

ENHANCED EFFICIENCY IN RAILWAY DRIVING: DEVELOPMENT OF ECOS (ENERGY CONSUMPTION OPTIMIZATION SOFTWARE) TOOL

FELIX ARIZA, JOSE M. MERA, EDUARDO CASTELLOTE & ISRAEL GÓMEZ-REY
CITEF (Railway Technologies Research Centre), Spain

ABSTRACT

Railway transport consumes large amount of energy and fuel every year. Increasing environmental concern and raising prices in diesel fuel and electrical energy has resulted in optimization techniques being a must for railway systems. This paper aims to explain the development of a software tool able of calculating the optimal consumption speed profile for a train service within a given topology and timetable. It also shall provide when and what actions (powering, braking, holding speed or coasting) the driver should follow for achieving it. Historically these objectives have been approached using computer algorithms based on very simplified models or dynamic programming, resulting in unrealistic or restrictively slow frameworks. We propose a really fast and accurate methodology based on Optimal Control theory. The problem is formulated as a differential system of equations stating train dynamics constraint by track features, as permitted speed or slope value in the various sections. Analytical solution for such a system of equations is not suitable due to its strictly non-linearity, being necessary the application of fast convergence iterative methods as the Interior Point method based on logarithm barrier penalty functions. Several real-life scenarios have been tested using the explained tool and its output have been compared with validated results, having achieved a high degree of accuracy and speed. Accordingly is valid to conclude that this approach can be used to develop a Driver Advisory System for real trains.

Keywords: railway, efficiency, optimization, consumption, ECOs (Energy Consumption Optimization software), DAS (Driver Advisory System), train dynamics, optimal control, interior point method, environmental concern.

1 INTRODUCTION

Railway transport operation involves expenses that can be divided between the depreciation of the rolling stock and its operational cost, including staff salaries, maintenance and energy/fuel consumed. All these expenses are mandatory along operational life and hardly diminishable except for the consumption that is likely to be reduced using an efficient driving strategy.

Railway operators are recently investing in efficient driving systems that allow them to reduce this factor by using DAS (Driver Advisory Systems) devices, usually a computer based system with a screen monitor where an optimal speed profile is shown to the driver in order to keep train velocity as close as possible to calculated optimal speed at any moment while satisfying a time schedule.

Calculation of these optimal speed profiles is a complex task due to the strictly non-linearity nature of train dynamics differential equations, being necessary the application of fast convergence iterative methods for solving it.

One of the most mature and fastest to converge among them is the Interior Point method based on logarithm barrier penalty functions. This method can be found in Ipopt package which implements an interior point line search filter method algorithm able to find feasible solutions to nonlinear problems.

Prior to apply any iterative solver the continues-time problem must be transformed into a nonlinear optimization problem using an appropriate discretization method.



An Orthogonal Collocation discretization has been chosen to divide the system in a number of finite elements containing piecewise low-order polynomial representations of the state and control variables in each element, allowing a simultaneous approach for this optimization problem.

This method evolves from sequential approaches for discretization by considering the dynamic optimization problem as a multi-period system where the periods themselves are represented by finite elements in time and can be considered as a special case of implicit Runge–Kutta methods, where their concepts and properties apply directly.

Using this formulation has a number of pros and cons. The problem size increases in direct proportion to the number of finite elements and to the polynomial degree. On the other hand, this formulation provides a great deal of sparsity and structure, along with efficient sparse linear solvers available allow to find a solution efficiently.

To evaluate this methodology, the results obtained are checked by two ways: (i) using the already validated third party CITEF's tool Hamlet Analysis Suite that has been used in more than one hundred electrical railway projects, and (ii) comparing with real data provided by some major railway operators.

The main contributions of this paper are: (i) the development of a train dynamics model that captures the real behaviour of any train, (ii) the development of a computer framework based on Ipopt that allows fast finding of accurate solutions, and (iii) the validation of the outputs using real life operation data and third-party simulation tools.

