

Document downloaded from:

<http://hdl.handle.net/10251/126942>

This paper must be cited as:

Macian Martinez, V.; Guardiola, C.; Pla Moreno, B.; Reig, A. (2018). Application and benchmarking of a direct method to optimize the fuel consumption of a diesel electric locomotive. Proceedings of the Institution of Mechanical Engineers Part F Journal of Rail and Rapid Transit. 232(9):2272-2289. <https://doi.org/10.1177/0954409718772133>



The final publication is available at

<http://doi.org/10.1177/0954409718772133>

Copyright SAGE Publications

Additional Information

Application and benchmarking of a direct method to optimize fuel consumption of a diesel electric locomotive

V Macian, C Guardiola, B Pla, A Reig

CMT-Motores Térmicos, Universitat Politècnica de València, Spain.

Abstract

This paper addresses the optimal control of a long-haul passenger train to deliver minimum fuel operation. Contrarily to the common Pontryagin minimum principle approach in railroad related literature, this work addresses this optimal control problem with direct method optimization, whose use in this field is still marginal. A particular implementation of a direct method based on Euler collocation scheme and its transcription into a nonlinear problem is described in detail. In this paper, this optimization technique is benchmarked with well-known optimization methods in literature, namely dynamic programming and Pontryagin minimum principle, simulating a real route. The results showed that direct methods are on the same level of optimality than other algorithms while requiring reduced computational time and memory, and being able to handle very complex dynamic systems. Direct method performance is also compared to the real trajectory followed by the train operator exhibiting up to 20% of fuel saving in the example route.

Index terms— optimal control; direct method; diesel electric locomotive; speed control

Abbreviations

DM	Direct Method
DP	Dynamic Programming
HJB	Hamilton-Jacobi-Bellman equation
HVAC	Heating, Ventilating and Air Conditioning
IM	Indirect Method
MINLP	Mixed Integer Nonlinear Programming
NLP	Nonlinear Programming
NP	Nondeterministic Polynomial time problem
OCP	Optimal Control Problem
ODE	Ordinary Differential Equation
PMP	Pontryagin Minimum Principle

1 Introduction

Optimal control provides tools to operate a system in the most efficient way it can behave. However, either it is generally too hard to be implemented or the system is not clear enough to define a proper optimal control problem (OCP).

Train operation is a field in which optimal control is especially attractive for several reasons: the problem to optimize is generally well known, it is usually applied many times under similar conditions and there may be enough time to solve (and recalculate if necessary) an OCP. In particular, diesel electric locomotives are an interesting target since their performance (especially prime mover's) is strongly affected by the way they are operated.

There is an extensive literature in optimal control for trains from the late 60's, when [1] first introduced the optimal control in railroad transportation, particularly for diesel locomotives, which was lately followed by [2]. The topic

The authors acknowledge the support of Spanish Ministerio de Economía, Industria y Competitividad through project TRA2016-78717-R.

gained interest especially during the last decade [3], with a variety of different approaches being used. Nowadays, works are spread over two main fields of optimization: scheduling and routing of trains [4–7], and efficient operation throughout the tracks. Regarding the latter, several works present heuristic control algorithms [8–10], but most of the literature focuses on optimal control theory, with different applications of the Pontryagin minimum principle (PMP) [11–16] or the dynamic programming method (DP) [17, 18]. Some of these works focus on discrete control locomotives optimization: [19] proposed an algorithm based on finding switching times for different notch settings, introducing speed limits in [20], track slope in [21] and a varying slope in [22]. All these PMP theory-based are founded around what some authors call the *necessary conditions for optimality*. In short, the common methodology is to apply PMP theory analytically to some extent and, then, deduct from the formulation a set of predefined operation modes (such as constant speed, full acceleration, coasting, etc.). At the end, the idea is to reduce the optimal control to the choice of the best operation mode at each moment. Of course, such a simple control scheme is only possible when using strongly simplified models; detailed train models are unapproachable with that methodology. Implementation of constraints and disturbances (track grade, speed limits, etc.) is significantly complex and, in some cases, impossible.

Direct methods (DM) in literature are marginal and have been applied on few works, mainly [23–25]. Those are based on the pseudospectral method to address ordinary differential equations (ODEs) and transcribing the OCP into a linear problem. These approaches permit a better handling of constraints but, still, the train model must be strongly simplified due to the problem linearization. Also, in those particular works, the computational requirements were significantly higher to those of PMP-based approaches.

The present paper tries to overcome the gaps found in the above works: (i) the requisite of a strongly simplified model, (ii) the complexity/inability to handle a number of constraints, and (iii) the increased computational requisites of powerful optimizers. In order to do so, this paper introduces the application of a DM to solve the

train OCP. Therefore, the contributions of this work are the ability of the presented methodology to:

1. Compute the optimal operation of the locomotive for a train model of nearly any complexity as long as DM can consider any type of dynamic model (the presented model is deliberately simple to provide a fair comparison to other optimization algorithms, but the same methodology applies for more complex dynamic models).
2. Include any kind of constraint in a straightforward manner, including slopes, speed limits, traffic, schedule, state constraints, etc., since DM intrinsically allow to include constraints explicitly in their formulation.
3. Find the optimal solution with little computational requirements that allows to quickly recalculate the optimal train trajectory if the track conditions have changed, and to do so in embedded systems. The use of Euler collocation scheme results in a very light nonlinear problem that can be handled efficiently.

In contrast to the existing literature, this method is based on the Euler’s collocation scheme, transcribing the original OCP into a nonlinear problem. This is achieved with an *ad hoc* framework designed to approach this kind of OCPs. Its performance is benchmarked against the two optimal control theory methods most commonly used in literature, DP and PMP. The efficiency of DM control algorithm on a long-haul route will be compared to the real operation of the train to quantify the potential fuel savings that optimal control offers.

The paper is organized as follows: first of all the diesel electric locomotive model to be used throughout this work is described and the corresponding OCP is stated. Then, the DM is introduced in detail followed by a brief description of the benchmarking methods: DP and PMP. Next, the performance on a real long distance route is shown and compared to the real operation of the train and, finally, some conclusions on the results are drawn.

2 Simulation tools

2.1 Train set model

The modeled train is a four axle Bo'Bo' type diesel electric locomotive coupled to 8 coaches with a total weight of 505 tonnes. The prime mover develops up to 2.4 MW with 8 different notch settings and an idle position. Brakes are of electro-pneumatic type with forced convection resistors on top of the locomotive.

Due to the inertia of the system as well as the low frequency of actuations, the prime mover can be modeled following a quasi-static approach. The power output at the wheels is a function of the notch setting ξ , so $P = P(\xi)$ as shown in figure 1. The fuel consumption is also a function of the notch setting, $F = F(\xi)$, as indicated in figure 2. These values are also listed in table 1. This approach considers only the fuel consumption required to deliver a traction effort; auxiliary devices such as compressors, HVAC and other peripherals are not considered for simplicity. Note also that no recuperation of brake energy is available on board for this type of locomotive, instead all braking energy is dissipated in resistors (no regenerative electrodynamic braking). Therefore, the locomotive is assumed to deliver a given constant power output and fuel consumption for each notch setting. This quasi-steady approach neglects engine dynamics which, on the one hand results in a simpler model that may be accurate enough for systems with slow dynamics (in fact this approach is widely used in railway control literature and state-of-the-art algorithms), but on the other hand could bring overestimation of engine efficiency under quite dynamic situations. This effect will be discussed later in the paper.

Accordingly, the locomotive tractive effort is a function of the power output, P , and the speed, v :

$$T_{trac} = \frac{P}{v} \quad (1)$$

This force cannot exceed the maximum tractive effort of electric motors $\bar{T}_{trac}(v)$ (maximum traction that motors can deliver by design due to temperature and integrity reasons), which are shown in figure 3.

Drag and track resistance are calculated with a modified form of the original *Davis* equation [26] as a function of speed v , mass m , track slope α

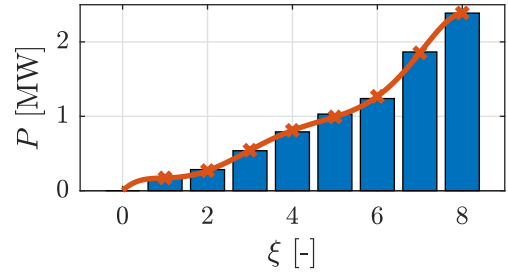


Figure 1: Prime mover's output power at wheels for each notch setting ξ . Zero notch is idling position (no power is forwarded to wheels). Red crosses are values fitted to the first three terms of a Fourier series.

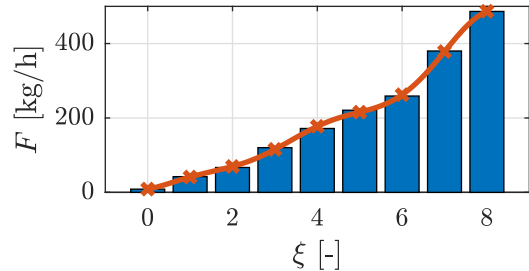


Figure 2: Prime mover's instantaneous fuel consumption for each notch setting ξ . Zero notch is idling position. Red crosses are values fitted to the first three terms of a Fourier series.

and radius R , according to locomotive manufacturer information:

$$T_{res} = A + Bv + Cv^2 + mg(1000\alpha + D/R) \quad (2)$$

where A , B , C and D are coefficients depending on the characteristics of the train set— A is a constant friction term, B and C are part of a parabolic model for speed-dependent friction, and D accounts for additional friction when track is not a straight line. The particular values for these coefficients are shown in table 2—they are fitted by the locomotive manufacturer. Note that the introduction of α as a per mille term instead of as $\sin \alpha$ introduces an error $< 0.002\%$ for the values considered in this work, which is negligible and simplifies the expression. The validity of this equation has been discussed in the past, especially for high-speed trains. Works like [27] have proposed alternative forms of the original Davis equation to even include energy dissipation by

ξ [-]	P [MW]	F [kg/h]
0	0	8.6
1	0.16	42
2	0.28	67
3	0.54	120
4	0.79	172
5	1.03	221
6	1.24	259
7	1.87	380
8	2.39	486

Table 1: Tabulated values of prime mover’s power output and fuel consumption for the different notch settings.

