque T_m,
2) The power function P_m is a continuous and nonincreasing function of the torque T_m,
3) The final cost P_s is nonincreasing w.r.t. SoC_0 and nondecreasing w.r.t. SoC^r, and the final cost P_F is nonincreasing.

Theorem 3. *The value functions of the continuous and discretized micro problems satisfy the above monotonicity property.*

Proof. It is enough to obtain the result for the micro problem with discretization of the distance. The monotonicity w.r.t. SoC^r is an obvious consequence of hypothesis 3). We next establish the monotonicity w.r.t. SoC_0. Consider a feasible policy T_m^k with associated state SoC^k (with k indicating the distance step). These are random variables, depending on the realization of speed and acceleration. Consider now the perturbed problem with perturbed initial state of charge $SoC'_0 = SoC_0 + \varepsilon$, with $\varepsilon \geq 0$ and $SoC'_0 \leq 1$. We consider a perturbed trajectory $(\hat{T}_m^k, \hat{SoC}^k)$ such that for all k, $\hat{SoC}^k$ is the smallest possible majorant of SoC^k. This obtained by forward induction: we choose the control T_m^{k-1} as the largest feasible one such that $\hat{SoC}^k \geq SoC^k$. By hypothesis 2) and (23), the variations of state of charge of the perturbed trajectory are not greater than the original ones. Therefore thanks to hypothesis 1), we have a non greater consumption at each step k. By hypothesis 3), the final cost for the perturbed trajectory is not greater as the original one. The conclusion follows, using similar arguments in the case of the final segment. □

Remark 4. *Possible expressions of functions P_s and P_F, that satisfy assumption 3), are*

$$P_s(SoC_{t_f}, SoC_s^r) = \lambda \max(SoC_s^r - SoC_{t_f}, 0) \tag{61}$$

$$P_F(SoC_{t_f}) = \lambda \max(SoC_f - SoC_{t_f}, 0) \tag{62}$$

for some parameters $\lambda > 0$ and $SoC_f \in [0,1]$.

Remark 5. *For the macro problem, from the previous assumptions it follows that the cost function $\nu_s(\cdot,\cdot)$ is nonincreasing w.r.t. its first variable and nondecreasing w.r.t. its second variable. Assuming ν_F to be nonincreasing, it follows easily that the value of the macro problem is itself nonincreasing w.r.t. SoC_0.*

V. Numerical simulations

Numerical simulations have been conducted on a passenger vehicle, for which the values of the parameters are presented in Table I [32]. For the functions P_s and P_F we take the expressions given in Remark 4, with $\lambda = 0.1$.

TABLE I: Parameters used in simulations

m	r_w	α_0	α_1	α_2
$1190kg$	0.31725m	113.5	0.774	0.4212

i	1	2	3	4	5
G_{R^i}	3.416	1.809	1.281	0.975	0.767
G^i_{Eff}	1	1	1	1	1

Pa_{ratio}	Pa_{eff}	R
59/13	0.95	3.3077

A. *Traffic data*

As already mentioned, the probabilistic traffic model is defined by (speed,acceleration) probability distributions, and the random variables $(\mathbf{V}(t), \mathbf{A}(t))$ define the vehicle driving behavior. In this work the probability distributions representing the traffic conditions were obtained from real driving data collected via the smartphone application "Geco air" [33]. This application has recorded so far more than 60 million kilometers of driving data in real world conditions, and the data are mostly concentrated in French metropolitan areas.

The simulations were conducted on a 4 km stretch of highway in the south of Lyon, France see Fig. 7, which gets heavily congested at peak hours due to a merge with the city beltway. A total of about 1000 trips were recorded over a period of about 16 months (from January 2017 to May 2018). The considered road portion was split into 27 smaller segments (whose length ranges from 50 meters to 450 meters) according to the procedure detailed in [18]. In each segment, the traffic conditions were processed into 4 clusters.

