DEVELOPMENT AND ANALISYS OF TRAIN BRAKE CURVE CALCULATION METHODS WITH COMPLEX SIMULATION

B. Vincze[a), G. Tarnai[b)

[a) *Budapest University of Technology and Economics, Dept. of Control and Transport and Automation*
Address: Bertalan L. u. 2., Budapest, Hungary H-1111
Phone: (+36-1) 463-1044 Fax (+36-1) 463-3087, e-mail:vincze.bela@t-online.hu web:web.t-online.hu/egzo
[b) *Budapest University of Technology and Economics, Dept. of Control and Transport Automation*
Address: Bertalan L. u. 2., Budapest, Hungary H-1111
Phone: (+36-1) 463-1990 Fax (+36-1) 463-3087, e-mail: tarnai@kaut.kka.bme.hu

Summary This paper describes an efficient method using simulation for developing and analyzing train brake curve calculation methods for the on-board computer of the ETCS system. An application example with actual measurements is also presented.

1. INTRODUCTION

1.1. Context of the study

The ETCS system is now spreading through Europe. As of now, there are ETCS level 1 and level 2 installations in European countries in operation. [1] Despite this fact, the ultimate aim of the ETCS system - the full interoperability - is still far off. The major reason for this comes from "vehicle" side: all interoperable railway vehicles will have to be equipped with an ETCS on-board computer (EVC), which can safely and efficiently co-operate with the trackside equipment in all countries and under circumstances. The study described in this paper focuses on the design, examination and validating process of a truly interoperable EVC with complex simulation.

1.2. Overview of the problem

The specifications for the trackside equipments are now clear, strict, well developed and documented [2]. However, this is not the case with the on-board equipment. The most important basic function of the EVC is the speed supervision, which forces the train to keep the maximum allowed speed limits. This requires a proper on-board brake curve calculation, to determine the current intervention speed limits in real time. At this point, the specifications still have a loophole [3], and significant issues are still open. The safety and the performance of the braking curves which are currently implemented in the existing EVCs are not fully defined by the current ETCS specification, and therefore have to be assessed for each railway network on a national level, which can be a very complicated and time-consuming process. Different calculation methods must be implemented in the EVC for different countries, vehicles, or even EVC manufacturers.

On the other hand, the safety of the brake curve calculation is the most important, but not the only one which has to be checked. Generally, it is easy to develop an algorithm which is absolutely safe (e.g. derived from worst-case analysis), but such an algorithm may be too strict, causing unnecessary brake interventions. To avoid this, extensive testing is required, which cannot be done with "real" test runs. As a result, a quick and efficient method is required in a simulated environment for developing and testing the basic algorithms implemented in the EVC.

2. METHODOLOGY

2.1. The tool

Nowadays, many railway companies and institutes in Europe use simulators for designing and testing various railway equipments, including station interlocking systems and train-on board computers [4].

During the research activity, a design and a real-time simulation tool have been developed, which is characterized by a fully modular architecture [4, 5, 6]. This architecture was designed to fulfil all needs related to train control system design and testing, in a three dimensional environment.

2.2. Modeling the environment

The virtual environment for the test runs included a test track, a test train with a locomotive and coaches (equipped with electro-magnetic and disc brakes or shoe brakes), and ETCS EVC on-board computer attached to the locomotive and two ETCS balises for transmitting the movement authority to the train. All object handler programs were written in C++ or Pascal.

The dynamics calculation is done by the simulation kernel [6], based on the parameters supplied by the vehicle objects. To achieve high accuracy, the dynamic model was calibrated using test measures.

