Preprints of the 21st IFAC World Congress (Virtual)
Berlin, Germany, July 12-17, 2020

Forecasting ECMS for Hybrid Electric Vehicles

Jean Kuchly*, Dominique Nelson-Gruel*, Alain Charlet*, Antoine Simon****,**
Thierry Jaine, Cédric Nouillant**, Yann Chamaillard***

**PRISME lab, University of Orléans, France*
***Groupe PSA, France (e-mail: jean.kuchly@mpsa.com)*

Abstract: This paper aims to propose a real-time suitable method to tackle the problem of energy and pollutant management of Hybrid Electric Vehicles. Methods proposed in the literature often limit the underlying optimal control problem to single-instant optimizations (Paganelli, 2002) due to the difficulty of taking future into account and to onboard limited computational resources. The point of the present paper is to propose an online oriented method based on a long-term vehicle speed prediction, using cartographic information such as speed limitation, road curvature, traffic and road signs. Pontryagin Maximum Principle applied on this speed prediction signal allows to convert the optimal control problem into a root-finding problem. This problem is solved using a Pegasus algorithm initialized by a black-box method trained offline, allowing high computational efficiency. The results are near-optimal and significantly better than classical methods: in the real-driving trip used in this paper, forecasting-ECMS showed a consumption 1.1% better and NO_x emissions 4.4% better than a SOC-feedback adaptive-ECMS.

Keywords: Hybrid Electric Vehicles, Energy Management, Pollutants Management, Real-Time

1. INTRODUCTION

Global warming and air pollution problems led to challenging regulatory consequences for the automotive industry, through for example CAFE objectives in Europe. Hybrid Electric Vehicles (HEVs) are a promising solution proposed by car manufacturers in this context. HEVs have brought a new control layer, related to the relative use of the Electric Motor (EM) and the Internal Combustion Engine (ICE). The control signal is here the torque ratio of a parallel HEV, where both engines are mechanically connected to the road:

$$\mathbf{u(t)} = \frac{\tau_{\mathbf{electric\ motor}}(\mathbf{t})}{\tau_{\mathbf{driver\ request}}(\mathbf{t})}. \qquad (1)$$

Methods proposed in the literature often try to minimize only fuel consumption (Beck, 2007; Hadj-Said, 2016 and 2017). This approach is tempting because fuel consumption can be expressed as a quadratic function of the ICE torque (Nüesch, 2014), leading to a conveniently quadratic, convex optimization problem. Solving it using Quadratic Programming guarantees excellent computational performance. However, this approach may be considered incomplete because fuel consumption and pollutants emissions are antagonist problems, as it is illustrated on Fig. 1 and 2 showing specific fuel consumption and NO_x generation of a gasoline engine of PSA group:

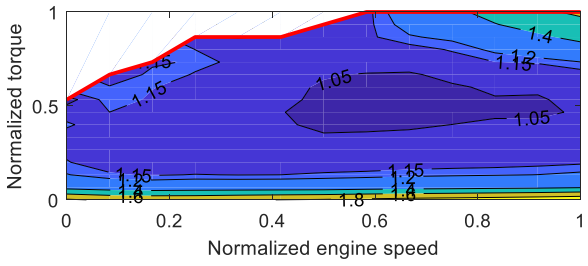

Fig.1: Normalized specific fuel consumption map

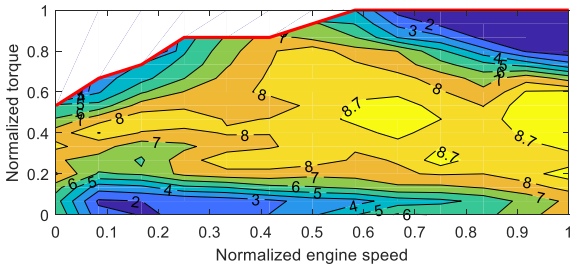

Fig.2: Normalized specific NO_x generation

Since the best region for fuel is nearly the worst for NO_x, minimizing fuel consumption will end up maximizing pollutant generation. We chose consequently to minimize a weighted compromise of NO_x and fuel consumption (Michel, 2012 and 2014a; Simon, 2015):

$$\boldsymbol{J} = \int_0^T (\dot{\boldsymbol{m}}_{\boldsymbol{fuel}}(\boldsymbol{t}) + \boldsymbol{\alpha} \cdot \dot{\boldsymbol{m}}_{\boldsymbol{NOx}}(\boldsymbol{t})) \cdot \boldsymbol{dt} \qquad (2)$$

Numerous methods are able to handle this problem of HEV supervision, as shown in the survey paper of Martínez (2016). A vast majority of the proposed methods for real-time application does not consider solving explicitly (2) due to computational time restriction onboard and to the difficulty to deal with future, therefore degrading the optimality of the solution. The point of this paper is to propose an online-meant solving of (2) expected to give near-optimal performances thanks to the prediction of the long-term future vehicle speed, and using real-time suitable methods.

