On Trajectory Optimization of an Electric Vehicle

Eligius M. T. Hendrix[1], Ana Maria A. C. Rocha[2(✉)], and Inmaculada García[1]

[1] Computer Architecture, Universidad de Málaga, 29080 Málaga, Spain
{eligius,igarciaf}@uma.es
[2] ALGORITMI Center, University of Minho, 4710-057 Braga, Portugal
arocha@dps.uminho.pt

Abstract. The efficient control of electrical vehicles may contribute to sustainable use of energy. In recent studies, a model has been analyzed and several algorithms based on branch and bound have been presented. In this work, we discuss a reformulated model on the control of an electric vehicle based on the minimization of the energy consumption during an imposed displacement. We will show that similar results can be obtained by applying standard software. Moreover, this paper shows that the specified control problem can be handled from a dynamic programming perspective.

Keywords: Electric vehicle · Energy optimization · Dynamic programming · Nonlinear optimization · Optimal control · Real-time control

1 Introduction

Efficient use of energy for transportation is of utmost importance. Efficient control of electric vehicles may lead to a more sustainable world. The literature on control of hybrid and electric vehicles has increased considerably over the last decade [6,10,11,14]. Literature on control typically focuses on continuous control using theory on the use of the Hamiltonian and co-states.

A simplified electric vehicle energy consumption model is presented in [7] and evaluated on standard driving cycles, whereas [15] presents a method to evaluate the real driving energy consumptions of electric vehicles. A simple electric vehicle energy consumption model that captures instantaneous braking energy regeneration as a function of the vehicle deceleration level can be found in [3]. A computationally efficient simulation model for estimating the energy consumption of electric vehicles is developed in [4]. In [13], a general formulation of energy-efficient driving of electric vehicles is presented, that covers several distinct scenarios and the most-adopted solution techniques.

In [8,9], a model for the control of a trajectory of a completely electric vehicle has been described. The final methodology suggested in those papers is a branch

© Springer Nature Switzerland AG 2019
S. Misra et al. (Eds.): ICCSA 2019, LNCS 11621, pp. 249–260, 2019.
https://doi.org/10.1007/978-3-030-24302-9_18

and bound (B&B) approach. This approach is counter-intuitive, as it does not make use of the dynamic characteristic of the problem. Similarly, a Mixed Integer Nonlinear Programming (MINLP) approach has been investigated in [12]. A more thorough mathematical analysis of the model can be found in [2]. As it does not feel that the problem has a multi-extremal character, the use of branch and bound does not seem appropriate. Our research question is whether standard software of implemented nonlinear optimization routines can reach a similar result.

To investigate the question, we first rewrite the model as an optimization problem in Sect. 2. From there we will investigate two possible approaches. A nonlinear optimization approach on a continuous relaxation of the problem in Sect. 3 and a state variable trajectory optimization as suggested in [9] is investigated in Sect. 4. Section 5 discusses a dynamic programming approach of the problem. Section 6 summarizes our findings.

2 Model

In the literature around this topic, the model to be studied is encapsulated in the convention of technical literature where subscripts are used for as well indices as indicators and small caps and capitals are mixed with greek letters. From the context, we first distinguish the technical data from the state and control variables. Therefore, we apply a notation following the convention of mathematical programming trying to distinguish between parameters and variables. Moreover, for the dynamics of the model, we will apply difference equations rather than differential equations based on a step size of δ seconds.

Indices

t Moment in time with δ second slots, $t = 0, \ldots, T$

Parameters

H Final control horizon in seconds
δ Time discretization slot, e.g. $\delta = 0.001\,\text{s}$.
T Number of periods (slots) in the horizon $T = \frac{H}{\delta}$
P Target position to be reached in control horizon
R Radius of the wheels, $R = 0.33\,\text{m}$.
B Resistance of the battery, $B = 0.05\,\Omega$
S Voltage of power supply, $S = 150\,\text{v}$
Tr Transmission coefficient motor to wheels, $Tr = 10$
C Resistance depending on air density, surface car and aerodynamics, $C = 0.517\,\text{N}$
L Inductance rotor, $L = 0.05\,\text{H}$
I Inductor resistance, $I = 0.03\,\Omega$
Q Coefficient motor torque, $Q = 0.27\,\text{Nm/A}$ (Newton-Meters per Ampere)
M Mass vehicle, $M = 250\,\text{kg}$
G Gravity constant, $G = 9.81\,\text{m/s}^2$
F Friction coefficient of the wheels, $F = 0.03$

