Reducing the Computation Effort of a Hybrid Vehicle Predictive Energy Management Strategy

Sebastien Delprat, Mohamed Riad Boukhari

Reducing the Computation Effort of a Hybrid Vehicle Predictive Energy Management Strategy

Sébastien Delprat, Mohamed Riad Boukhari

***Abstract*—The present paper is dedicated to the investigation of a predictive Equivalent Consumption Minimization Strategy. The objective is to determine the torque split between the internal combustion engine and the electric machine of a hybrid vehicle. The energy management is formulated as a receding optimization problem. To avoid a complex prediction of the vehicle speed and acceleration over time, the slow dynamic of their distribution is exploited. A rational tuning of the algorithm parameters is proposed as well as some improved implementations. The number of individual operations (additions, multiplications, interpolations, etc) required per seconds is discussed. Finally, the energy management algorithm energy consumption are assessed over different driving cycles, including one with a 15406 km length obtained using GPS measurements. A comparison with an adaptive Equivalent Consumption Minimization Strategy is provided. The predictive Equivalent Consumption Minimization Strategy allows controlling the state of charge close to a (possibly time varying) set point while providing low fuel consumption.**

***Index Terms*—Hybrid Energy Management, Optimal Control, Pontryagin's Minimum Principe, predictive-ECMS**

I. Introduction

Hybrid vehicles use different energy sources to reduce their fuel consumption and CO_2 emissions. They require an energy management algorithm (EMA) to split the driver power demand between the different sources [1], [2]. EMA can be divided into two categories: off-line and on-line. In simulation, off-line algorithms use the *a priori* knowledge of the driving scenario to compute the optimal solution with a minimum fuel consumption. Different approaches have been investigated in the literature such as linear programming [3], dynamic programming [4], genetic algorithms [5], particle swarm optimization [6], and Pontryagin's Minimum Principle (PMP) [7], [8]. Despite the effectiveness of the off-line optimal control strategies, they remain non causal and can't be used for real time control.

On-line control strategies do not rely on future information and can be used in real time but they provide sub-optimal results. Rules based energy management algorithms implement heuristic control rules such as the deterministic rule based strategy [9], and variants of fuzzy rule based control strategies [10], [11].

Manuscript received 07, 07, 2020; revised 29, 01, 2021; accepted 18, 05, 2021

This work has been funded by the ELSAT 2020 project co-financed by the European Union with the European Regional Development Fund, the French state and the Hauts-de-France Region Council.

Sébastien Delprat (corresponding author) is with the INSA Hauts-de-France and Mohamed Riad Boukhari is with the University Polytechnique Hauts-de-France. Both are also with the LAMIH UMR CNRS 8201, 59313 Valenciennes CEDEX 09, France. e-mails : sebastien.delprat@uphf.fr, boukhari.mr@gmail.com

Finally, some others real time EMA are based on optimization. The Equivalent Consumption Minimization Strategies (ECMS) are derived from PMP. The control is obtained by minimizing a total consumption consisting in the Internal Combustion Engine (ICE) fuel consumption plus the electric energy consumption multiplied by an equivalence factor [12], [13]. This equivalence factor is similar to the PMP costate. It does not follow the optimal dynamic provided by PMP optimality conditions but it is instead used, in real time, to control the state of charge. Different control approaches lead to different ECMS variants such as Adaptive-ECMS [14] [15] [16] or the Telemetric-ECMS [17].

In this work, the predictive-ECMS is studied. Similarly to classical Model Predictive Control, an optimization problem is solved in a receding way [18]–[21]. It does not provides the control values to be applied to the powertrain at each instant, but the costate associated with the optimal control problem. This costate value is an essential parameter that drives the control strategy energy usage. It can be updated at a very low rate but it is used to compute the control value at a faster rate according to the actual driving conditions [22]. To design such a control law, two key elements are needed: (i) the predictor that provides a prediction of the future driving conditions (e.g. driver acceleration request and vehicle velocity) and (ii) an optimization algorithm that can be implemented online. Several predictive-ECMS variants have been proposed for hybrid vehicle energy management by combining different predictors and optimization algorithms:

- (i) Predictors : the Frozen-Time MPC has no need of *a priori* knowledge of future driving behavior and assumes that the power demand is constant over the time horizon [23]. Prescient MPC in contrast assumes that the future trip information are accurately known [24]. Exponential Varying MPC is based upon the assumption that the future driver torque demand is exponentially decreasing [25]. Stochastic MPC take advantage of the Markov chain to model the driver behavior and predicting the vehicle velocity and power demands [26], [27]. Neural network have also been used [28]. Velocity can be used base on pattern recognition [29]. The accuracy of the aforementioned predictors decreases when the prediction horizon increases. Connected vehicles (using GPS or V2x) tackle this issue by using additional traffic information [30] but still, the driver behavior remains difficult to predict.
- (ii) Many standard optimization algorithms can be used: mixed integer or sequential quadratic programming solvers, Interior Point methods [31]. Their implementa-

tion within industrial automotive applications remains yet often difficult, not to say impossible, due to either their computational time requirement and/or intrinsic complexity.

Overall, the energy consumption of the existing predictive control strategies are limited by the predictor performance. Standard optimization algorithms have a computational requirement that are far from being met by current automotive Electronic Control Unit. To the best of the authors' knowledge there is no study in the literature that investigate in depth the actual computational cost of hybrid vehicle predictive energy management strategies. In this work, our main contributions are to provide a rational tuning methodology of the predictive-ECMS parameters and a description on how to take benefits from singular control to reduce the computational load.

The manuscript is organized as follows: in section II, the considered hybrid architecture is presented and its optimal control using PMP is described in section III. Section IV is devoted to the presented *predictive-ECMS*. The prediction principle is introduced and validated in simulation. The control strategy tuning methodology is described. A single driving cycle with a wide range of driving conditions is sufficient for the tuning, therefore easing the tuning process. Different implementations are discussed in section V and allow reducing the computational requirement of the *predictive-ECMS*. In section VI, an *adaptive-ECMS* is introduced as a reference and the proposed implementations of the *predictive-ECMS* are compared in term of computational load, fuel consumption and state of charge control. Simulation results are provided over long driving cycles (> 200 km and up to 16406 km) that cover a wide range of encountered driving situations. Finally, conclusive remarks and perspectives are given in section VII.

II. Considered application

As an illustrative example, a parallel single shaft architecture is considered although the presented concepts can be applied to other hybrid vehicle architectures. The system topology is depicted in Fig. 1. Control signals are depicted in red.

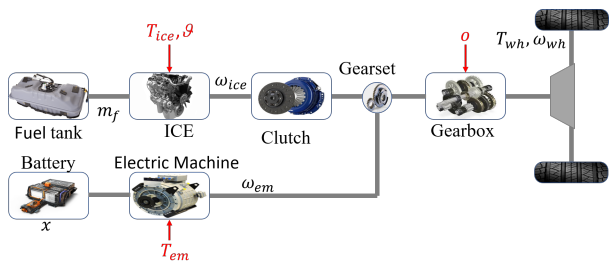

Fig. 1. Parallel hybrid vehicle model

The plug-in hybrid vehicle is built upon a conventional vehicle comprising an Internal Combustion Engine (ICE) coupled to a 5 gears robotized gearbox through a clutch. The Electric Machine is connected to the powertrain using gears. Further details on the powertrain and the vehicle's parameters are given in Table I.

TABLE I
Vehicle and powertrain parameters

Description	Value
Vehicle Mass	1469 kg
Engine Power	75.58 kW
Drag Coefficient	0.7280
Tire radius	0.3065 m
Tire rolling resistance coefficient	0.008
Nominal Battery voltage	390 V
Battery capacity	33.1 A h
Electric machine power	55 kW

The brake and throttle pedal are interpreted as an acceleration request. For each possible gears, a corresponding torque request is computed using a vehicle model. The EMA has to split this wheel torque request between the combustion engine and the electric machine, in such a way that the fuel consumption is minimized. As a result, the EMA controls the electric machine and ICE torques, ICE ignition signal (on/off) and the gears shifting. The combination of the ICE ignition on/off and engaged gear is denoted as an operating mode. The following notations are used: ω denotes a speed, T a torque, η an efficiency, x the battery state of charge and I an electric current. Subscript ice is used for Internal Combustion Engine, em for electric machine, gb for gearbox, gs for gearset, wh for wheels, bat for the battery and r for request. $R(o)$ is the gearbox ratio when the o^{th} gear is engaged. n_{gb} is the number of gears. ρ is the reduction ratio between the electric machine and the ICE shafts. ϑ is the engine injection signal (on/off). $\underline{(.)}$ and $\overline{(.)}$ denote a minimum and maximum respectively. When there is no ambiguity, in order to lighten expressions, the dependence on the time variable t is omitted. The exogenous variable is $z(t) = [v(t), a_{\mathrm{r}}(t)]^T$. i is a discrete time index, k is a measurement or prediction index, j is an index within a measurement or prediction horizon.

The EMA performances are evaluated using a quasi-static vehicle simulation model suitable for energy consumption studies [32]. The vehicle dynamic is given by :

$$\begin{aligned} M_{\mathrm{eq}} \cdot a(t) = &\frac{T_{\mathrm{wh}}(t)}{R_{\mathrm{tire}}} - MgC_{\mathrm{rr}} \cdot cos(\beta(t)) \\ &- 0.5 \cdot \rho_{\mathrm{air}} A_{\mathrm{f}} C_d v^2(t) - Mg \cdot sin(\beta(t)) \end{aligned} \tag{1}$$

where $v(t)$ and $a(t)$ are the vehicle longitudinal velocity and acceleration, T_{wh} the torque at the wheel, β the road slope, R_{tire} the tire radius, M the vehicle mass, g the gravitational acceleration, C_{rr} the rolling resistance coefficient, ρ_{air} the air density, A_{f} the front vehicle area, C_{d} is the drag coefficient. M_{eq} comprises the vehicle mass and the equivalent mass of the rotating inertia. It is defined by the equation (2).

$$M_{\mathrm{eq}} = M + J_{\mathrm{P}} \cdot r^{-2} \tag{2}$$

where $J_{\mathrm{P}} = (J_{\mathrm{ice}} + \rho^2 J_{\mathrm{m}}) R(o)^2$ is the powertrain inertia connected to the wheel shaft, r the wheel radius and $R(o)$ the ratio of the engaged gear o.

