On Minimizing the Energy Consumption of an Electrical Vehicle

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

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

Abstract. The electrical vehicle energy management can be expressed as a Bang-Bang optimal control problem. 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 focus on, is the minimization of the energy consumption of an electrical vehicle 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 to compute the minimal energy consumption achievable by an electrical vehicle on a given driving cycle. The main classical approaches have been tested on this Bang-Bang optimization problem, including the direct and indirect methods. For the moment, they did not provide any satisfactory result. Indirect methods, based on the Pontryagin maximum principle are efficient for their speed and accuracy, [9], [10]. However, their implementation using shooting methods may, in practice, deal with some difficulties, for example when the structure of control is Bang-Bang, as in our case. Indeed, these methods transform the original problem by solving a system of nonlinear equations. If this system is non regular, then its numerical solution becomes extremely sensitive to the choice of an initial point. Note also that the presence of a constraint on state variables increases the complexity of its use. Direct methods, traditionally involves total or partial discretizations of the problem, and then use various approaches (SQP or interior point techniques for example) to solve the large scale optimization problem arising. Nevertheless, they are generally 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 switches (as in our case). In this work, 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. In the literature about electrical engineering, some optimal control problems were addressed as in [1], [5], [8]. However, they concerns only hybrid vehicles and discussed about the efficiency of some adaptations of classical direct, indirect or dynamical programming approaches to solve different optimal control problems than here.

In order to solve this problem, we reformulate it following the construction of a current regulator technique to obtain a global optimization problem which is first discretized and then solved using a Branch-and-Bound type algorithm. Note that in [3], Esposito and FLoudas proposed a deterministic Branch-and-Bound algorithm based on extensions of their own $\alpha-$BB code. This new method makes it possible to find the global optimum of some optimal control problems. Unfortunately, this method cannot directly be applied to solve optimal control problem of Bang-Bang type as in our case.

In Section 2, the differential equations which come from the electrical and mechanical parts of the system are presented yielding the optimal problem that we have to solve. In Section 3, we show a method to approximate the optimal control problem by considering a system of current regulator. By this way, we provide a continuous global optimization problem. By discretization, we propose in Section 4 a method to solve, using a Branch-and-Bound algorithm, the problem of the minimization of the energy consumption of an electrical vehicle during an imposed time and displacement. In Section 5, some numerical tests for a displacement of 100 meters in 10 seconds are performed and discussed. This experiments validates our methodology of resolution. In Section 6, we conclude.

2 Model of the electrical vehicle

The model that we used here, comes from standard equations about electrical rotating machines with a continuous alimentation, see [2], [4], [6] for details.

On Figure 1, the standard traction links between the different components of the constitution of an electrical vehicle are presented. The model of this transmission chain has two parts: (i) an electrical part in association with battery, converter and motor; (ii) a 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 the rotating velocity of the motor).

The energy confined in the battery is regulated in the converter using the control parameter u, and the current, which is delivered to the motor, is depending on the following differential equation:

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

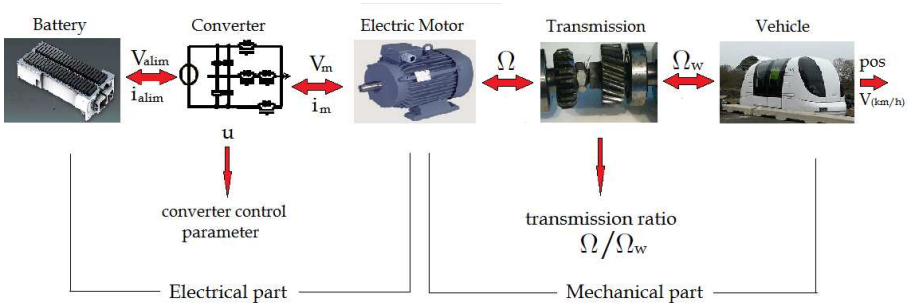

Fig. 1. Standard traction link

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 by the velocity of the rotor:

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

Note that there is a constant term in equation (2). Thus the vehicle always moves even if the velocity is null. We can correct that by changing the constant K_f by a function depending on velocity (for example, by returning 0 iff $\Omega(t) = 0$ and else the constant K_f; we can also introduce some continuous functions); this has no impact in our methodology.

In order to know the position of the vehicle, we introduce the following differential equation:

$$\frac{dpos(t)}{dt} = \frac{\Omega(t) \times r}{K_r}. \tag{3}$$

The linear velocity of the vehicle in km/h is given by: $V(t) = \frac{3.6 \times r}{K_r}\Omega(t)$.

The performance index is given by the following energy formula:

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

where E represents the electrical energy consummated during the displacement on the cycle $[0, t_f]$ (where t_f denotes the final time). 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 a Bang-Bang optimal control problem as follows:

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

The state variables are: (i) i_m, the current inside the motor; (ii) Ω, the angular velocity; (iii) pos, the position of the vehicle. 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 ($|i_m(t)| \leq 150A$). The other terms are fixed parameters and represent some physical things: $K_r = 10$, the coefficient of reduction; $\rho = 1.293kg/m^3$, the air density; $C_x = 0.4$, the aerodynamic coefficient; $S = 2m^2$, the area in the front of the vehicle; $r = 0.33m$, 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; $R_m = 0.03Ohms$, the inductor resistance; $L_m = 0.05$, the inductance of the rotor; $M = 250kg$, the mass; $g = 9.81$, the gravity constant; $J = M \times r^2/K_r^2$; $V_{alim} = 150V$, the battery voltage; $R_{bat} = 0.05Ohms$, the resistance of the battery. This problem is subject to the boundary conditions. The initial conditions are given by the 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 $I\!R^3$ but one or two variables could be free: for example, just the final position must be equal to $100m$; see Section 5.