2.3. Modeling the EVC

The EVC object was programmed in C++, in a highly modular style (Fig.1). In the following

sections, our study will focus on the "dynamic speed profile (DSP) calculation" module, which takes the actual static speed profile and train parameters as input, and calculates the intervention speeds for the output. For simplicity, we have removed the service brake intervention from the final EVC model.
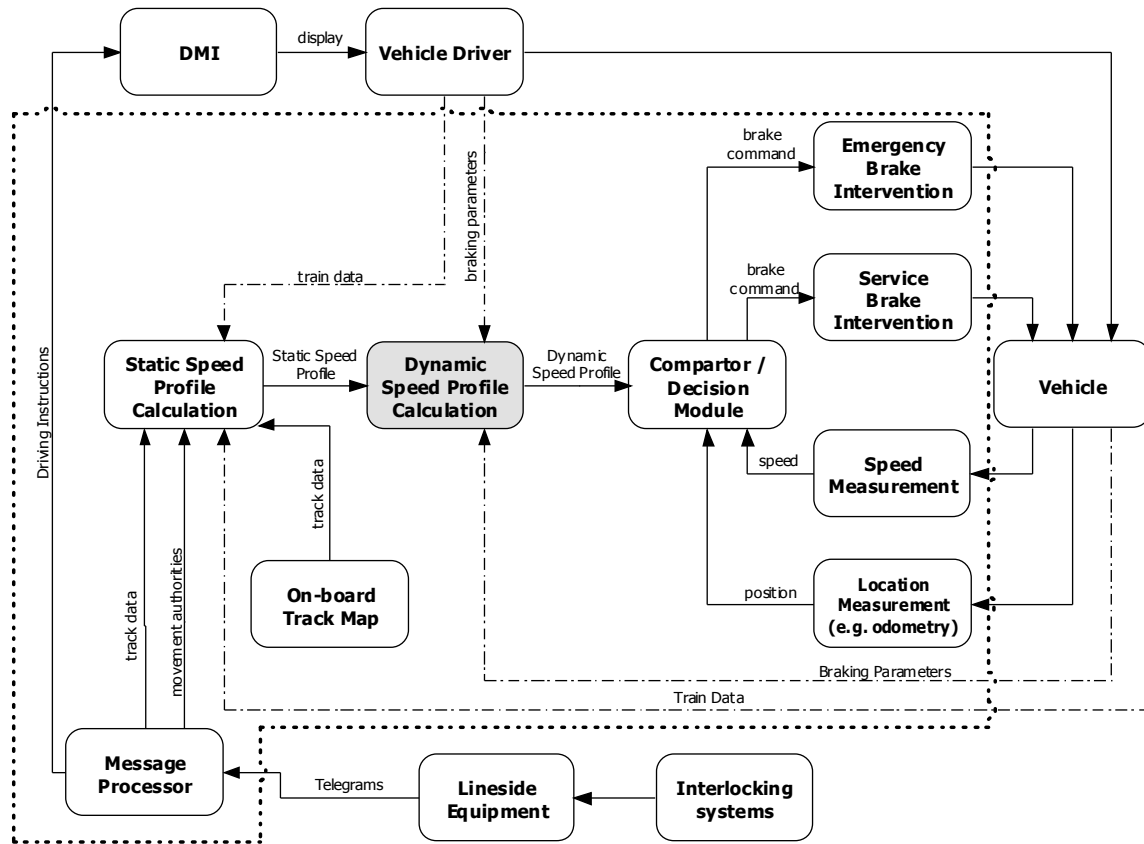

Fig. 1. Functional diagram of the simulated EVC unit (dotted frame) and the connected objects.

3. DEVELOPING AN ALGORITHM

3.1. Vehicle selection

For multiple units or passenger trains running in fixed consists (locomotive, coaches and driving trailer), brake curve calculation methods can be built up with very high accuracy by analyzing and processing actual test runs. Since train parameters does not change significantly, the brake distances will be affected only by the weather conditions [3].

The situation gets much more complicated, as we are trying to develop an algorithm for locomotives running with non-fixed train consists (usually freight trains or long-distance passenger trains), where different brake types can be used by each vehicle. In the following sections, this case will be discussed.

3.2. Input data

Again, taking a look at the current practice used by European railways for handling brake parameters, even more problems arise. Practically, apart from the parameters summarized in Table 1., there is no data available which could be used for brake curve calculations.

