OPTIMAL CONTROL APPLICATIONS AND METHODS
Optim. Control Appl. Meth. 2016; **37**:537–555
Published online 23 September 2015 in Wiley Online Library (wileyonlinelibrary.com). DOI: 10.1002/oca.2210

Online energy management for a solar car using pseudospectral methods for optimal control

Enrique Guerrero Merino*,† and Manuel A. Duarte-Mermoud

Electrical Engineering Department, University of Chile. Santiago, Chile

SUMMARY

This paper deals with the problem of energy planning for a solar car race. Therein, an electric vehicle powered only by means of a battery pack and the sun's energy must cross usually long distances in several days, facing disturbances along the road such as road slopes, winds, cloudy weather, and vehicle failures. Because the battery pack energy is insufficient to end the race, teams must be very careful with their energy consumption planning, so as to drive as fast as possible to reach the goal in minimum time without emptying the battery pack. The proposed methodology consists of a real-time implementation that measures the current state of the vehicle and then executes a quick energy planning for the long term based on the available battery energy and current solar radiation forecast, afterwards executing a continuous optimal control problem stage for the short term solved by means of pseudospectral methods. This last inclusion allows for fast calculation of trajectory updates while keeping modeling detail. Copyright © 2015 John Wiley & Sons, Ltd.

Received 24 September 2013; Revised 11 August 2015; Accepted 17 August 2015

KEY WORDS: solar car; Radau pseudospectral optimal control; real-time optimal control; race

1. INTRODUCTION

The World Solar Challenge is a worldwide known event that congregates every two years teams from all over the world that wish to test their solar vehicles. The goal of the race is crossing Australia from north to south, which sums a total running distance of 3000 km of highway, using only a battery pack with a certain initial load established in the event rules and the energy given by the sun during the race execution. Due to its extension and the exacting energy constraints, a rational decision about how much energy to use, how and when, becomes necessary, making the best use of the available means. It is natural, then, the use of schemes based on parametric optimization and optimal control. It is necessary, on the other hand, to count with an adequate modeling of the vehicle and its components, so as to accurately estimate its energy consumption.

A solar car is a very complex system, but this paper is mainly focused on its primary subsystems: a solar energy conversion system, responsible for transforming the solar radiation into electricity; the battery pack; and the powertrain, consisting of standard mechanical components (suspension, steering, wheels, and brakes) and an electric motor. A complete monitoring system is in charge of estimating important variables such as the battery state of charge (SoC) and all circulating currents and voltages on the electric circuit, as well as temperatures and other important data concerning the different car devices. Consequently, the problem is to drive this given solar vehicle to the finish line in minimum time, given the real-energy availability and the technical and regulatory constraints. To achieve this, several different strategies have been successfully tried. Researchers have already developed analytical methods for advising the best driving strategies under different scenarios [1–3], resorting to Pontryagin's maximum principle [4, 5]. Studies have been made, which include most

*Correspondence to: Enrique Guerrero Merino, Av. Tupper 2007, Casilla 412-3, Santiago, Chile.
†E-mail: enrique.guerrero.merino@gmail.com

kinds of disturbances into account, such as slopes and cloudy days, spatially varying radiation [6], and incorporating complex battery models. While it is true that the developed analytic methods provide powerful tools for strategy planning, the varying conditions that are faced among different teams, races, and route segments makes it very necessary to include numerical optimization methods for real planning. On the other hand, the great complexity some case studies achieve makes them also necessary. Among the numerical approaches available on the literature, stochastic solutions can be found [1, 7], which stand out in their modeling simplicity, as well as deterministic ones which resort to different numerical methods. Some of them make use of indirect shooting methods to solve the maximum principle conditions [8]; others have preferred to iterate on a single constant speed for the whole race [9, 10] using simulation and optimization techniques separately, and some others have preferred to use dynamic programming adaptations for this purpose [7, 8, 11]. Finally, other authors have based their planning on the empirical knowledge of the vehicle's energy consumption and have used very simple tools for planning the energy budget of the vehicle [12, 13].

The proposed work coincides with [7, 8, 12, 13] in pointing out the great importance of the solar resource as a limiting factor of the vehicle's performance but, at the same time, as a facilitator and simplifier on the calculation strategy to implement. This is how the proposed scheme is based primarily on estimating the future available energy and afterwards only making the most urgent calculations (i.e., the control decisions to apply in the current instant). The schemes proposed in [12, 13] have great simplicity and automation because they are based on empirical knowledge of the vehicle, in a fashion that each energy consumption has a cruising speed associated through a look-up table. The present work, however, proposes to insist in that dynamics of the vehicle vary slow enough to sacrifice a small amount to computational speed in exchange for a greater level of detail that allows team members to take the expected evolution of the relevant variables into account, allowing for detection of possible failures or disturbance effects. Instead of the look-up table revision of those references, the proposed method includes an optimal control stage based on pseudospectral methods, not found in the consulted literature. This stage allows for incorporating dynamic detail of the car's modeling while using less computing resources as in a simulator–optimizer scheme based on Runge–Kutta and parametric optimization schemes [14].

The document is organized as follows. Section 2 briefly describes the car's modeling, stating all the effects and assumptions included in the present paper. Section 3 describes the proposed algorithm, discussing relevant inputs and outputs on its different stages. Section 4 afterwards shows simulation results showing the algorithm's performance in both normal and abnormal operating conditions and discusses the goodness of its reaction against disturbances. In Section 5, a comparison is made between the proposed algorithm and other techniques found on the literature. Finally, Section 6 outlines the most relevant conclusions and future work to continue improving the proposed algorithm.

2. CAR MODELING

The car modeling will be divided into two parts: first, the mechanical aspects will be presented, and then the electrical part will be described.

(a) Mechanical forces

In this paper, we shall consider the effects of the opposing forces applied to the vehicle. Namely, those are the rolling resistance, aerodynamic drag, and gradient force [15]. The vehicle's movement is assumed to occur in one dimension. [‡] Solar cars are thought for highway traffic, and therefore, a smooth path will be considered, and no wheel slip, which is also a reasonable assumption for rainy days at the speed the vehicle drives. The present model is based on that found in [3]. If the car speed is v and the applied force on the rear axle is F, the speed's rate of change is given by

[‡]Most of the World Solar Challenge event is ran this way, so neither tracking nor track curvatures are considered. In case of difference, the fast feedback characteristic of the algorithm is to compensate any mismatch in the driver's behavior, assuming that the direction will be kept straight on the following instants.

DOI: 10.1002/oca

Table I. Mechanical forces parameters: meaning and values.

Symbol	Meaning	Value
A	Vehicle front area	0.96 $[m^2]$
C_D	Drag coefficient	0.08 [–]
C_{r1}	First rolling resistance coefficient	0.008 [–]
C_{r2}	Second rolling resistance coefficient	$1.789 \cdot 10^{-4}$ $[s/m]$
m	Total mass (pilot+vehicle)	302 $[kg]$
n_w	Number of wheels	3 [–]
ρ_a	Air density	1.2043 $[kg/m^2]$

$$\frac{dv}{dt} = \frac{1}{m}(F - F_{ae}(v) - F_g(r) - F_r(v, r)), \tag{1}$$

where $F_{ae}(v)$, $F_g(r)$, and $F_r(v, r)$ represent the aerodynamic resistance at the speed v, the gradient force at position r, and the rolling friction at the same speed and position. Their expressions are given by

$$\begin{aligned} F_{ae}(v) &= \frac{1}{2} C_D \rho_a A (v + v_w)^2, \\ F_g(r) &= mg \sin(\theta_r), \\ F_r(v, r) &= mg(C_{r1} \text{sign}(v) + C_{r2} v) \cos(\theta_r) \end{aligned} \tag{2}$$

Parameter meanings and values are given in Table I.

The presented work takes road slopes θ_r into account, treating them as known. This information can be easily estimated from road data available from maps. A useful tool for these means can be Google Earth[§] ®. Therefore, it is adequate to include a dynamic equation for the travelled distance of the vehicle.

$$\frac{dr}{dt} = v. \tag{3}$$

It is important to know that wind speed v_w appears as an unknown disturbance. Due to this unawareness, the proposed algorithm treats this term as zero.

(b) Electric system

A schematic view of the vehicle's electric system is shown in Figure 1.

The electric system of the car is very complex. Modeling it in detail can be very cumbersome and tremendously increases the calculation time required to perform an iteration of the algorithm, with arguable benefits from doing it. Therefore, the solar panel's behavior is approximated considering a static efficiency, $\eta_{cells} = 0.15$. On the other hand, the high efficiency of the maximum power point tracker system (MPPT) and battery pack suggests it is a good approximation to consider them as perfectly efficient. However, the DC Brushless motor used deserves more attention. In [3], a loss model of the electric motor used in the vehicle is obtained, allowing therefore an expression for the axle power discussed on the previous section to be built. If P_{in} is the power input to the motor inverter and the current vehicle speed is v, it holds that

$$P_{in}(F, v) = Fv + k_0 v + k_1 |F| v + k_2 F^2. \tag{4}$$

[§]The tool was used only for its availability and ease of usage. In the case of the race under consideration, the slope variation is not too high, and most of the race is carried out in large open areas, where satellite estimations are expected to be more accurate as, for example, steep hills or forest areas.

