Bachelor Thesis

Simulating and Evaluating
the Lötschberg Train Traffic System

Department of Computer Science
University of Applied Sciences Rapperswil

Spring Term 2013

Version for Publication

Authors:	Alain Kaeslin
	Michael Gerber
Advisor:	Prof. Dr. Luc Bläser
Project Partner:	systransis AG, Rotkreuz
	Dr. Farhad Mehta
	Dd. Xiaolu Rao

June 14, 2013

Abstract

The Lötschberg Base Tunnel is a 34.6 km long railway tunnel, consisting of a 19 km single track section. This poses particular challenges for the tunnel operation to minimize track occupation conflicts. For this reason, a train traffic control system is in operation that forecasts train movements, calculates optimal speed trajectories and sends corresponding advisory speeds to the train drivers, who can in turn adjust their speed.

The aim of this thesis is to analyze the added benefit of the existing control system compared to an operation without speed recommendations, and investigate on further possible improvements.

We have developed software to simulate conflict situations with and without an advisory speed system. The results are shown in combined speed-distance, force-distance and time-distance plots. This visualization enables an in-depth interpretation of the conflict situation and driver behavior. To evaluate the overall impact, we calculated four main railway line capacity indicators („number of trains", „average speed", „heterogeneity" and „stability") using a quantified model. They can be compared in a spider chart in our web-based analysis tool.

Simulation results show, that in conflict cases, approximately 25% of the trains are able to leave the single track section at least 15 seconds earlier thanks to the current advisory speed system. Additionally, we estimate that at least 12.5% of energy is saved.

Our evaluations also show that a detailed vehicle dynamics model is very beneficial for the crossing conflict optimization algorithm. We therefore propose an enhanced algorithm, which consistently respects engines' limitations to improve the resulting speed trajectories towards capacity gains. When such a model is used, we anticipate that in around 65% of the conflict cases, the affected train can clear the single track section at least 15 seconds earlier.

Declaration

We, Michael Gerber and Alain Kaeslin declare

- That this thesis and the work presented in it is our own, original work.

- All the sources we consulted and cited are clearly attributed. We have acknowledged all main sources of help.

- No copyright is infringed.

Rapperswil, June 14, 2013

Alain Kaeslin   Michael Gerber

4

Acknowledgements

At the end of our thesis we would like to thank all those people who made this thesis possible and an unforgettable experience for us.

We would like to express our gratitude to our supervisor Prof. Dr. Luc Bläser from HSR Hochschule für Technik Rapperswil for his continuing advice and encouragement throughout this thesis.

Our gratitude also goes to Dd. Xiaolu Rao, Dr. Farhad Mehta and the whole team at *systransis AG* for making this thesis possible and introducing us to this interesting topic.

Contents

Chapter 1

Background Information

1.1 Lötschberg Base Tunnel

The Lötschberg Base Tunnel (LBT) is a 34.6 km long railway tunnel put in service in 2007 and operated by *BLS AG*. It accommodates passenger and freight trains. The tunnel connects Frutigen (Berne) with Raron (Valais). The tunnel is part of the Bern–Lötschberg–Simplon axis and the Swiss AlpTransit (NEAT - Neue Eisenbahn-Alpentransversale) initiative.

In the time period between its opening in 2007 and the end of 2011, more than 128'000 trains have traveled trough the tunnel. Its occupancy rate is between 80 and 100% (Source: bls [4]). Maximum speeds of 250 km/h are allowed inside the tunnel. But as of 2013, regular trains only reach maximum speeds of 200 km/h[1].

In 1996, the Swiss federal council decided to put forward the construction of the tunnel. For financial reasons, it was decided to only install railway equipment in one of the two parallel tubes in full length. Therefore, 19 km of the tunnel are a single track section. This topology (see Figure 1.1) introduces particular challenges in the operation of the tunnel.
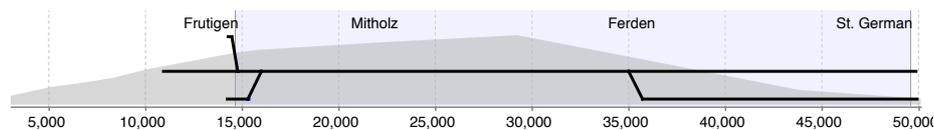
Figure 1.1: Topology of the Lötschberg Base Tunnel. 19 km is comprised of a single track section. (Data Source: Wägli [14].)

1.2 European Train Control System

The Lötschberg Base Tunnel is Switzerland's second railway line that uses the European Train Control System (ETCS) Level 2 as its signalling, control and

---

[1] The only vehicle on this line allowed for 250 km/h, the ETR 610, does not currently „fulfill all requirements" (bav [3]).

train protection system. It is Switzerland's first installation where no conventional fallback system is installed.

The ETCS L2 features are:

Cab signalling In ETCS L2, no conventional line side signals are used. Instead, the maximum allowed speed and other information on the track ahead is transmitted to the train using GSM-R (see bellow) and shown on the Driver Machine Interface (DMI) inside the driver's cab.

GSM-R In ETCS L2, GSM-R (Global System for Mobile Communications - Railway) is used to enable communication between the train's On-Board-Unit (OBU) and the radio block centre (RBC). Among other information exchanged between the two communication parties, the train reports information about its speed and position to the RBC.

An overview of the ETCS L2 components are shown in Figure 1.2:
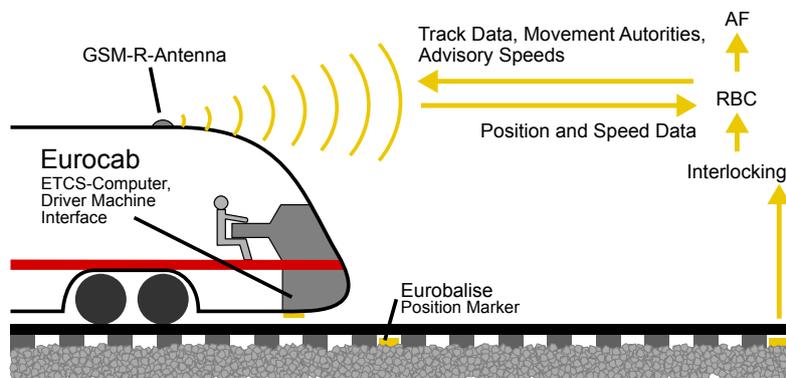
Figure 1.2: Components of the European Train Control System (ETCS) Level 2 as installed in the Lötschberg Base Tunnel. (Picture source see page 68.)

1.3 Automatic Functions

The *Automatic Functions* (AF) train traffic control system was developed by *systransis AG*, a subcontractor of *Thales Rail Signalling Solutions AG* on behalf of *BLS Alptransit AG*. The purpose of AF is to automate part of the operation of the tunnel. This includes solving track occupation conflicts. In other environments, this task is usually performed by the train dispatcher („Fahrdienstleiter").

Disturbances such as delays and unscheduled trains may lead to track occupation conflicts. AF detects a conflict by forecasting a train's movement and solves it by calculating an optimal speed trajectory. AF then sends an advisory speed to the train driver who can in turn adjust the current speed.

This thesis focuses on crossing conflicts (see Figure 1.3). A crossing conflict arises when a train approaching the single track section (blue) is prohibited from

entering it, because the section is still occupied by another train traveling in the opposite direction (orange). In crossing conflict situations, advisory speeds are sent to the train approaching the single track section.

Figure 1.3: Illustration of a crossing conflict. The delayed orange train causes the conflict while the blue one is affected by it.

The impact of advisory speeds is illustrated in Figure 1.4. In the non-optimized case, the train travels at full speed through the double-track section and is forced to come to a full stop in front of the single track section until the conflict train has released the shared track segment. As soon as the train is allowed to enter the single track section, the driver accelerates from 0 to the maximum allowed speed.

Thanks to the advisory speeds, the driver becomes aware of the conflict in advance. He adjusts the speed while passing trough the double-track section and therefore prevents a full stop, which saves energy. Moreover, the train can enter the single track section faster and thus minimizes its occupation time.
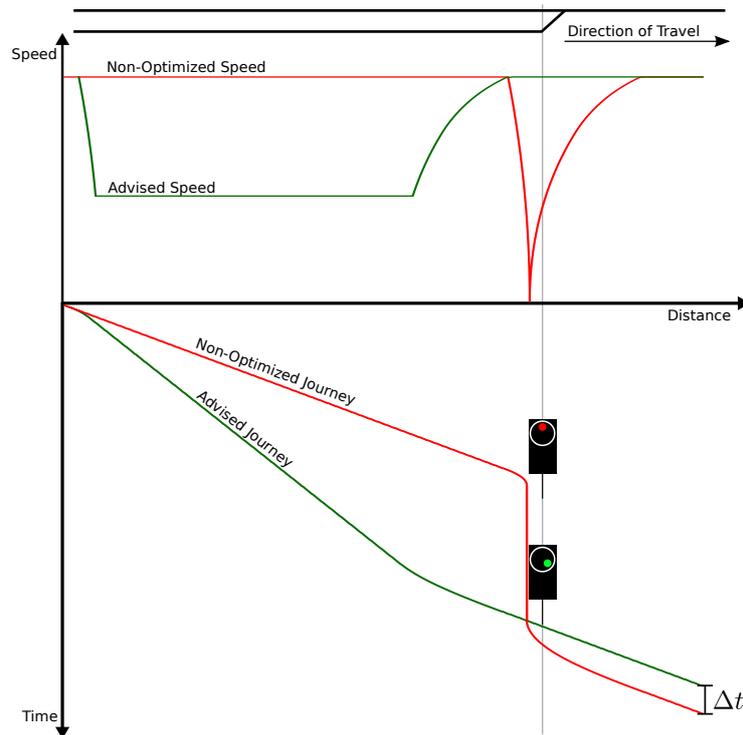
Figure 1.4: Trajectories of a train affected by a conflict (shown in blue in Figure 1.3). The red line represents the non-optimized scenario, while the green line shows the optimized scenario. Δt is the time saving.

1.4 Driver Behavior

No supervision of the drivers' adherence to the speed recommendations by AF is performed. Drivers are therefore not forced to follow the proposed trajectories and are free to drive above or below the advised speed.

However, ETCS will enforce the maximum possible braking curve to ensure that a train affected by a conflict will always come to a full stop before the single track segment, in case it has not yet been released by the approaching train.

Figure 1.5: Mockup of an ETCS Driver Machine Interface (DMI). The advisory speed („Vopt 100") is shown to the driver in the text message area on the left bottom. (Picture source see page 68.)

Chapter 2

Goals and Problem Statement

This chapter outlines the scope and goals of this thesis. For every addressed field, the objectives for this project and current solutions are compared.