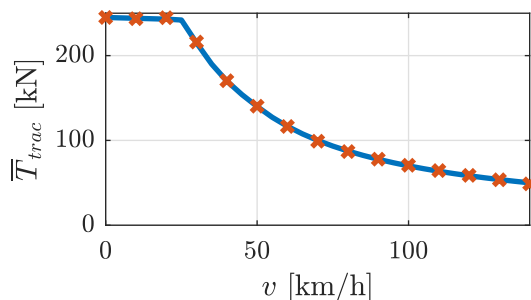


Figure 3: Electric motors maximum tractive effort as a function of locomotive speed. Red crosses are values fitted to a sum of three Gaussian functions.

hysteresis and through dumpers. However, the modified form shown in (2) is still well-accepted for long-haul freight trains and commonly used in control-oriented works. Therefore, in line with the current literature, (2) will be used for the purpose of this paper.

Electro-pneumatic brakes can be actuated to any value between release position ($T_{brk} = 0$) and maximum braking force ($T_{brk} = \bar{T}_{brk} = 380$ kN).

Train set dynamics are the balance of the above forces:

$$\dot{v} = \frac{T_{trac} - T_{res} - T_{brk}}{m} \quad (3)$$

Note that the effect of rotating masses is neglected in the above dynamic equation. However, the inertia of mainly axes and motors is low compared to the gross mass of the train set (505 tonnes).

A	7703	N
B	157.14	N/(m/s)
C	8.68	N/(m ² /s ²)
D	0.7	m

Table 2: Davis equation coefficients used in this work. These values are fitted by the locomotive manufacturer according to their measurements.

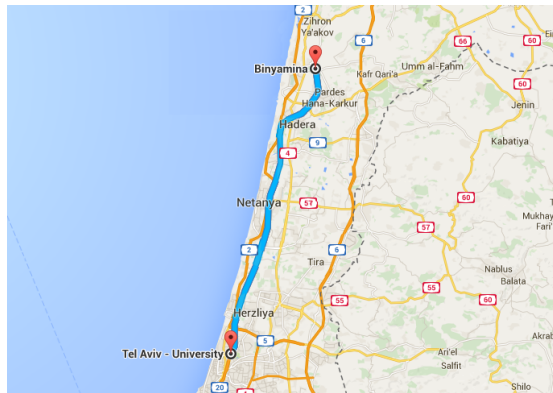


Figure 4: Benchmarking route between Binyamina and Tel Aviv University stations (Israel). Source: Google Maps

2.2 Benchmarking problem

A benchmarking problem is proposed to evaluate the performance of the direct method (DM) against dynamic programming (DP) and indirect method (IM), on the current train OCP. Fuel consumption, constraint violation, computational requirements (memory and time) and ease of use will be compared in the following sections.

This benchmarking problem consists of a 50 km track between Binyamina and Tel Aviv University stations in Israel as shown in figure 4. The most relevant data of this route (as provided by locomotive manufacturer) is summarized in table 3 whilst track radius and slope profiles are shown in figure 5. The specifications of the benchmarking computer are shown in table 4.

3 Optimal control problem

The formulation of an optimal control problem consists of a cost index \mathcal{L} to be minimized subject to several constraints. The common formulation in literature is expressed in terms of states

Total distance [km]	50
Available time [min]	27
Speed limit [km/h]	140
Elevation gain [m]	5
Maximum elevation difference [m]	45

Table 3: Summary of benchmarking route between Binyamina and Tel Aviv University stations (Israel). Source: locomotive manufacturer.

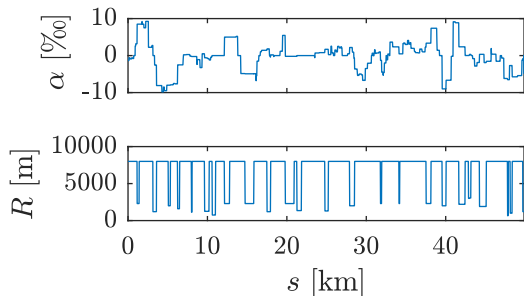


Figure 5: Track slope and radius profiles of benchmarking route between Binyamina and Tel Aviv University stations (Israel). Note that radius is saturated at 8000 m for numerical reasons (the difference in resistance between $R=8000$ and $R=\infty$ is negligible, about 0.02% at 100 km/h). Source: locomotive manufacturer.

x , actuators u and a problem domain s (in the case at hand, distance):

$$\mathcal{L} = \int_0^S L(x, u, s) ds \quad (4)$$

such that:

$$\frac{\partial x}{\partial s} = f(x, u, s) \quad (5)$$

with boundary conditions:

$$\begin{aligned} x(0) &= x_0 \\ x(S) &= x_s \end{aligned} \quad (6)$$

path constraints:

$$g(x, u, s) \leq 0 \quad (7)$$

and integral constraints:

$$\int_0^S N(x, u, s) ds = 0 \quad (8)$$

CPU	Intel Core i5-4440 @ 3.10 GHz
Memory	8 GB DDR3
OS	Ubuntu 14.04
Software	MATLAB R2015a with <i>ad hoc</i> direct optimization framework

Table 4: Benchmarking computer specifications

3.1 Application to the train optimal control problem

The OCP consists in minimizing the cost index \mathcal{L} defined as the fueling rate of the prime mover along a trajectory of length S :

$$\mathcal{L}_s = \int_0^S \frac{F(\xi)}{v} ds \quad (9)$$

Note that fueling rate $F(\xi)$ is measured in mass of fuel per unit of time.

Like in any OCP, the system is subject to several constraints. First of all, the train must cover the distance S within a time T (integral constraint):

$$\int_0^S \frac{1}{v} ds = T \quad (10)$$

Initial and final speeds are fixed since the train must be stand still at the departing and arriving platforms (boundary constraints):

$$\begin{aligned} v(0) &= 0 \\ v(S) &= 0 \end{aligned} \quad (11)$$

Track speed limits have to be respected (path constraint):

$$v(s) \leq g(s) \quad (12)$$

Limits of prime mover (maximum power output \bar{P}), traction motors (maximum tractive effort \bar{T}_{trac}) and brakes (maximum braking force \bar{T}_{brk}) are also considered:

$$\begin{aligned} P(s) &\leq \bar{P} \\ T_{trac}(s) &\leq \bar{T}_{trac}(v(s)) \\ T_{brk}(s) &\leq \bar{T}_{brk} \end{aligned} \quad (13)$$

In addition, there are two disturbances to the problem that affect the dynamic response of the system, namely the track slope $\alpha(s)$ and radius

$R(s)$. These are included in the traction effort calculation.

Therefore, the OCP consists in minimizing \mathcal{L} (9) throughout a trajectory of length S , fulfilling the constraints (10)-(13).

4 Direct method application

DM consist in approximating the optimal state and actuator trajectories to known functions such as polynomials, piecewise functions, or functions of any other form. The original OCP can be then transcribed into a large and extremely sparse nonlinear programming (NLP) problem whose unknowns are the coefficients of these functions [28–32]. In other words, this family of methods first discretize and then optimize the OCP. Thanks to the efficiency of nowadays sophisticated NLP optimizers, DMs are powerful methods. In fact, OCPs that have been traditionally unaffordable with other methods (either because of computational or mathematical aspects), for instance the complete optimization of all actuators in an internal combustion engine [33] or the calculation of minimum-fuel orbit transfer for a small spacecraft [34], can be approached with a transcription into an NLP problem.

The current paper is focused on Euler’s collocation method to transcribe the ODEs into a large and sparse NLP. Collocation is a numerical method to solve ODEs and consists in splitting the domain of the problem (space in this case) into n subintervals $[s_i, s_{i+1}]$ (collocation points) of length $h = S/n$, and approximating the state and actuator trajectories to known functions (generally polynomials) with a finite number of coefficients. Then, these functions are evaluated at the collocation points to solve the unknown coefficients [32]. For this particular OCP, state and actuator trajectories are approximated to a finite set of discrete values:

$$\begin{aligned} x &\approx [x_0, x_1, \dots, x_{n-1}, x_n] \\ u &\approx [u_0, u_1, \dots, u_{n-2}, u_{n-1}] \end{aligned} \quad (14)$$

where $x_i = x(s_i)$ and $u_i = u(s_i)$. ODEs can be approximated by any numerical method such as Euler or any Runge-Kutta method. For the sake of simplicity the Euler method is used in this work. Therefore, ODEs are transcribed into:

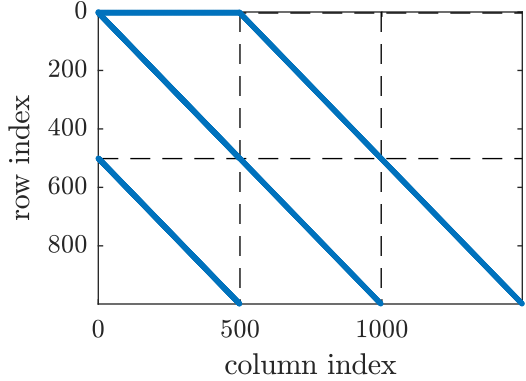


Figure 6: Jacobian matrix of the example OCP. Blue dots are the non-zero elements of the matrix. Quadrants are, from left to right: speed (state), notch setting and braking force (actuators) collocation points; from top to bottom: initial, final and integral constraints (thin upper rows), system dynamics, tractive effort limit (path constraint) and discretizing function (path constraint).

$$\begin{aligned} \frac{\partial x}{\partial s} &\approx \left[\frac{\partial x_0}{\partial s}, \dots, \frac{\partial x_{n-1}}{\partial s} \right] \\ \frac{\partial x_i}{\partial s} &= \frac{x_{i+1} - x_i}{h} \end{aligned} \quad (15)$$

and accordingly system dynamics end up as:

$$\frac{x_{i+1} - x_i}{h} = f \left(\frac{x_i + x_{i+1}}{2}, u_i \right) \quad (16)$$

Therefore, ODEs and constraints may be transcribed and arranged as a large and sparse NLP with unknowns $\mathbf{x} = [x_0, \dots, x_n, u_0, \dots, u_{n-1}]$, which can be solved with an NLP solver. In this work, IPOPT [35] (an open source NLP solver based on the interior point method available to download at its website) was used to solve the NLP. As an example, the Jacobian’s structure of the NLP corresponding to the train OCP is depicted in figure 6 to show the high level of sparsity of DM (blue dots are non-zero elements).