The mechanical powertrain architecture is described by the

following equations:

$$\omega_{\mathrm{wh}}(t) = \frac{v(t)}{R_{\mathrm{tire}}} = \frac{\omega_{\mathrm{ice}}(t)}{R(o(t))} = \frac{\omega_{\mathrm{em}}(t)}{R(o(t)) \cdot \rho} \tag{3}$$

$$\begin{aligned} T_{\mathrm{wh}}(t) = (\eta_{\mathrm{gb}})^{sign(T_{\mathrm{wh}}(t))} \cdot R(o(t)) \cdot (T_{\mathrm{ice}}(t) \cdot \vartheta(t) \\ + \rho \cdot (\eta_{\mathrm{gs}})^{sign(T_{\mathrm{em}}(t))} \cdot T_{\mathrm{em}}(t)) \end{aligned} \tag{4}$$

The engine and electric machine speed and torque are limited by mechanical constraints:

$$\begin{cases} \underline{\omega}_{\mathrm{ice}}(t) < \omega_{\mathrm{ice}}(t) < \overline{\omega}_{\mathrm{ice}}(t) \\ 0 < \omega_{\mathrm{em}}(t) < \overline{\omega}_{\mathrm{em}}(t) \end{cases} \tag{5}$$

$$\begin{cases} \underline{T}_{\mathrm{ice}}(\omega_{\mathrm{ice}}(t)) < T_{\mathrm{ice}}(t) < \overline{T}_{\mathrm{ice}}(\omega_{\mathrm{ice}}(t)) \\ \underline{T}_{\mathrm{em}}(\omega_{\mathrm{em}}(t)) < T_{\mathrm{em}}(t) < \overline{T}_{\mathrm{em}}(\omega_{\mathrm{em}}(t)) \end{cases} \tag{6}$$

The engine fuel mass flow (denoted as $\dot{m}_{\mathrm{f}}$) is a function of T_{ice} and ω_{ice} and is computed using linear interpolations between values experimentally measured on a test bench. The total fuel consumption (denoted m_{f}) to be minimized is given by the following equation:

$$m_{\mathrm{f}}(t) = \int_0^t \dot{m}_{\mathrm{f}}(T_{\mathrm{ice}}(t), \omega_{\mathrm{ice}}(t)) \cdot \vartheta(t) \cdot dt \tag{7}$$

The battery state of charge dynamic is given by the equation (8) where the battery current is a function of T_{em} and ω_{em}, [28], [31] :

$$\dot{x} = -\frac{1}{C} I_{\mathrm{bat}}(T_{\mathrm{em}}, \omega_{\mathrm{em}}) \tag{8}$$

where C is the battery pack capacity in As. Both the ICE engine fuel consumption $\dot{m}_{\mathrm{f}}$ and the battery current I_{bat} are computed using interpolations within look-up tables.

All the simulation results presented in this study have been obtained using a Simulink model similar to the ADVISOR software model [33]. It is a forward model, so a driver provides the acceleration request to the energy management algorithm such that the driving cycle is followed by the vehicle. The IC engine fuel consumption and electric machine current map have been measured on test bench. The simulated battery pack is built using LiFePo cells. In the operating range, the open circuit voltage is assumed to be constant [34].

III. Optimal energy management

Energy management can be formulated as an optimal control problem. Pontryagin's Minimum Principle is one of the classical approaches used to derive a solution [35]. In practice, PMP and Dynamic Programming provide very similar solutions [36]. The considered problem does not account for the battery state of charge limitations but a state constrained solution can be derived using the presented algorithm and the approach from [37].

The driver throttle and brake pedal request are interpreted as an acceleration request $a_r(t)$. From this request, the torque at the wheel $T_{\mathrm{wh}}(t)$ can be computed for each gear o using (1). To lighten the expression, the dependence of T_{wh} and ω_{wh} on the engaged gear is omitted when there is no ambiguity. The considered optimization horizon is $[t_0, t_{\mathrm{f}}]$ and the exogenous variable $z(t) = [v(t), a_r(t)]^T$ is supposed known over this horizon. The control variable is $u(t) = [T_{\mathrm{ice}}(t)\ \vartheta(t)\ o(t)]^T$. The optimal control problem to be solved is derived from (3)-(8):

$$\begin{aligned} \underset{u}{\text{minimize}} \quad & J[u] = \int_{t_0}^{t_{\mathrm{f}}} l(u(t), z(t)) \cdot dt \\ & \dot{x} = f(u(t), z(t)) \\ \text{subject to} \quad & u(t) \in \Phi(z(t)), \\ & x(t_0) = x_0, \\ & x(t_{\mathrm{f}}) = x_{\mathrm{f}} \end{aligned} \tag{9}$$

$\Phi(z)$ is the set of admissible control computed from (3)-(6). l is the fuel consumption obtained from (7) and (3). Similarly, f is the state of charge dynamic derived from (3), (4) and (8).

First, the Hamiltonian function associated with the optimal control problem (9) is defined.

$$H(u, z, \lambda) = l(u, z) + \lambda^T f(u, z) \tag{10}$$

where $\lambda : [t_0, t_{\mathrm{f}}] \to \mathbb{R}$ is called the costate. Assuming that the Hamiltonian is convex in u, Pontryagin's Minimum Principle provides optimality conditions fulfilled by the optimal solutions:

$$\dot{\lambda} = -\frac{\partial H(u, z, \lambda)}{\partial x} = 0 \tag{11}$$

$$u = \underset{u \in \Phi(z)}{\operatorname{argmin}}\, H(u, z, \lambda) \tag{12}$$

From (11), the costate λ is constant. Let us denote this constant as λ_0. From (12), the optimal policy Π that minimizes the Hamiltonian is written as :

$$\Pi(\lambda_0, z) = \underset{u \in \Phi(z)}{\operatorname{argmin}}\, H(u, z, \lambda_0) \tag{13}$$

The state of charge at the end of the optimization horizon is obtained by integrating the state dynamic:

$$x(t_{\mathrm{f}}) = x_0 + \int_{t_0}^{t_{\mathrm{f}}} f(\Pi(\lambda_0, z(t)), z(t)) \cdot dt \tag{14}$$

In order to facilitate the real time computation of the optimal solution, the integral in (14) is approximated using the Euler quadrature. Let i be the sampling index, s the sampling period with $t_0 = i_0 \cdot s$, $t_{\mathrm{f}} = (i_0 + N) \cdot s = i_{\mathrm{f}} \cdot s$:

$$x(t_{\mathrm{f}}) = x_0 + \sum_{i_0}^{i_{\mathrm{f}}-1} f(\Pi(\lambda_0, z(i \cdot s)), z(i \cdot s)) \cdot s \tag{15}$$

As a result, the optimal costate is obtained by solving the final state of charge constraint $x(t_{\mathrm{f}}) = x_{\mathrm{f}}$ and it is the root of the following defect function:

$$g_1(\lambda_0) = 0 \tag{16}$$

$$g_1(\lambda_0) = x_{\mathrm{f}} - x_0 - \sum_{i_0}^{i_{\mathrm{f}}-1} f(\Pi(\lambda_0, z(i \cdot s)), z(i \cdot s)) \cdot s. \tag{17}$$

Under mild assumptions, the function g_1 is monotonic [37] and so eq. (16) can be easily solved, for instance using a bisection approach. The resulting algorithm provides the optimal solution in simulation (since the driving cycle needs to be known beforehand) that can be used as a benchmark

for real time control strategies design. GPS measurements of a single vehicle have been collected during 3 years. These measurements have been processed to extract the different trip sections (between two vehicle rests) and inaccurate data (due for instance to GPS signal lost) have been discarded. The road slope is reconstructed from the altitude measurements after a careful data filtering. A first driving cycle has been synthesized and will be used for control strategy tuning and explanation purposes. It is denoted as the *Test driving cycle* and is depicted in Fig. 2. It lasts 4h17mins and 213.5 km and covers a wide range of driving conditions.

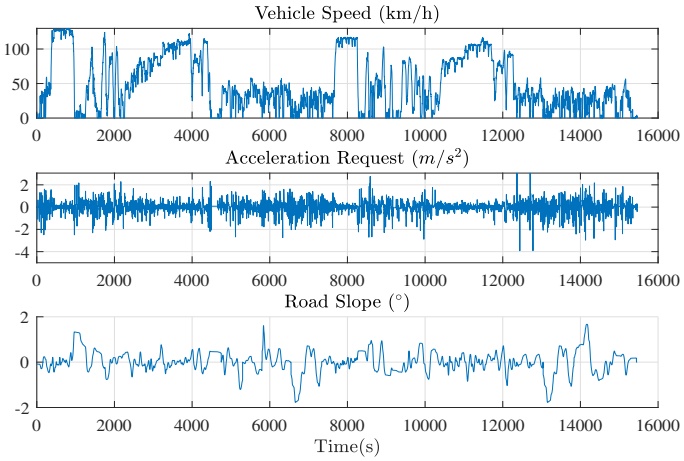

Fig. 2. *Test driving cycle* used for control strategy tuning

The optimal control results are depicted in Fig. 3. The initial state of charge is $x_0 = 50\%$ and the obtained final state of charge is $x(T) = 50.0049\%$ according to the chosen bisection algorithm accuracy (0.01%). The optimal costate, solution of $g_1(\lambda_0) = 0$ is $\lambda_0 = -12747.53$. The corresponding fuel consumption is 5.35 l/100km.

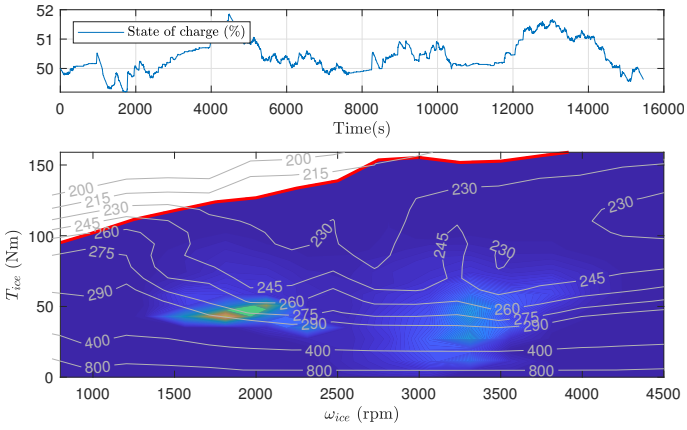

Fig. 3. Optimal control over the *Test driving cycle*. Top graph : state of charge. Bottom graph : heat-map of the IC engine operating points over its brake specific fuel consumption in g/kWh

IV. Predictive-ECMS

The *predictive-ECMS* strategy structure is depicted in Fig. 4. A predictor estimates the future driving conditions z over a prediction horizon of length N_{pred} samples. Every T_λ seconds, the problem (9) is solved along the prediction horizon using $i_0 = i$, $i_{\text{f}} = i_0 + N_{\text{pred}}$, $x_0 = x(t)$ such that the final state $x(i_{\text{f}} \cdot s)$ reaches a given reference value x_{f}. Finally, every T_s seconds, the control is computed using the last computed costate λ and the current driving conditions $z(t)$:

$$u(t) = \Pi(\lambda, z(t)) \tag{18}$$

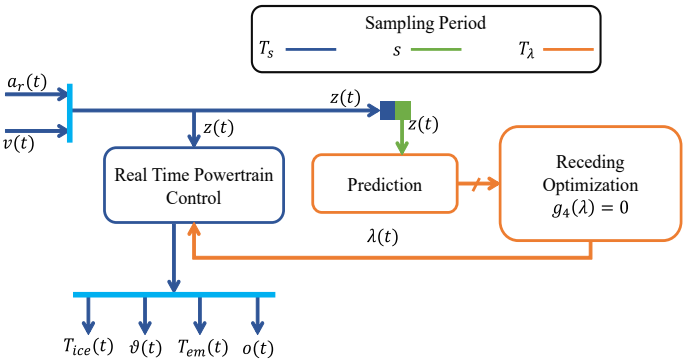

Fig. 4. Structure of the *predictive-ECMS*

A. Prediction of the exogenous variable

The optimal control problem to be solved for the EMA is highly dependent on the time varying exogenous variable z. In contrast to classical MPC, $z(t)$ cannot be simply considered as a disturbance whose effect has to be rejected and instead it has to be predicted. Unfortunately this prediction over a long enough prediction horizon is very difficult, most of the literature report $20s$ as a maximum prediction horizon [38]. In [22], the authors suggest to reformulate the defect function g_1 using the distribution (i.e. discrete probability density function) μ^+ of the quantified exogenous variable z over the prediction horizon $[i, i + N_{\text{pred}}]$:

$$g_2(\lambda) = x_{\text{f}} - x(i \cdot s) - N_{\text{pred}} \cdot \sum_{z \in Z} \mu^+(z) f(\Pi(\lambda, z), z) \cdot s \tag{19}$$

with $Z = \Phi_v \times \Phi_a$ and Φ_v (resp. Φ_a) a set of N_{v} (resp. N_a) values linearly spaced between the minimum and maximum vehicle speed (resp. acceleration). N_{v} and N_{a} provide control over the quantification step. The distribution has the following property $\sum_{z \in Z} \mu^+(z) = 1$. For a given value $\nu \in Z$, $\mu^+(\nu) = \frac{n_\nu}{N_{\text{pred}}}$ with n_ν the number of times that the quantified exogenous variable z takes the value ν over the interval $[i, i + N_{\text{pred}}]$.

