Coasting advice based on the analytical solutions of the train motion model

Cunillera, Alex; Jonker, Harm H.; Scheepmaker, Gerben M.; Bogers, Wilbert H.T.J.; Goverde, Rob M.P.

DOI
10.1016/j.jrtpm.2023.100412

Publication date
2023

Document Version
Final published version

Published in
Journal of Rail Transport Planning and Management

Citation (APA)
Cunillera, A., Jonker, H. H., Scheepmaker, G. M., Bogers, W. H. T. J., & Goverde, R. M. P. (2023). Coasting advice based on the analytical solutions of the train motion model. *Journal of Rail Transport Planning and Management*, *28*, Article 100412. https://doi.org/10.1016/j.jrtpm.2023.100412

Journal of Rail Transport Planning & Management 28 (2023) 100412

Contents lists available at ScienceDirect

Journal of Rail Transport Planning & Management

journal homepage: www.elsevier.com/locate/jrtpm

Coasting advice based on the analytical solutions of the train motion model

Alex Cunillera [a,*], Harm H. Jonker [b], Gerben M. Scheepmaker [b], Wilbert H.T.J. Bogers [b], Rob M.P. Goverde [a]

[a] *Department of Transport and Planning, Delft University of Technology, Delft, The Netherlands*
[b] *Nederlandse Spoorwegen, Utrecht, The Netherlands*

ARTICLE INFO

Keywords:
Driver Advisory System
Energy-efficient train driving
Coasting
Train motion dynamics
Train operation

ABSTRACT

Supervision, data analysis and communication algorithms monitor trains, exploiting most of their available computational power. On-board eco-driving algorithms such as Driver Advisory Systems (DAS) are no exception, as the computational power available limits their complexity and features. This was the case of *Roltijd*, the in-house developed DAS based on coasting advice of NS, the main Dutch passenger railway undertaking. This platform calculated the coasting curves at every second by integrating the equations of motion numerically, assuming that the track is flat. However, generating more complex driving advice required replacing this coasting curve calculation by a more computationally-efficient algorithm. In this article we propose a new coasting advice algorithm based on the analytical solutions of the train motion model, assuming that gradients and speed limits are piecewise constant functions of the train location. We analyse the qualitative properties of these solutions using bifurcation theory, showing that bifurcations arise depending on the value of the gradient and the applied tractive effort. We validate the proposed algorithm, finding that our algorithm is accurate and can be 15 times faster than the previous method. This allowed NS to implement our algorithm on their trains, contributing daily to the sustainable mobility of 1.3 million passengers.

1. Introduction

Nowadays, trains are equipped with several computers and central processing units that continuously monitor the rolling stock, analyse and store the data measured by sensors implemented on-board, and communicate with external systems such as trackside equipment, the traffic management system and other platforms used by the railway undertaking and the infrastructure manager. All this consumes a large part of the computational power of trains, so efficient software and algorithms are an important asset whenever on-board computations are required.

On-board eco-driving algorithms such as Driver Advisory Systems (DAS) and Automatic Train Operation (ATO) are no exception. Some of these energy-efficient algorithms use Train Trajectory Optimization (TTO) to compute in real time the reference trajectories in order to improve the timeliness and energy usage. Scientific research on TTO aims to minimize total traction energy consumption of a train run given the running times of the timetable. We refer to Scheepmaker et al. (2017) and Yang et al. (2016) for a comprehensive overview on the topic of TTO. A DAS translates the computed trajectories into driving advice that can be accepted and followed by drivers. At NS, energy-efficient train driving is only applied if a train is not delayed. In turn, ATO algorithms calculate

[*] Corresponding author.
E-mail addresses: a.cunilleraperez@tudelft.nl (A. Cunillera), harm.jonker@ns.nl (H.H. Jonker), gerben.scheepmaker@ns.nl (G.M. Scheepmaker), wilbert.bogers@ns.nl (W.H.T.J. Bogers), r.m.p.goverde@tudelft.nl (R.M.P. Goverde).

https://doi.org/10.1016/j.jrtpm.2023.100412
Received 1 June 2023; Received in revised form 18 September 2023; Accepted 19 September 2023
Available online 3 October 2023

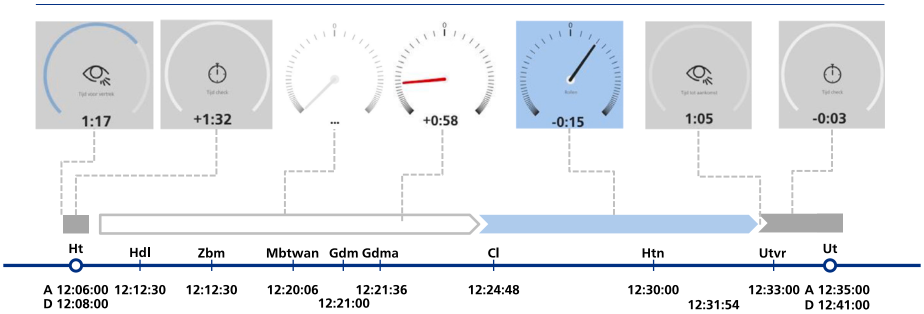


Fig. 1. *Roltijd* coasting advice during a long distance train run between the stops Ht ('s-Hertogenbosch) and Ut (Utrecht Central). The blue screen (number 5 from right) indicates that the train can start coasting and will be 15 s too early.

the right amount of tractive and brake effort required to track the reference trajectory. Furthermore, online model calibration is required to maintain over time the accurate performance of the mentioned eco-driving algorithms (Cunillera et al., 2022, 2023), exerting further pressure on the processing units.

In some cases, the restrictive computation power can be a major limitation when updating eco-driving algorithms to enhance their functionalities and increasing their complexity. This was the case of *TimTim*, the in-house real-time route information display app on a tablet for drivers developed by *Nederlandse Spoorwegen* (NS). The eco-driving strategy at NS is based on coasting. This is most convenient for the train drivers, and also the eco-driving strategy that has been applied several times in history at NS (i.e., even in the time of the steam engines train drivers at NS were stimulated to reduce the amount of coals used in order to save energy). Furthermore, Scheepmaker et al. (2020) showed that for the Dutch case a maximal coasting strategy (following cruising at the speed limit) leads to higher energy savings compared to a timetable based on a reduced cruising speed driving strategy (without coasting), mainly due to the relatively short distance between two stops. For a short distance train with a stop distance of 5 km, mechanical braking and 5% running time supplements, the maximum coasting driving strategy saves about 2.7% more energy than the reduced maximum speed (cruising), while for a long distance train with a stop distance of 50 km, mechanical braking and 5% running time supplements, the maximum coasting strategy saves about 8.2% more energy than the reduced maximum speed strategy (Scheepmaker et al., 2020). More information about the history of eco-driving at NS can be found in Luijt et al. (2016). The current eco-driving strategy implemented in the DAS of NS, i.e., *Roltijd* on the TimTim app is based on the UZI method (Universal energy-efficient driving idea or in Dutch *Universeel Zuinig rijden Idee*) developed by a train driver at NS (Scheepmaker and Goverde, 2015). The UZI method gives information using static tables to the train driver when to start coasting, depending on the available running time and speed limit between two stops. *Roltijd* computes the coasting curves dynamically in real-time. This is used to indicate to the driver the most appropriate time to start coasting in order to arrive on time to the next stopping point. This contributes to saving energy, while helping drivers to arrive on time and preventing unnecessary delays. Every second the system determines the train speed and location based on GPS measurements and calculates the expected arrival time, provided that the train starts coasting now. Then, the difference between the expected arrival time and the target time is displayed to the driver. The coasting advice is therefore provided as a countdown to start coasting. An example of the different screens shown by *Roltijd* during a train ride between two stops can be found in Fig. 1.

In *Roltijd* the journey between two stops is divided in seven phases, as shown in Fig. 1. From left to right, we show first the countdown to the scheduled departure time and the report of the actual departure time versus the scheduled one, both in grey. Then, in white, if the train is accelerating, the coasting curve is not calculated. If the train is cruising, the expected arrival time is calculated, provided that the train starts coasting now. If the train would arrive late, the information is shown in white. If the train would arrive on time or early, the information is shown in blue. The displayed information is shown in grey when the train is braking, since the driver must focus on the outside, and a countdown to the scheduled arrival time is shown. Last, the difference between the actual and the scheduled arrival times is displayed once the train has stopped.