2.1 Project Goal

The overall project goal was described by Farhad Metha in the task description as follows:

> „The analysis will demonstrate the added benefit concerning capacity usage and delay reduction of a dynamic train traffic management system using speed recommendations for the AF in the Lötschberg Base Tunnel as an example. Therefore, the achieved driving behavior and accuracy in case of an optimization must be determined based on the available data. In addition, further strategies to improve controllability and reactivity of trains as part of dynamic train traffic management systems should be developed."
>
> *- Farhad Metha, systransis AG*

2.2 Scope

Based on the task description authored by our supervisor and project partners we have identified the following fields that need to be addressed in this project.

2.2.1 Data Preparation

The system will use the AF (server) log file created during operation as input to extract information on train runs. The current software for parsing the log file is not extensible. We will therefore develop a modular and flexible log parsing solution.

2.2.2 Conflict Scenarios Simulation

For every conflict identified in the log file, the system must simulate and visualize the following scenarios under the assumption, that the train affected by the conflict enters the ETCS L2 area at the same time and speed as read from the log file:

Reality This scenario represents the actual operation in the tunnel. As tunnel operation is always performed with the AF turned on, it considers the advisory speeds and the drivers' compliance to them. It is based on the train position and speed information reported by the trains and which is written in the log file.

AF-optimized This hypothetical scenario is simulated under the assumption, that drivers always follow speed recommendations accurately and immediately.

Non-optimized This hypothetical scenario is simulated under the assumption, that no optimization is performed and that drivers are not notified about conflicts in advance. Here we assume that the line is used with ETCS Level 2 and cab signaling, but with no advisory speeds from AF.

2.2.3 Speed-Distance-Time Trajectories for Conflicts

The goal is to visualize the speed-distance-time (v-x-t) trajectories (see Figure 1.4) for all simulated conflict scenarios (section 2.2.2). A v-x-t trajectory consists of a $v(x)$ and a $t(x)$ plot with shared x-axis. The relationship between the two plots is given in Equation 2.1. The $v(x)$ plot is sometimes referred to as „Fahrschaulinie". The $t(x)$ plot is often used in „graphische Fahrpläne".

$$t(x) := -\int \frac{1}{v(x)}\, dx \tag{2.1}$$

We have identified several possibilities to enhance the existing log analysis software that we will address in the following ways:

- The software should be able to list all trains for a time period in an overview table.

- The software should automatically detect conflicting trains for a train selected by the user.

- The diagrams should include additional useful information such as the tunnel's topology and the track's inclination in order to fully interpret the conflict.

Based on these findings, we propose an improved graphical user interface and enhanced diagrams that simplify conflict analysis.

Comparisons between the non-optimized and reality scenario have been performed by Mehta et al. [10]. This calculation could not be performed in a web application and does not provide visual output at the start of our project. The difference between the AF-optimized scenario and the other scenarios had not been analyzed until now.

2.2.4 Vehicle Dynamics

In order to simulate the two hypothetical scenarios described in section 2.2.3, a model of the train's vehicle dynamics is required. *systransis* has developed Java software system that will be used and extended in our project.

2.2.5 Visualization of Tractive and Braking Force

To improve the explanatory power, we will include a force-distance ($F(x)$) plot in the trajectories generated by the software. The combination of the three plots give a very detailed view of all scenarios. Especially for the reality scenario, the extended trajectory enables an understanding of the driver behavior as the application of tractive and braking force can be seen in the diagram.

The diagram will be include plots of the forces that an engines can exert for propulsion and recuperation respectively (envelope of electrical traction). This provides the ability to verify that a hypothetical trajectory is indeed feasible. And for trajectories extracted from the operation of the tunnel the mode of operation chosen by the driver (e.g. coasting, regenerative braking, etc.) can be read from the diagram.

The force-distance ($F(x)$) plot is conceptually explained in Figure 2.1. Analysis of Figure 2.1a and 2.1b leads to the following conclusions about the sample $v(x)$ plot shown:

- The first deceleration is performed by coasting, as neither a tractive nor a braking force is applied.
- The second deceleration required dissipative braking as the maximum recuperation force is exceeded.
- The acceleration is not achievable as the tractive force exceeds the engine's capabilities.

2.2.6 Estimation of Energy Consumption

The goal is to perform an estimation of the energy consumption for each scenario described in section 2.2.3. Similar calculations had already been published by Mehta et al. [10], although only the „Reality" and „Non Optimized" scenario were compared.

2.2.7 Benefits of Improved Vehicle Dynamics Model

The „Automatic Functions" (AF) project was initiated in 2003. Its primary goal is to avoid unplanned stops inside the tunnel (as safety concern). In this context, a simplified vehicle dynamics model was used. In order to further improving railway capacity and energy efficiency, applying a detailed vehicle dynamics model is necessary. In this regard, an alternative algorithm focusing on enhanced vehicle dynamics model will be proposed in this thesis, resulting in better capacity performance.

2.2.8 Capacity Analysis

In contrast to the analysis functions described in section 2.2.3, which focuses on single conflict situations only, the capacity analysis aggregates the results from multiple conflicts over a given time period.
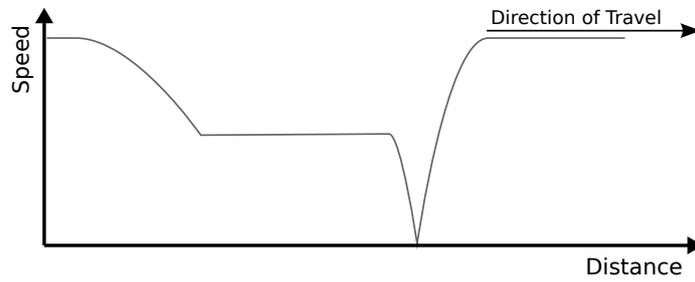
(a) Sample $v(x)$ plot.
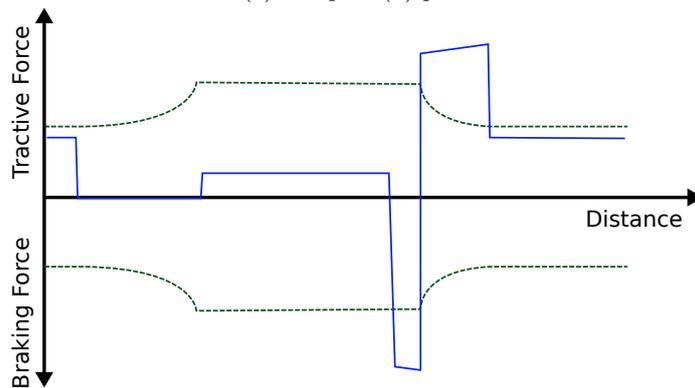
(b) $F(x)$ plot for the $v(x)$ plot shown in (a). The green line delimits the forces that an engines can exert for propulsion and recuperation respectively (envelope of electrical traction) and is extracted from the engine's force-speed characteristics (see (c)). The blue line shows the tractive force required to follow the speed profile shown in the $v(x)$ plot.
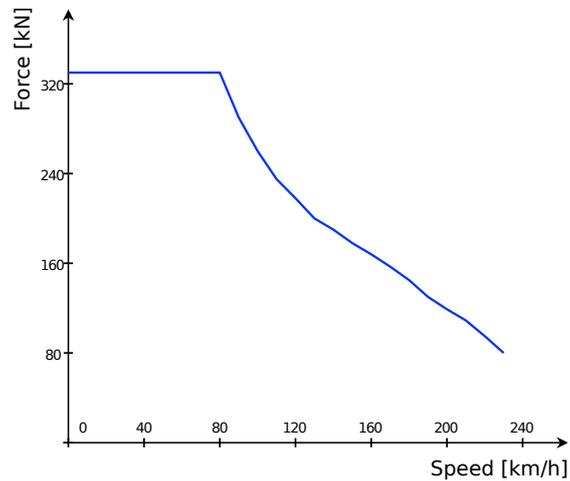
(c) Sample force-speed plot (tractive effort curve) for an Re 465 locomotive. These type of diagram is commonly used to describe engine characteristics. The continuous tractive effort is inversely proportional the to speed.

Figure 2.1: Illustration of the concepts of the $F(x)$ plot.

Defining railway capacity is complex, because numerous aspects have to be taken into account. Landex [7] lists eleven definitions from different authors to illustrate „how capacity has been defined differently over time". A widely accepted definition of capacity was published in UIC [1] by the „Union Internationale des Chemins de fer" in 2004:

> „Capacity as such does not exist. Railway infrastructure capacity depends on the way it is utilised. The basic parameters underpinning capacity are the infrastructure characteristics themselves and these include the signalling system, the transport schedule and the imposed punctuality level."

According to UIC [1], the capacity of a railway infrastructure is based on the interdependencies existing between:

- the number of trains per time interval.

- the average speed per time interval.

- the heterogeneity, in particular the frequency of trains, their speeds and their stopping patterns.

- the stability, which depends on the margins and buffer times available to recover from disturbances.

An example for the interdependency is given in Lüthi [8]: „A higher heterogeneity of trains results in a change in average speed and in varying headways causing a lower number of trains using the infrastructure compared to the case with homogenous (identical) trains."

The vision of *systransis* is to add an additional capacity analysis function into the existing log analysis tool, that presents the four factors mentioned above in a „capacity balance spider" chart (see Figure 2.2). The spider gives an overall view on capacity. Our analytics tool will enable the user to get insight on how the capacity is utilized. UIC [1] defines the „capacity balance spider" as follows:

> „A chord links the points on the axes, corresponding to the value of each parameter. The length of the chord represents the capacity. Capacity utilisation is defined by the positions of the chord on the four axes. Increasing capacity means increasing the length of the chord."

Quantified Model

UIC [1] only defines the „capacity balance" in a qualified way. This thesis will however use a quantified model for the „capacity balance spider" featuring normalized axes. This extension was proposed by Xiaolu Rao.
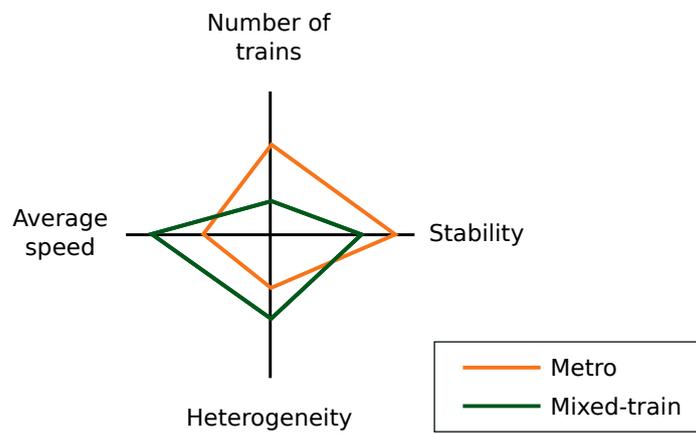
Figure 2.2: „Capacity balance spider" chart comparing the characteristics of two different network types.

Chapter 3

Methodology and Design

This chapter describes the methodologies applied and design decisions made to solve the problems introduced in chapter 2.