Considering (19), it is not necessary to predict the vehicle velocity and acceleration request, but only their distribution. As this distribution is slowly varying over time, it is assumed to be locally constant. As a result, the predictor is reduced to the computation of the distribution μ^- obtained over the measurement horizon $[i - N_{\text{meas}}, i]$:

$$\mu^+ = \mu^- \tag{20}$$

The measurement length N_{meas} should be long enough to capture the dynamic of the exogenous variable. Finally, the

predicted costate λ is periodically updated by computing the root of the following defect function:

$$g_3(\lambda) = x_{\mathrm{f}} - x_0 - N_{\mathrm{pred}} \cdot \sum_{z \in Z} \mu^-(z) f(\Pi(\lambda, z), z) \cdot s \quad (21)$$

At last, for the sake of implementation, it should be noticed that, over the measurement horizon $[(i - N_{\mathrm{meas}}) \cdot s, i \cdot s]$, $z(t)$ has exactly the distribution μ^-. The following defect function implements the prediction assumption (20) while avoiding the quantification error:

$$g_4(\lambda) = x_{\mathrm{f}} - x_0 - \frac{N_{\mathrm{pred}}}{N_{\mathrm{meas}}} \cdot \sum_{i-N_{\mathrm{meas}}}^{i-1} f(\Pi(\lambda, z(i \cdot s)), z(i \cdot s)) \cdot s \quad (22)$$

B. Validation of the prediction scheme

An illustrative video of the considered prediction scheme is available online: https://pod.uphf.fr/video/1505-reducing-computation-effort-of-a-predictive-ecms. In order to assess the distribution based predictor, comparisons with other predictions schemes are proposed: (i) a constant acceleration prediction [39] and (ii) a decaying profile [40], [41]. Over each receding horizon, the predicted decaying acceleration $\hat{a}(t)$ is given by the following rule:

$$\hat{a}(t + j \cdot s) = a_r(t) \cdot e^{j \cdot s \cdot (\tau_{\mathrm{decay}})^{-1}} \quad j = 1 \cdots N_{\mathrm{pred}} \quad (23)$$

The velocity is computed using the vehicle dynamic (1). The costate λ is updated by solving $g_4(\lambda) = 0$ every $T_\lambda = 200$ s. In order to limit the number of operations, the receding optimization problem is solved using a sampling rate $s = 1$ s and the distribution is computed using $N_{\mathrm{meas}} = 200$ samples. The prediction horizon is $N_{\mathrm{pred}} = 200$. The receding optimization problems are solved every $T_\lambda = N_{\mathrm{pred}} \cdot s$. With the obtained λ, the actual powertrain control u is computed every $T_s = 0.1$ s to ensure a good tracking of the driver power demand. The initial state of charge is $x_0 = 50\%$. A sinusoidal state of charge reference is considered to emphasize the prediction errors:

$$x_{\mathrm{f}}(t) = 0.5 + 0.1 \cdot sin(2 \cdot \pi \cdot t/4000) \quad (24)$$

If the prediction was exact, the state of charge would exactly reach $x_{\mathrm{f}}\left((k+1) \cdot s \cdot N_{\mathrm{pred}}\right)$ at the end of the k^{th} prediction horizon. Between two measurement horizons, road slope variations, road traffic and driver behavior are likely to generate small variations in the exogenous variable distribution. As a result, the state of charge error at the end of the k^{th} prediction horizon is :

$$\epsilon_x(k) = x_{\mathrm{f}}\left(k \cdot N_{\mathrm{pred}} \cdot s\right) - x\left(k \cdot N_{\mathrm{pred}} \cdot s\right) \quad (25)$$

The state of charge trajectories and the state of charge target x_{f} are depicted in Fig. 5 for the three studied prediction schemes. The Root Mean Square of the error ϵ_x is 1.70% for the proposed scheme, 9.40% for the constant acceleration scheme and 10.03 % for the prediction based on the decay profile. These results confirm the effectiveness of the proposed scheme even when considering relatively long prediction horizons (200 s).

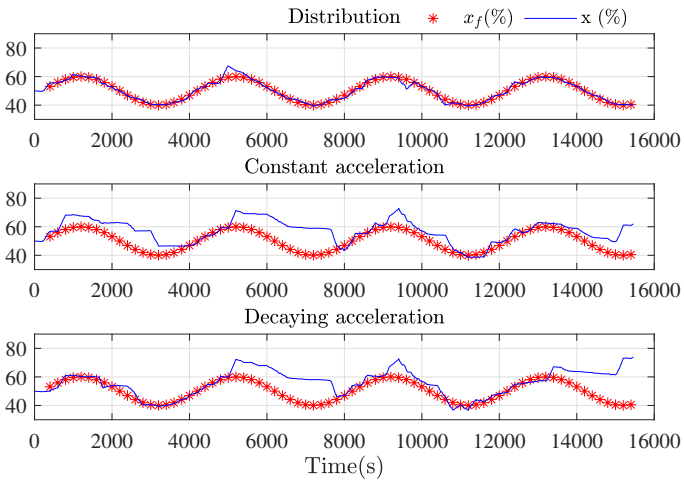

Fig. 5. Effect of the prediction accuracy over the *Test driving cycle*.

To further analyze the impact of the prediction horizon length N_{pred} on the prediction error, Fig. 6 summarizes the RMS values of the state of charge error ϵ_T obtained for different N_{pred} values and different prediction schemes. The simulations have been conducted for $N_{\mathrm{pred}} = N_{\mathrm{meas}} = T_\lambda/s$. Obviously using the velocity obtained by very simple prediction schemes such as constant acceleration or decaying acceleration does not make any sense for long prediction horizon. The suggested method provides significantly better results and moreover it does not require any additional computational effort for the prediction or any additional hardware (GPS, V2X devices).

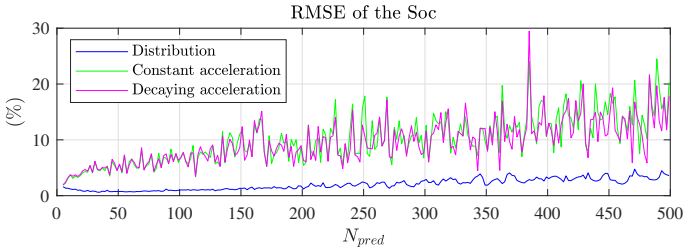

Fig. 6. Influence of the prediction horizon length N_{pred} on the RMS value of the SOC error ϵ_x.

C. Control strategy tuning

The real time powertrain control is operated using $T_s = 0.1$ s in order to provide a responsive feedback to the driver. As the energy consumption of the vehicle has a low bandwidth, sampling the exogenous variable z using $s = 1$ s is enough for the receding optimization problem. The choice of the other parameters has to ensure a tradeoff between potential prediction errors (that could lead to large deviation of the state of charge from its reference), fuel economy and computational requirements. The measurement horizon N_{meas} should be large enough to accurately capture the distribution of the exogenous variable z.

$N_{\mathrm{pred}} \cdot s$ is the theoretical time required for the state of charge to reach the target value if the prediction was perfect. Small values leads to very dynamic controller while longer values allows larger state of charge deviations. The costate update period T_λ should be small enough to capture sudden changes in the distribution of the exogenous variable, for instance when the vehicle leaves an urban area and enters a highway or when the road slope change significantly. The admissible values for T_λ are limited by the available computing power (since smaller T_λ leads to more computations).

Simulations have been conducted for different values of N_{pred}, N_{meas} and T_λ. The state of charge target x_{f} is kept constant all over the driving cycle and adjusted such that the actual final state of charge is $50\% \pm 0.5\%$. The criterion ϵ_x allows assessing the prediction errors but it does not represent well the instantaneous deviation from the state of charge target x_{f}. Instead, Δ_x, the RMS value of the instantaneous state of charge deviation, is introduced:

$$\Delta_x = \sqrt{\frac{1}{N}\sum_{i=0}^{N-1}\left(x(i\cdot s) - x_{\mathrm{f}}\right)^2} \tag{26}$$

with N the number of samples over the whole driving cycle.

The obtained results are depicted in Fig. 7. The upper graphs represent the RMS value of the instantaneous state of charge deviation Δ_x as a function of N_{meas} and N_{pred} for different T_λ values. Δ_x is quite sensitive to T_λ and, as expected, also slightly increases with N_{pred}. For $T_\lambda > 20s$, small values of N_{meas} or N_{pred} lead to larger Δ_x due to issues related to singular control (that will be explained in section V-C). It should be noticed that the state of charge error needs to be correlated with the large battery capacity (33.1 A h - 13 kW h): 1% of state of charge error still represent roughly 0.13 kW h of energy that allows to propel the vehicle at 50 km h^{-1} over 1.5 km. The lower graphs represent the corresponding fuel consumptions. For every considered T_λ, the minimum fuel consumption is depicted with a blue star. Lower fuel consumptions are obtained for large N_{pred} and N_{meas} with small T_λ. Unfortunately, this setting leads to a high computation load. Inside the domains limited by the red lines, the fuel consumption is almost constant and lower than 5.57 l/100km. Choosing $N_{\mathrm{pred}} = N_{\mathrm{meas}} = 80$ s for $T_\lambda \in \{5, 20\}$ leads to similar fuel consumptions, but $T_\lambda = 5$ s requires almost four times the computing power than $T_\lambda = 20$ s. As a result, the suggested tuning is $N_{\mathrm{pred}} = N_{\mathrm{meas}} = 80$ s and $T_\lambda = 20$ s.

V. Reduction of the computational load

As depicted in Fig. 4, the receding optimization algorithm consists in finding the roots of the defect function $g_4(\lambda)$. This is the most computational demanding part of the control strategy. We propose four different implementations of the receding optimization algorithm, ranging from a very basic and simplistic one up to the most efficient one.

A. Baseline algorithm

The core of the optimization algorithm is the Hamiltonian minimization in (12). To reduce the computational cost, an approach consists in fitting analytical models to the engine fuel consumption $\dot{m}_{\mathrm{f}}$ and electric machine current I_{bat} data [42], [43]. The choice of model is restricted by the need of analytical solution to (12). In general, to achieve a good fitting accuracy, piecewise quadratic models are considered but for a given engine and electric machine there is no prior guaranty on the resulting accuracy. Instead, we consider a more classical and generic approach that consists in solving the Hamiltonian minimization numerically using interpolation in look-up tables. The computational cost and accuracy of this approach is of course highly depending on the grid density.