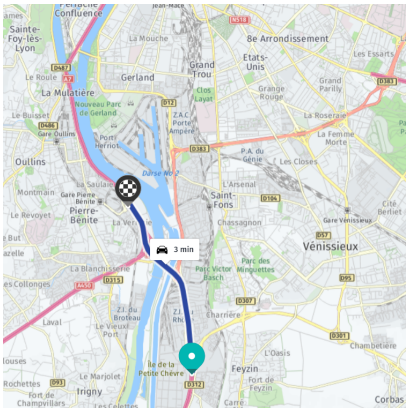


Fig. 7: Macro problem: simulated travel

B. *Micro problem*

We start the numerical simulations with the so-called micro problem defined in III-A. We recall that this problem consists in optimizing the consumption of the vehicle over a single road segment, for a prescribed initial and final SoC.

Traffic conditions and Value function In Fig. 8 we illustrate the micro problems on one segment with four clusters illustrating various traffic conditions. We consider here the iso-SoC problem with $SoC_0 = SoC_f = 0.25$. For each cluster, we plot the recorded speed profiles on the left-hand side graph. Middle graph shows the cluster barycenter and a scatter plot in the (speed,acceleration) plane. On the right-hand side graph we show the value function $\mathcal{V}$ on the (SoC,distance) state grid. Results can be interpreted in terms of traffic conditions:

- Cluster 1: fluid traffic with rather high and constant speed.
- Cluster 2: jammed traffic with much lower speed and frequent stops.
- Cluster 3/4: intermediate traffic.

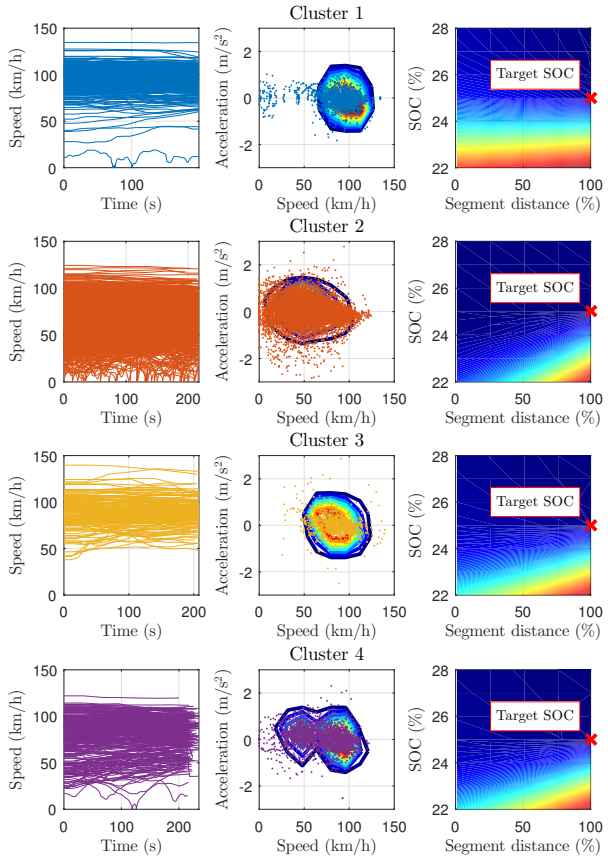


Fig. 8: Illustration of the micro problem on one segment (#9) with four traffic conditions clusters. On each graph, left to right: recorded speed profiles of the vehicles, (v,a) distribution and barycenter, value function on the (SoC,distance) state grid.

The value function is null if, when staying in fully electric mode, the inequality $SoC(t_f) \geq SoC_s^r$ holds almost surely. As expected from remark 5, for lower initial SoCs, the value function increases. As expected again, the value function is an increasing function of the distance. For cluster 1, the slope of the value function is smaller, which is due to the general high speed behavior: the required power cannot be provided only by the ICE, and the electric motor has to contribute to traction. Therefore recharging the battery is unlikely, and the penalization term for the final SoC constraint has a higher impact on the value function compared to the consumption from the ICE (increasing in distance).

Analysis of vehicle modes Let us interpret the information from the primary, motor and engine torques with respect to (v, a), in terms of modes of the hybrid vehicle:

- Regenerative breaking occurs for negative primary torques.
- Recharging is when the thermal engine is used to increase the state of charge on top of powering the vehicle.
- Pure electric is when the vehicle is powered by the electric motor only, with the thermal engine shut off.
- Hybrid mode uses both engine and motor together.

