Noname manuscript No.
(will be inserted by the editor)

On Minimizing the Energy Consumption of an Electrical Vehicle

Abdelkader Merakeb [1,2] · Frédéric Messine [2] · Mohamed Aidène [1].

Received: date / Accepted: date

Abstract The electrical vehicle energy management problem can be expressed as an optimal control one. In this work, we discuss on a new formulation and about the way to approximate this optimal control problem of Bang-Bang type via a discretization technique associated with a Branch and Bound algorithm. The problem that we focuss on is to compute the minimal energy consumption of an electrical car achievable on a given driving cycle. Some numerical experiments validates our methodology.

Keywords Discretization techniques · Optimal Control · Bang-Bang problem · Branch and Bound · Electrical vehicle · Energy consumption.

1 Introduction

Electrical vehicle uses an electrical energy source for its displacement which can be reversible. The aim of this work is to develop a method to find the control strategies and compute the minimal energy consumption achievable by the electrical vehicle on a given driving cycle. The main approaches have been studied, including the direct and indirect methods. Indirect methods, based on the Pontryagin maximum principle are efficient for their speed and accuracy. However, their implementation using shooting methods may, in practice, deal with some difficulties, for example when the structure of control is bang-bang type. Indeed, these methods transform the original problem by solving a system of nonlinear equations. In this system is no regular, and its numerical solution is extremely sensitive to the choice of an initial point. Note also that the presence of a constraint on the state variables increases the complexity of

[1] Université Mouloud Mammeri de Tizi-Ouzou, Algérie.
[2] Université Toulouse, ENSEEIHT-IRIT, France.
E-mail: merakeb_kader@yahoo.fr
E-mail: Frederic.Messine@n7.fr
E-mail: aidene@mail.ummto.dz

its use. Direct methods, traditionally involves total or partial discretizations of the problem, and then use various approaches (SQP and interior point techniques for example) to solve the large scale optimization problem arising. Nevertheless, they are relatively imprecise and can lead to problems of large sizes depending on the used step of discretization. Thus, these methods are less suitable for certain special cases, including problems with a bang-bang structure yielding a large number of switching operations. We discuss about the way to solve efficiently the problem of the minimization of the energy which is consummated by an electrical vehicle during an imposed displacement, see [1] for an overview on this type of problems. In order to solve this problem, we reformulate it following the construction of a current regulator technique to obtain a global optimization problem that can be solved using Branch-and-Bound algorithm. Using this method, the benefit obtained is the reduction of the computation requirements and the algorithm can be embedded within a real time predictive control framework.

2 Model of the electrical vehicle

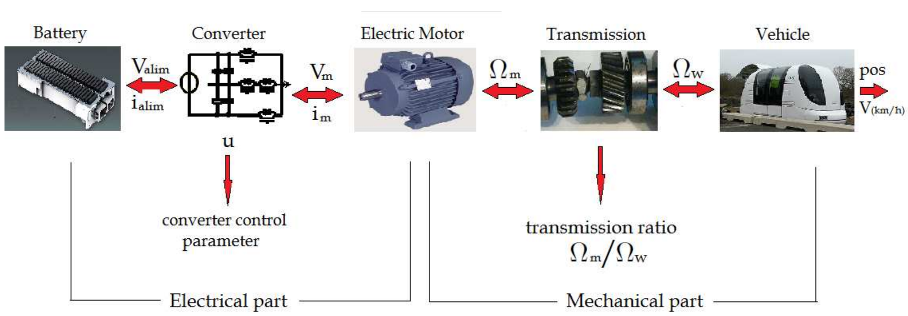

Fig. 1 Standard traction link

Fig.1 concerns the standard traction link between the different components of the constitution of the vehicle in question. The modeling of this transmission chain consists of two parts: electrical part in association with battery, converter and motor; and mechanical part in association with transmission and vehicle. Each part is described by a differential equation (one for the current inside the motor and one for speed).
The energy confined in the battery is regulated in the converter using the control parameter u, and the current delivered to the motor is submitted to the differential equation (1)

$$\frac{di_m(t)}{dt} = \frac{u(t)V_{alim} - R_m i_m(t) - K_m \Omega_m(t)}{L_m} \tag{1}$$

The movement of the motor is transmitted to the vehicle via the transmission provided with a coefficient ratio. The differential equation in this part is given in the speed rotor parameter by (2)

$$\frac{d\Omega_m(t)}{dt} = \frac{1}{J}\left(K_m i_m(t) - \frac{r}{K_r}\left(MgK_f + \frac{1}{2}\rho S C_x \left(\frac{\Omega_m(t) r}{K_r}\right)^2\right)\right) \qquad (2)$$

To know the position of the vehicle, we can infer it from the third differential equation (3)

$$\frac{dpos(t)}{dt} = \frac{\Omega_m(t) \times r}{K_r} \qquad (3)$$

$V(t) = \frac{3.6 \times r}{K_r}\Omega_m(t)$ gives linear celerity of the car in km/h.