3.1 Domain Model

In order to gain an in-depth understanding of the problem domain and the data available in the log file, the following domain model was created:
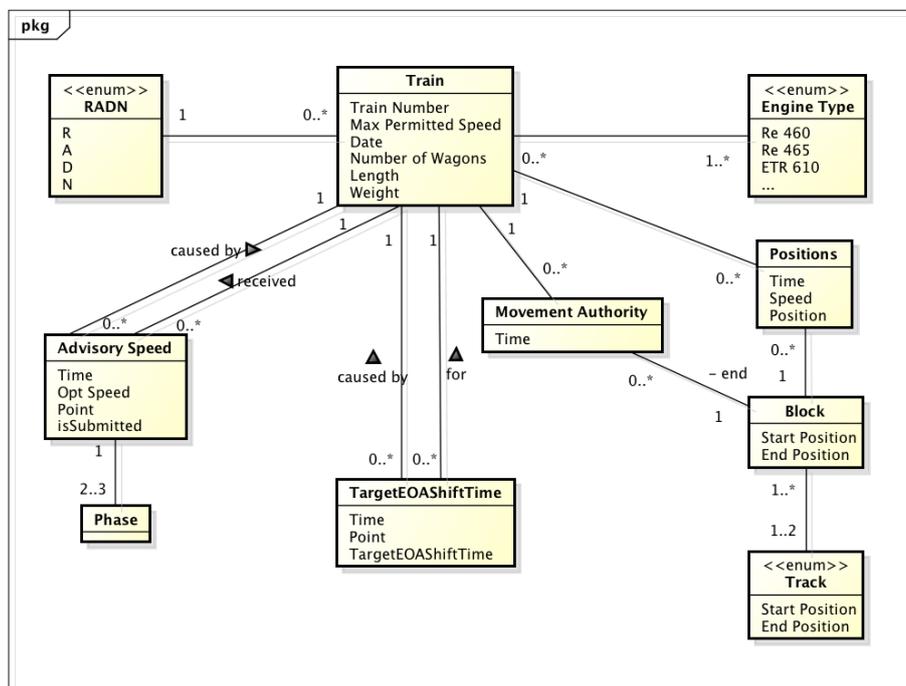
Figure 3.1: The domain model used in this project. Note that this diagram does not represent software classes.

3.2 Data Preparation

The goal is to extract data from the log file and structure it according to the domain model shown in section 3.1. The log file's characteristics are as follows:

- The log file is sorted in a chronological order.

- All log entries are associated with time stamps.

- The log file consists of XML as well as unstructured text entries.

- Only a subset of the entries in the log file are of interest for this project.

3.2.1 Solutions Considered

A solution was required to filter the relevant entries and extract the required information according to the domain model (section 3.1).

Handling the XML entries using a XML parser and plain text entries using regular expressions (RegEx) would introduce two fundamentally different approaches for log parsing. We therefore decided to use RegEx for all entries. This allows us to treat all entries in a unified way.

Java's extensive regular expression libraries (especially „matcher groups") enable a clean and straightforward way to extract the relevant information from either XML and unstructured plain text entries.

3.3 Conflict Scenarios Simulation

This section describes the models for the scenarios in detail.

3.3.1 Reality Scenario

No model or simulation is required for the reality scenario, because all position, speed and time information can be read from the log file.

3.3.2 AF Optimized

AF three phase trajectories consist of a deceleration, a constant speed traveling and an acceleration phase. The last phase ends with the maximum speed permitted at the brake entrance point (BEP) to ensure that a full stop is possible in case the conflict is not resolved at the time predicted. This is illustrated in Figure 3.2.

Speed and position information is available for the start and end point for each phase in the log file. This information is extracted by our software. After the last point extracted from the log file, the trajectory is extended with a simulation of the journey to the end of the tunnel.
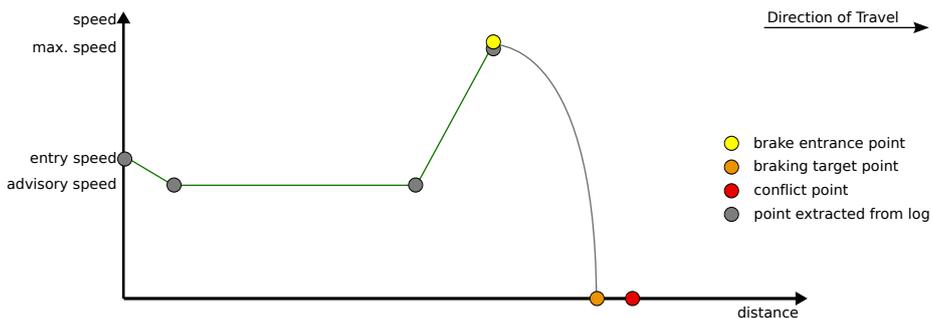
Figure 3.2: Illustration of a three phase trajectory calculated by AF. The trajectory always ends at the brake entrance point with the maximum speed permitted. The four points shown in gray are read from the log file.

3.3.3 Non Optimized Scenario

The steps shown below are repeated for every potential conflict point on the train's path trough the tunnel. The number of conflict points depends on the path a train travels trough the tunnel (see Figure 1.1).

1. In the first step we assume, that the conflict is only solved after the train has come to a full stop. We construct the trajectory shown in Figure 3.3 consisting of the following phases:

   (a) accelerate to the maximum speed permitted

   (b) continue traveling at maximum speed permitted until we reach the brake entrance point

(c) decelerate until coming to a full stop

It is possible that the phase in step 1. (b) does not exist as the train cannot reach its maximum speed permitted and come to a full stop at the braking target point. The deceleration phase will then start immediately after the acceleration phase.

2. In the second step, we determine at which point in time, the train received the movement authority to travel over the conflict point. This step is illustrated in Figure 3.4.

   (a) The trajectory from item 1 is backtracked to the time where the movement authority was granted.

   (b) The trajectory remaining after step 2. (a) is extended with an acceleration phase to the maximum speed permitted and then continues with constant speed until the next conflict point or the end of the tunnel.

   If the train comes to a full stop, a reaction time of 20 seconds is added before the train starts to accelerate. The driver requires this time to react and to release the brakes (see FDV [2]). This is an estimated average value, as the exact time required can vary between a few seconds and one minute.
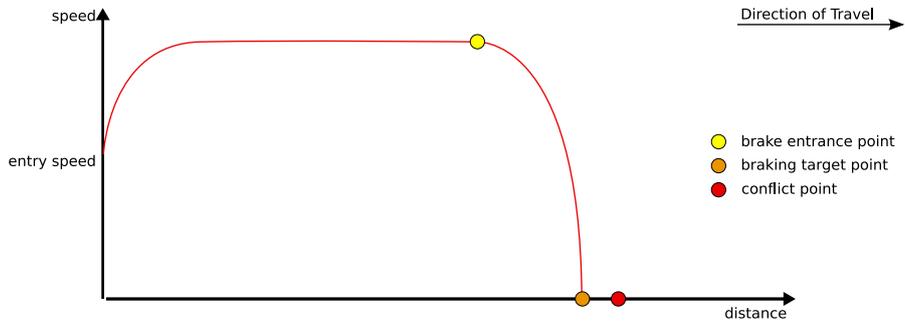
Figure 3.3: First step of the non-optimized scenario simulation.
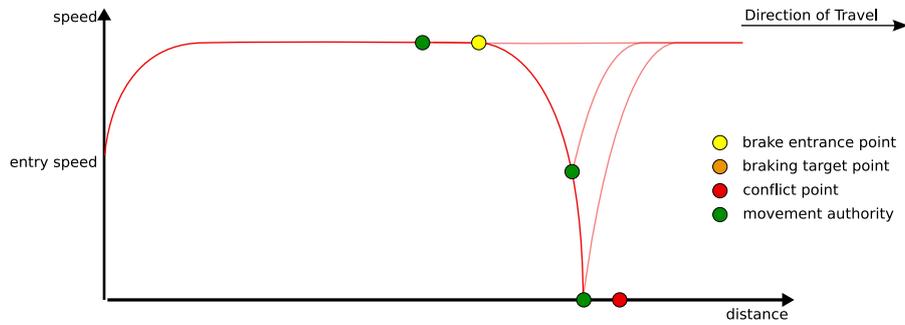
Figure 3.4: Second step of the non-optimized scenario simulation. This illustration shows three different points where the movement authority is given to the train and the resulting trajectory for each point. It is noteworthy that the movement authority can be given at any point along the trajectory calculated in Figure 3.3.

3.4   Vehicle Dynamics

systransis developed Java code to numerically calculate the vehicle dynamics. This model takes the following aspects into account:

- The engine's characteristics: for each engine type the tractive effort curve ($F(v)$-curve) is available.

- The rolling resistance is determined by using the Davis formula, which is based on empirical data.

- The tunnel resistance can be determined for a given speed and train type (passenger or freight).

- The inclination resistance can be determined for a given position.

- The deceleration is assumed to be constant. For passenger trains: $-0.8\,\mathrm{m/s^2}$ and for freight trains: $-0.6\,\mathrm{m/s^2}$

Some experimental comparisons performed in this project between the calculations performed with the existing source code and real log data showed that the simulations yield good results.

During this project, the demand for a more detailed simulation of the braking behavior came up. According to Hürlimann [6] and Wende [15] either empirical data on the braking behavior or detailed information about the physical properties of the vehicles brakes (engines and wagons) are required to perform this calculation. Furthermore, a detailed simulation would require knowledge on the driver's and the engine's on-board computer's decisions when and how strongly to apply which braking system (regenerative braking with traction motors, mechanical braking, track brake etc.). For the reasons mentioned, a detailed braking simulation is outside of this project's scope.

As mentioned in section 2.2.4, the existing source code of *systransis* provided a good starting point to simulate the vehicle dynamics of a train.

3.5 Force and Energy Analysis

The following sections describe the necessary steps to calculate and visualize the tractive force as a function of position. Moreover, the method to perform an energy estimation for each scenario is described.

Sense of Direction

In the following formulas, forces are expressed relatively to the direction of travel. Forces greater than 0 are oriented in the direction of travel. Forces less than 0 are oriented opposite to the direction of travel. The absolute orientation is ignored. This is illustrated in Figure 3.5.
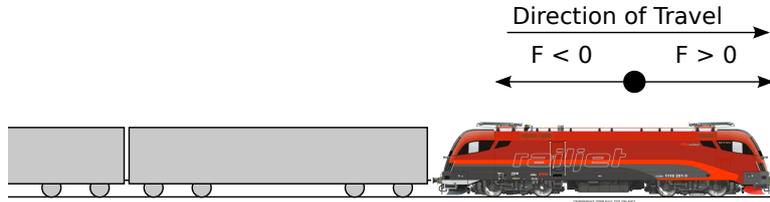
Figure 3.5: Forces are expressed relative to the direction of travel. (Picture source see page 68.)