Let us define a grid of ICE torque values regularly-spaced between the minimum $\underline{T}'_{\mathrm{ice}}$ and maximum $\overline{T}'_{\mathrm{ice}}$ torque according to (6) and (4), using δ_{Tice} as an increment.

$$T_{grid}(z,o) = \overline{T}'_{\mathrm{ice}}(z,o) \cup \{T_i \in \left[\underline{T}'_{\mathrm{ice}}(z,o), \overline{T}'_{\mathrm{ice}}(z,o)\right], \; i \in \mathbb{Z}^+\} \tag{27}$$

with $T_i = \underline{T}'_{\mathrm{ice}}(z,o) + i \cdot \delta_{Tice}$,

$$\overline{T}'_{\mathrm{ice}}(z,o) = min\left(\overline{T}_{\mathrm{ice}}\left(\omega_{\mathrm{ice}}\left(t\right)\right), \frac{T_{\mathrm{wh}}\left(t\right)}{R\left(o\left(t\right)\right)} \cdot \left(\eta_{\mathrm{gb}}\right)^{-sign(T_{\mathrm{wh}}(t))} - \frac{\rho}{\eta_{\mathrm{gs}}} \cdot \underline{T}_{\mathrm{em}}\left(\omega_{\mathrm{em}}\left(t\right)\right)\right) \tag{28}$$

$$\underline{T}'_{\mathrm{ice}}(z,o) = max\left(\underline{T}_{\mathrm{ice}}\left(\omega_{\mathrm{ice}}\left(t\right)\right), \frac{T_{\mathrm{wh}}\left(t\right)}{R\left(o\left(t\right)\right)} \cdot \left(\eta_{\mathrm{gb}}\right)^{-sign(T_{\mathrm{wh}}(t))} - \rho \cdot \eta_{\mathrm{gs}} \cdot \overline{T}_{\mathrm{em}}\left(\omega_{\mathrm{em}}\left(t\right)\right)\right) \tag{29}$$

The whole control grid, denoted as U_{grid}, is created by considering the torque grid and the pure electric mode for all gears. The Hamiltonian minimization (12) is approximated by:

$$u = \underset{u \in U_{grid}}{\operatorname{argmin}} H(u, z, \lambda) \tag{30}$$

Obviously, the deviation between the actual optimal control (12) and the gridded approximation (30) depends on the chosen grid increment δ_{Tice}. A compromise between accuracy and computation load has to be found.

Let us first investigate the number of f and l evaluations required to compute the value of λ. For a given value of the exogenous variable z, the fuel consumption and electric machine current have to be evaluated for every usable gears and all control values in $U_{grid}(z,o)$ (+1 evaluation for the electric machine current if the pure electric mode is feasible). This process has to be repeated for all the N_{meas} samples and also for all the iterations required by the bisection. This number is limited to $m_{bs} = 200$ to limit the computational load. This algorithm only requires very little additional memory to store the intermediate data but it is extremely demanding due to the imbrication of the different loops as depicted in Algorithm 1.

B. Algorithm with precomputation

The Hamiltonian is an affine function in λ, as a result, for a given optimization problem, the values of the function l and f can be precomputed and stored for all N_{meas}, n_{gb} and all the

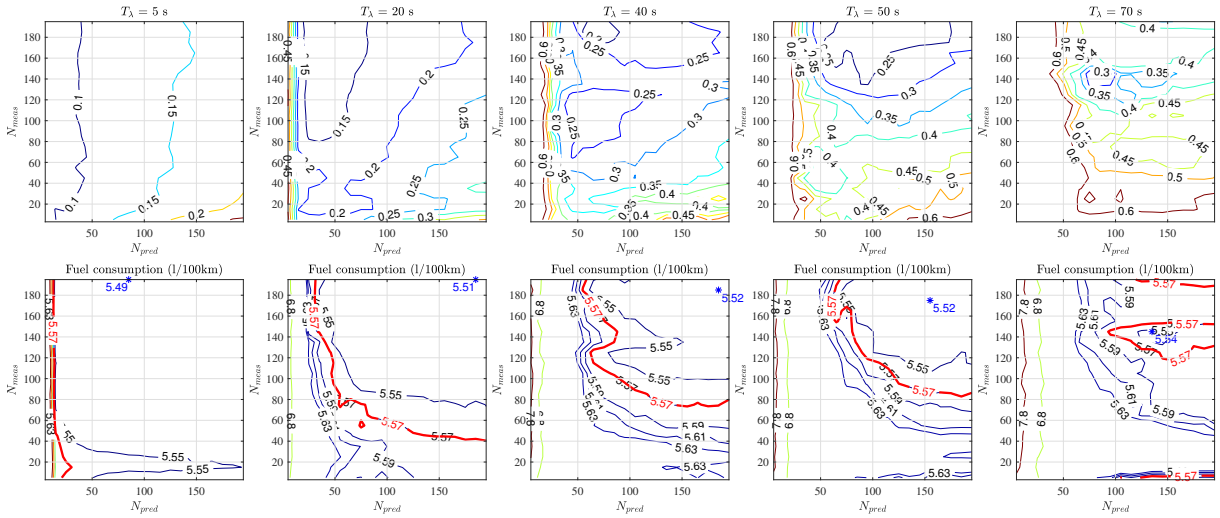

Fig. 7. Effect of the control strategy tuning on the fuel consumption and RMS value of the instantaneous state of charge error Δ_x. For every T_λ values, the minimum fuel consumption is depicted using a blue star. The fuel consumption inside the domain limited by the red thick line has limited variations.

Algorithm 1: Baseline algorithm

Parameters: $\underline{\lambda}$, $\overline{\lambda}$, Tol_x, Tol_λ
Input: z
Output: λ
$ended \leftarrow false$
while *not(ended)* **do**
  $\lambda \leftarrow (\underline{\lambda} + \overline{\lambda})/2$
  $g \leftarrow g_4(\lambda)$
  $ended \leftarrow |g| < Tol_x \quad OR \quad \left|\overline{\lambda} - \underline{\lambda}\right| < Tol_\lambda$
  $\left\{\underline{\lambda}, \overline{\lambda}\right\} \leftarrow BisectionUpdate(g, \underline{\lambda}, \overline{\lambda})$
Function $g_4(\lambda)$**:**
  for $\forall j \in \{1, ..., N_{\text{meas}}\}$ **do**
    for $\forall o \in \{1, ..., n_{\text{gb}}\}$ **do**
      for $\forall p \in \{1, ..., card\left(T_{grid}\left(z_{i-j}, o\right)\right)\}$ **do**
        $u_{on} \leftarrow [T_{\text{ice}}(p), 1, o]^T$
        $H_{on}(j, o, p) \leftarrow l\left(u_{on}, z_j\right) + \lambda \cdot f\left(u_{on}, z_j\right)$
      $u_{off} \leftarrow [\underline{T_{\text{ice}}}, 0, o]^T$
      $H_{off}(j, o) = f\left(u_{off}, z_j\right)$
      $u(j) \leftarrow argmin\{H_{on}(j, o, :) \cup H_{off}(j, o)\}$
    $g_4\left(\lambda\right) \leftarrow$
    $x_{\text{f}} - x_0 + \frac{N_{\text{pred}}}{N_{\text{meas}}} \cdot \sum_{j=0}^{N_{\text{meas}}-1} f\left(u(j), z_{i-j}\right) \cdot s$

values in U_{grid}. The computation of $g_4(\lambda)$ is then significantly improved. The resulting program structure is depicted in the Algorithm. 2.

C. Singular control

Singular control issues arise when the Hamiltonian minimization does not provide enough information to determine the control value. It has been analyzed in depth in [7] for the particular case of the hybrid vehicle energy management. When look-up tables with linear interpolations are used to model the fuel consumption $l(u, z)$ and the battery current

Algorithm 2: Algorithm with precomputation

Parameters: $\underline{\lambda}$, $\overline{\lambda}$, Tol_x, Tol_λ
Input: z
Output: λ
$[F_{on}, L_{on}, F_{off}] \leftarrow Precompute$
$ended \leftarrow false$
while *not(ended)* **do**
  $\lambda \leftarrow (\underline{\lambda} + \overline{\lambda})/2$
  $ended \leftarrow |g_4(\lambda, F_{on}, L_{on}, \underline{F_{off}})| < Tol_x$
  $ended \leftarrow ended \quad OR \quad \left|\overline{\lambda} - \underline{\lambda}\right| < Tol_\lambda$
  $\left\{\underline{\lambda}, \overline{\lambda}\right\} \leftarrow BisectionUpdate(g_4(\lambda), \underline{\lambda}, \overline{\lambda})$
Function $Precompute \rightarrow [F_{on}, L_{on}, F_{off}]$**:**
  for $\forall j \in \{1, ..., N_{\text{meas}}\}$ **do**
    for $\forall o \in \{1, ..., n_{\text{gb}}\}$ **do**
      for $\forall p \in \{1, ..., card\left(T_{grid}\left(z_{i-j}, o\right)\right)\}$ **do**
        $u_{on} \leftarrow [T_{\text{ice}}(p), 1, o]^T$
        $L_{on}(j, k, o) \leftarrow l\left(u_{on}, z_j\right)$
        $F_{on}(j, k, o) \leftarrow f\left(u_{on}, z_j\right)$
      $u_{off} \leftarrow [\underline{T_{\text{ice}}}, 0, o]^T$
      $F_{off}(j, o) \leftarrow f\left(u_{off}, z_j\right)$
Function $g_4\left(\lambda, F_{on}, L_{on}, F_{off}\right)$**:**
  for $\forall j \in \{1, ..., N_{\text{meas}}\}$ **do**
    for $\forall o \in \{1, ..., n_{\text{gb}}\}$ **do**
      for $\forall p \in \{1, ..., card\left(T_{grid}\left(z_{i-j}, o\right)\right)\}$ **do**
        $u_{on} \leftarrow [T_{\text{ice}}(p), 1, o]^T$
        $H_{on}(j, o, p) \leftarrow$
          $L_{on}\left(j, k, o\right) + \lambda \cdot F_{on}\left(j, k, o\right)$
      $H_{off}(j, o) = F_{off}\left(j, o\right)$
      $u(j) \leftarrow argmin\{H_{on}(j, o, :) \cup H_{off}(j, o)\}$
    $g_4\left(\lambda\right) \leftarrow$
    $x_{\text{f}} - x_0 + \frac{N_{\text{pred}}}{N_{\text{meas}}} \cdot \sum_{j=0}^{N_{\text{meas}}-1} f\left(u(j), z_{i-j}\right) \cdot s$

$f(u,z)$, the Hamiltonian (10) is a piecewise linear function as depicted in Fig. 8. Let us denote by α_l (resp. α_f) the number of torque vertices of the fuel consumption (resp. battery current) lookup table. The Hamiltonian contains at most $\alpha_l+\alpha_\mathrm{f}$ vertices. These vertices are the fuel consumption look-up table vertices and the ICE torque corresponding to the electric machine torque used to construct the battery current look-up table (They can be computed by solving (4) for given $z(t)$ and all T_em in the battery current look-up table vertices).

As depicted in Fig. 8, depending on the costate λ, two different situations can occur. For some singular costate values, denoted by λ_s, the Hamiltonian minimum is undetermined. For other costate values $\lambda \neq \lambda_s$, the Hamiltonian minimum is one of its vertices. The same situation occurs with the ICE state ϑ : due to the linear interpolations in the look-up tables, the Hamiltonian is an affine function of the ICE state and for some costate value, both $\vartheta(t)=0$ and $\vartheta(t)=1$ are simultaneously optimal.