The most important state in the system is the battery State of Charge (SOC). Other states can be taken into account, particularly the catalyst converter temperature (Michel, 2014b), in order to model its conversion efficiency. The catalyst converter is not taken into account in this work, but will be the focus of further studies.

There are three main ways to solve explicitly an optimal control problem of this nature:

- Dynamic Programming (DP; Bellman, 1954), that explores all the possible state values. Its important computational resources and memory needs restrain it to offline applications, but it is commonly used to evaluate the potential of a system architecture or to provide a reference to be compared with actual online strategies. In this paper, the algorithm presented in Simon (2018) provides a benchmark.

- Direct methods, where the minimization criterion is expressed as a cost function depending on the successive values of the time-discretized control signal. A numerical optimization algorithm reaches the minimum of the cost function and gives the optimal control values. The drawback of this approach is the large dimension that the optimization problem may attain if the sampling period of the control signal is short compared to the system dynamics. This is typically the case of the HEV torque-split signal and its battery SOC, and particularly in the large-battery case of a Plug-In HEV (PHEV), such as the vehicle taken as example in this paper. Many direct approaches handling the HEV supervision problem have already been tried (Kuchly, 2019; Di Cairano, 2013).

- Indirect methods, based on the Pontryagin Maximum Principle (PMP; Pontryagin, 1962). As developed in the next section, the PMP allows to sum up the whole optimal control of the HEV torque split to the determination of a single scalar constant λ. This strong advantage led Paganelli (2002) to propose the current reference method for the HEV energy management: the Equivalent Consumption Minimization Strategy (ECMS), which is simply a realization of the PMP (Serrao, 2009). The method presented in this paper is a pollutant-including ECMS, an Equivalent Consumption and Pollutants Minimization Strategy (ECPMS; Simon, 2015). ECMS and ECPMS methods have proven very good performances in a real-time context, but have to face two major difficulties: on the one hand, the little computational time available discourages the use of too demanding optimization processes. On the other hand, it is necessary to know the future vehicle speed to solve the optimal control problem, and most methods avoid facing this problem by adapting continuously to the current situation only. The main contributions of this paper consist in two parts: first an algorithm intends to deal with the future by predicting roughly the vehicle speed using cartographic information (see section 3), allowing to obtain near-optimal control performances. Then the use of an efficient numerical process described in section 4 allows competitive computational performances.

2. ECMS/ECPMS

2.1 Principle

A Hamiltonian is defined as the minimization criterion, taken only at the current time step, and augmented by the costate λ, multiplied by the state variation $\dot{SOC}$:

$$\boldsymbol{H(u(t),t) = \dot{m}_{fuel}(t) + \alpha \cdot \dot{m}_{NOx}(t) + \lambda(t) \cdot \dot{SOC}(t)} \quad (3)$$

It follows the general formulation of the PMP, but it has also a physical meaning: it is a compromise between a thermic cost $\dot{m}_{fuel}(t) + \alpha \cdot \dot{m}_{NOx}(t)$ and an electric cost $\lambda \cdot \dot{SOC}(t)$. The co-state λ is a weighting factor, an *equivalence factor* between the prices of both energies. Thus, λ will be the key of the problem and its determination is the point of the various existing ECMS/ECPMS methods. The current optimal control value is obtained by minimizing the Hamiltonian:

$$\boldsymbol{u^*(t) = \underset{u}{argmin}\ (H(u(t),t))}. \quad (4)$$

The basic principle of ECMS is to minimize H at each real time step. To be able to evaluate H, it is necessary to know the value of λ. Its variation is given by the following canonical equation of Hamilton:

$$\boldsymbol{\dot{\lambda}(t) = -\frac{\partial H(u(t),t)}{\partial SOC(t)}} \quad (5)$$

The first term of H, including instantaneous fuel consumption and NOx generation, is independent of the current SOC value. The second term $\lambda \cdot \dot{SOC}$ is dependent on the open-circuit voltage and the internal resistance of the battery, that both depend on the SOC value. However the dependency is sufficiently small to be neglected, and $\frac{\partial H(u(t),t)}{\partial SOC(t)} \approx 0$ is an often-made assumption (Michel, 2014). λ is hence considered constant.

