

## An algorithm for the calibration of running time calculation on the basis of GPS data

S. de Fabris, G. Longo & G. Medeossi

*Dept. of Civil and Environmental Engineering, University of Trieste, Italy*

### Abstract

Increasing the precision in timetable planning is a key success factor for all infrastructure managers, since it allows us to minimize delay propagation without reducing usable capacity. Since most running time calculation models are based on standard and deterministic parameters an imprecision is implicitly included, which has to be compensated by running time supplements.

At the same time, GPS or even more precise trackings are continuously stored in the event recorders of most European trains. Unfortunately, this large amount of data is normally stored but not used except for failure and maintenance management.

To consider real running time variability in running time calculation, an approach has been developed, which allows us to calibrate a performance factor for each motion phase.

Given the standard motion equation of a train, and a mesoscopic model of the line, the tool uses a simulated annealing optimisation algorithm to find the best regression between calculated and measured instant speed. To increase precision, the motion is divided into four phases: acceleration, braking at stops, braking for speed reductions/signals and cruising. By performing the procedure over a number of train runnings, a distribution of each performance parameter is obtained. Once the infrastructure model is defined and the trackings are imported, the procedure is completely automated.

The approach can be used in both stochastic simulation models and as a basis for advanced timetable planning tools, where stochastic instead of deterministic running times are used. The tool has been tested in the north-eastern part of Italy as input for both running time calculation and microscopic simulation.

*Keywords:* railway simulation, railway planning, GPS, train performance, calibration.



## 1 Introduction

Increasing the precision in timetable planning is a key success factor for all infrastructure managers, since it allows us to minimize delay propagation without reducing usable capacity. When setting up a timetable, it is necessary to estimate running times for different rolling stock using the current infrastructure. Conventional running time calculators solve the motion equation, which is based on a number of empirical parameters. Such parameters have been measured for years for different kinds of rolling stock [1]; therefore it is possible to calculate train speed profile in high detail. However, many influences on running times are not deterministic, such as human behaviour, weather conditions and even different trains of the same series could show different performances. To cope with this variability, recovery times are inserted, implicitly including an imprecision in the representation of train motion.

While a deterministic running time calculation is sufficient to plan timetables, which of course must be deterministic, a more detailed representation is required in micro-simulation and for ex-ante evaluate timetable robustness estimations, which aim at reproducing train behaviour with highest detail.

A *performance factor*, which introduces a stochastic element in the motion equation, has been proposed by some authors and inserted in proven simulation tools. This factor is multiplied by the tractive effort, the speed limit and the braking deceleration during acceleration, cruising and braking respectively.

To estimate the distributions of such parameters, an iterative approach was proposed by the Authors [3]. A software tool which allowed us to compare on-board collected data and running time calculation was developed from scratch and tested on a line in Northern Italy, demonstrating the benefits of calibrated motion equation in stochastic micro-simulation.

The results appeared very promising, although calibration of motion equation was performed manually by the user, graphically comparing simulated and real speed profiles at given points.

To overcome this weakness, and to allow a more precise calibration based on a higher number of records, a new software tool has been developed, which allows an automatic calibration of performance factors.

## 2 Approach and software tool

The proposed method includes a number of steps that are required to prepare data sets, filter the on-board collected data, calculate running time using deterministic motion equations and then calibrate them. All steps and data are included in a single tool to avoid exporting and importing data from in different software, reducing the number of steps performed by the user.

The software tool has been developed using Java programming language in order to use on different operating systems without compatibility problems, while the Netbeans IDE development environment was chosen for the simple windows drawing procedure and the good built-in debugging and profiling features.



The tool integrates a microscopic infrastructure model, rolling stock and timetable data to calculate running times. Real world data are filtered and then imported. A simulated annealing algorithm is used to calibrate the performance factor over a number of records. The distributions of performance factors are then presented and stored, to be used in the tool or exported to other simulation software supporting stochastic performance factors. The working flow of the approach is shown in Figure 1.