NLP solvers require the objective function $\mathcal{L}_s(\mathbf{x})$, the constraints $c(\mathbf{x})$ and generally first derivatives, namely the gradient $g(\mathbf{x})$ of the objective function:

$$g_i = \frac{\partial \mathcal{L}_s(\mathbf{x})}{\partial \mathbf{x}_i} \quad (17)$$

and the Jacobian $J(\mathbf{x})$ of the constraints:

$$J_{i,j} = \frac{\partial c_i(\mathbf{x})}{\partial \mathbf{x}_j} \quad (18)$$

Some solvers also accept second derivatives, i.e. the Hessian:

$$H_{i,j} = \frac{\partial^2 \varphi(\mathbf{x})}{\partial \mathbf{x}_i \partial \mathbf{x}_j} \quad (19)$$

where $\varphi(\mathbf{x})$ is the objective function $\mathcal{L}_s(\mathbf{x})$ or any of the constraints $c(\mathbf{x})$. The availability of second derivatives may improve the convergence of the NLP solver.

Formally, functions $\mathcal{L}_s(\mathbf{x})$ and $c(\mathbf{x})$ must be twice continuously differentiable for the collocation method to be applied. However, this is not a *hard* requisite from a numerical point of view; discrete or non-differentiable functions (such as maps, piecewise functions, etc) might be included providing numerical derivatives and substituting non-differentiable points with arbitrary finite values. Although this can be done with some success, providing continuous functions is recommended as long as the NLP solver convergence improves with function smoothness.

Therefore, discrete functions $P(\xi)$, $F(\xi)$ and $\bar{T}_{trac}(v)$ from the train model are fitted to continuous expressions. Prime mover's power output P and fuel consumption F are represented by the first 3 terms of a Fourier series, as shown in figures 1 and 2:

$$P(\xi) \approx \frac{p_{10}}{2} + \sum_{i=1}^3 \left[p_{1i} \cos\left(\frac{2\pi i \xi}{p_3}\right) + p_{2i} \sin\left(\frac{2\pi i \xi}{p_3}\right) \right] \quad (20)$$

$$F(\xi) \approx \frac{f_{10}}{2} + \sum_{i=1}^3 \left[f_{1i} \cos\left(\frac{2\pi i \xi}{f_3}\right) + f_{2i} \sin\left(\frac{2\pi i \xi}{f_3}\right) \right] \quad (21)$$

and maximum tractive effort is fitted to the sum of three Gaussian functions, which can be appreciated in figure 3:

$$\bar{T}_{trac}(v) = \sum_{i=1}^3 t_{1i} e^{-\left(\frac{v-t_{2i}}{t_{3i}}\right)^2} \quad (22)$$

Note that the notch setting actuator ξ is a discrete quantity but in the context of an NLP it should be considered as a continuous variable, otherwise objective or constraint functions will not be continuous.

Although all variables have been approximated to continuous functions, the train OCP is actually a problem combining continuous and integer variables, whose transcription into an NLP is known as mixed integer NLP (MINLP). There are several solvers available to manage a MINLP such as BONMIN [36] or NOMAD [37]. All these algorithms combine the solution of the equivalent relaxed NLP (NLP obtained after removing the constraints such that some variables are integer) with an integer programming technique such as the cutting plane or the branch and bound methods. Unfortunately, MINLPs are considered to be NP-hard in computational complexity theory, i.e. at least as hard to solve as the hardest nondeterministic polynomial time (NP) problem (a class of problems that can be solved in polynomial time with a nondeterministic Turing machine), which generally means that an algorithm would need super-polynomial time (e.g. exponential time as a function of the number of inputs to the problem) to find a solution or even that an algorithm simply does not exist and only suboptimal guesses can be achieved [38]. Therefore, the fact that MINLPs are NP-hard informally means that any algorithm to approach the solution in that class of problems might be potentially time consuming. Particularly, the above MINLP solvers (and generally any integer programming algorithm) take prohibitive computational times for a large MINLP with thousands of integer variables such as the one at hand.

There are several possible workarounds for such a large MINLP. The straightforward solution is to round the variables with integer constraints from the result of the relaxed NLP to the nearest integer value. However, neither it is possible to guarantee that constraints will be fulfilled with this solution nor its optimality. Another solution is to include some kind of *ad-hoc* heuristics to the relaxed NLP in order to iteratively search for the appropriate integer values. Constraints can be fulfilled but, again, optimality cannot be guaranteed with this solution.

In this paper a third workaround is proposed. It consists of three steps: first, the relaxed NLP

is solved; second, the variables with integer constraints are rounded to the nearest integer value; and third, a new NLP is constructed by substituting variables with integer constraints for the rounded trajectories from step two as fixed disturbances. Therefore, the first NLP provides an initial estimation about where the appropriate values of the integer variables are, while the second guarantees constraints fulfilling as well as an optimal solution for those given integer trajectories. Note that although this last solution is optimal, the fact that integer trajectories are fixed introduces suboptimality; however, the impact on the performance of the system is small compared to the optimal solution as it will be verified in section 6.

The main advantages that DM shows are:

- Handling of large problems with many states and actuators in a compact way.
- Profiting of state-of-the-art NLP solvers.
- Use of first and second derivatives of OCP objective function and constraints adding knowledge to the problem and speeding up the algorithm.
- Lower memory and time requirements than dynamic programming.
- Robustness and (often) convergence after few iterations.
- Use of previous solution of similar problems as a seed to improve speed.
- Implementation of state constraints directly in the formulation.

However some troubles can arise when solving OCPs with DM, especially when dealing with discrete quantities or a poor initial solution:

- NLP solvers give a local optimum which may differ from the global optimum.
- Convergence and solution may depend on initial seed.
- Extremely nonlinear problems might complicate the algorithm convergence.
- OCP cost index and constraint functions must be twice continuously differentiable, which might be troublesome for some particular problems.
- First and, especially, second derivatives could be difficult to calculate in some situations (numerical derivatives can be used but it may impact on algorithm performance).

- Discrete quantities are difficult to include in the optimization; MINLP can be used but OCP is much more expensive to solve.

This direct collocation method has been used to transcribe the train OCP into an NLP. Then, the problem is solved with IPOPT (interior point method) twice, one for the relaxed NLP and another time with fixed integer trajectories. Results of this optimization are depicted for the benchmarking route in figure 7, showing a strong acceleration at the beginning (notch 7, since notch 8 has lower efficiency), cruising around 130 km/h (between notch 2 and 6), then coasting for several kilometers (and eventually cruising with notch 1) and finally arriving to the platform with a strong braking at the very last moment, using the full available braking effort.

5 Methods for dynamic optimization

There exist three main families of numerical methods to calculate the optimal set of actuators that minimize a given cost index over the time [32, 39]: the Hamilton-Jacobi-Bellman (HJB) equation, whose main application is the dynamic programming algorithm (DP); indirect methods (IM) commonly applied as the Pontryagin minimum principle (PMP); and direct methods (DM), which have already been applied to the train OCP in the above section.

5.1 Dynamic programming

The DP algorithm is the numerical implementation of the HJB equation which is a sufficient condition for a global optimum. Although DP is not very common in railroad-related problems with some exceptions like [17, 18], in the automotive field it is widely used, especially in works like [40–44]. Informally, it consists in discretizing the problem into n domain steps (in this case steps in distance) and gridding the states (\mathcal{X}) and controls (\mathcal{U}) space to build a finite set of possible trajectories between consecutive time steps. Then, the single-step problem at a given step (distance) s_i is:

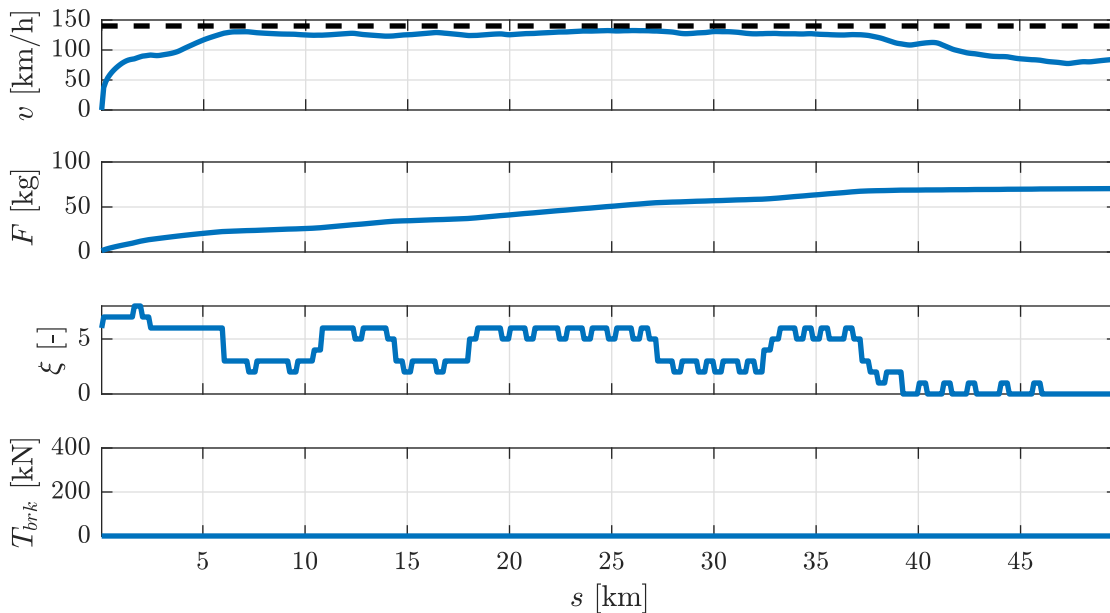


Figure 7: Train OCP solution calculated with DM. Main quantities are displayed: speed (state), accumulated fuel consumption (cost index), notch and braking force (actuators). Dashed line is the track speed limit.

$$L_s^*(s_i) = \min_u \left\{ L_s(\mathcal{X}, \mathcal{U}, s_i) + \sum_{k=i+1}^n L_s^*(s_k) \right\} \quad (23)$$

whose solution consist in the selection of the minimum among a finite set of candidates for L_s^* . Note that the minimization must be performed backwards to compute the sum in parallel. Therefore the OCP can be solved iteratively by evaluating the single-step problem from $s_n = S$ to $s_0 = 0$. The interested reader may find an extended introduction to DP at [45].