DOI: 10.1002/oca

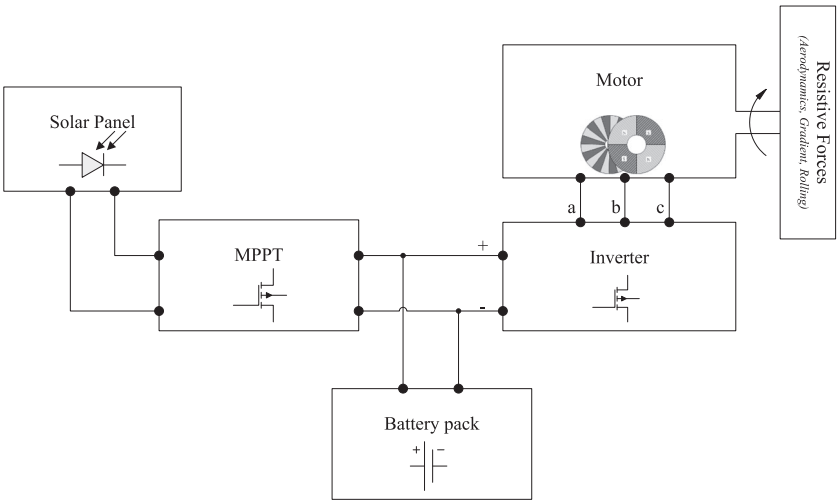

Figure 1. A typical solar car electric system.

Table II. Motor losses model parameters.

Parameter	Value
k_0	0.059(N)
k_1	0.043 (−)
k_2	0.00785(s/kg)

The values for the parameters k_0, k_1, and k_2 are given in Table II. In order to obtain a smooth model, the absolute value and sign functions are approximated as

$$\text{sign}(x) \approx \tanh\left(\frac{x}{5N}\right), \quad |x| \approx \left(\frac{1}{N}\log(2\cosh(Nx))\right), \tag{5}$$

with $N = 0.1$.

According to Figure 1, the electrical system is actually an energy-transferring system. If the available energy in batteries is E_b and $P_s(t)$ is the net solar energy income, then it holds that

$$\frac{dE_b}{dt} = P_s(t) - P_{in}. \tag{6}$$

It is important to include this variable in the model of the car due to the limited capacity of the batteries.

Summarizing, the complete model of the car used in this paper corresponds to

$$\begin{aligned}
\frac{dv}{dt} &= \frac{1}{m}\left(F - \frac{1}{2}C_D\rho_a A v^2 - mg\sin(\theta_r) - mg(C_{r1}\tanh\left(\frac{v}{5N}\right) + C_{r2}v)\cos(\theta_r)\right)\\
\frac{dr}{dt} &= v\\
\frac{dE_b}{dt} &= P_s(t) - P_{in}(F, v).
\end{aligned} \tag{7}$$

Adding the operational limits for batteries and inverter, the regulatory speed limits and the initial and terminal conditions for the traveled distance and battery energy, we have that if t_0 and t_f are the start and finish times of the race,

DOI: 10.1002/oca

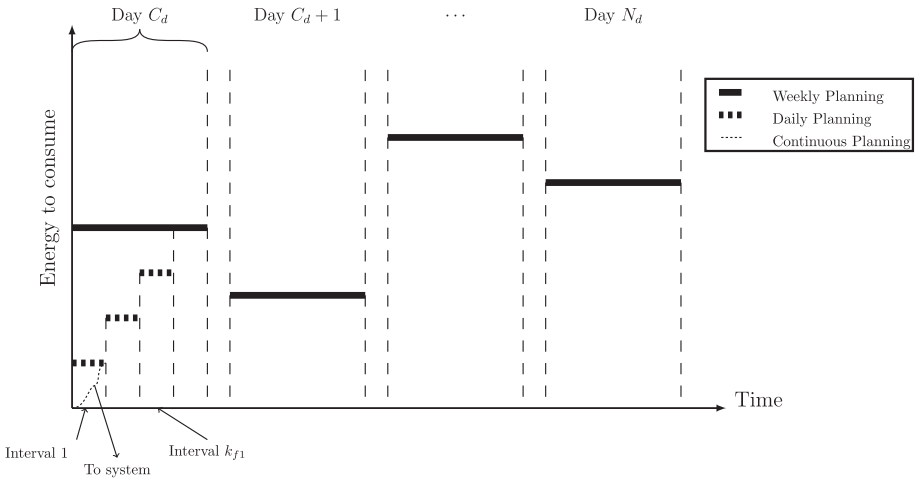

Figure 2. Summarized description of the proposed algorithm.

$$\begin{array}{rcl} 0 & \leqslant v(t) \leqslant & v_{(max)}, \quad \forall t \in [t_0, t_f] \\ E_{b(min)} & \leqslant E_b(t) \leqslant & E_{b(max)}, \quad \forall t \in [t_0, t_f] \\ F_{(min)} & \leqslant F(t) \leqslant & F_{(max)}, \quad \forall t \in [t_0, t_f] \end{array} \tag{8}$$

3. PROPOSED METHODOLOGY

If we were about to solve the optimal control problem of getting to the finish line in minimum time and we wanted to add enough detail, we would soon find out many difficulties that arise with this kind of treatments: first of all, the time scales of the different dynamic variables (for example, the speed) are orders of magnitude faster than the whole time horizon these kind of races use to take to be finished, which is in the order of several days. Because numerical methods for solving continuous-time optimal control problems resort themselves to parametrizations of the control and/or state variables and then solve a nonlinear programming problem it would thus take too many decision variables to have a detailed enough solution that can appropriately approximate the aforementioned dynamics, hence degrading the performance of an online formulation of the algorithm by making iterations too slow and the problem too hard for the numerical algorithms involved, requiring additionally an enormous amount of computing resources. On the other hand, knowing a detailed solution of what is expected to happen in the long term loses sense in the presence of such unpredictable disturbances as the ones present in a solar car race. Recognizing that it is important to predict what will happen in the long-term future but also spotting that the actual limiting resource is the available energy, we can therefore summarize this future behavior by just considering the evolution of this variable in both its shapes: the one that is stored in the batteries and the one expected to be received from the sun. Still not abandoning the idea of having some sort of detail, we can still include more accurate calculations for the moment in which they matter the most: the current race time and a small period of time immediately adjacent.

The proposed algorithm, thus, consists in three stages, as shown in Figure 2: the first one consists in planning a long-term strategy, in the time span of several days, by just considering the solar energy income predictions for the next race days and the available battery energy at the beginning of the current day. This requires the team to forecast the number of days the car will take to reach the finish line, because the algorithm will force by the end of the final day the battery pack to be empty.[¶]

[¶]If the battery pack was not empty at the end of the race, it means the remaining energy could have been used to reduce the time required to reach the goal. On the other hand, depleting the battery pack on a previous day may not necessarily guarantee that the solar energy to receive the following days is enough to reload it and be able to sustain high speeds. Because all these decisions depend on the weather forecast, the energy consumption for the rest of the days is left as a decision variable.

DOI: 10.1002/oca

If C_d is the current racing day, N_d is the user-defined final race day, $\hat{E}_s^{(d)}$ is the estimated solar energy income for day (d), $\hat{E}_b^{(d)}$ is the battery energy estimation for the start of day (d), and $E_m^{(d)}$ is the energy consumed by the inverter-motor set during day (d), the first stage of the algorithm, called *weekly planning*, is executed by solving

$$\begin{aligned}
&\min_{\left\{E_m^{(d)}\right\}_{d=C_d}^{N_d}} \quad -\sum_{d=C_d}^{N_d} E_m^{(d)} \\
&\text{s.t.} \\
&\hat{E}_b^{(d+1)} = \hat{E}_b^{(d)} + \hat{E}_s^{(d)} - E_m^{(d)}, \ \forall d \in \{C_d, \ldots, N_d\}, \\
&E_{b_{(min)}} \leqslant \hat{E}_b^{(d+1)} \leqslant E_{b_{(max)}}, \quad \forall d \in \{C_d, \ldots, N_d\}, \\
&E_{m_{(D-min)}} \leqslant E_m^{(d)} \leqslant E_{m_{(D-max)}}, \quad \forall d \in \{C_d, \ldots, N_d\}, \\
&\hat{E}_b^{(N_d+1)} = E_{b_{(min)}}.
\end{aligned} \tag{9}$$