The rest of the paper is organized as follows. In Section 2 previous studies are evaluated, in Section 3 a mathematical background for interior point method is briefly described, in Section 4 the train model is developed, in Section 5 the applied methodology is introduced and in Section 6 the results obtained are analysed. Finally, Section 7 draws the conclusions of this tool and presents future research.

2 LITERATURE REVIEW

Energy consumption optimization in vehicles has received growing attention in recent years. Increasing use of railway transportation for both passengers and freight at world scale has focused the attention of railway operators in decreasing energy consumption, in an attempt to reduce operational costs and become more environmentally friendly.

Investing in any optimal driving development accounts due to large scale energy consumption done by them along their operational life, therefore its reduction becomes a profitable investment in the short term.

There are many studies focusing on the energy efficient driving of trains. The earliest research was based on simplified models, based in the assumption of ignoring speed limits or slopes in the lines. Then those models were reformulated to take gradients and speed limits into consideration, but still used a very simplified train dynamic simulation. Also, efforts were applied into finding the coasting position by search algorithms, including genetic ones.

In recent years, various algorithms have been successfully applied to efficient operation of railway trains from energy viewpoint. Rodrigo et al. [1] remark about the value on the use of regeneration energy; a Lagrangian multipliers method was proposed to solve the problem by optimizing the speed profile. Huang et al. [2] propose the use of a multi-population genetic algorithm to optimize the train operation for multiple interstation's, taking into account both the trip time and the driving strategy. Cabrera-Montie et al. [3] go one step further and propose the use of dynamic programming method for solving this problem for a truck in a road, although their conclusions are totally valid for railway, as demonstrated in later studies carried out by CITEF investigation group. Larrañaga et al. [4] aim in the comparison between a direct method and an indirect method used to determine an optimal speed profile. They also

use an improved train model for their tests. Ben-Chaim et al. [5] propose an analytical method of evaluating vehicle fuel consumption under different operating conditions, as cruising at constant speed and acceleration. Their conclusions are valid for large diesel engines used in some railway locos.

All the above papers present some problems that make them unsuitable for developing a general optimizing framework for railway trains. Such a general framework must meet two essential requirements, being very fast in order to be implemented as a DAS on-board device and being very accurate in order to be used as an office analysis tool to make decisions about optimal trip time and expected savings.

In [1] a feasible method is proposed but it finally relies in Matlab optimization package, making it slower and third-party dependant. In [2] the authors make use of simplified assumptions for their calculations. In [3] an interesting approach to dynamic programming is presented. Dynamic programming always guarantees to find an optimal absolute minimum solution, but it is a slow method with a solving speed directly proportional to the complexity and grid size used for solving the problem. In [4] direct and indirect methods are compared. The direct method describes train model in detail, but discretization of the problem is done using a naive Euler method for then relying in third party programming language AMPL. This decision reduce accuracy and make algorithm third party dependant.

Finally, it is noteworthy that most of the previous studies ignores the influence of trip time on the energy consumption, which must be taken into account as an average consumption applied all over the trip time. This energy is used for auxiliary system of the train, as electrical engine refrigeration, air conditioning or lighting.

3 MATHEMATICAL BACKGROUND

Optimization problems can be faced from different solving strategies. In the last years growing interest in optimization methods has led to the development of interior point methods for large-scale nonlinear continuous systems, based in barrier logarithmic penalty functions. These methods has been implemented using efficient algorithms together with widely available multi-core processors, resulting in several mature open and proprietary libraries able to solve the problem while fast converging to a global solution.

One of the most popular state of the art framework is IPOPT, short for "Interior Point OPTimizer", which can be used to solve general nonlinear programming problems of the form:

$$\min_{x \in \mathbb{R}^n} f(x), \quad (1)$$

$$s. t. \quad g^L \leq g(x) \leq g^U, \quad (2)$$

$$x^L \leq x \leq x^U, \quad (3)$$

where $x \in \mathbb{R}^n$ are the optimization variables (possibly with lower and upper bounds, $x^L \in [-\infty, +\infty]^n$ and $x^U \in (-\infty, +\infty]^n$), $f: \mathbb{R}^n \rightarrow \mathbb{R}$ is the objective function, and $g: \mathbb{R}^n \rightarrow \mathbb{R}^m$, with $m \leq n$, are the general nonlinear constraints. The functions $f(x)$ and $g(x)$ can be linear or nonlinear and convex or non-convex (but should be twice continuously differentiable). The constraints, $g(x)$, have lower and upper bounds, $g^L \in [-\infty, +\infty]^n$ and $g^U \in (-\infty, +\infty]^n$.