The performance index is given by the energy formula (4)

$$E(t_f, i_m, u) = \int_0^{t_f} (u(t) i_m(t) V_{alim} + R_{bat} u^2(t) i_m^2(t)) dt \qquad (4)$$

where E represents the electrical energy consummated during the displacement on the cycle $[0, t_f]$. The quadratic term reflects the losses due to the internal resistance of the battery. The system allows to recover the kinetic energy under deceleration to recharge the battery.

The problem that we are interested with can be formulated as an optimal control problem (5):

$$\begin{array}{l}
\min_{i_m(t), \Omega(t), pos(t), u(t)} \quad E(t_f, i_m, u) \\
s.t. \\
\qquad \begin{cases}
\dot{i_m}(t) = \frac{u(t)V_{alim} - R_m i_m(t) - K_m \Omega(t)}{L_m} \\
\dot{\Omega}(t) = \frac{1}{J}\left(K_m i_m(t) - \frac{r}{K_r}\left(MgK_f + \frac{1}{2}\rho S C_x \left(\frac{\Omega(t) r}{K_r}\right)^2\right)\right) \\
\dot{pos}(t) = \frac{\Omega(t) r}{K_r}
\end{cases} \\
\\
\qquad |i_m(t)| \leq 150 \\
\qquad u(t) \in \{-1, \ +1\} \\
\\
\qquad (i_m(0), \ \Omega(0), \ pos(0)) = (i_m^0, \ \Omega^0, \ pos^0) \in^3 \\
\qquad (i_m(t_f), \ \Omega(t_f), \ pos(t_f)) \in \mathcal{T} \subseteq^3
\end{array} \qquad (5)$$

The state variables are: (i) i_m the current inside the motor; (ii) Ω the angular speed; (iii) pos is the position of the car. The control u is in $\{-1, 1\}$ (a Bang-Bang structure). In this problem, we have a constraint on a state variable to limit the current inside the motor in order to discard the possibility to destroy it. The other terms are fixed parameters and represent some physical things: $K_r = 10$, the coefficient of reduction; $\rho = 1.293 kg/m^3$, the air density;

$C_x = 0.4$, the aerodynamic coefficient; $S = 2m^2$, the area in the front of the car; $r = 0.33m$, the radius of the wheel; $K_f = 0.03$, the constant representing the friction of the wheels on the road; $K_m = 0.27$, the coefficient of the motor torque; $R_m = 0.03\Omega$, the inductor resistance; $L_m = 0.05$, inductance of the rotor; $M = 250kg$, the mass; $g = 9.81$, the gravity constant; $J = M \times r^2/K_r^2$; $V_{alim} = 150$, the battery voltage; $R_{bat} = 0.05\Omega$, the resistance of the battery. This problem is subject to the boundary conditions. The initial conditions are given by the starting point $(i_m^0,\ \Omega^0,\ pos^0)$ at the starting time $t_0 = 0$, but the target set $\mathcal{T}$ at the final time t_f is free and depends on the instances of the problem; it could be a point of R^3 but one or two variables could not be fixed: for example just the final position equal to $100m$ is required (see the numerical section).

This problem is hard to solve directly by using classical optimal control techniques. We try to solve it by using the Pontriagin method based on shooting techniques and also by using a direct shooting algorithm, [3]. For the moment, the fact that we have a constraint on the state associated with the fact that it is a Bang-Bang control involves a lot of difficulties which does not permit to obtain solutions (even local ones).

With direct methods, the procedure leads to problems of large sizes depending on the used step of discretization. In our case, if we discretize all the cycle of time by fixing the value of the control, it is necessary to have very small steps else the value of the current inside the motor will change too roughly. So, these methods are less suitable for certain special cases, including problems with a bang-bang structure yielding a large number of switching operations.

Dynamic programming using Hamilton- Jacobi Bellman equation, is a technique which compares the optimal solution with all the other solutions. This global comparison, therefore, leads to optimality conditions which are sufficient. The only disadvantage of DP (which often rules out its use), is that it can easily give rise to enormous computational requirements, which is the case with our problem [2].

Thus, in this paper we propose another original methodology to solve this problem yielding to some discretized problems which are solved using an exact Branch and Bound algorithm. This new method provide exact results for the discretized formulations which correspond to approximations of the global solutions of Problem (5).

3 Approximation of the Problem

First we remark that the energy formula is only depending on the current and the control. Therefore, it is just required to search the trajectory of the current that minimizes the consumption of the energy. If we discretize all the interval of time $[0, t_f]$ by fixing the value of the control u, it is necessary to have very small steps about 10^{-3} for at least to be able to control the current through the motor (else the value of the current will change too roughly). That will generate a very huge mixed integer non-linear global optimization problem which is, for the moment, impossible to solve using direct methods of optimal control.

