Model-based Systems Engineering with Matlab/Simulink in the Railway Sector

Alexander Nitsch
Universität Rostock
Alexander.Nitsch@uni-rostock.de

Benjamin Beichler
Universität Rostock
Benjamin.Beichler@uni-rostock.de

Frank Golatowski
Universität Rostock
Frank.Golatowski@uni-rostock.de

Christian Haubelt
Universität Rostock
Christian.Haubelt@uni-rostock.de

Abstract

Model-based systems engineering is widely used in the automotive and avionics domain but less in the railway domain. This paper shows that Matlab/Simulink can be used to develop safety-critical cyber-physical systems for railway applications. To this end, an executable model has been implemented which allows for train movement simulation such as automatic emergency braking.

Keywords: model-based, cyber-physical system, Matlab/Simulink, safety-critical, executable model

Acknowledgement: This work was funded by the German Federal Ministry of Education and Research (Grant No. 01IS12021) in the context of the ITEA2 project openETCS.

1. Introduction

Model-based system engineering has proven to be a well suited methodology to develop embedded systems and especially safety-critical cyber-physical systems. Model-based approaches are widely used in the automotive and avionics domain but still uncommon in the railway sector. The increasing complexity of software in locomotive on-board units renders software development with traditional methods nearly impossible. We propose model-based engineering techniques as a means to ease this process.

One critically safety-relevant software module is the control system of the train. Protection systems like this are getting developed since the very beginning of railway operation. Consequently, trains run by different countries use mostly non-interoperable train control systems. Especially in the converging European Union this leads to a problem: all trains that need to cross borders also need to be equipped with several expensive train control systems.

The European Train Control System (ETCS), which was designed in the early 1990s, is the designated solution to overcome this problem within the European borders. ETCS includes a set of modern concepts for train control to achieve high speed and high utilization of the rail. Besides this, ETCS aims to be flexible to address all requirements of the national railway operators. The resulting ETCS standard became rather complex and, as the standard currently is only available as natural, non-formal language document, is very difficult to implement. Consequences of this are high development costs and incompatible implementations of different vendors caused by ambiguities of the specification.

In this environment, the openETCS project was created with the goal of an open source implementation of the on-board unit software. To achieve this, model-based systems engineering methods are employed. In this paper we present our efforts to analyze and develop the Speed and Distance Monitoring, which is part of the ETCS standard. The chosen modelling tool is Simulink, which is widely used in industrial applications especially in the automotive sector. The result of this paper is an executable model of the braking curve calculation, which is part of the Speed and Distance Monitoring. Moreover, the developed model offers a simulation framework for the train movement characteristics.

This paper is structured as follows. After the motivation of the topic in section 1, section 2 gives an overview of railway related publications. Section 3 introduces the basics of the European Train Control System and presents the Speed and Distance Monitoring as a subsystem of ETCS. Braking curves are used to predict the movement behavior of a train especially in case of emergency. The principle of the Emergency Brake Deceleration (EBD) curve and its calculation in the form of an executable model are described in section 4. As a case study, the model of the EBD calculation is used in section 5 to simulate the braking behavior of a moving train. Section 6 summarizes the acquired knowledge and briefly discusses future work.

2. Related Work

Since the first release of the ETCS standard several publications examined different aspects of the ETCS specification. Many of them deal with real-time properties and reliability of the communication link between train and track-side equipment. In [zHHS13, JzHS98, ZH05, HJU05] Petri net extensions are used to investigate the functional properties and stochastic guarantees of the communication. Modeling and calculation of speed and distance monitoring of ETCS were covered in [BT11, Fri10]. These works focus on the functional properties of the computation and use of an application-specific modeling methodology.

Other publications in the ETCS context focus on formalization and safety analysis. The authors in [CPD$^+$14] show in three case studies how formal languages can ease the verification process of safety-critical systems. They show how the SPARK language and its toolset can be integrated into the existing development process to decrease the effort of system certification in the railway domain. In [CA14] a formal model in form of a Time Extended Finite State Machine is developed. This model is used to represent safety properties of the ETCS requirements and allows to derive tests for checking these properties. Within the scope of model-based systems engineering, SysML is a widely used language for graphical system description [OMG12]. A large number of publications use SysML to describe, test and verify architectural and functional system properties in context of ETCS and generally railway, e.g. [BFM$^+$11, MFM$^+$14, BHH$^+$14].However, SysML cannot produce executable code. Simulink is another graphical programming environment for model-based design and in contrast to SysML, it enables simulation of dynamic systems and provides automatic code generation for the integration into other applications [Mat14]. This paper focuses on Simulink to develop a ETCS related model which is executable and therefore usable for dynamic analysis tasks such as train movement.