3.5.1 Tractive and Regenerative Braking Force

The tractive force exerted by the engine motors and braking systems, as a function of position $F_T(x)$, can be determined by using Equation 3.1.

$F_T(x)$ never exceeds $\hat{F}_T(v(x))$, which is the sum of the tractive forces of all engines. The maximum tractive forces of each engine is determined using the corresponding tractive effort curve.

$$F_T(x) = F_A(v(x)) + F_R(v(x)) + F_I(x) + m \cdot e_i \cdot a(x) \tag{3.1}$$

$$\hat{F}_T(v(x)) = \sum_{k=1} \hat{F}_{T,k}(v(x)) \tag{3.2}$$

$$\tag{3.3}$$

where...

$F_T(x)$	is the tractive force exerted at point x
$v(x)$	is the speed at point x
$F_A(v(x))$	is the air resistance at point x
$F_R(v(x))$	is the rolling resistance at point x
$F_I(x)$	is the inclination resistance at point x
$m \cdot e_i \cdot a(x)$	is the acceleration resistance at point x
e_i	is the mass factor due to moment of inertia of the rotating mass
$\hat{F}_T(v(x))$	is the maximum available tractive force at point x from all engines
$\hat{F}_{T,k}(v(x))$	is the maximum available tractive force at point x for engine k

To estimate how much braking effort can be exerted by the regenerative brake ($\check{F}_{regen}(v(x))$), we assume that the braking force for any given speed, is approximately equal to the tractive force at the same speed:

$$\check{F}_{regen}(v(x)) \approx -\hat{F}_T(v(x)) \tag{3.4}$$

3.5.2 Work

The mechanical work performed by a train to travel from point x_1 to x_2 can be determined by integrating the tractive force over the distance traveled as shown in Equation 3.5:

$$W(x_1, x_2) = \int\limits_{x_1}^{x_2} F_T(x)\, dx \tag{3.5}$$

Our goal here is to determine the engine and braking forces involved from an x-v-t trajectory in order to verify, that some given advisory profile is indeed feasible. Further, this allows the estimation of the overall energy consumption. First of all, we distuinguish between a propulsion force exerted by the electric motors and a braking force (Equation 3.6). Secondly, for trains with electric traction, braking can occur by regenerative braking of the engine, by dissipative braking or by a combination of the two (Equation 3.7)[1]. We assume that drivers resort to dissipative braking only when the imposed deceleration exceeds the regenerative capabilities of the engine. This is illustrated in Figure 3.6, where a pair of dashed lines delimits the forces that an engines can exert for propulsion ($\hat{F}_T(v(x))$) and recuperation ($\check{F}_{regen}(v(x))$), respectively (envelope of electrical traction). Using Equations 3.8 to 3.11 and Table 3.1, the mode of operation can be determined for any situation.

$$W(x_1, x_2) = \int\limits_{x_1}^{x_2} F_{prop}(x)\, dx + \int\limits_{x_1}^{x_2} F_{brake}(x)\, dx \tag{3.6}$$

$$W(x_1, x_2) = \int\limits_{x_1}^{x_2} F_{prop}(x)\, dx + \int\limits_{x_1}^{x_2} F_{regen}(x)\, dx + \int\limits_{x_1}^{x_2} F_{diss}(x)\, dx \tag{3.7}$$

$$F_{prop}(x) = \begin{cases} F_T(x) & F_T(x) > 0 \\ 0 & \text{otherwise} \end{cases} \tag{3.8}$$

$$F_{brake}(x) = \begin{cases} F_T(x) & F_T(x) < 0 \\ 0 & \text{otherwise} \end{cases} \tag{3.9}$$

$$F_{regen}(x) = \begin{cases} F_{brake} & F_{brake} \geq \check{F}_{regen}(v(x)) \\ \check{F}_{regen}(v(x)) & \text{otherwise} \end{cases} \tag{3.10}$$

$$F_{diss}(x) = \begin{cases} F_{brake}(x) - \check{F}_{regen}(v(x)) & F_{brake} < \check{F}_{regen}(v(x)) \\ 0 & \text{otherwise} \end{cases} \tag{3.11}$$

---

[1]Dissipative braking is meant to include both rheostatic and mechanical braking as no energy gets recuperated in either case.
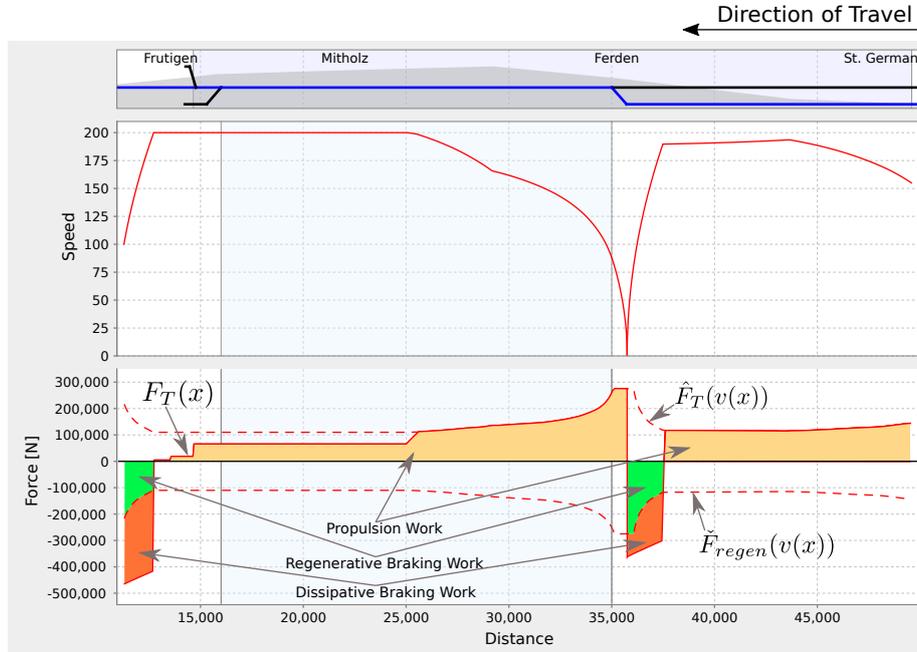
Figure 3.6: Illustration of the functions and integrals from Equation 3.7.

	Forces applied ($\neq 0$)		
Mode of Operation	F_{prop}	F_{regen}	F_{diss}
Coasting	false	false	false
Powered	true	false	false
Recuperating	false	true	false
Combined Braking	false	true	true
(Dissipative Braking Only)	false	false	true

Table 3.1: Modes of operation for electrical vehicles. Dissipative braking only mode is ignored in this thesis, as it is a bad behavior leading to a suboptimal energy consumption, because the regenerative braking capabilities of the engines are not exploited.

3.5.3 Electrical Energy

The electrical energy consumed by a train to travel from point x_1 to x_2, is the sum of the propulsion and regenerative braking work, multiplied by their respective energy conversion efficiency ratio (η):

$$E(x_1, x_2) = \eta_{prop} \int_{x_1}^{x_2} F_{prop}(x)\, dx + \eta_{regen} \int_{x_1}^{x_2} F_{regen}(x)\, dx \qquad (3.12)$$

As we lack information on the energy conversion efficiency ratio for each engine type, we cannot estimate the electrical energy. Therefore, we only approximate the mechanical work (mechanical energy) performed as described in section 3.5.2.

3.5.4 Sampling and Quantization of Speed and Position Data

The speed and position information available in the log file is sampled in irregular time intervals. Speed entries are quantized with 5 km/h and position entries with 1 m steps. This turned out to be a challenge, when calculating the tractive force numerically as the acceleration between two values might be far too high. The result was, that in the force-distance plot, huge peaks appeared whenever the speed value jumped from one 5 km/h quantization step to the next.

As a workarround, the values missing due to irregular sampling are filled up by repeating the last available information. Subsequently, a triangular weighted moving average filter is applied to every speed information in the log.

The results of this technique are illustrated in Figure 3.7a and 3.7b. The difference between the two diagrams is significant. However, this solution should only be considered as a workaround due to not fully detailed log information. In order to increase the analysis quality in the future, we recommend that the quality of the speed and position information logged is increased.
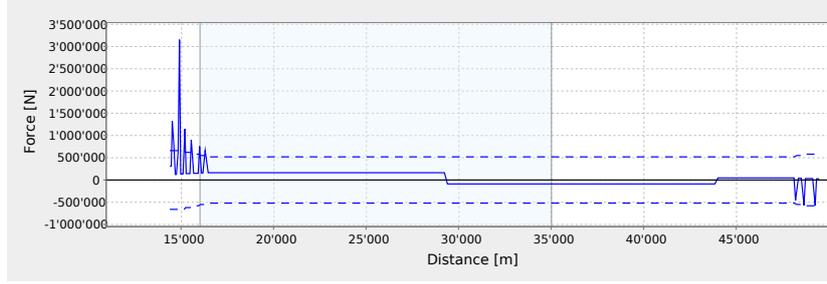
(a) Without filter applied.
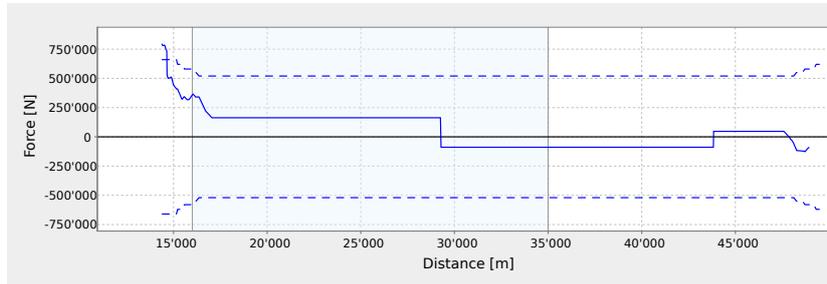
(b) With filter applied.

Figure 3.7: The same distance-force plot with and without the triangular weighted moving average filter applied to the speed-position information.

3.6 Benefits of Improved Vehicle Dynamics Model

3.6.1 Definition of the Optimization Problem

The conditions for our definition of an optimal three phase trajectory (deceleration, traveling at constant speed, acceleration) are described in the objective function 3.13 and the constraints 3.14 to 3.15. The variables are illustrated in Figure 3.8.

$$\max\{v(p_{bep})\} \qquad (3.13)$$

$$\Delta t_{decel} + \Delta t_{const} + \Delta t_{accel} \overset{!}{\geq} t_{resolution} \qquad (3.14)$$

$$p_{current} + \Delta d_{decel} + \Delta d_{const} + \Delta d_{accel} + \Delta d_{fullstop} \overset{!}{\leq} p_{eoa} \qquad (3.15)$$

where...