An idea, which directly comes from the numerical simulation of the behavior of the car, is to impose during some short sample of time the value of the current inside the electrical motor of the vehicle. This is possible using the control parameter $u(t)$. Thus, if we impose a reference current $iref$, if $i_m(t) > iref + \frac{\Delta}{2}$ then $u(t) := -1$ and if $i_m(t) < iref - \frac{\Delta}{2}$ then $u(t) := 1$ (see fig.2).

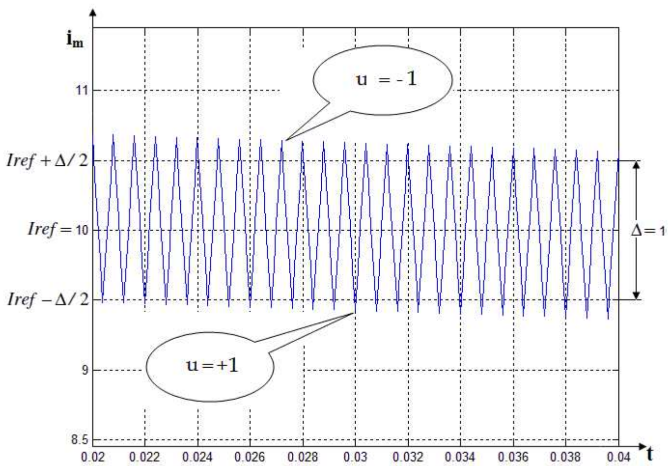

Fig. 2 Management principle by reference current

The control switches between the two values of u when the current inside the motor exceeds the value of iref with respect to the tolerance Δ.

The exceeding boundaries of the Δ band over a step time is due to the fact that the current is close to the borders of the Δ band at the end of the previous step. The maximum overflow is 0.3 amps for 10^{-4} time discretization stepsize.

This technique is just a way to construct a regulator of current which is a first step before making a speed regulator for an electrical car. Hence, using this, the following differential system of equations can be solved:

$$VS_{t_0,iref}(t) := \begin{cases} \dot{E}(t) = u(t)i_m(t)V_{alim} + R_{bat}u^2(t)i_m^2(t) \\ \dot{i_m}(t) = \frac{u(t)V_{alim} - R_m i_m(t) - K_m \Omega(t)}{L_m} \\ \dot{\Omega}(t) = \frac{1}{J}\left(K_m i_m(t) - \frac{r}{K_r}\left(MgK_f + \frac{1}{2}\rho S C_x \left(\frac{\Omega(t) r}{K_r}\right)^2\right)\right) \\ \dot{pos}(t) = \frac{\Omega(t) r}{K_r} \\ u(t) := \begin{cases} -1 \text{ if } i_m(t) > iref + \frac{\Delta}{2} \\ +1 \text{ if } i_m(t) < iref - \frac{\Delta}{2} \\ u(t) \text{ else.} \end{cases} \\ (E(t_0), i_m(t_0), \Omega^(t_0), pos(t_0)) = (E^{t_0}, i_m^{t_0}, \Omega^{t_0}, pos^{t_0}) \in R^4 \\ u(t_0) := 1; \end{cases} \tag{6}$$

where t_0 is the initial time which is not necessary equal to 0.

This system of differentiable equations can be efficiently solved using a classical differentiable integrator such as for example *Euler, RK2, RK4* with a step of time less than 10^{-3}. The function $VS_{(t_0,iref}(t)$ will compute in theory all the values for $E(t), i_m(t), \Omega(t), pos(t)$, for all $t \in [t_0, t_f]$ but in practice only values for a discretized time $t_i \in [t_0, t_f]$ is available. Here, we are interested by the final values of the state variables, hence we define a function:

$$VSF(iref, t_0, t_f) := (E(t_f), i_m(t_f), \Omega(t_f), pos(t_f)) \in R^4,$$

all the computations are performed using the function $VS_{(t_0,iref}(t)$ which solves the system of differential equations (6) under the initial conditions $(E^{t_0}\ , i_m^{t_0},\ \Omega^{t_0},\ pos^{t_0})$.

4 The Global Optimization Problem

The main idea of this work is to subdivide the cycle of time $[0, t_f]$ into P subintervals. In each sample of time $[t_{k-1}, t_k]$ with $k \in \{1, \cdots, P\}$ ($t_k = k \times \frac{t_f}{P}$), we apply a reference current $iref_k$ which takes values in $[-150, 150]$ in order to directly satisfy the constraint on the state variable of Problem (5).

