Optimization techniques applied to railway systems

Maialen Larrañaga, Jonatha Anselmi, Urtzi Ayesta, Peter Jacko, Asier Romo

Optimization Techniques Applied to Railway Systems

Maialen Larrañaga[2,6], Jonatha Anselmi[1], Urtzi Ayesta[2,3,5,6], Peter Jacko[1], Asier Romo[4]

[1]BCAM – Basque Center for Applied Mathematics, Bilbao, Spain,
[2]CNRS, LAAS, 7 Avenue du Colonel Roche, F-31400 Toulouse, France,
[3]IKERBASQUE, Basque Foundation for Science, 48011 Bilbao, Spain,
[4]INGETEAM Traction, S.A., Derio, Spain,
[5]Univ. of the Basque Country,
Department of Computer Science and Artificial Intelligence,
Manuel de Lardizabal, 1, 20018 Donostia, Spain,
[6]Univ. de Toulouse, INP-ENSEEIHT, LAAS, F-31400 Toulouse, France.

January 24, 2013

Abstract

We study the problem of minimizing the usage of electrical energy in rail systems. The aim is to determine a train speed profile that minimizes energy consumption given a time schedule. In collaboration with an industrial partner, we propose a new model that is more complete than the ones existing in the literature, in particular the model takes into account several non-linearities that emerge in a real setting. First, we formulate our problem within the framework of optimal control where our solution approach consists in discretizing the control problem and solving numerically the finite-dimensional optimization problem that is obtained out of the discretization. To do so we develop a platform based on AMPL and Ipopt that allows a fast and accurate solution. We then reformulate the problem within the framework of Dynamic Programming which allows to get the optimal action for any initial point. Solving the Dynamic Programming is very time consuming and we develop a C++ code to solve some simple examples. We finally implement our solution in a train simulator in order to estimate the energy reduction obtained in several real examples provided by INGETEAM S.A. The results obtained by the simulator indicate that the energy reduction is between 8% and 25%. We thus conclude that our first approach represents a scheme that could be implemented by industry to solve real-life cases.

1 INTRODUCTION

The problem of reducing energy consumption in railway systems has received lot of attention in recent years because of its known impact on economy and environment; see, e.g., [10] for a monograph. This task can be accomplished by providing train drivers with speed profiles that optimize the usage of electrical power while satisfying some timetables. The problem of finding the optimal speed profiles can be formulated as an optimal control problem where the utility function is the consumed energy and the control actions are the power and brake levels

Maximum acceleration	power= P_{max}, brake= 0
Speed-Holding	power s.t. $\frac{dv}{dt} = 0$, brake= 0
Coast	power= 0, brake= 0
Maximum brake	power= 0, brake= Q_{max}

Table 1: Optimal Driving Modes

selected by the driver. Unfortunately, given the complex relations between train motion and energy consumption, which are non-linear, analytical results are scarce and are available only in simplified cases.

However, we can mention the following very relevant results. In [8], it was shown that for solar powered cars the optimal speed profile is a combination of four different driving modes. This result was adapted (see [10]) to show that the optimal speed profiles of trains are also a combination of four driving strategies (see Table 1): maximum power, speed-holding, coast and maximum brake. A simplified model was considered in [7] and the existence of an optimal solution of the optimal-control problem is shown by applying Pontryagin's Maximum principle. In [13], it is shown that the optimal speed profile is unique in simple cases such as level-track trajectories, i.e., a completely flat path between origin and destination. The complexity of the problem reduces considerably by combining the uniqueness result and the fact that the solution contains only four possible driving modes, since now the problem reduces to finding the switching points.

Given the complexity of the problem mentioned above, a large deal of effort has been devoted to the development of approximate numerical schemes and algorithms. From a computational perspective, an important stream of research has been based on the dynamic programming approach; see, e.g., [4]. In this case, the decision process is simplified by dividing the problem into simpler subproblems. This approximation implies a great computational cost. Another research direction aims at designing speed profiles that are energy efficient by exploring all the possible combinations allowed by the so-called ATO commands. The ATO commands are not designed for minimizing energy but for passenger comfort reasons. Measurements in Metro de Madrid show that the reduction in energy can be up to 13%, see [6].

The goal of this paper is to provide a robust and efficient computational framework to find an optimal speed profile that train drivers should follow to optimize the usage of electrical energy within a given time schedule. We formulate this energy minimization problem for one train in two different ways: (1) as an Optimal Control (OC) problem, (2) as a Dynamic Programming (DP) problem. Unlike the models that have been studied in the literature, our approach takes into account relations between train motion and energy consumption that are more realistic. In particular, the characteristic curve of the tractive and brake efforts that we consider have a piece-wise non-linear behavior that is train-dependent. This behavior is typical in vehicles with electric motor. It turns out that these relations have a significant impact on the structure of the optimal solution and yield more accurate results. In our approach, we also take into account other essential aspects such as resistance to air, gradient force, comfort constraints, maximum acceleration and different maximum speed limits.