Two main challenges had to be overcome before tackling the plans of the company to expand the functionalities of the DAS. First, the computational burden of *Roltijd* was limiting the implementation of more complex driving advice procedures. The coasting curves were calculated by integrating numerically the train motion model, which is the equation that describes the dynamics of a train running on a track. A small time step was used to keep the calculations accurate enough at the cost of the mentioned high computation time. However, some inaccuracies due to the numerical discretization procedure had been noticed. The forward Euler method was used as an integration method, which has a global error proportional to the integration step (Butcher, 2016). Second, *Roltijd* considered a flat track. Despite the fact that the Netherlands is one of the flattest countries in the world, this was leading to non-negligible deviations between the expected and the actual arrival times. For instance, drivers reported a deviation of 17 s at Schiphol Airport station, one of the most critical bottlenecks of the Dutch railway network, because Schiphol is located in a downhill section in a tunnel, and trains reached the brake curve earlier than expected since the track was considered flat in the calculation of the arrival time. Hence, there was a need to find a more efficient way of calculating the speed of a coasting train at a certain location and the time required to reach it. Consequently, the height profile of the track should also be introduced as input to the algorithm in the new coasting curve calculation method in order to increase the accuracy of the coasting advice.

These issues have been addressed and solved in a joint effort between Team Newton of NS, and the Department of Transport and Planning of Delft University of Technology. In this article we present the main results of this collaboration. We have developed a coasting curve calculation method based on the analytical solutions of the train motion model. The new algorithm uses as input the current speed and location of the train, the speed limits of the track, the distance to the next stopping point, the target arrival time, the brake curve to be followed based on the Dutch Automatic Train Protection system ATB, and the location of signs and signals. Regarding the gradients, a description of the height profile of the track consisting of piecewise constant gradients is also used as input. This assumption allows to find closed explicit expressions of the speed as a function of the time and vice versa, and of the location as a function of the speed and the time. The expressions of the speed and the time as a function of the location have been obtained as implicit formulas, that can be solved by means of a Golden-section Search algorithm (Kiefer, 1953). When the track is considered flat, the intersection between the coast and brake curves can be calculated in one go, reducing the computation time by a factor of 15 with respect to the numerical integration procedure. When gradients are considered, the speed and time have to be computed at the locations where the gradients change. Moreover, the analytical calculation of the coasting curves removes the global integration error of the numerical algorithm proportional to the integration step used.

Some analytical solutions of the train motion model have already been described in the existing literature. Franke et al. (2000) solved the train motion model analytically using the kinetic energy as the state variable instead of the speed. The running resistance is usually modelled as a quadratic function of the speed called the Davis equation (Davis, 1926). The model was simplified by removing the linear term of this speed-dependent running resistance. When the kinetic energy is used instead of the speed in the train motion model, the mentioned linear term is transformed into a term that depends on the square root of the kinetic energy. Thus, this nonlinearity was neglected in that approach. Ye and Liu (2017) solved the train motion model using the speed as the independent variable and piecewise constant gradients and speed limits. However, they modelled the maximum tractive effort and brake as piecewise linear functions of the speed. The maximum effort is generally considered to depend linearly on the speed at low speeds, since adhesion limits the maximum effort that can be applied before the wheels slip. At higher speeds, however, the maximum tractive effort is usually limited by the maximum power of the engine, so the maximum effort depends hyperbolically on the train speed (Brünger and Dahlhaus, 2014). Schank (2011) and Jaekel and Albrecht (2014) modelled the running resistance and the tractive effort as piecewise quadratic functions of the speed. This leads to a quadratic acceleration in the resulting differential equation, which is solved analytically in both references. Becker and Schreckenberg (2018) described analytical solutions of the train motion model, studying maximum traction, cruising, coasting and braking separately to compute the running times. They considered constant deceleration when braking, and disregarded the running resistance when applying maximum tractive effort. They modelled the maximum tractive effort as a piecewise quadratic function of the speed for low speeds, and as the power divided by the speed for high speeds. Moreover, they provide a closed expression for the mechanical energy consumed. Brünger and Dahlhaus (2014), derived similar analytical solutions of the train motion model considering also a maximum tractive effort that depends quadratically on the speed for low speeds. They also considered the hyperbolic dependence of the maximum tractive effort on the speed at high speeds, including the running resistances, however, the solutions of the resulting integrals are not shown. These solutions are expanded in Körner and Dahlhaus (2007), describing the acceleration as a polynomial, which is factorized after calculating its roots. This factorization allows to calculate the analytical solutions of the train motion model, however, the analytical formulas of the location as a function of the speed when maximum tractive power is applied, that is to say, when the maximum effort depends hyperbolically on the speed, have not been published yet in the existing literature.

In this article we propose a real-time coasting advice generator based on analytical solutions of the train motion model's differential equations. We extend the formulas mentioned in the previous paragraph and calculate the analytical solutions of the train motion model when the maximum tractive and brake effort is limited by the maximum power of the engine, including the mentioned formulas that up to this date were missing in the literature. We use the polynomial factorization introduced in Körner and Dahlhaus (2007) to calculate such formulas. Although the analytic solutions that arise when maximum engine power is applied are not used in the coasting advice algorithm, we include them in this article for completeness. These formulas could be used in the future for extending the driving advice calculation including acceleration phases at high speeds that require using the maximum power of the engine. Furthermore, we present a novel analysis of the analytical solutions of the train motion model differential equations based on the theory of dynamical systems (Strogatz, 2018) and the Fundamental Theorem of Algebra (Fine and Rosenberger, 1997). These solutions show bifurcations depending on the value of the gradient and the effort or power applied. The proposed coasting advice algorithm and the formulas implemented are also valid in the case of a constant applied tractive and brake effort, which will allow us to extend these analytical calculations in the near future to provide more complex driving advice.

We have tested the proposed coasting curve calculator in a case study. We have used MATLAB's ode45 function to validate it. This function solves ordinary differential equations using an adaptive fifth-order Runge–Kutta method. We have compared it with the previously-used numerical calculator and the coasting curves performed by a train trajectory optimizer based on a pseudospectral method (Scheepmaker et al., 2020; Goverde et al., 2021; Wang and Goverde, 2016), finding that the three of them produce similar results, but the proposed method is computationally more efficient than the other two. Although the formulas corresponding to a maximum tractive or brake power have not been implemented in the proposed coasting curve calculator, we also validate them with respect to ode45. The proposed coasting curve calculation method has already been implemented in *Roltijd* and in the current version of *TimTim*, so it is already contributing to transport timely 1.3 million passengers daily in a more sustainable way.

To sum up, the main contributions of this article are:

- A computationally-efficient coasting advice algorithm that indicates drivers when to start coasting to arrive on time while saving energy.

- A complete overview of the analytical solutions of the train motion model's differential equations, with closed formulas describing the location as a function of the speed under constant gradient, tractive effort that depends hyperbolically on the speed, and quadratic running resistance.
- A novel analysis of the qualitative behaviour of these solutions, interpreting the train motion model as a vector field and using the theory of dynamical systems to observe the bifurcations described by these solutions. Moreover, we prove the existence of fixed points that constitute terminal speeds of a train under constant gradient and applied tractive effort or power.

The structure of the remainder of this paper is outlined as follows: In Section 2 we further describe *TimTim*, *Roltijd*, the previous way of calculating the coasting advice, and the train trajectory optimizer used to validate the new method. In Section 3 we derive the analytical solutions of the train motion model's differential equations, analyse their algebraic properties and describe the new coasting curve calculation algorithm. Section 4 gathers the results of the validation of the proposed algorithm and the analytical formulas with respect to the other two methods. Last, in Section 5 we summarize the main conclusions of this research and mention our future steps.

2. Driving advice at Nederlandse Spoorwegen

2.1. Onboard driving advice applications: TimTim and Roltijd

TimTim is a Driver Advisory System application developed by the NS IT department. Drivers use the application on their personal tablet or on a built-in screen in the cabin. TimTim informs the driver about the timetable to follow, route settings, temporary speed restrictions and information about surrounding trains. This information helps the driver to drive the train in a punctual and energy efficient way. The application has a wireless connection with the back offices of NS and ProRail (the infrastructure manager) for receiving real time updates of this information.

In addition, TimTim contains *Roltijd*, a basic speed profile calculation module that continuously evaluates the current position of the train (based on the tablet or train GPS) with respect to the target arrival at the next station and indicates to the driver when to start coasting in order to arrive on time to the next stop. The expected arrival time is recalculated every second and the corresponding coasting advice is updated in the application, helping drivers to save energy.

Two different stop approaches are considered:

1. If, on the one hand, the train must stop and there are speed reductions due to speed signs or signals, the coasting curve ends at a specific location and then the train brakes following an ATB braking curve with constant deceleration. For instance, this is used to approach a switch or a curve that forces the train to reduce its speed.
2. If, on the other hand, there is no explicit speed reduction, the train coasts until it meets a comfort braking curve, which again is based on a constant deceleration.

The main difference between these two scenarios is that the end of the coasting curve in the latter case is not fixed, but speed-dependent. In both situations, the train is expected to drive at 40 km/h at the start of the platform and stops at the stopping point by following a comfort brake curve. The arrival time is calculated as the sum of coasting time to the braking location, the ATB or comfort braking time until reaching 40 km/h, the time to drive along the platform at that speed and the time to stop from 40 km/h to a standstill following a comfort braking curve. The locations of signs, signals and stopping points are known with an accuracy of a cm.