To our best knowledge, no comparable solution exists for train movement analysis and simulation with respect to conformity of the ETCS standard.

3. ETCS - Speed and Distance Monitoring

One of the main tasks of ETCS is to supervise the speed and position of trains to ensure that the train stays in the permitted speed ranges. Because of the low friction between steel wheels and rail and the relative high mass of the train, the braking distance is very large compared with e.g. automobiles. As a consequence, a human train driver is not even able to perceive the brake distance on the track including railway signals or other trains.

An established approach in train control systems are track side equipment like multiple combined signals and mutual exclusive track usage of trains. The size of the track segments significantly effects the utilization and possible throughput and therefore the profitability of a track. Since the signal equipment is fixed at the track side, a customization for different rolling stock is effectively impossible. This becomes a serious problem if trains with significant different maximum speed and braking abilities are used on a track.

To prevent a human failure of the perception of such safety-critical information, all modern train control systems must have an automatic intervention possibility for dangerous situations. More sophisticated train control systems like ETCS make usage of customized signaling with displays within the train cab. This so called "cab signalling" helps to customize the speed and distance limits for every train. The challenge of such a calculation on the onboard unit of the train control system is to ensure the safe operation of the train. This includes the functional safety and the time critical aspects of this calculation speed and distance limits.

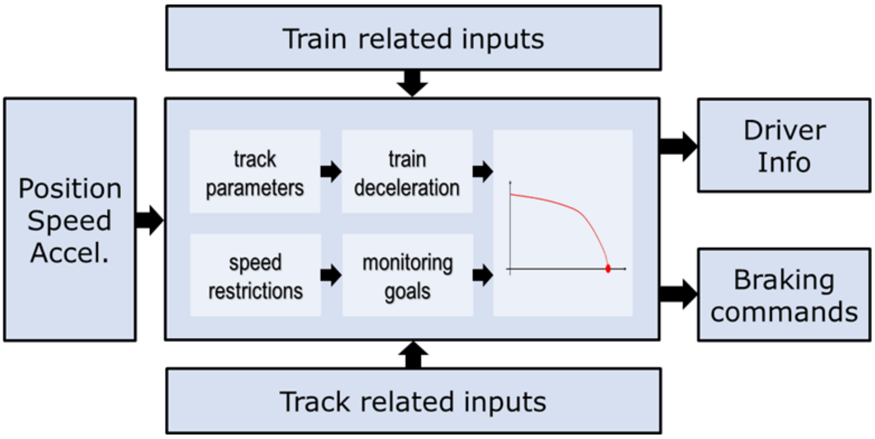


Figure 1: Overview of the ETCS Speed and Distance Monitoring

An overview of the SaDM is shown in Figure 1. The tasks of the Speed and Distance Monitoring (SaDM) are defined within the System Requirements Specification [UNI12] within chapter 3.13. The main results of the SaDM are information for the driver, e.g. the current permitted speed, monitoring targets and for critical situations the SaDM issues automatic braking commands.

In order to determine this information, SaDM needs several inputs such as the dynamic values of current position, speed and acceleration of the train. Moreover a certain number of other train and track related inputs are needed, which have a lower dynamic as position or speed. The most important train related inputs are the braking abilities of a train.

Modern trains have multiple sets of brakes, which have different operating principles, and are used in several combinations according to various conditions. According to this the applicable braking deceleration in a dangerous situation needs to be defined for all possible combinations.

Other important characteristics such as curve tilt abilities, maximum train speed or the train length are also needed to be considered in order to calculate the train dependent impact on the speed and distance limits. All train related inputs are combined to a function called A_{safe}, that assigns a braking acceleration to the two independent parameters of speed and distance. All described inputs are piece-wise constant functions or so called step function of speed or position, so that the $A_{safe}(v, d)$ have also the characteristics of a step function in a two dimensional manner.