This problem is hard to solve directly by using classical optimal control techniques. We tried to solve it by using the Pontryagin method based on shooting techniques, [9], [10]. 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 satisfactory solutions (even local one).

With direct methods, the procedure leads to consider large scale optimization problems 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. Thus, these methods are less suitable for certain special cases, including problems with a Bang-Bang structure yielding a large number of switching operations. In collaboration with Sager, we used the algorithm devel-

oped in [7] to solve Problem (5). This method based on direct multiple shooting technique, is dedicated to solve Mixed-integer optimal control problems (also in [7], methods for computing bounds are proposed); it works perfectly well applied on Problem (5), if the control is relaxed yielding $u(t) \in [-1, 1]$. However, for the moment, it does not provide correct results for a Bang-Bang control which is the case that we have to deal with on this application.

Dynamical programming could also be used here to try to solve Problem (5), see [10]. However, its application in our case will yield to deal with an important computational effort. That is why, we prefer not to pursue in this way of investigation. Thus, in this paper we propose another original methodology to solve this problem yielding to some small-sized discretized optimization problems which are solved using an exact Branch-and-Bound algorithm. This new approach provides solutions which are the exact or close to the exact ones for the discretized formulations which correspond to approximations of the global solutions of Problem (5).

3 Approximation using a current regulator system

First, we remark that the energy formula is only depending on the current and on the control. Therefore, it is just required to search the trajectory of the current which minimizes the consumption of energy. If we discretize all the interval of time $[0, t_f]$ by fixing the value of the control u; note that it is important to have very small steps about 10^{-4} to be able at least to control the current through the motor (else the value of the current change too roughly). If we discretize it directly, this will generate a very huge mixed integer non-linear global optimization problem which is, for the moment, very difficult to solve using direct methods of optimal control.

Our idea, which directly comes from the numerical simulation of the behavior of the vehicle, is to impose during some short sample time the value of the current inside the electrical motor of the vehicle. This is possible using the control parameter $u(t)$. Thus, imposing a reference current $iref$ yields to: (i) $i_m(t) > iref + \frac{\Delta}{2}$ involves $u(t) := -1$; (ii) $i_m(t) < iref - \frac{\Delta}{2}$ involves $u(t) := 1$; (iii) no change for u if $i_m(t) \in \left[iref - \frac{\Delta}{2}, iref + \frac{\Delta}{2}\right]$, see Figure 2.

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 of 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 a 10^{-4} time discretization stepsize. Remark that on this example, the time separating two switches is very small; in 2.10^{-3} seconds we have 5 switches, see Figure 2.

This technique is a way to construct a current regulator which is a first step before making a velocity regulator of an electrical vehicle. Hence, using this, the following differential system of equations can be written as follows:

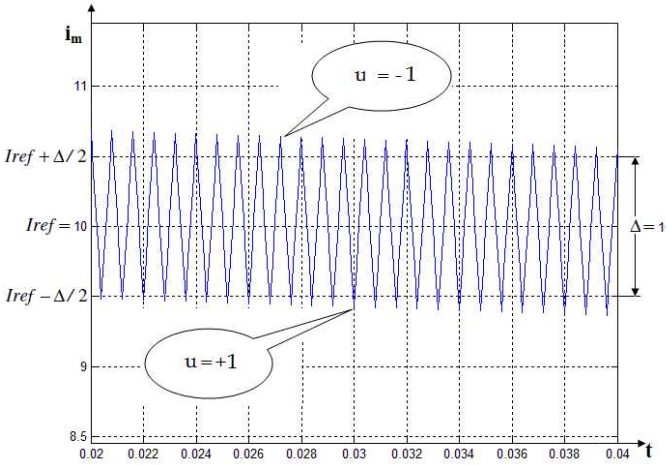

Fig. 2. Management principle by reference current

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

where t_0 is the initial time which is not necessary equal to 0 and $u(t^-)$ represents the previous value of the control which is equal to -1 or 1 (using an integrator with a step h, one has $t^- = t - h$).

This system of differential equations can be efficiently solved using a classical differential integrator such as for example *Euler, RK2, RK4*, with a step of time less than 10^{-4}; no numerical difficulty occurred when we solved this system VS using these three differential integrators with the double precision floating point number representation. The function $VS_{t_0,iref}(t)$ will compute all the values for $E(t), i_m(t), \Omega(t), pos(t)$, for all discretized times $t_i \in [t_0, t_f]$. Here, we are mainly interested by the final values of the state variables. Hence, we define the following function:

$$VSF(iref, t_0, t_f) := (E(t_f), i_m(t_f), \Omega(t_f), pos(t_f)) \in I\!R^4. \quad (7)$$

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 ($E^{t_0}, i_m^{t_0}$, Ω^{t_0}, pos^{t_0}).

One of 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:

$$\begin{cases} \min_{iref \in [-150,150]^P} \sum_{k=1}^{P} E_k \\ u.c. \\ \qquad (E_k, i_k, \Omega_k, pos_k) := VSF(iref_k, t_{k-1}, t_k) \\ \qquad (E_0, i_0, \Omega_0, pos_0) = (E^0, i_m^0, \Omega^0, pos^0) \in I\!R^4 \\ \qquad (i_P, \Omega_P, pos_P) \in \mathcal{T} \subseteq I\!R^3 \end{cases} \quad (8)$$

If the sample times are sufficiently small, Problem (8) will provide a good approximation of the initial Problem (5). Moreover, sample times of $1s$ seems to be generally sufficient, and so this approach will generate a few number of variables equal to 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 inside the vehicle.

4 A Branch-and-Bound based method

For the moment, we are not able to solve exactly Problem (8), thus we need to discretize also the possible values for the reference current: $iref \in D^P = \{-150, -150 + s, -150 + 2 \times s, \cdots, 150\}^P$; we will take values for s which divide exactly interval $[-150, 150]$. Therefore, the set of solution becomes finite and could be enumerated. Nevertheless, if we want to have a good approximation of the global optimum of Problem (8), 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 limit the exploration all the finite set of points.