Variables

$i_t \in [-150, 150]$ Induction (current) of the engine
ω_t Angular velocity in radians per second. So velocity $s_t = \frac{R}{Tr}\omega_t$ meter per second
$p_t \in [0, P]$ Position of the vehicle at time t
$u_t \in \{-1, 1\}$ Control, switch.

The control is of interest. One can switch very frequently, so one of the approaches is to consider a relaxation where $u_t \in [-1, 1]$. However, the most important in our approach is that we limit the real value of the current in simulating the dynamic system such that we keep the current $i_t \in [-150, 150]$.

The objective is given by the energy consumption E in Joules

$$E = \delta \sum_{t=0}^{T-1} (S \cdot u_t \cdot i_t + B \cdot u_t^2 \cdot i_t^2). \tag{1}$$

The dynamics is given by difference equations taking the time step size δ into account. The position of the vehicle at time t is given by

$$p_t = p_{t-1} + \delta s_t \tag{2}$$

and the induction of the engine at time t is modelled as

$$i_t = i_{t-1} + \delta \frac{S u_t - I i_{t-1} - Q\omega_{t-1}}{L}. \tag{3}$$

The dynamics of the angular velocity at time t is given by

$$\omega_t = \omega_{t-1} + \delta \frac{Tr}{R} \left(\frac{QTr}{RM} i_{t-1} - GF - \frac{C}{M} v_{t-1}^2 \right). \tag{4}$$

Based on this model, we can build a simulation program, which evaluates the energy consumption of a certain control rule to determine u_t. To keep the induction into boundaries, one can limit the actual control using bounds. Rewriting (3) and limiting $i_t \in [-150, 150]$ one can determine the limit parameter

$$\Delta_t = \frac{150L + (\delta I - L) i_{t-1} + \delta Q \omega_{t-1}}{\delta S}$$

and limit the values of the control to

$$u_t \in min\{-1, \operatorname{sgn}(\Delta_t) \min\{|\Delta_t|, 1\}, \max\{1, \operatorname{sgn}(\Delta_t) \min\{|\Delta_t|, 1\}$$

with $\operatorname{sgn}(x)$ being the sign of real number x. This means we can send a larger control to the engine, but the reaction is limited such that the induction is not too high in absolute value preventing the engine to be destroyed.

3 Nonlinear Optimization on Continuous Control

Our first approach discretizes the horizon H in a finite number of $n < T$ control slots with a global control continuous variable $v_k, k = 1, \ldots, n$ such that $u_t = v_k$ if moment t falls into control slot k. In this way, we can apply simulation based optimization, where the optimizing routine sends trials of vector $v = (v_1, \ldots, v_n)$

and the simulation program returns the energy $E(v)$ and the deviation from the target $g(v) = P - p_T$. Following the convention of mathematical programming, the constraint is written as $g(v) \leq 0$, so the function g tells us how far we are off target.