2.2 How to compute λ ?

Say the vehicle speed and consequently the powertrain successive operating points (engine speed and torque) are known over the upcoming trip. One can arbitrarily choose a value for λ and simulate the trip: at each time step, the optimal control value $u^*(t)$ is computed by minimizing the current Hamiltonian (4). The SOC value is updated consequently by applying the control to the powertrain model, and the process is done successively until the end of the simulated trip. It is then possible to compute the final SOC value for a given value of λ. This SOC evolution simulation will allow to compute a λ leading to a desired SOC value.

The final SOC is indeed constrained: in the charge-sustaining case of a non-plug-in HEV, it should end up around an average value, for example 50%. In the charge-depleting case of a PHEV, considering a recharge at the end of the trip, the final SOC should be the lowest acceptable value plus a safety margin in order to ensure all the energy available in the battery has been used. A function $r(\lambda)$ is defined (6) and displayed in Fig.3, representing the difference between the final desired SOC percentage and the SOC value actually obtained for a given value of λ:

$$\boldsymbol{r(\lambda) = SOC_{final}(\lambda) - SOC_{desired}} \quad (6)$$

Extreme values of λ will value too much the price of thermic or electric energies and drive the SOC to respectively the lowest or highest values achievable during the trip, leading to a sigmoid shape for $r(\lambda)$. r admits one root: it is the balanced value of λ allowing to reach the desired SOC value at the end of the trip. It is possible to find λ and then solve the optimal control problem thanks to a root-finding algorithm.

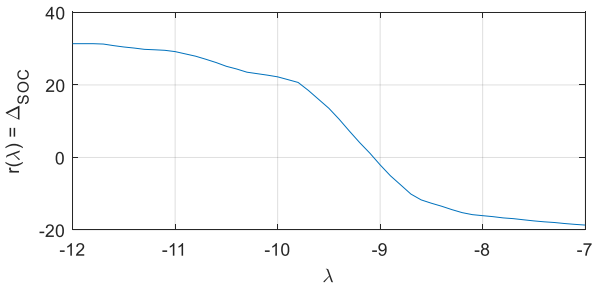

Fig.3: Shape of $r(\lambda)$

This approach solves explicitly the PMP problem, but has two drawbacks. First, it is necessary to know or estimate a priori the vehicle speed over the considered trip. Moreover, solving the problem may require prohibitive computational time, regarding the real-time requirement and the low embedded computational resources in the automotive context. Hence, most methods do not rely directly on the PMP to compute λ, but try instead to adapt its value depending on the current SOC or other information. Three main families of methods exist:

- SOC-feedback methods, which update λ in order to regulate the SOC around a certain value (Onori, 2010). It relies only on the SOC value and rarely on cartographic information.

- Pattern recognition methods, where past information are used in order to deduct the driving conditions and choose accordingly a relevant value for λ (Gu, 2006).

- Vehicle speed prediction methods, like in Kazemi (2017), where a short-term 1-minute prediction is used. The present paper aims to estimate the future vehicle speed over a large 20-minutes horizon based on cartographic information, in order to solve explicitly the aforementioned PMP problem using a root-finding algorithm. By rebuilding an a priori knowledge of the future, it is hoped to approach closely optimal performances. The proposed method will be compared to both Dynamic Programming and a SOC-feedback method.

3. MODELS

3.1 Vehicle

A PHEV model is obtained from the model structure presented in Simon (2018). The battery is modelled using only an ideal generator and an internal resistance (Guzzela, 2007). Both engines are modelled using maps of engine speed and torque: efficiency for the EM, and fuel consumption and NO_x generation for the ICE. The present study does not consider yet other pollutants, nor the catalyst converter. The catalyst converter will be the focus of further studies (see section 6).

3.2 Driver

To predict the vehicle speed, it is necessary to model the driver behaviour in response to external elements: road curvature, traffic, speed limitations,… Road slopes are not yet taken into account. A speed reference is set to the current speed limitation. This reference will then be lowered by limiting factors. First, the maximum comfortable curve speed at a position x depends on the curve radius r_c (Polus, 2000):

$$\boldsymbol{V_{\max curve}(x) = K - \frac{1}{a \cdot r_c(x) + b}}, \qquad (7)$$

The black-box parameters K, a, and b are calibrated using real-driving data.