This method shows several advantages:

- It provides a sufficient condition, so the solution to the OCP is the global optimum.
- It is able to deal with systems of any complexity.
- It is *differential-less*, i.e. partial differential of states or actuators are not needed, which permits handling discontinuous systems.
- It is discrete by definition, so discrete systems are well handled.
- It is suitable for closed loop control as a space of optimal trajectories are precomputed.

However, some difficulties arise when dealing with it, especially with problems with many states and/or actuators:

- It suffers from the so called *curse of dimensionality*. The number of candidates to evaluate at any single step is $L_N = \prod_{N_x} X_{N,i} \cdot \prod_{N_u} U_{N,j}$ (where N_x and N_u are the number of states and actuators, and X_N and U_N the number of discrete points of each gridded state and actuator) which quickly rises as the number of states/actuators or the density of their grids increase. In practice it means that problems with $N_x + N_u \gtrsim 5$ involve prohibitive computational times.
- The discrete nature of the method generally prevents from fulfilling equality constraints, requiring an acceptance threshold.

The OCP was solved with a comprehensive DP algorithm implementation for MATLAB by Sundström and Guzzella [46]. The DP solution for the train OCP at the benchmarking route can be appreciated at figure 9 (in green). Grid discretization and a list of states and actuators included in the formulation of the problem may be found in table 5. Note that elapsed time,

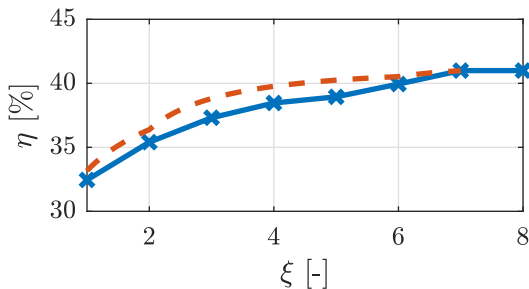


Figure 8: Prime mover’s efficiency for each notch setting ξ , in blue. Red dashed line represent the virtual efficiency curve resulting of switching from notch position 7 to null with different weights.

albeit it is an integral constraint of the form $\widehat{N} = \int_0^S N(x, u, t) dt$, has been included as a state $\partial x / \partial s = N(x, u, t)$ with conditions $x(0) = 0$ and $x(S) = \widehat{N}$ as a straightforward solution.

The solution for the benchmarking route roughly shows three phases: maximum acceleration, cruise speed and coasting. This is a very common speed profile for dissipative systems as it can be appreciated [47] (i.e. a system that can only spend energy on dissipative forces with no chances to regenerate energy back). Accordingly, for the first 4 minutes the notch setting stays at position 7 until track limit speed is reached; afterwards, the speed is held as close to track limits as possible by switching from notch setting 7 (the most efficient position, check figure 8) to null (the position spending the least fuel). Finally, notch is set to idle position and the train coasts until reaching the station, where brakes are applied to stop the train.

Note that switching between notch settings 7 and null when holding speed allows to virtually operate at an efficiency level resulting from the linear combination of fuel consumption and power output of both notch settings. This virtual efficiency (red dashed line in figure 8) is higher for any power output than the efficiency of the corresponding notch settings (blue line in figure 8). However, the model neglects engine dynamics that actually have an effect in the real engine, so this operation may not be suitable for a real world application since switching from full load to idle would neither be practical nor efficient.

The above DP calculation took 2277 seconds

which may be considered a large computational requirement for such a short trip. Fortunately, this can be improved removing states that are consequence of integral constraints. To do so, the integral constraint (elapsed time in this particular problem) is adjoined to the cost index by means of a weighting parameter ψ :

$$\mathcal{L}_\psi = \int_0^S \left(\frac{F}{v} + \psi \frac{1}{v} \right) ds \quad (24)$$

Formally, as stated in [39], an integral constraint on the form of $\widehat{N} = \int_0^T N(x, u, t) dt$ can be interpreted as a state x_{n+1} such that $\dot{x}_{n+1} = N(x, u, t)$. According to Euler-Lagrange equations, a state x_{n+1} may be adjoined to the cost index by means of a weighting function ψ such that $\dot{\psi} = -\partial H / \partial x_{n+1}$, with $H = L + \psi x_{n+1}$. In the case of the integral constraint, L is not a function of x_{n+1} and consequently $\dot{\psi} = 0$. Therefore, an integral constraint can be introduced as an additional term in the cost index weighted with a constant parameter ψ as done in (24).

The above artifact might be interpreted as a penalization on the cost index for solutions taking large amounts of time to reach the end of the track. It allows to reduce the current number of states in one unit. However, the value for ψ such that $\int_0^S 1/v ds = T$ must be found. This is generally done by iterating, which unfortunately spoils some of the computational time savings. The results for this second DP approach at the benchmarking route are shown in red at figure 9, while blue trajectories correspond to the DP approach with 2 states (speed and remaining time). The results are pretty similar to those of DM, showing the acceleration/cruise/coasting/braking sequence. The main difference is the way DP handles the cruising section, with a high frequency switch between notch zero and 7 as discussed above.

5.2 Indirect methods

Indirect methods (IM) are based on necessary conditions for optimality and, in few words, first they try to solve the differential equation and afterwards to discretize the solution. The most representative algorithm of this family is the PMP [39, 48]. This is by far, the most common method in railroad journey optimization in literature [12–

Variable	Type	Range	Elements
Distance	domain	0 ~ 49.8 km	499
Train speed	state	0 ~ 140 km/h	141
Elapsed time	state	0 ~ 1633 s	328
Notch setting	actuator	0 ~ 8	9
Braking effort	actuator	0 ~ 380 kN	15

Table 5: Grid discretization at DP problem definition. Note that elapsed time is actually an integral constraint included in formulation as a state for convenience.

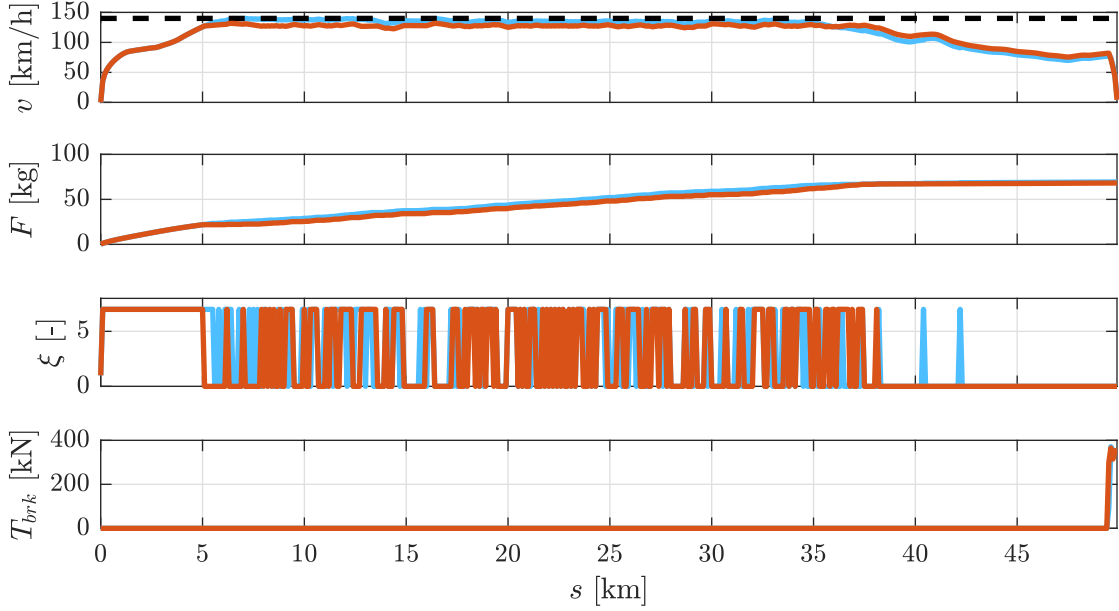


Figure 9: Train OCP solution calculated with DP (problem with 1 state in red, with 2 states in green). Main quantities are displayed: speed (state), accumulated fuel consumption (cost index), notch and braking force (actuators). Dashed line is the track speed limit.

16]. This method arises from the idea of adjoining the states to the cost function (same philosophy than adjoining integral constraint in DP), constructing the so-called Hamiltonian:

$$H = L_s(x, u, s) + \sum_n \lambda_i(s) \frac{\partial x_i(s)}{\partial s} \quad (25)$$

where the Lagrangian multipliers λ_i are known as *costates*. It can be verified that the optimal trajectory minimizes the value of the Hamiltonian at any position:

$$H(\lambda^*, x^*, u^*, s) \leq H(\lambda^*, x^*, u, s) \quad (26)$$

The trajectory of the *costates* follows the dynamics:

$$\frac{\partial \lambda_i}{\partial s} = -\frac{\partial H}{\partial x_i} \quad (27)$$

Therefore, the original OCP is translated into an equivalent problem with $2N_x$ ODEs. Generally this is solved by shooting techniques, which can be summarized as: guess initial values for λ_i , find actuations that minimize Hamiltonian, update states by integrating $\partial x / \partial s = f(x, u, s)$ and update *costates* also by integrating (27); if final conditions are met the problem is solved; if not, a different set of initial λ_i should be used.