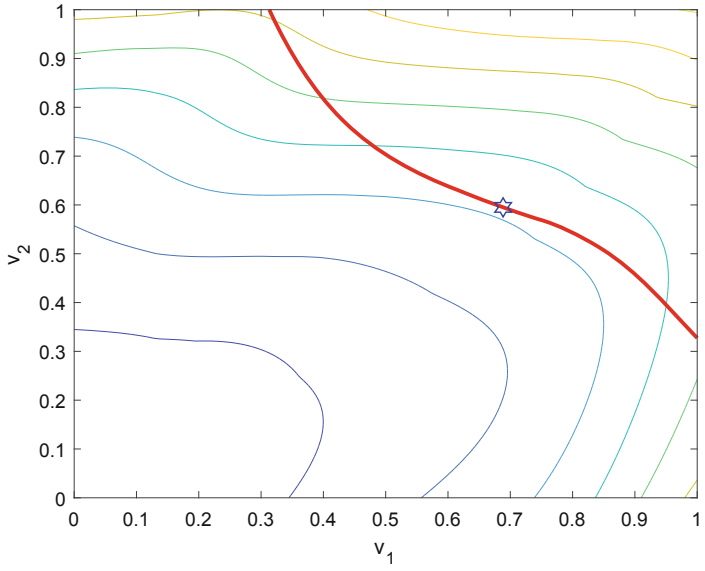

Fig. 1. Energy contours of $E(v_1, v_2)$ and the location constraint $g(v_1, v_2) \leq 0$ in red when optimizing over two slots. Using less energy in the left-down direction and feasible area in the right upper corner; the star gives the optimum found by NLP. (Color figure online)

Figure 1 provides a picture of the optimization problem considering only $n = 2$ slots. The contours of the total energy used $E(v)$ are given and the restriction on reaching the target position of $P = 100\,\text{m}$ is depicted in red. The functions expose a smooth behavior despite the evaluation is based on a finite step size simulation model. The value of the time step (time discretization slot) used in the simulation is $\delta = 0.001$ and the time horizon is $H = 10\,\text{s}$. The optimum is determined running the `fmincon` routine in Matlab version 2016b. The best point found is indicated for the two dimensional example. Of course, using only two slots is not energy friendly although the final target is reached. In the end the energy consumption reaches 31,589 J and the final speed is more than 10 m/s.

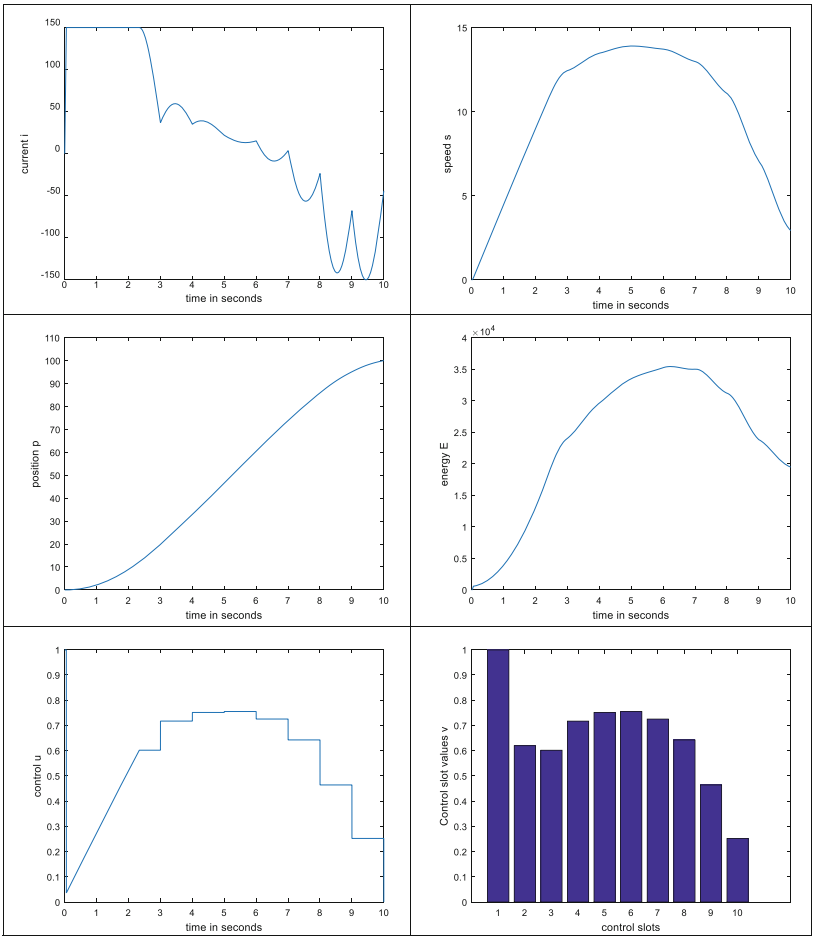

Fig. 2. Simulation result after optimizing v (result in bar graph) over 10 slots. The graphs give the development in time up to the horizon of the control u, speed s, position p, energy use and actual current i.