$v(p_{bep})$	is the speed at the brake entrance point
Δt_{decel}	is the time required to decelerate to the recommended speed
Δt_{const}	is the time traveled at the recommended speed
Δt_{accel}	is the time required to accelerate
$t_{resolution}$	is the time until the conflict is expected to be resolved
$p_{current}$	is the current position
Δd_{decel}	is the distance required to decelerate to the recommended speed
Δd_{const}	is the distance traveled at the recommended speed
Δd_{accel}	is the distance required to accelerate
$\Delta d_{fullstop}$	is the distance required to perform a full stop after the acceleration
p_{eoa}	is the end position of the movement authority until the conflict is resolved
$\overset{!}{\geq}$	demands the left hand side to greater or equal than the right hand side.
$\overset{!}{\leq}$	demands the left hand side to less or equal the right hand side.

The objective function 3.13 states, that $v(x)$ is to be maximized at $x = p_{bep}$ to ensure that the highest possible speed is reached before entering the single track section to ensure minimal occupation time of this track segment. Constraint 3.14 states that the three phases must not be finished before the conflict is expected to be resolved. Constraint 3.15 ensures that it is possible for the train to perform a full stop before the end of its movement authority, if the conflict is not resolved at the expected time.

3.6.2 Constraints Not Considered

We assume that an optimal speed trajectory consists of a deceleration, constant speed traveling and an acceleration phase to ensure that the comparability between our proposal and the existing algorithm is guaranteed since we want to demonstrate the impact of a more detailed vehicle dynamics model. The consequences of switching to a different pattern than the three phases mentioned are not considered.

In our definition of the optimization problem (described in Equation 3.13 to 3.15), the only objective function is the speed, which has to be maximized at the brake entrance point (BEP). It is important to stress, that in the real world operation, other aspects influence the „optimal solution". For example, it is not desirable to use very low advisory speeds, such as 15 km/h in the tunnel operation. According to Wyss [18], this might „block following trains" or „damage the locomotives" hauling heavy trains. Such constraints are outside of the scope of this thesis and thus ignored here.

3.6.3 Solutions considered

To solve the optimization problem described in 3.13 to 3.15, either an algorithm based on symbolic computation or a numerical approach can be used.

speed

p_{now} Δd_{decel} Δd_{const} Δd_{acel} $\Delta d_{fullstop}$ p_{eoa}

○ brake entrance point
○ braking target point
● conflict point
● expected movement
authority for next
single track section

entry speed

distance

Δt_{decel}

Δt_{const}

Δt_{accel}

$t_{resolution}$
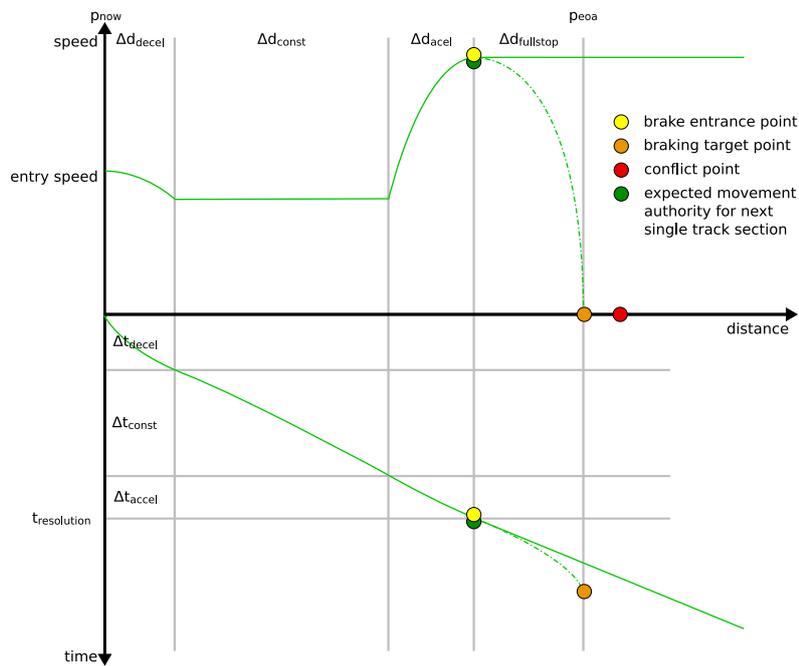
time

Figure 3.8: Illustration of the times and distances mentioned in Equation 3.13 to 3.15. In this figure, two situations are shown: 1) in the main case, the train is granted a movement authority before reaching the braking application point to perform a full stop and is therefore able to travel over the conflict point. 2) The dashed line illustrates the trajectory, if the train is forced to stop, because the time prediction for the movement authority to travel over the conflict point was inaccurate.

As previously mentioned, the vehicle dynamics functions in the existing software are implemented numerically and it is therefore not possible to solve the optimization problem using symbolic computation. Due to its computational intensity when approximated numerically, it would be interesting to investigate, if the optimization problem could be solved faster when calculated symbolically. Unfortunately, rewriting the vehicle dynamics functions in Mathematica or a similar computational software is beyond the scope of this project.

3.6.4 Algorithm implemented

The algorithm implemented approximates the solution by adjusting multiple variables and probing the validity of the resulting trajectories. The proposed algorithm can be divided into the following steps listed below. The process is illustrated in Figure 3.9.

1. An acceleration curve from the current position[2] to the end of the current movement authority is simulated in order to determine the maximum possible speed that can be reached at the conflict point. This speed is used as the initial target speed v_{target} guess to be reached at the conflict point $p_{conflict}$.

2. The current speed is used as an initial advisory speed v_{opt} guess.

3. The brake entrance point p_{bep} for v_{target} speed which ensures a stop at $p_{conflict}$ is calculated.

4. A trajectory consisting of a deceleration phase from the current speed to the v_{opt} speed[3], a constant traveling and an acceleration phase to the v_{target} at the p_{bep} point are calculated.

    4.1. If no trajectory can be found, because v_{target} cannot be reached before p_{bep} point, the algorithm lowers the v_{target} speed.

        - If the new v_{target} speed is > 0, the algorithm continues with step 2.
        - If the new v_{target} speed is $= 0$, no solution could be found and the algorithm terminates.

    4.2. If the time criteria in constraint 3.14 is not met (train arrives at the conflict point before the conflict is expected to be resolved), the algorithm lowers the v_{opt} speed.

        - If the new v_{opt} speed guess is > 0, the algorithm continues with step 4.
        - If the new v_{opt} speed guess is $= 0$, the algorithm lowers the v_{target} speed and continues with step 2.

    4.3. Otherwise, the algorithm has found a solution and terminates.

---

[2]This is the affected train's position, when the system performed a conflict resolution time ($t_{resolution}$) forecast.

[3]In the first iteration, the deceleration phase does not exist, as the advisory speed guess v_{opt} is initialized to the current speed.

As acceleration curves can only be calculated numerically, from their starting point and not from their end point, an additional step is required in the algorithm's implementation, to determine the start point of the acceleration phase for a given end point.
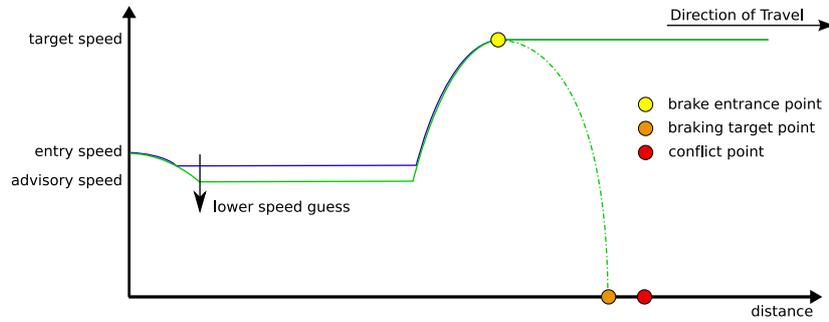
Figure 3.9: Illustration of the algorithm. An optimal trajectory is searched which meets constraint 3.13 and 3.14. In case the time criteria in constraint 3.13 is false (blue), the advisory speed v_{opt} is lowered.

3.7 Capacity Calculations

This section describes the methods used to calculate the values of the four axes of the „capacity balance spider" shown in Figure 2.2.

3.7.1 Area of Interest

The area of interest for the capacity analysis consists of all single track section blocks (722 to 739) and the four blocks making up the two points W60 and W44 which surround the single track section. The three parts are the „shared track segment" in crossing conflict situations.
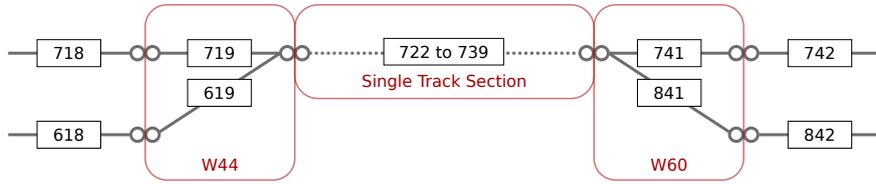
Figure 3.10: The topology of tunnel and its division into blocks illustrated using a double-vertex graph as suggested by Montigel [11]. The area of interest for the capacity analysis is marked red.

3.7.2 Stability

The stability of railway operations depends on the buffer times between trains, time supplements and the unused time in the timetable. This time reserve can be used to recover from disturbances such as delays and conflicts. All of the three time reserve types are reflected in the occupancy rate. We therefore suggest the following definition for stability (Equation 3.16):

$$Stability = 1 - Occupancy\ Rate \tag{3.16}$$

The stability axis of the capacity spider ranges from 0 to 1. A high stability value means, that there is a large time reserve available. A low stability corresponds to a minimal time reserve, which indicates that the ability to recover from disturbances is limited.

Occupancy Rate

We use the „UIC capacity method" (defined in UIC [1]), which is widely accepted in the railway industry, to calculate the occupancy rate. This method is based on compressing the timetable using the „blocking stair". The occupancy rate K is defined as follows (Equation 3.17):

$$K = \frac{t_k}{t_u} \tag{3.17}$$

where... [4]

K is the occupancy rate
t_k is the total capacity consumption measured in time
t_u is the total time observed

The total capacity consumption measured in time t_k is determined by summing up the shortest buffer time between the block occupation times for each consecutive pair of trains in the „blocking stair". This process is illustrated in Figure 3.11 and 3.12.

Block Occupation Time

In order to perform the occupancy rate calculation described above, the occupation time for each block needs to be estimated for every scenario. We use the following assumptions:

- Blocks are occupied 80 seconds before the train enters it. We have determined this value by adjusting it, until it approximately matched the time when movement authorities are granted.

- Blocks are released approximately 4 seconds after the end of the train has cleared the block. This is an empirical value for „modern infrastructure" from Heister et al. [5].

Empirical Value

According to Pachl [12], experience shows that the occupancy rate for stable operation should be between 0.5 and 0.8 (rush hour). We therefore expect the value of the stability ($Stability = 1 - Occupancy\,Rate$) axis on the capacity spider to be between 0.2 and 0.5.