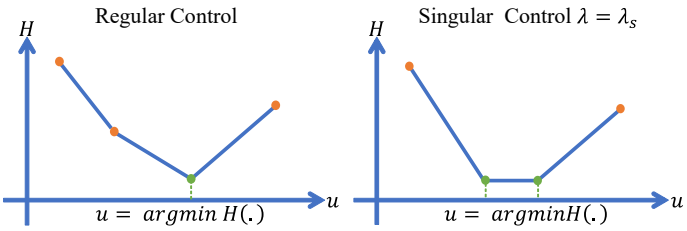

Fig. 8. Illustration of regular and singular control cases. For some costate λ, the Hamiltonian minimum is not unique. The optimal control cannot be determined from the Hamiltonian minimization alone

The effect of non unique optimal control values on the final state of charge is illustrated in Fig. 9. For the singular costate values λ_s, as multiple control values are optimal, multiple final state of charge values are obtained. The function $g_4(\lambda)$ is then discontinuous and $g_4(\lambda)=0$ cannot be accurately solved for some x_f values. The bisection in the Algorithm 1 will not converge since the problem has no solution if a constant sampling period is considered [7]. In practice, the algorithm will stop after a maximum number of iterations is reached. The shape of the function g_4, subject to multiple discontinuities, enforces the choice of a derivative-free algorithm such as the bisection, for the root-finding algorithm.

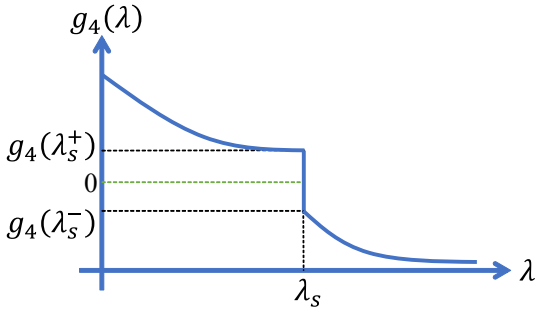

Fig. 9. Defect function $g_4(\lambda)$ in the vicinity of a singular costate value λ_s. The defect function is discontinuous and $g_4(\lambda)=0$ does not admit any solution.

From this analysis, the algorithm with precomputation can be improved. First, for $\vartheta=1$, the grid of ICE torque candidates $T_{grid}(z,o)$ (27), used to construct $U_{grid}(z,o)$ should contain only the fuel consumption look-up table vertices and the ICE torques obtained by solving (4) for given $z(t)$ and all T_em in the battery current look-up table vertices. This allows reducing the number of control candidates while simultaneously canceling the error due to the grid in (30).

Second, the bisection algorithm convergence toward $g(\lambda_s^+)$ and $g(\lambda_s^-)$ instead of 0 should be carefully monitored so the singular control issue depicted in Fig. 9 can be detected as early as possible. Algorithmic details about this detection are available in [7].

D. Exploiting the distribution to reduce the computational load

It is possible to reduce even further the computational load, but at the price of a higher memory requirement. The costate λ can be computed as a solution to $g_3(\lambda)=0$ with g_3 given in (21). The Hamiltonian needs to be minimized only for $z \in Z$ corresponding to the cell centers of a rectangular grid Z constructed using N_v speed values linearly spaced between the vehicle minimum and maximum speed and N_a values between the vehicle minimum and maximum acceleration.

The Hamiltonian minimization can be computed off-line and stored in memory as depicted in the Algorithm 3 (F_{on}, L_{on} and F_{off} are the precomputed values). As the set Z is finite, a quantification errors will occur and a tradeoff between memory usage and additional fuel consumption resulting from the quantification error has to be found. Several simulations have been conducted for different N_a and N_v values and results are summarized in Fig. 10. For each simulation, the initial state of charge is 50% and the state of charge target x_f is adjusted such that the final state of charge is $50\% \pm 0.5\%$. The memory requirement is proportional to $N_\mathrm{a} \cdot N_\mathrm{v}$. In order to assess the impact of the quantification errors, a criterion Δ_λ is adopted. This criterion represents the RMS deviation between the costate value $\lambda_{distrib}$ obtained using the distribution (by solving $g_3(\lambda_{distrib})=0$) and the costate value λ_{ref} obtained using all the measured samples (by solving $g_4(\lambda_{ref})=0$) :

$$\Delta_\lambda = \sqrt{\frac{1}{N}\sum_{i=0}^{N-1}\left(\lambda_{distrib}(i\cdot s)-\lambda_{ref}(i\cdot s)\right)^2} \qquad (31)$$

Within the area delimited by the red line in Fig. 10 (corresponding to $N_\mathrm{a} \geq 30$ and $N_\mathrm{v} \geq 20$), the costate error Δ_λ, the fuel consumption and the RMS state of charge error Δ_x remain low. The suggested tuning is therefore $N_\mathrm{a}=30$ and $N_\mathrm{v}=20$ and requires 2.9Mb of memory to store the precomputed data. Slightly smaller N_v values could also lead to good fuel consumption but larger Δ_λ and Δ_x.

The distribution $\mu(z)$ is very sparse: for instance, when driving in highway conditions, all the $\mu(z)$ for z corresponding to 'low speed' will be zero and will not contribute to the final state of charge. This allows reducing further the computational load of $g_3(\lambda)$. For instance, over the *Test driving cycle*, with $N_\mathrm{a}=30$ and $N_\mathrm{v}=20$, as depicted in Fig. 11, at most 36 cells are used simultaneously and on average only 14 cells are

Algorithm 3: Algorithm with Distribution

Parameters: $\underline{\lambda}$, $\overline{\lambda}$, Tol_x, Z, F_{on}, L_{on}, F_{off}
Input: μ^-
Output: λ
$ended \leftarrow false$
while *not(ended)* **do**
  $\lambda \leftarrow (\underline{\lambda} + \overline{\lambda})/2$
  $g \leftarrow g_3(\lambda, \mu^-, F_{on}, L_{on}, F_{off})$
  $ended \leftarrow |g| < Tol_x \quad OR \quad \left|\overline{\lambda} - \underline{\lambda}\right| < Tol_\lambda$
  $\left\{\underline{\lambda}, \overline{\lambda}\right\} \leftarrow BisectionUpdate(g, \underline{\lambda}, \overline{\lambda})$
end
Function $g_3\ (\lambda, \mu^-, Z, F_{on}, L_{on}, F_{off})$**:**
  $g_3 \leftarrow 0$
  for $\forall j \in \{1, ..., card(Z)\}$ **do**
    if $\mu(j) \neq 0$ **then**
      for $\forall o \in \{1, ..., n_{\text{gb}}\}$ **do**
        for $\forall p \in \{1, ..., card\,(T_{grid}\,(z_{i-j}, o))\}$ **do**
          $u_{on} \leftarrow [T_{\text{ice}}(p), 1, o]^T$
          $H_{on}(j, o, p) \leftarrow$
            $L_{on}\,(j, k, o) + \lambda \cdot F_{on}\,(j, k, o)$
        end
        $H_{off}(j, o) = F_{off}\,(j, o)$
        $u(j) \leftarrow argmin\{H_{on}(j, o, :$
          $) \cup H_{off}(j, o)\}$
      end
      $g_3 \leftarrow g_3 + N_{\text{pred}} \cdot \mu^-(j) \cdot f\,(u(j), z(j)) \cdot s$
  end
end

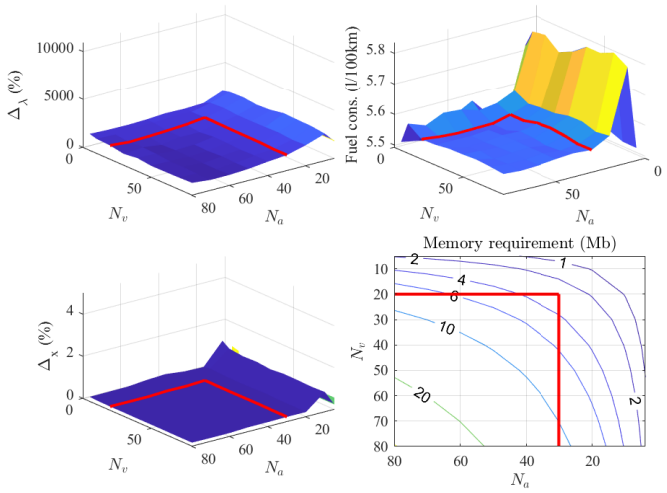

Fig. 10. Analysis of the quantification effects. N_{a} and N_{v} control the number of values contained in the set Z used to estimate the distribution of the exogenous variable. The area inside the red line corresponds to tunings with roughly similar performances

used. The number of operations (additions and multiplications) required by the algorithm over different driving cycles in the particular case $N_a = N_v$ is represented in Fig. 12.

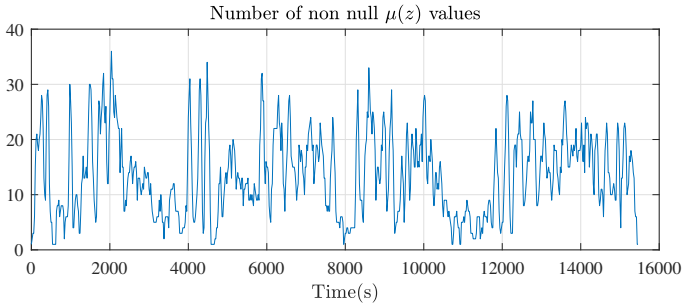

Fig. 11. Sparsity of the distribution $\mu(z) \forall z \in Z$

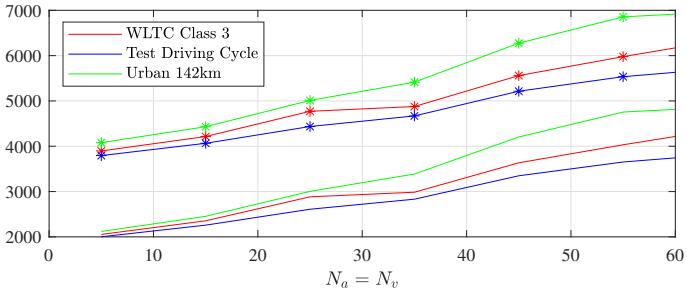

Fig. 12. Computation load as a function of $N_a = N_v$

VI. Algorithms Assessment

Laboratory equipments such as dSpace MicroAutobox, National Instrument Compact RIO, or even a PC have enough computational power to easily implement predictive energy management algorithms. Nevertheless, automotive electronic control units have less computing power available for the energy management and a careful estimation and optimization of the computational load is needed. The time required to perform a Simulink simulation on a desktop computer is not a reliable indicator of the computational load. Instead, we suggest to count the number of basic operations (additions, multiplications and interpolations) per simulated seconds as an indicator of the computational load. In the first subsection, to analyze the different implementations of the *predictive-ECMS*, the *adaptive-ECMS* is introduced and will be used as a reference. Then the computational load and the memory requirement of the studied algorithms are evaluated on the *Test driving cycle*. Finally, the fuel consumptions obtained on various driving cycles are discussed.

A. Adaptive-ECMS

The *adaptive-ECMS* strategy from [16] is chosen as a reference. It uses the real time powertrain control block depicted in Fig. 4. The costate λ is computed using a discrete time update rule, where k is the discrete time index :

$$\lambda(k+1) = \frac{\lambda(k-1) + \lambda(k-2)}{2} + K_p \frac{x(k) - x_{\text{f}}(k)}{T_\lambda} \quad (32)$$

This control strategy has two parameters : The costate update period T_λ and the control gain K_p. After an exhaustive search on the *Test driving cycle*, the following tuning is chosen $T_\lambda = 200$ s and $K_p = 2 \cdot 10^7$. This control strategy appears to be very sensitive to the initialization of the costate (i.e. the

$\lambda(-1)$ and $\lambda(-2)$ values in (32) required to compute the first costate value for $k = 0$). These initial values have been chosen in the vicinity of the optimal costate: $\lambda(-1) = \lambda(-2) = -15000$. The numerical Hamiltonian minimization from the *predictive-ECMS* 'singular control' implementation is reused. Simulations have been conducted over the *Test driving cycle*. Results are presented in Fig. 13. The state of charge target $x_{\mathrm{f}} = 49\%$ has been adjusted so the final state of charge is 49.63%. The obtained fuel consumption is $5.88\ \mathrm{l/100km}$ and the RMS of the state of charge error is $\Delta_x = 2.71\%$. These results confirm that the strategy performs well on the *Test driving cycle*.