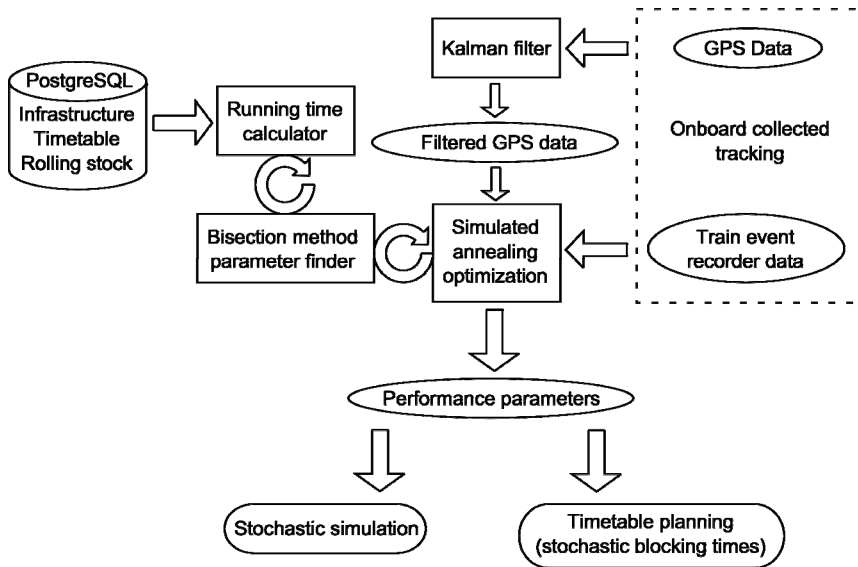


Figure 1: Block diagram of the working flow and software tool.

## 2.1 Microscopic infrastructure model

The infrastructure, rolling stock and timetables are stored in a PostgreSQL database. Compared to binary files, the database allows a simpler data model, higher exchangeability with other tools, and a smarter management of large amounts of data. Moreover, PostgreSQL represents a freeware, yet performing and reliable alternative to commercial DBMS. Infrastructure is represented according to the RailML standard [5], allowing a simple import and export from the most common simulation tools. The well-known and proven double-vertex graph [6] is used simplify the definition of itineraries.

A microscopic infrastructure model has been chosen rather than a mesoscopic one, which is sufficient for running time calculation, because not only line speed, but also signal aspects and the supervision curve of the protection systems are considered. To achieve the highest precision in the calibration process, the speed profile has to be calculated precisely for the entire train route: therefore the

itineraries within station and the position (and aspects) of each signal are required.

Also timetable data structure is entirely RailML compatible, while rolling stock is defined using all parameters required to solve the motion equation and consider the Italian ATP and ATC (SSC, SCMT, RSC, ETCS L2).

## 2.2 Data filtering and import

The calibration of motion equation can be performed on the basis of on-board collected GPS tracking or on the archives of digital train event recorders. Train event recorders store very accurate data based on odometers; endowed with a wider range of other data, such as the speed limit calculated by the protection system, the signal aspects the throttle and brake percentage use. These files do not require any filtering process, since tracking data are filtered in real time comparing two odometers and using balises or a discrete calibration of distances.

Significantly more complex to perform is the preparation of GPS data. Trackings are first post-processed using a proven algorithm based on the Kalman filter already in use to reconstruct the trajectories of race cars [8]. The filtered data show a sufficient precision, but do not contain any information on signal aspects, which is required in order to reconstruct the drivers' behaviour. To fill this gap, the recorded speed profile is compared to the planned one (which considers no route conflict), in which the train brakes only for speed reductions and at stops.

Once unplanned brakings are found, the planned profile is recalculated, considering a yellow aspect at the corresponding signal. On lines with discrete ATPs, the braking curve comprehending the approach speed and distance at next signal is computed, since the train is not allowed to re-accelerate before passing the corresponding main signal balise. On lines equipped with continuous ATP, where the train is allowed to accelerate as the man-machine interface shows the green aspect again, the instant when the train re-accelerates is obtained as local minimum of the real speed profile. The planned speed profile is then modified considering braking and then re-accelerating. The weakness of this method is represented by the impossibility to know exactly whether an unplanned braking is due to a "normal" traffic conflict, or to a failure, where the protection system activated the emergency braking.

