Preprints of the 20th World Congress
The International Federation of Automatic Control
Toulouse, France, July 9-14, 2017

Handling State Constraints in Fast-computing Optimal Control for Hybrid Powertrains

Jihun HAN*, Antonio SCIARRETTA*, Nicolas PETIT**

** IFP Energies Nouvelles, 92852 Rueil Malmaison, France*
(e-mail: {jihun.han, antonio.sciarretta}@ifpen.fr),
***Centre Automatique et Systèmes, Unité Mathématiques et Systèmes, MINES ParisTech, France*
(e-mail: nicolas.petit@mines-paristech.fr).

Abstract: To optimally design hybrid powertrains, optimal energy management strategies must be automatically and rapidly generated. Pontryagin's minimum principle-derived optimization tool called Hybrid Optimization Tool (HOT) can guarantee the fast computing of minimal fuel consumption using an array operation as well as Picard's method. However, in presence of state constraints (e.g., the battery state of charge limitations), the near-optimality of HOT no longer holds. Herein, we use the interior- and exterior-penalty method to impose the state constraints in HOT and highlight numerical difficulties encountered in their implementation. Then, a factor that causes the numerical difficulties is optimized by quantifying trade-off between the state constraints violation and computational demanding. Finally, through a case study of a parallel hybrid electric vehicle, the results show that despite of a complex problem with rapidly changing dynamics, the penalty methods are able to generate results comparable with dynamic programming ones while guaranteeing the low computational burden.

Keywords: hybrid vehicles, energy management, optimal control, state constraints, simulation.

1. INTRODUCTION

Due to the increasing complexity of hybrid powertrains, including various topologies and architectures, and many dimensional parameters, Hybrid electric vehicles (HEV) design is not a trivial task. A design optimization based on mathematical modelling is therefore needed to achieve the best performance (e.g. fuel economy, emissions, etc.) (Pourabdollah, 2013; Jianning and Sciarretta, 2016). Furthermore, the performance heavily depends on a supervisory control, known as energy management strategy (EMS), because the EMS determines the power split ratio between multiple power sources (Sciarretta et al., 2007). Therefore, in order to optimize the design variables and topology, it is necessary to generate a near-optimal EMS for each configuration, which allows computing its optimal performance. In addition, the design optimization process should guarantee a reasonably fast computing time in order to be effectively adopted in industrial practice.

A method for fast computing of minimal fuel consumption on a given driving scenario has been presented in (Chasse et al., 2011), called a hybrid optimization tool (HOT). The HOT uses a generic hybrid powertrain architecture that can be parameterized to represent various configurations. The HOT is based on Pontyagin's minimum principle (PMP) (Serrao et al., 2011), and finds iteratively the initial value of co-states by a shooting method. Typically, the battery state of charge (SoC) is a single state variable, and the physical meaning of corresponding co-state is a fuel/electricity equivalence factor. Furthermore, it is possible to extend to multiple-state optimization or to multiple objectives, as in (Serrao et al., 2013; Michel et al., 2015). The HOT as a fast PMP-based solution does not use multiple for-loop algorithm (time, control variable, initial values of co-states), but uses an array operation for reducing the computational time within the acceptance level of memory use. Moreover, HOT uses Picard's method for considering state-dependent system dynamics (Sciarretta et al., 2015).

However, when the state constraints become active, the near-optimal performance of HOT becomes questionable due to a non-trivial co-state dynamics (Kim et al., 2011). In PMP framework, the state constraints can be handled in two ways. First, an analytical approach defines Lagrangian, and solves the constrained optimal control problem (OCP) using Lagrangian-based necessary conditions for optimality (Hartl et al., 1995). This approach would require a model that is sufficiently simple to be solved analytically (Pérez et al., 2016), leading to a lack of accuracy because of the unconsidered nonlinear properties. Secondly, a numerical approach transforms the constrained OCP into an unconstrained OCP, and then solves it. In (Van Keulen et al., 2014), the constrained OCP is reformulated as a sequence of unconstrained sub-problems and is recursively solved, but it is only applicable to scalar state-constrained OCP.

Alternatively, the state constraints can be replaced by introducing a penalty function that penalizes the state constraints violation in the cost function (Xing et al., 1989; Malisani et al., 2014). Solving the penalized OCP is an effective technique because it does not require the knowledge of the sequence of constrained (or unconstrained) arcs. This motivates the use of the penalty method in combination with the fast computing, PMP-based HOT. However, the penalty method causes numerical difficulties as the penalty parameter goes to extremely low/high values. Thus, this paper is aimed to discuss how to implement the penalty function into the HOT.