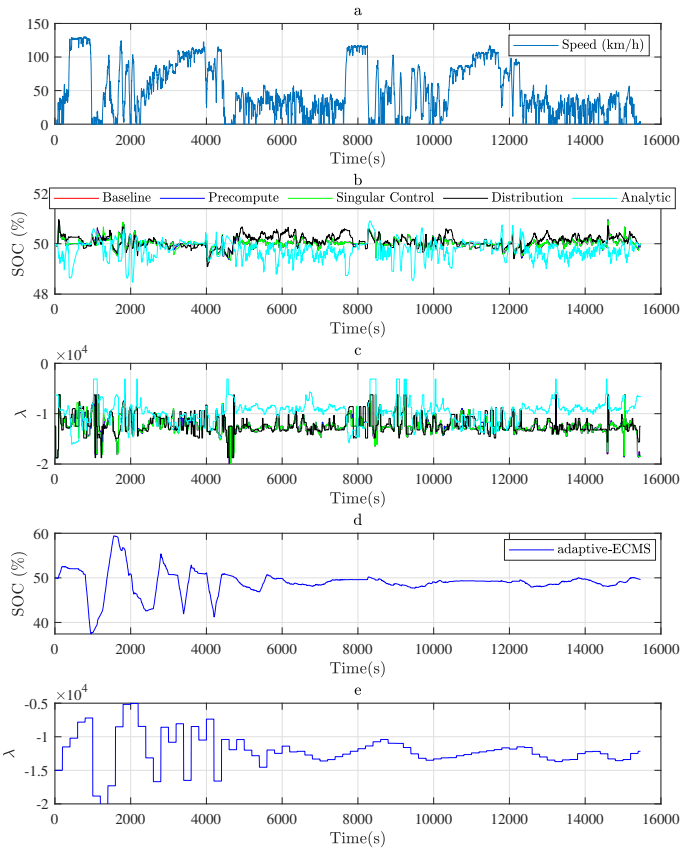

Fig. 13. Result obtained on the *Test driving cycle*: **(a)** vehicle speed, **(b)** state of charge of the predictive ECMS strategies, **(c)** costate of the predictive-ECMS strategies, **(d)** state of charge of the adaptive-ECMS, **(e)** costate of the adaptive-ECMS.

B. Computational load assessment on the Test driving cycle

Five control strategies have been presented so far : *adaptive-ECMS* and four different implementations of the *predictive-ECMS* described in section V. They are denoted as *Base Line*, *Precomputation*, *Singular control* and *Distribution*. The main objective of this study being the computational load evaluation, a sixth algorithm denoted as *analytic* is also considered. It is similar to the *Singular control* but uses polynomial models of the fuel consumption and electric machine current as described in [44]:

$$\dot{m}_{\mathrm{f}} = \sum_{\mathrm{i}=0}^{2}\sum_{\mathrm{j}=0}^{5} a_{\mathrm{ij}}(T_{\mathrm{ice}})^i \cdot (\omega_{\mathrm{ice}})^j \tag{33}$$

$$\begin{aligned} I_{\mathrm{bat}} = \sum_{\mathrm{j}=0}^{5} &(\alpha_{\mathrm{j}} \cdot (T_{\mathrm{em}})^2 + \beta_{1\mathrm{j}} \cdot \Gamma(T_{\mathrm{em}}) \cdot T_{\mathrm{em}} \\ &+\beta_{2\mathrm{j}} \cdot \Gamma(-T_{\mathrm{em}}) \cdot T_{\mathrm{em}}) \cdot (\omega_{\mathrm{em}})^j \end{aligned} \tag{34}$$

with Γ the Heavyside function. For these analytical models, the optimal policy (13) can be determined straightforwardly [44]. As the Hamiltonian minimization is computed explicitly, the computation load is significantly reduced but the result accuracy depends on the quality of the fitting.

These six algorithms have been implemented using pure Simulink coding. The computational requirements and the fuel consumption of the *adaptive-ECMS* are used as a reference. The costate update rule (32) being very simple, most of the required computations are due to the real time powertrain control bloc depicted in Fig. 4. Any ECMS variant will have at least to implement this block. So the obtained results are representative of most of the basic ECMS with lightweight costate update algorithm.

The considered algorithms have been simulated over the *Test driving cycle* using the same initial state of charge $x_0 = 50\%$. The state of charge target x_{f} has been adjusted such that the final state of charge is $50\% \pm 0.5\%$. The initial costate values of the *adaptive-ECMS* have been chosen $\lambda(-1) = \lambda(-2) = -15000$ and the initial distribution used by the *predictive-ECMS* has been set to $\mu(z) = 0$. In order to compare the fuel consumptions despite the small deviation of the final state of charge, a corrected fuel consumption is computed using the deviation of the final state of charge deviation and the optimal co-state. The obtained results are depicted in Fig. 13. For every simulation, the number of individual operations (additions, multiplications and interpolations) has been counted and divided by the cycle duration ($15460s$) to obtain the average computational load. Interpolations have been implemented using Pre-Lookup to reduce the computational load. When operations are applied to a vector, the number of operations is multiplied by the vector size. The total operations per seconds refers to the total number of block executions (e.g. Relational operator, saturation, etc. including additions, multiplications and interpolations). The required computation time is also given as an indication. Results are summarized in Table II. The *adaptive-ECMS* performances are recalled in the first column. The obtained corrected fuel consumption, $5.89\ \mathrm{l/100km}$, is close to the optimal one ($5.35\ \mathrm{l/100km}$) but the state of charge regulation is quite poor: $\Delta_x = 2.71\%$) which is about six times higher than the result of any *predictive-ECMS* implementation. The *Baseline* implementation of the *predictive-ECMS* is quite inefficient since the number of required operations is 24 times greater than the one from the *adaptive-ECMS*. The fuel economy is slightly better: $5.44\ \mathrm{l/100km}$ with a very good state of charge tracking $\Delta_x = 0.18\ \%$. The *Precomputation* implementation allows reducing the computational requirement approximately by a factor two compared with the *Baseline* implementation for the same energy usage (fuel consumption and state of charge regulation). This comes at the price of a higher memory usage, about $1.6\ \mathrm{Mb}$. The *Singular control* implementation allows

reducing the number of elementary operations even further: it is now only 2.4 times greater than the *adaptive-ECMS* and also allows reducing the memory usage to 0.4Mb. This is due to the fact that the stored control grid U_{grid} is built using only the meaningful look-up table vertices, thereby reducing the grid size. Finally, the *Distribution* implementation is the most computational efficient version of the *predictive-ECMS*. It only requires 1.2 times the number of total operations than *adaptive-ECMS* and this comes at the price of a higher memory usage, 2.8 Mb. On this driving cycle, the main *predictive-ECMS* benefits is not the fuel consumption but the very good state of charge tracking control. Finally, the *analytic* implementation has a low number of operations (14 190), in the same range as the *distribution* implementation (13 305). However, the quality of the polynomial fitting is not good enough to capture accurately the non-linearities of the fuel consumption (they can be clearly seen in the brake specific fuel consumption map in Fig. 3). As a result, the corrected fuel consumption of the *analytic* implementation, $6.18\ \mathrm{l/100km}$, is significantly higher than the *distribution* implementation ($5.44\ \mathrm{l/100km}$). However, it should be noticed that, when the analytic model fits well the data, the *analytic* implementation is very efficient and leads to performances similar as the other implementations [43]–[45].

As a conclusion, only the *Singular control* and *Distribution* implementations of the *predictive-ECMS* should be considered depending on the available memory and computational power.

C. State of charge control and fuel consumption assessment

The state of charge reference signal x_{f} can be chosen as a constant, for instance for vehicle equipped with a small battery capacity [16]. It can also be speed and altitude dependent to account for the vehicle potential energy storage. For plug-in hybrids, different approaches can be considered : after a full recharge, the battery is typically depleted until a given threshold is reached, then the energy management algorithm switches to charge sustaining until the next recharge. The charge depleting may be more or less sophisticated according to the vehicle connectivity and available information. Whatever the chosen setpoint is, the state of charge should be controlled nearby the reference while providing the lowest fuel economy. Two different simulation settings are considered: a constant state of charge setpoint and a charge depleting approach.

Different driving cycles are also considered. First, a $142\ \mathrm{km}$ long *urban driving cycle* has been created by concatenating trip sections from our database with a maximum speed lower than $60\ \mathrm{km/h}$. Second, a $8862\ \mathrm{km}$ long *highway driving cycle* has been created by concatenating trip sections with a max speed greater than $110\ \mathrm{km}/h$. Finally, a driving cycle containing the whole database ($16406\ \mathrm{km}$) is also considered.

1) Constant state of charge target: In order to evaluate the energy management algorithms performance, a simple simulation case is considered. Using *adaptive-ECMS* and *predictive-ECMS* with singular control or distribution implementation, 41 simulations have been conducted, with the same initial state of charge $x(0) = 50\%$ and different state of charge targets x_{f} in the $0-100\%$ range. Fig. 14 depicts the obtained fuel consumption as a function of the final state of charge. Optimal control results (solution to (16)) are also plotted. The RMS of the fuel consumption deviations with respect to optimal control results are given in Table III. All the different approaches rely on the same Hamiltonian minimization and so they mostly differ by the considered co-state controller. Simple controllers such as (32) do not capture well the effect of the exogenous variable distribution on the costate and poor performances are obtained. The two implementations of the *predictive-ECMS* have similar behavior and hereby confirm that the energy consumptions are not significantly impacted by the quantification of the distribution μ. The *Urban driving cycle* remains quite challenging since driving conditions are changing a lot over time. The two *predictive-ECMS* have a $6-7\%$ fuel consumption increase with respect to the optimal result whereas the *adaptive-ECMS* leads to very poor fuel consumption except for some final state of charge values around 50 %. Over the *highway driving cycle* and the *whole database* both the *predictive-ECMS* implementation provides slightly better fuel consumption than the *adaptive-ECMS*.