Input data	Unit	Deviation	Notes
Train length*	[m]	±5%	Affects brake delay
Train mass*	[t]	0 – (-20)%	
Braking efficiency*	[%]	0 – (+15)%	Has no physical meaning. Valid only when breaking to stop.
Train brake mode	G, P, R, Mg	--	Affects delay and performance.
Distance to target	[m]	<3%	When using odometry
Target speed	[km/h]	0%	--
Gradient	[‰]	--	--

Obvious, that the available input data does not allow the creation of a method, which is highly accurate under all possible circumstances. Therefore, we have to find an algorithm which provides an adequate level of safety, while it does not trigger undesired, disturbing interventions in most cases.

3.3. An example algorithm

As an example, let's start from the UIC 546. formula (1) for brake distance:

$$s = \frac{\varphi \cdot V^2}{1,09375 \cdot \dfrac{\lambda\%}{100} + 0,127 - 0,235 \cdot (\pm i) \cdot \varphi} \quad (1)$$

Where V is the speed of the train, $\lambda\%$ is the braking efficiency, i is the gradient and φ is a speed dependent constant. Assuming constant deceleration, the deceleration value (a_c) can be determined by rearranging the formula (1), taking into account that

$$a = \frac{v^2}{2 \cdot s} \quad (2)$$

and the mean value of φ (φ_c) for the speed range, the deceleration is

$$a_c = \frac{1}{2370} \cdot \frac{\lambda\%}{\varphi_c} + \frac{1}{200 \cdot \varphi_c} - \frac{1}{111} \cdot i \quad (3)$$

To check the accuracy this formula, test runs were made in the simulator. The results of the test run were used to refine the algorithm.

4. ANALSYS OF THE ALGORITHM

4.1. Model calibration

Before running tests on the simulator, the dynamic mode must be calibrated. For this purpose, various speed measurements were carried out with a GPS device (Fig.2.).
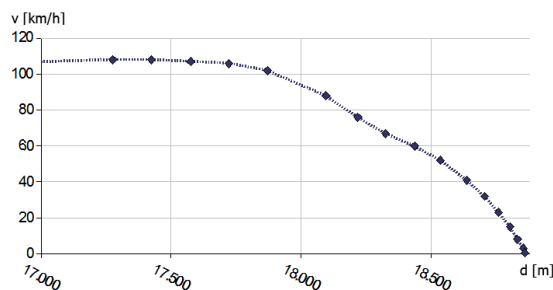

Fig. 2. Example of a Speed-Distance diagram, measured with GPS, for model calibration.

Based on the measurements, the braking model of the virtual locomotive and coaches can be fine-tuned, with a separate software module. The braking model has more than 5 parameters, including friction coefficients for each brake type of the test train and reacting force constants. These parameters are optimized by checking all possible values such that the difference between the GPS measured and simulated brake curves become minimal (Fig. 3.).

Since the brake curve cannot be expressed in an explicit form, the optimization is very time consuming. This due to the fact that each optimization step requires the simulation of a full braking process with a virtual test train. In case more parameters are implemented in the model, a faster optimizing algorithm will be required (e.g. application of generic algorithms should be considered).
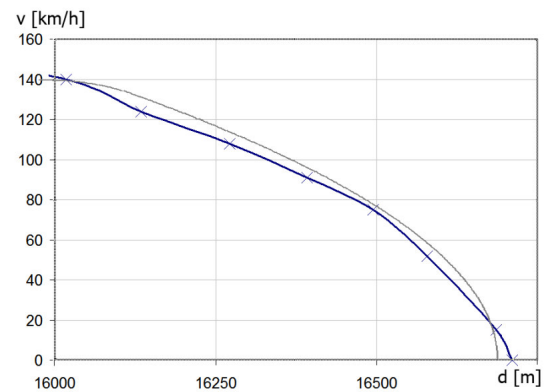

Fig. 3. Comparison of a real brake curve measured with GPS (dark line), and a simulated brake curve (light line) after calibration.

4.2. Testing and refining the algorithm by using the simulator

As soon as the model is calibrated, test runs can be made in the simulator to identify and counter weak points of the EVC model. During test runs, the simulator logs the selected parameters (position, speed, interventions, state codes etc.) into a text file, for further processing.