The remainder of this paper is organized as follows: Section 2 gives a brief overview of optimal EMS and introduces the penalty method. In Section 3, basic features of HOT are addressed. Section 4 presents the implementation issues and Section 5 shows results of a parallel HEV as a case study. Finally, in Section 6, conclusions are presented.

2. OPTIMAL CONTROL PROBLEM FOR HEV

A generic state-constrained OCP can be written as follows:

$$\min J = \int_0^T L(u,x,\tau)\,d\tau \tag{1}$$
$$\text{s.t. } \dot{x} = f(u,x,t), \tag{2}$$
$$x \in X, \quad u \in \Omega \tag{3}$$

where L is the cost function, x and u represent state variable and control input, respectively; X and Ω are admissible state and control ranges.

In order to focus on how to handle the state constraints, a standard OCP, which minimizes fuel consumption, is considered, where L, x, and u are defined by the fuel power (P_f), battery state of charge (SoC), and engine torque, respectively. The standard OCP with the battery internal resistance model (Guzzella and Sciarretta, 2013) can be formulated, as follows:

$$\min\left\{\int_0^T P_f\,(u,\tau)d\tau\right\} \tag{4}$$
$$\text{s.t. } \dot{x} = f_b(u,x,t) = -I_b(u,x,t)/Q_0, \tag{5}$$
$$x(0) = x_0, \quad x(T) = x_T, \tag{6}$$
$$x \in X, \quad u \in \Omega, \tag{7}$$

where I_b is the battery current as a function of open-circuit voltage ($V_{b.OC}$), resistance (R_b), and battery power (P_b), ($I_b = \left(V_{b.OC} - \sqrt{V_{b.OC}^2 - 4R_bP_b}\right)/(2R_b)$). Q_0 is the nominal charge capacity of the battery.

Using Pontryagin's minimum principle (PMP), optimal control input can be evaluated as

$$u^*(t) = arg\,min_{u\in\Omega} H(u,x,t), \tag{8}$$

where H denotes the Hamiltonian.

2.1 *State-unconstrained Problem*

In the absence of state constraints, the Hamiltonian is defined, and the corresponding boundary value problem (BVP) is formulated with (8), as follows:

$$\begin{aligned} H(u,x,t) &= P_f(u,t) + \lambda(t)f_b(u,x,t) \\ &= P_f(u,t) + s(t)P_{ech}(u,x,t), \end{aligned} \tag{9}$$
$$\dot{x}^*(t) = f_b(u^*,x^*,t) \approx f_b(u^*,t), \tag{10}$$
$$\dot{\lambda}^*(t) = -\lambda^*(t)\frac{\partial f_b}{\partial x} \approx 0, \tag{11}$$
$$x^*(0) = x_0, \quad x^*(T) = x_0, \tag{12}$$

where s represents the equivalence factor that is an explicit function of co-state, $s = -\lambda/(Q_0V_{b.OC})$. The resulting electric battery power, P_{ech}, is computed by $P_{ech} = I_bV_{b.OC}$.

With assumption that the battery loss depending on SoC is negligibly small, the PMP framework is also called equivalent consumption minimization strategy (Sciarretta et al., 2007). In this context, constant s must be properly computed because it is highly dependent on the driving cycle.

2.2 *State-constrained Problem based on a Penalty Method*

A ith pure state inequality constraint is described as,

$$h_i(x,t) < 0, \quad i = 1,2,\cdots l. \tag{13}$$

A penalty method is introduced due to its benefit to easily solve complex OCP with the state constraints. Using the penalty function, the state constraints can be augmented in original cost function, thereby transforming the constrained OCP (1-3) into an unconstrained OCP:

$$\min J = \int_0^T L(u,x,\tau) + r\sum_{i=1}^{l}\gamma\big(h_i(x,\tau)\big)\,d\tau \tag{14}$$
$$\text{s.t. } \dot{x} = f(u,x,t), \tag{15}$$
$$u \in \Omega, \tag{16}$$

where r and γ denote the penalty parameter and the penalty function, respectively.

Sub-optimal unconstrained solution by the augmented cost function can converge to the solution of the original constrained OCP as r increases or decreases depending on its type. There exists two types of penalty functions, which are distinguished by the convergence direction: 1) interior, or 2) exterior. Interior penalty function (IPM) is non-negative for an interior (feasible) region ($h < 0$) and has a diverging asymptotic behaviour near the state boundary condition ($h = 0$); otherwise, this function is zero. Thus, each IPM solution is always in the interior region, and approaches the state boundary condition if $r \to 0$. On the other hand, an exterior penalty function (EPM) is engaged to add a high cost only if the state constraints become active; thus, solutions start from the exterior region ($h > 0$), and finally approach the state boundary condition if $r \to \infty$.