Thus, we focus on the resolution of the following global optimization problem:

$$\begin{cases} \min_{iref \in [-150,150]^P} \sum_{k=1}^{P} E_k \\ u.c. \\ \qquad (E_k, i_k, \Omega_k, pos_k) := (iref_k, t_{k-1}, t_k) \\ \qquad (E_0, i_0, \Omega_0, pos_0) = (E^0, i_m^0, \Omega^0, pos^0) \in R^4 \\ \qquad (i_P, \Omega_P, pos_P) \in \mathcal{T} \subseteq R^3 \end{cases} \tag{7}$$

Problem (7) is a good approximation of the initial problem (5) which generates just a few number of variables: P. In fact, we use a current regulator system to control the vehicle; this is also interesting in itself for a future implementation of the system in the car.

5 Dedicated Branch and Bound Algorithm

For the moment, we are not able to solve exactly the global optimization problem (7), thus we need to discretize also the possible values for the reference current: $iref \in \{-150, -150+s, -150+2\times s, \cdots, 150\}^P$; we will take integer values for s which divide exactly $[-150, 150]$. Therefore, the set of solution becomes finite and could be enumerated. Nevertheless, if we want to have a good approximation for the resolution of the global optimization problem (7) we have to discretize into small samples and the finite set of possible points becomes rapidly too huge to be entirely enumerated in a reasonable CPU-time. The idea is then to use a Branch and Bound algorithm in order to not explore all the finite set of solutions.

5.1 Computing Bounds Technique

For using such an algorithm, we have to elaborate a technique to compute bounds for the four main parameters: $E_k, i_k, \Omega_k, pos_k$ over a box $IREF \subseteq \{-150, -150+s, -150+2\times s, \cdots, 150\}^P$ and for given t_0 and t_f. In order to be more efficient, in a previous sample, we compute 4 matrices: $M_E, M_i, M_\Omega, M_{pos}$ where the columns corresponds to values when $iref$ is fixed with $i_m^{t_0} = iref$ and the rows provides values for the entities when a speed Ω^{t_0} is given (we discretize also the possible values of the speed).

		$iref = -150 + (j-1)s$
		$\vdots$
$\Omega^{t_{k-1}} = (i-1)pasV$	$\cdots$	$m_\Theta(i,j)$

$pasV$ is a discretization step of the speed values.
Θ represents one of these symbols $E,\ \Omega,\ pos$.
$e_E = (1,0,0,0),\ \ e_\Omega = (0,0,1,0),\ \ e_{pos} = (0,0,0,1)$

$$m_\Theta(i,j) =< VSF(iref, t_{k-1}, t_k),\ e_\Theta >$$

taken when computing the function $VS_{t_{k-1},iref}(t)$ over a sample time $[t_{k-1},\ t_k]$ and under the initial conditions

$$(E^{t_{k-1}},\ i_m^{t_{k-1}},\ \Omega^{t_{k-1}},\ pos^{t_{k-1}}) = (0,\ iref,\ \Omega^{t_{k-1}},\ 0).$$

For example $m_E(i,j)$ represents the value of the energy which is consummated during a sample of time $[t_{k-1}, t_k]$ when $iref$ corresponds to the jth components of the set $\{-150, -150+s, -150+2\times s, \cdots, 150\}$ with $i_m^{t_0} = iref$ and the ith row corresponds to the discretized value of the speed, the other initial values are taken equal to 0: i.e., $E^{t_{k-1}} = pos^{t_{k-1}} = 0$.

When a box $IREF$ is considered, we can compute bounds for E, i, Ω and pos by computing the integer sets I and J of the indices corresponding to the possible values of the speed at the previous sample and the possible values of $iref$. Then, we have to compute the bounds which correspond to the minimal and maximal values of $m_E(i,j), m_i(i,j), m_\Omega(i,j), m_{pos}(i,j)$ with $(i,j) \in I \times J$. To obtain the final value for E and pos, we have to sum all the lower and upper bounds.

The rest of the Branch and Bound algorithm that we develop is simple and uses the following classical principle: (i) subdivision into two (distinct) parts of the enumerate set $IREF$ (which represents the possible values for $iref$); (ii) the upper bound is updated by taking the middle of the box $IREF$ if the constraints are satisfied and if its value is better than the previous one (we start with $+\infty$); (iii) we branch following the heuristic of lowest lower bound of the energy.

5.2 Heuristics Alternative

The Branch & Bound algorithm uses the data pre-processing when computing matrices. We analyze these data in order to find properties that allow us to reduce the computing time bounds through these matrices. Indeed, the exact method described above to compute bounds is expensive for CPU-time. So, interest was paid for two heuristics $H1$ and $H2$.

