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

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

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 to optimize the consumption of a hybrid electric vehicle taking into account the 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. The solutions show a reasonable over-consumption compared with deterministic optimization, and manageable computational times for both the offline and online parts.**

***Index Terms*—Hybrid 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]. 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 [10], [11] and to establish speed predictors. Those predictors either combine deterministic and stochastic approaches [12], [13], or are fully based on stochastic processes such as Markov chains [14], [15], [16], or are determined through independent and identically distributed (i.i.d.) sequences [17]. These probability distributions are often obtained from standard driving cycles [18], [13] or real driving data [19], [15], [20], [17]. Their ability in reproducing 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 methods, but seldom used in practical implementations due to the high computational cost [21]. The online methods, such

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

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 [22]. 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 [23]. 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. Hierarchical MPC has been applied to the EMS problem for HEVs [24], [25] and proved compatible with real-time implementation. However, 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.

The contributions of this work can be summarized as follows. Firstly, the statistical traffic model based on joint speed-acceleration probability distributions and proposed in [17] is used here for the 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. Secondly, 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 executed at lower rate and yields 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 in each road segment.

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 discusses the numerical simulations performed using actual traffic 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) [26]. 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 in order to recharge the battery, at the cost of an increased fuel consumption. This option allows for optimization of the global consumption if some information is available on traffic conditions along the travel.

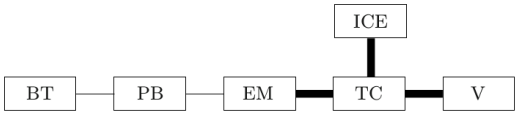

Fig. 1: Parallel hybrid configuration where ICE = internal combustion engine, BT = battery, PB = Power link, EM = electric motor, TC = torque coupler, V = vehicle. Bold lines indicate mechanical links, solid lines indicate electrical links.

Neglecting the slope effect and knowing the speed v and the acceleration a, the torque T_w and the rotation speed ω_w of the wheel at each time t can be computed using the following formula (1):

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

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

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):

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

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

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

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.

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

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

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

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:

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

This reformulation allows us to take 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:

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

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:

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

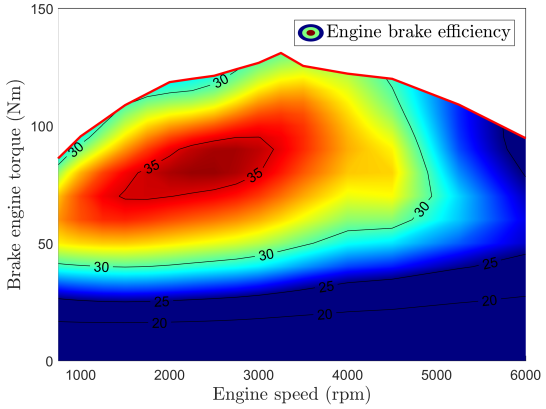

Fig. 2: Efficiency map of the engine

B. Probabilistic traffic model

We consider a subdivision of he roads into small *segments*, typically delimited by topological characteristiscs. 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 $t_{s,k} := t_s + kh_0$, for $k \in \mathbb{N}$. We assume that the traffoc variables (speed and acceleration) are random variables $(\mathbf{V}(t), \mathbf{A}(t))$, constant over each *grid intervals* $(t_{s,k}, t_{s,k+1})$, $k \in \mathbb{N}$, with discrete i.i.d. distributions μ^s, called the *traffic distribution*, whose (finite) support $\text{supp}(\mu^s)$ is included in $\mathbb{R}_+ \times \mathbb{R}$. Fig. 3 shows an example of such a distribution.