IM shows several advantages:

- Can manage large scale problems that DP cannot in a compact form.
- Suitable for online optimal control since they do not need to evaluate the whole problem at once but optimize step by step separately (if λ_i are known).

Unfortunately, only very simple problems can be solved with IM [49] (generally those where all λ_i are constant or where an explicit solution for all ODEs exist). PMP drawbacks are:

- *Costate* ODEs are usually ill-conditioned, resulting in divergent trajectories when integrated numerically, even with a good initial guess [50].
- Initial λ_i values are generally hard to find and computationally demanding.
- Discrete quantities cannot be managed since functions must be continuous and differentiable.
- Active state constraints introduce discontinuities in *costates* that cannot be managed with this kind of optimization methods.
- Only necessary conditions for a local optimum are provided so optimality of solution is only guaranteed for convex problems.
- If the cost index is a linear function of u , in the differential $\partial H/\partial u = 0$ (calculated to find the minimum of Hamiltonian) the control u vanishes, resulting in a bang-bang control or a singular-control problem which is generally hard to solve [39, 51].

Due to the above drawbacks, the common use of PMP in rail transportation literature is not a direct application of the original principle but a set of algorithms or *modes* that drains from the theory, known as *necessary conditions for optimality*. This consists in analyzing the analytical solution of the control variable for different ranges of λ_i . This analysis shows (after several model assumptions) that an optimal trajectory typically consists of a set of full power/hold speed/coast segments, where only the switching times are to be found [15, 16, 19]. It is useful in practical applications as a first approach to optimal operation with low computational requirements and, in fact, algorithms proposed by [11–16] are already used in commercial devices. Unfortunately, this analytical analysis is only possible for very simple models where optimal control

is of bang-bang type (as stated before, this is the case when $\partial H/\partial u = 0$). For a more complex model, such as the one at hand in this paper, the necessary conditions for optimality-based algorithms are suboptimal and, therefore, the full PMP theory must be considered.

The complete PMP method has been formulated and implemented to solve the train OCP. One state (speed) and two actuators (notch setting and braking effort) are considered to minimize the index at (24). The Hamiltonian is:

$$H = \frac{F}{v} + \psi \frac{1}{v} + \lambda_v \frac{\dot{v}}{v} \quad (28)$$

with λ_v the *costate* associated to the speed.

This OCP shares space domain discretization with DP to produce fair comparisons (check domain discretization in table 5). Note that notch setting actuator is relaxed to a continuous function, albeit it can only take discrete values, because PMP formulation requires all functions to be continuous and differentiable.

It is well known that *costate* ODEs are generally ill-conditioned. That brings numerical issues when integrating (27), resulting in divergent $\lambda_i(s)$ trajectories. An example of such a divergent trajectory may be appreciated at figure 10 for $\lambda_v(s)$ (in red) as calculated with PMP. As a workaround to this numerical issue, if an optimal solution is already available, *costates* can be pre-calculated from optimal state and actuator trajectories as [41] introduced:

$$\lambda_i(s) = -\frac{\partial L_s^*}{\partial(\partial x_i/\partial s)} \quad (29)$$

The *costate* $\lambda_v(s)$ has been calculated for the train OCP with the above expression using L_s^* and x_i trajectories from DP solution. Results may be found in figure 10 (in blue). Despite $\lambda_v(s)$ is noisy in this case (caused by the disturbances introduced to the problem, i.e. track slope and radius profile) it is easy to see that it shows a convergent behavior. The comparison between this result and the divergent evolution of the numerically integrated $\lambda_v(s)$ indicates that the proper $\lambda_v(s)$ trajectory cannot be guessed by means of numerical integration but from an existing solution to the problem. This makes PMP virtually lacking of utility for this particular OCP.

Since PMP method lacks of a minimum robustness to obtain a solution to the train OCP, the

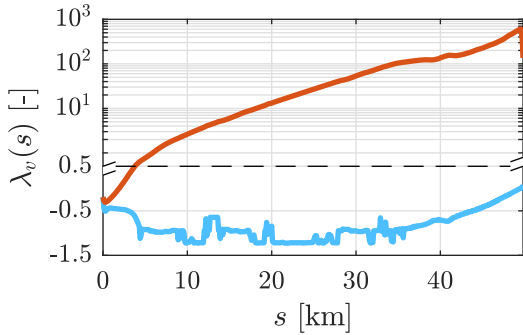


Figure 10: *Costate* trajectory of PMP method, calculated from results of DP method using (29) (in green) and by numerical integration with its formal definition from (27) (in red).

$\lambda_v(s)$ trajectory calculated with (29) has been used to show that, at least, the method is still able to reach the optimal solution when getting rid of some sources of numerical issues. Figure 11 shows the speed profile, fuel consumption, notch setting and braking effort of the resulting PMP solution to the train OCP. Even so, it may be appreciated that around $s = 23$ km train slightly over-speeds. Speed limit constraint is not respected as long as state constraints cannot be directly considered in PMP formulation but through the value of the *costate*: since λ_v is calculated by numerical differentiation from DP, discretization issues may lead to a suboptimal value that violate state constraints. Notch setting behavior is erratic compared to DP although speed profile holds many similarities.

To provide a workaround to the inability of guessing proper values for $\lambda_v(s)$ without the need of a previous solution to the OCP, a constant *costate* value has been used instead. A sweep was performed, varying values in the range of $[\bar{\lambda}_v - 1, \bar{\lambda}_v + 1]$ with $\bar{\lambda}_v$ the average of the actual $\lambda_v(s)$ trajectory. The resulting speed profiles can be appreciated in figure 11 in brown scale (darker colors are lower values). Despite the acceleration/cruising/coasting/braking pattern might be somehow identified in the notch trajectory, the resulting speed profile is unrecognizable. In fact, none of these fulfill constraints, especially the terminal speed, since a constant λ_v value means a roughly constant actuation as long as the Hamiltonian reaches a steady state situation, and there will be no chances to brake.

It may seem strange that after so many works that successfully employed PMP to extract an optimal control policy for diesel locomotives [11–14], it is still unable to solve the current OCP. The answer is that these works extract operation modes (full throttle, coasting, full brake, etc.) as a function of the *costate* with no need to solve the actual PMP set of ODEs. Then, they combine those modes to build a complete control trajectory which is a simplification of the optimal trajectory. To calculate the actual solution to the OCP the complete set of ODEs must be solved, and at this point PMP method falters.

6 Discussion on benchmarking results

Two methods have been capable of solving the current OCP in an standalone form: DP and the presented DM. The main performance characteristics of both algorithms are summarized in table 6 as well as constraint fulfilling in table 7. On the other hand, PMP method is unable to find a solution because of the ill-conditioning of the ODEs that drive the *costates* to diverge. However, if a previous solution is available, *costates* can be guessed and PMP is able to deliver a proper solution. Even so, PMP throws the worst fuel result (73 kg versus 68.2 of the DP and 70.4 of the DM) with computational requirements somewhere between DP and DM, while slightly violating speed limits (state constraints such as a speed limits cannot be directly considered in formulation but through *costates*).

DP, with its current discretization, solved the OCP after 38 minutes of calculation. Although this solution is expected to be the global optimum of the problem, it must be taken into account that actually it is the optimum among the discrete spaces of candidates \mathcal{X} and \mathcal{U} into which the continuous problem has been gridded. As long as the system does not go only through the grid but also through the continuous space, OCP solution may be suboptimal or even violate some constraints due to interpolation of gridded values along the continuous path of states and actuators. In fact, it may be appreciated that DP solution in figure 9 ends at 10 km/h while the final constraint is set to 0 km/h. This issue might be reduced as the grid density increases, but at

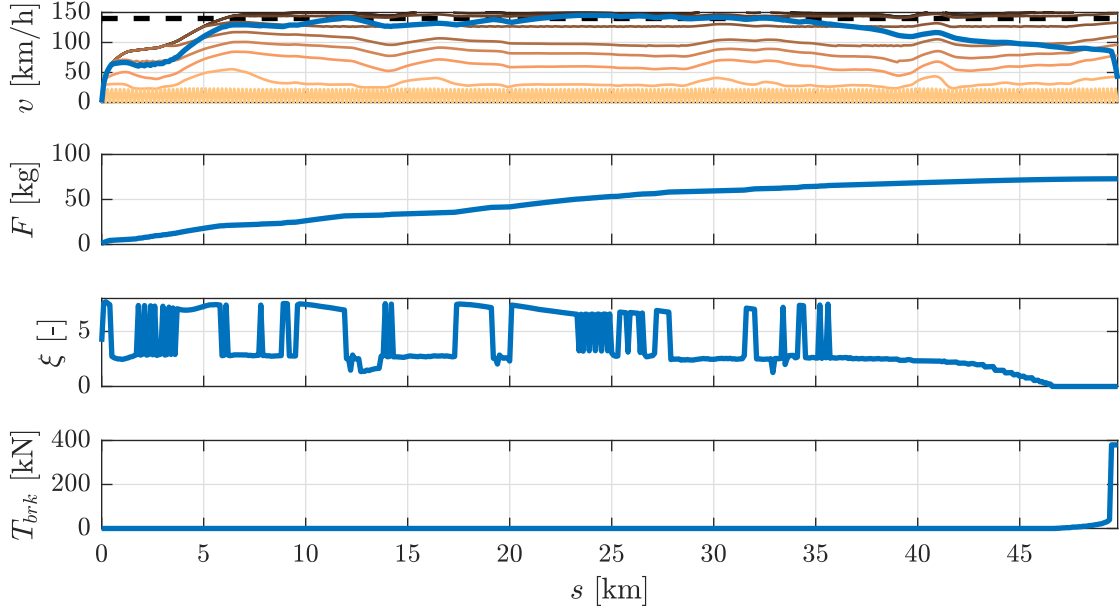


Figure 11: Train OCP solution calculated with PMP.