3.7.3 Heterogeneity

A timetable's heterogeneity or homogeneity depends on the frequency of the trains, their speeds and their stopping pattern.

Landex [7] describes several formulas to define a timetable's heterogeneity. The final solution proposed takes into account all aspects mentioned and is described as follows:

> „The homogeneity can be calculated taking both the variation in headway times and speed into account by taking the ratio between the headway time at the departure station ($h_{t,i}^D$) and the following headway time ($h_{t,i+1}^D$). The same ratio is taken at the arrival station using ($h_{t,i}^A$) and ($h_{t,i+1}^A$) and multiplied by the ratio at the departure station."

Samples for this metric are illustrated in Figure 3.13a to 3.13c. The formulas for this metric are presented in Equation 3.18 to 3.20.

---

[4] Symbol names from UIC [1].
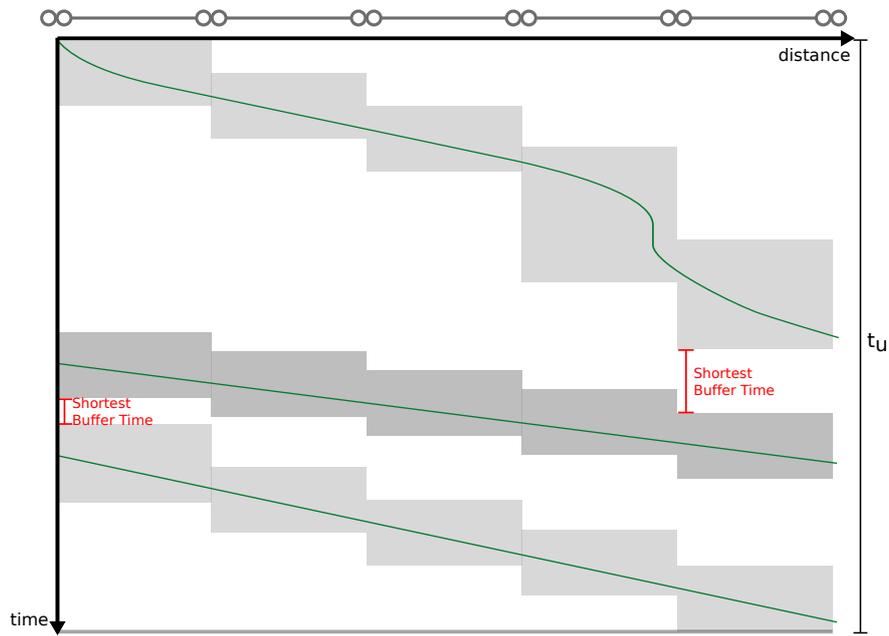
Figure 3.11: Illustration of block occupations (grey) over time shown in a graphical timetable (blocking stairs). Between every consecutive train pair, the shortest buffer time is highlighted in red.
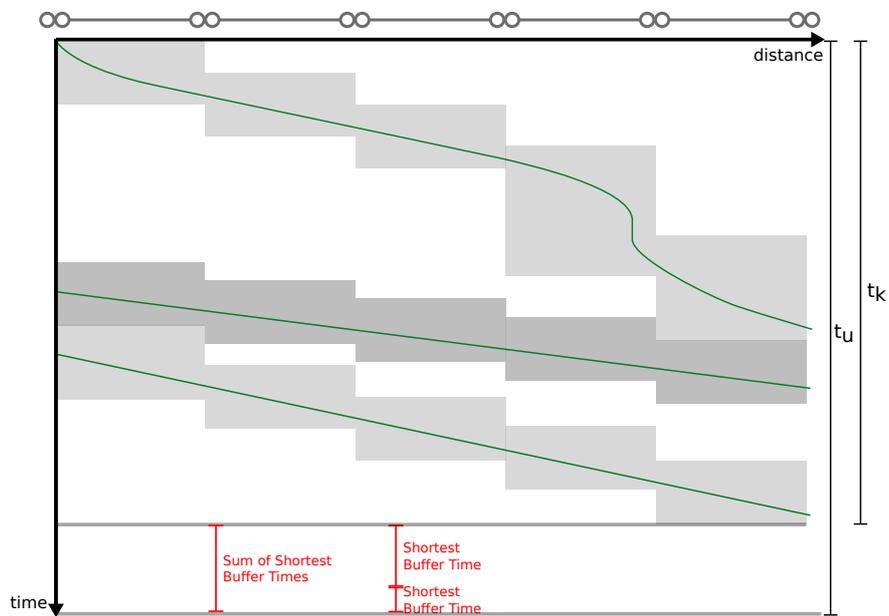
Figure 3.12: Illustration of the graphical timetable shown in Figure 3.11 after compression. The sum of all buffer times is shown in the bottom.

$$Heterogeneity = 1 - Homogeneity \tag{3.18}$$

$$h_N = t_N - 1 \tag{3.19}$$

$$Homogeneity = \frac{\sum_{i=1}^{h_N-1}\left(\min\left(\frac{h_{t,i}^D}{h_{t,i+1}^D};\frac{h_{t,i+1}^D}{h_{t,i}^D}\right)\cdot\min\left(\frac{h_{t,i}^A}{h_{t,i+1}^A};\frac{h_{t,i+1}^A}{h_{t,i}^A}\right)\right)}{h_N-1} \tag{3.20}$$

where...

h_N	is the number of headways observed
t_N	is the number of trains observed
$h_{t,i}^D$	is the ith head way time observed at point D
$h_{t,i}^A$	is the ith head way time observed at point A

Due to its division by $h_N - 1$, the heterogeneity value will always be between 0 and 1. This range is used for the axis in the spider.

3.7.4 Number of Trains

Calculating the number of trains is straightforward. The maximum value of the axis is $\frac{30}{\text{h}} \cdot \Delta t_u$, where Δt_u is the duration of the observed time window.

The number of 200 km/h trains[5] that the tunnel can accommodate per hour is approximately 30, assuming that a block is occupied for around two minutes and the time buffer between the trains is zero.

3.7.5 Average Speed

Calculating the average speed is straightforward. The maximum value of the axis is 250 km/h, i.e. the tunnel's maximum line speed.

3.7.6 Adaptive Advisory Speeds

For the capacity analysis, an adaptive version of the proposed crossing conflict optimization algorithm is used. The trajectory is updated every time a new conflict resolution time forecast becomes available (see Figure 3.14). This is required to ensure, that no infeasible trajectories are used in the capacity estimation.

However, we were not able to create an adaptive version of the existing AF algorithm that can be used in the simulation due to the following reasons:

- The existing source code cannot be reused in our simulation software due to its dependencies to other system components.

- Rewriting the algorithm would not ensure that exactly the same trajectories would result from the existing and the rewritten version.

---

[5]Generally the line's maximum speed should be used here. But as of today 200 km/h is the maximum speed reached by regular trains in the LBT (see section 1.1).
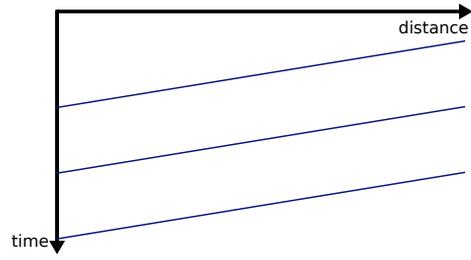
(a) Homogeneity = 1
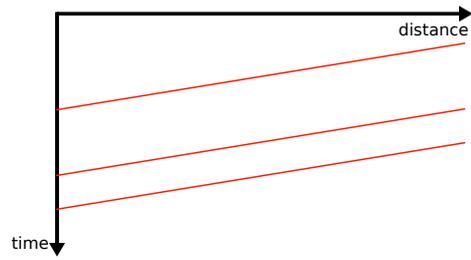
(b) Homogeneity = 0.25
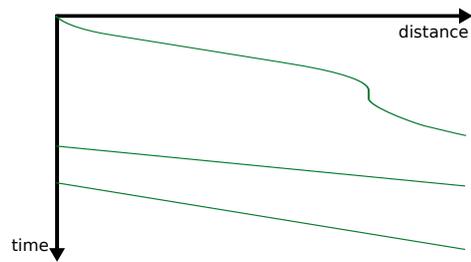
(c) Homogeneity = 0.22

Figure 3.13: Sample timetables and their heterogeneity values as of Equation 3.18.

This will result in some inaccuracy in the simulation results of the existing AF algorithm. However, it is noteworthy that in the current tunnel operation, the algorithm works adaptively, i.e. continuously sends updated advisory speeds, when the conflict resolution time forecast changes or the driver does not follow the advised trajectory.
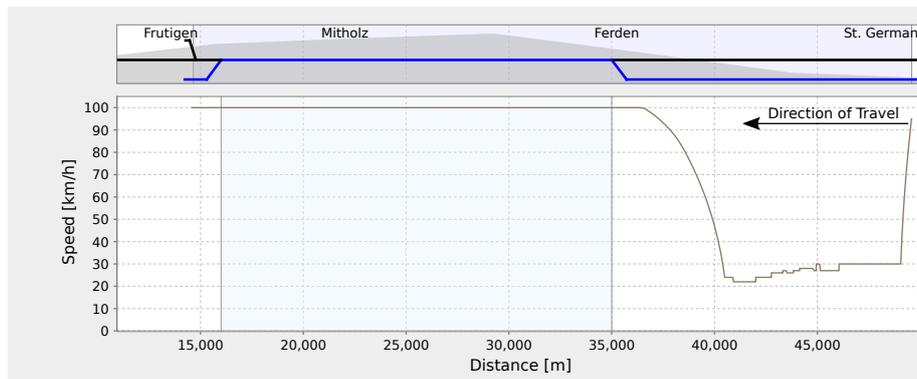
Figure 3.14: Trajectory of the adaptive version of the proposed crossing conflict optimization algorithm. Speed changes during the constant speed traveling phase represent changes in the conflict resolution time forecast.

Chapter 4

Software Architecture

4.1 Overview

4.1.1 Packages and Layers

A graphical overview of the packages and layers used in the system are shown in Figure 4.1. Packages are described in Table 4.1.

Package	Description
User Interface Layer	
web	„Java Server Pages" User Interface: invokes the data collection, the chart visualization and the analysis functions.
charts	visualizes scenarios.
Logic Layer	
collector	extracts the data from the log file and populates the domain layer.
scenarios	simulation of different scenarios.
analysis	aggregates the results from multiple scenarios.
Domain Layer	
domain	stores domain data.
Existing systransis classes	
vehicle dynamics, engine types, etc.	