GPS trackings, filtered and endowed with the corresponding signal aspects, and train event recordings are saved and used as basis for the calibration procedure.

## 2.3 Performance factors and motion phase

As stated in [3] a single performance factor does not allow a precise representation of the different motion phases. Separated parameters for acceleration, cruising and braking phases were suggested and used in the test case.

Knowing the signal aspects and the planned timetable it is possible to separate braking into three different phases: braking at a halt, for a line speed reduction



and for a restrictive signal aspect. This separation has been decided after a first analysis of real braking curves, which appeared significantly different depending on the motivation of braking. Braking at halts showed lower deceleration compared to braking for signals, and even lower values were recorded at speed limit changes. Moreover, especially when heavy trains brake at a speed reduction often run even lower speeds than allowed or show very variable deceleration to avoid this excessive braking.

As a result, compared to conventional running time calculation, where the braking distance is continuously calculated and stored, three values are computed.

## 2.4 Calibration algorithm

During calibration, the best fitting set of performance factors for a train ride is calculated. This computation is based on three assumptions:

- 1) The infrastructure model represents exactly the infrastructure used by the train, in particular concerning signal positions, relative distances and speed limits.
- 2) At the end of calibration, calculated running time correspond exactly to the measured one. In other words, the set of performance factor must lead to an exact calculation of the running time at the end of the journey.
- 3) The integration period in running time calculation must correspond to the tracking sampling period. This simplifies the calibration significantly, since to each recorded value corresponds to a calculated one, permitting a simple comparison in order to obtain an indicator of the goodness of the estimated parameters.

The method used to compare the two arrays is the simple mean squared error estimator. The software tool implements the algorithm (1)

$$MSE = \sum_{t=1}^N (v_{GPS}(t) - v_C(t))^2 \quad (1)$$

To compute a fixed length speed vector one dependent variable and four independent variables have to be considered. The software tool uses cruising performance as dependent variable for three reasons:

- 1) It has a small variation and relative high value (always higher than 90% in the test case)
- 2) Cruising running phase is the longest during a train run, therefore a minimum variation of its value has a great impact on total train running time
- 3) The value of cruising performance is inversely proportional to train running time.

Assuming 3), the value of the dependant variable value can be found using the bisection method.

## 2.5 Simulated annealing

In order to find the best performance parameters, an optimisation can be performed. It is necessary to find the minimum mean square error given four independent variables of a not linear function resolving the equation (2).

$$\begin{cases} \min \left\{ \sum_{t=1}^N [v_{GPS}(t) - v_c(t)]^2 \right\} \\ v_c = f(a, b, c, d, e) \\ size(v_{GPS}) = size(v_c) = N \end{cases} \quad (2)$$

In the equation  $a, b, c, d, e$  are the five performance parameters,  $v_c$  is the arrays of the calculated speeds and  $v_{GPS}$  is the array of measured speeds.

This appears as a nonlinear optimization problem: an adequate method to solve it has to be found; moreover, since it is difficult to determine a priori the properties of the target function a robust method must be used. A number of proven algorithms to perform this computation can be found in literature.

In the first tests, using only one parameter for breaking performance, the map represented in figure 2 has been obtained. It represents the value of target function (z-axis) with the variation of acceleration and braking factors. Obviously the value of cruising performance is not represented because it depends on the other two values. It's possible to notice the optimum value point for this train run.

The simulated annealing (SA) optimization method [9] is used in the software tool to find the best performance parameters that reduce the difference between computed and measured train speed. The algorithm begins finding a solution at a random point of the feasible region; after that, many steps are performed in order to find a better solution. It is possible to limit the maximum number of steps to be performed in order to limit computation time.