Ipopt implements a primal-dual interior point method that uses a line-search based on filter method. It has been designed to exploit 1st and 2nd derivative (Jacobians and Hessians).

The main features of Ipopt are that it is open software released under the EPL license (although it may use some proprietary sparse linear solvers), it can be called from various

modelling environments (C++ and Python frameworks available) and if no Hessians are provided, it will be able to approximate them using a quasi-Newton method, specifically a BFGS update.

Initially, interior-point methods were developed using exact penalty merit functions to enforce progress toward the solution. In recent years filter methods has been proposed, offering an alternative to merit functions, as a tool to guarantee global convergence in algorithms for nonlinear programming and to allow convergence from poor starting points. The underlying concept is that trial points are accepted if they improve the objective function or improve the constraint violation instead of a combination of those two measures defined by a merit function.

More recently, this filter approach has been adapted to barrier methods in a number of different ways, as considering a trust region filter method that bases the acceptance of trial steps on the norm of the optimality conditions or using heuristics based on the idea of filter methods. In the other hand, Ipopt was designed on a primal-dual interior-point algorithm with a filter line-search, specifically a primal-dual barrier method to solve nonlinear optimization problems of the stated above form. Ipopt performance compares favourable to other state of the art nonlinear optimization packages as KNITRO and LOQO.

In [6] a more comprehensive explanation about interior point method implemented in Ipopt package and all its mathematical approach can be consulted.

4 MODEL DESCRIPTION

Train cinematics will be subject to the next continues-time state equations:

$$v(t) = \frac{dx(t)}{dt}, \quad (4)$$

$$a(t) = \frac{dv(t)}{dt}, \quad (5)$$

$$j(t) = \frac{da(t)}{dt}, \quad (6)$$

where $x(t)$ denotes the position, $v(t)$ the speed, $a(t)$ the acceleration and $j(t)$ the jerk.

On the other hand, train dynamics states that there are two resistive forces that affects to train movement. The first resistive force is the air resistance force that always opposes to movement and is expressed by well-known Davis formula:

$$R_{air}(t) = K_A + K_B \cdot v(t) + K_C \cdot v(t)^2, \quad (7)$$

where K_A , K_B and K_C are the drag coefficients.

The second resistive force is the slope force and it may oppose or favor train movement depending of the positive (uphill) or negative (downhill) value of the slope:

$$R_{slope}(t) = g \cdot \alpha(t) \cdot m, \quad (8)$$

where g is the gravitational acceleration, m is the static mass of the train and $\alpha(t)$ is the slope of the track in ‰ units (per mille), that is one vertical millimetre per horizontal meter.

Applying a force balance to the train dynamics equations:

$$F_{train}(t) = \sum F_i = F_{mot}(t) - R_{air}(t) - R_{slope}(t), \quad (9)$$

$$M \cdot a(t) = F_{mot}(t) - K_A - K_B \cdot v(t) - K_C \cdot v(t)^2 - g \cdot \alpha(t) \cdot m, \quad (10)$$

where M is the dynamic mass of the train, resulting from adding rotatory mass to static mass of the train and F_{mot} is the force applied by the engines of the train, positive values for powering and negative values for braking.