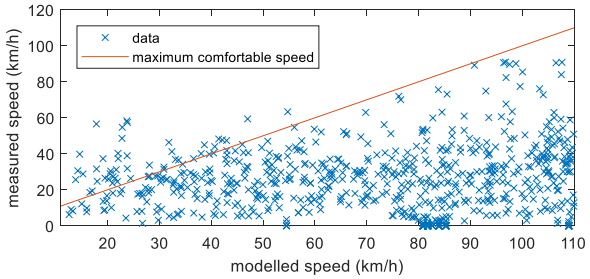

Fig.4: Maximum speed in curve against real data

The modelled maximum speed is plotted against real driving data in Fig. 4. (7) aims to provide a superior bound imposed to the vehicle speed by the curve radius, and this bound will be lowered by other factors such as local traffic. Therefore, the points obtained in real-driving (blue in Fig. 4) should be inferior to the output of (7) (red in Fig. 4). It is the case, except for high-speed points measured in low-radius cases, that were voluntarily excluded of the calibration because they correspond to hardly-predictable behaviour of drivers in urban context, caused by the surrounding traffic. The new speed reference is then chosen by saturating the current speed limit L by the maximum comfortable curve speed obtained:

$$\boldsymbol{V_{ref}(x) = \min\left(V_{max\,curve}(x), L(x)\right)} \qquad (8)$$

The local traffic level I_{tr} will be graded from 1 (fluid) to 4 (blocked), levels 2 and 3 being intermediate levels in-between these extremes. The vehicle speed imposed by the traffic $V_{ref\,tr}$ is obtained this way:

$$\boldsymbol{V_{ref\,tr}(x) = V_{ref}(x) - K\left(L(x), I_{tr}(x)\right)R(x) - A\left(L(x), I_{tr}(x)\right)}, \qquad (9)$$

where K and A are parameters whose value is selected accordingly to the current speed limit and local traffic level. The different selectable values are calibrated using real-driving data. $R(x)$ is a random, space-filtered value between 0 and 1 attributed to each position of the longitudinally-discretized vehicle path, once at the beginning of the trip. This value allows to take into account the stochastic impact of traffic on speed, without modelling other vehicles.

Finally, at each time step the model compute the required distance to stop. If a road element forcing the vehicle to stop (traffic light, stop sign) is in range, the speed reference is put to 0. The vehicle will hence progressively slow down to stop at the required position.

The driver is modelled by a proportional regulator acting directly on the speed value, and having a coefficient P_A in accelerations and P_B in decelerations. The maximum speed variation is saturated by two constants: the driver maximum comfortable acceleration amx and its equivalent in braking bmx. By applying the whole process it is possible to obtain a very rough speed prediction. Fig. 5 show both real and predicted vehicle speeds for a 65 km trip between Plaisir and

Bussy Saint-Georges (France), including both highways and dense, urban driving.

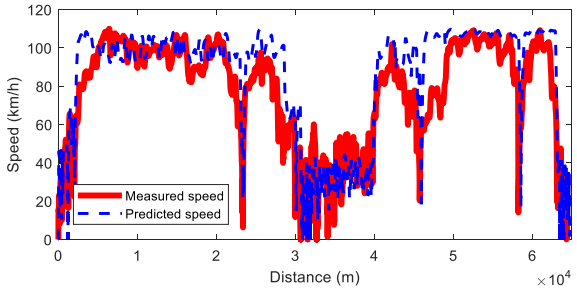

Fig.5: Modelled speed against real data

The speed prediction is locally false but it is representative of the energy spent during the trip. There is indeed a difference of 10.5% only in the mechanical energy produced by the powertrain between prediction of real trip. The dynamics are close enough to reality to allow good optimization performance (see section 5). If necessary, driver dynamic can be improved by recursively updating the parameters of the model to adapt online to the actual driver behaviour.

4. REAL-TIME OPTIMAL CONTROL

The method proposed here considers a finite, receding horizon, over which the vehicle speed is predicted and the PMP solved. The method presented in section 2 to handle the PMP needs to solve two numerical problems. It must be able to minimize the successive Hamiltonians (4) and to find the root of $r(\lambda)$ (6).