Fig. 9 shows the vehicle modes for the segment 9 (out of 27) and cluster 1, with an iso-SoC constraint, both at the beginning and end of the segment (90% distance covered). The main difference is the recharging mode, which occurs much more frequently at the beginning of the segment, and almost never at the end. This is consistent with the intuition that in a stochastic context, it is optimal to try to satisfy the final SoC constraint right from the beginning, rather than trying to reach it only near the end, which is likely to fail. Therefore the optimal torque selection policy is such that recharging is favored at the beginning of the segment, not at the end. Overall, the full electric mode appears to be predominantly used for null or low acceleration phases. Accelerating phases, on the other hand, typically involve the thermal engine.

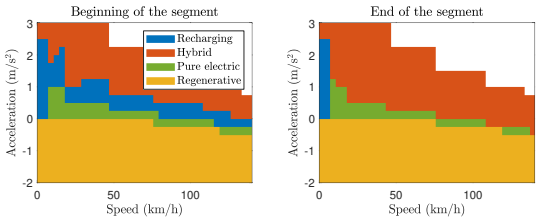


Fig. 9: Hybrid vehicle modes: optimal policy at the beginning and end of a segment.

C. Macro problem

We next present the numerical simulations with the proposed bi-level method. We recall that this problem consists in optimizing the fuel consumption of the vehicle over a whole trip, with a prescribed initial and final SoC.

Cost map from the micro problems As explained in III-B, the macro problem relies on a cost map v_s that gives an estimate of the consumption over one segment, with given traffic conditions, for fixed boundary conditions (SoC_0, SoC_f). In our case this map is built by solving the corresponding family of micro problems. With our choice of the functions P_s and P_F, the value of the micro problem is invariant w.r.t a translation on (SoC_0, SoC_f), whenever the bound constraints of the SoC are almost surely non active. We use this property and define the cost maps as 1D functions of $\Delta_{SoC} := SoC_f - SoC_0$

Fig. 10 shows the cost maps for the four traffic clusters of segment 9. The shape is similar for all clusters, with a cost increasing with the required Δ_{SoC}, as expected. The cost curves are divided into three parts:
- a first flat cost for low (negative) Δ_{SoC}, corresponding to travels with the thermal engine completely shut off. The fuel consumption is then zero and the small cost comes from the numerical diffusion of the penalty for the final SoC constraint.
- then the cost typically increases with Δ_{SoC}, since the thermal engine is used more in order to satisfy the SoC requirement.
- for larger Δ_{SoC} we observe a sharp increase in the cost function as the final SoC constraint becomes infeasible, resulting in a large penalization in the cost.

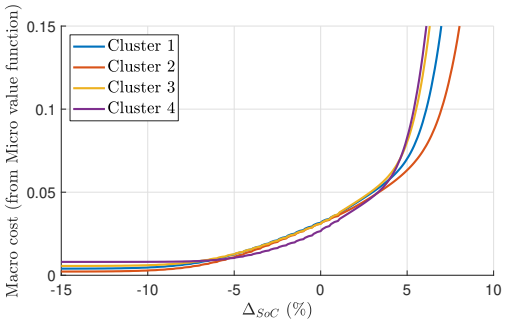


Fig. 10: Macro cost map v_s from the family of micro problems.

D. Validation of the bi-level method

An illustration of the bi-level approach on a travel using real traffic data is presented in V-E. Here we want to compare the solutions of the bi-level strategy with the global optima found by a deterministic dynamic programming (DDP) with full traffic knowledge. We generate a set of 1000 fictitious (v,a) profiles by i.i.d. sampling using the probabilities from the statistical traffic model. These profiles have an average duration of about 20 minutes, with a sampling frequency of 1Hz.

We show in Fig 11 the results of the bi-level and DDP method on this set of travels, for the $SoC_0 = SoC_f = 25$ case. We observe that the final SoC constraint is always respected by the DDP, as expected from a global method. The bi-level approach shows more variation in the final SoC, however the constraint is well satisfied overall, with an average value of 24.99. The consumption distributions have a similar (Gaussian) shape, with the bi-level method showing a slight over-consumption compared to the DDP.

Table II summarizes the comparison in terms of average final SoC $\bar{SoC_f}$ and consumption over the 1000 travels, for the three cases $SoC_f = 20, 25, 30$. We observe similar results for the three SoC_f, which indicates consistency of the bi-level approach. The average over-consumption w.r.t. to the DDP is about 4.5%, and the final SoC constraint is well respected.