2.2. Train trajectory optimization based on a pseudospectral method

We used the solver PROMO to compute the energy optimal train trajectory. The aim of the train trajectory optimization problem is to minimize total traction energy consumption of a train given the amount of running time in the timetable (Scheepmaker et al., 2017), while considering rolling stock and track characteristics such as the speed-dependent running resistance, the maximum tractive and brake effort and power that can be applied, the speed limits and the height profile of the track. PROMO is a solver developed in MATLAB using the GPOPS (General Purpose OPtimal control Software) toolbox version 4.1 (Rao et al., 2010). A Radau pseudospectral method is implemented in GPOPS to solve the optimal control problem (Rao et al., 2011). More details of PROMO can be found in Scheepmaker et al. (2020b). To discretize the optimal control problem, the Radau pseudospectral method applies orthogonal collocation at the Legendre–Gauss–Radau (LGR) points. Afterwards, the optimal control problem is transcribed into to a nonlinear programming (NLP) problem, which can be solved using standard NLP solvers (Betts, 2010). GPOPS has the possibility to evaluate the costates and Hamiltonian to validate whether the results are in line with optimal control theory. Wang and Goverde (2016) and Goverde et al. (2021) give more details regarding the discretization of the optimal control problem. We applied PROMO in different case studies to compute the energy-optimal driving strategy as well as other driving strategies (Scheepmaker et al., 2020; Scheepmaker and Goverde, 2020) and include the energy-efficient train trajectories into timetable design (Scheepmaker et al., 2020b; Scheepmaker and Goverde, 2021). In this article, we use PROMO to validate the coasting advice algorithms.

2.3. Numerical coasting curve calculation based on the forward Euler's integration method

Roltijd's coasting curve calculation was performed by integrating numerically the equations of motion of the train (Brünger and Dahlhaus, 2014),

$$m\rho \frac{dv}{dt} = F(t) - R(v) - W(s), \tag{1}$$

$$\frac{ds}{dt} = v, \tag{2}$$

where s is the train location (in m), v the train speed (in m/s), t the time (in s), m the train mass (in kg), ρ the rotating mass factor that accounts for the inertia of the rotating parts of the train (it is dimensionless), $F(t)$ is the tractive effort and brake applied by the engine (in N), $R(v)$ is the running resistance of the train (in N) and $W(s)$ is the resistance to motion due to the track geometry (in N). In order to simplify (1), we can divide both sides of the equation by $m\rho$, obtaining a mass-specific version of the equation,

$$\frac{dv}{dt} = f(t) - r(v) - w(s), \tag{3}$$

where $f(t) = F(t)/(m\rho)$, $r(v) = R(v)/(m\rho)$ and $w(s) = W(s)/(m\rho)$.

The maximum tractive effort that can be applied is usually modelled as constant or linearly-dependent on the speed at low speeds, and limited by the maximum power of the engine $p_{\max}$ at higher speeds, where $p_{\max}$ is in W/kg.

$$f(t) \leq \begin{cases} f_{\max}, & \text{for } 0 \leq v < p_{\max}/f_{\max}, \\ p_{\max}/v, & \text{for } v \geq p_{\max}/f_{\max}. \end{cases} \tag{4}$$

Regarding the bounds on the brake effort, we assume for simplicity that the braking capacity of the train is large enough so that the train can follow predefined brake curves at any time and under any operating condition. The analytical solutions of the train motion model presented in Section 3 would also be valid if the maximum brake is modelled similar to the maximum traction.

The running resistance is usually modelled as a quadratic function of the speed, and it is usually called the Davis equation (Davis, 1926),

$$r(v) = r_0 + r_1 v + r_2 v^2, \tag{5}$$

where $r_0, r_2 > 0$ and $r_1 \geq 0$, in N/kg, N/(kg(m/s)^2) and N/(kg m/s), respectively.

The term related to the track geometry is usually modelled as

$$w(s) = \frac{g}{\rho} \sin(\theta(s)), \tag{6}$$

where $g = 9.81$ m/s^2 is the gravity acceleration and $\theta(s)$ is the slope in rad of the track at location s. Therefore, $w(s)$ is positive when the train is running on an uphill section, negative on a downhill section and zero on a flat track.

When the train is coasting, f is zero. Then, if a train starts coasting at time t_0 and considering the state of the train at that time, $s_0 = s(t_0)$ and $v_0 = v(t_0)$, the location and speed of the train can be calculated numerically by means of the forward Euler integration method. The acceleration between two consecutive calculations is assumed to be constant, so from (2) and (3) we deduce

$$s_{k+1} = s_k + v_k \Delta t - \frac{1}{2}(r(v_k) + w(s_k))\Delta t^2, \tag{7}$$

$$v_{k+1} = v_k - (r(v_k) + w(s_k))\Delta t, \tag{8}$$

where Δt is the integration time step, $s_k = s(t_k)$, $v_k = v(t_k)$ and $t_k = t_0 + k\Delta t$. In *Roltijd*, Δt is equal to 1 s. Fig. 2 illustrates the computation of a coast curve using the forward Euler method. Each vertical line represents a time step where the speed is calculated from the speed and location at the previous time step. The dark blue curve in the upper figure represents the train speed during coasting, while the yellow lines represent the speed limit. In the lower figure we show the resistance to motion due to the gradients in red. The full line at zero resistance shows that the old algorithm considers the track flat. As we can observe in this example, the gradients shown as red dotted lines are neglected. This implementation utilizes only four parameters, r_0, r_1, r_2 and m, which are known for each rolling stock set and are considered to be constant along each journey. *Roltijd* is based on the assumption that in the last phase of reaching the stopping point location the train is always forced to reduce speed from maximum track speed (or cruising speed) to 40 km/h through a series of speed reduction signals or signs (e.g. 80, 60, 40). Exact locations of these speed reduction points are known in the data and speed reduction is performed strictly following the ATB braking criteria. This brake curve is precomputed by considering constant deceleration rates. At every location update Eqs. (7) and (8) are utilized recursively until the coasting curve intersects the brake curve or it reaches the location where the coasting curve must end. This is used to calculate the expected arrival time to the next stopping point if the train started coasting now, that is to say, at t_0. The difference between the expected and the target arrival time is displayed to the driver as a countdown to start coasting. *Roltijd* updates this information every second by repeating this procedure.

Despite the simplicity of this algorithm, the error of the forward Euler method per integration step is proportional to the square of the integration time step, while the accumulated error over all the iterations is proportional to the integration time step (Butcher, 2016). Therefore, the error of this algorithm at the brake curve may be in the order of a few seconds. This numerical coasting advice calculation method has some shortcomings. It requires too many computations per second, which compromises the limited

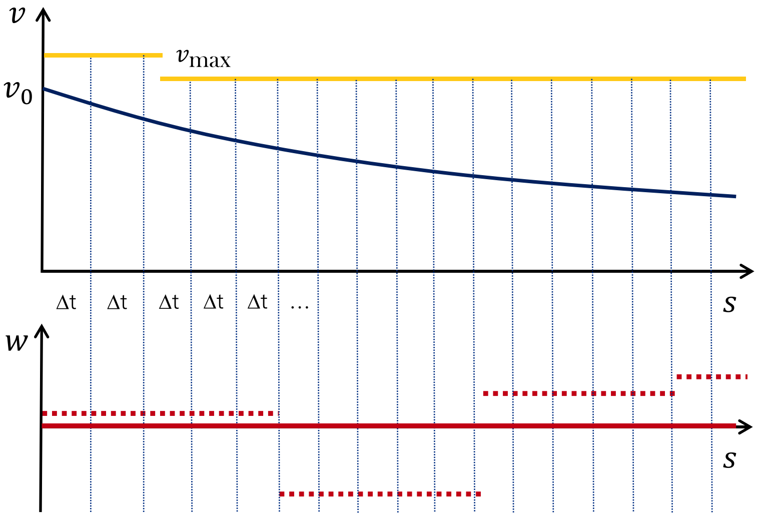


Fig. 2. Schematic representation of a coast curve calculated by means of the forward Euler method with a time step equal to Δt s.

computational resources of the tablet. NS aims to extend the coasting advice into driving advice along the entire trip, including multiple coasting sections over intermediate timing points and traction, brake and cruise advice. This will require more efficient calculations, as well as supporting varying track gradients. Moreover, NS moved from a minute based timetable to a 6 s interval based timetable, which requires more accuracy from drivers. Inaccuracies on the coasting advice that lead to deviations of more than 6 s are therefore not desirable.