4.1 Hamiltonian minimization: gradient algorithm

For given values of λ, engine speed and torque, the Hamiltonian depends only on the current value of u. A single dimension optimization problem has then to be solved, and a gradient algorithm using a polynomial interpolation line search is used (Wright, 1999). The Hamiltonian may present a local minimum at $u = 0$; in order to handle this, the initial guess is chosen as follows:

$$\begin{cases} if\ H\left(u = \frac{u_{max}}{2}\right) < H(0): u = \frac{u_{max}}{2} \\ else: u = argmin(H(0), H\left(\frac{u_{min}}{2}\right)) \end{cases} \quad (10)$$

4.2 Root-finding: Pegasus algorithm

$r(\lambda)$ admits only one root (Fig. 3). In order to find it, a Pegasus algorithm is implemented (King, 1973). As the dichotomy method for example, it is looking for the root between two bounds λ_a and λ_b, $r(\lambda_a)$ and $r(\lambda_b)$ having opposite signs. The new candidate λ_c is chosen at the root of a linear approximation of r between $r(\lambda_a)$ and $r(\lambda_b)$. If $r(\lambda_c)$ is not close enough to 0, it replaces the same-sign bound and a new step is taken. If successive steps bring same-sign candidates, the value of the opposite bound (say $r(\lambda_b)$) is artificially reduced:

$$\boldsymbol{r'(\lambda_b)} = \frac{\boldsymbol{r(\lambda_b) \cdot r(\lambda_a)}}{\boldsymbol{r(\lambda_a) + r(\lambda_c)}}. \quad (11)$$

This update allows to deal with solution retention problems. Considering the general shape of $r(\lambda)$ in Fig. 3, this situation may occur if one bound is close or in the linear part of $r(\lambda)$, and the other far away in the flat part of $r(\lambda)$. Pegasus algorithm appears as a relevant method in this context to solve the root-finding problem.

4.3 Pegasus initial bounds

Starting Pegasus with bounds close to the root allows to gain considerable computational time. Considering a horizon length large compared to the update period n, only small difference should occur between two successive speed predictions. Then, the previous λ computed at the last update is already a good approximation of the current solution.

How to obtain a second bound ? The ideal would be to obtain a good approximation of the solution, computable fast enough to get quickly Pegasus started. This estimation of λ allows also to provide a guess at the initial instant, and to adapt to an important change detected in the trip prediction.

A black-box method is used. N artificial roads are randomly generated offline, defined by randomized but realistic curves, speed limitations, local traffic, etc… Variable initial vehicle speed and initial SOC values are assigned to each road. The driver model is used to obtain the vehicle speed corresponding to each road. An input vector including the length of the trip, the mean speed, the initial speed and ΔSOC the variation between the initial and the desired final SOC is associated to each road. The optimal control problem is solved over the whole trip for each road, that can then be assigned a value of λ. A black-box method can now be used in order to link the input vectors to the corresponding values of λ.

Numerous methods could have been used to estimate λ, but this paper does not aim to provide a detailed discussion about the choice of an optimal method. We chose to use the Neural Network Fitting Matlab Toolbox as a first approach, because it was easy and fast to use, and because the generated function gave decent accuracy and computational time.

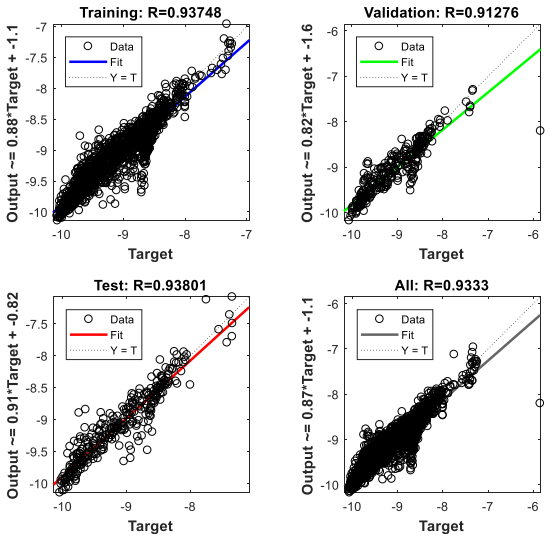

Fig.6: Neural network fitting performance

The result of the training process are exposed in Fig. 6: the target data are the solutions λ of the optimal control problems corresponding to $N = 1000$ randomly generated routes, and the output is the estimation provided by the neural network.

Validation data are used to ensure the network is generalizing and to stop training before overfitting, and test data are used to provide a final estimation of the neural network accuracy. Here the correlation between test data and estimation is above 90%, allowing to solve Pegasus algorithm in one step in a vast majority of cases and providing good results as presented in section 5. We did not investigate further the question of the estimation method, considering that the imperfection of the estimation is merely due to the incompleteness of the input variables. To improve the correlation, parameters representing the driving aggressiveness should be taken into account, such as the mean absolute vehicle acceleration for example.