(SoC_0, SoC_f)	(25,20)	(25,25)	(25,30)
$\bar{SoC_f}$	19.99	24.99	29.99
ΔC_{mM}^{ddp}	+4.40%	+4.45%	+4.56%

TABLE II: Bi-level vs DDP, on 1000 i.i.d travels

Numerical settings and CPU times We recall in Table III the state and control variables discretization for the DDP and bi-level methods. The CPU times on a standard laptop (i7-6700U @2.8GHz) are summarized in Table IV. The bulk of the computations is done offline for the micro value functions.

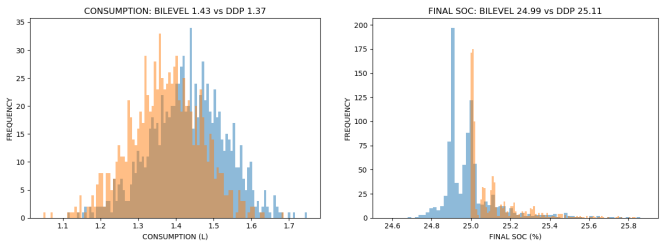


Fig. 11: Bi-level (blue) vs DDP (orange), on 1000 i.i.d generated trips with conditions $SoC_0 = SoC_f = 25$

Since they are independent for all (segment,cluster) pairs, they can be massively parallelized. As an indication, the value functions for our 27 segments, each with 4 clusters, take up to 1.7GB when stored as uncompressed text files.
Concerning the online part, the macro optimization at the start of the travel takes about 3.5s on average, while the torque selection performed at each time step of the micro resimulation takes a few milliseconds. Since the time step is set to 1s (1Hz frequency), the torque selection is clearly fast enough for real-time use.

	Discretization for state and control variables
DDP	$SoC \in [10, 40]$ (30 steps), $T_{mot} \in [-28, 28]$ (56)
micro	$SoC \in [10, 40]$ (300), $d \in [0, L](10)$, $T_{mot} \in [-28, 28]$ (56)
macro	$SoC \in [10, 40]$ (300), $\Delta_{SoC} \in [-5, 5]$ (100)

TABLE III: Discretization for DDP and bi-level methods. Time resolution is 1 second for DDP/micro. Space resolution is 1 segment for macro.

DDP	4.3s (average on 1000 travels)
micro value function	1 day (27 segments x 4 clusters)
macro optimization	3.2s (average on 1000 travels)
torque selection	3ms per decision (1Hz frequency)

TABLE IV: CPU times for DDP and bi-level approach.

Sensitivity to cluster information. We try now to check the sensitivity of the bi-level method with respect to the traffic clusters. Instead of using the correct cluster information for the road segments, we set a unique cluster index for the whole travel. Table V shows the results, and we see that the over-consumption vs DDP worsens by 25 to 40% when using the incorrect cluster information. The loss in terms of absolute consumption is rather small, however, which may indicate that there is some redundancy in the considered clusters. A possible way to verify this hypothesis would be to recompute the traffic model with fewer clusters and compare the results.

clusters	correct	always 1	always 2	always 3	always 4
ΔC_{mM}^{ddp}	+4.40%	+5.53%	+5.57%	+5.97%	+6.17%
rel. loss	0	25.7%	26.6%	35.7%	40.2%

TABLE V: Sensitivity to clusters - $SoC_0 = SoC_f = 25$

E. Comparison with ECMS on real speed profiles

The proposed strategy is now validated on real data instead of generated travels. In order to consider sufficiently long trips, the 1000 recorded trips coming from the 4-km stretch of highway are concatenated to form continuous speed profiles of at least 15 minutes. Following this methodology, 1000 speed profiles containing the 27 segments and 27x4 traffic clusters are generated. For all the trajectories, bot the initial SoC and the final SoC target are set to 25%.

Fig. 12 presents a result of the bi-level strategy on a particular speed profile, with both SoC trajectories corresponding to the macro optimization and the torque selection along the travel. The torque selection is computed in practice by resimulating the corresponding sequence of micro problems, using the stored value functions.