With SoC limitations ($l = 2$), the augmented Hamiltonian can be defined, and the corresponding BVP is formulated with (8), as follows:

$$\begin{aligned} H(u,x,t) = P_f(u,t) &+ r\sum_{i=1}^{2}\big[\gamma\big(h_i(x,t)\big)\big] \\ &+ s(t)P_{ech}(u,x,t), \end{aligned} \tag{17}$$
$$\dot{x}^*(t) \approx f_b(u^*,t), \tag{18}$$
$$\dot{s}^*(t) \approx -r\cdot\sum_{i=1}^{2}\left[\frac{\partial\gamma\big(h_i(x^*,t)\big)}{\partial x}\right] = f_s(x^*,t), \tag{19}$$
$$x^*(0) = x_0, \quad x^*(T) = x_0. \tag{20}$$

where $h_1 = x - x_{max}$ and $h_2 = x_{min} - x$ represent maximum and minimum SoC bound. Note that, with respect to (10-12), the equivalence factor dynamics (19) is non-zero.

The IPM and EPM are chosen as (21) and (22), respectively, for guaranteeing the smooth property at $h = 0$,

$$\gamma_I\big(h(x,t)\big) = \begin{cases}(-h)^{-n_I} & \text{if } h < 0 \\ 0 & \text{else}\end{cases}, \tag{21}$$
$$\gamma_E\big(h(x,t)\big) = \begin{cases}0 & \text{if } h < 0 \\ h^2 & \text{else}\end{cases}, \tag{22}$$

where subscripts I and E denote the interior and the exterior penalty function, respectively. n_I determines IPM shape.

3. HYBRID OPTIMIZATION TOOL (HOT)

3.1 *Generic Hybrid Powertrain Architecture*

To cover the possible configurations of HEVs, HOT uses a generic hybrid powertrain architecture which composed of engine, two electric machines, battery, transmission, and two connectors for torque or speed coupling and one connector for pure electric mode; the details are in (Chasse et al., 2011).

3.2 *Vectorization*

As explained in (Sciarretta et al., 2015), the HOT must execute a triple-nested loop algorithm including an outer loop (s_0-loop) for initial co-state search, a middle loop (t-loop) for satisfying the desired final SoC, and an inner loop (u-loop) for finding the optimal control input. However, such algorithm is computationally demanding; thus, it is replaced by a vector operation. Hamiltonian is vectorized and thus formed as a three-dimensional array, $H(u,t,s_0)$. Firstly, minimizing Hamiltonian w.r.t. the u-dimension results in $u^*(t,s_0)$ as two-dimensional array. This optimal control policy generates both trajectories of the fuel power ($P_f^*(t,s_0)$) and the battery current ($I_b^*(t,s_0)$). Then, summing the battery current over the t-dimension results in a one-dimensional array for the final SoC, $x_f^*\,(s_0)$. Finally, the optimal value of s_0 that satisfies the terminal condition, s_0^*, is selected and then the minimal fuel consumption is obtained as

$$x_T^*\,(s_0) = x_0 + \frac{1}{Q_0}\sum_{t=0}^{T}[-I_b^*(t,s_0)], \tag{23}$$

$$m_{f.T}^*(s_0) = \frac{1}{H_f}\sum_{t=0}^{T}\left[P_f^*(t,s)_0\right], \tag{24}$$

$$m_{f.T}^{min} = m_{f.T}^*(s_0^*) \text{ with } x_T^*\,(s_0^*) = x_0. \tag{25}$$

where H_f is the fuel lower heating value.

3.3 *Picard's Method*

Because of state-dependent Hamiltonian, it is necessary to solve a state-dependent ordinary differential equation (ODE) with an initial condition, i.e., an initial value problem (IVP),

$$\dot{y}(t) = f(y,t) \text{ with } y(0) = y_0, \tag{26}$$

that can be rewritten as an integral equation,

$$y(t) = y_0 + \int_0^t f(y,\tau)d\tau. \tag{27}$$

The Picard's method iteratively solves state-dependent ODE based on the form (27). This iterative process converges to the solution of the IVP under some conditions (Falb, P. L., 1969; Nagle et al., 2000). The general Picard iteration process can be expressed by the following equations,

$$\bar{y}^{(0)}(t) = y_0, \quad \forall t \in [0,T], \tag{28}$$

$$\bar{y}^{(k+1)}(t) = y_0 + \int_0^t f\left(\bar{y}^{(k)},\tau\right)d\tau, \tag{29}$$

where k denotes the iteration number.