If the two bounds have the same sign, a straight line is extrapolated from the two points. A new λ value is taken at the root of the line, and the function r is evaluated at this point. The curvature of the function (Fig. 3) ensures that the new point is of opposite sign, or at least closer to the root of r. Checks are performed: if the two bounds values are positive, the step must be done toward $\lambda = 0$, an eventual positive λ is pointless so it is brought back to zero, etc…

4.4 SOC reference for PHEV

Energy available in the battery has to be allocated over all the trip (Martinez, 2016). The final SOC desired at the end of the horizon is then linearly approximated this way:

$$\boldsymbol{SOC_{desired}} = \frac{(\boldsymbol{d_r - d_h})\cdot(\boldsymbol{SOC_0 - SOC_{final}})}{\boldsymbol{d_r}} + \boldsymbol{SOC_{final}}, \quad (12)$$

where d_r is the remaining distance of the trip, d_h is the predicted distance completed at the end of the horizon, SOC_0 is the current SOC value, and SOC_{final} is the desired SOC value at the end of the trip.

At the beginning of the trip the tolerance used for Pegasus algorithm is set to 4.5% of SOC. After the range of the prediction becomes larger than the remaining part of the trip, accuracy is linearly decreasing to 0.5%. The idea is to quickly obtain a suitable λ in the beginning of the trip, and to refine progressively the result in the end of the trip. The upper limit of 4.5% is chosen so that it can be satisfied most of the time by the neural network of section 4.3, allowing good computational performance at the initial instant.

The simplest way to improve performance is to ensure all the electric energy initially available has been used at the end of the trip. Hence, EM use is forced if the SOC is still above a user-defined threshold for a too short remaining distance.

4.5 Real-time strategy

The general algorithm functions as follows:

Algorithm 1: Forecasting-ECPMS (F-ECPMS)

Predictive phase: Every n seconds

Vehicle speed is predicted over a finite horizon

The desired SOC at the end of the horizon is computed

Pegasus algorithm is used to find the root of $r(\lambda)$ on the predicted horizon

ECPMS phase: Every second or less

The current Hamiltonian is minimized to obtain the optimal control value $u^*(t)$, using the value λ computed through root finding

To sum up, the algorithm is functioning as a regular ECPMS, but the value of λ is regularly updated accordingly to the speed prediction. The value of the time period for the update of λ can be tuned accordingly to the embedded computational resources available. In the simulation presented in section 5 the time period for the torque split computation is set to 1 second, but it can be reduced in order to ensure that the powertrain is always able to provide the necessary torque, particularly in the case of a sudden and important variation in the torque request of the driver. Finally the consumption-pollutant compromise α is a parameter left to be set by the car manufacturers pollutant specialists in order to respect the considered standard.

5. RESULTS

In order to assess the performances of the proposed method, a simulation is performed. The system considered is a Diesel PHEV and the trip is the one presented in Fig. 5, featuring both urban and highway driving. It is a 65 km trip, lasting 4100 seconds. The initial SOC is 90%, and the final desired value is 20%. On the first hand, a Dynamic Programming is applied offline, considering an exact and total knowledge of the future vehicle speed and operating points. This algorithm is discretized as follows: $\Delta u = 0.005$, and $\Delta SOC = 0.005\%$. It provides results very close to the best performance achievable for this trip, that are used as a reference. On the other hand, the F-ECPMS algorithm described in this paper is used. It uses a 20 minutes horizon length, and a 100 seconds time period between two λ updates. The current control signal is computed every second by minimizing the Hamiltonian.

For comparison purpose, an Adaptive-ECPMS (A-ECPMS; Onori, 2011) is implemented using a SOC-feedback with a PI controller and a reference linearly decreasing with the remaining distance. A basic Charge Depleting – Charge Sustaining (CD-CS) method is also implemented: pure electric driving is forced until a low SOC threshold is attained. Then, the SOC is regulated around this threshold until the end of the trip, using A-ECPMS.

	DP	F-ECPMS	A-ECPMS	CD-CS
m_{fuel}	2072.6 g	-0.1%	+1%	+3.4%
m_{NOx}	25.49 g	+0.2%	+4.6%	+10%

Table 1: Compared performances

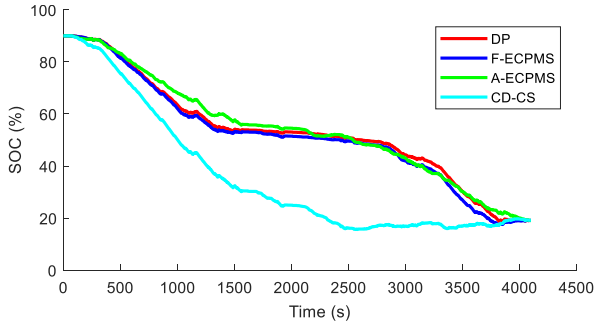

Fig.7: SOC signals

Fig.7 shows that DP, F-ECPMS and in a less remarkable way A-ECPMS generate similar SOC signals. F-ECPMS provides consumption and NO_x performances really close to the best result achievable provided by DP (Tab. 1), and performs better than classical methods such as Adaptive ECPMS or CD-CS.