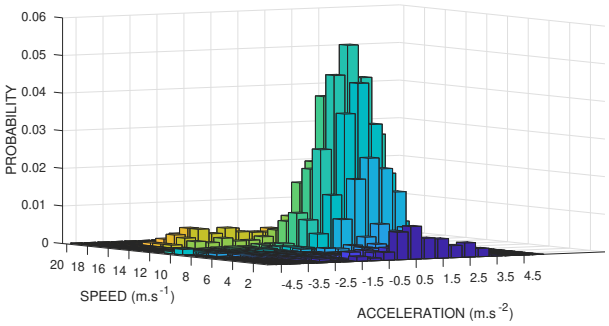

Fig. 3: Illustrative example of (speed,acceleration) distribution obtained from the traffic simulator SUMO[2].

The assumption that traffic distributions do not depend on time is an important simplification, allowing to reduce considerably the burden of computations. 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 reaction time of vehicles.

We make the central assumption that drivers 'follow the traffic', meaning that their speed and acceleration coincide with the random variables $(\mathbf{V}(t), \mathbf{A}(t))$. In [17] we showed how to construct this statistical traffic model and derived estimations for the energy consumption of a given car. In particular, we illustrate how these traffic distributions at different times of the day, for a given segment, can be clustered in order to reduce the size of information. It was observed that an accurate prediction of the energy consumption can be obtained with a small number of clusters.

C. Global minimal expected consumption problem

Consider the problem of minimizing the total engine consumption over a given route, defined by the successive traveling across segments 1 to S, in this order. Denote by L_F the total length of the travel, 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 minimal 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

[2]Source: https://sumo.dlr.de

over each grid interval (t_k, t_{k+1}). In this formulation are implicitly defined the random entry times t_s over each segment s, with $t_1 = 0$, t_{s+1} is the minimal 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 whatever is the initial state of charge $SoC_0 \in [0, 1]$, 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. 4. The micro level is offline and computes cost maps for all the 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.

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, in a Hazard-Decision framework as in the global problem: the control variable depends on the present value of the random variables $(\mathbf{V}(t), \mathbf{A}(t))$ (speed and acceleration). We add a final cost that penalizes the constraint to reach a final SoC at least equal to the reference value. The resulting problem $(\mathcal{P}_{micro})$ reads as

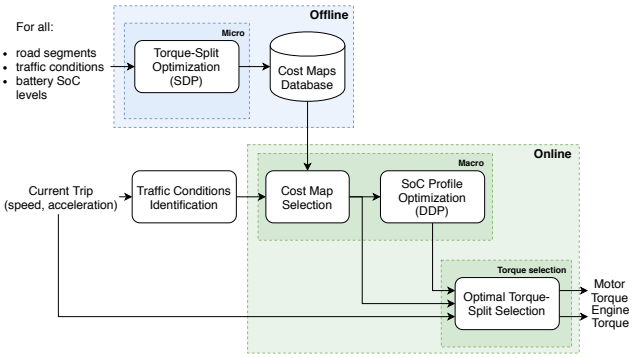

Fig. 4: Overview of the bi-level consumption optimization

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

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

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

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

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

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

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

Alternatively we can reformulate the above problem in a purely 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 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 [27]. 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

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

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

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

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

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

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

$$U(L_s, SoC) = P_s(SoC, SoC^r). \tag{38}$$

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 $0 \leq k_D \leq N_D$, $0 \leq k_{SoC} \leq N_{SoC}$ and extended to $[0, L_s] \times [0,1]$ by the usual multidimensional linear interpolation formula, see e.g. [27, 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

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

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

$$U_k(SoC) := U(D_k, SoC); \;\; U_k^h(SoC) := U^h(D_k, SoC); \tag{40}$$

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

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

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

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

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

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

with

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

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

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

$$|\Delta_2| \leq e_I(h_D, h_{SoC}). \tag{46}$$

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

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

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

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

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*

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

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

$$hv \text{ is a multiple of } h_D, \tag{50}$$

then

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

Proof. It follows from the previous discussion that

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

Equivalently

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

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 [28].*

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

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

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 $k \in \{0, \ldots, N_D\}$, 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

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

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 nonincreasing 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 $(\hat{T}_m^k, \hat{S}oC^k)$ such that for all k, $\hat{S}oC^k$ 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 $\hat{S}oC^k \geq SoC^k$. 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 simulation have been conducted on a passenger vehicle from [29], for which the value of the parameters are presented in Table I. 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_{Eff}^i	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" [30]. This application has recorded so far more than 40 million kilometers of driving data in real world conditions, and the data are mostly concentrated in the French metropolitan areas.