The penalty method also causes a state dependency in s dynamics (19). The Picard's iteration starts at a constant initial guess of state and the equivalence factor trajectories.

$$\bar{x}^{(0)}(t) = x_0,\ \bar{s}^{(0)}(t) = s_0, \quad \forall t \in [0,T], \tag{30}$$

and then proceeds as

$$\bar{x}^{(k+1)}(t) = x_0 + \frac{1}{Q_0}\sum_0^t\left[-\bar{I}_b^{*(k)}\left(t,\bar{x}^{(k)},\bar{s}^{(k)}\right)\right], \tag{31}$$

$$\bar{s}^{(k+1)}(t) = s_0 + \sum_0^t\left[\bar{f}_s^{*(k)}\left(t,\bar{x}^{(k)},\bar{s}^{(k)}\right)\right], \tag{32}$$

where f_s indicates the s dynamics (19). The Picard's method convergence is achieved when the difference between two successive trajectories is smaller than a prescribed tolerance.

In order to guarantee the convergence of the Picard's method within a finite number of iterations, the IVP must have an unique solution. However, the IPM cannot guarantee Lipschitz continuity at the state boundary condition unlike the EPM, resulting in the numerical difficulty to find s_0^* in HOT. This numerical issues are discussed in Section 4.

4. IMPLEMENTATION ISSUES

4.1 *Convergence*

The array-based HOT constructs terminal relation, $x_T^*(s_0)$, accurately and rapidly in order to find s_0^*. To this end, the penalty method implementation into the HOT should converge Picard's iterations. However, the IPM causes non-convergence issue due to discontinuity at $h = 0$. This is due to the fact that the initial guess of state trajectory (e.g. initial SoC) cannot enforce the SoC boundary constraints and thus the second iteration trajectory would be in the exterior region. Having non-converged SoC trajectories cannot fully construct the terminal relation and consequently deteriorates the accuracy to predict minimal fuel consumption. Therefore, a transition point, h_0, is introduced for guaranteeing the continuity, leading to *extended interior penalty method* (eIPM), as follows:

$$\gamma_{I.n} = \begin{cases} (-h)^{-1} & h \le h_0 \\ (-h_0)^{-1}\left\{\left[\frac{h}{h_0}\right]^2 - 3\left[\frac{h}{h_0}\right] + 3\right\} & h > h_0 \end{cases}, \tag{33}$$

$$\frac{\partial\gamma_{I.n}}{\partial h} = \begin{cases} -(-h)^{-2} & h \le h_0 \\ -\left\{\frac{(2h-3h_0)}{h_0^3}\right\} & h > h_0 \end{cases}, \tag{34}$$

where n_I must be larger than 1 theoretically (Malisani et al., 2014), but n_I is set to 1 in order to construct the eIPM.

As an example in Fig. 1, the terminal relation of IPM has a zigzag trend because of non-converged trajectories, whereas the eIPM converges to the fixed solution within a finite number of Picard's iteration and the final SoC is a monotonically increasing function of s_0. Note that the EPM always guarantees the convergence of Picard's iteration because it allows starting from the infeasible solutions.

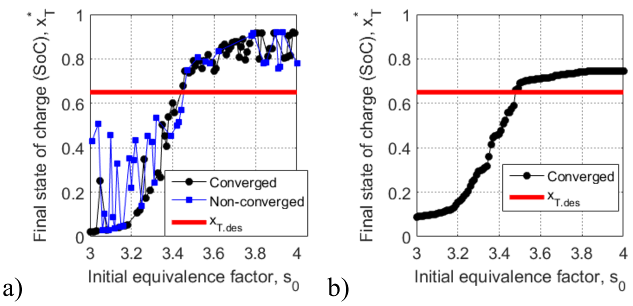

Fig. 1. Final SoC, $x_T^*(s_0)$, for $Q_a = 1.6$ Ah: a) the IPM ($r = 10^{-5}$), and b) the eIPM ($r = 10^{-5}$ and $h_0 = -10^{-3}$), where Q_a denotes allowable energy usage of battery ($Q_a = (x_{max} - x_{min})Q_0$).