Table 4.1: Layers and packages used in the developed software.
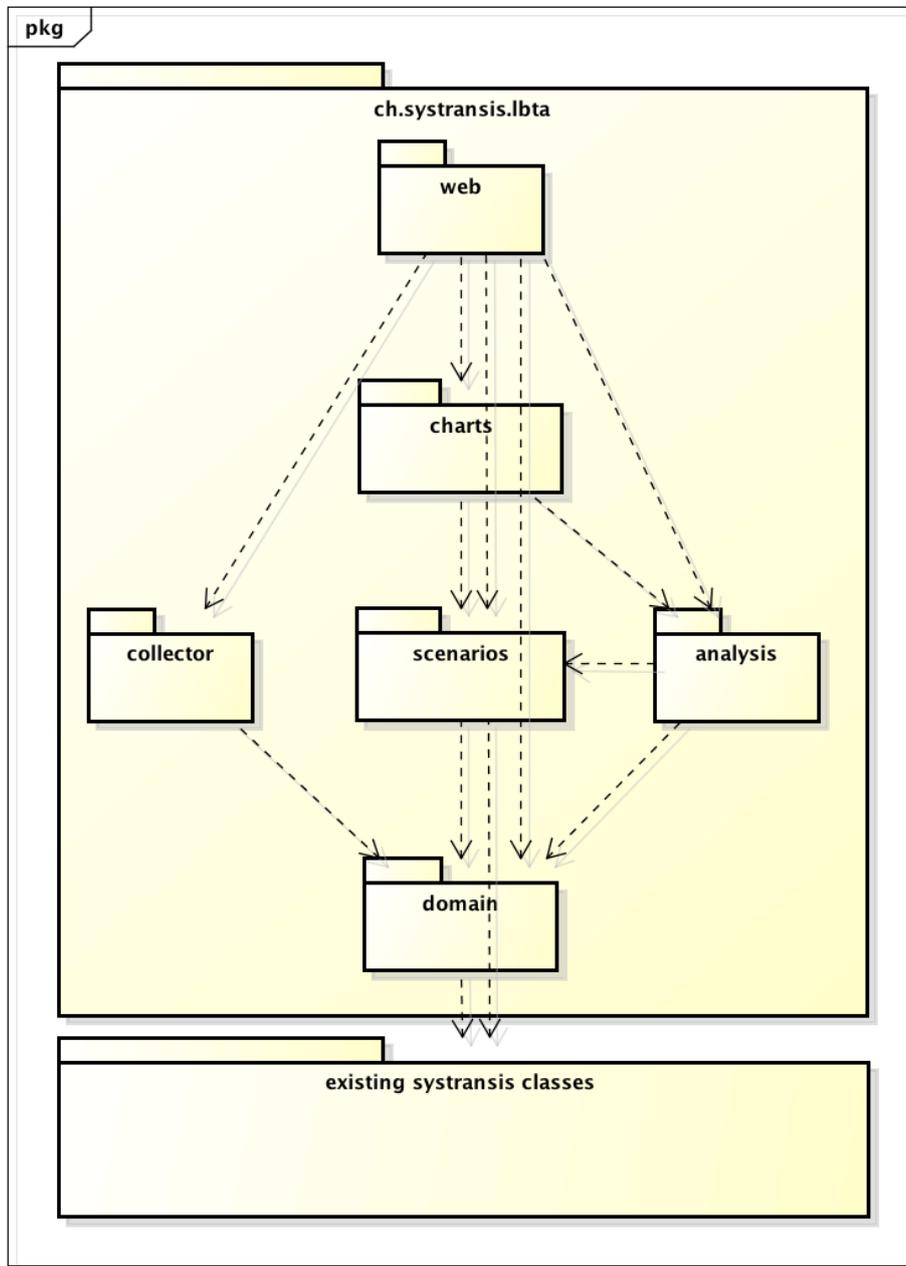
Figure 4.1: Overview of the packages and layers used in the system.

4.2 Domain Layer

An overview of the domain layer is shown in Figure 4.2. The structure resembles the one of the domain model from section 3.1.

4.2.1 Key Classes

Class	Description
`TrainInfoStore`	This class encapsulates a collection of `TrainInfo` instances. It serves as a „database" as it contains a list of references to all `TrainInfo` instances.
`TrainInfo`	This class holds information about a specific train run, such as the travel data, train number, weight etc. Some information is extracted to other `*Info` classes (see bellow).
`AdvisorySpeedInfo`, `MovementAuthorityInfo`, `TrainPositionInfo`	These classes contain collections and provide helper methods to search them. For example, the `TrainPositionInfo` aggregates `TrainPosition` objects and provides a method to find the first `TrainPosition` inside the ETCS area.

Table 4.2: Key classes of the domain layer and their description.
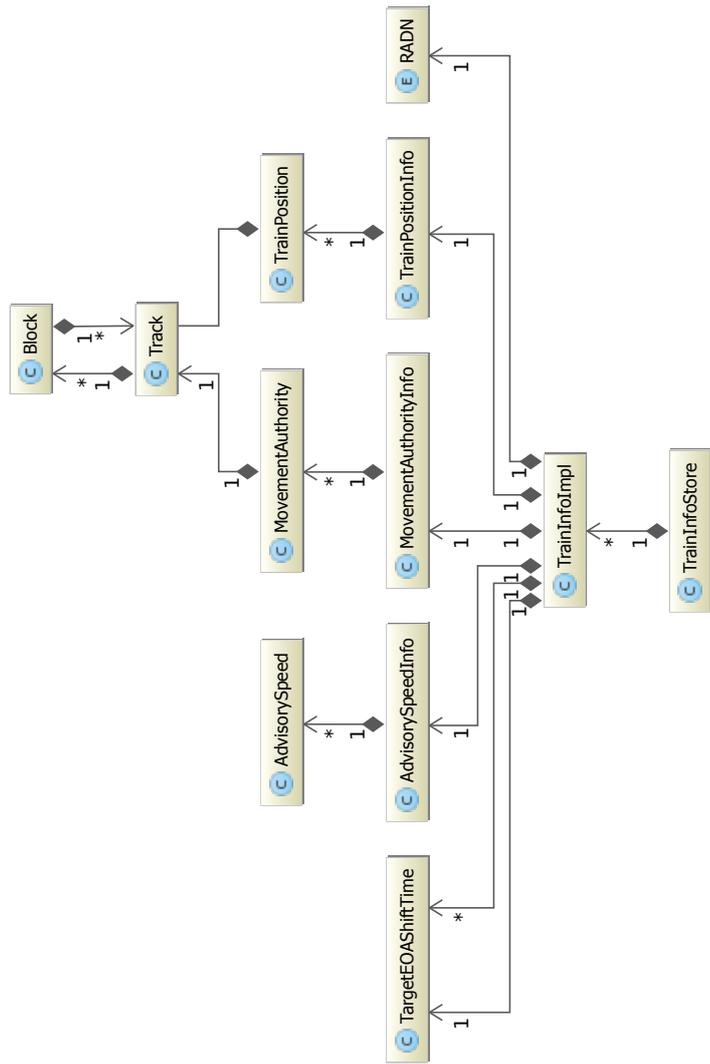
Figure 4.2: UML diagram of the software classes in the domain layer. The `TrainInfoStore` class encapsulates a collection of `TrainInfo` instances. A `TrainInfo` instances represent a train run. Additional details are explained in Table 4.2.

4.3 Business Logic layer

4.3.1 Collector Package

The UML diagram in Figure 4.3 shows the solution implemented for the log file extraction.
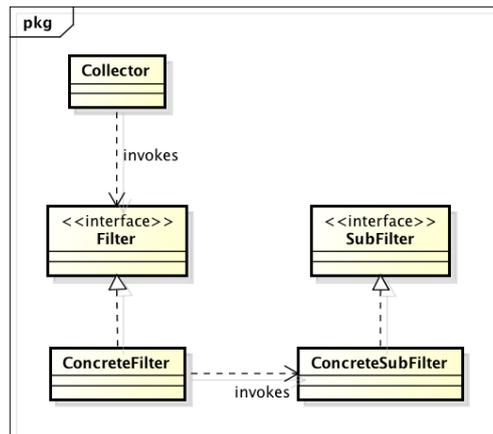
Figure 4.3: The collector UML diagram.

The key interfaces in this package are:

Collector This interface provides methods to invoke a data collection process and access the results using an instance of the `TrainInfoStore` class. During the collection process, this class delegates the processing of log entries to the appropriate `Filter` implementation.

Filter For each log entry type, an implementation of the `Filter` interface exists. They employ a RegEx pattern to extract the important information (e.g. train number, train length etc.) and add it to a `TrainInfo` object.

SubFilter Implementations of the `SubFilter` interface serve as delegates by `Filter` implementations when a log entry contains more than one kind of information. For example the `TrainInfoFilter` delegates the extraction of the „length", „max. speed" etc. to `SubFilter`s.

All filters except the `ConflictFilter` are stateless. The `ConflictFilter` class is statefull, because not every log entry type handled by this filter contains the required information to unambiguously assign the extracted information to the corresponding train.

Sample Filter

The code snipped in Listing 4.1 shows an example of a `SubFilter`. Like all other filters, it makes use of Java's `Pattern` and `Matcher` classes to extract the required data.

```java
public class MaxSpeedFilter implements SubFilter {
    private static final Pattern PATTERN =
        Pattern.compile("<Attribute field=\"19\"\"
        value=\"(\\d+)\"/>");

    @Override
    public void process(String entryString, TrainInfo
        trainInfo) {
        Matcher matcher = PATTERN.matcher(entryString);
        if (matcher.find()) {
            int maxSpeed =
                Integer.parseInt(matcher.group(1));
            trainInfo.setMaxPermittedSpeed(maxSpeed);
        }
    }
}
```

Listing 4.1: Implementation of MaxSpeedFilter class

4.3.2 Collector Package: Query Caching

Problem Description

The log file extraction (described in section 4.3.1) takes approximately 50 seconds of processing time to handle a log period of one hour. This means that 20 minutes are required for one day. This is especially undesirable if users want to analyze the same period multiple times.

Solutions Considered

- The user-friendliest solution is a scheduled task which regularly parses the new log entries incrementally and stores the resulting information into a database. This solution would however introduce constant data duplication. This approach is therefore not desired by the customer.

- Another solution is to cache the results of all log file extractions performed. The benefit is, that only the first user has to wait for the process to finish and all subsequent users can access the same data fast.

Solution

We decided to use the „Ehcache"[1], a light weight open source java based cache. The cache maps time periods to results of the extraction process.

The UML diagram in Figure 4.4 shows how the cache is implemented. The method `getTrainInfo(Date, Date)` on the `CachedCollector` first checks the cache for the requested data. If this results in a cache hit, the method immediately returns the cached result. In case of a cache miss, it invokes the `getTrainInfo(Date, Date)` on the `CollectorImpl` which in turn extracts the data from the log files and stores the results in the cache.