It is just necessary to run this stage once a day; with the reading of the battery SoC at the beginning of each race day, that makes therefore $\hat{E}_b^{(C_d)}$ actually a measured variable. $\hat{E}_s^{(d)}$ is assumed given by an appropriate radiation forecasting system. The author should note here that the decision variable is in this problem the consumed energy of the motor in each day. A natural question that arises next is then how is the car supposed to distribute this whole consumption during each day. Again, it is much more important to plan the current race day instead of the following ones, assuming that the remaining battery energy at the end of each day would remind the same when planning the following days with more detail. The second stage is this way introduced. We will call it *daily planning* and make it consist in solving

$$\begin{aligned}
&\min_{\left\{E_m^{(C_d,k)}\right\}_{k=k_c}^{k_f}} \quad -\sum_{k=k_c}^{k_f} E_m^{(C_d,k)} \\
&\text{s.t.} \\
&\hat{E}_b^{(C_d,k+1)} = \hat{E}_b^{(C_d,k)} + \hat{E}_s^{(C_d,k)} - E_m^{(C_d,k)}, && \forall k \in \{k_c, \ldots, k_{f1}\}, \\
&\hat{E}_b^{(C_d,k+1)} = \hat{E}_b^{(C_d,k)} + \hat{E}_s^{(C_d,k)}, && \forall k \in \{k_{f1}+1, \ldots, k_{f2}-1\}, \\
&\hat{E}_b^{(C_d,k_{f2})} = \hat{E}_b^{(C_d+1)}, \\
&E_{b_{(min)}} \leqslant \hat{E}_b^{(C_d,k)} \leqslant E_{b_{(max)}}, && \forall k \in \{k_c+1, \ldots, k_{f2}\}, \\
&E_{m_{(S-min)}} \leqslant E_m^{(C_d,k)} \leqslant E_{m_{(S-max)}}, && \forall k \in \{k_c, \ldots, k_{f1}\}.
\end{aligned} \tag{10}$$

At this point, the current day has been divided into an arbitrary number of time intervals k_{f2}. These are intended to include the energy evolution from the early morning, before the sun rises and the racing day starts, till the late night, after the racing day has finished and the sun has set. This allows to predict the battery-stored energy trajectory during the morning and evening. On the other hand, k_{f1} makes reference to the last of those intervals in which the vehicle can run, which depends on the race regulations.

This stage is updated periodically through the day, and because this implies that at a certain point some time intervals will be obsolete, it is not necessary to include those in the planning, motivating that k_c, the current interval, changes with the current race time. This way, the closer the clock is from the end of the racing day, the less this stage will take to be solved.

In this work, the time intervals were chosen to have a length of 15 min. This allows a still long prediction horizon for the third stage of the algorithm to be included and the number of intervals to be kept small enough for each day. It is important to notice that up to this point, all stages are based on the following principle: the time optimal solution will be based on spending the greatest amount of energy, during each day and time interval, in such a way that the batteries are empty at the end of the race and the energy is used in the most efficient way, which means being able to run the longest distance possible with a given energy budget for a given time interval. This last problem is to be

DOI: 10.1002/oca

solved by the last stage of the algorithm, but before introducing it, the reader should take into account that the energy limits used for the decision variables are chosen based on the datasheet maximum operating values for the inverter-motor set, considering regenerative braking and forward operation. For the weekly planning, the limits are normally to be set considering a lower continuous maximum power than for the daily planning. Finally, another thing to take into account when applying this algorithm is the fact that the present formulation of (9) and (10) is perfectly suitable for nearly flat tracks, but it still allows for considering a different objective function that relates energy, time, and space, in order to, for example, include information on spatial variation of the solar radiation or slopes and thus enabling the algorithm to make better decisions.

The third and last stage is called *continuous planning* and consists of applying a continuous-time optimal control problem solution algorithm to the problem

$$
\begin{aligned}
\min_{F} \quad & -r\left(t_f^{(C_d,k_c)} + \delta t\right) \\
\text{s.t.} \quad & \\
\frac{dv}{dt} \quad & = \frac{1}{m}\left(F - \frac{1}{2}C_D\rho_a A v^2 - mg\sin(\theta_r) - mg(C_{r1}\tanh\left(\frac{v}{5N}\right) + C_{r2}v)\cos(\theta_r)\right) \\
\frac{dr}{dt} \quad & = v \\
\frac{dE_m}{dt} \quad & = P_{in}(F,v) \\
0 \quad & \leqslant v(t) \quad \leqslant v_{(max)}, \forall t \in \left[T_c, t_f^{(C_d,k_c)} + \delta t\right] \\
0 \quad & \leqslant E_m(t) \quad \leqslant E_{mf}^{(C_d,k_c)}, \forall t \in \left[T_c, t_f^{(C_d,k_c)} + \delta t\right] \\
F_{(min)} \quad & \leqslant F(t) \quad \leqslant F_{(max)}, \forall t \in \left[T_c, t_f^{(C_d,k_c)} + \delta t\right] \\
v(T_c) \quad & = v_{T_c} \\
r(T_c) \quad & = r_{T_c} \\
E_m(T_c) \quad & = E_{m,T_c} \\
E_m(t_f^{(C_d,k_c)} + \delta t) \quad & = E_{mf}^{(C_d,k_c)},
\end{aligned}
\tag{11}
$$

in which

$$
E_{mf}^{(C_d,k_c)} = E_m^{(C_d,k_c)} + \frac{\delta t}{\left(t_f^{(C_d,k_c+1)} - t_0^{(C_d,k_c+1)}\right)} E_m^{(C_d,k_c+1)}. \tag{12}
$$

As a remark, please note that the system (11) is identical to (7), except for a change in the direction of the energy convention and the extraction of $P_s(t)$ from it. This is a consequence of having already considered the solar energy contribution on the previous phases.

As it has been said before, the daily planning is based on a fixed, previously chosen discretization of the current race day. On the other hand, the continuous planning is thought to be executed with the highest periodicity of all the other stages, and therefore most of the time, it will be ran at a time (denoted as T_c in (11)) that will not coincide with the starting instant of the current time interval $t_0^{(C_d,k_c)}$, letting a difference δ appear, as shown in Figure 3. If we emulate a moving-horizon model predictive control strategy, we find out that this difference also causes the problem to include a small portion of the next interval. This stage is thought to respect the previously made planning, and so this is why we add an extra term in the definition of the energy limit for this problem, $E_{mf}^{(C_d,k_c)}$, which includes the energy planning results for both time intervals. We also stress that in this stage, the control problem is a free final state one, and the initial state is obtained from the information obtained from the car's telemetry system, v_{T_c}, r_{T_c}, and E_{m,T_c}. The last variable makes reference to the measure in the consumed motor energy in the current time interval. While it is true that power integrators may develop a big measuring error with time, the formulation of the problem makes it

DOI: 10.1002/oca

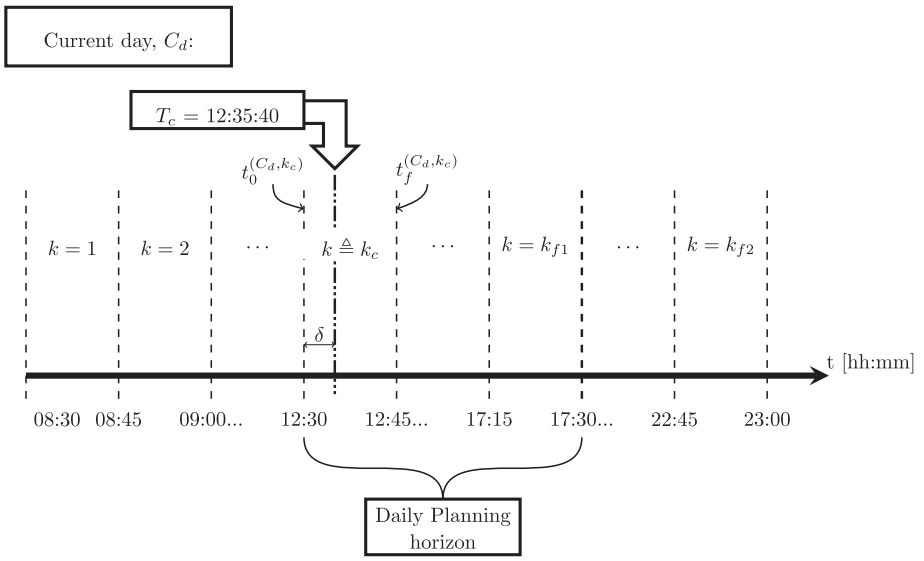

Figure 3. Location of the continuous planning in the daily planning discretization.

mandatory to reset this measure to zero when a new time interval is reached, reducing therefore the global measuring error along the race. According to all the previous, the final travelled distance for this problem is also evaluated at the end of the moving horizon. $t_f^{(C_d,k_c)}$ represents the final time of the current time interval.