For $H1$, it is taken as lower bounds, the values of each sub-matrices induced corresponding to the first row and first column $m_E(i_1,j_1)$; $m_\Omega(i_1,j_1)$; $m_{pos}(i_1,j_1)$. As upper bounds, the values corresponding to the last rows and last columns $m_E(i_n,j_m)$; $m_\Omega(i_n,j_m)$; $m_{pos}(i_n,j_m)$.

For $H2$, we keep the same bounds as the heuristic $H1$ for the position. For the energy and the speed, as lower bounds, we compute the minimum value on the first row. As upper bounds, the maximum value on the last row.

Lower bounds $\min_{j \in J} m_E(i_1,j);\ \min_{j \in J} m_\Omega(i_1,j);\ m_{pos}(i_1,j_1)$.
Upper bounds: $\max_{j \in J} m_E(i_n,j);\ \max_{j \in J} m_\Omega(i_n,j);\ m_{pos}(i_n,j_m)$.

Time computing matrices depends on the sample of time $t_k - t_{k-1} = \frac{t_f}{P}$, on the $pasV$ (step of discretization of the speed), on the integer value of s (which sudivide the intervalle $[-150,150]$ of reference current), on the $RK4$ integrator step of time and te value of Δ. If we fix the $RK4$ integrator step to 10^{-4}, $pasV = 0.1$ and $\Delta = 1$, the preprocessing time to compute matrices is proportional with $(t_k - t_{k-1})$ and inversely with s.

5.3 Algorithme B&B

1: Initialization: Let $L :=$ the initial domain in which the minimum is searched, $Emin := \infty$ the upper bound of the minimum, $pas\_Iref := s$ discretization step of reference current, $posmin := 0$ the initial position, $solmin := 0$ the initial solution.
2: Initialization of matrices: $pasV :=$ is the discretization step of the speed, $pastf := P$ denotes the length of the sample of time.
3: Computing matrices $Posf, \quad Ef, \quad Vf$.
4: **while** $L \neq \emptyset$ **do**
5:   Extracting the first element X of L.
6:   Compute the maximum width H of X and its index ν.
7:   $m :=$ compute the middle of X_ν.
8:   **for** $i = 1 \;\; 2$ **do**
9:     breakup of the element X into two blocks
10:     **if** $(i = 1)$ **then**
11:       $X := \lfloor \frac{m}{pas\_Iref} \rfloor pas\_Iref$ (first block)
12:     **else**
      $X := \lfloor \frac{m}{pas\_Iref} \rfloor pas\_Iref + pas\_Iref$ (second block)
13:     **end if**
14:     Computing lower bounds biE, $bipos$, Vbi, and upper bounds bsE, $bspos$, Vbs using heuristics $H1$, $H2$ or the exact method ME.
15:     **if** $(bspos \geq posf)$ and $(bipos \leq posf)$ and $(biE < Emin)$ **then**
16:       $midi :=$ midpoint of the block
17:       $sol := [midi/pas\_Iref] * pas\_Iref$
18:       Computing bounds $Esol, possol, Vsol$
19:       **if** $(possol \geq posf)$ and $(Esol < Emin)$ **then**
20:         $Emin = Esol$, $posmin = possol$, $solmin = sol$
21:         **if** $(H \neq 0)$ **then**
22:           $L = (L; X)$ insertion of the block X in the list L
23:         **end if**
24:       **end if**
25:     **end if**
26:   **end for**
27: **end while**
28: Results : $Emin$, $posmin$, $Vsol$.

$[.]$ denotes the integer part. $\lfloor . \rfloor$ denotes the floor function.

The general structure of this algorithm is shown schematically in four stages achieved in the following order: extraction, division, analysis, insertion. The extraction of an element of L is following the FIFO accounting: extraction on tops of the list and insertion on tail.

On step 9 of the algorithm, the technique used to split the element X into two blocks is to select the component with the largest width and divide X in the middle of this component. The loop **for** of line 8 is the core of the algorithm. Line 14 concerns the verification of constraint by computing bounds. This is

the stage of analysis that uses one of the heuristics ME, $H2$ ou $H1$. If the condition of line 15 holds, then it is certain that the global minimum is in the block X, which justifies its insertion in the list, otherwise the block is directly deleted (line 22).
Finding a good feasible solution consists to evaluate the objective function at the midpoint of X. If this solution satisfies the constraints, we compare it to the best current solution of the problem. This one is updated if it improves the current solution (line 19). The stop condition of the algorithm occurs when the whole list L is empty (line 21). This algorithm is of course exponential complexity in time and memory.

6 Numerical Experiments

For all numerical tests, we use a standard notebook with 4GB of RAM using MatLab 9. In order to compare the results in cases of variations on speed constraints, we adopt the same profile for all simulations. Our method is thus evaluated for a displacement of 100 meters and a cycle of time $t_f = 10$ seconds: $(i_m(0), \Omega(0), pos(0)) = (0, 0, 0)$; $(i_m(t_f), \Omega(t_f), pos(t_f)) \in \mathcal{T} = R \times R \times \{100\}$.