The optimum solution can be used as starting value for finding the optimum trajectory when using more slots. Actually, we run the optimization for $n = 5, 10, 20$ control slots which reduces the energy use $E(v)$ to 24,094, 19,476 and 19,322 J respectively.

Figure 2 shows the result for $n = 10$ control slots. The optimum trajectory is found within 13 s using as starting value the optimum from the $n = 2$ case and the resulting control vector v can be observed in the bar-graph at the right down corner. As said, the energy consumption went down to 19,476 J. One can observe that the internal control u follows the external control v most of the time apart from the first slot where the accelerator is pressed completely. The

vehicle after first accelerating regenerates energy by using the engine brake with a negative induction reaching a final speed of 3 m/s.

4 Optimizing the Reference Current as State Variable

A second way to consider the control problem in [9] is to include an internal regulator that tries to keep the value of the induction i close to a reference value r. This means if we follow a similar procedure of defining a number of n control slots, the procedure sets reference values $r_1, \ldots, r_n$ and the regulator tries to stay close to that. In [9] they suggest apply a threshold value ε and use as regulator

$$u_t = \left\{ \begin{array}{lll} 1 & \text{if} & i_{t-1} < r_t - \varepsilon \\ -1 & \text{if} & i_{t-1} > r_t + \varepsilon \\ u_{t-1} & \text{else} & \end{array} \right\}. \tag{5}$$

For the illustration, we built a simulator including regulator (5). We were not able to obtain exactly the same values as [9], but feeding the program with the

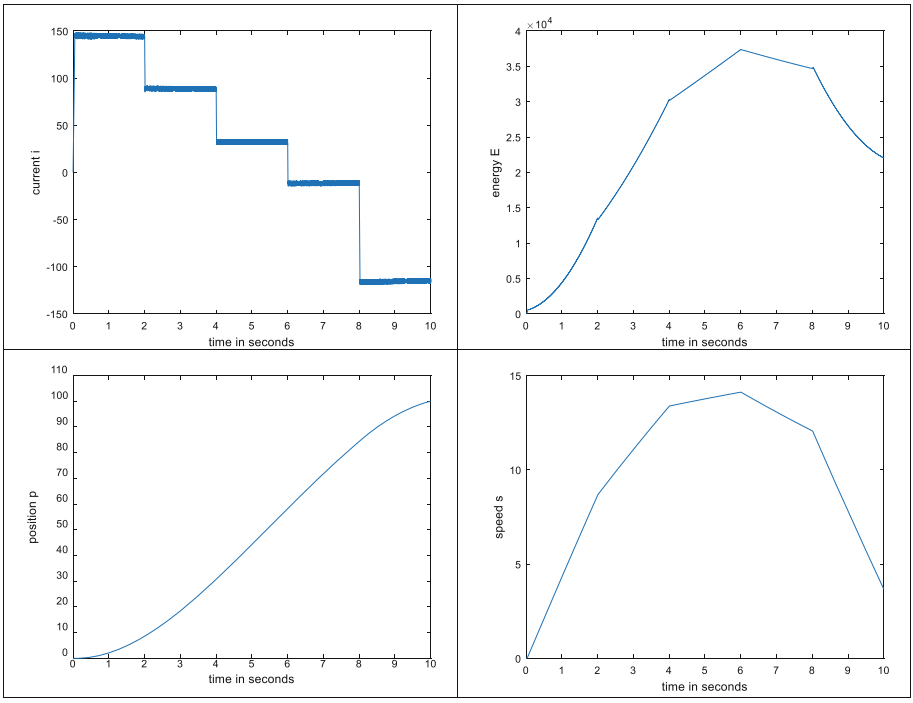

Fig. 3. Simulating current reference vector $r = (145, 90, 34, -10, -115)$ of 5 control slots during 10 s with time steps of $\delta = 0.0001$ s.