All the above equations yield to the next constraints:

$$\begin{cases} x(0) = 0, \\ x(T) = L, \end{cases} \quad (11)$$

$$\begin{cases} v(0) = 0, \\ v(T) = 0, \end{cases} \quad (12)$$

$$v(t) \leq \begin{cases} V_{max1} & \text{if } x(t) \leq X_1, \\ \vdots \\ V_{maxN} & \text{if } x(t) \leq X_N = L, \end{cases} \quad (13)$$

$$\alpha(t) \leq \begin{cases} A_{max1} & \text{if } x(t) \leq Y_1, \\ \vdots \\ A_{maxN} & \text{if } x(t) \leq Y_N = L, \end{cases} \quad (14)$$

$$|j(t)| \leq J_{MAX}, \quad (15)$$

$$-A_{SB} \leq a(t) \leq A_{MAX}. \quad (16)$$

The meaning of all the constants used in eqns (11) to (16) can be consulted in Table 1 below.

The behaviour of the electrical engines can be described as a set of hyperbolic curves limited by available adhesion. Next equations show the tractive and brake effort curves used by the model as constraints depending of train speed:

$$F_T(t) \leq \begin{cases} F_{T_max} & \text{if } v(t) \leq V_1, \\ \frac{F_{T_max} \cdot V_1}{v(t)} & \text{if } v(t) \leq V_2, \\ \frac{F_{T_max} \cdot V_1 \cdot V_2}{v(t)^2} & \text{if } v(t) \leq V_{MAX}. \end{cases} \quad (17)$$

$$F_B(t) \leq \begin{cases} 0 & \text{if } v(t) \leq V_3, \\ F_{B_max} & \text{if } v(t) \leq V_4, \\ \frac{F_{B_max} \cdot V_4}{v(t)} & \text{if } v(t) \leq V_{MAX}. \end{cases} \quad (18)$$

The meaning of all the constants used in eqns (17) and (18) can be consulted in Table 1 below. Note that electrical engines loss braking capacity at low speed due to its intrinsic design as can be observed from brake effort curve (18). This is automatically counterbalanced by control unit in real trains by using pneumatic brake blending. Therefore this effect must be considered for calculations in the model.

Table 1: Train constants.

Constant:	Meaning:
L	Trip distance.
T	Available trip time.
$X_1 \dots X_N$	Positions where speed profile changes.
$Y_1 \dots Y_N$	Positions where gradient profile changes.
$V_{\max 1} \dots$	Speed steps belonging to the speed profile.
$A_{\max 1} \dots$	Gradient steps belonging to the gradient profile.
J_{MAX}	Maximum jerk.
A_{MAX}	Maximum powering acceleration.
A_{SB}	Maximum service brake deceleration.
V_{MAX}	Maximum train speed.
F_{T_max}	Maximum tractive effort available.
F_{B_max}	Maximum braking effort available.
$V_1 \dots V_N$	Speeds where effort curves change behaviour.

It is necessary to mention that an extra constraints must be introduced in the model to avoid engines power and brake at the same time:

$$F_{mot}(t) = F_T(t) - F_B(t), \quad (19)$$

$$F_T(t) \cdot F_B(t) = 0. \quad (20)$$

Finally it is necessary to define the objective function to minimize, in this case the consumption:

$$c(t) = \int_0^T p(t) \cdot dt, \quad (21)$$

where $c(t)$ denotes the consumption and $p(t)$ denotes the power used in the engines. Now is the moment to differentiate between regenerative and rheostatic engines. Regenerative engines are able to send energy generated during braking back to the catenary so it can be used by other train or stored in the electrical substation if an energy storage system is available. In the other hand, rheostatic engines get rid of braking generated energy dissipating it as heat in a resistor bank

$$c_{reg}(t) = \int_0^T F_{mot}(t) \cdot v(t) \cdot dt \quad (22a)$$

$$c_{rhe}(t) = \int_0^T F_T(t) \cdot v(t) \cdot dt. \quad (22b)$$

From all the above exposed it is easily concluded that the control variables of this model are the tractive and brake efforts, $F_T(t)$ and $F_B(t)$, so now an optimal control formulation can be presented by using objective function defined in (21):

$$\min_{F_T(t), F_B(t) \in [0, T]} c(t) = \min_{F_T(t), F_B(t) \in [0, T]} \int_0^T F(t) \cdot v(t) \cdot dt. \quad (23)$$

Again, it is remarkable that $F(t)$ shall be $F_{\text{mot}}(t)$ or $F_T(t)$ depending of the rheostatic or regenerative nature of the system. Also, note that above formulation allows the insertion of electro-mechanical performance coefficients if available, minoring the energy regenerated and increasing the energy used for motion.