3. Generating coasting advice using the analytical solutions of the train motion model

3.1. Analysis of the analytical solutions of the train motion model

3.1.1. Constant tractive and brake effort

Eqs. (2) and (3) can be considered as a dynamical system depending on two parameters: the applied effort and the resistance due to the track geometry. Considering these two parameters to be constant, we proceed to analyse the dependence of solutions of the train motion model on these two parameters using the theory of dynamical systems (Strogatz, 2018). This analysis is essential for calculating the analytical solutions and for the coasting advice algorithm, as we will show later in this article.

We can consider the acceleration as a quadratic polynomial with the speed as its independent variable in each track interval where the applied effort f and the track resistance term w are constant,

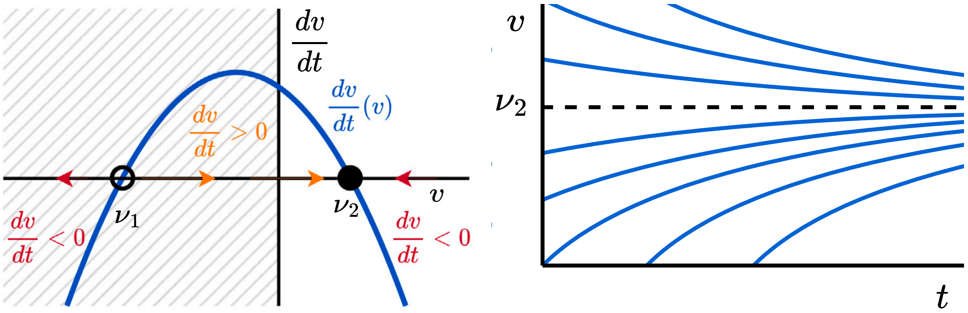
$$\frac{dv}{dt} = f - w - r_0 - r_1 v - r_2 v^2. \tag{9}$$

In fact, (9) can be interpreted as a one-dimensional vector field when the speed is considered as its independent variable, and we can find easily its fixed points, which are the speeds at which the acceleration is zero. According to the Fundamental Theorem of Algebra (Fine and Rosenberger, 1997), a quadratic polynomial always has two roots and three possible scenarios may occur, depending on the values of f and w:

- **Case a:** The roots of the polynomial are two complex conjugates, $v_1, v_2 \in \mathbb{C}$, with $v_2 = \bar{v}_1$.
- **Case b:** The polynomial has a non-positive real double root at the maximum of the parabola described by the polynomial, $v_1 = v_2 = -\frac{r_1}{2r_2} \in \mathbb{R}$. We define α based on this double root, $\alpha := \frac{r_1}{2r_2} \geq 0$, since $r_1 \geq 0$ and $r_2 > 0$.
- **Case c:** The polynomial has two real, simple solutions, v_1 and v_2, with $v_2 > v_1$. Moreover, v_1 will always be negative.

Each of these three cases lead to a different formula when calculating the analytical solution of (9). Therefore, the roots v_1 and v_2 of the acceleration polynomial have to be calculated first in order to identify which is the formula to be used.

We have to observe the sign of the acceleration as a function of the train speed in order to understand the physical meaning of each of these three cases. In cases a and b, the acceleration is negative for any value of the speed, except at speed $-\alpha$ in case b, where the acceleration would be zero. However, this speed is negative, so it is not physically meaningful. In case c, the acceleration polynomial has two fixed points: v_1, an unstable fixed point, and v_2, a stable one. The acceleration is negative for any speed below v_1 and for any speed larger than v_2, and positive between v_1 and v_2. Fig. 3(a) shows the phase portrait of case c with positive v_2, where the acceleration polynomial is represented in blue in an acceleration-speed diagram (see Fig. 3).



(a) Phase portrait of case c.

(b) Vector flow around a terminal speed ν_2.

Fig. 3. Dynamics of the acceleration polynomial in case c, where the stable fixed point ν_2 constitutes a terminal speed.

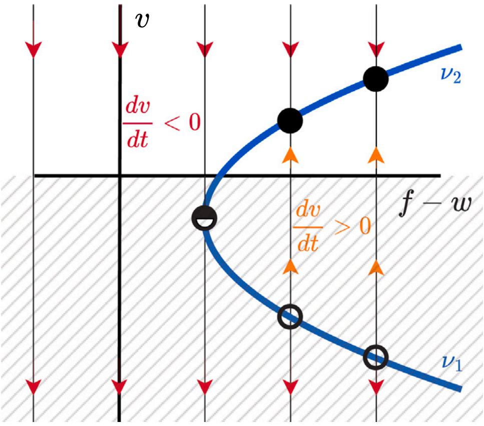


Fig. 4. Bifurcation diagram of (9).

In Fig. 3(a) the negative speeds part of the phase portrait is shaded since it is not physically meaningful. The unstable and stable fixed points, ν_1 and ν_2 are represented by empty and full dots, respectively. ν_2 is the terminal speed for the effort f and the gradient resistance w. This fact is particularly relevant if ν_2 is positive, for instance, when coasting on a steep donwhill section or when applying traction on a non-steep section. If a train is running at a speed higher than ν_2, it will decelerate towards ν_2, and, if it is running below that speed, it will accelerate towards it. Nevertheless, this terminal speed can never be reached at a finite time if f and w are kept constant, as ν_2 is approached asymptotically. This is due to the fact that ν_2 is a stable fixed point of the vector field (9). In the coasting advice algorithm we used this fact to avoid calculating the location or passing time at speeds that cannot be reached by coasting under the considered gradient. Fig. 3(b) shows the vector flow of (9) around a positive terminal speed ν_2. Thus, case c can only be found in particular scenarios, i.e. when applying constant traction on non-steep uphill track sections or when a train is coasting on steep downhill sections. The deceleration over speed ν_2 takes place at high speeds, where the applied effort is not enough to compensate the running and track resistances. In turn, cases a and b may take place when the train is braking or coasting at non-steep downhill gradients, or, again, when the applied effort is insufficient.

Fig. 4 shows the bifurcation diagram of (9), where the vertical axis represents the train speed and the horizontal axis the difference between the applied effort and the gradient resistance, $f - w$. The vertical lines represent constant values of $f - w$, and the arrows over them show the sign of the acceleration at those speeds. Since the running resistance parameters are considered constant and fixed, only variations in the term $f - w$ may lead to qualitative changes in the solutions of the dynamical system (9). The blue line shows the fixed points of the vector field as a function of $f - w$. At any fixed value of $f - w$, the solid dot represents the stable fixed point ν_2 and the open one the unstable fixed point ν_1. We observe a bifurcation: for negative and the smaller positive values of $f - w$, the dynamical system has no real roots, leading to case a. Then, for a particular value of $f - w$, the dynamical system has only a semistable fixed point, corresponding to case b, with $\nu_1 = \nu_2$. Last, for higher values of $f - w$, case c arises and the dynamical system has two real simple roots.

3.1.2. Constant tractive and brake power

When the train is accelerating at maximum power, $f = p_{\max}/v$, with $p_{\max} > 0$, (3) turns into

$$\frac{dv}{dt} = \frac{p_{\max}}{v} - w - r_0 - r_1 v - r_2 v^2 = \frac{1}{v}(p_{\max} - (w + r_0)v - r_1 v^2 - r_2 v^3). \tag{10}$$

Taking $1/v$ as a common factor in the right-hand-side term of (10), we can write it as $1/v$ multiplied by a cubic polynomial. The analysis of (10) is similar to the one presented for the constant effort, so we will cover it briefly. It shows that this vector field always has a real positive root v_3, and the other two can be either complex conjugates, a real negative double root, or two real negative simple roots. Again, these three scenarios lead to analytical solutions with different formulas. Regarding the physics of the fixed points, v_3 is the only one that may be of interest, since it is a stable fixed point that, if it is large enough, may constitute again a terminal speed.

The case in which a train is braking at the maximum braking power $f = -p_{\max}/v$, with $p_{\max} > 0$ is analogous to the maximum tractive power, but changing the signs of all the roots. A negative real root is always found, and, two positive simple roots may be found if the track is sufficiently downhill. Then, the largest of the roots would be a terminal speed where the downhill gradient balances the running resistance and the brake power, and the other positive root would be an unstable fixed point, where any perturbation would deviate the train speed from that fixed point. Nevertheless, the Fundamental Theory of Algebra can be used to prove that braking at maximum power allows for a special case in which a negative triple root is found at $v_1 = -2\alpha/3$. This special case can only arise if r_1 is strictly positive in sections where $r_0 + w = 4r_2\alpha^2/3 > 0$ and $p_{\max} = -8r_2\alpha^3/27$.