Given the non-linear nature of all the features mentioned above, an analytical solution of the resulting problem is extremely difficult to obtain. Therefore, we propose two methods of a very different nature to solve the problem numerically. In the first approximation we perform a discretization of the model converting the continuous-time model into a standard (finite-dimensional) non-linear optimization problem. This allows us to obtain efficiently an approximate solution of the original continuous-time problem using well-established interior-point methods from optimization theory. A software prototype has been built using AMPL [11] and Ipopt [3]. In the second approximation we discretize all the variables considered in the model obtaining a discrete state space and we adapt it so that we get a recursive objective function where the cost adds up in time. This reformulation of the model within the framework of Dynamic Programming can be solved by backward recursion and an optimal speed profile can be obtained for every possible initial point of the state space. This can be very useful, since once the DP is solved, the driver could instantaneously know how to react to any change or modification along the trip (a longer break than expected, lowering the speed due to bad meteorological conditions etc.). Solving the DP numerically can be very expensive computationally. In [4] they allow only 10 possible values for the control variables to reduce the computational cost, in our case we use the information of the solution obtained by AMPL/Ipopt to reduce cost without adding any restriction to the control variable. We have developed a software package in C++ that allows to solve real life examples, although the accuracy is not good enough for a possible implementation in a real system. We believe this is a very promising research avenue for the future, since more research is needed in order to make this technique suitable for implementation in industry.

To evaluate the impact of our study, we compare our speed profiles with real speed profiles taken from a metro of a major city in Europe and provided by INGETEAM Traction S.A. This comparison is performed on top of a train simulator developed by INGETEAM Traction S.A. The results obtained by this simulator over several trip profiles indicate that the energy reduction of a typical trip is between 8% and 25%.

The main contributions of the paper are: (i) the development of a new model that captures more closely, than the ones available in the literature, the real and desired dynamics of the system for short journeys as in metro systems, (ii) the development of a numerical platform based on AMPL and Ipopt that allows a fast and accurate solution, (iii) the development of a C++ code that provides an optimal solution under any possible circumstance, and (iv) the implementation of our solution in a train simulator in order to estimate the energy reduction obtained in several real examples.

The rest of the paper is organized as follows. In Section 2 we develop the model, in Section 3 we briefly describe the main numerical methods to solve control problems in the context of railway systems, in Section 4 we introduce our first solution approach that is based on a discretization of the original problem, in Section 5 we present our second approach that consists on the discretization of the whole state space and the definition of a recursive objective function, and in Section 6 we present the numerical results obtained with the train simulator. Finally, Section 7 draws the conclusions of our work and outlines future research.

2 MODEL DESCRIPTION

As in other works, e.g., [10], we assume a *point-mass* train. Let $p(t) \geq 0$ be the power applied at time t to accelerate the train and $q(t) \geq 0$ be the power at time t to decelerate it. Then, the dynamics of a point-mass train is described by the following state equations,

$$\frac{dx(t)}{dt} = v(t),\ \ x(0) = 0, \tag{1}$$

$$m\frac{dv(t)}{dt} = \frac{p(t) - q(t)}{v(t)} - R(x(t), v(t)),\ \ v(0) = 0, \tag{2}$$

where $x(\cdot)$ denotes the position, $v(\cdot)$ the speed and $R : \mathbb{R}^2 \to \mathbb{R}$ is the counterforce given by

$$R(x, v) := m\,(a + bv + cv^2) + m\,g\,\sin\alpha(x), \tag{3}$$

where a, b and c are drag coefficients, m is the mass of the train, g is the gravitational acceleration and $\alpha(x)$ is the slope of the track in point x. The first summand in (3) captures the air resistance and is known as Davis formula, and the second summand corresponds to the gradient force.

The objective is to find function $(p(t), q(t))$, i.e., a power profile, that minimizes the energy consumed by a train in a journey while ensuring that the train reaches destination in no more than T time units. This yields the following optimal control formulation

$$\min_{p(t),q(t),v(t),\,\forall t\in[0,T]} \int_0^T p(t)\ \mathrm{d}t, \tag{4}$$

subject to state equations (1) and (2),

$$x(T) = X, \tag{5}$$

$$p(t)q(t) = 0, \tag{6}$$

$$v(t) \leq V_{max}, \tag{7}$$

the comfort or jerk constraints

$$\left|\frac{d^2v(t)}{dt^2}\right| \leq 0.8\ \mathrm{m/s^3}, \tag{8}$$

$$-1.2\ \mathrm{m/s^2} \leq \frac{dv(t)}{dt} \leq 1.15\ \mathrm{m/s^2}, \tag{9}$$

and the tractive/brake effort constraints,

$$\frac{p(t)}{v(t)} \leq \begin{cases} E_{\max} & \text{if } v(t) < \omega_0, \\ \frac{E_{\max}\omega_0}{v(t)} & \text{if } \omega_0 \leq v(t) < \omega_1, \\ \frac{E_{\max}\omega_0\omega_1}{v^2(t)} & \text{if } \omega_1 \leq v(t), \end{cases} \tag{10}$$

$$\frac{q(t)}{v(t)} \leq \begin{cases} E_{\max} & \text{if } v(t) < \omega_2, \\ \frac{E_{\max}\omega_2}{v(t)} & \text{if } \omega_2 \leq v(t), \end{cases} \tag{11}$$

$$p(t), q(t), v(t) \geq 0, \tag{12}$$

where $E_{\max}$ is the constant that defines the maximum tractive effort and brake effort and the values ω_0, ω_1, ω_2 represent the speed at which the characteristic curve of the tractive effort changes.

The incorporation of constraints (10)-(11) represent an important contribution of the paper. Their expression show that the power consumed when accelerating or braking the train strongly depends on the speed of the train and on other train-dependent coefficients. In existing works (see for example [10], [13]) (10)-(11) have the more simple form:

$$p(t) \leq P_{\max} \text{ and } q(t) \leq Q_{\max}, \tag{13}$$

where the constraints become a restriction in the useable power directly. Making this assumption yields the possibility of giving power without taking the speed of the train into account.