Finally, there is a last significant energy consumption due to auxiliary devices, as engine refrigeration or air conditioning

$$\int_0^T P_{\text{aux}} \cdot dt = P_{\text{aux}} \cdot T. \quad (24)$$

This one cannot be reduced but it affects global calculations, so it should be considered in the model. It is defined by an average constant power P_{aux} along travel time.

5 METHODOLOGY

Once the train model has been exposed, next step consists in using it within Ipopt package. Ipopt has interfaces to several programming languages as C/C++ or Python, third party algebraic modelling languages as AMPL and also allows other frameworks link to it to add an extra layer with additional features.

Interfacing Ipopt directly from a programming language is a very hard task due to tight data insertion format employed. It is expected that all the discretized state equations and constraints, the initial values, and first and second derivatives of state and control variables matrixes are provided prior to solve the problem as the next input data:

- Number of variables and constraints.
- Variable and constraints bounds.
- Initial values for the primal x variables and for the multipliers (only required for a warm start option).
- Number of nonzeros and sparsity structure in the Jacobian of the constraints.
- Number of nonzeros and sparsity structure in the Hessian of the Lagrangian function.
- Objective function, $f(x)$ and its gradient, $\nabla f(x)$.
- Constraint function values, $g(x)$ and its Jacobian, $\nabla g(x)^T$.
- Hessian of the Lagrangian function, $\sigma_f \nabla^2 f(x) + \sum \lambda_i \nabla^2 g_i(x)$, $i \in [1, m]$.

Therefore, calculating all these inputs manually is a time consuming and error prone task that only should be done for very simple problems. In this case, the model must be discretized as a multi-period problem, where the variables and equations increases accordingly with the discretization method chosen. For this kind of models, several thousands of variables and constraints are expected, so interface directly with a programming language must be discarded.

Fortunately, there are several libraries that calculate the above input using a more understandable input entry and freeing developers of most of complex math. These libraries directly interface with Ipopt, either calling to its binary application directly or statically linking with it.

This project uses Python SciPy packages for fast prototyping and C++ CasADi library for high performance code. Their main features are that they provide discretization methods ready to use and an automatic differentiation method to calculate gradient, Jacobian and Hessian matrixes. It is remarkable that all this software is open source and free to use for both research and commercial projects.



5.1 Model insertion

The train model shall be defined by eqns (4) to (24). All these equations must be included into the library and the model parameters must be set to match with real train.

The first issue that must be solved is that the model is a subject to time one, while all the track data is subject to distance. It is easy to transform the model into a continuous-distance system using eqn (4) in the rest of the equations. Especial care must be taken in boundaries of the problem with division by zero issues, taking into account that now the velocity variable is in the denominator of other terms.

Once the model is stated as a continuous-distance nonlinear differential equations system, it is straightforward to insert the speed profile and gradient profile as distance dependant constraints. Most operators apply speed limits while any part of the train is within, so they should be adjusted to affect all train length if necessary.

Also, changing the model to distance dimension prepares it to be easily modifiable to add new resistance forces if they affect appreciably the dynamics calculus, per example low adhesion areas, tunnel resistance or curve resistance. This last one is usually negligible for most trains except monorail train and big freight trains where its effects are really noticeable.

5.2 Discretization

Once the train dynamics model and all its configuration parameters have been inserted into the library, the next step is to choose a suitable discretization method. The best option for this kind of optimal control problems is to discretize by direct collocation with orthogonal polynomials method.

Direct collocation essentially trades nonlinearity for problem size, transforming the original problem into another larger but less nonlinear and with a sparsity structure that can be exploited efficiently by adding more degrees of freedom.