3.2. Analytical solution of the train motion model's differential equations

3.2.1. Constant tractive and brake effort: analytical solutions

Considering a track section in which the applied tractive effort and brake, and the gradient are constant, we may rewrite the right-hand-side term of (3) as a quadratic function of the speed,

$$\frac{dv}{dt} = (f - w - r_0) - r_1 v - r_2 v^2. \tag{11}$$

This equation can be integrated easily since its right-hand-side term does not depend on the time. Indeed, since $r_2 > 0$,

$$\int_{v_0}^{v} \frac{dv}{v^2 + \frac{r_1}{r_2} v - \frac{f-w-r_0}{r_2}} = -r_2 \int_{t_0}^{t} dt. \tag{12}$$

Completing the square of the denominator of the integrand of the speed term, we find

$$\int_{v_0}^{v} \frac{dv}{\left(v + \frac{r_1}{2r_2}\right)^2 - \left(\frac{r_1}{2r_2}\right)^2 - \frac{f-w-r_0}{r_2}} = -r_2 \int_{t_0}^{t} dt. \tag{13}$$

Taking again $\alpha = \frac{r_1}{2r_2}$ and defining $\beta^2 = \alpha^2 + \frac{f-w-r_0}{r_2}$, we observe that $\alpha \geq 0$ since $r_2 > 0$ and $r_1 \geq 0$. However, β^2 shows a more complex behaviour, since it can be zero, positive or negative. In this latter case, we use $\gamma^2 = -\beta^2$, so γ is real. This leads to three different formulas (Gradshteyn and Ryzhik, 2014). If $\beta > 0$,

$$r_2\beta(t - t_0) = \frac{1}{2}\log\left(\frac{\beta + v + \alpha}{\beta - v + \alpha}\,\frac{\beta - v_0 + \alpha}{\beta + v_0 + \alpha}\right), \tag{14}$$

where log is the natural logarithm. If $\beta = 0$, then

$$r_2(t - t_0) = \frac{1}{v + \alpha} - \frac{1}{v_0 + \alpha}. \tag{15}$$

If β is complex, then

$$-r_2\gamma(t - t_0) = \arctan\left(\frac{v + \alpha}{\gamma}\right) - \arctan\left(\frac{v_0 + \alpha}{\gamma}\right). \tag{16}$$

This allows to calculate explicitly $v(t)$ and $t(v)$.

From (2) and (3), we can obtain a differential equation that describes the evolution of the train speed with respect to its location, which is particularly relevant in railways since several elements of the track are located at specific location, not times. Stopping locations, signals, points, gradients and speed limits are some examples.

$$\frac{dv}{ds} = \frac{1}{v}(f(s) - r(v) - w(s)). \tag{17}$$

This equation can also be integrated to find formulas that relate the speed and the location. $s(v)$ is found explicitly, however, this is not the case with $v(s)$, which has to be calculated iteratively by means of a search algorithm such as the Golden-section Search (Kiefer, 1953). The remaining two formulas, $s(t)$ and $t(s)$, can be calculated as $s(v(t))$ and $t(v(s))$, respectively.

If $\beta > 0$, then

$$-2r_2\beta(s - s_0) = \beta\log\frac{(v + \alpha)^2 - \beta^2}{(v_0 + \alpha)^2 - \beta^2} + \alpha\log\left(\frac{\beta + v + \alpha}{\beta - v + \alpha}\,\frac{\beta - v_0 + \alpha}{\beta + v_0 + \alpha}\right). \tag{18}$$

If $\beta = 0$, then

$$-r_2(s - s_0) = \log\frac{v + \alpha}{v_0 + \alpha} + \alpha\left(\frac{1}{v + \alpha} - \frac{1}{v_0 + \alpha}\right). \tag{19}$$

If β is complex, then

$$-r_2\gamma(s-s_0)=\frac{\gamma}{2}\log\frac{(v+\alpha)^2+\gamma^2}{(v_0+\alpha)^2+\gamma^2}-\alpha\left(\arctan\left(\frac{v+\alpha}{\gamma}\right)-\arctan\left(\frac{v_0+\alpha}{\gamma}\right)\right). \tag{20}$$

The analysis of the bifurcation of (9) shows that when the train is accelerating β can only be positive, while when it is decelerating β can be real and positive, zero or complex. Thus, the appropriate formulas have be selected based on the value of β when the train decelerates. A special case arises when the forces on the train balance and the acceleration is zero. In this case, the time, location and speed can be calculated by considering the speed to be constant.

3.2.2. *Constant tractive and brake power: analytical solutions*

For completeness, we show the analytical solutions of (3) and (17) under a constant gradient and $f=p_{\max}/v$. We can rewrite (3) and (17) as

$$\frac{dv}{dt}=\frac{-r_2}{v}(v-v_1)(v-v_2)(v-v_3), \tag{21}$$

$$\frac{dv}{ds}=\frac{-r_2}{v^2}(v-v_1)(v-v_2)(v-v_3), \tag{22}$$

where v_1, v_2 and v_3 are the roots of the cubic polynomial that arises after extracting an extra $-r_2/v$ factor from the right-hand-side of (3) and (17). Again, according to the Fundamental Theorem of Algebra, one of these roots (for instance, v_1) is always real, and the other two can be complex conjugates, a real double root or two real single roots. These three cases lead to the following formulas (Gradshteyn and Ryzhik, 2014), which are explicit only when the speed is the independent variable:

If the polynomial has three real simple roots, v_1, v_2 and v_3, then

$$-r_2(t-t_0)=\frac{1}{(v_1-v_2)(v_1-v_3)(v_2-v_3)}\left(v_1(v_2-v_3)\log\frac{v-v_1}{v_0-v_1}+v_2(v_3-v_1)\log\frac{v-v_2}{v_0-v_2}+v_3(v_1-v_2)\log\frac{v-v_3}{v_0-v_3}\right), \tag{23}$$

$$-r_2(s-s_0)=\frac{1}{(v_1-v_2)(v_1-v_3)(v_2-v_3)}\left(v_1^2(v_2-v_3)\log\frac{v-v_1}{v_0-v_1}+v_2^2(v_3-v_1)\log\frac{v-v_2}{v_0-v_2}+v_3^2(v_1-v_2)\log\frac{v-v_3}{v_0-v_3}\right). \tag{24}$$

If the polynomial has a simple real root v_1 and a double real one, v_2, then

$$-r_2(t-t_0)=\frac{v_1}{(v_1-v_2)^2}\log\left(\frac{v-v_1}{v_0-v_1}\frac{v_0-v_2}{v-v_2}\right)+\frac{v_2}{v_1-v_2}\left(\frac{1}{v-v_2}-\frac{1}{v_0-v_2}\right), \tag{25}$$

$$-r_2(s-s_0)=\frac{v_1^2}{(v_1-v_2)^2}\log\frac{v-v_1}{v_0-v_1}+\frac{v_2^2-2v_1v_2}{(v_1-v_2)^2}\log\frac{v-v_2}{v_0-v_2}+\frac{v_2^2}{v_1-v_2}\left(\frac{1}{v-v_2}-\frac{1}{v_0-v_2}\right). \tag{26}$$

If the polynomial has a real root v_1 and two complex conjugate roots, $v_2=a+bi$, $\bar{v}_2=a-bi$, then

$$-r_2(t-t_0)=\frac{1}{(v_1-v_2)(v_1-\bar{v}_2)}\left[v_1\log\frac{v-v_1}{v_0-v_1}-\frac{v_1}{2}\log\frac{v^2-2av+a^2+b^2}{v_0^2-2av_0+a^2+b^2}+\frac{a^2+b^2-v_1a}{b}\left(\arctan\frac{v-a}{b}-\arctan\frac{v_0-a}{b}\right)\right], \tag{27}$$

$$\begin{aligned}-r_2(s-s_0)={}&\frac{1}{(v_1-v_2)(v_1-\bar{v}_2)}\left[v_1^2\log\frac{v-v_1}{v_0-v_1}+\frac{a^2+b^2-2av_1}{2}\log\frac{v^2-2av+a^2+b^2}{v_0^2-2av_0+a^2+b^2}+\right.\\&\left.+\frac{v_1b^2+a(a^2+b^2-av_1)}{b}\left(\arctan\frac{v-a}{b}-\arctan\frac{v_0-a}{b}\right)\right].\end{aligned} \tag{28}$$

Last, when applying constant brake power, if the polynomial has a negative triple root, $v_1=-2\alpha/3$, then

$$-r_2(t-t_0)=-\frac{1}{2}v_1\left(\frac{1}{(v-v_1)^2}-\frac{1}{(v_0-v_1)^2}\right)-\left(\frac{1}{v-v_1}-\frac{1}{v_0-v_1}\right), \tag{29}$$

$$-r_2(s-s_0)=-\frac{1}{2}v_1^2\left(\frac{1}{(v-v_1)^2}-\frac{1}{(v_0-v_1)^2}\right)-2v_1\left(\frac{1}{v-v_1}-\frac{1}{v_0-v_1}\right)+\log\frac{v-v_1}{v_0-v_1}. \tag{30}$$

3.2.3. *A remark on alternative ways of modelling the tractive and brake effort*

In case that the maximum tractive effort is modelled as a linear or quadratic function of the speed, the acceleration polynomial could be carefully rewritten as a quadratic function of the speed and the formulas proposed in Section 3 could still be used (Ye and Liu, 2017; Becker and Schreckenberg, 2018). If the quadratic term is cancelled with r_2, solving the resulting integrals becomes trivial.