The proposed algorithm is thought to be executed on an iterative basis. The continuous planning is to be executed several times before one iteration of the daily planning is made, but the switching criterion between stages may vary as well. While some users may prefer to set it to a number of iterations of the continuous, others may wait for some elapsed time between executions of the daily planning, while it is also possible to include a tolerance criterion for the deviation of the measured variables with respect to their predicted values. In the present paper, the last criterion is used.

For the solution of this problem, we have used the MATLAB-based, free software GPOPS 4.1 [16] that implements the Radau pseudospectral method [17] with the multiple interval discretization and *hp* mesh refinement algorithm of [18]. In summary, this method approximates the control and state trajectories by means of the Lagrange polynomials. In a single interval formulation of this method, the problem's time interval is transformed to the $[-1, +1)$ interval, and then the Lagrange polynomials are evaluated on the corresponding Legendre–Gauss–Radau (LGR) points. For a multiple interval formulation of the problem, the problem's time interval is divided in an arbitrary segment partition and each segment is transformed as before, evaluating thus the Lagrange polynomials corresponding to each sub-interval on the respective (LGR) points. In each case, the maximum polynomial degree for each interval is user-defined. For the current application, a trial and error test yielded a $N_i = 9$ uniformly distributed interval set and led to a choice of polynomials of degree $N_{npi} = 20$ in each one of them.

In this last sense, it is important to justify the aforementioned choice. Table III shows a summary of tests that were applied in order to evaluate the behavior of the solution with different configurations. These tests were performed for solving problem (11) at an arbitrary initial state of null speed, 100 km far away from the starting point and an energy limit of 0.1596 kWh, as shown in Figure 4. If the initial battery energy is 3.0 kWh, the final allowed battery energy corresponds to 2.8404 kWh. Due to the practical uncertainty on the road slopes, the tests are run on flat terrain and on a time span of 15 min. It is desired to maximize the travelled distance. It is noticeable that the objective function value does not depend on the collocation configuration, but rather, the difference appears to be in the shape of the solution itself and in the time required to calculate it (in all cases, the initial guess used in the optimization method was the same). In that same sense, from what follows from test number 2, it is possible to observe that increasing the number of nodes does not necessarily lead to

DOI: 10.1002/oca

Table III. Several tests on the collocation interval and node numbers.

Test no.	N_i	N_{npi}	Refined $N_i\,N_{npi}$	Solution time (s)	Objective value (km)	Comment
1	9	20	180	1.43	-1.46E+2	Well defined singular arc, no oscillations.
2	25	20	500	13.25	-1.46E+2	Same solution as in 1.
3	4	20	689	3.77	-1.46E+2	Significant oscillatory solution distortion.
4	9	15	135	2.49	-1.46E+2	Slight oscillatory solution distortion.
5	9	12	108	2.29	-1.46E+2	Slight oscillatory solution distortion.
6	9	10	540	3.89	-1.46E+2	Significant oscillatory solution distortion.

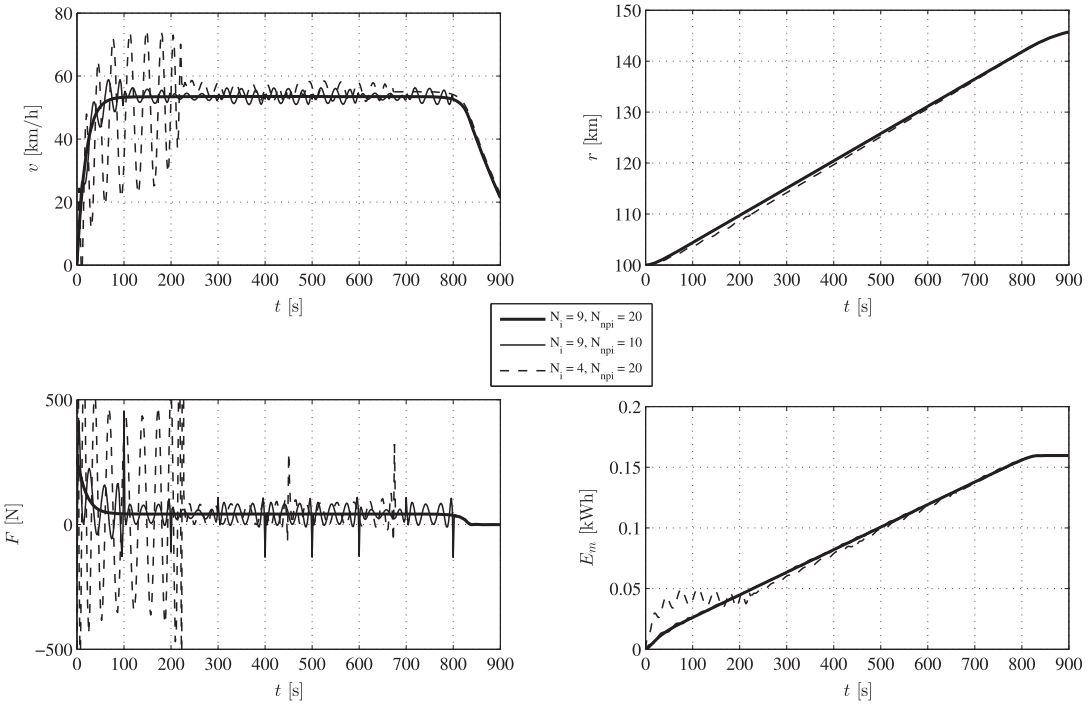

Figure 4. Effect of the pseudospectral method's configuration on the solution.

better solutions, and therefore, there is indeed an upper bound for the aforementioned magnitudes. Tests number 4 and 5 show that there exists a range in which the respective obtained solutions are still acceptable and could be under tolerance assumptions used in practice to obtain fast results, but a slightly oscillatory behavior typical of collocation methods starts to appear and increases as the number of nodes decreases. This phenomenon is shown in Figure 4. In Table III, the term 'singular arc' makes reference to that region of the state space where the value $E_m(t) = E_{mf}^{(C_d,k_c)}$ holds true.

The first three configurations in Table III consider a fixed number of nodes per interval with a variable number of intervals. The last three consists of the analogous for fixed number of intervals. For generating solutions from test number 2 and on the initial guess has been obtained from the solution of test number 1. It is important to notice that it might seem as if the smallest numbers of nodes would yield slower solutions, but that is actually due to the oscillatory distortion effect observed in Figure 4, in which a remarkable effect is that reducing the number of intervals to a half has a greater impact than doing the same with the number of nodes contained on each one. The initial guess is therefore farther from the actual solution than in the other cases. The difference on the solution time between tests 1 and 2, and 4 and 5 follow the logic that a greater number of variables requires, as expected, a bigger number of operations. Between tests 1 and 2, the number of nodes is almost triplicated, whereas the solution time is multiplied almost 10 times. This is not

DOI: 10.1002/oca

only due to the fact that more calculations are needed but also due to the fact that the inclusion of additional intervals also implies the satisfaction of additional matching conditions between them.

On GPOPS 4.1, the automatic mesh refinement algorithm of [18] is implemented. Besides the empirical tests performed, the algorithm is allowed to start from the corresponding guesses and increase or decrease the number of intervals and nodes. The results are as well included in Table III. Column *Refinement* $N_i N_{npi}$ includes the number of nodes obtained after a maximum of 10 refinement iterations.

From the tests performed, the preferred configuration to be used corresponds to $N_i = 9$ and $N_{npi} = 20$. It can be seen in the aforementioned table that the number of nodes in this case remains unaltered. Furthermore, if the thus obtained force is interpolated by Hermite piecewise cubic polynomials [19] and the system is integrated, the relative state error at the collocation points corresponds satisfies

$$e := \max_{k \in \{1 \ldots N_i N_{npi}\}} \frac{|x_{\text{int}}(k) - x_{\text{sol}}(k)|}{|x_{\text{sol}}(k)|} < 10^{-3}.$$

$x_{\text{int}}(k)$ is the result of the integration, while $x_{\text{sol}}(k)$ denotes the test's solution, both evaluated in node k. In other cases, the increase on the total node number was very significant. These solutions are thus discarded. The other non-varying solutions might have been taken as well, but in this paper, a compromise is made for additional detail, thus choosing N_i and N_{npi} as previously mentioned.

The Radau pseudospectral method has the advantage that the continuous-time dynamics of the solution are greatly contained in the polynomials used to approximate the solution, leaving a reduced information calculation burden for the numeric solver. The continuous-time problem is in this fashion transformed in a parametric nonlinear programming problem that has a large-scale sparse structure [20], which is very convenient for the application of methods based on sequential quadratic programming (SQP) such as SNOPT [21]. All this results advantageous in allowing the inclusion of a greater amount of detail in the short-term solution at the use of low computational resources and short calculation times.