To illustrate the impact of constraints (8)-(11), we consider a real case-study for a short trip. This refers to a track which presents uphill inclines during all the ride, see Figure 1. Figure 2 shows three different profiles for tractive effort and speed: i) the real profiles (RP),

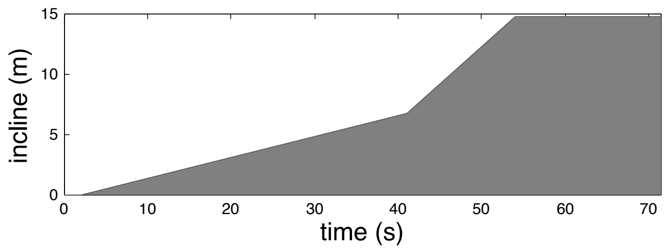
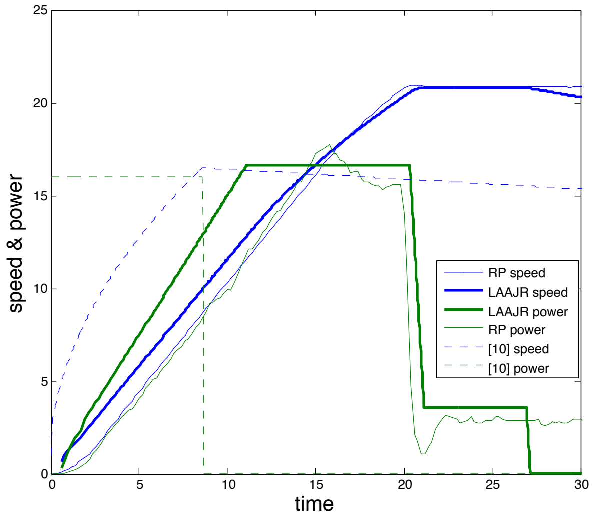


Figure 1: Track profile for example 1.

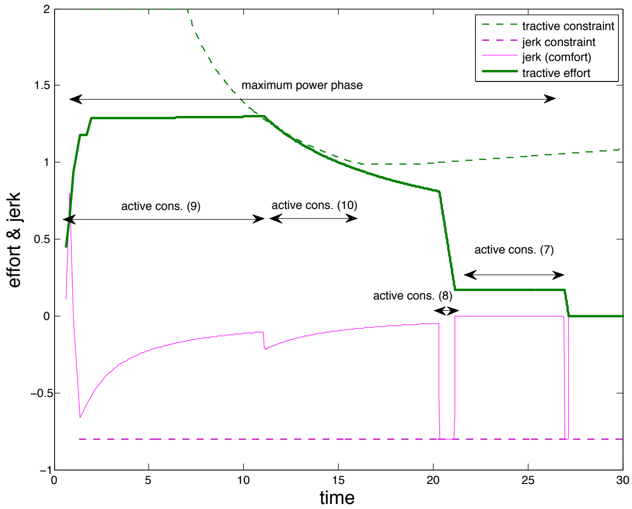
which are measured in a real example, ii) the profiles obtained by applying the model in [10], and iii) the profiles obtained with our model (LAAJR). We plot the profiles obtained in the first 30 seconds of the ride (the ones for all ride will be described in Section 6). In Figure 2 we highlight the impact of constraints (8)-(11). Due to the fact that all the constraints are considered, the speed profile obtained in LAAJR captures the physics of train very closely. In Figure 2a we observe that the simplified model, as used in [10], does not capture the dynamics of trains accurately. Since RP and LAAJR are close, we conclude that the current profile used in reality, in the first 30 seconds of the ride, is very close to the optimum.

3 NUMERICAL METHODS FOR OPTIMAL CONTROL PROBLEMS

As we have explained in the introduction, given the complex relations between train motion and energy consumption, analytical results are available only in simplified cases. Due to this difficulty, a large deal of effort has been devoted to the development of approximate numerical schemes and algorithms. In this section we briefly mention some of the most important approaches undertaken within the context of optimization of railroad systems. We refer to, e.g., [1] and [5] for a more exhaustive overview on numerical methods applied to optimal control.



(a) Speed and power profiles.



(b) Active constraints.

Figure 2: Comparison between the three profiles for a real example.

3.1 Direct Methods

Direct methods do not require a previous knowledge about the structure of the solution. The first step is to discretize the problem to obtain a finite dimensional problem and then Nonlinear Programming (NLP) techniques can be used. The idea of these techniques is to solve simpler subproblems that converge to the original solution in a finite number of iterations or in the limit. Two different type of algorithms are addressed:

1. *Interior-Point Method and Penalty Function methods:* The problem is reformulated to convert it into an unconstraint optimization problem. Afterwards, unconstraint optimization methods can be used to find a solution, such as gradient based methods.

2. *Newton-like Methods:* The problem is solved by finding a point that satisfies the Karush-Kuhn-Tucker conditions (necessary conditions for optimality). In [2] quadratic programming was used to solve the simplified model of trains introduced in [10], but we note that the method did not converge for various examples.

3.2 Indirect Methods

Indirect methods require to have knowledge about the structure of the solution, because a good initialization is needed. As an advantage, the discretization of the problem is not needed. The most important method is the so-called *shooting method*. The idea is to iteratively improve the estimates of the adjoint values, the Lagrange multipliers and the terminal time, so that the Euler-Lagrange equations are satisfied. In [9], a shooting method is used to find the switching points at which the driving mode is changed from speed-hold to maximum power or coast.

4 SOLUTION APPROACH: BY DIRECT METHOD

As explained in the introduction, our aim is to develop a numerical scheme that could be implemented in a metro in order to assist the driver. Thus, the solution approach needs to be fast, efficient, robust and completely automatic, that is, without requiring any initialization or human assistance. This motivates us to approximate the optimal control as a finite dimensional optimization problem, as said in Section 3.1.