The top figure represents the vehicle speed profile associated with the powertrain states chosen by the energy management (torque selection). The bottom-left graph shows the target and the realized SoC trajectories. The bottom-right figure presents the location of the engine operating points on the efficiency map. We see that the macro stage specifies a target SoC trajectory satisfying the final condition $SoC_f = 25\%$. The circles represent the end of each road segment, showing the level of discretization of the macro dynamic programming. This SoC trajectory is used in the low-level controller to decide the energy split between the motor and the engine at each time step. We observe that the controller succeeds in following the SoC trajectory and satisfying the final constraint. A difference between target and realized SoC can be seen at the beginning of the trip. It can be explained by the sharp deceleration, implying a negative electric torque charging the battery. While satisfying the SoC constraint, the energy management optimizes the engine use as shown on the bottom-right figure: most of the engine operating points are located in the high efficiency area.

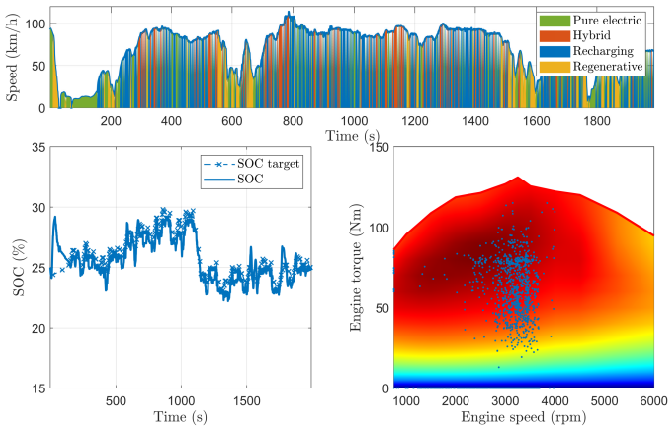


Fig. 12: Illustration of a bi-level solution, for a travel with conditions $SoC_0 = SoC_f = 0.25$.

We now compare the proposed bi-level method with both a deterministic global optimization and an established industrial method. First, a deterministic dynamic programming (DDP) strategy is computed as a reference, knowing the exact vehicle speed and acceleration profiles all along the trip. Then, an Energy Management System Strategy (ECMS) [6] is implemented for further comparison.

The three strategies are compared over the 1000 speed

profiles, with the results shown in Fig. 13 and Table VI. The two online strategies (bi-level and ECMS) show an increase of consumption of about 10% compared to the optimal consumption of the DDP, with distributions similar in shape (Gaussian-like). Moreover, the final SoC constraint is well satisfied on average by both strategies, with the bi-level method appearing to bracket the target SoC_f more accurately than the ECMS. The over-consumption of the bi-level method, although tolerable, is significantly larger than the one observed in section V-D for a set of i.i.d generated travels. This could be explained by the reliability of the stochastic traffic model, and improving the model could reduce the overall consumption.

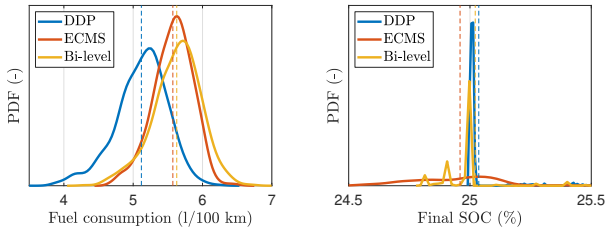


Fig. 13: Comparison of fuel consumption and final SoC on 1000 real speed profiles for DDP, ECMS and bi-level strategies.

	Average consumption	Average SoC_f
DDP	5.12 l/100 km	25.04%
ECMS	5.57 l/100 km (+9.1%)	24.96%
Bi-Level	5.63 l/100 km (+10.1%)	25.02%

TABLE VI: Comparison of average fuel consumption and final SoC on 1000 real speed profiles for DDP, ECMS and bi-level strategies.

It is worth noting that although presented quite differently, ECMS and bi-level strategies present interesting similarities. On the one hand, ECMS determines the optimal decision by minimizing a Hamiltonian function, which requires some information on the so-called adjoint state associated to the SoC. On the other hand, the torque selection part of the bi-level strategy also performs a minimization to find the optimal decision, using cost maps derived from the value functions of the micro stochastic problems. In this sense the bi-level may be seen as a new variant of the ECMS method, using global optimization at both the micro and macro levels. This is supported by well-known link in control theory between the adjoint state of the Pontryagin minimum principle and the gradient of the value function (see for instance [34]).