Method	Fuel consumption [kg]	CPU time [s]	Total memory [kB]
DP (2 states)	69.4	2277	224000
DP (1 state)	68.2	44	680
PMP (precalculated λ_v)	73.0	12	40
DM (relaxed)	69.5	1.5	60
DM (discrete)	70.4	0.5	45

Table 6: Main performance indices of DP, PMP and DM algorithms used to solve the train OCP at the benchmarking route. Note that CPU time includes possible iterations in DP and PMP to find values for the parameter ψ .

Method	Max speed [km/h] (limit: 140)	Trip time [s] (limit: 1633)	Final speed [km/h] (limit: 0)
DP (2 states)	139	1618	10
DP (1 state)	132	1632	4
PMP (precalculated λ_v)	145	1605	0
DM (relaxed)	131	1631	0
DM (discrete)	132	1623	0

Table 7: Constraint fulfilling of DP, PMP and DM of the OCP solution at the benchmarking route. Values violating constraints are in bold numbers and red color.

the expense of strongly increasing computation time.

The general DP method shows a robust optimization of the OCP but calculation takes a large amount of time. The modified DP with one adjoined state at the cost index reached a pretty similar solution but reducing the calculation time from 38 minutes to just 3.5 seconds. However, no explicit method exists to estimate the value of the additional parameter ψ so an iterative calculation is needed until final constraints are met, shooting several DP instances that could jeopardize any time saving. In this case 12 iterations were needed with a total time of 44 seconds.

DM solved the OCP after 2 seconds (1.5 seconds for the relaxed problem plus 0.5 seconds for the problem with integer trajectories). This calculation time strongly depends on the quality of the initial solution provided to the NLP solver. In this case the initial solution is set to constant speed and notch settings that does not necessarily fulfill the constraints. In addition, for the sake of simplicity, this calculation does not include second derivatives, making it harder for the NLP solver to converge to the problem solution. Even so, the NLP converges quickly to the optimal solution and is in fact much faster than DP or PMP. Moreover, even a quicker calculation can be reached including some heuristics to guess a better initial solution or providing second derivatives. It is remarkable that there is a difference of 0.9 kg of fuel between the relaxed and the discrete DM. This difference is not only because of the integer approximation carried out but also due to the unrealistic fractional notch settings selected in the relaxed DM. Therefore, the penalty in fuel consumption introduced by the integer constraint might be under 1.3%.

DP and DM solutions are pretty much the same in terms of speed profile and fuel consumption as it can be appreciated in table 7 (DP spent 68.2 kg of fuel and DM 70.4 kg). However, there are two remarkable differences between both: the number of switches in actuators and constraints fulfilling.

Regarding the first difference, the fact that DP handles all state and actuation spaces (\mathcal{X} and \mathcal{U}) as discrete variables brings on solutions with switching actuators, especially if several economic poles exist and no switching costs are specified (e.g. efficiency versus notch setting as

shown in figure 8), but also if grid density is poor. On the other hand, DM treats quantities as continuous variables and therefore optimal solution might be smoother. Even after rounding variables with integer constraints, the number of switches are definitely lower than that in DP. This behavior brings a major drawback for DP since a switching solution that is numerically the optimum, may be unpractical for real world conditions. In this case, the operator obviously cannot be constantly switching from notch 7 to idle, not only because the operator itself but also because of the capabilities, wear, safety and efficiency of the prime mover. In this sense, it is also interesting to remark that optimal solutions could be considered uncomfortable from a passenger point of view: as shown in figures 7 and 9, there are many notch switches (way less in the DM solution but still a great number of switches) that may produce a bouncing trip, and heavy braking for a full stop—it saves fuel not to brake until the very end and, then, brake as strong as possible, since it maximizes the average speed almost for free. These issues, which could limit the applicability of these optimal trajectories in the actual locomotive, can be overcome introducing additional constraints that take into consideration ride comfort concerns, such as a limit on the number of switches or an acceptance threshold for negative accelerations. Although the implementation of such constraints is pretty straightforward following the presented methodology, it falls out of the scope of this paper.

Regarding the second difference, as long as DM simulates the system at the same time it is being optimized, it can be guaranteed that the solution fulfills all constraints in a forwards simulation within very tight tolerances. By contrast, due to the discrete set of candidates that DP manages and since OCP is optimized first in a backwards fashion and then simulated, it may happen that the optimal trajectory calculated by the solver (which is restricted to the set of discrete candidates) does not fit the forwards simulation and, therefore, some constraints might not be fulfilled. Table 7 summarizes constraint fulfilling for all algorithms showing that DM is the only method that does not violate any constraint and, in particular, it may be appreciated that DP does not match zero speed at the end of the route while DM does.

As a general conclusion, DP could be the choice for simple problems with few states and actuators and featuring discrete quantities. OCPs with integral constraints would be on the borderline as long as they might not directly jeopardize computational time but introduce additional parameters to calibrate (which vary from problem to problem, even when changing a single model parameter) that may be troublesome and require the calculation of several DP problems. OCPs with increasing complexity or number of states/actuators will involve prohibitive computational times. DM, on the other hand, can deal with OCPs of any complexity and many states and actuators. Integral constraints are handled directly on the formulation without any additional parameter and no significant impact on calculation time. Although discrete quantities must be handled indirectly, NLP convergence is usually strong and it takes less time than DP to find the optimal solution to the OCP. In addition, since DM deals with continuous variables, the solution is usually smoother matching constraints better than DP.

Therefore, unless the target OCP is a simple function with a couple of states/actuators, no integral constraints, discrete quantities and some discontinuities, DM should be the choice to solve the OCP.

In the following section, the optimization of a full trip of 127 km with stops in 13 stations is compared to the real utilization of the train. DM is used to solve the OCP since it featured the best performance over all methods in the benchmarking route.

7 Optimization of a long-haul route

As introduced in the previous discussion, DM is the choice to deal with the vast majority of OCPs due to its efficiency, simplicity and flexibility to manage large amounts of constraints, states and actuators [33]. Despite DP method shown a similar performance under some conditions, DM is selected because:

- It keeps a similar performance even if additional constraints or states are considered.
- Train routes are generally repeated multiple

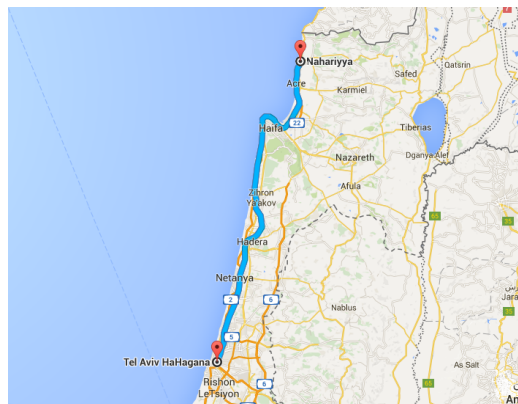


Figure 12: Route between Nahariyya and Tel Aviv HaHagana stations (Israel). Source: Google Maps

times a day with a high degree of predictability, so information of previous trips may be used as initial solution to speed up the algorithm.

- DM is faster and uses reduced memory, making it attractive for online applications with limited memory.

In order to test the performance and applicability of DM to a complete route, the train OCP is implemented for a trip between Nahariyya and Tel Aviv HaHagana stations in Israel whose route may be appreciated in figure 12. The route is daily operated by diesel electric locomotives (same type than the one modeled in this work), and it covers a distance of 127 km in 111 minutes with stops in 13 stations. The slope and radius of the track (disturbances to the OCP) are displayed in figure 13.

This OCP is similar to the problem formulated for the benchmarking route. The minimization objective is the total fuel consumption of the prime mover (9). This is subject to a time limit T_i to reach each station:

$$\int_{s_{i-1}}^{s_i} \frac{1}{v} ds = T_i \quad (30)$$

where s_i is the position of the i -th station ($s_0 = 0$ and $s_{12} = S$). Consequently, total travel time at the actual route and the optimal trajectory will be the same. Speed is also constrained at stations (train must remain stopped at platforms):

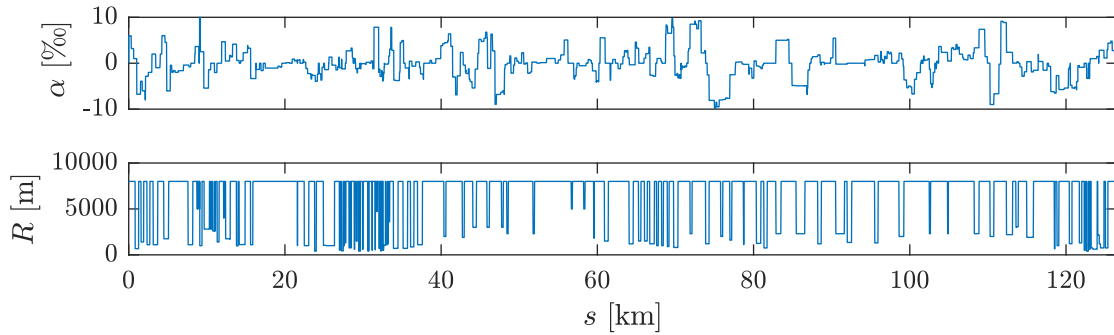


Figure 13: Slope α and track radius R profile at the full route between Nahariyya and Tel Aviv HaHagana stations (Israel).

$$v(s_i) = 0 \quad (31)$$

Additionally, track speed limit (12) and technical limits of the locomotive (13) are also included.