4.1 Bounding Technique

For using such a Branch-and-Bound algorithm, we have to elaborate a technique to compute bounds for the four main parameters: E_k, i_k, Ω_k, pos_k over $IREF \subseteq D^P$ and for given initial and final times t_0 and t_f. In a previous stage, we compute 4 matrices: $M_E, M_i, M_\Omega, M_{pos}$ where the columns corresponds to values where $iref$ is fixed with $i_m^{t_0} = iref$ and the rows provide values for the entities when a velocity Ω^{t_0} is given (we discretize also the possible values of the velocity). We call *discrete box* X a subset of D^P.

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

st_V is a discretization step of the velocity values in km/h, Θ represents one of these three symbols $E,\ \Omega,\ pos$ and with $e_E = (1,0,0,0),\ e_\Omega = (0,0,1,0),\ e_{pos} = (0,0,0,1)$. Note, that we do not need to compute matrices for i_m because it will be taken equal to *iref*.

Thus, we obtain:

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

where $< .,. >$ denotes scalar product and $m_\Theta(i,j)$ depends on computing 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 $[t_{k-1}, t_k]$ when *iref* corresponds to the jth components of set $D = \{-150, -150+s, -150+2\times s, \cdots, 150\}$ with $i_m^{t_0} = iref$ and the ith row corresponds to the discretized value of the velocity, the other initial values are taken equal to 0; i.e., $E^{t_{k-1}} = pos^{t_{k-1}} = 0$.

When a discrete box *IREF* is considered, we can compute bounds for E, Ω and pos by calculating the integer sets $I = \{i_1, ..., i_n\}$ and $J = \{j_1, ..., j_m\}$ of the indices corresponding to the possible values of the velocity at the end of the previous sample and the possible values of *iref*. Thus, we have to compute the bounds which correspond to the minimal and maximal values of $m_E(i,j), m_\Omega(i,j), m_{pos}(i,j)$ for all $(i,j) \in I \times J$; we call this method EM for exact method. In order to obtain the final value for E and pos, we have to sum all the lower and upper bounds for each sample time $[t_{k-1}, t_k]$.

The remaining part of the Branch-and-Bound algorithm that we developed, is standard and uses the following classical principle: (i) subdivision into two (distinct) parts of the discrete box *IREF* (which represents the possible values for *iref*); (ii) the upper bound is updated by taking the middle of *IREF* if the constraints are satisfied and if its value is better than the previous one (we start with $+\infty$); (iii) in this work, we branch following the breadth-first or depth-first heuristic.

4.2 Heuristics Alternative

Our Branch-and-Bound code uses the data corresponding to the computations of the matrices M_Θ. The exact method EM for computing bounds, is expensive

in CPU-time, then we were interested by the following 4 heuristics $H1$, $H2$, $H3$ and $H4$:

- For $H1$, the values of each sub-matrices at the first row and first column $m_E(i_1, j_1), m_\Omega(i_1, j_1), m_{pos}(i_1, j_1)$ are taken as lower bounds and we take for 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 heuristic $H1$ for position. However considering the energy and the velocity, we compute the minimum value on the first row for lower bounds and 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: $\max_{j \in J} m_E(i_n, j), \max_{j \in J} m_\Omega(i_n, j), m_{pos}(i_n, j_m)$.
- For $H3$, we keep the same bounds as heuristic $H1$ for position and velocity. However, for the energy, the value of the sub-matrix at the last row and the first column $m_E(i_n, j_1)$ is taken as lower bound and respectively $m_E(i_1, j_m)$ as upper bound.

- For $H4$, we keep the same bounds as the heuristic $H2$ for position and velocity. Nevertheless, for the energy, as lower bound, we compute the minimum value on the last row and as upper bound, the maximum value on the first row:
  - Lower bounds: $\min_{j \in J} m_E(i_n, j), \min_{j \in J} m_\Omega(i_1, j), m_{pos}(i_1, j_1)$.
  - Upper bounds: $\max_{j \in J} m_E(i_1, j), \max_{j \in J} m_\Omega(i_n, j), m_{pos}(i_n, j_m)$.

Note that these values are possibly not lower or upper bounds. However, this is a first study and the efficiency of these heuristics provide information on functions of Problem (5).

The time for computing matrices depends on the sample time $t_k - t_{k-1} = \frac{t_f}{P}$, on st_V (step of discretization of the velocity), on the integer value of s (which sudivides interval $[-150, 150]$ of reference current), on the $RK4$ integrator step of time and on Δ. If we fix the $RK4$ integrator step to 10^{-4}, $st\_V = 0.1$ and $\Delta = 1$, the preprocessing time to compute matrices is proportional to $(t_k - t_{k-1})$ and inversely proportional to s.

4.3 Branch-and-Bound Algorithm

1: Initialization: Let $\mathcal{L} :=$ the initial domain D^P in which the minimum is searched, $Emin := \infty$ the upper bound of the minimum, $st\_Iref := s$ discretization step of reference current, $posmin := 0$ the initial position, $solmin := 0$ the initial solution, $st\_V :=$ the discretization step of the velocity (in km/h), $st\_tf := \frac{P}{t_f}$ denotes the length of the sample time.
2: Compute matrices $Posf, \quad Ef, \quad Vf$.