4.2 Numerical Stability

If $r \rightarrow 0$ (or $r \rightarrow \infty$), the numerical instability might be caused in solving the penalized OCP (8, 18-20). Therefore, this section addresses the procedure to select an appropriate r.

Using the penalty method, the terminal relation becomes a function of two variables such as $x_T^*(s_0, r)$. Furthermore, s_0^* is also a function of the penalty parameter, $s_0^*(r)$, as shown in Fig. 2 for five values of Q_a. Note that the batteries with large Q_a (e.g. 1.8 Ah and 2 Ah) represent inactive state constraints case, and others represent active state constraints case.

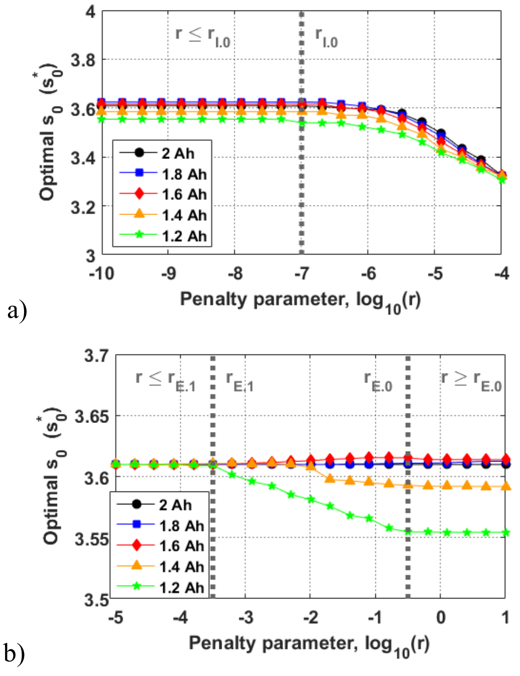

Fig. 2. Optimal value of initial equivalence factor, $s_0^*(r)$, for varying Q_a: a) the eIPM, and b) the EPM.

If there exists a lower bound, $r_{E.0} > 0$, (or an upper bound, $r_{I.0} > 0$) such that for $r \geq r_{E.0}$ (or $r \leq r_{I.0}$) any solution to the penalized OCP is also the solution of the original constrained OCP, $r_{E.0}$ (or $r_{I.0}$) is called an *exact* penalty parameter (Xing, 1994). Figure 2 shows that both penalty methods have such exact penalty parameter. In case of the eIPM, s_0^* monotonically decreases as r increases. On the other hand, in case of the EPM, there exists another threshold, $r_{E.1}$, such that each $r \leq r_{E.1}$ is too small to have an effect on the cost in the penalized OCP. In the transition region ($r_{E.1} < r < r_{E.0}$), the violation level of state constraints decreases with the increase in r. For this reason, the transition region becomes wider and deeper when the Q_a decreases. Note that s_0^* is also constant if Q_a is large enough not to activate state constraints.

In case of the EPM, however, r affects the stiffness of the equivalence factor dynamics (19), and thus using $r_{E.0}$ (or $r_{I.0}$) might not numerically provide the optimal solution because HOT uses Euler ODE solver with a typical time step of 1 s. To solve this stiffness problem in which a large change of the final SoC occurs in the vicinity of s_0^*, a smaller time step or another ODE solver could be used (Rao, 2014). However, both approaches significantly increase the computational time, thus, the direct use of $r_{E.0}$ (or $r_{I.0}$) is inconvenient, time consuming, and not robust.

If $r \rightarrow r_{E.0}$, the state constraints are nearly preserved, but the numerical difficulties illustrated above occurs; otherwise, it is easy to find BVP solution but that generally yields some violation of state constraints. Therefore, the trade-off relation between the preservation of the state constraints and reduction of the computational time should be quantified to optimally choose r. To this end, three performance indices are used. A level of stiffness (L.o.S.) is defined by the sensitivity of final SoC to the initial equivalence factor at s_0^*,

$$\mathrm{L.o.S.} = [\partial x_T^* / \partial s_0]_{s_0 = s_0^*}. \qquad (35)$$

A level of violation of the state constraints (L.o.V.) is defined as the maximum difference between violating SoC and any SoC bound,

$$\mathrm{L.o.V.} = \max(0, x^* - x_{max}, x_{min} - x^*)) \times 100\ [\%]. \qquad (36)$$

And the level of the accuracy loss (L.o.A.) is defined as,

$$\mathrm{L.o.A.} = |x_T^* - x_0| / x_0 \times 100\ [\%]. \qquad (37)$$

From the fact that higher stiffness leads to higher computational time, Fig. 3 shows that the computational time trades off with the violation of state constraints for the EPM. As r decreases, L.o.V. decreases, whereas the computational time increases with the stiffness. For very large r, the BVP solution cannot be accurately computed (L.o.A. > 3 %) unless the time step size decreases because of the numerical instability (red square in Fig 3).