Beside the train characteristics, the track related information is the other important input data. A train equipped with ETCS receives information about the track properties while moving on it. This includes a profile of the track slopes and a set of static speed restrictions, which are caused by the shape of a track. Furthermore dynamic speed restrictions (e.g. in areas which are under maintenance) are transmitted to the train. This collection of location based data defined speed restrictions are compressed to a single data structure called Most Restrictive Speed Profile (MRSP), which contains a single allowed speed for every position on the track ahead. From this profile the particular targets for the supervision are derived by getting all points with a decreasing allowed speed. An additional special target is derived from the limited permission of a train to move on the track. This End of Authority is derived from the Movement Authority, which needed to be transmitted to the train while operation.

Every of the described supervision targets are forwarded to the calculation of the target specific braking curve. To predict the behavior of the train in an emergency case the Emergency Brake Deceleration (EBD) curve is one of the most important calculations, which is therefore in the focus of following sections.

4. Braking Curve Calculation

In this section, we detail the braking curve calculation, which is part of the Speed and Distance Monitoring and presented as Simulink model. To best of our knowledge, this is first executable model derived from ETCS SRS.

4.1. EBD Calculation

The Emergence Brake Deceleration curve (EBD) is called the parachute of ETCS because this curve represents the braking behavior in case of emergency. In case of emergency and from a certain speed the system has to use all available brakes to reach zero speed at a concrete location. In addition, there exist several constraints, e.g. there is a slippery track, which leads to a reduced brake performance, or the system is not able to use all brakes but only a specific combination, which also results in a reduced brake performance, the system has to calculate the position of brake initiation to reach the target position under any circumstances. The influence of the brake performance on the braking distance is shown in Figure 2.

By a lower brake performance (1) the train will not stop at the desired position on track, that means the system has to brake earlier to stop at the desired position. In contrast to that, a higher brake performance will earlier stop the train (2) or the initial braking can be done later. The braking distance depends on a brake deceleration value (3).

If the stop location and the brake performance on each section of the track are known, the latest possibility of braking can be calculated to stop at the desired position. Hence, there is a need of a

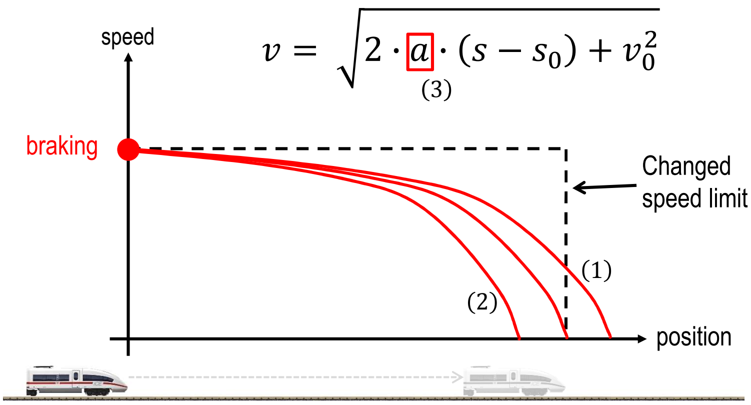


Figure 2: Brake performance and its influence on the brake distance

backward calculation algorithm, which starts its calculation from the target location and calculates backwards to the front end of the train Figure 3.

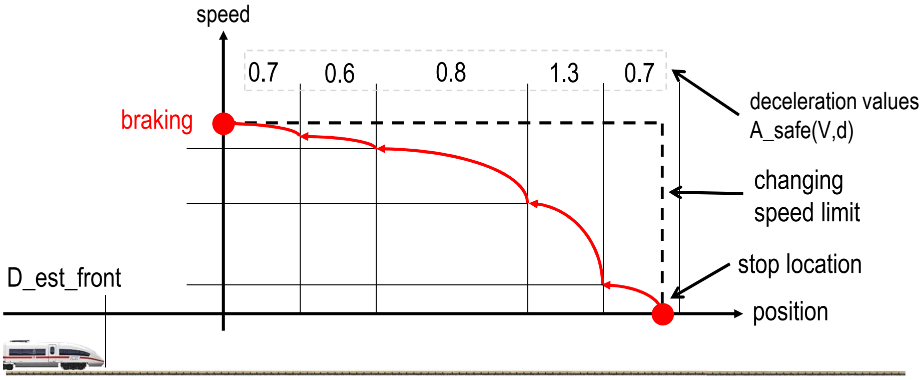


Figure 3: Backward calculation of the brake initiation depending on brake performance

The result of the algorithm is the maximum speed of the train on a specific position on track. By exceeding this speed limit the train will fail to stop at the desired location. This information is known as EBD. After knowing the maximum speed in comparison to the actual speed, the ETCS onboard computer can intervene and brake automatically.