Usually collocation methods use Gauss-Legendre or Radau roots as collocation points. In this paper Radau roots are used because they allow large time steps and present a smaller truncation error, as it can be deduced of its properties:

- Collocation methods are A-stable, and both Gauss–Legendre and Radau collocation are AN-stable, or equivalently algebraically stable. As a result, there is no stability limitation for stiff problems.
- Radau collocation has stiff decay. Consequently, large time steps are allowed for stiff systems that capture steady state components and slow time scales.
- Both Gauss–Legendre and Radau collocations are among the highest order methods and their truncation error are $O(h^{2k})$ for Gauss–Legendre and $O(h^{2k-1})$ for Radau, where k is the polynomial degree.

The number of finite elements and the number of collocation points (related with the degree of the polynomial) must be chosen so it is possible to solve the problem with high accuracy but avoiding to oversize the problem increasing solving time and not contributing to accuracy. Heuristics and experience allow to choose the right values considering distance to travel and number of steps in the profiles.

It is noteworthy that the continuous set of discretized values do not have to be necessarily of fixed step. This is an interesting property that allows to add more discretization points in areas where the static speed profile or the gradient profile change, using less finite elements and resulting in smaller and faster to solve problems.

5.3 Solving and results

Once the discretization method has been applied on the model, Ipopt is ready to solve the problem and to provide the optimal speed profile and the state variables values as outputs.

Firstly the available travel time must be set so all the equation yields to it. The available travel time is chosen by railway operators and depends of a lot parameters as distance to travel, desirable speed and possible delay penalizations.

Then, the initial value for all the state variables must be set. Interior point methods are very sensitive to this configuration, converging slower if improper values are chosen or even becoming infeasible in extreme cases. It has been a normal practise for this kind of problems to use the average speed as initial values for velocity and its corresponding calculated timing for time.

By experience faster solutions can be obtained using the calculated state variables for the minimum time problem as initial values, where a train must run the travel distance as fast as permitted by static speed limits constrains. This can be done in an efficient way by approximate simulation as a previous stage.

Besides obtaining the optimal speed profile, the algorithm is prepare to calculate the optimal driving strategy at any moment that can be used to implement a DAS device.

Based in the physics principle of avoiding transitory regimes due to the loss of performance while they take place, four driving strategies have been defined: maximum powering, maximum braking, coasting and speed holding.

For maximum powering and braking driver sets train controls to apply either all the power of the engines or the maximum service brake. For speed holding strategy driver cruises at constant speed either powering or braking enough, while for coasting strategy driver stops applying any effort.

The main advantage of just having four pure strategies is that they are common for all rolling stock in service and understandable by any driver of any country. They are also applicable without further modifications in train or driving desk.

6 OUTCOMES

All the tests in this section have been carried out with a laptop mounting an Intel® Core™ i5-3210M @2.5 GHz and 4 GB DDR3 RAM running Windows 7 OS without any kind of tweak or overclocking to enhance performance. Even using this outdated device, all the tests have been solved in times under 12 seconds depending of track distance and complexity.

The tests have been done using a high-speed train model configured as indicated in Table 2 and Fig. 1 below.

Table 2: High speed train model.

Coefficient	Value	Coefficient	Value	Coefficient	Value
L (km)	170	A_{MAX} (m/s ²)	0.6	$F_{T\ max}$ (kN)	198
T (mm:ss)	51:34	A_{SB} (m/s ²)	0.6	$F_{B\ max}$ (kN)	180
V_{MAX} (km/h)	330	J_{MAX} (m/s ³)	1.0	P_{AUX} (kW)	130
L_{TRAIN} (m)	200	K_A (N)	2885	$\mu_{elect-mech}$	0.85
m (t)	357	K_B (N/(m/s))	96.12	μ_{reg}	0.95
M (t)	367	K_C (N/(m/s) ²)	6.3504	-	-