3: **while** $\mathcal{L} \neq \emptyset$ **do**
4: Extract an element X of $\mathcal{L}$.
5: **for** $i = 1$ to 2 **do**
6: Bisect X into two discrete subboxes $X^{(1)}$ and $X^{(2)}$ following the largest edge of the convex hull of X.
7: Compute lower bounds lbE, $lbpos$, lbV, and upper bounds $ubpos$, ubV of $X^{(i)}$ using $H1$, $H2$, $H3, H4$ or EM.
8: **if** $(ubpos \geq posf)$ and $(lbpos \leq posf)$ and $(lbE < Emin)$ **then**
9: $midi :=$ midpoint of the subbox $X^{(i)}$
10: $sol := [midi/st\_Iref] * st\_Iref$
11: Compute bounds $Esol, possol, Vsol$
12: **if** $(possol \geq posf)$ and $(Esol < Emin)$ **then**
13: $Emin := Esol$, $posmin := possol$, $solmin := sol$
14: **end if**
15: **if** $X^{(i)}$ is not reduced to a point **then**
16: Insert $X^{(i)}$ in $\mathcal{L}$
17: **end if**
18: **end if**
19: **end for**
20: **end while**
21: Results : $Emin$, $posmin$, $Vsol$.

where [.] returns the closest integer value of a real number.

The general structure of this algorithm can be explained schematically in four stages: (i) extraction, (ii) division, (iii) analysis and (iv) insertion. The list $\mathcal{L}$ can have a LIFO or FIFO management, for example.

In step 4.3 of the algorithm, the technique to bisect X into two discret subboxes proceeds by a selection of the component with the largest width of the convex hull of X and by bisecting it by the middle of this component. The loop **for** in line 4.3 is the main part of a Branch-and-Bound algorithm. Lines 4.3-4.3 concern the verification of constraint by computing bounds. This is the stage of analysis that uses one of the bounding techniques $H1, H2, H3, H4$ or EM. If the condition of line 4.3 holds, then the global minimum may occur in the discrete subbox $X^{(i)}$ which justifies its insertion into the list (line 4.3), otherwise this subbox is directly discarted.

Finding a good feasible solution consists in evaluating 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 $Emin$. This one is updated if it improves the current solution (line 4.3). The stopping criterion condition of the algorithm occurs when the whole list $\mathcal{L}$ is empty (line 4.3). This algorithm has an exponential complexity in time and in memory (as all discrete Branch-and-Bound algorithms).

5 Numerical Experiments

For all numerical tests, we used an Intel(R) Core(TM)2 Duo CPU 2.13GHz DELL laptop with 4GB of RAM, Windows7 64-bits operating system, and this first version of the code was done in MatLab 9. In order to compare the results in cases of variations on velocity constraints, we adopt the same profile for all simulations. Our method is then evaluated for a displacement of 100 meters and a cycle of time $t_f = 10$ seconds: $(i_m(0), \Omega(0), pos(0)) = (0, 0, 0)$; $(i_m(t_f), \Omega(t_f), pos(t_f)) \in \mathcal{T} = I\!R \times I\!R \times \{100\}$. We implemented our code with a LIFO and a FIFO management of $\mathcal{L}$. According to some preliminary numerical tests, it occurs that using FIFO heuristic the behavior of the Branch-and-Bound code is the most efficient; this also depends on our implementation of our code in MatLab. Therefore, all the following numerical experiments are performed using a Branch-and-Bound code where a FIFO management of $\mathcal{L}$ is taken into account.

5.1 Case without constraint on velocity

In Table 1, we compare all the heuristics $H1$ to $H4$ and the exact method EM with parameters $P = 5$ and $s = 1$. We obtain for the discretized problem, the exact solution $iref = (145,\ 86,\ 34,\ -16,\ -112)$ corresponding to the minimum energy $Emin(10) = 23,045$ J. Moreover, we have $pos(10) = 100.01$ m. These computations were performed for an integration step equal to 10^{-4} and $st\_V = 0.1\ km/h$. The pre-processing for computing matrices needed 659 seconds.

	CPU (s)	E_{min} (J)	$posf$ (m)	Vf (km/h)	$iref$ $(amps)$	$Iterations$
EM	3,217	23,045	100.01	13.20	(145, 86, 34, −16, −112)	12,511,713
$H1$	54	23,151	100.00	13.68	(149, 83, 21, 2, −116)	717,752
$H2$	473	23,054	100.00	11.72	(149, 86, 29, −17, −116)	2,201,152
$H3$	551	23,045	100.01	13.20	(145, 86, 34, −16, −112)	7,215,459
$H4$	1,524	23,045	100.01	13.20	(145, 86, 34, −16, −112)	7,232,821

Table 1. *Comparison of the heuristics for P=5, s=1.*

In Table 1, the solutions obtained by EM, $H1$, $H2$, $H3$ and $H4$ for the parameters $P = 5$ and $s = 1$ are presented. The CPU-time running is given in seconds; E_{min} is the minimum energy consumption given in joules; $posf$ is the distance performed by the vehicle in meters; Vf is the final velocity of the vehicle given in km/h; $iref$ gives the reference currents in amps. Last column gives the number of iterations required for the execution of the algorithm. The graphs corresponding to the optimal solution are given on Figure 3.

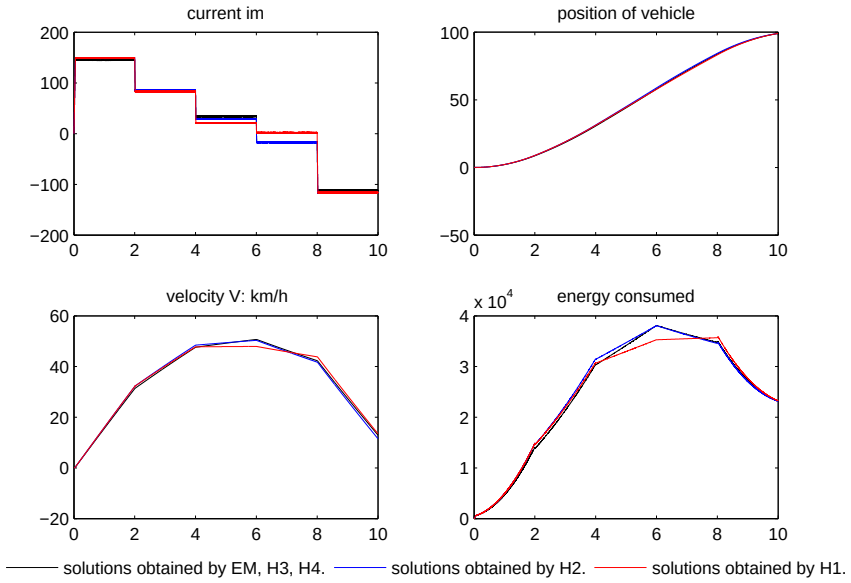

Fig. 3. Case: P=5; s=1; posf=100 m.