$n = 5$ slot reference value $r = (145, 90, 34, -10, -115)$ reaches the 100 m with an energy consumption of 22,039 J. The resulting speed, energy consumption and position are sketched in Fig. 3. One can observe the current fluctuating around the reference values $r_1, \ldots, r_n$ due to regulator (5).

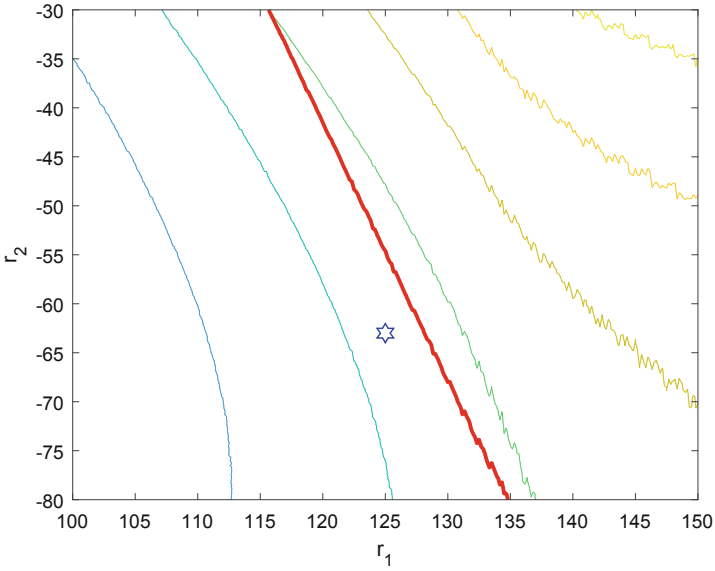

Fig. 4. Energy contours of $E(r_1, r_2)$ and location constraint $g(r_1, r_2) \leq 0$ in red optimizing over two slots. The star gives the "optimum" found by a genetic algorithm. Less energy use in the left-down direction and feasible area in the right upper corner. (Color figure online)

Optimizing the reference values for a finite number of slots becomes a challenge. Considering a continuous optimization approach for $n = 2$ slots provides us the energy contours in Fig. 4. At the left of the figure, the contours seem smooth, but at the right we observe strange behaviour. This phenomenon has been described in [5] and is due to simulation based optimization where `if..then..else` structures are included. This means that changing one coordinate of a coordinate r_k of reference value vector r a small step, does not change the control behaviour and the simulation returns more or less the same value for the energy used and position, i.e. the simulation does not change. We focus on this phenomenon by changing in the $n = 5$ dimensional reference vector r only the value of r_3 observing the development of the energy consumption $E(r)$ and the deviation $g(r) = P - p_T$ from the target of $P = 100$ m in Fig. 5.