4.2. Simulink Model for the EBD Calculation

For the calculation of the EBD curve a Simulink model has been implemented. The algorithm for the calculation process can be seen in Figure 4. For a given target distance the algorithm calculates the maximum allowed speed of the train to stop at that target location. The Simulink model use numerous inputs, provided by a balise, which is integrated in the rail bed in front of a possible target. The inputs are: the distance to the target location (`d_target`), the desired speed at the target location (`V_target`) and the estimated front end (`d_est_front`) of the train (distance covered yet).
Another input is a two dimensional step function organized as an array named `A_safe(V,d)` containing information about deceleration values (`curAcc`) of the train depending on the track position (`curDis`) and the speed of the train (`curVel`). Therefore a specific deceleration value depending on both, a particular speed and position category (`dis_cat`, `vel_cat`) is returned.

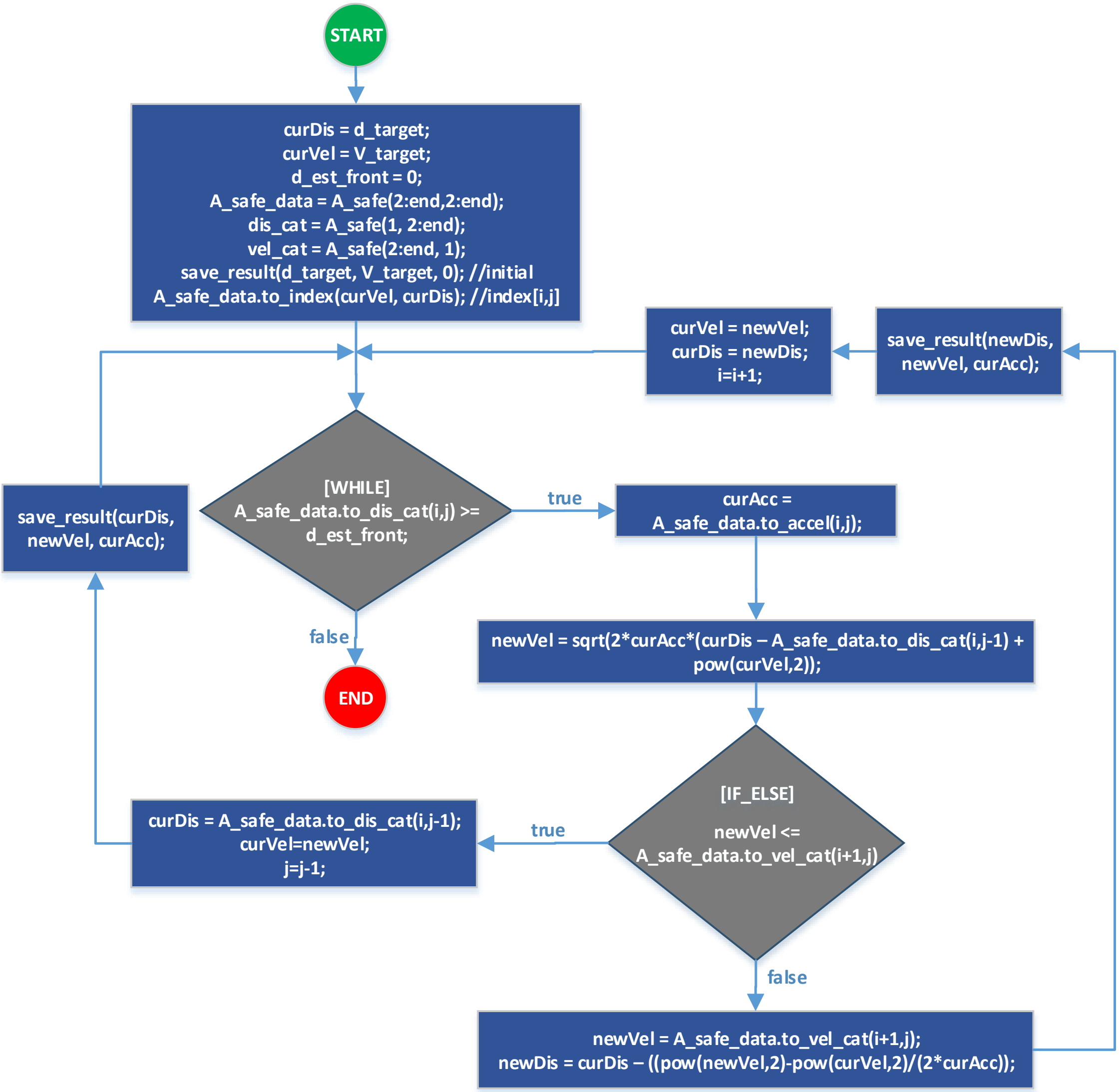
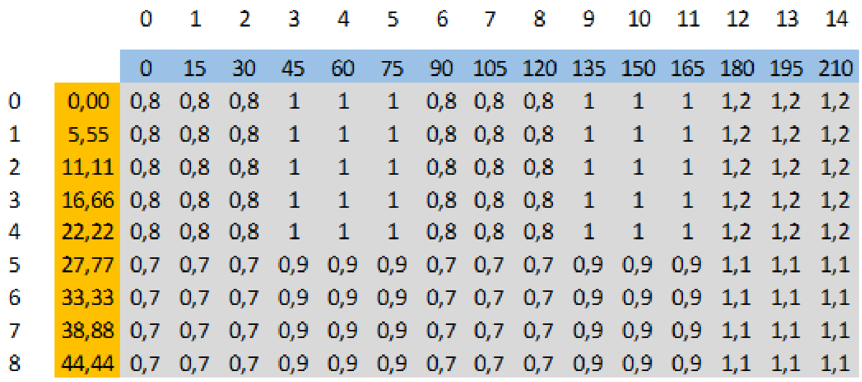