However, in contrast with previously presented schemes, we propose to solve the finite-dimensional optimization problem directly. This approach was perhaps too time consuming a few years ago, but due to the tremendous technological progress of recent years we will show that it is possible to solve real-life examples in a few seconds with a personal computer. There are three main steps in our approach. First, we perform a discretization of the original continuous-time formulation. Then, we formulate the problem as a finite-dimensional optimization problem and, finally, we develop a software prototype to solve the problem based on AMPL and Ipopt, which are consolidated tools for the formulation and the solution of finite-dimensional optimization problems.

4.1 Discretization

The problem needs to be adapted due to the fact that both the gradient force and the speed limits are determined by the position at which the train is located. Therefore, the state equations and the cost functional are reformulated in order to make them space dependent. The conversion of the continuous optimal control problem is then performed by discretizing the state space uniformly.

Let N be the number of space stages in the interval $[0, X]$ of length $\Delta x = X/N$. Then, by using a midpoint rule for the discretization of the cost functional, we obtain

$$J(\overrightarrow{p}, \overrightarrow{v}) := \sum_{\ell=0}^{N-1} \frac{p_\ell + p_{\ell+1}}{v_\ell + v_{\ell+1}} \Delta x,$$

and using Euler's method for the state equation, we obtain for $\ell = 0, \ldots, N-1$,

$$m \frac{v_{\ell+1} - v_\ell}{\Delta x} = \frac{p_\ell - q_\ell}{v_\ell} - R(\ell \Delta x, v_\ell),$$

subject to the initial and boundary conditions,

$$v_0 = 0, v_N = 0, t_0 = 0, t_N = T,$$

4.2 Finite dimensional constraint optimization

After the discretization process the following nonlinear optimization problem is obtained,

$$\text{Problem (P1)}: \quad \min_{p_\ell, q_\ell, v_\ell, \ell=0,\ldots,N} J_{x_0, v_0} \tag{14}$$

subject to for $\ell = 0, \ldots, N-1$

$$\begin{bmatrix} v_{\ell+1} \\ t_{\ell+1} \\ \Delta t_\ell \end{bmatrix} = \begin{bmatrix} v_\ell + \frac{\Delta x}{m} \left(\frac{p_\ell - q_\ell}{v_\ell} - R(\ell \Delta x, v_\ell) \right) \\ t_\ell + \Delta x \frac{2}{v_\ell + v_{\ell+1}} \\ t_{\ell+1} - t_\ell \end{bmatrix},$$

$$p_\ell q_\ell = 0, \tag{15}$$

$$v_\ell \leq V_{max}(\ell \Delta x), \tag{16}$$

$$\left| \frac{(v_{\ell+1} - 2v_\ell + v_{\ell-1})v_\ell^2}{\Delta x^2} \right| \leq 0.8, \tag{17}$$

$$-1.2 \leq \frac{(v_{\ell+1} - v_\ell)v_\ell}{\Delta x} \leq 1.15, \tag{18}$$

$$\frac{p_\ell}{v_\ell} \leq \begin{cases} E_{\max} & \text{if } v_\ell < \omega_0, \\ \frac{E_{\max}\omega_0}{v_\ell} & \text{if } \omega_0 \leq v_\ell < \omega_1, \\ \frac{E_{\max}\omega_0\omega_1}{v_\ell^2} & \text{if } \omega_1 \leq v_\ell, \end{cases} \tag{19}$$

$$\frac{q_\ell}{v_\ell} \leq \begin{cases} E_{\max} & \text{if } v_\ell < \omega_2, \\ \frac{E_{\max}\omega_2}{v_\ell} & \text{if } \omega_2 \leq v_\ell, \end{cases} \tag{20}$$

$$p_\ell, q_\ell, v_\ell \geq 0,$$

for all $\ell = 0, \ldots, N$. Let P^ℓ_{max} denote the maximum traction power that can be allocated at step ℓ while the constraints (16)-(20) are satisfied, respectively, let Q^ℓ_{max} be the maximum braking power that can be allocated at step ℓ making sure constraints (16)-(20) are satisfied. The formulated model is solved using Ipopt and we obtain the optimal speed and power profiles that are energy efficient.

4.3 Implementation and Results

We program in AMPL/Ipopt the optimization problem (P1), and we solve it in a Core i3 @ 3.2 GHz, 4GB AM 1333 MHz DDR3 computer. Here, Ipopt (see [3]) is a nonlinear optimization solver that provides with a local optimal solution, it is programed in C++ and it is open source, it requires third party code in order to compile it, such as AMPL (see [11]). AMPL is a mathematical programming language that provides automatic differentiation functionality and it allows the programmer to model with the same mathematical notation used in regular optimization problems. With this software prototype that we have built we analyse a large variety of track profiles, some taken from examples available in the literature, and some other real profiles provided by INGETEAM S.A. Based on the extensive numerical computations carried out we observe that the obtained power profiles are characterized by four possible actions: maximum power, speed hold, coast and brake, see Table 2. This observation is known to hold for a simplified version of the problem (see [10]), but remains to be proven under the more realistic constraints. We note that the value of power when maximum acceleration is applied in Table 2 is the maximum that the constraints in tractive efforts, comfort, maximum speed and maximum acceleration allow, *i.e* P^ℓ_{max}, analogously the value for q will be computed by taking the maximum that these constraints allow, *i.e* Q^ℓ_{max}, see Figure 2.