The simulations were conducted on a 4 km stretch of highway in the south of Lyon, France see Fig. 5, 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 [17]. In each segment, the traffic conditions were processed into 4 clusters.

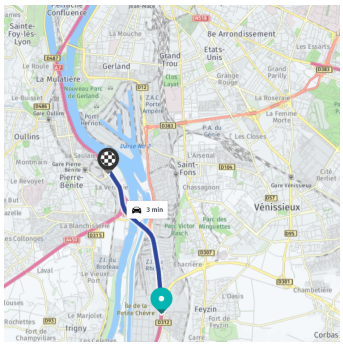

Fig. 5: Macro problem: simulated travel

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 energy consumption of the vehicle over a single road segment, for a prescribed initial and final SoC.

Traffic conditions and Value function On Fig. 6 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 speed profiles of all recorded vehicles on the left graph. Middle graph shows the cluster barycenter and scatter plot in the (speed,acceleration) plane. On the right graph we draw 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.

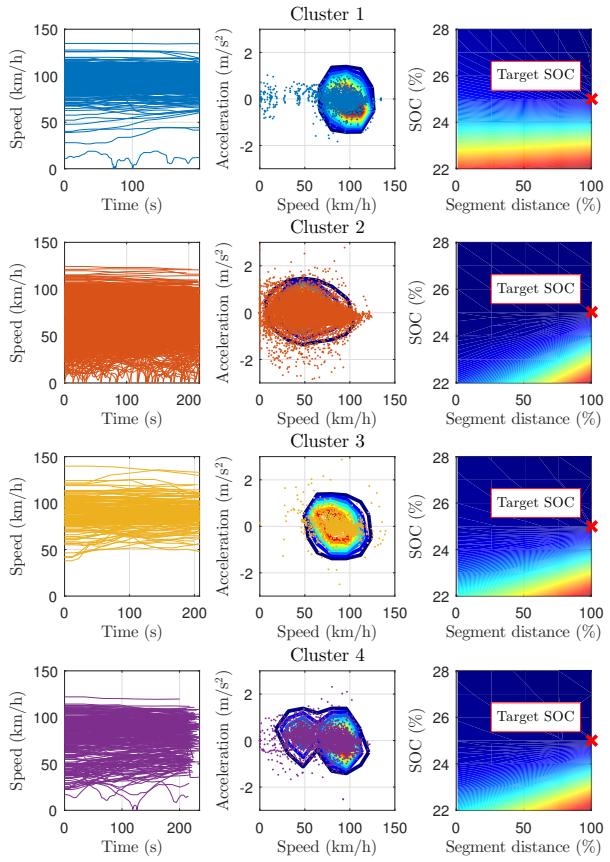

Fig. 6: Illustration of the micro problem on one segment (#9) with four traffic conditions clusters. On each graph, left to right: recorded speed profiles of the vehicles, (v,a) distribution and barycenter, value function on the (SoC,distance) state grid.

The value function is null if, when staying always in fully-electric mode, we have that 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 high speed: 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, thus 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 Fig. 7 interprets the information from the primary, motor and engine torques with respect to (v, a), in terms of modes of the hybrid vehicle:

- Regenerative breaking (light blue) occurs for negative primal torques.
- Recharging (dark blue) is when the thermal engine is used to increase the state of charge on top of powering the vehicle. This occurs only in cluster 1 (fluid traffic, high average speed) at low constant speeds.
- Pure electric (green) is when the vehicle is powered by the electric motor mostly (i.e. more than 90% of the required torque). We encounter this when acceleration is almost zero, except for cluster 1.
- Pure engine (red) is when the vehicle is powered by the thermal engine mostly (i.e. the motor provides less than 10% of the required torque). Such a situation arises mostly for cluster 1 with a high average speed.
- Hybrid (orange) corresponds to both engine and motor being used together. This is the main mode when the vehicle is accelerating ($a > 1\, m/s^2$).