START
curDis = d_target;
curVel = V_target;
d_est_front = 0;
A_safe_data = A_safe(2:end,2:end);
dis_cat = A_safe(1, 2:end);
vel_cat = A_safe(2:end, 1);
save_result(d_target, V_target, 0); //initial
A_safe_data.to_index(curVel, curDis); //index[i,j]
curVel = newVel;
curDis = newDis;
i=i+1;
save_result(newDis, newVel, curAcc);
save_result(curDis, newVel, curAcc);
[WHILE]
A_safe_data.to_dis_cat(i,j) >=
d_est_front;
true
curAcc =
A_safe_data.to_accel(i,j);
false
END
newVel = sqrt(2*curAcc*(curDis – A_safe_data.to_dis_cat(i,j-1) + pow(curVel,2));
[IF_ELSE]
newVel <=
A_safe_data.to_vel_cat(i+1,j)
true
curDis = A_safe_data.to_dis_cat(i,j-1);
curVel=newVel;
j=j-1;
false
newVel = A_safe_data.to_vel_cat(i+1,j);
newDis = curDis – ((pow(newVel,2)-pow(curVel,2)/(2*curAcc));

Figure 4: Algorithm of the EBD curve calculation

Based on the given inputs the Simulink model calculates iteratively the maximum allowed speed of the train regarding to a specific position on track, which must not been exceeded by the train to stop at the desired target location.

The structure of an example `A_safe`-data-set is depicted in Figure 5. This function contains a matrix (gray) which represents the deceleration values of the train in m/s^2, for a distance category in m (blue) and a speed category vector m/s (orange). Additionally, the left axes represents the speed category index (`i`) and the upper axes represents the distance category index (`j`).

Now the EBD calculation starts as follows. The deceleration value in the target region has to be determined at first. Therefore within a Simulink look-up-table the given target distance in the distance vector is approximated (`A_safe_data.to_index(curVel,curDis)`) and also the relevant index is returned. Because zero speed at the target is considered, the first deceleration value can be derived at the speed index zero and the distance category index selection which depends on the target distance. Thereafter, the the maximum allowed speed is calculated by the formula which is given in Figure 2. The resulting speed (`newVel`) is calculated until the next lower distance