3.3. *The new coasting advice algorithm*

Algorithm 1 summarizes the calculation of the coast curve. This algorithm aims to determine the arrival time at the next stop if the train starts coasting at the current location. It runs every second and takes as input the current speed v_0, location s_0 and time t_0, the target arrival time, the track gradients and speed limits, the train mass and its running resistance parameters. Since the train is coasting, we consider $f=0$ in all the calculations. We split the distance to the next stop or target location in sections where the gradient and speed limit are constant and calculate the parameter β or γ in each section, as explained in Section 3.2.1. Then, the speed and time at the end of each section is calculated by means of (14)–(16), (18)–(20) until a fixed target location or a comfort brake curve is reached. As we mentioned in Section 2.1, the coast curve must end at certain target locations when there are speed reductions ahead due to speed signs and signals. The train must decelerate after reaching the target location by following an ATB brake curve. In turn, comfort brake curves are used when there are no explicit speed reductions. Both types of brake curves are precomputed and stored beforehand. These brake curves utilize constant deceleration rates, so the speed of a brake curve can be calculated explicitly as a function of the location. It can be introduced in (18)–(20) to obtain the intersection point between the coast and the brake curves. The algorithm also accounts for the possibility of reaching a speed limit while coasting under steep downhill sections. In this scenario, the driver is supposed to brake the train to maintain a speed equal to the mentioned speed limit and avoid overspeeding. After that, the algorithm checks whether a comfort brake curve is reached. The expected arrival time is calculated by summing the time spent on coasting, braking to the restricted speed 40 km/h, braking from 40 km/h to a standstill and cruising at 40 km/h between these two latter decelerations. The driver is updated with the difference between the calculated expected arrival time and the target one.

Algorithm 1 Coasting advice algorithm based on analytical formulas

```
1:  Initialize parameters and measure train state: $v = v_0$, $s = s_0$, $t = t_0$
2:  for each section do
3:      Update section gradient, speed limit $v_{\max}$, length, $\beta$ or $\gamma$
4:      $v \leftarrow$ Calculate speed at the end of section using (18)–(20)
5:      if Speed limit exceeded then
6:          Calculate location at which speed limit is reached using (18)–(20)
7:          $v \leftarrow v^{\max}$
8:          $v^{\max}$ to be held until end of section
9:      end if
10:     if Comfort brake curve reached then
11:         Calculate intersection between coast and comfort brake curves using (18)–(20)
12:         $t \leftarrow t +$ Calculate time until brake curve using (14)–(16) and/or constant speed
13:         go to 18
14:     end if
15:     $t \leftarrow t +$ Calculate time until end of section using (14)–(16) and/or constant speed
16:     $s \leftarrow s +$ section length
17: end for
18: $t \leftarrow t +$ Calculate remaining time to stopping to a standstill
19: Show difference between $t$ and target arrival time to the driver
```

Fig. 5 illustrates the new algorithm for computing coast curves. The train speed during coasting, the speed limit and the resistance to motion due to track gradients w are shown in dark blue, yellow and red, respectively. The vertical lines show the locations where the new algorithm calculates the state of the train, namely when either the gradients or the speed limits change. Compared to the old algorithm shown in Fig. 2, the state of the train is only calculated at the locations where the gradients or the speed limit change. Moreover, if the track is not flat, the computations of the old algorithm may differ significantly with respect to the new algorithm's and with reality.

4. Validation of the new method and the constant tractive and brake power formulas

4.1. Validation of the new method

In this section we show the results of the validation of the proposed coasting advice algorithm. We have simulated three scenarios with different track geometries and compared the performance and accuracy of the proposed method with the one implemented previously in *Roltijd*, which is based on the forward Euler method, and with the coasting curves generated by PROMO.

Table 1 gathers the track geometry configuration of these scenarios. In each of them, the track is 5 km long. The track is flat before 1.1 km and after 2.4 km in all scenarios. Regarding the track between 1.1 km and 2.4 km, in the first scenario the track is slightly downhill, with a −0.681 permil slope. This corresponds to the $\beta = 0$ case. The second and third scenarios have a 9.33 permil downhill and uphill slope, between 1.1 km and 2.4 km, respectively. Under flat track and in the considered uphill slope, β^2 is negative, so Eqs. (16) and (20) have to be used to calculate the speeds and locations on those track intervals. In turn, coasting at the downhill gradient leads to a positive β^2, so Eqs. (14) and (18) are the formulas required in this case. Coasting at the −0.681 permil slope requires using Eqs. (15) and (19). We considered an NS Sprinter train rolling stock type SLT-6 in this case study. Its characteristics are outlined in Table 2.

We have implemented both coasting advice algorithms in the same platform, using the same language code. Figs. 6(a) and 6(b) show the coasting curves calculated using these three methods. For this comparison, we have included the gradient resistance in the forward Euler method. Fig. 6(a) represents the speed curves as a function of the location, and Fig. 6(b) the passing times. We have represented in yellow the $\beta = 0$ scenario, in turquoise the downhill one, and in purple the uphill. We show in full-coloured dashed

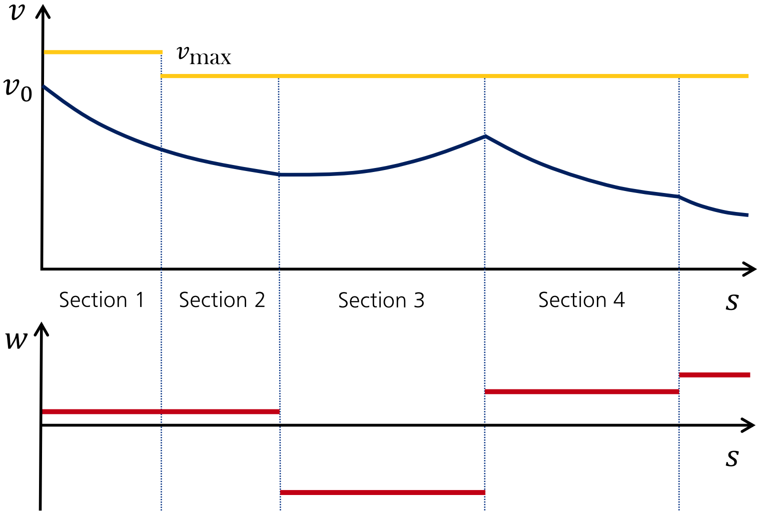


Fig. 5. Schematic representation of the new coasting advice algorithm, where the time and speed of the train are calculated at the points where either the speed limit or the track gradient change.

Table 1
Track geometry of the scenarios used for validating the algorithm. All slopes are in permil, start at the mentioned km and end at the next one, while the last flat track ends at the line end in all scenarios.

Case	Slope at km 0	Gradient at km 1.1	Gradient at km 2.4
$\beta = 0$ section	0	−0.681	0
Downhill section	0	−9.33	0
Uphill section	0	+9.33	0

Table 2
Basic parameters of a NS Sprinter train rolling stock type SLT-6 (Scheepmaker et al., 2020).

Property	Value
Train mass [t]	198
Rotating mass supplement [-]	1.08
Max. speed limit [km/h]	140
Train resistance [N] (v: [m/s])	$1375 + 37.48v + 6.75v^2$

lines the coasting curves calculated using the forward Euler method and in pale coloured solid lines the coasting curves extracted from the pseudospectral train trajectory optimizer. The blue markers represent the speeds and times calculated using the analytical formulas. The circle marker shows the points calculated in the $\beta = 0$ scenario, the x markers correspond to the downhill scenario, and the + markers to the uphill one. For each track section with constant gradient and constant applied effort, including the case when the train is coasting, the proposed algorithm only needs to calculate the speed and time at the end of the track section. In contrast, the other method require several intermediate calculations at each track section. We observe that the forward Euler method and the proposed analytical algorithm show a similar accuracy, while there are some small qualitative differences with respect to the coasting curves generated by the train trajectory optimizer, probably due to the sparser and unevenly distributed locations of the points where the trajectory is calculated by the pseudospectral method (see Fig. 6).

Last, Fig. 7 shows a comparison of the accuracy of these methods with respect to the analytical solutions. In the upper plots we show the error in speed, and in the lower ones, the cumulative time error. Again, we represent in yellow the scenario where $\beta = 0$, in turquoise the downhill, and in purple the uphill. The dashed line corresponds to the error of the forward Euler method with respect to the analytical formulas, and the continuous line, PROMO's error. Furthermore, we compared the analytical formulas with respect to MATLAB's ode45 function, which solves the equations of motion (1), (2) using an adaptive Runge–Kutta numerical method (Shampine and Reichelt, 1997). This is represented as a continuous blue line in all the plots. The error of ode45 is several orders of magnitude lower than the other two numerical methods and never exceeds 10^{-4} m/s. This validates the analytical formulas (14)–(16), (18)–(20).