We finally note that in all real examples provided, the computation time required is less than 8 seconds, which makes the method very interesting in view of a real application.

Table 2: Optimal Driving Modes for the Proposed Model

Maximum power (a)	$q_\ell = 0, p_\ell = P^\ell_{max}$=maximum value allowed by constraints (16)-(20)
Speed-Holding (h)	p_ℓ s.t $\frac{dv}{dt} = 0$, $q_\ell = 0$
Coast (c)	$p_\ell = 0$, $q_\ell = 0$
Maximum brake (b)	$p_\ell = 0$, $q_\ell = Q^\ell_{max}$=maximum value allowed by constraints (16)-(20)

5 SOLUTION APPROACH: BY DYNAMIC PROGRAMMING

The direct method outlined in Section 4 allows to obtain the optimal solution given an initial point (x_0, v_0). From a practical point of view this is quite limited. Indeed imagine the situation where a train is following the optimal trajectory and an unexpected event takes place (for instance slowing down the speed due to the presence of an obstacle). This implies that the formerly obtained solution might no longer be optimal. A possible solution will be to solve problem (P1) again on-line. Another solution, which we present here, will be to solve off-line the optimization problem for all initial points in an efficient way. This can be done with a DP technique.

Dynamic Programming (DP) techniques allow to solve the control problem without any initialization of the problem and under any given circumstances the optimal solution can be computed, this is one of the major advantages of the DP approach, it goes through the whole state space to provide with a solution from any possible point of the state space until destination. The idea is to divide the complex problem into simpler subproblems, and each time a subproblem is solved the solution is stored in the memory to help solve bigger subproblems. The major drawback of using this method is that it implies a very expensive computational cost, to overcome this issue in [4] they propose to allow only 10 possible values for the control variable for all the points of the state space. However, in the solution we obtained by the direct approach we observed that the optimal speed profile is always characterized by four different driving modes, see Table 2. Therefore, for each point in the state space only 4 possible values should be considered for the control variable, letting this 4 values vary according to the point in the state space in which we are. This trick allows to reduce the number of possible control variables without adding any restriction on the values that the control can take. It allows a considerable reduction of the memory required to store the information, compared to the existing solution presented in [4].

5.1 Dynamic Programming formulation

The DP formulation of the energy minimization for one train problem consists on discretizing the model proposed in Section 2 by defining a uniform time, space and speed discretization and adapting the objective function. We divide the time interval $[0, T]$ into N_t intervals of length Δ_τ, $[0, X]$ into N_x and $[0, v_{max}]$ into N_v steps, where v_{max} is the maximum value $V_{max}(x)$ takes

$$v_{max} = \max\{V_{max}(0), \dots, V_{max}(X)\}.$$

Therefore the state of the system for $k = 0, \ldots, N_t$ is

$$\delta_k = (\tau_k, i_k, v_k, v_{k+1}),$$

where

$$\tau_k = \begin{cases} T - t_k & \text{if } t_k \leq T, \\ -1 & \text{if } t_k > T, \end{cases}$$

so τ_k represents the time left until the limiting time T is reached, and

$$i_k = \begin{cases} X - x_k & \text{if } x_k \leq X, \\ -1 & \text{if } x_k > X, \end{cases}$$

so i_k represents the distance left until destination X is reached. Finally,

$$v_k \equiv \text{ speed at time } \tau_k, \quad v_{k+1} \equiv \text{ speed at time } \tau_{k+1}.$$

Thus, k defines the backward stage, $i.e$ $k = 0$ defines the discretization point corresponding to time T and $k = N_t$ defines the discretization point corresponding to the initial time 0. Once the state space has been set we define the four possible controls that can be applied in each point of the state space,

$$\text{maximum acceleration } (a) \rightarrow p_k = P_{max}^k, q_k = 0,$$

$$\text{speed hold } (h) \rightarrow p_k = P_h^k \text{ s.t } \frac{dv_k}{d\tau} = 0, q_k = 0,$$

$$\text{coast } (c) \rightarrow p_k = 0, q_k = 0,$$

$$\text{maximum brake } (b) \rightarrow p_k = 0, q_k = Q_{max}^k.$$

where, as explained in Table 2, P_{max}^k is the maximum value that the traction power p_k can take in stage k due to the constraints (16)-(20), and respectively, Q_{max}^k is the maximum value that the braking power q_k can take in stage k. P_h^k denotes the power at stage k that is needed in order to keep the speed constant. Having determined the state and the control space we define the objective function for the DP approach. First, we define

$$J_k = \min_{u_k, u_{k+1}} \{J_{k-1} + m(\delta_k, u_k, u_{k+1})\},$$

which is the cost to reach destination starting at stage k, where $u_k = (p_k, q_k)$. Here m_k represents the immediate cost to transfer the system from stage k to $k-1$,

$$m(\delta_k, u_k, u_{k+1}) = \Delta_\tau \left(\frac{p_k(i_k) + p_{k+1}(i_{k+1})}{2} \right),$$

and J_{k-1} the cost from stage $k-1$ until destination in stage 0. This objective function can not be solved by backward recursion due to the presence of the control in stage $k+1$ which is not known. We adapt the objective function by

$$\tilde{J}_k = J_k + \frac{\Delta_t}{2} p_k,$$

obtaining the recursive formula

$$\tilde{J}_k = \min_{u_k \in \{a,h,c,b\}} \{\tilde{J}_{k-1} + \tilde{m}^{u_k}(\delta_k)\}, \tag{21}$$

where $\tilde{m}^{u_k}(\delta_k) = \Delta_\tau p_k(i_k)$, with $u_k = (p_k, q_k)$ and

$$a = (P^k_{max}, 0),\ h = (P^k_h, 0),\ c = (0, 0),\ b = (0, Q^k_{max}).$$

We will then denote by m^{u_k} the immediate cost to transfer the system from a state in stage k to a state in stage $k-1$. Observe that the immediate cost in the case of $u_k = c$ or b is 0.