		0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
		0	15	30	45	60	75	90	105	120	135	150	165	180	195	210
0	0,00	0,8	0,8	0,8	1	1	1	0,8	0,8	0,8	1	1	1	1,2	1,2	1,2
1	5,55	0,8	0,8	0,8	1	1	1	0,8	0,8	0,8	1	1	1	1,2	1,2	1,2
2	11,11	0,8	0,8	0,8	1	1	1	0,8	0,8	0,8	1	1	1	1,2	1,2	1,2
3	16,66	0,8	0,8	0,8	1	1	1	0,8	0,8	0,8	1	1	1	1,2	1,2	1,2
4	22,22	0,8	0,8	0,8	1	1	1	0,8	0,8	0,8	1	1	1	1,2	1,2	1,2
5	27,77	0,7	0,7	0,7	0,9	0,9	0,9	0,7	0,7	0,7	0,9	0,9	0,9	1,1	1,1	1,1
6	33,33	0,7	0,7	0,7	0,9	0,9	0,9	0,7	0,7	0,7	0,9	0,9	0,9	1,1	1,1	1,1
7	38,88	0,7	0,7	0,7	0,9	0,9	0,9	0,7	0,7	0,7	0,9	0,9	0,9	1,1	1,1	1,1
8	44,44	0,7	0,7	0,7	0,9	0,9	0,9	0,7	0,7	0,7	0,9	0,9	0,9	1,1	1,1	1,1

Figure 5: Example `A_safe`-data

category index because at this position may the deceleration value change and consequently the brake performance also change.

Due to the two dimensional characteristic of the `A_safe`-data the brake performance also depends on speed. The algorithm has to check whether the new calculated maximum speed matches into the related speed category, which has been realized through an Simulink If-Condition-Block. There are two possible scenarios. If the calculated maximum speed is lower or equal to next speed stage, the speed category index is incremented by one, the distance category index will be decremented and its value represents the new distance as input to the next iteration of the algorithm. The else-case is triggered if the calculated maximum speed is greater than the next speed stage. It must be assumed that there is change in deceleration value on that speed level. In consequence of that, the concrete position regarding to the actual speed level have to be calculated by transposing the formula which is given in Figure 2. The results are saved in a table and the backward calculation algorithm processes as long as the actual train position is reach.

5. Case Study

In this section, the Simulink model of the EBD calculation is used to simulate the braking behavior of a moving train in case of a speed limit change to zero speed Figure 6. The simulation is interactive, therefore the user is able to manipulate the train movement by setting the actual acceleration value of the train. A positive value will accelerate the train to a desired speed, zero acceleration leads to a constant speed and a negative acceleration is used to decelerate the train until zero speed is reached and the train stands still. The outputs of the train movement block are the actual train speed and the actual train position on the track.

At the start of the simulation, the braking curve of the whole track is calculated. This curve represents the upper speed limit of the train to really stop at the desired location. By reaching the EBD curve the train will automatically brake by using the deceleration of the `A_safe`-data, which depends on the actual speed and track position of the train. The output of the EBD Calculator is a matrix which consists of 4 column vectors: start and end position of a deceleration section, its corresponding deceleration value and the calculated maximum speed at the end of each section.

The EBD Sampler calculates, corresponding to actual train position, a maximum speed value by using the formula which is presented in Figure 2. The maximum speed regarding to a specific position is given as an output to the Speed Limiter. The Speed Limiter compares the actual speed of the train with the braking curve speed limit and feeds back a boolean value to the train movement.