Fig. 8 displays the computational time needed to predict the vehicle speed and solve the PMP over the 20-minute horizon. The computational time of the predictive phase is never above 20 ms, and since it is done every 100 seconds it can be spread over a large period of time in the vehicle control unit. Besides, the ECPMS phase at each instant of the trip requires on average 25 µs. This computational performance has been obtained on a computer and using Matlab, but the order of magnitude of the computational time makes the proposed method relevant for a real-time context in a vehicle. It can also be compared to the order of magnitude of the 200 seconds needed in Michel (2015) to solve the PMP over a 30-minute WLTP cycle, or to the 13 minutes needed by the DP algorithm used in the present paper to optimize a 20-minute cycle.

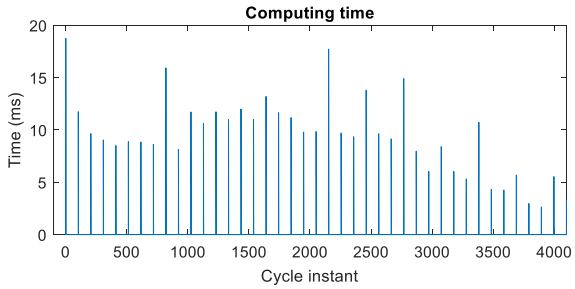

Fig.8: Computational time

6. CONCLUSION

F-ECPMS is quite demanding: it needs cartographic information over a large range. The system should also be able to predict the trip destination, eventually based on the driver habits for daily trips, or on big data information. However, it is able to provide near-optimal performances and is computationally efficient. Hence, F-ECPMS can be considered a relevant competitor in the context of energy and pollutants management of HEVs.

An ECPMS strategy has to take into account the catalyst converter efficiency to be truly relevant. Oncoming work will then aim to include the conversion efficiency of the catalyst converter in the system, thanks to the estimation of its temperature (Simon, 2019). Two solutions may be considered: either the conversion efficiency is estimated from temperature and simply taken into account in the Hamiltonian used in this work, or the catalyst converter temperature could be added in the application of the PMP as a second state, leading to a second co-state λ_{CAT}. The root-finding problem would then be solved by searching the minimum of $r(\lambda, \lambda_{CAT})^2$.

Other important objectives are to provide a better SOC reference taking into account road information, and to implement F-ECPMS in a vehicle in order to prove its real-time ability. Further work will also aim to propose a classical A-ECPMS method taking into account cartographic information about the future through its SOC reference.

REFERENCES

Beck, R., Bollig, A., & Abel, D. (2007). Comparison of two real-time predictive strategies for the optimal energy management of a hybrid electric vehicle. *Oil & Gas Science and Technology-Revue de l'IFP*, *62*(4), 635-643.

Bellman, R. (1954). *The theory of dynamic programming* (No. RAND-P-550). RAND Corp Santa Monica CA.

Di Cairano, S., Bernardini, D., Bemporad, A., & Kolmanovsky, I. V. (2013). Stochastic MPC with learning for driver-predictive vehicle control and its application to HEV energy management. IEEE Transactions on Control Systems Technology, 22(3), 1018-1031.

Gu, B., & Rizzoni, G. (2006, January). An adaptive algorithm for hybrid electric vehicle energy management based on driving pattern recognition. In ASME 2006 International Mechanical Engineering Congress and Exposition (pp. 249-258). American Society of Mechanical Engineers.

Guzella, L., Sciaretta, A., et al. (2007). *Vehicle propulsion systems*, volume 1. Springer.

Hadj-Said, S., Colin, G., Ketfi-Cherif, A., & Chamaillard, Y. (2016). Convex optimization for energy management of parallel hybrid electric vehicles. *IFAC-PapersOnLine, 49*(11), 271-276.

Hadj-Said, S., Colin, G., Ketfi-Cherif, A., & Chamaillard, Y. (2017). Analytical solution for energy management of parallel hybrid electric vehicles. *IFAC-PapersOnLine, 50*(1), 13872-13877.