6.1 Case without constraint on the speed

In this simulation, we compare the three heuristics with parameters $P = 5$ and $s = 10$. We obtain for the discretized problem, the exact solution $iref = (150,\ 90,\ 30,\ -30,\ -110)$ corresponding to the minimum energy $Emin(10) = 23272$ J. In addition, we have $pos(10) = 100.25$ m. This calculation is performed for an integration step equal to 10^{-4} and $pasV = 0.1\ km/h$. The pre-processing for calculating matrices took 374 seconds.

	CPU (s)	E_{min} (J)	$posf$ (m)	Vf (km/h)	$iref$ $(amps)$	$Iterations$
ME	23.69	23272	100.25	11.31	$(150,\ 90,\ 30,\ -30,\ -110)$	31440
$H1$	0.70	23559	100.12	13.89	$(150,\ 90,\ 0,\ 20,\ -120)$	8819
$H2$	4.27	23272	100.25	11.31	$(150,\ 90,\ 30,\ -30,\ -110)$	15982

Table 1 *Comparison of ME, H2 and H1 for P=5, s=10.*

Table 1 shows the solutions obtained by ME, $H1$ and $H2$ for the parameters $P = 5$ and $s = 10$ which graphs are shown in figure 3. CPU time running in seconds; E_{min} is the minimum energy consumption given in joules; $posf$ is the distance traveled by the vehicle in meters; Vf is the final speed of the vehicle given in km/h; $iref$ gives the reference currents in amps. The last column gives the number of iterations required for the execution of the algorithm.

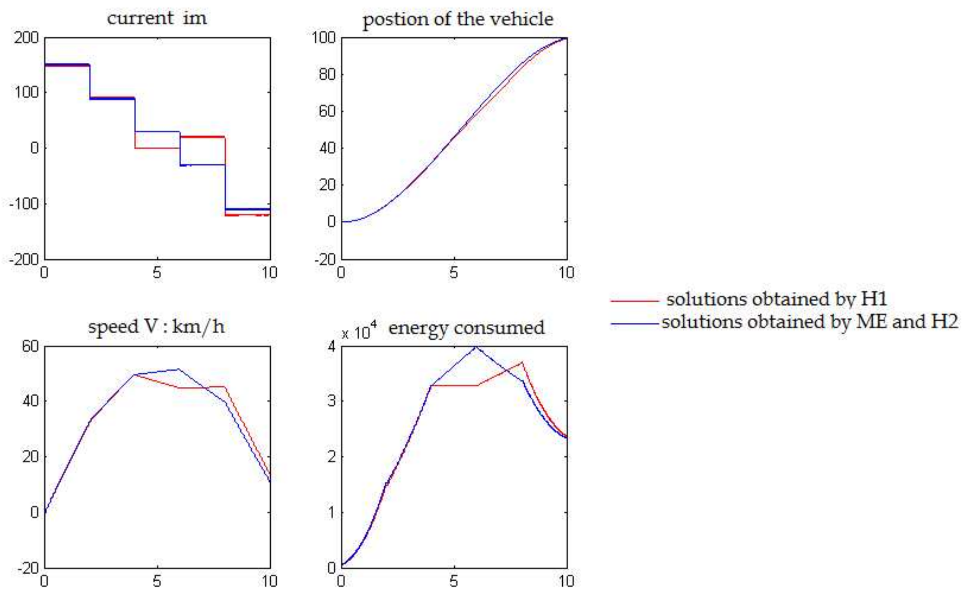

Fig. 3 Case: P=5; s=10; posf=100 m.

Note that if we calculate the solution directly using the $RK4$ numerical integrator (without using matrices), we obtain $\overline{Emin} = 23313$ J for a position $\overline{pos} = 99.37$ m. The calculation error is about 0.17% for the energy and 0.87% for the position.
Although the heuristics $H1$ gives results quite close, it is not reliable for determining the exact solution, especially in this case. Nevertheless, it can be used when we can settle for a less accurate result to reconcile computational time and solution quality. The heuristic $H2$ gives the same solution as compared to the exact method ME, it is quite robust, fast and gives exact solutions in most cases we have tested.
When we evaluate our algorithm for parameters $P = 5$ and $s = 1$, we obtain the solution $iref^* = (149,\ 86,\ 29,\ -17,\ -116)$ corresponding to the minimum energy $Emin^*(10) = 23054$ J. In addition, we have $pos^*(10) = 100.00$ m. The computation time is about 48129 s, which must be added the 659 s pre-processing, corresponding to $1,925,783$ iterations of the Branch&Bound algorithm. This very long computing time depends on parameters s and P, which is understandable for a Branch&Bound code whose complexity is exponential in P.
Therefore, an idea to get more accurate solutions, is to run the Branch and Bound code iteratively by defining more specific areas around the exact solutions obtained above, by increasing the parameter P and decreasing the parameter s.