4. RESULTS

The first interesting result[‖] of the algorithm is that, using an Intel® Core™ 2 Duo - E6850 CPU @ 3.00GHz, 64-bit, 8.00 GB RAM, it takes about 5 s to finish a complete cycle consisting in all its stages. This time is strongly influenced by the initial guess' closeness to the optimal solution of the continuous planning stage, which motivates reusing the solution of an iteration as the initial guess of the next one.

On a real implementation, the calculation of the optimal control and its application to the real system can be performed by separate devices. Each result of the continuous planning stage might be transmitted to a suitable real-time controller which will store and apply it until the next curve is received. Because each optimal control is obtained for long enough time horizons, the same horizons are the upper bounds for an acceptable calculation time. This corresponds to 15 min in this case. It is therefore not required to have a real-time operating system perform the optimal control calculations but only ensure that its perfomance is high enough, because the calculation time impacts the system's ability to keep optimality in the presence of disturbances.In this case, it proves sufficient to use Windows 7 Home Premium® or Ubuntu 14.04®. The time scales considered until now make data transferring times negligible.

In a normal operating situation, a complete cycle of the algorithm yields the results shown in Figures 5 and 6, in which expression (10) is used for the daily planning. Figure 7 shows, on the other hand, the effect of applying the moving horizon window. Therein it is possible to appreciate how the moving horizon strategy works like. At 08:30 h, then at 08:35 h, and then at 08:40 h, the continuous planning stage is executed respecting the daily planning results. In the first continuous planning

[‖]All results are obtained by computational calculations.

DOI: 10.1002/oca

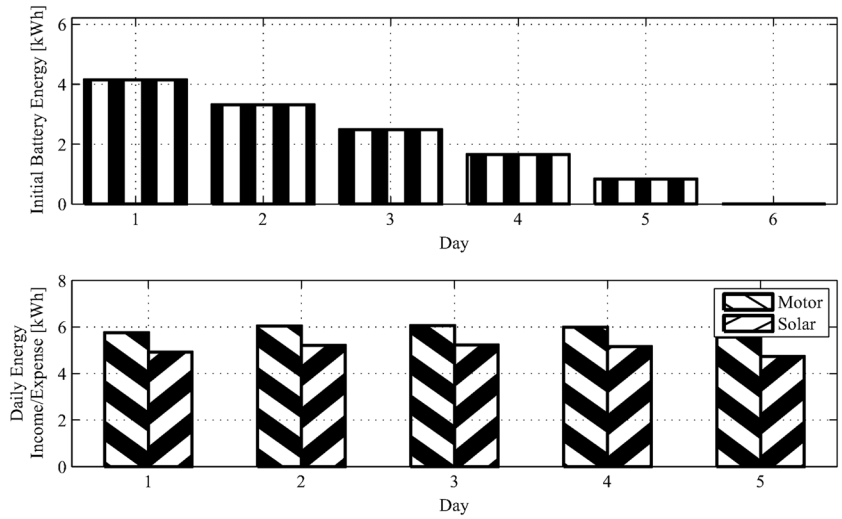

Figure 5. Weekly planning for a 5-day long race.

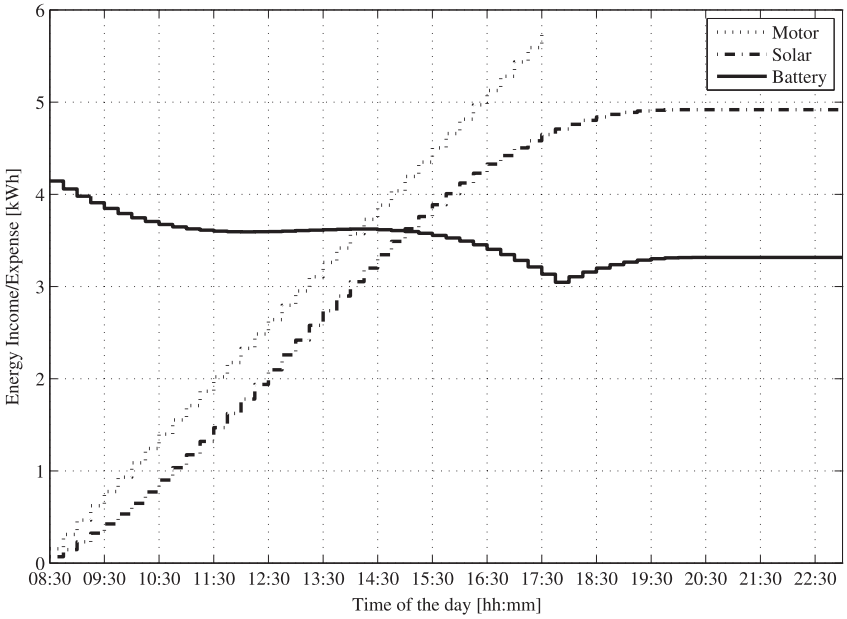

Figure 6. Energy planning for the first day.

iteration, the algorithm accelerates the vehicle from a null speed to an optimal one, then holds a constant speed, and finally lets the car decelerate freely in order to maximize the traveled distance in the considered period without going further in the use of energy. It is important to note that this comes only from the limited time scope and energy availability of problem (11). In the same figure, it is possible to see, on the other hand, how the iterative characteristic of the method overcomes this effect and keeps the horizon in which the speed is held near an optimum also moving, thus never allowing the car to freely decelerate.

Given now the case that the algorithm works properly in the absence of disturbances, let us have a closer look into what happens when the vehicle is in the presence of those. In that case, the model the algorithm uses to obtain the optimal curves and the car dynamics will not coincide. It is possible to distinguish two different kinds of scenarios: one in which the disturbances directly affect the performance of the car, such as winds, unpredicted slopes, mechanical failures and so on; and one in

DOI: 10.1002/oca

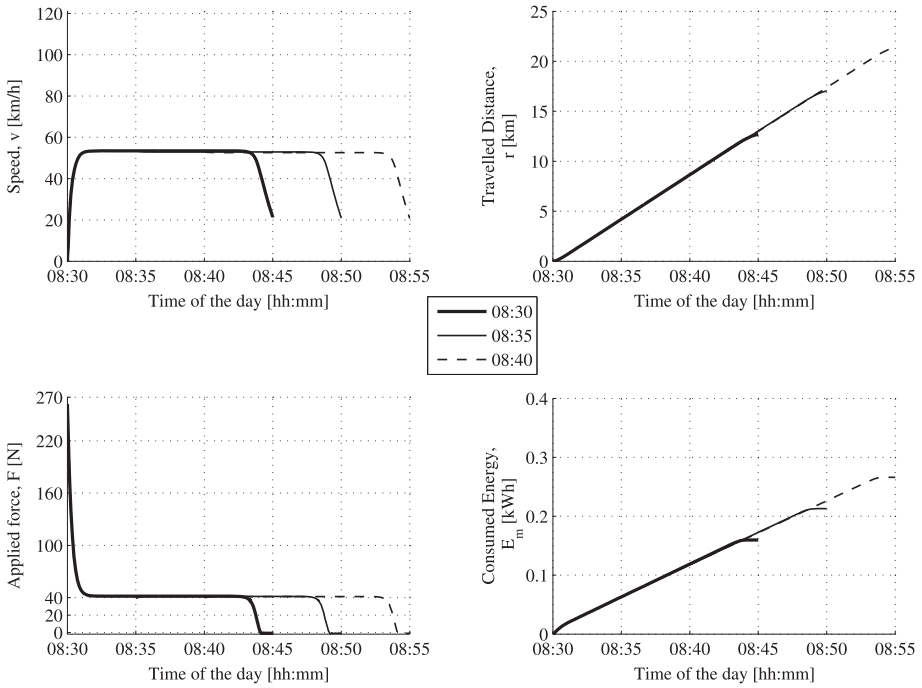

Figure 7. Continuous planning. Moving horizon effect when executing the stage at 08:30, 08:35 and 08:40 h.