Note that from the solution obtained in Table 1 by EM, if we directly compute the characteristical values using RK4 (without using matrices), we obtain $\overline{E}min = 23,166\ J$ for a position $\overline{pos} = 99.04\ m$; the error comparing with results using the method based on matrices, is about 0.52% for the energy and 0.97% for the position.

Although the heuristics $H1$ and $H2$ give results quite close, they are not reliable for determining the exact solution; this case is a counter-example for them. Nevertheless, $H1$ and $H2$ can be used when we can settle for a less accurate result to reconcile computational time and solution quality. In this case, the heuristic $H3$ and $H4$ provide the same solution as the exact one EM; they are much more reliable, than $H1$ and $H2$.

In order to test the efficiency of the heuristics $H2, H3, H4$ and EM ($H1$ is discarded because it is not efficient enough), we performed 101 tests where the final position varies from $20m$ to $120m$ in increment of $1m$ (all other constraints are the same).

On Figures 4 and 5, the performance of heuristics $H2$, $H3$, $H4$ and the exact method EM are presented. These methods are compared by the number of iterations and the CPU time. The profile on Figure 5 shows an important gain in time for $H2, H3$ and $H4$ compared to the exact method EM.

In Table 2, we present comparisons between all our five methods (heuristics and the exact one). These comparisons are done using: (i) the percentage of

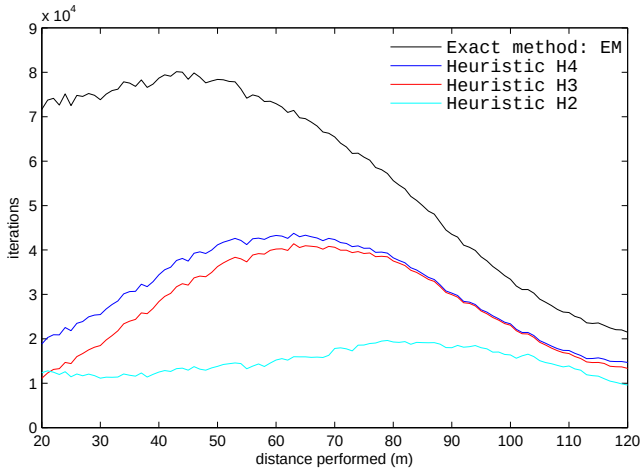

Fig. 4. Number of iterations, case: P=5, s=10.

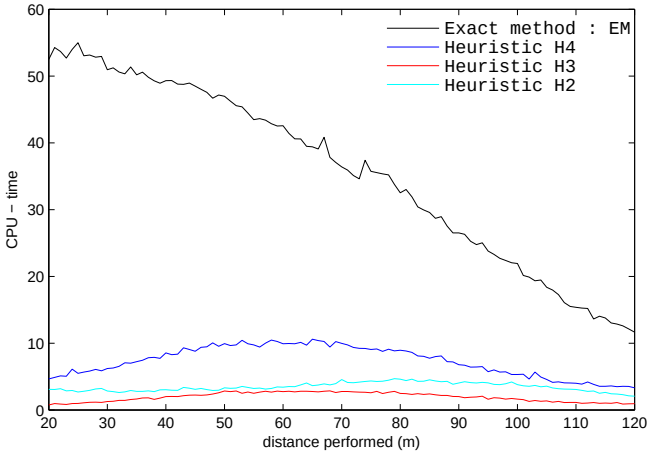

Fig. 5. CPU-time, case: P=5, s=10.

success on the solutions, (ii) the average CPU-time, and (iii) the maximum error on the solution compared to the exact one (computed with EM).

$Heuristics$	$H1$	$H2$	$H3$	$H4$	EM
$average\ CPU-time\ (s)$	0.18	3.44	1.92	7.31	35.59
$maximum\ error\ on\ energy\ (\%)$	4.45%	1.00%	0.69%	0%	//
$success\ (\%)$	0%	67%	97%	100%	//

Table 2. *Comparison of the heuristics.*

Comparing to EM and using $H3$, the average CPU time is divided by **18**. $H4$ seems to be completely reliable and gives in all cases the same results as using EM. Heuristic $H3$ is slightly less efficient than $H4$, but the maximum error committed is 0.69%. However, the computation time is divided by 4. Therefore, for all the following numerical tests we used only $H3$.

When we evaluate our algorithm for parameters $P = 10$ and $s = 10$ by using $H3$, we obtain the best solution provided so far in one day of computations: $iref^* = (150,\ 140,\ 90,\ 70,\ 30,\ 20,\ 20,\ -20,\ -80,\ -150)$, which corresponds to 8×10^8 iterations of the Branch-and-Bound algorithm; the corresponding minimum energy $Emin^*(10)$ is equal to $22,807J$ and position $pos^*(10)$ equals to $100.05m$. The computing time depends directly on parameters s and P, which is understandable for a Branch-and-Bound code where complexity is exponential in P: about $\left(\frac{300}{s}+1\right)^P$.

However, 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.