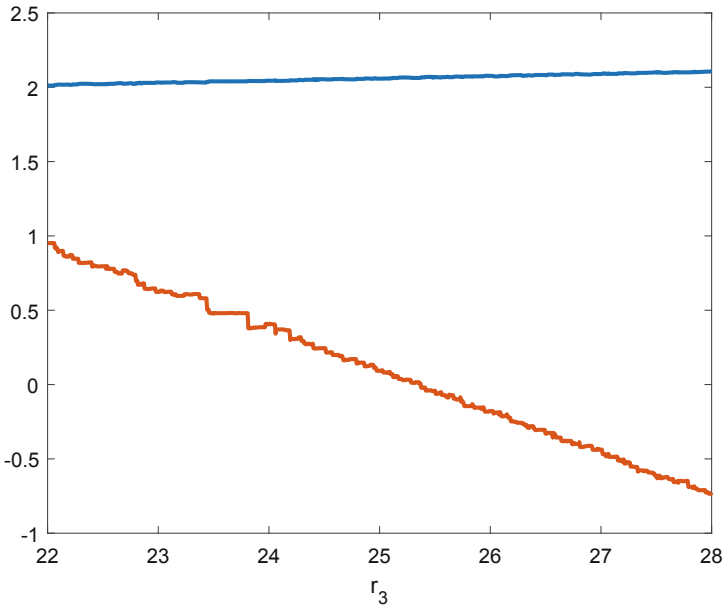

Fig. 5. Simulated energy use $E(r)$ in blue and deviation $g(r) = P - p_T$ in orange using regulator (5) as function of reference current r_3 of slot 3 keeping the other reference values constant. (Color figure online)

One can observe that the objective $E(r)$ and constraint function $g(r)$ are typically non-continuous. This means that software on nonlinear optimization including pattern search methods have a difficulty to find the optimum. [5] also provide a smoothing method to make the function continuous, although computationally this may lead to bad conditioned problems with high and low derivatives. [9] discretized the n dimensional search space and looked for methods to reduce (bound) the feasible area. Such a method is highly non-attractive, as the number of combinations to consider increases exponentially with the number of slots n.

Although the problem from a smoothed view does not have local non-global optima, one can run population based algorithms. The optimization toolbox of Matlab has several variants of genetic algorithms and particle swarm algorithms available. As the optimum is also relatively flat in terms of objective and constraint (see Fig. 4), in our experience, the algorithms take a lot of time (hours) to get close to the optimum. For dimension $n = 2$ both standard algorithms take minutes. Moreover, the best point found as depicted in Fig. 4, seems far off of the optimum in the space of the reference value.

We have seen so far, that a continuous relaxation of the control can be handled by standard software even when the number of control slots grows. For the second method, where one optimizes reference values of the induction, the discrete control rule for u turns the problem in a non-continuous optimization

problem creating a hard to optimize problem. However, if we are able to create a method that is linear in the slots, perhaps the optimization can be handled in reasonable time.

5 A DP Procedure to Derive an Optimal Control Rule

In their findings, [9] report that for this system, the change in reference value $r_k - r_{k-1}$ does not influence the search for the best solution. This finding creates the possibility to consider dynamic programming (DP) with only two state variables, i.e. the speed s_k and position p_k at the beginning of a control slot. Enumerating the possibilities for the induction reference value r_k over a discrete grid leads to a procedure which is linear in the number of control slots. We will illustrate that.

The dynamic programming procedure determines in fact the optimal trajectories for a complete grid of state values (k, s, p), i.e. the slot number, speed and current position. In fact, the procedure also provides an answer to the control when we deviate from the optimal trajectory. Following the DP principle of Bellman [1] for a finite horizon dynamic system we can build a valuation of the state space according to the relation

$$V_{k-1}(s,p) = min_r[E(r,s,p,\frac{H}{n}) + V_k(\Phi(r,s,p,\frac{H}{n}))], \tag{6}$$

where in our system we should define $E(r, s, p, \tau)$ as the energy use when using reference value r from starting position s and p during a period of τ seconds. Similarly, function $\Phi(r, s, p, \tau) : \mathbb{R}^4 \rightarrow \mathbb{R}^2$ describes the transition function providing the state (speed and position) we arrive at using the same arguments (r, s, p) during τ seconds.

If one is able to compute the so-called value function V for all possible state values, one can also retrieve the optimal control value for the reference induction $r_k(s, p)$ from any state according to

$$r_k(s,p) = \text{argmin}_r[E(r,s,p,\frac{H}{n}) + V_k(\Phi(r,s,p,\frac{H}{n}))]. \tag{7}$$