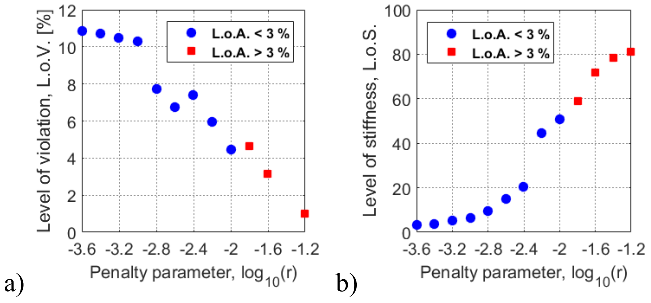

Fig. 3. Results of the EPM: a) L.o.V., and b) L.o.S.

In case of the eIPM, the transition point, h_0, plays same role as r of EPM in avoiding numerical difficulties. In this paper, h_0 (eIPM) or r (EPM) can be chosen to produce the desired balance between the performance objectives.

5. EVALUATION

5.1 Configuration of Parallel HEV

A pre-transmission parallel HEV is considered for HOT evaluation and its configuration is summarized in Table 1. Both the engine and electric machine are described by their efficiency map.

5.2 Simulation Environment

The Artemis Highway Cycle is chosen with two battery sizes (3 Ah and 4 Ah) for evaluation of the HOT performance. The simulation is performed on a standard laptop with a 3.50 GHz

Intel quad core chip and 16.0 GB RAM using Matlab 2015b. In simulation, maximum (or minimum) SoC bound is set to 85 % (or 45 %) and the initial SoC is set to 65 %.

Table 1. Configuration of parallel HEV

Vehicle parameter	Mass [kg]	1814
	Road load: c_0, c_1, c_2	93.5, 5.29, 0.536
Transmission ratio	Gear box (5 gears): R_d	[15.5, 8.21, 5.81, 4.43, 3.48]
	R_m, R_c, R_n, R_g	3.31, 1, 1(On)/0(Off), 0
Engine	Max torque [Nm]	131 @340 rad/s
	Max power [kW]	59.9 @550 rad/s
Electric machine	Max torque [Nm]	28
	Max power [kW]	37.8
Battery	Capacity [Ah]	3, 4

Note. the details are in (Chasse et al., 2011).

HOT set-up is summarized as follows: engine torque variable is discretized in 23 values such that 20 values are equally spaced between minimum and maximum engine torque corresponding to the engine speed, while the additional three values represent ICE-only torque, zero torque at nonzero (ICE-on) or zero engine speed (ICE-off), respectively. The gear variable is discretized in 6 values such that 5 values represent gear box (ICE-on) and the last one represents the second gear ratio in ICE-off mode. The engine on/off and gear-shift strategies can be also considered as additional control variables, but gear-shift strategy is fixed as one computed by the baseline HOT without the state constraints in order to focus on the effect of enforcing the SoC boundary constraints. The initial equivalence factor is firstly discretized in 101 values (1–5). This range is then narrowed using the HOT results at the end of the Picard iterations. This method guarantees a high accuracy of the penalty method without sacrificing the computational time too much.

5.3 *Results and Discussion*

Results of HOT in combination with the penalty method are presented w.r.t the following criteria: (1) computational time (T_c), (2) the level of optimality loss (L.o.O.) defined as

$$\mathrm{L.o.O.} = \left|m_{f.T}^{HOT} - m_{f.T}^{DP}\right| / m_{f.T}^{DP} \times 100\ [\%], \qquad (38)$$

where the value of $m_{f.T}^{DP}$ is calculated by dynamic programming (DP), (3) L.o.V., and (4) L.o.A.

The both penalty methods are compared to three other implementations: 1) DP (Sundstrom et al., 2010), 2) the baseline HOT without the state constraints (Baseline), and 3) the baseline HOT with a simple rule (S. rule), which avoids the violation of state constraints as follows: ICE-on mode is engaged if $SoC > SoC_{max}$ (or $SoC < SoC_{min}$) and during braking, mechanical braking is used instead of regenerative braking for preventing overcharging. Note that this simple rule must use t-loop because forcibly changed operation mode cannot converge the Picard's iterations.