Using the solution obtained above with $H3$ (see Figure 3) $iref = (145, 86, 34, -16, -112)$ over a sampling intervals of $2s$, $(P = 5; s = 1)$, we then construct the following $iref = (145, 145, 86, 86, 34, 34, -16, -16, -112, -112)$ over the sampling intervals of $1s$, $(P = 10; s = 1)$. Spreading it on a maximum range of 40 amps, a discrete box is generated where its convex hull is: $\overline{IREF} = [125, 150] \times [125, 150] \times [66, 106] \times [66, 106] \times [14, 54] \times [14, 54] \times [-36, 4] \times [-36, 4] \times [-132, -92] \times [-132, -92]$. The solution will then be searched in $IREF$ the corresponding discrete box. Note that point $iref$ is a possible solution for this new problem.

In Table 3, we present the refined solutions generated after 3 iterative runs of a Branch-and-Bound algorithm where P is increased and s decreased by using previous solutions to define smallest domain of research.

With the improved solutions, we obtain a gain of 2.68% on the quality on the global optimum for 323 seconds of additional CPU-time. The curves on Figure 6 are done using the latest refined solution (last row in Table 3). By validating this solution directly using $RK4$, this provides an energy $\overline{E}min = 22,876\ J$, and a position $\overline{pos} = 98.71\ m$. The calculation error is about 1% for energy and 1.3% for position.

$Instance$	$iref$	E_{min} (J)	$posf$ (m)	Vf (km/h)	CPU (s)	$range$ $(amps)$	$Iter.$
$P = 5$, $s = 10$	$(150,\ 90,\ 30, -30, -110)$	$23,272$	100.24	11.31	2.65	300	$23,069$
$P = 10$, $s = 10$	$(150,\ 140,\ 110,\ 70,\ 30,$ $20, -10, -30, -90, -130)$	$22,813$	100.02	11.48	34	40	$264,406$
$P = 10$, $s = 5$	$(150,\ 145,\ 105,\ 65,\ 20,$ $20,\ 0, -20, -80, -135)$	$22,698$	100.03	12.94	93	20	$702,222$
$P = 10$, $s = 1$	$(150,\ 147,\ 104,\ 63,\ 21,$ $20,\ 0, -21, -80, -136)$	$22,648$	100.00	12.73	193	4	$1,410,307$

Table 3. *Table of refined solutions: free final velocity.*

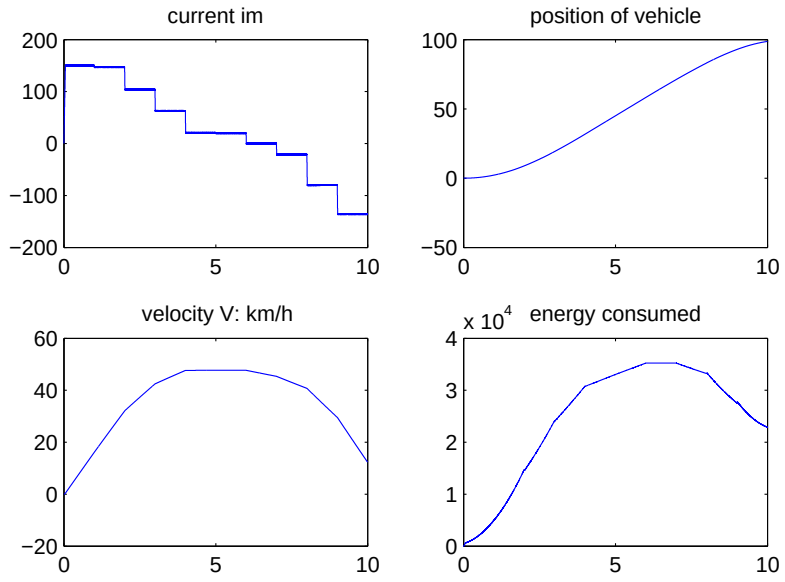

Fig. 6. Case: P=10; s=1; posf=100 m.

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 (about $3A$ every 10^{-3} seconds). Moreover, the curve of the energy decreases at the end of the cycle because this corresponds to a phase of recovery.

Note that the final velocity is not null, see Figure 6. However, our method can take this constraint into account. By setting : $(i_m(0), \Omega(0), pos(0)) = (0, 0, 0)$; $(i_m(t_f), \Omega(t_f), pos(t_f)) \in \mathcal{T} = \mathbb{R} \times \{0\} \times \{100\}$. The corresponding refined solutions are presented in Table 4.

$Instance$	$iref$	E_{min} (J)	$posf$ (m)	Vf (km/h)	CPU (s)	$range$ $(amps)$	$Iter.$
$P = 5$, $s = 10$	$(150,\ 110,\ 40, -70, -150)$	$26,517$	100.72	-1.17	0.24	300	$3,022$
$P = 10$, $s = 10$	$(150,\ 150,\ 130,\ 90,\ 30,\ 20, -50, -60, -150, -150)$	$25,646$	100.00	-0.89	8	40	$85,630$
$P = 10$, $s = 5$	$(150,\ 145,\ 130,\ 90,\ 35,\ 20, -50, -65, -140, -150)$	$25,199$	100.00	-0.10	12	20	$121,889$
$P = 10$, $s = 1$	$(150,\ 145,\ 131,\ 88,\ 36,\ 20, -50, -66, -138, -150)$	$25,156$	100.01	0.00	116	4	$1,077,629$

Table 4. *Table of refined solutions: null final velocity.*

We note that the minimum energy consumption for this case is about 10% greater than in the previous case without the constraint on the final velocity. The calculation of the latest refined solution of Table 4, by simulation using directly $RK4$, provides an energy $\overline{E}min = 25,568J$; a position $\overline{pos} = 98.83m$ and a final velocity $Vf = -0.34km/h$. The error is about 1.61% for energy, 1.17% for the position and $\pm 0.34km/h$ for velocity. Because the vehicle starts and ends with a velocity which must be equal to zero, it necessarily goes 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.