With the solution obtained above (fig.3) $iref = (150, 90, 30, -30, -110)$ over a sampling period $[0, 2]$, $(P = 5; s = 10)$, this one is equivalent to $iref = (150, 150, 90, 90, 30, 30, -30, -30, -110, -110)$ over the sampling period $[0, 1]$, $(P = 10; s = 10)$. Spreading it on a maximum range of 40 amps, we generate a box $IREF = [130, 150] \times [130, 150] \times [70, 110] \times [70, 110] \times [10, 50] \times [10, 50] \times [-50, -10] \times [-50, -10] \times [-130, -90] \times [-130, -90]$.
Each solution found is returned for a new iteration. The following table gives the refined solutions after 3 iterations.

$Instance$	$iref$	E_{min} (J)	$posf$ (m)	Vf (km/h)	CPU (s)	$range$ $(amps)$	$Iter.$
$P = 5,$ $s = 10$	$(150,\ 90,\ 30, -30, -110)$	23272	100.24	11.31	4	300	15982
$P = 10,$ $s = 10$	$(150,\ 140,\ 90,\ 90,\ 30,$ $30, -20, -20, -100, -130)$	22972	100.10	11.48	957	40	127549
$P = 10,$ $s = 5$	$(150,\ 140,\ 95,\ 85,\ 30,$ $30, -15, -30, -100, -130)$	22852	100.04	10.99	3024	20	227598
$P = 10,$ $s = 1$	$(150,\ 139,\ 97,\ 85,\ 30,$ $30, -16, -31, -101, -131)$	22817	100.00	10.68	807	4	124854

Table 2 *Table of refined solutions: free final speed.*

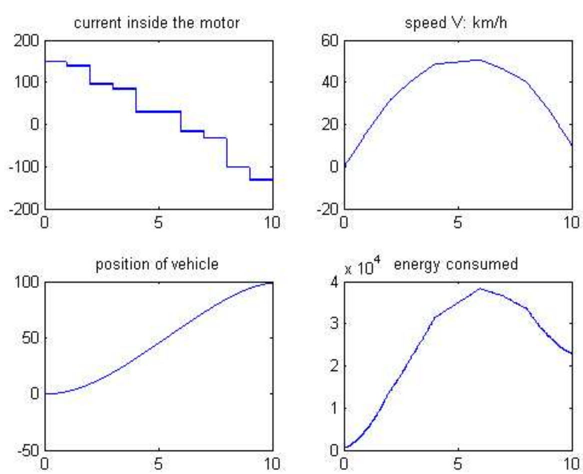

Fig. 4 Case: P=10; s=1; posf=100 m.

With the improved solutions, we obtain a gain of 2% on the performance index for $4788s$ more. The curves (fig.4) are from the latest refined solution. When computing this solution directly using the numerical integrator $RK4$

without using matrices, leads to an energy $\overline{E}min = 23,022$ J, and position $\overline{pos} = 98.80\ m$ and a final velocity $Vf = 10.28\ km/h$. The calculation error is about 0.89% for the energy, 1.2% for the position and $\pm 0.4\ km/h$ for speed.

We remark that the current i_m remains trapped around $iref$ with respect to the tolerance Δ. The values of u switches many times between -1 and $+1$; this is due to the fact that the current in the motor increases too quickly (around of $3A$ every $10^{-3}s$). Moreover, the curve of the energy decreases at the end of the cycle because this corresponds to the recovery phase.

6.2 Case with constraint on the final speed

The final speed in the curve of figure 4 is not equal to zero. However, our method by successive refinement can take this constraint into account. To compare the solutions, we simulate the same previous instances by adding a constraint on the final speed, -i.e-, a displacement of 100 m, and a cycle of time $t_f = 10$ seconds with final speed equal to zero: $(i_m(0), \Omega(0), pos(0)) = (0, 0, 0)$; $(i_m(t_f), \Omega(t_f), pos(t_f)) \in \mathcal{T} = R \times \{0\} \times \{100\}$. We obtain the following table which compute refined solutions.