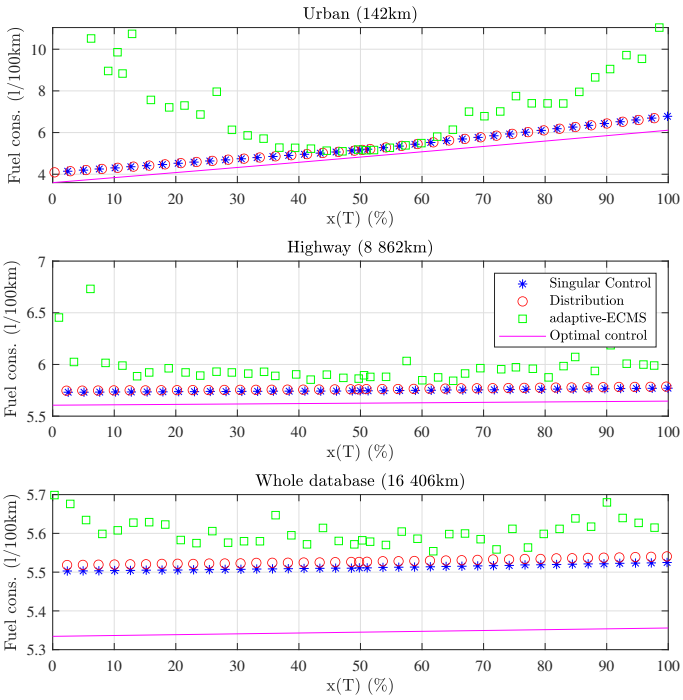

Fig. 14. Fuel consumption of the considered energy management algorithms for constant state of charge targets

2) Plug-in operations: Let us now consider a more realistic case where the vehicle can be recharged every $100km$ once the vehicle is stand-still. These recharges usually take place overnight or during a sufficiently long pause. In order to avoid an unnecessary increase of the driving cycle duration, the battery charging process is not simulated and the state of charge is instead reset to its nominal value (95%). The battery state of charge target x_{f} is set to discharge the battery from 95% down to 20% over $80\ \mathrm{km}$. Once the lower set-point $x_{\mathrm{f}} = 20\%$ is reached, it is kept constant until the next recharge. The obtained results are plotted in Fig.15 and summarized in

TABLE II
COMPUTATION REQUIREMENT OF THE DIFFERENT PREDICTIVE-ECMS IMPLEMENTATION

	adaptive-ECMS	*Baseline*	*Precomputation*	*Singular control*	*Distribution*	*Analytic*
Nb Additions/s	1933	56946	42432	6864	2685	3019
Nb Products/s	3712	77070	46984	9609	4515	8849
Total Operations/s	10967	258354	132322	26135	13305	14190
Computation time (s)	33.3	215.8	164.1	66.4	40.7	21.0
Fuel consumption ($\mathrm{l}/100\mathrm{km}$)	5.88	5.44	5.44	5.44	5.44	6.18
Corr. Fuel cons. ($\mathrm{l}/100\mathrm{km}$)	5.89	5.44	5.44	5.44	5.44	6.18
Δ_x (%)	2.71	0.18	0.18	0.19	0.25	0.44
Memory requirement (Mb)	0.0	0.0	1.6	0.4	2.8	0.0
Final state of charge (%)	49.63	49.99	49.99	50.01	49.86	49.97

TABLE III
RMS OF THE FUEL CONS. DEVIATION WITH RESPECT TO OPTIMAL RESULTS

	Urban	*Highway*	*Whole*	Units
Singular Control	0.44	0.07	0.07	$\mathrm{l}/100\mathrm{km}$
	6.97	1.16	1.26	%
Distribution	0.43	0.08	0.08	$\mathrm{l}/100\mathrm{km}$
	6.75	1.43	1.55	%
adaptive-ECMS	4.60	0.22	0.15	$\mathrm{l}/100\mathrm{km}$
	72.20	3.90	2.78	%

Table IV. This table includes a corrected fuel consumption m'_f that compensate for the final state of charge variation $x'_{\mathrm{T}} - x(T)$ and final network energy variations denoted as $E'_{\mathrm{f}} - E(T)$ with x'_{T} the average final state of charge values and E'_{f} the average final newtwork energy consumption.

$$m'_{\mathrm{f}} = m_{\mathrm{f}} + \lambda \cdot (x'_{\mathrm{T}} - x(t)) + \frac{\lambda \cdot \nu_{\mathrm{avg}}}{OCV \cdot C} \cdot (E'_{\mathrm{T}} - E(T)) \quad (35)$$

with ν_{avg} the average efficiency of the electric path. Compared *adaptive-ECMS*, the two *predictive-ECMS* have 7 to 10 % lower fuel consumptions while simultaneously providing a very good state of charge control. This is one of the major advantage of this control strategy: since it explicitly accounts for the driving distribution, its tuning is quite independent from the driving cycle and a correct state of charge tracking is obtained whatever the driving conditions are. The *adaptive-ECMS* tuning being dependent of the driving conditions, the state of charge tracking is quite poor and inadmissible state of charge values are reached. (-24.1% on the *Highway driving cycle* or -8.3% on the *whole database* driving cycle whereas the setpoint was set to 20% at that time).

VII. CONCLUSION

A *predictive-ECMS* has been presented. It exploits the slow variation of the driving conditions distribution to avoid a complex prediction of the vehicle speed and acceleration. The control strategy has been detailed and its parameters can be easily tuned in simulation using the presented analysis. In order to simplify the tuning process, a sufficiently long driving cycle is used and covers a wide range of the encountered driving conditions. The control strategy using explicitly the driving conditions to compute in real time the costate λ, its tuning on a single driving cycle (representative enough of encountered driving conditions) is sufficient to provide good fuel consumptions and good state of charge tracking on other driving cycles. The computation load of the presented *predictive-ECMS* has been studied and reduced by exploiting the properties of the piecewise Hamiltonian function that leads to singular control. The simulation results show that the computational load of the *predictive-ECMS* with *distribution* implementation can be made only 20% higher than the *adaptive-ECMS* (with an optimized implementation that re-uses results from the Singular Control implementation). This comes at the price of a 2.8 Mb memory usage. The compromise between the fuel consumption, the state of charge tracking performance versus the computational load and the memory usage is set by the choice of either the *distribution* or *singular control* implementation and the control strategy parameters. The energy management performances have been validated using a very long driving cycle (16406 km) constructed using actual measurements (and so representative of one particular vehicle usage). The proposed *predictive-ECMS* lead to a fuel consumption 6.7% lower than the *adaptive-ECMS* one while maintaining tightly the state of charge near the setpoint.

In this work, the presented validation uses only data recorded using a single driver. Similar performances have been obtained using other classical driving cycles such as those from the Hyzem or Artemis study. In general, a control strategy must be robust to different driving styles ('nervous','calm',etc.). Future work will be devoted to the building of a larger database with many more drivers involved. Also, additional developments will be needed to integrate thermal management (such as winter cold start, battery thermal management, coupling with air climate control).

ACKNOWLEDGMENT

This work has been funded by the ELSAT2020 project. It is co-financed by the European Union with the European Regional Development Fund, the French state and the Hauts de France Region Council. The authors gratefully thanks these organization for their suppor

TABLE IV
SUMMARY OF ENERGY MANAGEMENT ALGORITHM PERFORMANCES

	Singular Control	*Distribution*	*adaptive-ECMS*
	WLTC		
m_{f} (l/100km)	3.41 (100.0 %)	3.30 (96.8 %)	3.40 (99.8 %)
Final SOC (%)	78.3	77.1	76.2
m'_{f} (l/100km) with x'_T =77.2 % and E'_T =0 MJ	3.28 (100.0 %)	3.32 (101.0 %)	3.51 (107.1 %)
Δ_x (%)	1.5	1.4	3.7
Min state of charge (%)	78.1	76.8	74.8
Network energy (MJ)	0	0	0
	Urban (142 km)		
m_{f} (l/100km)	2.46 (100.0 %)	2.50 (101.7 %)	2.70 (109.9 %)
Final SOC (%)	62.4	62.5	61.2
m'_{f} (l/100km) with x'_T =62.0 % and E'_T =34 MJ	2.44 (100.0 %)	2.48 (101.7 %)	2.73 (111.9 %)
Δ_x (%)	0.6	0.7	2.3
Min state of charge (%)	19.7	19.8	17.6
Network energy (MJ)	70	70	68
	Highway (8 862 km)		
m_{f} (l/100km)	4.88 (100.0 %)	4.90 (100.4 %)	5.37 (110.0 %)
Final SOC (%)	94.0	94.1	95.2
m'_{f} (l/100km) with x'_T =94.4 % and E'_T =1316 MJ	4.87 (100.0 %)	4.89 (100.4 %)	5.38 (110.4 %)
Δ_x (%)	1.2	1.1	4.4
Min state of charge (%)	18.1	17.8	-24.1
Network energy (MJ)	34	34	34
	Whole database (16 406km)		
m_{f} (l/100km)	4.48 (100.0 %)	4.49 (100.4 %)	4.85 (108.3 %)
Final SOC (%)	19.8	19.8	20.2
m'_{f} (l/100km) with x'_T =20.0 % and E'_T =2572 MJ	4.47 (100.0 %)	4.49 (100.4 %)	4.85 (108.4 %)
Δ_x (%)	1.0	1.1	3.4
Min state of charge (%)	18.5	17.9	-8.3
Network energy (MJ)	1324	1324	1301

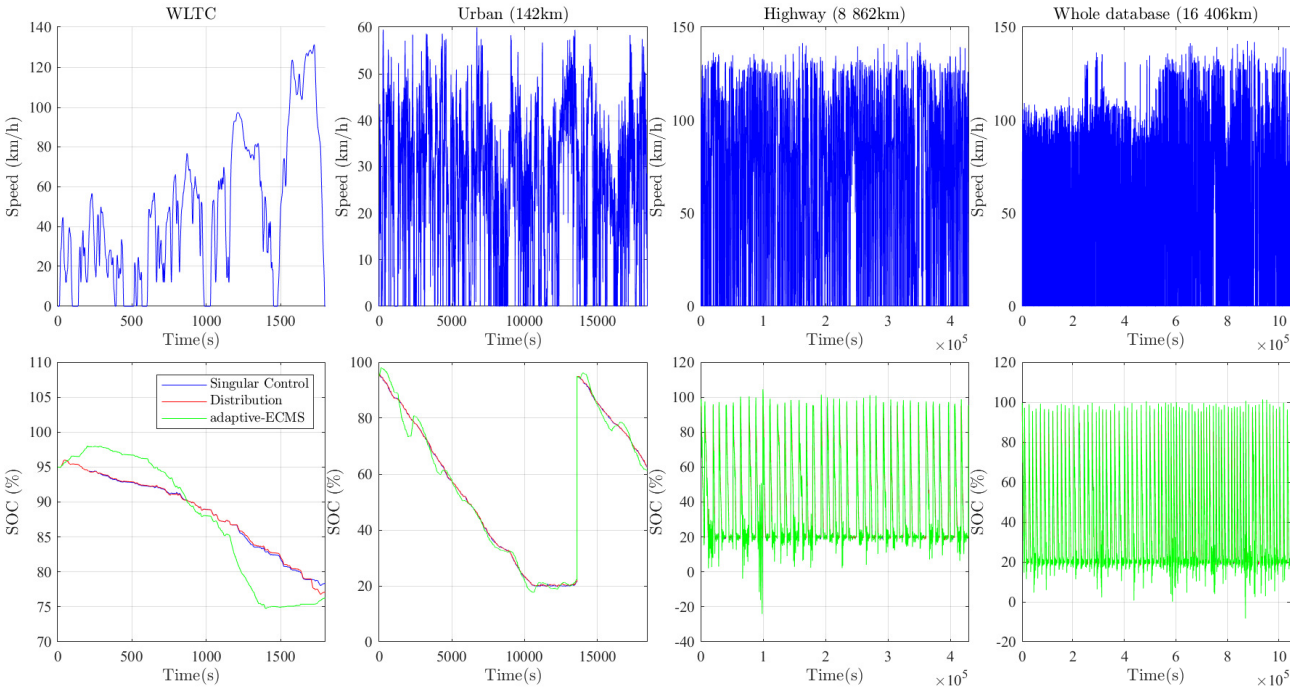

Fig. 15. State of charge profiles over different driving cycles for plug-in operations

REFERENCES

[1] J. Torres, R. Gonzalez, A. Gimenez, and J. Lopez, "Energy management strategy for plug-in hybrid electric vehicles. a comparative study," *Applied Energy*, vol. 113, pp. 816–824, 2014.