In theory, starting from $r_1(0, 0)$ we can find the optimal trajectory for the reference control r following (7). For a practical implementation, we have to define a grid in the state space (s, p) for each control slot k and follow the recursion (6). One of the challenges to get this working is that the transformation (motion) function Φ will not take us to another grid point, such that interpolation or extrapolation is required. Another challenge is that in fact, we are only interested in those state values where we can still reach the target distance P. This can be formalised by identifying those values (s, p) at the beginning of control slot k for which

$$\Phi_2(150, s, p, \frac{n-k}{n}H) \geq P, \tag{8}$$

i.e. a maximum reference current $r = 150$ from the current speed s and position p can still lead us to the target P (second element of Φ) given the rest of the time

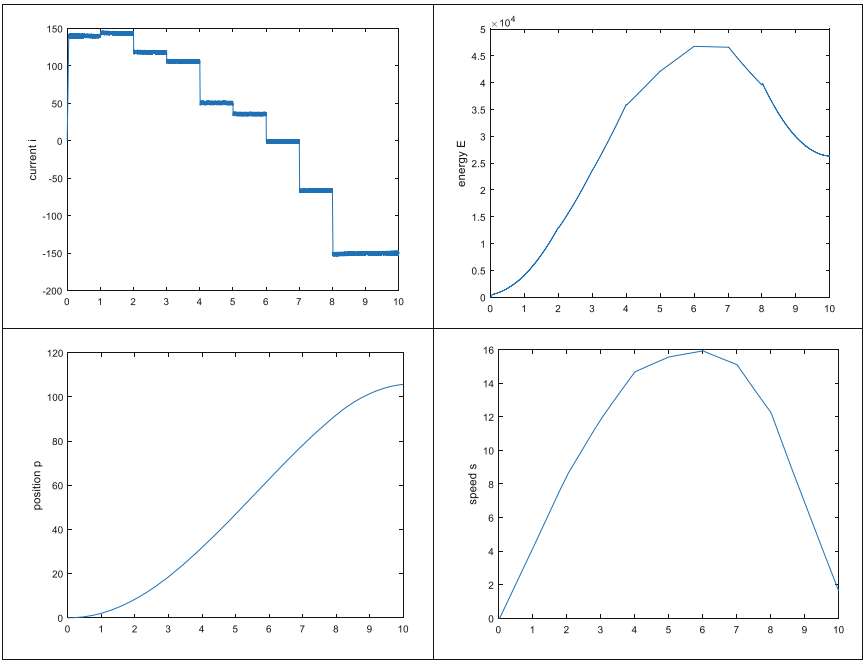

Fig. 6. Simulation result from DP reference induction values.

left. The state space can be bound on that providing lower and upper bounds $[\underline{s}_k, \overline{s}_k]$ and $[\underline{p}_k, \overline{p}_k]$. Basically, we cannot avoid infeasible combinations, as (8) is in fact a two-dimensional surface. However, maximum values $(\overline{s}_k, \overline{p}_k)$ can be generated by providing the maximum reference induction $r = 150$ from the start, i.e.

$$(\overline{s}_k, \overline{p}_k) = \Phi(150, 0, 0, \frac{k}{n}H). \tag{9}$$

In the implementation we used a grid of 15 equally distributed points over the ranges $[\underline{s}_k, \overline{s}_k]$ and $[\underline{p}_k, \overline{p}_k]$ for each control slot k. For the control variable r_k we used a grid of 301 points within $[-150, 150]$, such that the mesh size between two trials is 1, as suggested in [9], i.e. r_k has an integer value. Notice that for the evaluation of values for r, we can also find lower bounds $\underline{r}_k(s, p)$ using the monotonicity in the reference value r of the transition function Φ. For this, we run the possibilities from high to low and create a `while` construction such that as soon as a reference value provides infeasible positions, we reached a minimum reference value $\underline{r}_k(s, p)$ for the corresponding grid point.

We implemented the procedure in Matlab 2016b and run the dynamic programming recursion for $n = 10$ control slots. The computational time was 4 min and 11 s to reach a grid $r_k(s, p)$ of 10×15^2 control values. Starting from initial state $(s, p) = (0, 0)$, a simulation was run for the 10 control slots using small time steps of $\delta = 0.0001$. The result can be observed in Fig. 6 in terms of energy

use, speed, induction and position. The final energy use is estimated on 26,299 J and strange enough the reference trajectory reaches a position of 105.7 m.