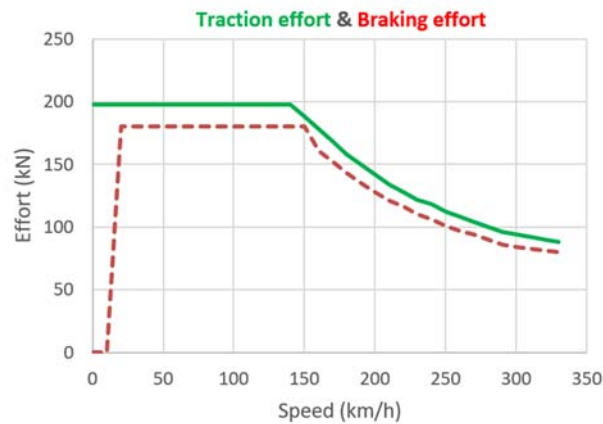


Figure 1: Traction and braking effort curves.

Where $\mu_{\text{elect-mech}}$ denotes electro-mechanical performance, including wheel-rail friction, gear train efficiency and catenary losses, and μ_{reg} denotes regenerative brake performance.

Fig. 2 shows the minimum time speed profile used as initial state while Fig. 3 shows the calculated optimal speed profile for given time schedule.

The figure shows speed limits, heights map, train speed and features the driving strategies represented as a colours schema over x-axis where green stands for powering, red for braking, grey for coasting and cyan for holding speed.

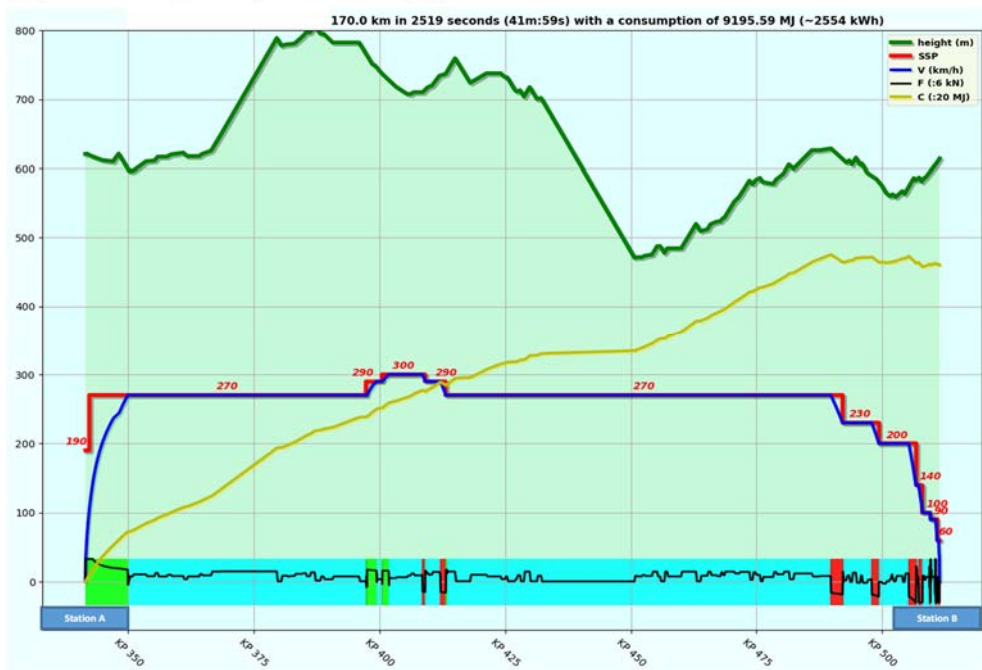


Figure 2: Minimum time problem.



Figure 3: Minimum energy consumption problem.

Observing the results is deduced that increasing travel time 22.8% the energy consumption reduces 31.6%. This is a stunning result valid only for the worst case, while normal driving employs longer travel times. Anyway, by experience a reduction of around 10% is expected using the right driving strategies for the same travel time.

As can be deduced this method works by trading travel time for energy saving in an efficient way. Here the reduction comes from using the coasting strategy in different sections and maintaining it along several kilometres taking advance of gradient profile.