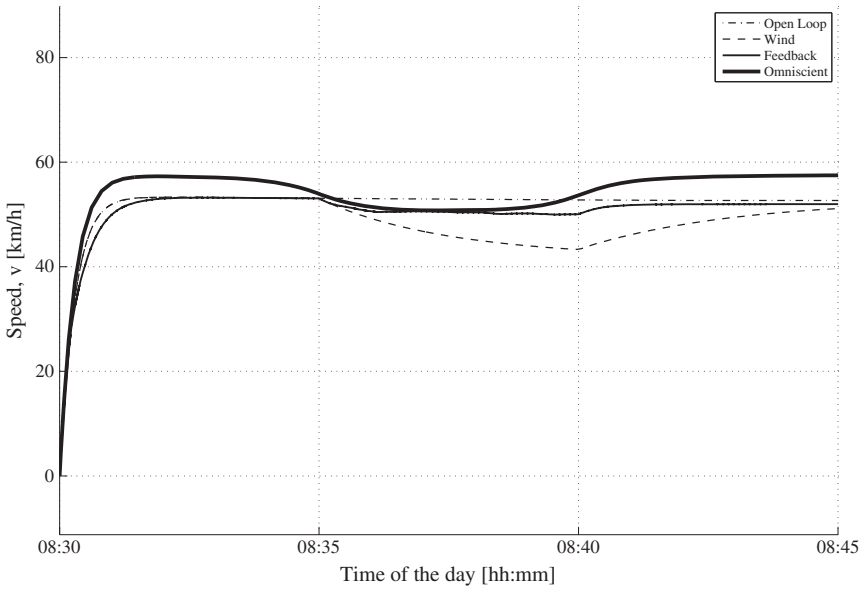

Figure 8. Results obtained while simulating a wind gust applied to the car when using different amounts of prediction information.

which the disturbances affect actually the energy availability of the car, such as cloudy weather, and failures in the battery pack or solar array. The proposed algorithm is able to cope with both kinds of scenarios.

For testing the first scenario, it will be assumed that the car begins the race in an environment such that all model parameters are accurate and the disturbances are null until 08:35 h, in which an unknown 15 km/h opposing wind gust is applied to the vehicle, lasting until 08:40 h. The algorithm is run considering one iteration of the daily and weekly planning stages to be fixed and applying

DOI: 10.1002/oca

Table IV. Summary of relevant results while testing the algorithm under a wind gust.

Scenario	Travelled distance at 08:40 h (km)	Consumed energy at 08:40 h (kWh)
Open Loop	8.614	0.118
Wind	8.085	0.111
Feedback	8.382	0.119
Omniscient	8.784	0.129

every 5 s the optimal control obtained from the continuous planning. The results for the speed are shown in Figure 8. If the measurement data v_{T_c}, r_{T_c}, and E_{m,T_c} had been taken from the planning itself due to lack of instrumentation, we would obtain the curve *Open Loop* in that figure as the result of the continuous planning. However, those results would not match reality after the wind gust, effect shown by curve *Wind* in the same figure. When the measurement data are accurately obtained from the car sensors, we obtain the *Feedback* curve. To enrich the discussion, we will also add the *Omniscient* curve, which is obtained by assuming a complete knowledge on the wind gust. Other relevant variables are summarized in Table IV.

The *Open Loop* and *Wind* scenarios show what would happen in the case we are trying to use an implementation that is too detailed to produce fast results: the predictions are too optimistic, and when disturbances come, the travelled distance ends being up much less as expected. On the other hand, having a whole – but impossible to get – knowledge of the disturbances to have along the way would allow to take much wiser decisions such as lowering the car speed during the gust and then increase it once again. The partial knowledge had in the *Feedback* curve, consequently, helps in improving the travelled distance for the given time interval, making it thus advisable to use the developed algorithm in the solar car. It must be noted, however, that the energy consumption in this last case is higher than the others due to the car's struggle against the wind force. This could have been prevented in case of counting with the appropriate instrumentation to predict this wind gust. In any case, the energy consumption still does not differ too much from the *Open Loop* and *Wind* scenarios.

In the case of energy income lacks or energy overspend, the main evidence for them will be the detection of a lower SoC than what is expected to be found during a particular time interval on the daily planning. The following study consists of three cases. In case 1, it is assumed that the car starts at 08:30 h with its full battery pack. Case 2 assumes that at 08:45 h it is found that the car's available battery energy is 90% of what was expected for that time in case 1, and case 3 assumes that at 10:30 h, the remaining energy equals 90% of the expected available energy in case 1 at the end of the day, thus having already consumed the whole day's planned energy. The algorithm reacts as shown in Figure 9, where it can be seen that the algorithm tries to keep the car moving at a lower speed, not stopping in spite of the low energy. It tries to consume less energy or to even be able to load energy so as to accomplish with the goals from the daily planning while still advancing towards the finish line.

5. DISCUSSION

In this section, the proposed method is compared with other optimal control techniques found on the literature. Some of the most used approaches so far consider piecewise constant speeds as decision variables on discrete-time processes, considering empirical loss models and planning with respect to the present energy availability and expected solar energy income. This strategy is included under the designation of *Steady State* and is in some cases implemented by the use of the dynamic programming algorithm. This time the solution is performed through a standard SQP line search algorithm. Other approaches integrate the dynamics of the vehicle as a continuous system, typically by means of methods analog to the single-shooting method, briefly explained in the subsequent discussion. In order to complement the discussion, the multiple shooting methodology is also included in the analysis.

DOI: 10.1002/oca

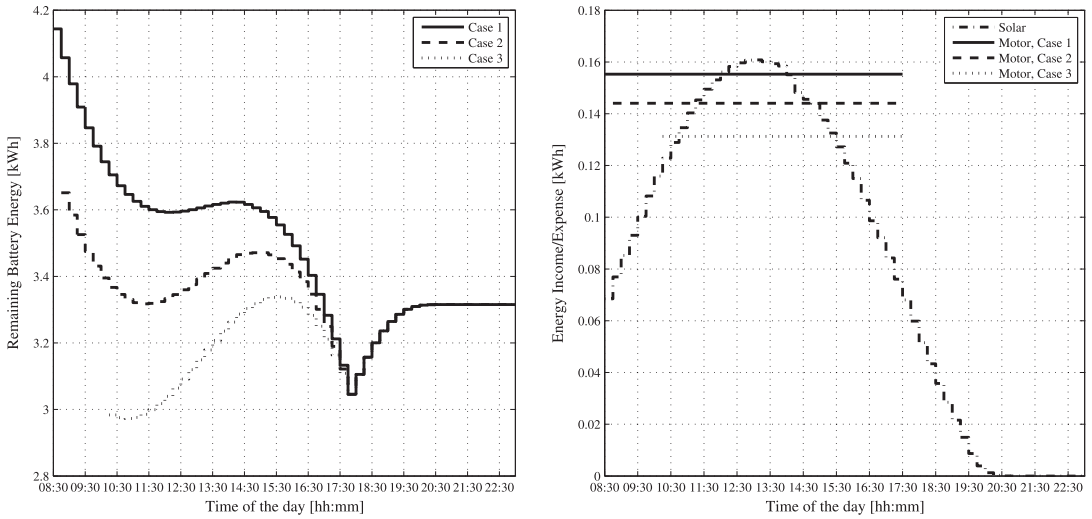

Figure 9. Daily planning reaction under different low battery SoC cases.

For simplicity, the comparison will be performed on a fictitious one-day race, and the aim of each algorithm will be to maximize the travelled distance during it. While all calculations concerning the proposed scheme are of course subject to the energy planning previously performed as described earlier, the other algorithms will be allowed to freely choose on this aspect. The following implementations are thus considered:

(a) *Single Shooting* [9, 10, 14]*:* On a fixed time, it is desired to find the control that maximizes the travelled distance subject to the previously explained constraints. For these means, the control function F in (7) is parameterized as constant q, and its value is on each iteration used to integrate the differential part of the problem in order to evaluate the constraints (border conditions and box constraints) and objective function to be minimized, thus obtaining a parametric optimization problem instead of an infinite dimension problem. Allowing the initial and final states to be decision variables as well with the corresponding constraints, the following system is obtained.

$$\begin{aligned}
&\min_{q,s_0,s_f} \quad -r(t_f) = -s_{f,2} \\
&\text{s.t.} \\
&\dot{x} \quad = f(t,x,q) \qquad \text{as in (7)} \\
&s_0 \quad = x_0 \\
&s_f \quad = s_0 + \int_{t_0}^{t_f} \dot{x}\,dt \\
&x_{(min)} \leqslant s_0, s_f \leqslant x_{(max)} \quad \text{as in (8)} \\
&F_{(min)} \leqslant q \quad \leqslant F_{(max)},
\end{aligned} \tag{13}$$

in which t_0 and t_f correspond to the start and end times of the race, concretely, 08:30 and 17:30 h of the day. The initial state is considered as the car starting at null speed, from the beginning of the track and with full battery pack. This corresponds to a vector $s_0 = [0,\ 0,\ 3]$. The vehicle speed must stay non-negative and under 120 km/h, and the battery pack remaining energy must remain non-negative and under the maximum capacity, 3 kWh. The force absolute value cannot exceed 500 N (the motor's maximum torque is 136 Nm and the wheel radius is 0.27 m).