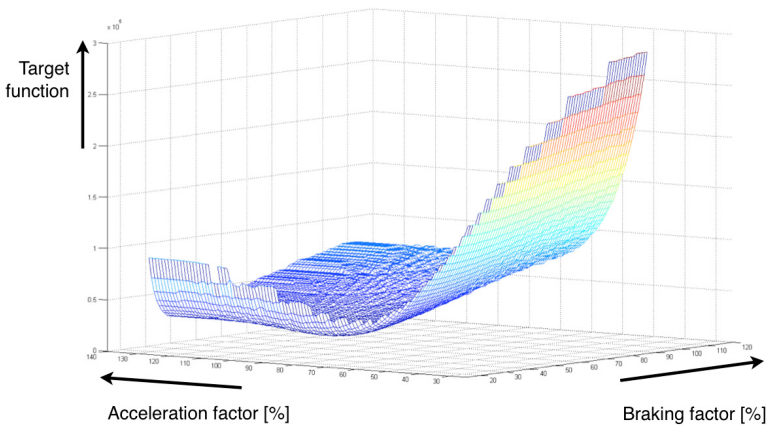


Figure 2: Target function (z-axis) as a function of acceleration and braking factors.

Each step of the SA algorithm replaces the current solution by a random “nearby” solution, chosen with a probability that depends on the difference between the corresponding function values and on a global parameter  $T$  (called the temperature), that is gradually decreased during the process. The dependency is such that the current solution changes almost randomly when  $T$  is large, but increasingly “downhill” as  $T$  goes to zero. The allowance for “uphill” moves saves the method from becoming stuck at local optima.

At the end of the optimization process both performance parameters are saved into a custom file format in order to perform aggregate data analysis. In the same files, the corresponding occupation steps are stored, to enable a precise representation of the real infrastructure usage.

## 2.6 Multithreading

Simulated annealing algorithm performs a number of experiments looking for value of target function in several points. Since each experiment is a train running time calculation with different performance parameters independent from each other, it's possible to run them concurrently using modern processors multithreading processing capacity. This permits to reduce total computation time, allowing us to use nearly 100% of the CPU power of modern multiprocessor workstations. The experiments to be computed are inserted in a queue, so each thread gets one experiment input data from the queue, computes the value of the target function, compares it with the best value ever found before, generates up to 8 new experiments and adds them in the queue. Synchronization between threads is needed only when accessing the queue to add and remove experiment input data and accessing best target function value. The synchronized code is less than 0.1% of total code lines of the optimization phase.

## 3 Test and applications

The tool has been tested in North-Eastern Italy, on the Trieste – Venice line. The double track, electrified line is about 130 km long and plays an important role in regional transport and as freight corridor between Eastern Europe and Italy. The line is endowed with SCMT, digital, discrete ATP corresponding to ETCS Level 1. About 100 train runs with the same rolling stock have been computed, obtaining the corresponding parameters sets. Each set represents a driving style, since it contains the way a driver handles in each motion phase.

Comparing the calibrated equation with 5 parameters with the conventional running time calculation, including 7% running time supplement, the difference between the two approaches has been pointed out, showing differences of more than 20% in the expected occupation steps.

In Figure 2, the speed limit (black), the planned speed profile (red) the real (green) and the calibrated (purple) ones are compared, clearly showing the differences among them. In particular, the goodness of the fitted equation compared to the planned one is remarkable.