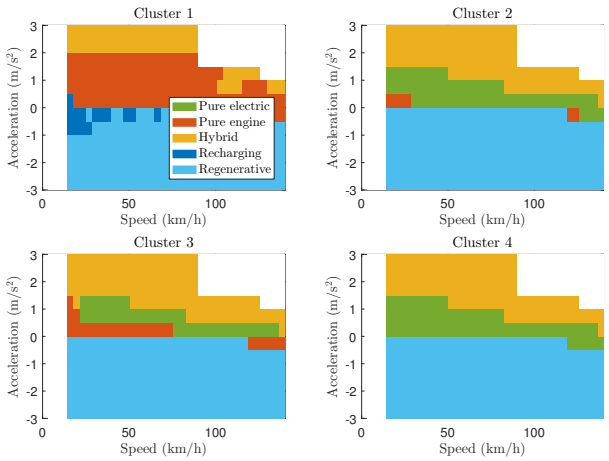

Fig. 7: Hybrid vehicle modes: different battery uses in the (speed,acceleration) plane. Graphs are for the 4 traffic conditions clusters and an iso-SoC constraint.

All in all, an interesting observation is that the thermal engine is predominantly used for i) fluid traffic conditions with a rather high average speed (cluster 1), where the engine often powers the vehicle alone and sometimes even recharges the battery; ii) accelerating phases, typically along with the electric motor.

Comparison with deterministic solution We now compare the stochastic method with a deterministic solution obtained by solving the micro problem when assuming a known (v, a) profile (DDP). The results are shown on Fig. 8-9 for two prescribed $\Delta_{SoC} = -0.04$ (4% discharge) and $\Delta_{SoC} = 0$ (iso-SoC). For the first 'easier' case (a negative Δ_{SoC} gives an easier final constraint), both methods find similar optimal trajectories. This means that the stochastic problem proposes the same strategy as the deterministic one in terms of torque.

The artificial speed profiles due to the i.i.d hypothesis do not influence the optimal strategy.

For the second 'harder' iso-SoC case, we observe an interesting discrepancy on the SoC trajectories:
i) the deterministic solution follows a rather conservative approach, maintaining a constant SoC all along in order to satisfy the final iso-SoC constraint.
ii) on the other hand, the stochastic solution starts with a more optimistic strategy, depleting the battery down to 49% after only a third of the segment. It then has to recharge the battery back to 50%, leading overall to a small over-consumption compared to the deterministic strategy. It is worth noting, though, that the gap in consumption remains quite small (below 5%). So the torque trajectories differ due to the excessive use of the motor at the beginning of the segment, with the engine torque being higher in the rest of the segment in order to recharge the battery.

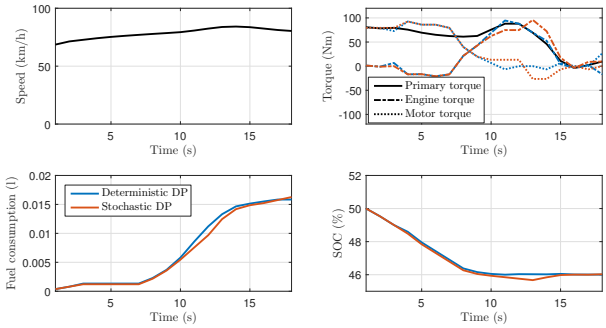

Fig. 8: Micro problem: SDP vs DDP. Sample optimal trajectories for cluster 2, discharge case (-4%).