In the present simulations the traffic data were taken from a highway portion mostly, and turned out to be rather regular overall. This was confirmed by the almost constant adjoint state observed in the ECMS method, and may explain the relatively low sensitivity to the cluster information of the bi-level method (see section V-D).

It would be interesting to perform another comparison using traffic data with more variations, to see if the global aspect of the bi-level approach performs better in this case than a standard (local) ECMS method.

VI. Conclusion

We have presented a bi-level method for the energy management of a hybrid vehicle. More precisely, the aim is to minimize the fuel consumption of the thermal engine over a fixed travel, assuming that the vehicle follows the (stochastic) traffic conditions of the road. We consider a subdivision of the road network into small segments, for which typical traffic conditions are modeled as probability distributions in the (speed,acceleration) plane. At the upper 'macro' level, a fast, online optimization is performed to compute a deterministic approximation of the optimal SoC trajectory. This optimization relies on a cost map over all road segments, for all possible initial and final SoC conditions. The cost maps are computed at the lower 'micro' level from the value functions obtained by solving offline a family of stochastic optimization problems under traffic constraints. Additionally, these offline solutions are used to recover the optimal control (electric torque) during the online optimization. Numerical simulations carried out using actual traffic data from a highway portion near Lyon (France) indicate that the bi-level approach performs in a satisfying manner. A limited over-consumption is observed compared to a deterministic solution using fully known traffic information, and performance is comparable to well-established industrial solutions such as the ECMS. The key point to obtain a method fast enough for online use is that the stochastic traffic conditions are completely handled at the micro level, leaving only a simple deterministic optimization at the macro level. Future works may include a comparison with other bi-level approaches using e.g. MPC techniques, as well as a deeper investigation of the link between the presented bi-level method and the existing ECMS. On the traffic model aspect, there is still room for improvement, as shown by the gap between the tests with the i.i.d. generated travels and the real ones. For the clustering part in particular, isolating data for a specific vehicle/driver could allow for some meta-clustering in order to further reduce the data size and/or to increase the relevance of the probability distributions.

References

[1] "Ambient air pollution: a global assessment of exposure and burden of disease," World Health Organization, Tech. Rep., 2016.
[2] "World Energy Outlook 2018," International Energy Agency, Tech. Rep., 2018.
[3] "Post-2020 co2 emission performance standards for cars and vans," https://ec.europa.eu/clima/policies/transport/vehicles/regulation_en, [Online; accessed July 2019].
[4] T. Gnann, T. S. Stephens, Z. Lin, P. Plötz, C. Liu, and J. Brokate, "What drives the market for plug-in electric vehicles ? - A review of international PEV market diffusion models," *Renewable and Sustainable Energy Reviews*, vol. 93, pp. 158–164, 2018. [Online]. Available: https://doi.org/10.1016/j.rser.2018.03.055
[5] "Driving into 2025: The Future of Electric Vehicles," J.P. Morgan, Tech. Rep. [Online]. Available: https://www.jpmorgan.com/global/research/electric-vehicles
[6] A. A. Malikopoulos, "Supervisory Power Management Control Algorithms for Hybrid Electric Vehicles: A Survey," *IEEE Transactions on Intelligent Transportation Systems*, vol. 15, no. 5, pp. 1869–1885, 2014.
[7] I. De Vlieger, D. De Keukeleere, and J. G. Kretzschmar, "Environmental effects of driving behaviour and congestion related to passenger cars," *Atmospheric Environment*, vol. 34, pp. 4649–4655, 2000.
[8] C. Fiori, V. Arcidiacono, G. Fontaras, M. Makridis, K. Mattas, V. Marzano, C. Thiel, and B. Ciuffo, "The effect of electrified mobility on the relationship between traffic conditions and energy consumption," *Transportation Research Part D*, vol. 67, pp. 275–290, 2019.