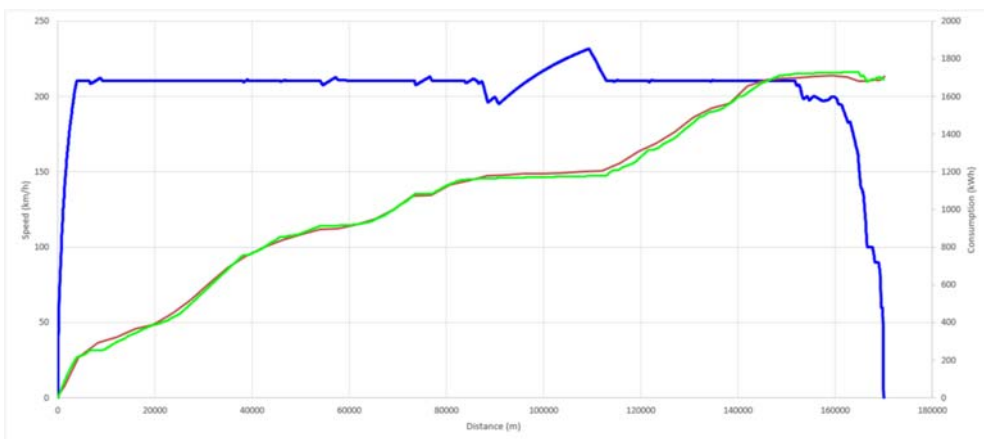


Figure 4: Optimal expected and real train consumptions.

7 CONCLUSIONS

The present paper demonstrates how a general algorithm has been successfully developed for solving the minimum energy consumption problem yield to time schedule, speed limits and gradients constraints of the track, and most important, how this can be solved efficiently.

The next complex requirements have been satisfied:

- Fast calculus in the order of seconds.
- High accuracy in estimated consumption and saving.
- Electric and diesel-electric detailed train modelling available.
- Relevant train features and track elements affecting consumption included.
- Able of running in any normal computer carried onboard.
- Able of recalculating an optimal speed profile if necessary due to delays.

Also, a very realistic train model has been developed that can be set using the information provided by the technical data sheet of any train, no matter its type or configuration.

This algorithm has been the base of ECOs tool, consisting in a framework specifically created from scratch for fulfilling all the above requirements. This tool has been designed to both calculate the expected energy saving prior to make any decision in office analysis or as a DAS main core within an on-board device.

Lastly, all the outputs obtained by the tool has been carefully validated through the use of a commercial railway electrical simulator suite and employing real information when the logs were available, as in Fig. 4, where green line is expected and red line is real consumption.

The error found was under 0.5% in the case of the simulator tool and under 5.0% in the case of real train data. The simulation error is due to using a different integrator method while the real life error is caused by natural human driving and normal mismatches between theoretical model and real train.

Future research is focus in improving fuel consumption estimation for diesel-electric trains, using a more realistic model that weighs the internal combustion engine speed change due to acceleration or brake. Currently implemented diesel model calculates fuel consumption by applying efficiency coefficients to the joints between the diesel and the electrical.

REFERENCES

- [1] Rodrigo, E., Tapia, S., Mera, J.M. & Soler, M., Optimizing electric rail energy consumption using the lagrange multiplier technique. *J. Transp. Eng.*, **139**, pp. 321–329. 2013.
- [2] Huang, Y., Ma, X., Su, S. & Tang, T., Optimization of train operation in multiple interstations with multi-population genetic algorithm. *Energies*, **8**, pp. 14311–14329. 2015.
- [3] Cabrera-Montie, W., Tapia-Fernandez, S., Jiménez-Alonso, F. & Aparicio-Izquierdo, F., Fuel optimization for road vehicles based on dynamic programming. *DYNA*, **87**(5), pp. 574–653, 2012.
- [4] Larrañaga, M., Anselmi, J., Ayesta, U., Jacko, P. & Romo, A., Optimization techniques applied to railway systems (hal-00780524), 2013.
- [5] Ben-Chaim, M., Shmerling, E. & Kuperman, A., Analytic modeling of vehicle fuel consumption. *Energies*, **6**, pp. 117–127, 2013.
- [6] Wächter, A. & Biegler, L.T., 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.