It may be clear that in contrast to the suggested enumeration and bounding method in [9], the computation is not exponential but linear in the number of time slots n. However, the final implementation requires further fine-tuning to reach a more exact target value.

6 Conclusions

This paper reformulated a model in literature on the control of an electric vehicle as an optimization problem. Relaxing the integrality of the control, provides an optimization problem which can be handled by standard nonlinear optimization software following an optimization-simulation approach. The optimization of reference values for one of the state variables (induction) provides a non-continuous optimization problem where bounding can be used to reduce the effect of the exponential enumeration in number of control slots. We have shown that the same control vision can be handled by a dynamic programming approach. The challenge for that method is to deal with the feasibility of candidate solutions. However, the computational time is linear in the number of time slots.

Acknowledgments. This paper has been supported by The Spanish Ministry (RTI2018-095993) in part financed by the European Regional Development Fund (ERDF) and by FCT – Fundação para a Ciência e Tecnologia within the Project Scope: UID/CEC/00319/2019.

References

1. Bellman, R.: Dynamic Programming. Princeton University Press, Princeton (1957)
2. Cots, O.: Geometric and numerical methods for a state constrained minimum time control problem of an electric vehicle. ESAIM: COCV **23**(4), 1715–1749 (2017)
3. Fiori, C., Ahn, K., Rakha, H.A.: Power-based electric vehicle energy consumption model: model development and validation. Appl. Energy **168**, 257–268 (2016)
4. Genikomsakis, K.N., Mitrentsis, G.: A computationally efficient simulation model for estimating energy consumption of electric vehicles in the context of route planning applications. Transp. Res. Part D Transport Environ. **50**, 98–118 (2017)
5. Hendrix, E.M.T., Olieman, N.J.: The smoothed Monte Carlo method in robustness optimization. Optim. Methods Softw. **23**(5), 717–729 (2008)
6. Lei, F., Bai, Y., Zhu, W., Liu, J.: A novel approach for electric powertrain optimization considering vehicle power performance, energy consumption and ride comfort. Energy **167**, 1040–1050 (2019)
7. Luin, B., Petelin, S., Al-Mansour, F.: Microsimulation of electric vehicle energy consumption. Energy **174**, 24–32 (2019)
8. Merakeb, A., Messine, F.: Toward global minimum solutions for the problem of the energy consumption of an electrical vehicle. In: Cafiery, S., et al. (eds.) Proceedings of the Toulouse Global Optimization Workshop, pp. 85–88. ENSEEIHT (2010)
9. Merakeb, A., Messine, F., Aidéne, M.: A branch and bound algorithm for minimizing the energy consumption of an electrical vehicle. 4OR **12**(3), 261–283 (2014)

10. Onori, S., Serrao, L., Rizzoni, G.: Hybrid Electric Vehicles: Energy Management Strategies. Springer, London (2016). https://doi.org/10.1007/978-1-4471-6781-5
11. Sabri, M., Danapalasingam, K., Rahmat, M.: A review on hybrid electric vehicles architecture and energy management strategies. Renew. Sustain. Energy Rev. **53**, 1433–1442 (2016)
12. Sager, S., Claeys, M., Messine, F.: Efficient upper and lower bounds for global mixed-integer optimal control. J. Glob. Optim. **61**(4), 721–743 (2015)
13. Sciarretta, A., De Nunzio, G., Ojeda, L.L.: Optimal ecodriving control: energy-efficient driving of road vehicles as an optimal control problem. IEEE Control Syst. Mag. **35**(5), 71–90 (2015)
14. Sciarretta, A., Guzzella, L.: Control of hybrid electric vehicles. IEEE Control Syst. Mag. **27**(2), 60–70 (2007)
15. Yuan, X., Zhang, C., Hong, G., Huang, X., Li, L.: Method for evaluating the real-world driving energy consumptions of electric vehicles. Energy **141**, 1955–1968 (2017)