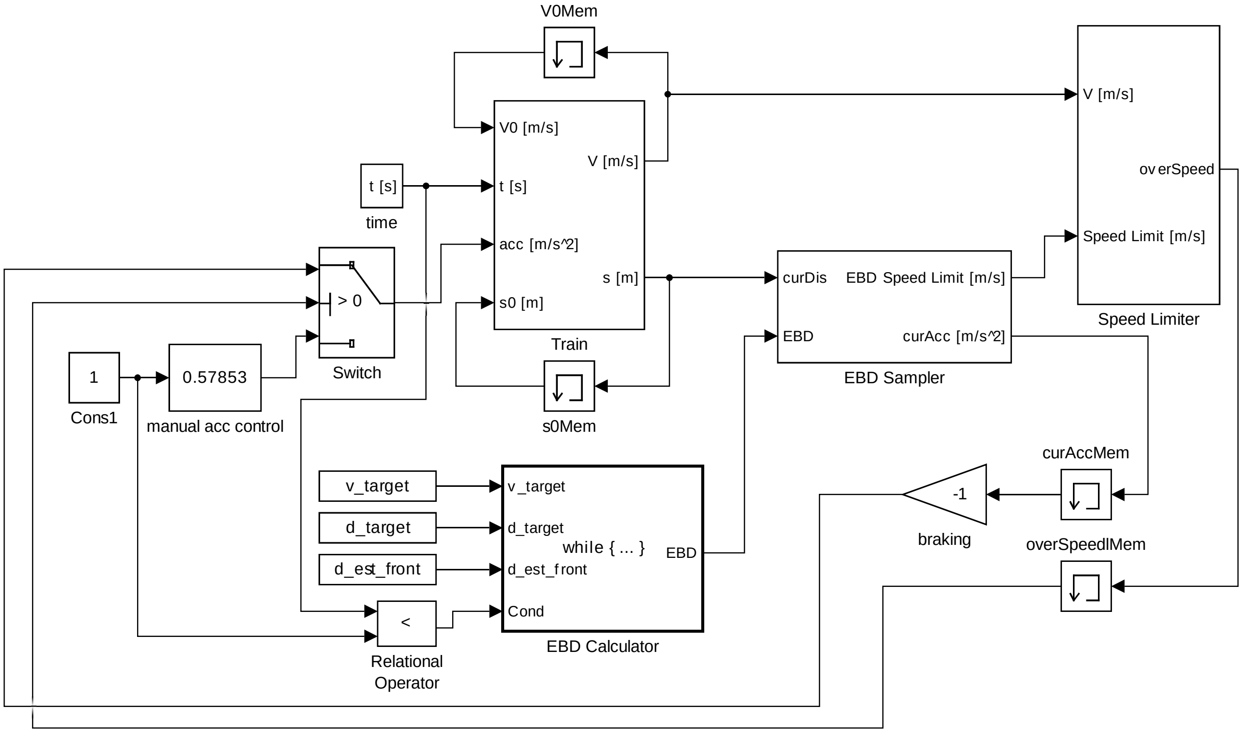


Figure 6: Train movement simulator block diagram

In case of equality of both values, the boolean is set to one. The acceleration input switches from manual user acceleration control to automatic braking. Consequently the train slows down and stops at the target location. The result of an example test case is depicted in Figure 7. Due to several impacts like the brake built up time and possible other tractional acceleration a certain safety margin is subtracted from the EBD. Therefore the both curves are not congruent while the automatic intervention.

6. Conclusion and Future Work

This paper has shown that model-based system engineering is suitable to develop complex safety-critical cyber-physical systems of the railway domain. We have proven that the desired functionality can be realized with Matlab/Simulink. Simulink was used to implement an executable model to calculate the Emergency Brake Deceleration curve, which is an important outcome of the Speed and Distance Monitoring. Additionally a simulation framework for the analysis of train movement was realized. With that result, we are now able to test different scenarios for automatic braking. To the best of our knowledge, this is the first executable model of the Speed and Distance Monitoring of the new European Train Control System. This model will be used in future experiments to verify and design parts of ETCS trains onboard unit on model-based approaches.