[9] C. M. Martínez, X. Hu, D. Cao, E. Velenis, B. Gao, and M. Wellers, "Energy Management in Plug-in Hybrid Electric Vehicles: Recent Progress and a Connected Vehicles Perspective," *IEEE Transactions on Vehicular Technology*, vol. 66, no. 6, 2017.

[10] F. Zhang, X. Hu, R. Langari, and D. Cao, "Energy management strategies of connected HEVs and PHEVs : Recent progress and outlook," *Progress in Energy and Combustion Science*, vol. 73, pp. 235–256, 2019.

[11] P. Cao, T. Miwa, and T. Morikawa, "Use of Probe Vehicle Data to Determine Joint Probability Distributions of Vehicle Location and Speed on an Arterial Road," *Transportation Research Record*, vol. 2421, no. 1, 2014.

[12] R. Liu and X. Zhu, "Statistical Characteristics of Driver Accelerating Behavior and Its Probability Model," *arXiv:1907.01747*, 2019.

[13] K. R. Bouwman, T. H. Pham, S. Wilkins, and T. Hofman, "Predictive Energy Management Strategy Including Traffic Flow Data for Hybrid Electric Vehicles," *IFAC PapersOnLine*, vol. 50, no. 1, pp. 10 046–10 051, 2017. [Online]. Available: https://doi.org/10.1016/j.ifacol.2017.08.1775

[14] G. Wu, K. Boriboonsomsin, M. J. Barth, and S. Member, "Development and Evaluation of an Intelligent Energy-Management Strategy for Plug-in Hybrid Electric Vehicles," *IEEE Transactions on Intelligent Transportation Systems*, vol. 15, no. 3, pp. 1091–1100, 2014.

[15] C. Sun, X. Hu, S. J. Moura, and F. Sun, "Velocity Predictors for Predictive Energy Management in Hybrid Electric Vehicles," *IEEE Transactions on Control Systems Technology*, vol. 23, no. 3, pp. 1197–1204, 2015.

[16] D. Karbowski, V. Sokolov, and A. Rousseau, "Vehicle Energy Management Optimization through Digital Maps and Connectivity," in *ITS World Congress*, 2015.

[17] M. Montazeri-Gh and M. Mahmoodi-K, "Optimized predictive energy management of plug-in hybrid electric vehicle based on traffic condition," *Journal of Cleaner Production*, vol. 139, pp. 935–948, 2016. [Online]. Available: http://dx.doi.org/10.1016/j.jclepro.2016.07.203

[18] A. Le Rhun, F. Bonnans, G. De Nunzio, T. Leroy, and P. Martinon, "A stochastic data-based traffic model applied to vehicles energy consumption estimation," *IEEE Transactions on Intelligent Transportation Systems*, pp. 1–10, 2019.

[19] S. J. Moura, H. K. Fathy, D. S. Callaway, and J. L. Stein, "A Stochastic Optimal Control Approach for Power Management in Plug-In Hybrid Electric Vehicles," *IEEE Transactions on Control Systems Technology*, vol. 19, no. 3, pp. 545–555, 2011.

[20] X. Jiao and T. Shen, "SDP Policy Iteration-Based Energy Management Strategy Using Traffic Information for Commuter Hybrid Electric Vehicles," *Energies*, vol. 7, no. 7, pp. 4648–4675, 2014.

[21] X. Fuguo, J. Yuan, and J. Xiaohong, "A Modified Energy Management Strategy Based on SDP Policy Iteration for Commuter Hybrid Electric Vehicles," in *2016 35th Chinese Control Conference (CCC)*. TCCT, 2016, pp. 2537–2541.

[22] Chan-Chiao Lin, Huei Peng, and J. W. Grizzle, "A stochastic control strategy for hybrid electric vehicles," in *Proceedings of the 2004 American Control Conference*, vol. 5, June 2004, pp. 4710–4715 vol.5.

[23] Y. Huang, H. Wang, A. Khajepour, H. He, and J. Ji, "Model predictive control power management strategies for hevs: A review," *Journal of Power Sources*, vol. 341, pp. 91 – 106, 2017. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0378775316316731

[24] A. Sinha, P. Malo, and K. Deb, "A review on bilevel optimization: From classical to evolutionary approaches and applications," *IEEE Transactions on Evolutionary Computation*, vol. 22, no. 2, pp. 276–295, April 2018.