From (7), the need for estimating the solar power and the road slope is seen. In Figure 10, the first one is defined as a look-up table whose input is the day time and the output is the quantity, while in the second case, the slope is defined by means of an analytical function. On the first

DOI: 10.1002/oca

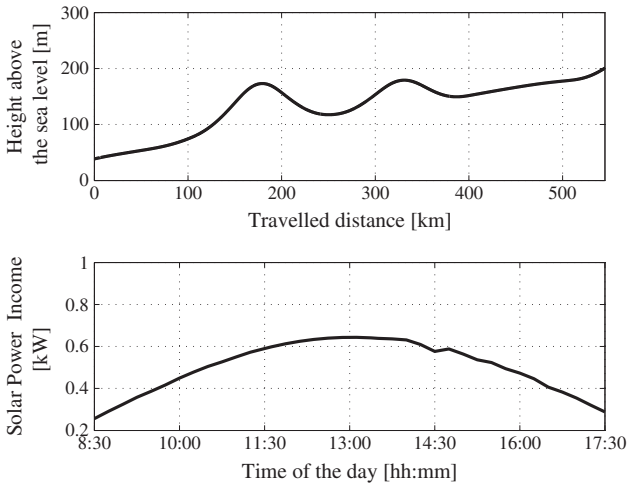

Figure 10. Altitude profile and solar power income used as basic scenario for comparison of the proposed method with other strategies.

case, thus, cubic spline interpolation has been used, in order to allow the dynamics right-hand side to keep being at least C^1, which greatly helps optimization and integration convergence. Approximation (5) is certainly used for the input power (4) with the same goal.

The problem is solved by means of Matlab's™ *fmincon* SQP line search algorithm. For derivative calculations, the analytical right-hand side derivatives with respect to state and controls are provided and the differential equations

$$\begin{aligned} \dot{G} &= G\frac{\partial f}{\partial x} \\ \dot{G}_q &= G_q\frac{\partial f}{\partial x} + \frac{\partial f}{\partial q} \\ G(t_0) &= I_{3\times 3} \\ G_q(t_0) &= (0,0,0) \end{aligned} \tag{14}$$

are integrated along with the state for obtaining thus $G(t_f) := \frac{\partial x(t_f)}{\partial x(t_0)}$ and $G_q(t_f) := \frac{\partial x(t_f)}{\partial q}$, which are in turn used to form the Jacobians for the constraint relating s_0 and s_f . Integrations are performed by means of the software's *ode45* method, which implements the Dormand–Prince Runge–Kutta (4,5) [22] pair.

(b) *Multiple Shooting* [9, 10, 14]*:* Implementation (a) might be improved by dividing the whole time horizon into a set of sub-intervals, applying single shooting to each one of them, and relating the corresponding subsystems to each other through corresponding continuity conditions.

$$\begin{aligned} &\min_{\{q_i\}_{i=0}^{N_i-1},\{s_i\}_{i=0}^{N_i}} -r(t_f) = -s_{N_i,2} \\ &\text{s.t.} \\ &\dot{x} = f(t,x,q_i) \qquad \forall t \in [t_i, t_{i+1}], \quad \forall i \in \{0,\ldots,N_i-1\} \qquad \text{as in (7)} \\ &s_0 = x_0 \\ &s_{i+1} = s_i + \int_{t_i}^{t_{i+1}} \dot{x}dt \quad \forall i \in \{0,\ldots,N_i\} \\ &x_{(min)} \leqslant s_i \leqslant x_{(max)}, \qquad \forall i \in \{0,\ldots,N_i\} \qquad \text{as in (8)} \\ &F_{(min)} \leqslant q_i \leqslant F_{(max)}, \qquad \forall i \in \{0,\ldots,N_i-1\} \end{aligned} \tag{15}$$

where N_i represents the number of intervals used in the discretization. In the present work, it has been considered $N_i = 80$. The numerical methodology in this case is the same as in case

DOI: 10.1002/oca

(a). A remarkable consequence of this parametrization is the growth on the number of integrations needed to evaluate the constraints. This is the strongest source of running time growth in the method with respect to Single Shooting. For the derivative calculation, we employ, on each node, the approach of (14).

(c) *Steady State* [6, 11–13]*:* Another way of obtaining optimal solutions found in the literature are methods in which the vehicle dynamics are absent and the problem is treated as in steady state. These approaches plan the energy consumption of the car and make use of loss models. Following our modeling, all this would correspond to consider all time derivatives null and therefrom obtain the required information. The following problem is obtained.

$$\begin{aligned}
&\min_{\{v_i\}_{i=0}^{N_i-1}} \quad -\sum_{i=0}^{N_i-1} \Delta t_i\, v_i \\
&\text{s.t.} \\
&E_{b,0} \;=\; E_{b0} \\
&E_{b,i+1} \;=\; E_{b,i} + E_{s,i} - \Delta t_i\, P_{in,i}(F_i, v_i), \quad \forall i \in \{0,\dots,N_i-1\} \\
&E_{b,(min)} \leqslant E_{b,i} \leqslant E_{b,(max)} \qquad\qquad \forall i \in \{0,\dots,N_i\},
\end{aligned} \tag{16}$$

where $P_{in,i}(F_i, v_i)$ is obtained from (4) with approximation (5). $E_{s,i}$ represents the net solar energy income corresponding to period $[t_i, t_{i+1}]$, $\forall i \in \{0,\dots,N_{i-1}\}$, obtained by integrating the solar power over it, $\delta t_i = t_{i+1} - t_i$, $\forall i \in \{0,\dots,N_{i-1}\}$ and F_i and r_i are intermediate variables obtained as follows. E_{b0} and r_0 remain the same as in the previous schemes.

$$\begin{aligned}
r_{i+1} &= r_i + \Delta t_i\, v_i, && \forall i \in \{0,\dots,N_i-1\} \\
F_i &= \frac{1}{2} C_D \rho_a A v_i^2 - mg\left(C_{r1} \tanh\left(\frac{v_i}{5N}\right) + C_{r2} v_i\right)\cos(\theta_{r_i+1}), && \forall i \in \{0,\dots,N_i-1\}
\end{aligned} \tag{17}$$

The resulting nonlinear optimization problem is this way completely independent from dynamic considerations, and therefore, all computations are performed quickly. One drawback of this method is that the lack of dynamics inclusion makes it necessary to have a very fine discretization grid in order to account for these effects. In this example, we are taking $N_i = 720$ for this method. The problem's structure allows for different optimization algorithms. As it is a continuous nonlinear problem, line search SQP methods [23] are adequate, and as in the previously presented schemes, that will be the preferred technique to use. However, in the literature, there are several examples in which such a problem is solved by means of dynamic programming.

All scenarios have the same constraints for controls and states and the same initial conditions for the state. For comparison between the proposed method against the techniques previously introduced, we consider the World Solar Challenge 2011 race elevation profile as depicted in Figure 10, where it can be seen it is a rather flat track. The considered solar radiation is visible in the same figure and corresponds to real data captured by the University of Chile's Meteorology Group at the Geophysics Department.

The results are shown in Figure 11. Different means of comparison are shown in Table V. It is noticeable that the energy consumption profiles, as well as the travelled distances, tend to be the same for all methods. The proposed algorithm yields a trajectory not worse than the others in these kinds of tracks in spite of the simplifications introduced. In terms of the interaction between force and speed, it can be seen that the single shooting solution, being constrained to a constant force, presents the most oscillating speed whereas the proposed algorithm, the multiple shooting, and steady state techniques show rather constant profiles, just consistent to the conclusions in [1]. On the applied forces, the jumps in the presented solution are due to the fact that feedback has been considered, different from the other scenarios. In this sense, Table V shows rows *Partial solution time* and *Whole solution time* that emphasize that for the present method, the whole solution is obtained as a sequence of partial solutions which are the ones calculated by the *continuous planning* stage, and they are available quicker than in the other cases, in which the whole race must be optimized. This might be advantageous along the race.

DOI: 10.1002/oca

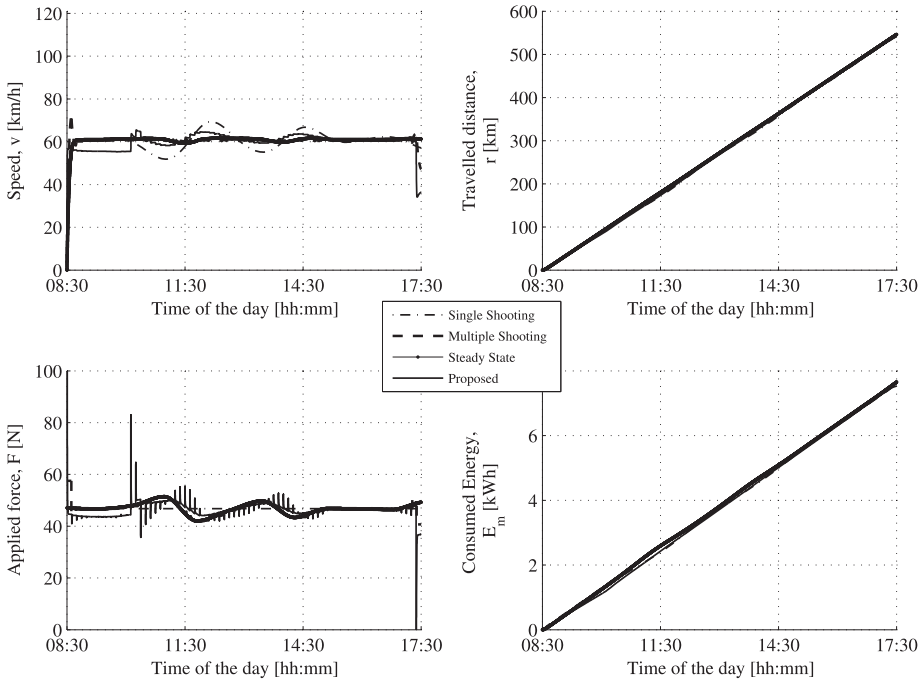

Figure 11. Proposed method's results in comparison with similar strategies.