Direct collocation is used to transcribe the OCP to a large sparse NLP, which is optimized with IPOPT. Calculations for the whole route take about 9 seconds for the relaxed OCP and 4 seconds for the discrete OCP. Total fuel consumption at this route is 182 kg which compared to the actual speed profile (228 kg) results in a saving of 46 kg. Figure 14 shows the resulting trajectories and table 8 summarizes the main results. Note that fuel consumption at the actual route is calculated by simulating the model stated in section 2.1 while using measurements of the notch settings selected by the operator onboard the real train, in order to provide a fair benchmark between optimal and real notch trajectories. Therefore, all differences shown in this benchmark are only because of the optimized controls. The actual data from the real operation of the train at this particular route and its simulated fuel consumption are also displayed in figure 14.

There are several facts that can be appreciated when comparing these results with the real operation of the train:

- DM solution tends to strongly accelerate the train at the beginning, then coast and brake only at the very end, while actual speed profile includes few coasting phases.
- Both trajectories brake in a similar way.
- DM never reaches notch 8 (highest power output) but notch 7 (highest efficiency) while the driver regularly does.

- And most importantly, DM operation provided a fuel saving of 46 kg, which is a 20% of total fuel consumption.

8 Discussion on the results and applicability of the optimization method

The benchmark showed that PMP can easily handle problems with many states and actuators, however due to its lack of robustness (ODEs are ill-conditioned), it has been unable to manage the problem properly. Results exhibited strong divergence and it is only possible to find a feasible solution using information from another OCP solver. On the other hand, DP is a robust method that provides the global optimum of the problem (informally speaking, it provides a *soft* global optimum since it searches among a gridded space of candidates instead of the whole continuous space). As a counterpart it suffers the so called *curse of dimensionality*, requiring prohibitive computation times for problems with many states/actuators. The current OCP is close to be unaffordable, however adjoining the integral constraint to the cost function reduced the computational burden by a factor of 50. Results, however, slightly violate final constraints due to the difference between the discrete backwards optimization and the continuous forwards simulation.

DM is able to transcribe a large scale problem with many states, actuators and constraints into a sparse NLP that can be optimized efficiently

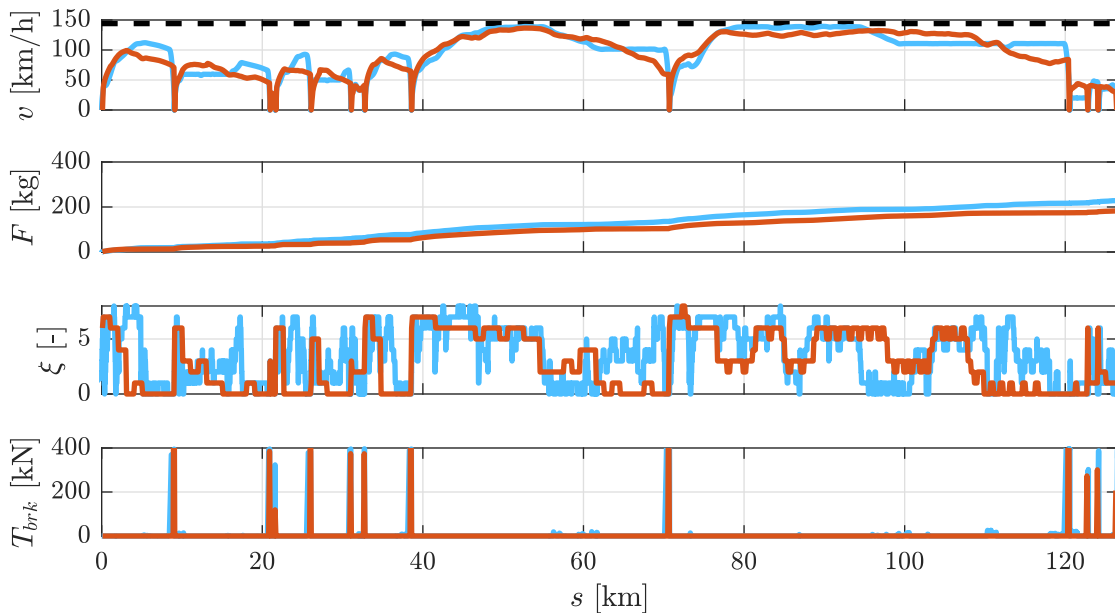


Figure 14: Main results of train operation in the full route between Nahariyya and Tel Aviv HaHa-gana stations in Israel (127 km). Green color corresponds to real train operator actuations and red to direct method OCP solution.

Method	Fuel consumption [kg]	CPU time [s]	Total memory [kB]	Max speed [km/h]	Trip time [min]
Human operator	228	—	—	139	111
Direct method	182	13	150	137	111

Table 8: Main results from the simulation of the long-haul route (13 stations), for the optimized control (DM) and the human operator.

with any state-of-the-art NLP solver. It shows robust convergence to the optimal solution, but is only able to reach a local optimum. Also, handling discrete quantities is troublesome since the NLP turns into a MINLP which is harder to solve. The workaround suggested in this work is to solve the NLP twice: first with the integer variables relaxed into continuous quantities, then round the result to the nearest integer and finally optimized the NLP again with the rounded variables as fixed disturbances. This involves a small penalization in terms of computational time compared to a fully continuous OCP without a significant impact in the cost of the minimization objective.

Recalling the main contributions of the paper as stated in the introduction, the presented method has been able to:

1. Compute the optimal trajectory for a long-haul route, delivering the optimal set of notch settings to drive the locomotive efficiently in a real route. Particularly, a significant reduction of fuel consumption has been achieved compared to the control performed by the train operator. Despite the model used in this work has been kept simple in line with the existing literature to provide a fair comparison of other optimization methods, more complex models (with additional states such as engine dynamics, coolant or motor coil temperatures, etc.) can be easily included (as long as they are continuous functions) in the presented formulation as in (5). Contrarily to DP, the inclusion of additional states only involves a linear increase in the Jacobian dimensions.

2. Include constraints to the problem, namely full stops at platforms (boundary constraints) track speed limits (path constraint), tractive effort limit (path constraint) and maximum travel time (integral constraint). These constraints were included in a straightforward manner in the formulation, simply by discretizing them following the methodology described in section 4. Any other boundary, path or integral constraint can be included following the same precise methodology. Note that, in the Jacobian matrix, boundary constraints only introduce a new element, integral constraints a new row, and path constraints a whole new matrix.
3. Find the optimal trajectory for the long-haul route with very little computational requisites. In fact, this problem just required 13 seconds to be optimized, suggesting that much more complex models and many additional constraints could still be considered while keeping a low computational burden. This also points out that, in case of any unexpected situation during the train operation (such as a red signal), the method could recalculate the remaining portion of the route in few seconds with updated arrival time constraints.

The above results showed promising performance and capabilities of direct methods for railroad control applications. However, it must be taken into account that for a final implementation many additional requisites must be fulfilled that, so far, are not addressed in this work. Among others, these are: a more detailed train set model considering additional dynamics, backup controls in case of unexpected situations that have not been included in the formulation or ride comfort considerations.

9 Conclusions

The optimal operation of a diesel electric locomotive has been approached as an OCP to minimize fuel consumption fulfilling time and distance constraints for a given route. A benchmarking route has been used to analyze and benchmark the three main families of methods able to deal

with OCPs: HJB equation (DP), IM (PMP) and DM. This particular problem consists of one state (speed), two actuators (notch setting and braking force), one integral constraint (elapsed time) and two disturbances (track slope and radius).

DM resulted the most effective and versatile algorithm, so it has been benchmarked with a complete route that is daily covered by diesel electric locomotives in Israel, between Nahariyya and Tel Aviv HaHagana stations. This OCP is interesting because it is a large problem, involves additional constraints (arriving on time at each station), and data of the real operation of the train is available at that route. DM spent 9 seconds to solve the relaxed OCP and 4 additional seconds to find a solution for the discrete OCP, resulting in a saving of 46 kg of fuel compared to the real operation of the train.

Optimal control is especially interesting in diesel electric locomotives because performance is strongly dependent on the way it is actuated and many information is known in advance (speed limits, track grade, etc.). For the case at hand (a typical route between two cities) it has been found that a proper optimal control saves about 20% of the fuel consumption compared to the actual current operation and that DM is probably the best performer algorithm for this kind of problems.

However, several things are still missing to provide a complete optimal control that can be used for real world applications. Future works may focus on closing this gap. Some interesting ideas might be applying the methodology presented in this paper with detailed locomotive models that would provide more accurate optimization trajectories, or including ride comfort constraints to the problem in order to guarantee a safe and realistic train operation.