The solutions were obtained using a total CPU-time about 2 minutes which is more efficient than results obtained without considering this constraint on the final velocity.

5.2 Case with constraints on the velocity state variable and on the final velocity

Our Branch-and-Bound algorithm can also take into account the constraint on the velocity parameter. We simulate the same previous instances by adding a constraint on the state variable $\Omega(t)$ by setting $\Omega(t) \leq \frac{K_r}{3.6 \times r} \times 50$ which indicates that a velocity limit is given to perform the displacement (for example,

in agglomeration where the speed must not exceed $50km/h$). Similarly to the previous subsection, the final velocity Vf is fixed but in this case it has to be equal to 50 km/h. The boundary conditions are written $(i_m(0), \Omega(0), pos(0)) = (0, 0, 0)$; $(i_m(t_f), \Omega(t_f), pos(t_f)) \in \mathcal{T} = I\!R \times \{\frac{K_r}{3.6 \times r} \times 50\} \times \{100\}$.

The solutions obtained using successive runs and refinement of a Branch-and-Bound code, are reported in Table 5 and drawn on Figure 7.

Instance	*iref*	E_{min} (J)	*posf* (m)	*Vf* (km/h)	*CPU* (s)	*range* $(amps)$	*Iter.*
$P = 5$, $s = 10$	(120, 60, 50, 30, 40)	43,836	100.10	51.49	0.70	300	8,100
$P = 10$, $s = 10$	(120, 100, 80, 70, 50, 40, 30, 40, 30, 30)	42,495	100.06	50.24	36	40	341,281
$P = 10$, $s = 5$	(120, 95, 85, 65, 50, 50, 35, 30, 35, 25)	42,475	100.03	50.04	67	20	618,853
$P = 10$, $s = 1$	(121, 96, 83, 63, 48, 50, 37, 30, 34, 24)	42,151	100.24	50.02	93	4	804,511

Table 5. *Table of refined solutions: constraint on the velocity state and final velocity (*$50km/h$*).*

To validate the solution obtained in Table 5, we use directly $RK4$, and the difference is about 1.06% for the energy ($\overline{E}min = 41,702J$), about 1% for the position ($\overline{pos} = 99.24m$) and $\pm 0.4km/h$ for the velocity (final velocity $\overline{V}f = 49.62km/h$).

On Figure 7, the scheme representing the current takes its maximum value early in the cycle, decreases as far as reaching a steady value, around $30A$ in the last seconds. This allows maintaining the velocity at its desired value. That corresponds to the phase of steady state of the motor and there is no recovery of the electrical energy on this profile.

5.3 Some discussions

Considering a free final velocity case, the system mainly depending on the current, appears to have an identical behavior; i.e., decreasing curves with negative currents at the end of the cycle which corresponds to the phases of electrical energy recovery. Furthermore, with a same fixed time and for different values for the final position, the slopes of these curves representing the current are more or less important depending on the distance performed, see Figure 8. Nevertheless, this behavior of the optimal current is not general, see Figure 9.

On Figures 8 and 9, we plot the optimal solutions of the current *iref*. The different lines join points which are the values of the optimal current which is plotted in the center of its corresponding sample time of $2s$. These numerical tests are done for fixed final positions from $40m$ to $90m$, representing the different lines in Figures 8-9.

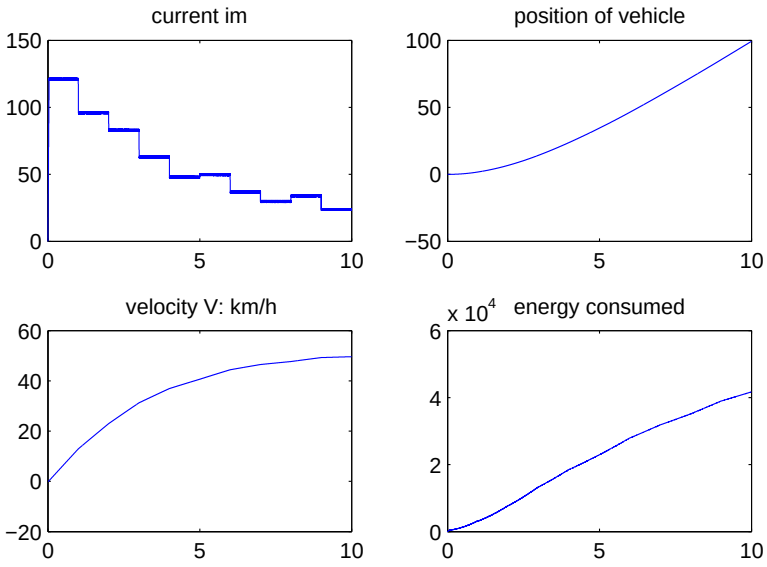

Fig. 7. Case: P=10; s=1; posf=100 m; $V(t) \leq 50$ km/h; $Vf = 50$ km/h.

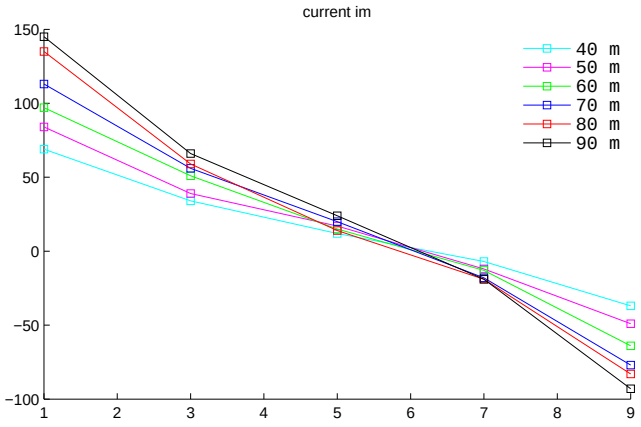

Fig. 8. Optimal current: free velocity.