Table V. Comparison of different methodologies for solving the optimal control problem.

Method/characteristic	Proposed	Single Shooting	Multiple Shooting	Steady State
Partial solution time (mm:ss)	00:05.00	03:53.41	18:16.40	00:34.70
Whole solution time (mm:ss)	01:26.38	03:53.41	18:16.40	00:34.70
Number of decision variables	720	7	323	720
Maximum distance (km)	540.9	541.8	543.6	541.2
Distance/energy ratio (km/kWh)	71.84	71.48	71.72	71.83
Average efficiency (%)	92.77	85.85	91.85	92.68

The proposed method is also close to the final optimal solutions computed, the small difference is explained by the introduction of the simplifications discussed earlier. Nevertheless, on a practical implementation, the faster feedback is expected to compensate this drawback, depending on how far away the race conditions are from the ideal.

The applied methods impact the performance of the car. The last two rows on Table V measure this fact. We define the average efficiency as

$$\eta_{average} = \frac{1}{N}\sum_{k=1}^{N}\frac{F_k v_k}{P_{in,k}}, \tag{18}$$

where the quantities are with respect to the solution of each problem. The distance/energy ratio is calculated from the terminal values of both the travelled distance and the consumed energy on each case. In these rows, the proposed method shows to yield better results. The consequence of making only short-term detailed optimizations is that the vehicle will tend to operate in points where the efficiency is maximized without considering future points. The other considered methods are presumably accessing less efficient operating points, accepting the cost of accelerated energy loss in exchange of additional mileage at the end of the race. This might also be risky if the modeling assumptions or race condition considerations stop being valid at a certain point, specially if they involve expected future energy recoveries.

DOI: 10.1002/oca

6. CONCLUSIONS

A new method for planning the energy consumption of a solar vehicle in a solar car race has been proposed. The method is able to include detailed information on the car's dynamics and is based on planning the energy consumption of the vehicle first to then include the more detailed dynamic model of the car into a continuous-time optimal control problem to be solved using pseudospectral methods.

It has been shown through simulations that the proposed algorithm is able to include the disturbed state information and thus produce trajectories fast enough, letting the car go through longer distances in a determined amount of time than schemes who take longer times to update their planning. The proposed method is able to obtain detailed solutions for the short term in the order of 5 s, allowing for slope differences and speed variations as described in analytic methods found in the literature. This gives the proposed algorithm advantages over less detailed methods, which are based on keeping a constant speed all along the way. The method has been successfully compared with other common approaches in the field of solar racing. The high degree of freedom of its control signal allows for reaching efficient operating modes with increased flexibility, and its sweeping nature allows for fast planning on the mid-term while keeping the number of decision variables constant.

The proposed approach is itself suboptimal, because it is based on an arbitrarily chosen time discretization of the current race day in order to execute the so-called daily planning. If this planning was continuous, the results would not be dependent on this choice. However, if the discretization intervals are chosen small enough, the energy predictions get more realistic, thus improving results.

It is important that the car model used for solving the continuous-time optimal control problem is accurate enough, and it is desirable to be able to predict the disturbances with the highest accuracy possible as well. This is due to the high dependence on this model the Radau pseudospectral method used for solving has.

In spite the present paper focusing on a specific application, the same reasoning can be used in any system involving solar energy: the current forecast information can be summarized to include the energy to receive on a certain time window, and then pseudospectral methods for optimal control could be used afterwards in order to solve the more detailed dynamics present in the given system.

ACKNOWLEDGEMENTS

The research reported in this paper was supported by CONICYT-Chile through grants FONDECYT Project 1120453 and the Basal Financing Program 'Center for Mining Technology' FB0809. The first author would like to thank the scholarship no. 22120217 granted by CONICYT for his Masters studies.

REFERENCES

1. Pudney P. Optimal energy management for solar-powered cars. *Ph.D. Thesis*, University of South Australia, 2000.
2. Santosa F, Guo H, Ketelsen C, Kletskin I, Li Q, Limon A, Mileyko Y. Solar car racing strategies. *6th PIMS-IMA Graduate Mathematics Modelling Camp,* Banff, AB, Canada, 2003; **1**:61–82.
3. Aranda González EJ. *Desarrollo de Estrategia para el Uso óptimo de la Energía en un Vehículo Solar*. Departamento de Ingeniería Eléctrica, Universidad de Chile: Santiago, 2008.
4. Athans M, Falb PL. *Optimal Control: An Introduction to the Theory and Its Applications*. Dover Publications, Inc.: Mineola, 1998.
5. Pontryagin L, Boltyanskii V, Gamkrelidze R, Mishchenko E. *The Mathematical Theory of Optimal Processes (International Series of Monographs in Pure and Applied Mathematics)*. John Wiley & Sons: New York, 1962.
6. Shimizu Y, Komatsu Y, Torii M, Takamuro M. Solar car cruising strategy and its supporting system. *JSAE Review* 1998; **19**(2):143–149.
7. Scheidegger A. Energy management optimization for a solar vehicle. *Master's Thesis*, École Polytechnique Fédérale de Lausanne, Lausanne, 2006.
8. Daniels MW, Kumar P. The optimal use of the solar powered automobile. *Control Systems, IEEE* 1999; **19**(3):12–22.
9. Boulgakov A. Sunswift IV strategy for the 2011 World Solar Challenge. *Bachelor's Thesis*, The University of South Wales, Sydney, 2012.
10. Mocking C. Optimal design and strategy for the SolUTra. *Master's Thesis*, Department of Electrical Engineering, University of Twente, Enschede, 2006.
11. Li CT, Dowling A. Solar car energy management strategy. *Momentum, SAE International*, Vol. 1, 2010; 6–7.

DOI: 10.1002/oca

12. Wright GS. Optimal energy management for solar car race. *Circuits and Systems, 1996., IEEE 39th Midwest Symposium on*, Vol. 3: IEEE, 1996; 1011–1014.
13. Cheng JH, Li CT. Backward energy management algorithm for a solar car. *World Renewable Energy Congress X and Exhibition*, Glasgow, UK, 2008.
14. Rao AV. Survey of numerical methods for optimal control. *AAS/AIAA Astrodynamics Specialist Conference*, Vol. 135, Pittsburg, PA, 2009; AAS Paper 09–334.
15. Gillespie TD. *Fundamentals of Vehicle Dynamics*. Society of Automotive Engineers, Inc.: Warrendale, 1992.
16. Rao A, Benson D, Darby C, Mahon B, Francolin C, Patterson M, Sanders I, Huntington G. *User's Manual for GPOPS Version 4.x: A MATLAB Software for Solving Multiple-Phase Optimal Control Problems Using hp-Adaptive Pseudospectral Methods*: University of Florida: Gainesville, 2011.
17. Kameswaran S, Biegler LT. Convergence rates for direct transcription of optimal control problems using collocation at Radau points. *Computational Optimization and Applications* 2008; **41**(1):81–126.
18. Darby CL, Hager WW, Rao AV. Direct trajectory optimization using a variable low-order adaptive pseudospectral method. *Journal of Spacecraft and Rockets* 2011; **48**(3):433–445.
19. Fritsch F, Carlson R. Monotone piecewise cubic interpolation. *SIAM Journal of Numerical Analysis* 1980; **17**: 238–246.
20. Rao AV, Benson DA, Darby C, Patterson MA, Francolin C, Sanders I, Huntington GT. Algorithm 902: GPOPS, a MATLAB software for solving multiple-phase optimal control problems using the gauss pseudospectral method. *ACM Transactions on Mathematical Software* 2010; **37**(2):1–39.
21. Gill PE, Murray W, Saunders MA. SNOPT: an SQP algorithm for large-scale constrained optimization. *SIAM Journal on Optimization* 2002; **12**(4):979–1006.
22. Dormand JR, Prince PJ. A family of embedded Runge–Kutta formulae. *Journal of Computational and Applied Mathematics* 1980; **6**(1):19–26.
23. Nocedal J, Wright S. *Numerical Optimization*. Springer Series in Operations Research: New York, USA, 2006.

DOI: 10.1002/oca