References

- [1] K. Ichikawa. “Application of optimization theory for bounded state variable problems to the operation of train”. In: *Bulletin of JSME* 11.47 (1968), pp. 857–865 (cit. on p. 1).
- [2] I. A. Asnis, A. V. Dmitruk, and N. P. Osmolovskii. “Solution of the problem of the energetically optimal control of the mo-

- tion of a train by the maximum principle”. In: *USSR Computational Mathematics and Mathematical Physics* 25.6 (1985), pp. 37–44 (cit. on p. 1).
- [3] X. Feng et al. “A review study on traction energy saving of rail transport”. In: *Discrete Dynamics in Nature and Society* 2013 (2013), p. 9 (cit. on p. 2).
- [4] A. Higgins, E. Kozan, and L. Ferreira. “Optimal scheduling of trains on a single line track”. In: *Transportation research part B: Methodological* 30.2 (1996), pp. 147–161 (cit. on p. 2).
- [5] X. Li and H. K. Lo. “An energy-efficient scheduling and speed control approach for metro rail operations”. In: *Transportation Research Part B: Methodological* 64 (2014), pp. 73–89 (cit. on p. 2).
- [6] X. Li and H. K. Lo. “Energy minimization in dynamic train scheduling and control for metro rail operations”. In: *Transportation Research Part B: Methodological* 70 (2014), pp. 269–284 (cit. on p. 2).
- [7] M. Samá et al. “Ant colony optimization for the real-time train routing selection problem”. In: *Transportation Research Part B: Methodological* 85 (2016), pp. 89–108 (cit. on p. 2).
- [8] H.-S. Hwang. “Control strategy for optimal compromise between trip time and energy consumption in a high-speed railway”. In: *IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans* 28.6 (1998), pp. 791–802 (cit. on p. 2).
- [9] X. Li et al. “Train energy-efficient operation with stochastic resistance coefficients”. In: *International Journal of Innovative Computing, Information and Control* 9.8 (2013), pp. 3471–3483 (cit. on p. 2).
- [10] K. Gao et al. “An online distributed game optimal control for heavy haul trains with limited communication”. In: *Mathematical Problems in Engineering* 2015 (2015), p. 9 (cit. on p. 2).
- [11] P. Howlett. “The optimal control of a train”. In: *Annals of Operations Research* 98.1 (2000), pp. 65–87 (cit. on pp. 2, 12, 13).
- [12] E. Khmelnitsky. “On an optimal control problem of train operation”. In: *IEEE Transactions on Automatic Control* 45.7 (2000), pp. 1257–1266 (cit. on pp. 2, 10, 12, 13).
- [13] R. R. Liu and I. M. Golovitcher. “Energy-efficient operation of rail vehicles”. In: *Transportation Research Part A: Policy and Practice* 37.10 (2003), pp. 917–932 (cit. on pp. 2, 10, 12, 13).
- [14] P. G. Howlett, P. J. Pudney, and X. Vu. “Local energy minimization in optimal train control”. In: *Automatica* 45.11 (2009), pp. 2692–2698 (cit. on pp. 2, 10, 12, 13).
- [15] A. Albrecht et al. “The key principles of optimal train control – Part 1: Formulation of the model, strategies of optimal type, evolutionary lines, location of optimal switching points”. In: *Transportation Research Part B: Methodological* 94 (2015), pp. 482–508 (cit. on pp. 2, 10, 12).
- [16] A. Albrecht et al. “The key principles of optimal train control – Part 2: Existence of an optimal strategy, the local energy minimization principle, uniqueness, computational techniques”. In: *Transportation Research Part B: Methodological* 94 (2015), pp. 509–538 (cit. on pp. 2, 10, 12).
- [17] M. Miyatake and H. Ko. “Optimization of train speed profile for minimum energy consumption”. In: *IEEE Transactions on Electrical and Electronic Engineering* 5.3 (2010), pp. 263–269 (cit. on pp. 2, 8).
- [18] S. Lu, S. Hillmanssen, and C. Roberts. “A power-management strategy for multiple-unit railroad vehicles”. In: *IEEE Transactions on Vehicular Technology* 60.2 (2011), pp. 406–420 (cit. on pp. 2, 8).
- [19] J. Cheng and P. Howlett. “A note on the calculation of optimal strategies for the minimization of fuel consumption in the control of trains”. In: *IEEE Transactions on Automatic Control* 38.11 (1993), pp. 1730–1734 (cit. on pp. 2, 12).

- [20] P. Pudney and P. Howlett. “Optimal driving strategies for a train journey with speed limits”. In: *The Journal of the Australian Mathematical Society. Series B. Applied Mathematics* 36.1 (1994), pp. 38–49 (cit. on p. 2).
- [21] P. Howlett. “Optimal strategies for the control of a train”. In: *Automatica* 32.4 (1996), pp. 519–532 (cit. on p. 2).
- [22] P. G. Howlett and J. Cheng. “Optimal driving strategies for a train on a track with continuously varying gradient”. In: *The Journal of the Australian Mathematical Society. Series B. Applied Mathematics* 38.03 (1997), pp. 388–410 (cit. on p. 2).
- [23] Y. Wang et al. “Optimal trajectory planning for trains—A pseudospectral method and a mixed integer linear programming approach”. In: *Transportation Research Part C: Emerging Technologies* 29 (2013), pp. 97–114 (cit. on p. 2).
- [24] Y. Wang et al. “Optimal trajectory planning for trains under fixed and moving signaling systems using mixed integer linear programming”. In: *Control Engineering Practice* 22 (2014), pp. 44–56 (cit. on p. 2).
- [25] P. Wang and R. M. P. Goverde. “Multiple-phase train trajectory optimization with signalling and operational constraints”. In: *Transportation Research Part C: Emerging Technologies* 69 (2016), pp. 255–275 (cit. on p. 2).
- [26] W. J. Davis. *The tractive resistance of electric locomotives and cars*. General Electric, 1926 (cit. on p. 3).
- [27] S. A. Bernsteen, R. A. Uher, and J. P. Romualdi. “The interpretation of train rolling resistance from fundamental mechanics”. In: *IEEE Transactions on Industry Applications* 1.5 (1983), pp. 802–817 (cit. on p. 3).
- [28] L. T. Biegler. “Solution of dynamic optimization problems by successive quadratic programming and orthogonal collocation”. In: *Computers and Chemical Engineering* 8.3-4 (1983), pp. 243–247 (cit. on p. 6).
- [29] D. Tieu, W. R. Cluett, and A. Penlidis. “A comparison of collocation methods for solving dynamic optimization problems”. In: *Computers and Chemical Engineering* 19.4 (1995), pp. 375–381 (cit. on p. 6).
- [30] D. A. Benson et al. “Direct Trajectory Optimization and Costate Estimation via an Orthogonal Collocation Method”. In: *Journal of Guidance, Control, and Dynamics* 29.6 (2006), pp. 1435–1440 (cit. on p. 6).
- [31] D. Garg et al. “A unified framework for the numerical solution of optimal control problems using pseudospectral methods”. In: *Automatica* 46.11 (2010), pp. 1843–1851 (cit. on p. 6).
- [32] J. T. Betts. *Practical methods for optimal control and estimation using nonlinear programming*. SIAM, 2010 (cit. on pp. 6, 8).
- [33] J. Asprion, O. Chinellato, and L. Guzzella. “Optimal control of diesel engines: Numerical methods, applications, and experimental validation”. In: *Mathematical Problems in Engineering* 1 (2014), pp. 1–21 (cit. on pp. 6, 16).
- [34] C. L. Darby and A. V. Rao. “Minimum-Fuel Low-Earth Orbit Aeroassisted Orbital Transfer of Small Spacecraft”. In: *Journal of Spacecraft and Rockets* 48.4 (2011), pp. 618–628 (cit. on p. 6).
- [35] A. Wächter and L. T. Biegler. “On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming”. In: *Mathematical Programming* 106.1 (2006), pp. 25–57 (cit. on p. 6).
- [36] P. Bonami et al. “An algorithmic framework for convex mixed integer nonlinear programs”. In: *Discrete Optimization* 5.2 (2008), pp. 186–204 (cit. on p. 7).
- [37] S. Le Digabel. “Algorithm 909: NOMAD: Nonlinear Optimization with the MADS algorithm”. In: *ACM Transactions on Mathematical Software* 37.4 (2011), pp. 1–15 (cit. on p. 7).
- [38] J. Van Leeuwen. *Handbook of theoretical computer science: algorithms and complexity*. Vol. A. Elsevier, 1990 (cit. on p. 7).

- [39] A. E. Bryson and Y.-C. Ho. *Applied optimal control: Optimization, estimation, and control*. Taylor & Francis, 1975 (cit. on pp. 8, 10, 12).
- [40] E. Hellström et al. “Look-ahead control for heavy trucks to minimize trip time and fuel consumption”. In: *Control Engineering Practice* 17.2 (2009), pp. 245–254 (cit. on p. 8).
- [41] C. Guardiola et al. “A new approach to optimally tune the control strategy for hybrid vehicles applications”. In: *Engine and Powertrain Control, Simulation and Modeling*. Vol. 3. 1. 2012, pp. 255–261 (cit. on pp. 8, 12).
- [42] T. Nuesch et al. “Optimal energy management and sizing for hybrid electric vehicles considering transient emissions”. In: *Engine and Powertrain Control, Simulation and Modeling*. Vol. 3. 1. 2012, pp. 278–285 (cit. on p. 8).
- [43] W. Dib et al. “Evaluation of the energy efficiency of a fleet of electric vehicle for eco-driving application”. In: *Oil & Gas Science and Technology – Revue d’IFP Energies nouvelles* 67.4 (2012), pp. 589–599 (cit. on p. 8).
- [44] J. M. Luján et al. “Cost of ownership-efficient hybrid electric vehicle powertrain sizing for multi-scenario driving cycles”. In: *Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering* 230.3 (2016), pp. 382–394 (cit. on p. 8).
- [45] R. Vinter. *Optimal control*. Springer, 2010 (cit. on p. 9).
- [46] O. Sundström and L. Guzzella. “A Generic Dynamic Programming Matlab Function”. In: *IEEE Control Applications, Intelligent Control*. 2009, pp. 1625–1630 (cit. on p. 9).
- [47] A. Sciarretta, G. De Nunzio, and L. L. Ojeda. “Optimal Ecodriving Control: Energy-Efficient Driving of Road Vehicles as an Optimal Control Problem”. In: *IEEE Control Systems* 35.5 (2015), pp. 71–90 (cit. on p. 10).
- [48] L. S. Pontryagin et al. *The mathematical theory of optimal processes*. Interscience, New York, 1962 (cit. on p. 10).
- [49] C. Guardiola et al. “Insight into the HEV/PHEV optimal control solution based on a new tuning method”. In: *Control Engineering Practice* 29 (2014), pp. 247–256 (cit. on p. 12).
- [50] H. Waschl et al. *Optimization and Optimal Control in Automotive Systems*. Springer, 2014 (cit. on p. 12).
- [51] D. Cobb. “Controllability, Observability, and Duality in Singular Systems”. In: *IEEE Transactions on Automatic Control* 29.12 (1984), pp. 1076–1082 (cit. on p. 12).