With a non-zero final velocity and a limit on the speed, the curves of the current are decreasing with the growth of distance performed. This trend is reversed with increasing curves in the case of short distances, see Figure 9.

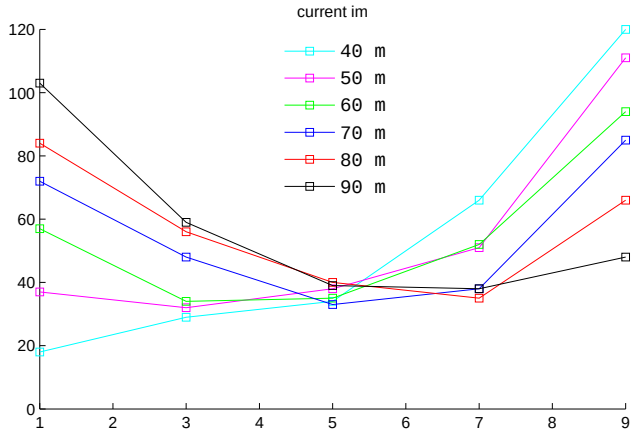

Fig. 9. Optimal current: with constraints on velocity.

5.4 A bi-criteria problem

We consider now a second criterion that corresponds to the maximization the distance performed. This yield to a bi-criteria optimal control problem. In this case, we add a constraint on the state variable $\Omega(t)$ in order to limit the velocity of the vehicle at $50km/h$. We then extend our methodology previously developed to construct the Pareto front presented on Figure 10. The results are obtained for a cycle of time $t_f = 10s$, with initial conditions $(i_m(0), \Omega(0)) = (0, 0)$ and final conditions $(i_m(t_f), V(t_f)) \in \mathcal{T} = I\!R \times \{50km/h\}$. The parameters of simulation are $P = 5$ and $s = 10$ and heuristic $H3$ is used. The total CPU-time for constructing the Pareto front is about 30 minutes, using the successive runs of refined solutions.

The two considered criteria, i.e., to minimize the energy consumed and to maximize the distance performed on a fixed cycle, are of conflicting type. By optimizing only one criteria without taking into account the other, we get the ideal point $(E^*_{min}, pos^*_{max}) = (31780J, 115.38m)$. For $E^*_{min} = 31780J$, the corresponding distance is $pos_* = 26.75m$. For $pos^*_{max} = 115.38m$, the corresponding consumed energy is $E_* = 47846J$.

To construct the Pareto front, see Figure 10, we discretized the values of position by step of $1m$ from pos_* to pos^*_{max}. The energy consumption is then

minimized by fixing iteratively the final position and by using our methodology based on approximations of Problem (5).

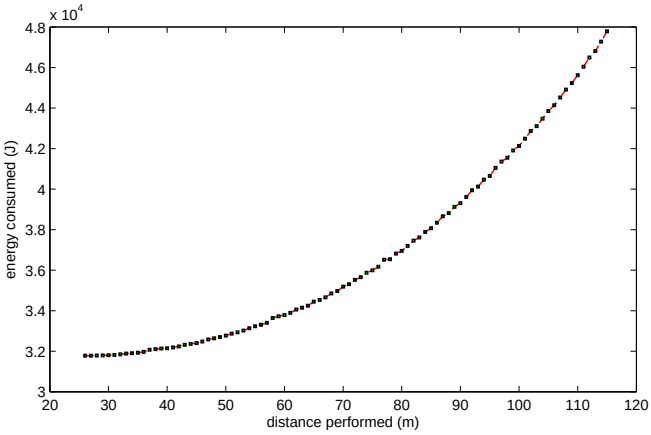

Fig. 10. Pareto front

In multiple criteria optimization, finding the best tradeoff used in Goal Programming method involves to minimize the value

$$\|(E(t_f, i_m, u), pos(t_f, \Omega)) - (E^*_{min}, pos^*_{max})\|_p$$

where $\|.\|_p$ is the norm of L^p. It is useful in this case to put the two criteria in the same size order. The solution is reached for: (i) $(37615J, 83m)$ with L^2 norm; (ii) $(36544J, 78m)$ with L^1 norm and, (ii) $(37892J, 84m)$ with L^∞ norm.

6 Conclusion

The energy management problem of an electrical vehicle can be formulated as an optimal control problem with a Bang-Bang control, which is actually difficult to solve using classical methods. In this paper, we show an original way based on discretization and a Branch-and-Bound method to solve an approximation of the optimal control problem. A methodology of resolution was derived in this paper and was validated on some numerical tests. These first numerical study tends to show the great efficiency of our approach to solve those difficult Bang-Bang optimal control problems. Some extensions and improvements are considered such as management of roads with slopes or using a regulator on velocity.

This concludes a first study on this subject yielding a methodology and a code to solve this kind of difficult Bang-Bang optimal control problem were done in MatLab.

References

1. 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.
2. J. Bonal, *Entrainements électriques vitesse variable,* ISBN 2-7430-0138-0, TEC&DOC, 1997.
3. W.R. Esposito, C.A. Floudas, *Deterministic Global Optimization in Nonlinear Optimal Control Problems*, Journal of Global Optimization 17, 97-126, 2000.
4. G. Lacroux, *Les Actionneurs électriques pour la robotique et les asservissements,* ISBN 2-85206-270-4, TEC&DOC Lavoisier, 1985.
5. 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.
6. S.A. Nasar, *Handbook of Electric Machines,* ISBN 0-07-045888-X, McGraw-Hill Book Company, 1987.
7. 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.
8. 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.
9. E. Trélat, *Contrôle optimal : théorie et applications,* Vuibert, Collection "Mathématiques Concrètes", 2005.
10. R. Vinter, *Optimal Control, Systems and Control: Foundations and Applications,* Birkhuser Boston, Inc, Boston, MA, 2000.