In the framework of this paper we can't go through each test, just present some of the interesting results. For testing the algorithm presented in the previous section, various scenarios were created, with different train compositions. Fig. 4. shows a speed-distance diagram of a test run with a freight train equipped with shoe brakes. The train had to brake from 100 km/h to standstill with automatic intervention.

According to the expectations, the deceleration of a train equipped with shoe brakes can't be treated constant in the entire speed range. However, this did not cause any problems since the train had to brake to standstill, and the position of the emergency brake intervention was derived from the

average deceleration, calculated with (3). The diagram clearly shows that there is some margin of safety in the calculated brake distance.
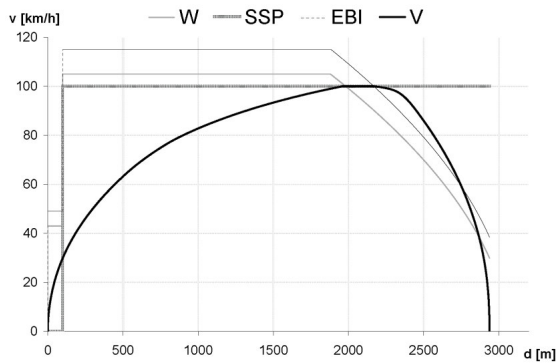

Fig.4. Results of a test run in the simulator. W is warning speed, SSP is the static speed profile, EBI is the emergency brake intervention speed and V is the actual speed of the train.

Another test run was carried out with the same train consist, with a target speed of 80 km/h (Fig.5.). In this case, the original algorithm failed, as it could not force the train to slow down to 80 km/h before the target. The reason for this lies in the input parameters: the formula (3) is based on the braking efficiency, which is valid only when the train brakes to stop.
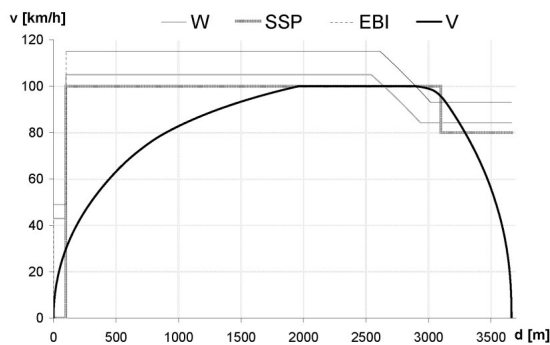

Fig. 5. Results of a test run with a freight train braking from 100 km/h to 80 km/h.

Further tests showed that the error is much smaller with electro-magnetic or disc brakes. This suggests that, there are several ways to correct the algorithm, e.g. by introducing some fixed or variable delay depending on the value of braking efficiency.

Summing up, the design process of algorithm for the EVC is an iteration, where each loop results in a more and more refined system. Without complex simulation, the loops would be much more expensive and much longer in time.

5. CONCLUSION

The presented simple example proved that a complex computer simulation [5] with calibrated dynamic model can be efficiently used for developing or testing basic functions and algorithms of an ETCS on-board computer. The most important objective of further development of the tool is to improve the calibration of the dynamic model, e.g. by the means of genetic algorithm.

REFERENCES

[1] SBB-CFF-FFS: *ETCS – Aktuelle Projekte in Europa (http://mct.sbb.ch/mct/en/etcs-projekte.pdf) (2005).*

[2] ERTMS/ETCS consortium: *ERTMS/ETCS – Class 1 System Requirements Specification. (2005).*

[3] ERTMS/ETCS consortium: *Interoperability-related consolidation on TSI annex A document, Issue 1-0-0 (2005)*

[4] Vincze, B., Tarnai, G.: *Functional and safety analysis of train control systems with simulation (in Hungarian) Vezetékek Világa 2005/4.*

[5] Vincze, B., Tarnai, G.: *Examination of level crossings on ETCS equipped lines with complex simulation. AEEE, Žilina, 2004*

[6] Vincze, B., Tarnai, G.: *Simulating train control systems in three dimensional environment, FORMS 2005 Conference Proceedings*