Kazemi, H., Fallah, Y. P., Nix, A., & Wayne, S. (2017). Predictive AECMS by utilization of intelligent transportation systems for hybrid electric vehicle powertrain control. IEEE Transactions on Intelligent Vehicles, 2(2), 75-84.

King, R. F. (1973). An improved Pegasus method for root finding. *BIT Numerical Mathematics*, *13*(4), 423-427.

Kuchly, J., Nelson-Gruel, D., Charlet, A., Chamaillard, Y., Nouillant, C. (2019). Projected Gradient and Model Predictive Control : Optimal Energy and Pollutants Management for Hybrid Electric Vehicles. In *Advances in Automotive Control (AAC), 2019*

Marquardt, D. W. (1963). An algorithm for least-squares estimation of nonlinear parameters. *Journal of the society for Industrial and Applied Mathematics*, *11*(2), 431-441.

Martinez, C. M., Hu, X., Cao, D., Velenis, E., Gao, B., & Wellers, M. (2016). Energy management in plug-in hybrid electric vehicles: Recent progress and a connected vehicles perspective. IEEE Transactions on Vehicular Technology, 66(6), 4534-4549.

Michel, P., Charlet, A., Colin, G., Chamaillard, Y., Nouillant, C., & Bloch, G. (2012). Energy management of HEV to optimize fuel consumption and pollutant emissions. In *11th International Symposium on Advanced Vehicule Control (AVEC), 2012*.

Michel, P., Charlet, A., Colin, G., Chamaillard, Y., Nouillant, C., & Bloch, G. (2014). 3WCC temperature integration in a gasoline-hev optimal energy management strategy. *Advances in Mechanical Engineering*, *6*, 802597.

Michel, P., Charlet, A., Colin, G., Chamaillard, Y., Bloch, G., & Nouillant, C. (2014). Catalytic converter modeling for optimal gasoline-HEV energy management. *IFAC Proceedings Volumes, 47*(3), 6636-6641.

Nüesch, T., Elbert, P., Flankl, M., Onder, C., & Guzzella, L. (2014). Convex optimization for the energy management of hybrid electric vehicles considering engine start and gearshift costs. Energies, 7(2), 834-856.

Onori, S., Serrao, L., & Rizzoni, G. (2010, January). Adaptive equivalent consumption minimization strategy for hybrid electric vehicles. In ASME 2010 dynamic systems and control conference (pp. 499-505). American Society of Mechanical Engineers.

Onori, S., & Serrao, L. (2011, December). On Adaptive-ECMS strategies for hybrid electric vehicles. In Proceedings of the international scientific conference on hybrid and electric vehicles, Malmaison, France (Vol. 67).

Paganelli, G., Delprat, S., Guerra, T. M., Rimaux, J., & Santin, J. J. (2002). Equivalent consumption minimization strategy for parallel hybrid powertrains. In *Vehicular Technology Conference, 2002. VTC Spring 2002. IEEE 55th* (Vol. 4, pp. 2076-2081). IEEE.

Polus, A., Fitzpatrick, K., & Fambro, D. B. (2000). Predicting operating speeds on tangent sections of two-lane rural highways. Transportation Research Record, 1737(1), 50-57.

Pontryagin, L., Boltânskij, V., Gamkrelidze, R., Misenko, E. (1962). *The Mathematical Theory of Optimal Processes*. Interscience, 1962.

Serrao, L., Onori, S., & Rizzoni, G. (2009, June). ECMS as a realization of Pontryagin's minimum principle for HEV control. *In 2009 American control conference* (pp. 3964-3969). IEEE.

Simon, A., Michel, P., Nelson-Gruel, D., Chamaillard, Y., & Nouillant, C. (2015, October). Gasoline-HEV equivalent consumption and pollutant minimization strategy. In *Vehicle Power and Propulsion Conference (VPPC), 2015 IEEE* (pp. 1-6). IEEE.

Simon, A., Nelson-Gruel, D., Charlet, A., Jaine, T., Nouillant, C., Chamaillard, Y. (2018). Optimal supervisory control of a Diesel HEV taking into account both DOC and SCR efficiencies. *IFA-PapersOnLine*, 51(9), 323-328.

Simon, A., Nelson-Gruel, D., Onori, S., Charlet, A., Jaine, T., Collin, G., Nouillant, C., Chamaillard, Y. (2019). An Observer Looks at the Temperature in the 3WCC. In *Advances in Automotive Control (AAC), 2019*

Wright, S., & Nocedal, J. (1999). *Numerical optimization*. Springer Science, 35(67-68), 7.