When it comes to the other two methods, we observe jumps at the locations where the gradients change. This is due to the fact that both PROMO and the forward Euler method do not calculate the state of the train exactly at these locations. Another source of error is the approximation error in both methods, which makes the error vary linearly in each interval with constant gradient.

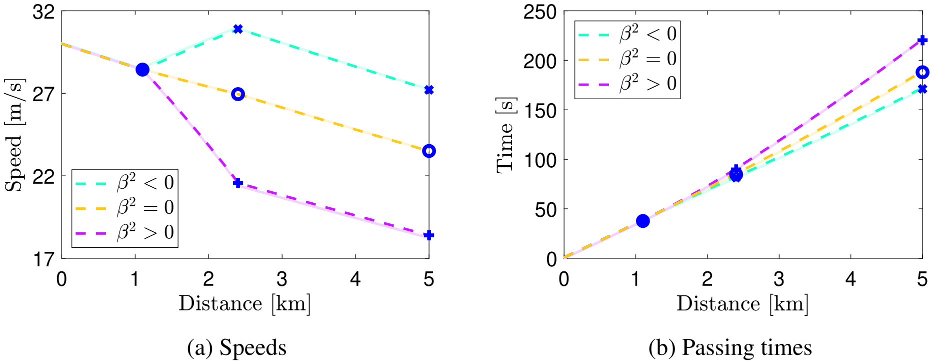


Fig. 6. Comparison of the coasting curves generated by the three methods.

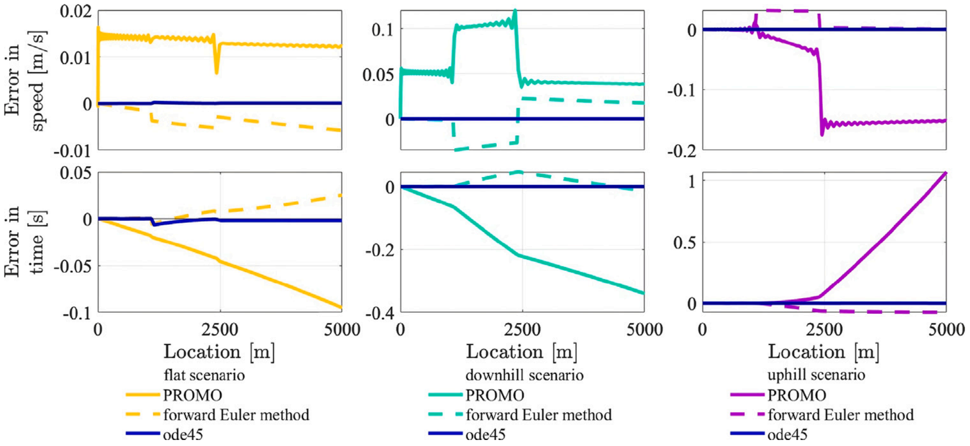


Fig. 7. Comparison of the error in speed (up) and time (down) at each location with respect to the analytical method for flat (left), downhill (middle) and uphill (right) slopes between 1100 and 2400 m. In each plot the dark blue line represents the error of MATLAB's ode45 function with respect to the analytical formulas, while the coloured dashed lines represent the error of the forward Euler method, and the coloured continuous line represent PROMO's error.

Particularly, PROMO shows oscillations in the speed since it is approximated as a finite sum of Lagrange polynomials. We also observe speeds up to 0.02 m/s higher than the initial speed while coasting in the first 40 m of the first flat track in the downhill scenario, and up to 0.007 m/s higher during the first 11 m in the scenario where $\beta = 0$. This is another numerical issue inherent to the pseudospectral discretization that constitutes PROMO's core and biases its error from the beginning of the coast curve. Still, the errors of these three methods are below 1 s and below 0.2 m/s in most cases after 5 km, which demonstrates that the accuracy of the three considered methods is adequate in practice for providing coasting advice.

Regarding the computation times, we used .NET framework's Stopwatch to determine which method is faster. We implemented both algorithms in the same programming language and run them on the same hardware. We compared the computation times of both algorithms in several scenarios, calculating coasting curves with diverse lengths and initial speeds. We run each scenario 30 times, observing that the computation times became stable after the third run in each scenario. Thus, we disregarded the first three warm-up runs and computed the average computation time. The proposed method was found to be up to 40 times faster than the forward Euler method. When aiming for the accuracy levels required in a practical implementation, the new method is 15 times faster. This relieves the computational burden on the application, allowing us to implement more complex driving advice routines in the near future. Moreover, the reduction of the computational power needed contributes to extending the battery life of the tablet in which this application is embedded. The computation time of the proposed method depends on the number of changes in gradients and speed limits, so filtering carefully this input data is important to keep the computation times low. Despite this, an accurate description of the track geometry is essential to guarantee the accuracy of the coasting advice. In turn, the forward Euler method's computation time is independent of the number of changes in gradients and speed limits. When it comes to implementing the proposed analytical method, it has been implemented in 250 lines of code, four times more than the forward Euler method. Still, the code is simple and the proposed method can be implemented easily.

Table 3
Constant tractive and brake power applied and track gradient resistance corresponding to the four different constant power formulas.

Scenario	Constant power [W]	Gradient [permil]	Initial speed [km/h]
(a) Three real simple roots	−800 000	−20	81
(b) A real double root	394 683.2	−10	60
(c) Two complex roots	1 755 000	20	60
(d) A triple root	−42.8	−6.7	80

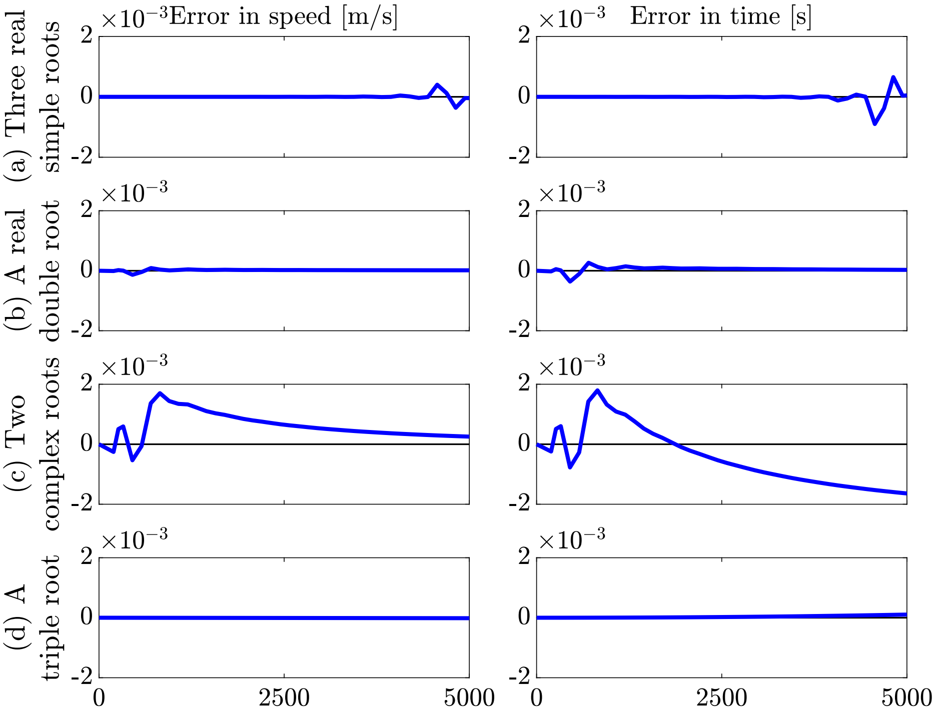


Fig. 8. Comparison of the error in speed (left) and time (right) at each location of MATLAB's ode45 function with respect to the analytical formulas in the following scenarios: (a) three real simple roots, (b) a real double root, (c) two complex roots and (d) a triple root.