As shown in Table 2, both penalty methods converge within the finite number of iterations, while satisfying a high accuracy (L.o.A. < 4 %). Moreover, they experience a small loss of optimality (L.o.O. < 1.5 %), compared to DP, and reduce the violation level compared to the baseline HOT, but the state constraints violation cannot be completely avoided because of the trade-off relation with the computational time. In case of the baseline HOT, eIPM, and EPM, the computational time strongly depends on the number of the Picard's iterations (in bracket in T_c column), For example, the eIPM with the 3 Ah battery requires 3 and 13 Picard's iterations for generating the baseline HOT solution to narrow the $s_0$range and the penalized solution, respectively. On the other hand, because of the t-loop, the simple rule results in large computational burden (around 20 s).

As shown in Fig 4 a), the penalty methods generate several jumps of the equivalence factor to preserve the SoC boundary constraints. However, the simple rule is instantaneously engaged if the SoC boundary constraints are active, leading to a different SoC trajectory in Fig 4 b). As a further example of problems induced by the simple rule, the ICE-on mode is forcibly operated (700–900 s), and thus the desired wheel torque cannot be satisfied due to the maximum engine torque limit, in Fig 4 c).

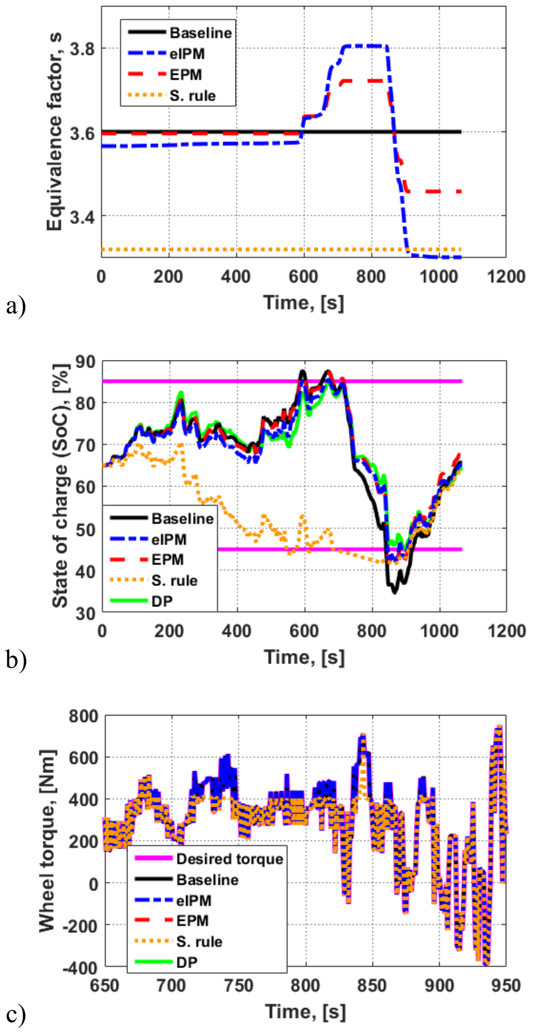

Fig. 4. Simulation results for $Q_0 = 3$ Ah: a) equivalence factor, b) SoC, and c) resultant wheel torque

At least for the small battery sizes considered, the baseline HOT seems to have a small loss of optimality, but this performance would be deteriorated for more aggressive driving scenario. On the other hand, the penalty method could guarantee the near-optimal fuel consumption while enforcing the state constraints, despite of the driving scenarios.

Table 2. Summary of the simulation results

Q_0		T_c [s]	L.o.O.	L.o.V.	L.o.A.
			[%]	[%]	[%]
3Ah	Baseline	1.47 (3)	0.95	10.4	0.97
	S. rule	20.8	2.29	3.47	2.08
	eIPM	8.45 (3, 13)	1.10	3.27	0.12
	EPM	6.90 (3, 10)	1.19	4.24	3.97
	DP	27.0	0	0	1.24
4Ah	Baseline	1.52 (3)	1.29	4.38	2.27
	S. rule	21.3	0.25	1.12	1.00
	eIPM	5.40 (3, 7)	1.31	1.13	1.92
	EPM	4.20 (3, 5)	1.26	1.50	0.96
	DP	26.8	0	0	1.27

6. CONCLUSIONS

This paper introduces the fast computing, PMP-based HOT with the penalty method in order to enforce the state constraints, and discusses the implementation issues. The penalty method has an advantage over the other indirect methods because it can solve a complex state-constrained OCP without a priori knowledge of the jump conditions. By quantifying the trade-off relation between the state constraints violation and the computational demanding in a systematic manner, the transition point and the exterior penalty parameter can be optimized. A case study of the parallel HEVs showed that both penalty methods yield the accurate, near-optimal solution that nearly preserves the state constraints while maintaining a small computational burden. Future works include further analysis on OCP with two state variables in HEV applications as well as design of real-time EMS based on the insights from the penalty method .