[25] H. Lim and W. Su, "Hierarchical Energy Management for Power-Split Plug-In HEVs Using Distance-Based Optimized Speed and SOC Profiles," *IEEE Transactions on Vehicular Technology*, vol. 67, no. 10, pp. 9312–9323, 2018.

[26] T. Liu, X. Tang, H. Wang, H. Yu, and X. Hu, "Adaptive Hierarchical Energy Management Design for a Plug-In Hybrid Electric Vehicle," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 12, pp. 11 513–11 522, 2019.

[27] C. Sun, S. J. Moura, X. Hu, J. K. Hedrick, and F. Sun, "Dynamic Traffic Feedback Data Enabled Energy Management in Plug-in Hybrid Electric Vehicles," *IEEE Transactions on Control Systems Technology*, vol. 23, no. 3, pp. 1075–1086, 2015.

[28] J. Fu, S. Song, Z. Fu, and J. Ma, "Hierarchical Model Predictive Control for Parallel Hybrid Electric Vehicles," *Asian Journal of Control*, vol. 20, no. 6, pp. 2331–2342, 2018.

[29] K. T. Chau and Y. S. Wong, "Overview of power management in hybrid electric vehicles," *Energy Conversion and Management*, vol. 43, no. 15, pp. 1953–1968, 2002. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0196890401001480

[30] M. Falcone and R. Ferretti, *Semi-Lagrangian Approximation Schemes for Linear and Hamilton–Jacobi Equations*. SIAM Publications, 2013.

[31] M. Bardi and I. Capuzzo-Dolcetta, *Optimal control and viscosity solutions of Hamilton-Jacobi-Bellman equations*. Birkhäuser, Boston, MA, 1997.

[32] G. De Nunzio and L. Thibault, "Energy-optimal driving range prediction for electric vehicles," in *Intelligent Vehicles Symposium (IV), 2017 IEEE*. IEEE, 2017, pp. 1608–1613.

[33] "Geco air," http://www.gecoair.fr/home_en/, [Online; accessed July 2019].

[34] V. R. Clarke, F., "The relationship between the maximum principle and dynamic programming." *SIAM J. Control Optim.*, no. 25, pp. 1291–1311, 1987.



Arthur Le Rhun received his M.Sc. degree in engineering from the ENSTA Paristech (France), and a M.Sc. degree in Optimization from Université Paris XI both in 2016. He is currently pursuing a Ph.D. in applied mathematics at the Ecole polytechnique and IFP Energies nouvelles. His research theme is dynamic optimization for stochastic systems, and applications to energy management of hybrid vehicles.



Frederic Bonnans received his engineering degree from the Ecole Centrale Paris, a PhD (docteur ingenieur) from the Université Technique de Compiègne, and the Habilitation from Université Paris IX Dauphine in 1979, 1982 and 1993 resp. He is currently a senior researcher at Inria Saclay and Ecole polytechnique. His research theme is dynamic optimization for deterministic and stochastic systems, and applications to energy management and biological models.



Giovanni De Nunzio received the B.Sc. and M.Sc. degrees in information and automation engineering from the University of L'Aquila, Italy, in 2007 and 2010. He received the PhD in automatic control from the Grenoble Institute of Technology, France, in 2015. He is currently a research engineer with IFP Energies nouvelles, France. His research activities focus on modeling, control, and simulation of traffic and transportation systems, vehicle powertrain modeling, advanced driver-assistance systems, energy consumption minimization, and graph theory.



Thomas Leroy graduated from ESSTIN and received the M.Sc. degree in control theory from the University Henri Poincaré in Nancy, France, in 2006. He received the doctorate degree in Control theory and Mathematics from the Ecole des Mines ParisTech, France, in 2010. He is currently a research engineer at IFP Energies nouvelles, Rueil-Malamaison, France. His research activities focus on modeling, control, and simulation of internal combustion engines and hybrid electric powertrains.



Pierre Martinon received his M.Sc. and Ph.D. degrees in applied mathematics from the ENSEEIHT (Toulouse, France) in 2001 and 2005. He is currently a researcher at Inria Paris and LJLL Sorbonne Universite. His research field is optimization for dynamical systems, more specifically the numerical aspects.