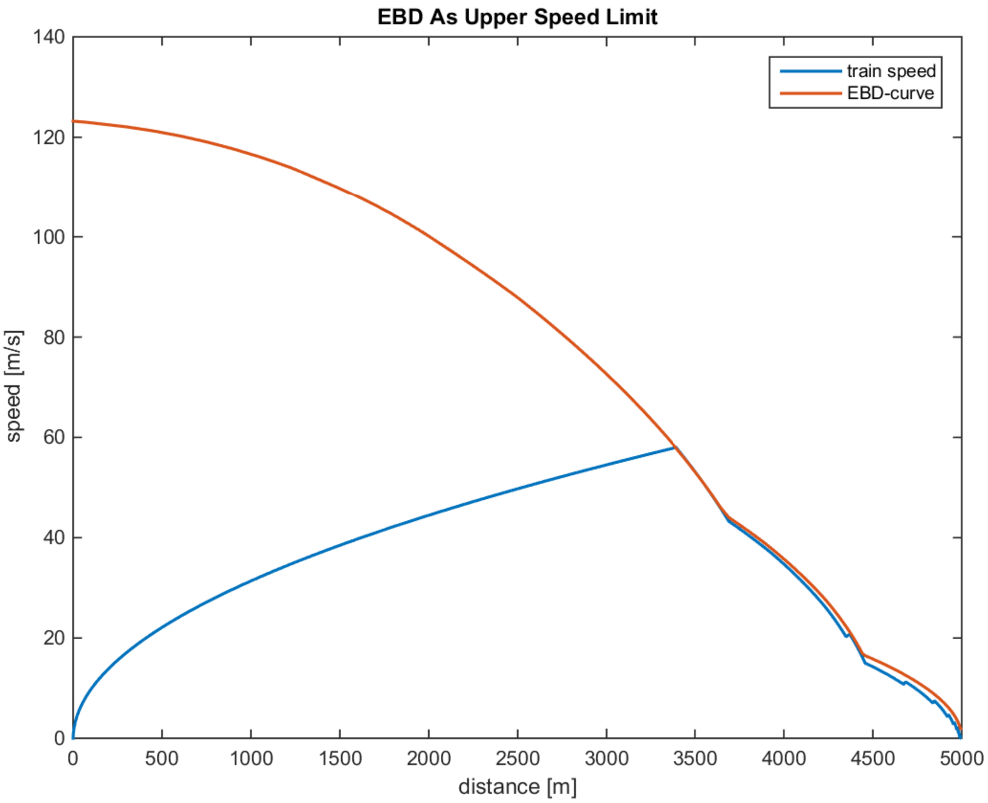


Figure 7: Simulation result of an example test case

References

[BFM+11] Bernardi, Simona, Francesco Flammini, Stefano Marrone, José Merseguer, Camilla Papa, and Valeria Vittorini: *Model-driven availability evaluation of railway control systems*. In Flammini, Francesco, Sandro Bologna, and Valeria Vittorini (editors): *Computer Safety, Reliability, and Security*, volume 6894 of *Lecture Notes in Computer Science*, pages 15–28. Springer Berlin Heidelberg, 2011.

[BHH+14] Braunstein, Cécile, AnneE. Haxthausen, Wen ling Huang, Felix Hübner, Jan Peleska, Uwe Schulze, and Linh Vu Hong: *Complete model-based equivalence class testing for the etcs ceiling speed monitor*. In Merz, Stephan and Jun Pang (editors): *Formal Methods and Software Engineering*, volume 8829 of *Lecture Notes in Computer Science*, pages 380–395. Springer International Publishing, 2014.

[BT11] B.Vincze and G. Tarnai: *Development and analysis of train brake curve calculation methods with complex simulation*. Advances in Electrical and Electronic Engineering, 5(1-2):174–177, 2011.

[CA14] C. Andrés, A. Cavalli, N. Yevtushenko J. Santos R. Abreu: *On modeling and testing components of the european train control system*. In *International Conference on Advances in Information Processing and Communication Technology - IPCT 2014*. UACEE, 2014.

[CPD$^+$14] Claire Dross, Pavlos Efstathopoulos, David Lesens, David Mentré, and Yannick Moy: *Rail, space, security: Three case studies for SPARK 2014*. Toulouse, February 2014.

[Fri10] Friman, B.: *An algorithm for braking curve calculations in ertms train protection systems*. Advanced Train Control Systems, page 65, 2010.

[HJU05] Hermanns, H., D.N. Jansen, and Y.S. Usenko: *A comparative reliability analysis of etcs train radio communications*, February 2005. AVACS Technical Report No. 2.

[JzHS98] Jansen, L., M. M. zu Hörste, and E. Schnieder: *Technical issues in modelling the european train control system*. Proceedings of the workshop on practical use of coloured Petri Nets and Design /CPN 1998, pages 103–115, 1998.

[Mat14] Mathworks: *Simulink - Simulation and Model-based Design, Version R14*. 2014.

[MFM$^+$14] Marrone, Stefano, Francesco Flammini, Nicola Mazzocca, Roberto Nardone, and Valeria Vittorini: *Towards model-driven v&v assessment of railway control systems*. International Journal on Software Tools for Technology Transfer, 16(6):669–683, 2014.

[OMG12] OMG, Object Management Group: *Systems Modeling Language (SysML), Version 1.3 Reference Manual*. 2012.

[UNI12] UNISIG: *SUBSET-026 – System Requirements Specification*. Srs 3.3.0, ERA, 2012.

[ZH05] Zimmermann, A. and G. Hommel: *Towards modeling and evaluation of etcs real-time communication and operation*. Journal of Systems and Software, 77(1):47–54, 2005.

[zHHS13] Hörste, M.M. zu, H. Hungar, and E. Schnieder: *Modelling functionality of train control systems using petri nets*. Towards a Formal Methods Body of Knowledge for Railway Control and Safety Systems, page 46, 2013.