REFERENCES

Chasse, A. and Sciarretta, A. (2011). Supervisory of hybrid powertrains: An experimental benchmark of offline optimization and online energy management. *Control Engineering Practice*, 19(11), 1253-1265.

Guzzella, L. and Sciarretta, A. (2013). *Vehicle propulsion systems, introduction to modelling and optimization*. Springer.

Falb, P. L. and de Jong, J. L. (1969). *Some successive approximation methods in control and oscillation theory*. Academic Press.

Hartl, R. F., Sethi, S. P. and Vickson, R. G. (1995). A survey of the maximum principles for optimal control problems with state constraints. *SIAM Review*, 37(2), 181–218.

Jianning, J. and Sciarretta, A. (2016). Design and control co-optimization for hybrid powertrains: development of dedicated optimal energy management strategy. *8th IFAC Symposium on Advances in Automotive Control AAC 2016*, 49(11), 277-284.

Kim, N., Rousseau, A., and Lee, D. (2011). A jump condition of PMP-based control for PHEVs. *Journal of Power Sources*, 196(23), 10380-10386.

Malisani, P., Chaplais, F., and Petit, N. (2014). An interior penalty method for optimal control problems with state and input constraints of nonlinear systems. *Optimal Control Applications and Methods*, 37(1), 3-33.

Michel, P., Charlet, A., Colin, G., Chamaillard, Y., Bloch, G., and Nouillant, C. (2017). Optimizing fuel consumption and pollutant emissions of gasoline-HEV with catalytic converter. *Control Engineering Practice*, 61, 198-205.

Nagle, R., Saff, E., and Snider, A. (2000). *Fundamental of differential equations*, Chapter 13, Addison-Wisley.

Pérez, L. V., De Angelo, C. H., and Pereyra, V. L. (2016). Determination of the equivalent consumption in hybrid electric vehicles in the state-constrained Case. *Oil and Gas Science and Technology*, 71(30).

Pourabdollah, M., Murgovski, N., Grauers, A., and Egardt, B. (2013). Optimal sizing of a parallel PHEV powertrain. *IEEE Transactions on Vehicular Technology*, 62(6), 2469-2480.

Rao, A. V. (2014). Trajectory optimization: a survey. In Waschl, H., Kolmanovsky, I., Steinbuch, M., and Re, L., *Optimization and optimal in automotive systems*, Chapter 1, Springer.

Sciarretta, A. and Guzzella, L. (2007). Control of hybrid electric vehicles. *IEEE Control Systems Magazine*, 27(2), 60-70.

Sciarretta, A., Dabadie, J. C., and Font, G. (2015). Automatic model-based generation of optimal energy management strategies for hybrid powertrains. *International Conference : SIA Powertrain*.

Serrao, L., Onori, S., and Rizzoni, G. (2011). A comparative analysis of energy management strategies for hybrid electric vehicles. *Journal of Dynamic Systems, Measurement and Control, Transactions of the ASME*, 133(3).

Serrao, L., Sciarretta, A., Grondin1, O., Chasse, A., Creff, Y., Di Domenico, D., Pognant-Gros, P., Querel, C., and Thibault, L. (2013). Open issues in supervisory control of hybrid electric vehicles: a unified approach using optimal control methods. *Oil and Gas Science and Technology*, 68(1), 23-33.

Sundstrom, O., Ambuhl, D., and Guzzella, L. (2010). On implementation of dynamic programming for optimal control problems with final state constraints. *Oil and Gas Science and Technology*, 65(1), 91-102.

Van Keulen, T., Gillot, J., De Jager, B., and Steinbuch, M. (2014). Solution for state constrained optimal control problems applied to power split control for hybrid vehicles. *Automatica*, 50(1), 187-192.

Xing, A.-Q. and Wang, C.-L. (1989). Applications of the exterior penalty method in constrained optimal control problems. *Optimal Control Applications and Methods*, 10(4), 333-345.

Xing, A-Q. (1994). The exact penalty function method in constrained optimal control problems. *Journal of mathematical analysis and applications*, 186, 514-522.