The objective function in equation (21) subject to the constraints (16)-(20) form the DP model, which can now be solved by backward recursion. We will refer to this problem as (P2).

5.2 Backward recursion

To perform the backward recursion in (P2) we first need to define the immediate and the final costs.

- **Final Cost:** We denote by final states the points in the state space for which the process stops, there is no need to apply any control on them. By setting the cost of falling into these final states high or low we make sure the system ends in the right state. If the train reaches destination beyond the limiting time, *i.e*

  $$\delta_0^1 = (-1, 0, v_k, v_{k+1}), v_k > 0,$$

  or on time with positive speed, *i.e*

  $$\delta_0^2 = (\tau_k, 0, v_k, v_{k_1}), \tau_k \geq 0, v_k > 0,$$

  we set the final cost $\tilde{J}_0^1$ incurred by δ_0^1 and $\tilde{J}_0^2$ incurred by δ_0^2 high. We set the final cost $\tilde{J}_0^3$ to zero if it reaches destination on time, *i.e*

  $$\delta_0^3 = (\tau_k, 0, 0, v_{k+1}), \tau_k \geq 0,$$

  therefore,

  $$\tilde{J}_0 = \begin{cases} TE_{max}v_{max} & \text{if } \delta_0 = \delta_0^1, \delta_0^2 \\ 0 & \text{if } \delta_0 = \delta_0^3 \end{cases},$$

Once the immediate costs, m^{u_k}, and the final costs, $\tilde{J}_0$, have been set we can code the recursion, given by equation (21), in C++ to obtain the optimal speed profile for any possible starting δ_k. The way to proceed is summarized in Algorithm 1.

As said in the introduction to this section the Algorithm 1 carries a big computational cost, since for each state δ_k we need to keep the information of the optimal action in A_k and optimal cost of each strategy in arrays of size $N_t \times N_x \times N_v \times N_v$.

Algorithm 1 Backward Recursion

$\tilde{J}^1{}_0, \tilde{J}^2{}_0 \leftarrow TE_{max}v_{max}$
$\tilde{J}^3{}_0 \leftarrow 0$
for $k \geq 1$ **do**
  given δ_k compute $\delta_k^a, \delta_k^h, \delta_k^c, \delta_k^b$
  and compute $m^a(\delta_k), m^h(\delta_k), m^c(\delta_k), m^b(\delta_k)$
  if $\delta_k^a, \delta_k^h, \delta_k^c$ or δ_k^b brake a constraint **then**
    recompute $\delta_k^a, \delta_k^h, \delta_k^c$ or δ_k^b
    recompute $m^a(\delta_k), m^h(\delta_k), m^c(\delta_k), m^b(\delta_k)$
  end if
end for
for $k \leq N_t$ **do**
  $\min_{u_k \in \{a,h,c,d\}}\{\tilde{J}_{k-1} + m^{u_k}(\delta_k)\}$
  store $A_k \leftarrow u_k$ and $C_k \leftarrow \tilde{J}_k$
end for

5.3 Implementation and Results

In order to perform the backward recursion in problem (P2) we developed a C++ code with the procedure explained in the previous section and solve the DP model in a Core i3 @ 3.2 GHz, 4GB AM 1333 MHz DDR3 computer. Note that by using this approach: (1) we do not obtain a good accuracy in the solution unless we take a very fine discretization of the time, space and speed intervals, incurring in a huge memory capacity (around $3 \times N_t \times N_x \times N_v^2$ bits) and that (2) the finer the discretization is the longer it takes to compute the solutions (around $1\mathrm{e}^{-5} \times N_t \times N_x \times N_v^2$ seconds). This approach will then be feasible for short journeys. We again take a great variety of examples of real profiles provided by INGETEAM S.A. and other examples in the literature and we compare the solutions with those obtained in the direct approach that was proposed in this report. Based on several experiments we have observed that the solution obtained solving (P2) converges to the solution obtained in (P1) as the discretization of the state space becomes finer, see an example in Figure 3. The DP solution presented in Figure 3 took around an hour of computation.

The accuracy obtained by the direct approach by solving (P1) could not be obtained by the approach proposed in Section 5 due to its memory requirements. In order to attain accuracy of this method more research is needed as for instance, the states that we know will not be reached in the recursion could be eliminated or we could leave the profiles that are unprovable aside, this would reduce the computational cost considerably.

6 NUMERICAL AND SIMULATIONS RESULTS

In this section, we report the simulation results we have obtained. As explained in Section 4 and 5, we first solve problem (P1) using AMPL/Ipopt, and later we solve (P2) by a software package developed in C++. Both results, the speed profiles obtained solving (P1) and the ones obtained by solving (P2), coincide for a fine discretization of the problem (P2). We have verified this for several real case examples. As a second step we use a train simulator developed by INGETEAM S.A. to quantify the reduction in energy obtained by our approach in two