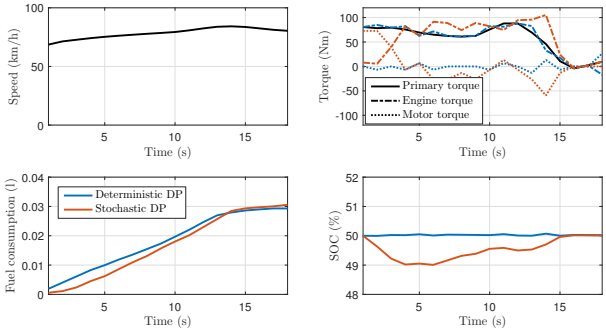

Fig. 9: Micro problem: SDP vs DDP. Sample optimal trajectories for cluster 2, iso-SoC case.

In order to obtain a more general view, we simulate a sample of 120 vehicles (about 10% of the total recorded vehicles) using both the stochastic and deterministic methods. All problems use the iso-SoC constraint $SoC_0 = SoC_f = 0.25$. We show on Fig. 10 the distribution of the consumption and actual $\Delta_{SoC} := SoC(t_f) - SoC(t_0)$. The stochastic method gives an average over-consumption of 5.54% (0.0150 vs 0.0142), which seems quite reasonable. We also observe that both methods manage to satisfy the iso-SoC constraint rather well, with the distributions clearly centered around zero, see also Table II. The stochastic method appears to have a very small bias towards positive Δ_{SoC}, which is likely due to the penalization only applying to the inequality constraint $SoC(t_f) \geq SoC_f$. The deterministic method uses the same penalization, but the knowledge of the (v, a) profile allows for a more precise targeting of the final SoC.

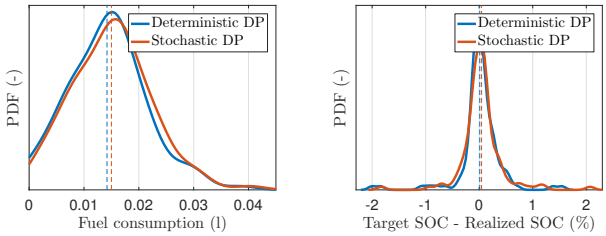

Fig. 10: Micro problem: SDP vs DDP. Distributions of the consumption (left) and Δ_{SoC} (right), for a sample of 120 simulated vehicles. The average values of the distributions are plotted as vertical lines.

Δ_{SoC}	median	5-centile	25-centile	75-centile	95-centile
Deterministic	0.0103	-0.5090	-0.0976	0.1234	0.5060
Stochastic	0.0445	-0.4033	-0.0967	0.1272	0.6600

TABLE II: Micro problem: SDP vs DDP. Distribution of actual Δ_{SoC} for the iso-SoC case.

C. Macro problem

We next present the numerical simulations with the proposed bi-level method. We recall that this problem consists in optimizing the energy consumption of the vehicle over a whole travel Fig. 5, with a prescribed initial and final SoC.
In the numerical example, we take the travel as a sequence of 27 road segments and the SoC^r is discretized with a step size of 1%.

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. We illustrate on Fig. 11 the consumption map for segment #9, for the 4 traffic clusters. Note that the hierarchy between the four cost maps is consistent with the traffic conditions for the clusters (see also Fig. 6 and 7): cluster 1 has the higher speed and relies more heavily on the engine, hence it has a higher cost overall; cluster 2 has the lowest speed and also the lowest cost; clusters 3 and 4 correspond to intermediate traffic conditions.

Remark 6. *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. This could provide a reduction of the computational cost for obtaining these maps.*

Optimal policy for macro problem We show in Fig. 12 a sample optimal trajectory for the macro problem, for which

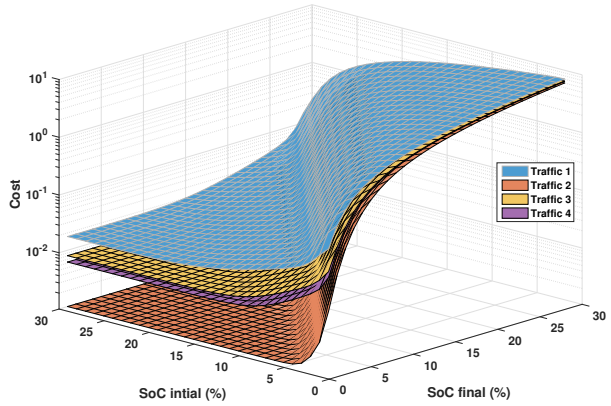

Fig. 11: Macro cost map v_s from the family of micro problems.