$Instance$	$iref$	E_{min} (J)	$posf$ (m)	Vf (km/h)	CPU (s)	$range$ $(amps)$	$Iter.$
$P = 5$, $s = 10$	$(150,\ 110,\ 40, -70, -150)$	26517	100.72	-1.17	14	300	37437
$P = 10$, $s = 10$	$(150,\ 150,\ 130,\ 90,\ 30,$ $20, -50, -60, -150, -150)$	25646	100.00	-0.89	543	40	97429
$P = 10$, $s = 5$	$(150,\ 145,\ 130,\ 90,\ 35,$ $15, -45, -55, -150, -150)$	25362	100.11	-0.20	472	20	91304
$P = 10$, $s = 1$	$(150,\ 143,\ 129,\ 92,\ 37,$ $14, -45, -55, -148, -150)$	25259	100.00	0.00	63	4	37764

Table 3 *Table of refined solutions: null final speed.*

The minimum energy consumption for this case is higher compared to when we not consider the constraint on the final speed. The calculation of this solution without using matrices (using $RK4$ numerical integrator), gives an energy $\overline{E}min = 25,592\ J$; a position $\overline{pos} = 98.97\ m$ and a final velocity $Vf = 0.20\ km/h$. The calculation error is about 1.30% for energy, 1.03% for the position and $\pm 0.2\ km/h$ for speed. As the vehicle starts and ends with zero velocity, it necessarily passes through a phase of deceleration corresponding to the recovery phase of electrical energy. The current is to the bottom at the first moments and ends with its minimum value at the end of the cycle to be able to stop the vehicle (final velocity zero).

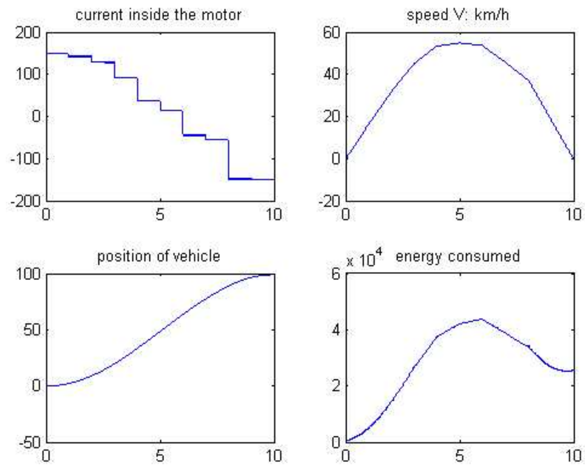

Fig. 5 Case: P=10; s=1; posf=100 m; Vf=0.

7 Conclusion

The energy management problem of an electrical vehicle has been written as an optimal control problem with bang bang control, which is, in general, difficult to solve using PMP and direct methods.
In this paper, we show an original way based on discretization and a Branch and Bound method to solve a hard global optimization problem which is an approximation of an optimal control problem. An algorithm was derived and it remains to be improve it by investigating topics such as management of routes with slopes and regulation using the reference speed.
Also, in a future work, we want to improve the efficiency of our Branch and Bound algorithm. Furthermore, we are interested by the resolution of Problem (7) directly by computing bounds.

References

1. A. Sciarreta, L. Guzzella, *Control of Hybrid Electric Vehicles - A Survey of Optimal Energy-Management Strategies*, IEEE Control Systems Magazine, Vol. 27, N. 2, pp. 60–70, 2007.
2. C. Musardo, G. Rizzoni, Y. Guezennec, B. Staccia, *A-ECMS: An Adaptive Algorithm for Hybrid Electric Vehicle Energy Management*, European Journal of Control, N. 11 (4–5), pp. 509–524, 2005.
3. J. Bernard, S. Delprat, T.M. Guerra, F. Buechi, *Fuel Cell Hybrid Vehicles: Global Optimization based on Optimal Control Theory*, International Review of Electrical Engineering, 1, 2006.
4. S. Sager, H.G. Bock, G. Reinelt, *Direct Methods With Maximal Lower Bound for Mixed-Integer Optimal Control Problems*, Math. Program. Ser. A, 118: 109-149, 2009.
5. W.R. Esposito, C.A. Floudas, *Deterministic Global Optimization in Nonlinear Optimal Control Problems*, Journal of Global Optimization 17, 97-126, 2000.

6. S.A. Nasar, *Handbook of Electric Machines,* ISBN 0-07-045888-X, McGraw-Hill Book Company, 1987.
7. G. Lacroux, *Les Actionneurs électriques pour la robotique et les asservissements,* ISBN 2-85206-270-4, TEC&DOC Lavoisier, 1985.
8. J. Bonal, *Entrainements électriques vitesse variable,* ISBN 2-7430-0138-0, TEC&DOC, 1997.
9. L. Idoumghar, D. Fodorean et A. Miraoui, *Simulated Annealing Algorithm for multi-objective optimization: Application to Electric Motor Design.* Proceeding of the 29th IASTED International Conference on Modelling, Identification and Control, pp. 190-196, February 15-17, 2010.
10. M.S. Couceiro, C.M. Figueiredo, C. Lebres, N.M. Fonseca Ferreira, and J.A. Tenreiro Machado, *Electric Vehicle Drive System with Adaptive PID Control.* Modelling, Identification, and Control - 2010.