[2] R. Wang and S. M. Lukic, "Review of driving conditions prediction and driving style recognition based control algorithms for hybrid electric vehicles," in *2011 IEEE Vehicle Power and Propulsion Conference*, (Chicago, USA), pp. 1–7, IEEE, September 2011.

[3] N. Robuschi, M. Salazar, P. Duhr, F. Braghin, and C. H. Onder, "Minimum-fuel engine on/off control for the energy management of a hybrid electric vehicle via iterative linear programming," *IFAC-*

PapersOnLine, vol. 52, no. 5, pp. 134 – 140, 2019. 9th IFAC Symposium on Advances in Automotive Control AAC 2019.

[4] H. Lee, C. Song, N. Kim, and S. W. Cha, "Comparative analysis of energy management strategies for hev: Dynamic programming and reinforcement learning," *IEEE Access*, vol. 8, pp. 67112–67123, 2020.

[5] M. Montazeri-Gh, A. Poursamad, and B. Ghalichi, "Application of genetic algorithm for optimization of control strategy in parallel hybrid electric vehicles," *Journal of the Franklin Institute*, vol. 343, no. 4-5, pp. 420–435, 2006.

[6] Z. Chen, R. Xiong, and J. Cao, "Particle swarm optimization-based optimal power management of plug-in hybrid electric vehicles considering uncertain driving conditions," *Energy*, vol. 96, pp. 197–208, 2016.

[7] S. Delprat, T. Hofman, and S. Paganelli, "Hybrid vehicle energy management: Singular optimal control," *IEEE Transactions on Vehicular Technology*, vol. 66, no. 11, pp. 9654–9666, 2017.

[8] C. Zheng, W. Li, and Q. Liang, "An energy management strategy of hybrid energy storage systems for electric vehicle applications," *IEEE Transactions on Sustainable Energy*, vol. 9, no. 4, pp. 1880–1888, 2018.

[9] H. S. Ramadan, M. Becherif, and F. Claude, "Energy management improvement of hybrid electric vehicles via combined gps/rule-based methodology," *IEEE Transactions on Automation Science and Engineering*, vol. 14, no. 2, pp. 586–597, 2017.

[10] V. I. Herrera, A. Milo, H. Gaztañaga, A. González-Garrido, H. Camblong, and A. Sierra, "Design and experimental comparison of energy management strategies for hybrid electric buses based on test-bench simulation," *IEEE Transactions on Industry Applications*, vol. 55, no. 3, pp. 3066–3075, 2019.

[11] F. Khoucha, M. Benbouzid, and A. Kheloui, "An optimal fuzzy logic power sharing strategy for parallel hybrid electric vehicles," in *2010 IEEE Vehicle Power and Propulsion Conference*, (Lille, France), pp. 1–5, IEEE, September 2010.

[12] G. Paganelli, S. Delprat, T.-M. Guerra, J. Rimaux, and J.-J. Santin, "Equivalent consumption minimization strategy for parallel hybrid powertrains," in *Vehicular Technology Conference. IEEE 55th Vehicular Technology Conference. VTC Spring 2002 (Cat. No. 02CH37367)*, vol. 4, (Birmingham, AL, USA), pp. 2076–2081, IEEE, 2002.

[13] A. Sciarretta, M. Back, and L. Guzzella, "Optimal control of parallel hybrid electric vehicles," *IEEE Transactions on control systems technology*, vol. 12, no. 3, pp. 352–363, 2004.

[14] C. Musardo, G. Rizzoni, Y. Guezennec, and B. Staccia, "A-ecms: An adaptive algorithm for hybrid electric vehicle energy management," *European Journal of Control*, vol. 11, no. 4, pp. 509 – 524, 2005.

[15] G. Rizzoni and S. Onori, "Energy management of hybrid electric vehicles: 15 years of development at the ohio state university," *Oil & Gas Science and Technology – Rev. IFP Energies nouvelles*, vol. 70, no. 1, pp. 41–54, 2014.

[16] S. Onori, L. Serrao, and G. Rizzoni, "Adaptive equivalent consumption minimization strategy for hybrid electric vehicles," in *Proceedings of the ASME 2010 Dynamic Systems and Control Conference*, pp. 499–505, ASME, 2010.

[17] A. Sciarretta, L. Guzzella, and M. Back, "A real-time optimal control strategy for parallel hybrid vehicles with on-board estimation of the control parameters," *IFAC Proceedings Volumes*, vol. 37, no. 22, pp. 489–494, 2004.

[18] N. Cui, J. Fan, C. Zhang, and J. Wu, "Research on predictive control based energy management strategy for hybrid electric vehicle," in *2012 3rd IEEE International Symposium on Power Electronics for Distributed Generation Systems (PEDG)*, pp. 642–646, IEEE, 2012.

[19] B. Sakhdari and N. L. Azad, "A distributed reference governor approach to ecological cooperative adaptive cruise control," *IEEE Transactions on Intelligent Transportation Systems*, vol. 19, no. 5, pp. 1496–1507, 2018.

[20] Y. Zheng, S. E. Li, K. Li, F. Borrelli, and J. K. Hedrick, "Distributed model predictive control for heterogeneous vehicle platoons under unidirectional topologies," *IEEE Transactions on Control Systems Technology*, vol. 25, no. 3, pp. 899–910, 2017.

[21] C. Zhai, F. Luo, Y. Liu, and Z. Chen, "Ecological cooperative look-ahead control for automated vehicles travelling on freeways with varying slopes," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 2, pp. 1208–1221, 2019.

[22] S. Kermani, S. Delprat, T.-M. Guerra, R. Trigui, and B. Jeanneret, "Predictive energy management for hybrid vehicle," *Control Engineering Practice*, vol. 20, no. 4, pp. 408–420, 2012.

[23] P. Golchoubian and N. L. Azad, "Real-time nonlinear model predictive control of a battery–supercapacitor hybrid energy storage system in electric vehicles," *IEEE Transactions on Vehicular Technology*, vol. 66, no. 11, pp. 9678–9688, 2017.

[24] Y. Huang, A. Khajepour, and H. Wang, "A predictive power management controller for service vehicle anti-idling systems without a priori information," *Applied Energy*, vol. 182, pp. 548–557, 2016.

[25] Y. Liu, J. Liu, D. Chen, and D. Qin, "An exponentially varying speed prediction method based on svm recognition," in *2018 International Conference on Energy, Ecology and Environment*, 2018.

[26] S. Di Cairano, D. Bernardini, A. Bemporad, and I. V. Kolmanovsky, "Stochastic mpc with learning for driver-predictive vehicle control and its application to hev energy management," *IEEE Transactions on Control Systems Technology*, vol. 22, no. 3, pp. 1018–1031, 2013.

[27] Y. Ma, J. Matuško, and F. Borrelli, "Stochastic model predictive control for building hvac systems: Complexity and conservatism," *IEEE Transactions on Control Systems Technology*, vol. 23, no. 1, pp. 101–116, 2014.

[28] C. Sun, F. Sun, and H. He, "Investigating adaptive-ecms with velocity forecast ability for hybrid electric vehicles," *Applied Energy*, vol. 185, pp. 1644 – 1653, 2017. Clean, Efficient and Affordable Energy for a Sustainable Future.

[29] C. Zhai, F. Luo, and Y. Liu, "A novel predictive energy management strategy for electric vehicles based on velocity prediction," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 11, pp. 12559–12569, 2020.

[30] J. Han, D. Kum, and Y. Park, "Synthesis of predictive equivalent consumption minimization strategy for hybrid electric vehicles based on closed-form solution of optimal equivalence factor," *IEEE Transactions on Vehicular Technology*, vol. 66, no. 7, pp. 5604–5616, 2017.

[31] S. East and M. Cannon, "Energy management in plug-in hybrid electric vehicles: Convex optimization algorithms for model predictive control," *IEEE Transactions on Control Systems Technology*, pp. 1–13, 2019.

[32] L. Guzzella and A. Sciarretta, *Vehicle Propulsion Systems: Introduction to Modeling and Optimization*. Berlin:Springer, 2007.

[33] S. Rezazade, M. Changizian, A. Saleki, and H. Moghbeli, "Investigation and comparison between phev and shev for sedan vehicle based on advisor," in *2018 9th Annual Power Electronics, Drives Systems and Technologies Conference (PEDSTC)*, pp. 404–409, 2018.

[34] A. Rezaei, J. B. Burl, and B. Zhou, "Estimation of the ecms equivalent factor bounds for hybrid electric vehicles," *IEEE Transactions on Control Systems Technology*, vol. 26, no. 6, pp. 2198–2205, 2018.

[35] S. Bauer, A. Suchaneck, and F. P. León, "Thermal and energy battery management optimization in electric vehicles using pontryagin's maximum principle," *Journal of Power Sources*, vol. 246, pp. 808–818, 2014.

[36] R. Schmid, J. Bürger, and N. Bajcinca, "A comparison of pmp-based energy management strategies for plug-in-hybrid electric vehicles," *IFAC-PapersOnLine*, vol. 52, no. 5, pp. 592 – 597, 2019. 9th IFAC Symposium on Advances in Automotive Control AAC 2019.

[37] T. van Keulen, J. Gillot, B. de Jager, and M. Steinbuch, "Solution for state constrained optimal control problems applied to power split control for hybrid vehicles," *Automatica*, vol. 50, no. 1, pp. 187 – 192, 2014.

[38] C. Sun, H. He, and F. Sun, "The role of velocity forecasting in adaptive-ecms for hybrid electric vehicles," *Energy Procedia*, vol. 75, pp. 1907 – 1912, 2015. Clean, Efficient and Affordable Energy for a Sustainable Future: The 7th International Conference on Applied Energy (ICAE2015).

[39] X. Hu, X. Zhang, X. Tang, and X. Lin, "Model predictive control of hybrid electric vehicles for fuel economy, emission reductions, and inter-vehicle safety in car-following scenarios," *Energy*, vol. 196, p. 117101, 2020.

[40] H. Borhan, A. Vahidi, A. M. Phillips, M. L. Kuang, I. V. Kolmanovsky, and S. Di Cairano, "Mpc-based energy management of a power-split hybrid electric vehicle," *IEEE Transactions on Control Systems Technology*, vol. 20, no. 3, pp. 593–603, 2012.

[41] K. Han, J. Tao, L. Xie, and R. Zhang, "Rule and mpc based hybrid energy allocation system for hybrid electric vehicle," in *2019 Chinese Automation Congress (CAC)*, pp. 57–62, 2019.

[42] C. Fontaine, S. Delprat, and T. M. Guerra, "Toward analytical solution of optimal control problems for hev energy management," in *2010 IEEE Vehicle Power and Propulsion Conference*, pp. 1–6, 2010.

[43] S. Hadj-Said, G. Colin, A. Ketfi-Cherif, and Y. Chamaillard, "Analytical solution for energy management of parallel hybrid electric vehicles," *IFAC-PapersOnLine*, vol. 50, no. 1, pp. 13872 – 13877, 2017. 20th IFAC World Congress.

[44] M. S. J. Ritzmann, A. Christon and C. Onder, "Fuel-optimal power split and gear selection strategies for a hybrid electric vehicle," in *14th International Conference on Engines and Vehicles*, pp. 1–11, SAE, 2019.

[45] C. Fontaine, S. Delprat, and T. M. Guerra, "Toward analytical solution of optimal control problems for hev energy management," in *2010 IEEE Vehicle Power and Propulsion Conference*, pp. 1–6, 2010.