we also recomputed the sequence of micro problems along the travel. Top graph displays a real speed profile recorded thanks to 'Geco air'. Middle graph shows the optimal policy for the macro problem (i.e SoC^r), as well as the actual SoC corresponding to the sequence of micro problems. Bottom graph indicates the engine and motor torques of the vehicle recovered thanks to the value function of the micro problems. Finally, we also optimize the whole travel using DDP, taking the speed profile from the micro problems. The corresponding SoC trajectory is also shown on the middle graph.

For this simulation we set the conditions $SoC_0 = 0.25, SoC_f = 0.24$. We observe that both the macro and DDP solutions satisfy the final SoC correctly, although the SoC trajectories are different. The consumption is 0.2363 for bi-level and 0.1991 for DDP. The over-consumption of 18% for the bi-level method seems reasonable against the deterministic solution with full knowledge of the speed profile. The SoC trajectory from the micro problems initially diverges a bit from the macro target, but catches up later.

As with the micro problems, we now compare the bi-level method with the deterministic solution over a thousand known speed profiles. The distributions of the consumption and $\Delta_{SoC} := SoC(t_f) - SoC(t_0)$ are shown on Fig. 13. The average consumption obtained by the bi-level method is $6.58L/100km$, and the deterministic optimization obtained an average consumption of $5.70L/100km$. The average consumption is 15% higher for the stochastic bi-level approach, which seems reasonable considering the relatively large step size used for the discretization of SoC^r. Also, the bi-level Δ_{SoC} is centered around -1%, as expected, even if less strongly than for the deterministic approach, see Table III.

Δ_{SoC}	median	5-centile	25-centile	75-centile	95-centile
Deterministic	-0.9726	-1.7242	-0.9899	-0.9442	-0.2915
Bi-Level	-1.0339	-6.3671	-1.3986	-0.7537	0.1881

TABLE III: Macro problem: Bi-Level vs DDP. Distribution of actual Δ_{SoC} for depleting 1% of the battery.

Overall, the bi-level method gives satisfying results compared to a method with full traffic information.

CPU times The simulation times are recapped in Table IV. For the offline micro level, one SDP problem gives the cost map for all possible initial SoCs, so we only need to solve the family of problems for all possible final SoCs. Furthermore, all the micro problems are independent. On the other hand, the CPU time for the macro problem is quite small, as required for an online implementation. The key element for this fast macro problem is the handling of the stochastic traffic conditions at the micro level, leaving only a simple deterministic optimization at the macro level. For comparison purposes, we also include the time to optimize a full travel using DDP with complete traffic information, as well as an estimate of the cost for solving the same problem with stochastic traffic conditions.