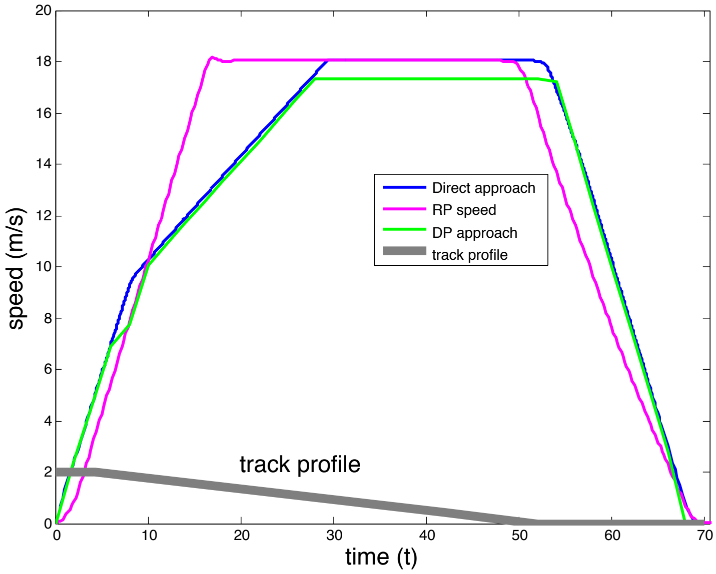


Figure 3: Comparison of the solution of problem (P1) and (P2).

different rides of a metro in a major city in Europe.

The main conclusions regarding the numerical results are the following:

- Our model (due to the additional constraints), (8)-(11), captures more accurately the dynamics of a train than existing models in the literature.
- The solution of (P1) is always formed by four possible actions which can be summarized in Table 2.
- The computation time required to solve (P1) for real examples is below 8 seconds, thus our approach represents a realistic solution that can be implemented in real systems.
- The computational time to solve (P2) is

  $$1\mathrm{e}^{-5} \times N_1 \times N_2 \times N_3^2 \text{ s.}$$

  For the example presented in Figure 3 the required time for an accurate solution would be 2 days an a half.
- The memory required to save all the optimal strategies is

  $$3 \times N_1 \times N_2 \times N_3^2 \text{ bits.}$$

- The energy reduction obtained is larger in tracks that present downhill sections.
- In real examples the energy reduction varies between 8% and 25%.

Table 3: Energy reduction compared to real profiles

Section	LAAJR energy	RP energy	Reduction
example 1	15.1485 kWh	16.6101 kWh	8.7995 %
example 2	15.2289 kWh	19.897 kWh	23.4613 %

6.1 Simulations

Simulations carried out in the simulator developed by INGETEAM S.A. allow us to (i) compare the structure of our profiles with real examples and (ii) to measure the reduction in energy that our solution would provide. The procedure with the simulator is the following; we provide the speed profile obtained with Ipopt and the simulator gives the power profile that a train would need to complete the journey determined by this speed and computes the energy consumption.

In Figures 5 and 6 we consider two different real examples from a metro in a major city in Europe. The track in the first example (example 1) presents steep uphill sections in most of the track, see Figure 1, and the track in the second example (example 2) presents a flat track with a steep downhill section in the end, see Figure 4. We compare the speed and energy consumptions obtained with the simulations with the profile used in reality and with that obtained by solving Problem (P1) and (P2).

In Table 3 we represent the energy consumption for the two trips considered. The results show that the solution profile obtained by solving (P1) and (P2) provide a larger reduction in energy in tracks that present downhill sections, i.e., these tracks are more sensitive, in view of energy efficiency, to a good switching strategy. The energy reduction varies between 8% and 25% depending on the track considered.

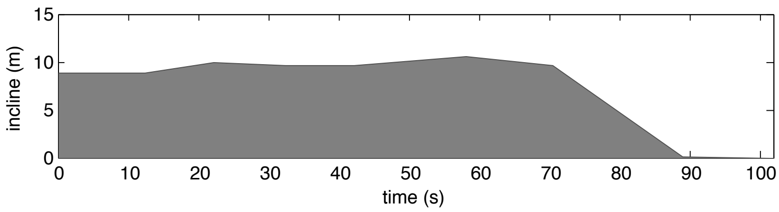
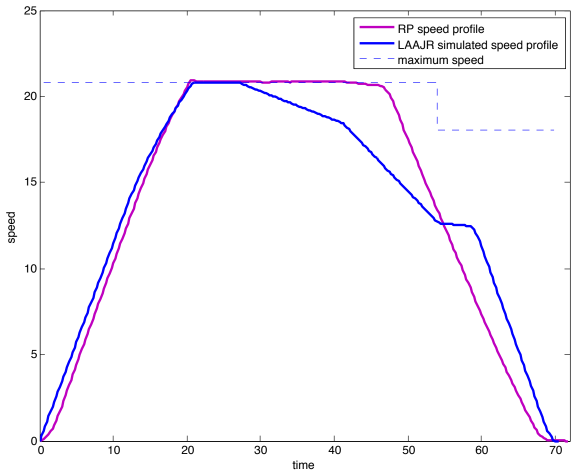


Figure 4: Track profile for example 2.

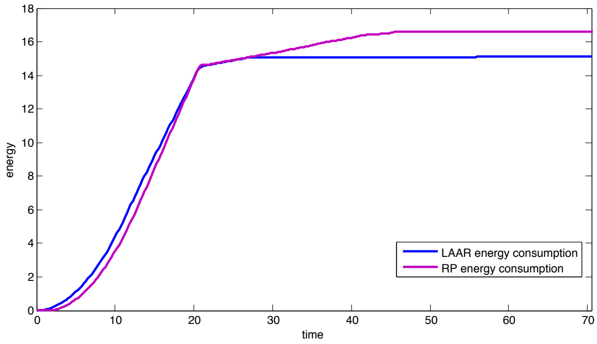
7 CONCLUSIONS AND FUTURE WORK

This paper proposes a robust model for the problem of finding the power and speed profiles that minimize the energy in short railway journeys. We propose two numerical methods to handle the difficult constraints that this problem presents avoiding simplifications in the model as it has been done in the literature.