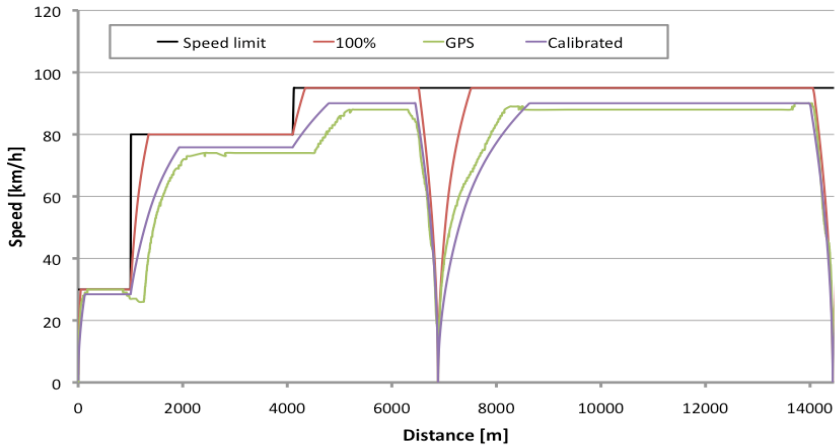


Figure 3: Speed limit, simulated speed profile with 100% performance, calibrated and real speed profiles.

### 3.1 Future applications

The obtained driving styles (calibration sets) over a statistically significant number of records can be used to compute the stochastic behaviour of a single train or a calibrated simulation of the interactions among trains. The second option would lead to a precise micro-simulation that could be either asynchronous or synchronous.

In the first option a representation of the blocking times over a number of realistic operational days are obtained, as tested on the Trieste-Venice line. The resulting blocking times can be stored and depicted in a timetable graph using transparencies to represent each run. As a result, a wider occupation staircase is obtained, which colour intensity is proportional to the probability to have the infrastructure used. While planning a timetable, the possibility to view the realistic infrastructure usage supports the definition of buffer times, which can be chosen on the basis of the real variability instead of empiric rules.

## 4 Conclusions and outlook

In this work, an approach for the accurate calibration of motion equation, which introduces a quantitative representation of real driving styles has been presented. The method has been implemented on a software tool, using standard representation of input data and taking advantage of powerful DBMS and multithreading to perform onerous computations in a competitive time.

The case study has shown a significant improvement in the stochastic micro simulation output, demonstrating the importance of a precise parameter calibration especially regarding train braking and acceleration. A further



improvement will come from the calibration of the motion equation parameters to fit the DIS data.

The approach appears very useful for the calibration of motion equation within micro-simulation tools, while a promising application is represented by the implementation of stochastic blocking time staircases instead of deterministic ones in timetable planning software.

## References

- [1] Wende, D. "Fahrndynamik des Schienenverkehrs". Wiesbaden, Teubner Verlag, 2003.
- [2] Hansen, I.A., Pacht, J., "Railway Timetable & Traffic" Hamburg, Eurailpress, 2008.
- [3] Medeossi, G. "Capacity and Reliability on railway networks, a simulative approach" University of Trieste, 2010-07-23.
- [4] de Fabris, S., G. Longo, et al. Automated analysis of train event recorder data to improve micro-simulation models. In: J. Allan, E. Arias, C. A. Brebbia et al., Computers in Railways XI, WIT Press, Southampton, 575-585, 2008.
- [5] Nash, A., Huerlimann, D., Schuette, J., Krauss, V.P., "RailML – A standard data interface for railroad applications", In: Allan, J., Hill, R.J., Brebbia, C.A., Sciutto, G., Sone, S. (eds.), Computers in Railways IX, WIT Press, Southampton, 45-54, 2004.
- [6] M. Montigel. Representation of Track Topologies with Double Vertex Graphs. In T.K.S. Murthy, F.E. Young, S. Lehman, W.R. Smith, editor, Computers in Railway, volume 2, Washington D.C., 1992. Computational Mechanics Publications.
- [7] Huerlimann, D., Nash, A., "Railway simulation using Opentrack", In: Allan, J., Hill, R.J., Brebbia, C.A., Sciutto, G., Sone, S. (eds.), Computers in Railways IX, WIT Press, Southampton, 45-54, 2004.
- [8] Robert Grower Brown, Patrick Y.C. Hwang, "Introduction to Random Signals and Applied Kalman Filtering" Second Edition, John Wiley & Sons, NavtechGPS, 1997.
- [9] Hillier, F. and Lieberman, G., Introduction to Operations Research (8<sup>th</sup> ed.), McGraw-Hill Science/Engineering/Math, 2005.