4.2. Validation of the constant tractive and brake power formulas

We continue by exploring the formulas that are not included in the coasting advice algorithm, i.e., those that involve a constant tractive or brake power term (23)–(30). We simulate 5 km of coast curves corresponding to the four possible constant power scenarios introduced in Section 3.1.2: (a) three real simple roots (23) and (24), (b) a real double root (25) and (26), (c) two complex roots (27) and (28), (d) a triple root (29) and (30). Although these four scenarios may take place for a countless amount of combinations of power and gradient, we only provide one example to test the formulas. Ultimately, validating the formulas is our main scope rather than testing them in a wide variety of scenarios. Besides the basic parameters in Table 2, the gradient and the applied power parameters used in the constant power simulations are displayed in Table 3. Fig. 8 shows the error in speed (left) and time (right) at each location of MATLAB's ode45 function with respect to the analytical formulas. We observe that in these four cases the error is kept within 0.002 s or 0.002 m/s in the case of the speed, which validates the analytical formulas (23)–(30). Still, reaching this accuracy requires solving the mentioned formulas iteratively using a Golden-section Search algorithm. Despite having only a linear convergence rate, it requires only one evaluation of the analytical formulas per iteration and it does not require evaluating their derivative. This step introduces an error in the calculated speed and passing time that is inversely proportional to the number of iterations of the search algorithm and, consequently, to the computation time. The low errors shown in Fig. 8 allow to reduce the number of iterations while keeping an error that is acceptable in practical implementations. We also noticed that ode45 calculates 30 speed points in the first 10 s in all scenarios, which would make it unfit for our practical application due to the limited computation power available unless a lower bound on the integration step is implemented.

5. Conclusions

In this article we have presented an accurate and computationally efficient algorithm for calculating coasting curves. We utilize this algorithm at each second to calculate the expected arrival time if the train starts to coast, and this information is shown to the driver as a countdown to start braking. We utilize a description of the track geometry based on piecewise constant gradients. The analytical solutions of the train motion model are used to calculate the train speed and time at the locations where the gradients change. We show closed formulas for the mentioned analytical solutions, extending the formulas already published in the literature. We present the analytical solutions of the train motion model when the applied tractive and brake effort is limited by the maximum power of the engine, which were still not available in the literature. We have also presented an analysis of the possible solutions based on the theory of dynamical systems, which is essential in the algorithm. When considered as a dynamical system, its solutions present bifurcations, which are qualitative changes in the dynamics of the solutions. These changes are translated into changes in the closed formulas, and different formulas have to be used in different scenarios. For instance, only one formula can describe the dynamics of an accelerating train under constant effort and gradient, however, three formulas arise when considering a decelerating train under those conditions. Therefore the algorithm identifies which of the formulas has to be used, based on whether the train is accelerating or decelerating, and the applied effort and gradient resistance in the current track section. Furthermore, in the maximum power case, we factorized the acceleration term, extracting from it a cubic polynomial which analytical solutions depend on the geometry of its roots.

We have validated the coasting advice algorithm with respect to MATLAB's ode45 in a simulated case study, and compared its performance and accuracy with the previous method and a train trajectory optimizer based on a pseudospectral method. The proposed method is 15 times faster than the previous algorithm, allowing us in the near future to implement more complex advice and several extra features in the application. Moreover, it shows an improved accuracy over the other two methods. Although the formulas corresponding to a constant tractive or brake power are not implemented in the proposed algorithm, we have also validated their accuracy.

This new coasting advice generator has already being implemented in *Roltijd*, the DAS at NS in the Netherlands, and is contributing to the sustainable mobility of around 1.3 million passengers every day.

The applicability of the analytical solutions of the train motion model goes way beyond a coasting advice algorithm. Their simplicity and accuracy can be exploited in many different areas of the railway industry that require reproducing or predicting the motion of a train. We believe these formulas have the potential to make an impact in the coming years in the field of timetable design and evaluation, simulation, DAS and ATO.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

We thank the Dutch railway undertaking Nederlandse Spoorwegen (NS) for co-funding this research.

References

Becker, M., Schreckenberg, M., 2018. Analytical method for the precise and fast prediction of railway running times and its applications. IEEE Trans. Intell. Transp. Syst. 19 (11), 3560–3569.

Betts, J.T., 2010. Practical methods for optimal control and estimation using nonlinear programming. In: Advances in Design and Control. SIAM, Philadelphia, PA, USA.

Brünger, O., Dahlhaus, E., 2014. Running time estimation. In: Hansen, I.A., Pachl, P. (Eds.), Railway Timetabling and Operations: Analysis, Modelling, Optimisation, Simulation, Performance, Evaluation. Eurail Press, Hamburg, Germany.

Butcher, J.C., 2016. Numerical Methods for Ordinary Differential Equations. John Wiley & Sons.

Cunillera, A., Bešinović, N., Lentink, R., van Oort, N., Goverde, R.M.P., 2023. A literature review on train motion model calibration. IEEE Trans. Intell. Transp. Syst. 24 (4), 3660–3677.

Cunillera, A., Bešinović, N., van Oort, N., Goverde, R.M.P., 2022. Real-time train motion parameter estimation using an unscented Kalman filter. Transp. Res. C 143, 103794.

Davis, W.J., 1926. The tractive resistance of electric locomotives and cars. General Electr. Rev. 29, 685–707.

Fine, B., Rosenberger, G., 1997. The Fundamental Theorem of Algebra. Springer Science & Business Media.

Franke, R., Terwiesch, P., Meyer, M., 2000. An algorithm for the optimal control of the driving of trains. In: Proceedings of the 39th IEEE Conference on Decision and Control (Cat. No. 00CH37187), Vol. 3. IEEE, pp. 2123–2128.

Goverde, R.M.P., Scheepmaker, G.M., Wang, P., 2021. Pseudospectral optimal train control. European J. Oper. Res. 292 (1), 353–375.

Gradshteyn, I.S., Ryzhik, I.M., 2014. Table of Integrals, Series, and Products. Academic press.

Jaekel, B., Albrecht, T., 2014. Comparative analysis of algorithms and models for train running simulation. J. Rail Transp. Plan. Manag. 4 (1–2), 14–27.

Kiefer, J., 1953. Sequential minimax search for a maximum. Proc. Amer. Math. Soc. 4 (3), 502–506.

Körner, H., Dahlhaus, E., 2007. Precise Determination of Travel Time of Rail Vehicles. European Patent 20 050 025 843.

Luijt, R.S., van den Berge, M.P.F., Willeboordse, H.Y., Hoogenraad, J.H., 2016. 5 Years of dutch eco-driving: Managing behavioural change. Transp. Res. A 98, 46–63.

Rao, A.V., Benson, D., Darby, C., Mahon, B., Francolin, C., Patterson, M.A., Sanders, I., Huntington, G.T., 2011. User's manual for GPOPS version 4.x: A MATLAB software for solving multiple-phase optimal control problems using hp-adaptive pseudospectral methods. Manual.

Rao, A.V., Benson, D.A., Darby, C., Patterson, M.A., Francolin, C., Sanders, I., Huntington, G.T., 2010. Algorithm 902: GPOPS, a MATLAB software for solving multiple-phase optimal control problems using the Gauss pseudospectral method. ACM Trans. Math. Softw. 37 (2), 22:1–22:39.

Schank, T., 2011. A fast algorithm for computing the running-time of trains by infinitesimal calculus. In: The Fourth International Seminar on Railway Operations Modelling and Analysis, RailRome 2011.

Scheepmaker, G.M., Goverde, R.M.P., 2015. The interplay between energy-efficient train control and scheduled running time supplements. J. Rail Transp. Plan. Manag. 5 (4), 225–239.

Scheepmaker, G.M., Goverde, R.M.P., 2020. Energy-efficient train control using nonlinear bounded regenerative braking. Transp. Res. C 121, 102852.

Scheepmaker, G.M., Goverde, R.M.P., 2021. Multi-objective railway timetabling including energy-efficient train trajectory optimization. Eur. J. Transp. Infrastruct. Res. 21 (4), 1–42.

Scheepmaker, G.M., Goverde, R.M.P., Kroon, L.G., 2017. Review of energy-efficient train control and timetabling. European J. Oper. Res. 257 (2), 355–376.

Scheepmaker, G.M., Pudney, P.J., Albrecht, A.R., Goverde, R.M.P., Howlett, P.G., 2020b. Optimal running time supplement distribution in train schedules for energy-efficient train control. J. Rail Transp. Plan. Manag. 14, 100180.

Scheepmaker, G.M., Willeboordse, H.Y., Hoogenraad, J.H., Luijt, R.S., Goverde, R.M.P., 2020. Comparing train driving strategies on multiple key performance indicators. J. Rail Transp. Plan. Manag. 13, 100163.

Shampine, L.F., Reichelt, M.W., 1997. The MATLAB ODE suite. SIAM J. Sci. Comput. 1997, 1–22.

Strogatz, S.H., 2018. Nonlinear Dynamics and Chaos: With Applications To Physics, Biology, Chemistry, and Engineering. CRC Press.

Wang, P., Goverde, R.M.P., 2016. Multiple-phase train trajectory optimization with signalling and operational constraints. Transp. Res. C 69, 255–275.

Yang, X., Li, X., Ning, B., Tang, T., 2016. A survey on energy-efficient train operation for urban rail transit. IEEE Trans. Intell. Transp. Syst. 17 (1), 2–13.

Ye, H., Liu, R., 2017. Nonlinear programming methods based on closed-form expressions for optimal train control. Transp. Res. C 82, 102–123.