The efficiency of this methods allow a real application in typical trips of metro systems, being able to: (1) compute new profiles given any circumstance by the approach proposed in Section 4 , and (2) compute off line all the optimal profiles for any possible circumstance by the approach proposed in Section 5. Our study shows that the method presented in Section 4

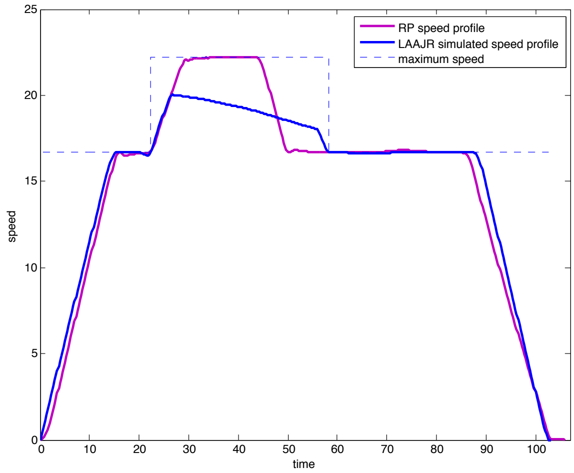


(a) Speed profiles.

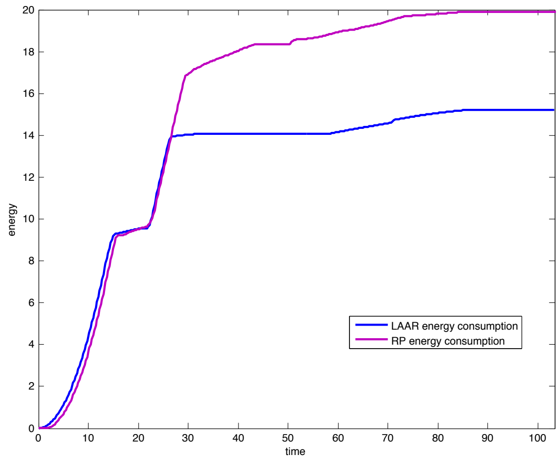


(b) Cumulative consumption.

Figure 5: Simulations for example 1.



(a) Speed profiles.



(b) Cumulative consumption.

Figure 6: Simulations for example 2.

could be implemented in a real system immediately, whereas more research is required in order to solve real-life problems with the method of Section 5. Besides, the simulations have shown to obtain a great reduction in the consumption of energy of around 8-25%.

8 ACKNOWLEDGMENTS

The authors gratefully acknowledge the financial support of INGETEAM Traction S.A. during the development of this research project and also Albert Gonzalez Manresa (INGETEAM Power Technology - Traction) for the simulations carried out.

References

[1] A. E. Bryson, Jr., Y.C. Ho, Applied Optimal Control: Optimization, Estimation and Control, *Hemisphere Wiley*, 1975.

[2] A. Grünig, Efficient Generation of Train Speed Profiles, Bachelor's Thesis, Institute for Operations Research, ETH Zurich, 2009.

[3] A. Wächter, L. T. Biegler, On the Implementation of a Primal-Dual Interior Point Filter Line Search Algorithm for Large-Scale Nonlinear Programming, *Mathematical Programming*, 106(1), pp. 25-57, 2006.

[4] H. Ko, T. Koseki and M. Miyatake, Application of Dynamic Programming to the Optimization of the Running Profile of a Train, *Computers In Railways IX*, vol. 15, pp. 103-112, 2004.

[5] J. T. Betts, Practical Methods for Optimal Control Using Nonlinear Programming (Advances in Design and Control), *Society for Industrial* & *Applied Mathematics*, 2nd edition edition , 2009.

[6] M. Domínguez, A. P. Cucala, A. Fernández, R.R. Pecharromán, J. Blanquer, Energy Efficiency on Train Control: Design of Metro ATO Driving and Impact of Energy Accumulation Devices, *9th World Congress on Railway Research - WCRR 2011*, Lille, France, 2011

[7] P.G. Howlett. An optimal strategy for the control of a train *Journal of Australian Mathematical Society Series B*, 31 (1990), pp. 454471.

[8] P.G. Howlett, P.J. Pudney, An Optimal Driving Strategy for a Solar Powered Car on an Undulating Road *Dynamics of Continuous, Discrete and Impulsive System.*, 4:553-567, 1998.

[9] P. G. Howlett, P. J. Pudney, Energy-efficient Driving Strategies for Long-Haul Trains, *Proceedings of CORE 2000 Conference on Railway Engineering*, Adelaide, 21-23 May 2000.

[10] P. G. Howlett, P. J. Pudney, Energy-efficient Train Control. *Springer-Verlag London* Ltd., 1995.

[11] R. Fourer, D. M. Gay and B. W. Kernighan, A Modeling Language for Mathematical Programming. *Management Science* 36 (1990) 519-554.

[12] W. Zhang, T. Inanc, A Tutorial for Applying DMOC to Solve Optimization Control Problems, *Systems Man and Cybernetics (SMC), 2010 IEEE International Conference*, 1857 - 1862, Istanbul, 2010.

[13] X. Vu (2006). Analysis of necessary conditions for the optimal control of a train. *Ph.D. thesis*, University of South Australia (see also www.amazon.co.uk/Analysis-necessary-conditions-optimal-control/dp/3639120000).