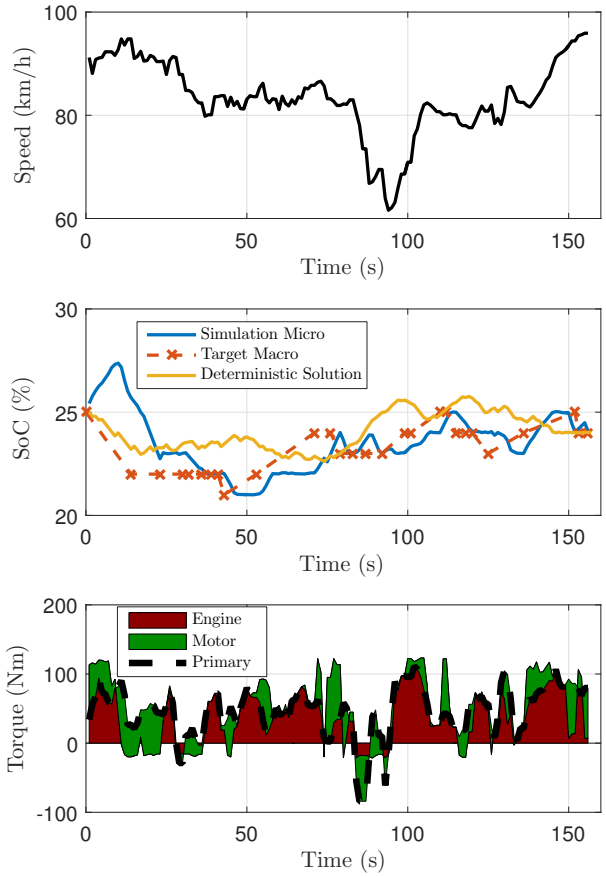

Fig. 12: Macro problem: optimal policy. Bi-level optimization vs DDP. Conditions $SoC_0 = 0.25, SoC_f = 0.24$.

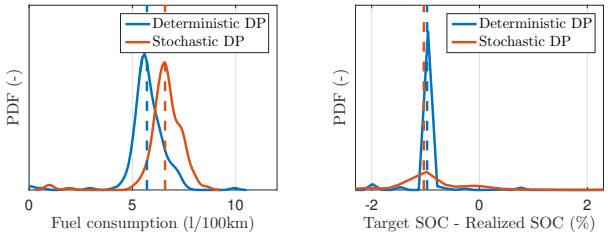

Fig. 13: Macro problem: Consumption distribution over 939 speed profiles. Bi-level optimization vs DDP. Conditions $SoC_0 = 0.25, SoC_f = 0.24$.

[3]with a 10×10 speed/acceleration discretization due to the linear complexity of SDP

	CPU time
Micro level (SDP, 1 segment, 1 SoC_f)	$\sim$ 1h
Macro level (DDP, deterministic SoC)	$\sim$ 10s
Reference: DDP over the whole travel	$\sim$ 2h
Estimation of the SDP over the travel [3]	($\sim$ 200h)

TABLE IV: CPU times

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 level, a fast, online optimization is performed to compute a deterministic approximation of the optimal SoC trajectory. This optimization relies on an energy cost map over all road segments, for all initial and final SoC conditions. We compute these costs at the lower level by solving offline a family of stochastic optimization problems under traffic constraints. Additionally, these offline solutions are also 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, with a limited over-consumption compared to a deterministic solution using fully known traffic information. Moreover, the obtained optimal policy for the vehicle (electric, thermal or hybrid mode) appears consistent with the traffic conditions observed on the road segments. The key point in order 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 work directions could include in particular a more accurate discretization for the macro problem in order to improve the prescribed SoC trajectory. It may also be interesting to perform a comparison with other bi-level approaches using e.g. MPC techniques. On the traffic model aspect, isolating data for a specific vehicle/driver could allow for some meta-clustering in order to further reduce the data size, as well as an increased 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] 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.

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

[12] 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

[13] 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.

[14] 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.

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

[16] 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

[17] 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.

[18] 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.

[19] 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.

[20] 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.

[21] 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.

[22] 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

[23] 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.

[24] C. Sun, F. Sun, X. Hu, J. K. Hedrick, and S. Moura, "Integrating Traffic Velocity Data into Predictive Energy Management of Plug-in Hybrid Electric Vehicles," in *American Control Conference (ACC)*, 2015.

[25] 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.

[26] 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

[27] M. Falcone and R. Ferretti, *Semi-Lagrangian Approximation Schemes for Linear and Hamilton–Jacobi Equations*. SIAM Publications, 2013.
[28] M. Bardi and I. Capuzzo-Dolcetta, *Optimal control and viscosity solutions of Hamilton-Jacobi-Bellman equations*. Birkhäuser, Boston, MA, 1997.
[29] 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.
[30] "Geco air," http://www.gecoair.fr/home_en/, [Online; accessed July 2019].

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.