---

[1]`www.ehcache.org`
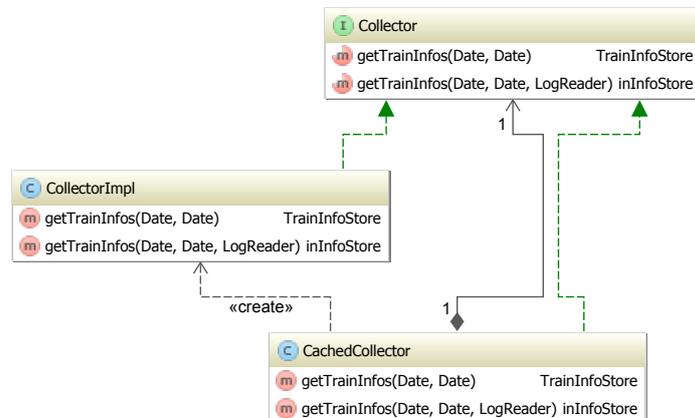
Figure 4.4: The collector proxy UML diagram.

Ehcache Configuration

The Ehcache is easily configurable through an XML file. Our configuration is
shown in Listing 4.2.

```
<cache name="cache"
       maxEntriesLocalHeap="500"
       maxEntriesLocalDisk="1000"
       eternal="true"
       overflowToDisk="true"
       diskPersistent="true">
</cache>
```

Listing 4.2: ehcache.xml

The meaning of the attributes are as folllows:

name an optional name for the CacheManager.

maxEntriesLocalHeap sets the maximum number of objects that will be cre-
ated in memory.

maxEntriesLocalDisk sets the maximum number of objects that will be main-
tained in the DiskStore.

eternal defines whether elements are eternal. If this is set to eternal, timeouts
are ignored and the element never expires.

overflowToDisk denotes whether elements are overflown to disk. If this is set
to true, the local disk is used as a data tier.

diskPersistent enables disk persistance. If this is set to true, the data file and
an index file are stored to the local disk.

4.3.3 Scenario Package

AbstractScenario Class

`AbstractScenario` (Figure 4.5) is the key class of this package and represents a simulation of a train run. Several concrete scenarios such as the `NonOptimizedScenario` extend this class.
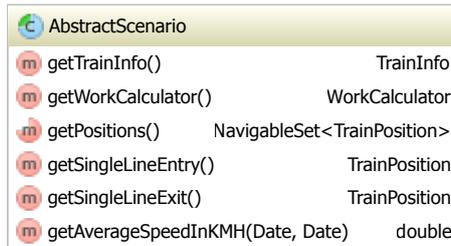
Figure 4.5: UML overview of the AbstractScenario class.

Method	Description
`getTrainInfo()`	returns the `TrainInfo` instance this simulation is based on.
`getPositions()`	gives access to the results of the simulation.
`getWorkCalculator()`	returns an instance of the `WorkCalculator` class from the `analysis` package for the simulation.
other methods	are helper methods to extract information from the simulation results.

Table 4.3: Methods of the AbstractScenario class.

AbstractScenarioFactory Class

The `AbstractScenarioFactory` is used to access concrete scenario factories such as the `NonOptimizedScenarioFactory`. Concrete scenario factories instantiate concrete scenarios (section 4.3.3) based on a `TrainInfo` object.

The concrete scenario factories contain logic to check, wether the scenario can be applied to the `TrainInfo` object passed. For example if `TrainInfo` objects that represent train runs not affected by crossing conflicts are passed, no scenarios can be calculated. In this case the concrete factories always return a `RealityScenario` instance.

Example usage is shown in Listing 4.3.

```
AbstractScenarioFactory factory = new
    AbstractScenarioFactory.NonOptimizedScenarioFactory();
AbstractScenario nonOptimizedScenario =
    factory.createScenario(trainInfo);
```

Listing 4.3: Example Usage of AbstractScenarioFactory

4.3.4 Analysis Package

ScenarioAnalysis Class

The `ScenarioAnalysis` class receives a collation of `TrainInfo` objects in the form of a `TrainInfoStore` object, time period and a `AbstractScenarioFactory` as input. The `ScenarioAnalysis` class performs simulation runs for all `TrainInfo`s using the scenario produced by the factory passed. Based on these results, the class calculates the four capacity indicators previously mentioned in this thesis.

CapacityAnalysis Class

This class aggregates the results from multiple `ScenarioAnalysis` instances. Its primary purpose is to make the results available to the user interface layer.

4.4 User Interface Layer

4.4.1 Background

systransis requested, that the web user interface is implemented using Java Server Pages (JSP), because this is their standard technology. Unfortunately, JSP is not well suited to construct rich user interfaces.

We therefore decided to use JQuery to construct a user interface that features enhanced functionality such as sorting and filtering of tables. The advantage of this approach, is that the JSP parts of the UI can easily be used without JQuery in case *systransis* choses to fall back to a pure JSP interface.

4.4.2 Libraries Used

The user interface is built with Java Server Pages (JSP)[2], JavaServer Pages Standard Tag Library (JSTL)[3], JFreeChart[4], JasperReport[5], DataTables[6], J-Query[7] and JQuery UI[8].

JSP is the main technology, that helps to dynamically create web pages based on HTML. It is licensed under Common Development and Distribution License (CDDL-1.0).

JSTL extends JSP by additional tag libraries for common tasks, such as iterating over elements, conditional execution, different formatters etc. It is licensed under Common Development and Distribution License (CDDL-1.0).

JQuery is a multi-browser Java Script library licensed under MIT License. It is applied instead of plain JavaScript. It helps to create browser independent JavaScript and it is used by DataTables and JQuery UI.

DataTables is a plugin-in for JQuery. It is used in our project for all tables to make them sortable, searchable and visually appealing. DataTables is licensed under GNU General Public License, version 2 and BSD (3-point) style license.

JQuery UI is a Java Script library, that helps building visually appealing web applications and it's also used by DataTables. It is licensed under MIT License.

JFreeChart is a free chart library which is used to create the charts and plots. It is licensed under GNU Lesser General Public Licence (LGPL).
Apache Batik[9] is used to to transform charts and plots to the Scalable Vector Graphics (SVG) format. This library is licensed under Apache License, Version 2.0.

---

[2]http://www.oracle.com/technetwork/java/javaee/jsp/index.html
[3]http://www.oracle.com/technetwork/java/index-jsp-135995.htmll
[4]http://www.jfree.org/jfreechart/
[5]http://community.jaspersoft.com/project/jasperreports-library
[6]http://www.datatables.net/
[7]http://www.jquery.com/
[8]http://www.jqueryui.com/
[9]http://xmlgraphics.apache.org/batik/

JasperReport is a light weight reporting engine. It is used to create the PDF reports. The reports are designed with iReport, a report designer for JasperReports. It is licensed under GNU Lesser General Public Licence (LGPL).

Chapter 5

Simulation and Evaluation

5.1 Benefits of the Proposed Algorithm

- Simulation results show, that the proposed crossing conflict optimization algorithm often results in more precise advisory speed trajectories, as the train dynamics model considers the engines' limitations better.

  An example is illustrated in Figure 5.1.
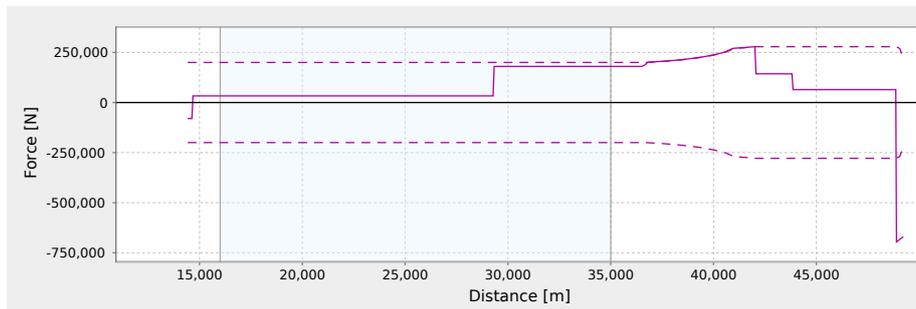
Figure 5.1: $F(x)$ plot of the trajectory calculated by the proposed alternative algorithm. The tractive force required (solid line) does fully exhaust the engine's capabilities (dashed line) during the acceleration phase, but does not exceed them.

5.2 Impact of Advisory Speeds

5.2.1 Simulated Period

In order to analyze the impact of advisory speeds over a longer period of time, the time from the 29th of October to the 18th of November 2012 has been evaluated. During this period, the daily schedule was repeated several times and various special situations were considered. The number of trains operated during this time frame is shown in Table 5.1.

Week	Number of Trains	Number of Trains Affected by Crossing Conflicts[1]	Percentage
First	565	59	10%
Second	613	73	12%
Third	657	80	12%
Sum	**1835**	**212**	**11%**

Table 5.1: Number of train runs and conflicts per week from the 29th of October to the 18th of November 2012.

5.2.2 Impact on Individual Conflicts

Table 5.2 shows how the four considered scenarios perform in conflict situations.

In scenario ↓

	Non Optimized	Reality	AF Optimized	Proposed AF
Non Optimized	-	119	87	50
Reality	45	-	51	24
AF Optimized	88	112	-	74
Proposed AF	86	145	81	-
n of 212 trains cleared the single track section at least 15 seconds earlier than in →	Non Optimized	Reality	AF Optimized	Proposed AF

Table 5.2: Comparison Matrix. Example: In scenario „Proposed AF", 145 of 212 trains cleared the single track section at least 15 seconds earlier than in „Reality". In scenario „Reality", 24 of 212 trains cleared the single track section at least 15 seconds earlier than in „Proposed AF". For 43 trains, the resulting time difference was below 15 seconds $(145 + 24 + 43 = 212)$.

---

[1]Received at least one advisory speed unequal to 0 and unequal to „v max" inside ETCS area for conflict on points W34, W44 or W60.

The following conclusions can be drawn from Table 5.2:

- The proposed AF algorithm with improved vehicle dynamics functions yields the best results, as in these scenarios, most trains can clear the single track section earlier.

  We therefore conclude, that an algorithm such as the one proposed, can even further reduce the traveling time inside the single track section than the existing one.

- The second best result in the comparison matrix is achieved by the „AF optimized" scenario.

- The reasons why the proposed AF algorithm is sometimes outperformed by the „Non Optimized" case can be one of the following:

  - Inaccurate conflict resolution time forecasts might influence the benefit of the optimization negatively.

  - For conflicts on points W34 and W44 at the northern end of the tunnel, it can be observed, that in the „Non Optimized" scenario, the deceleration phase in front of the conflict point is frequently interrupted, as soon as the conflict is resolved. The train then starts to accelerate immediately, which results in a very good performance.

    However, the greedy behavior of accelerating to the maximum possible speed followed by a full stop in front of the conflict point would probably not be followed by the train drivers for distances to the conflict point as short as at the northern end of the tunnel.

    An example is illustrated in Figure 5.2.

- In the „Reality" scenario, drivers often use coasting (see Figure 5.3) for deceleration in conflict situations and therefore require a longer distance to reach the advised speed than expected by the AF software. This causes AF to adapt its speed recommendation continuously to lower advisory speeds, which might reduce the capacity benefits.
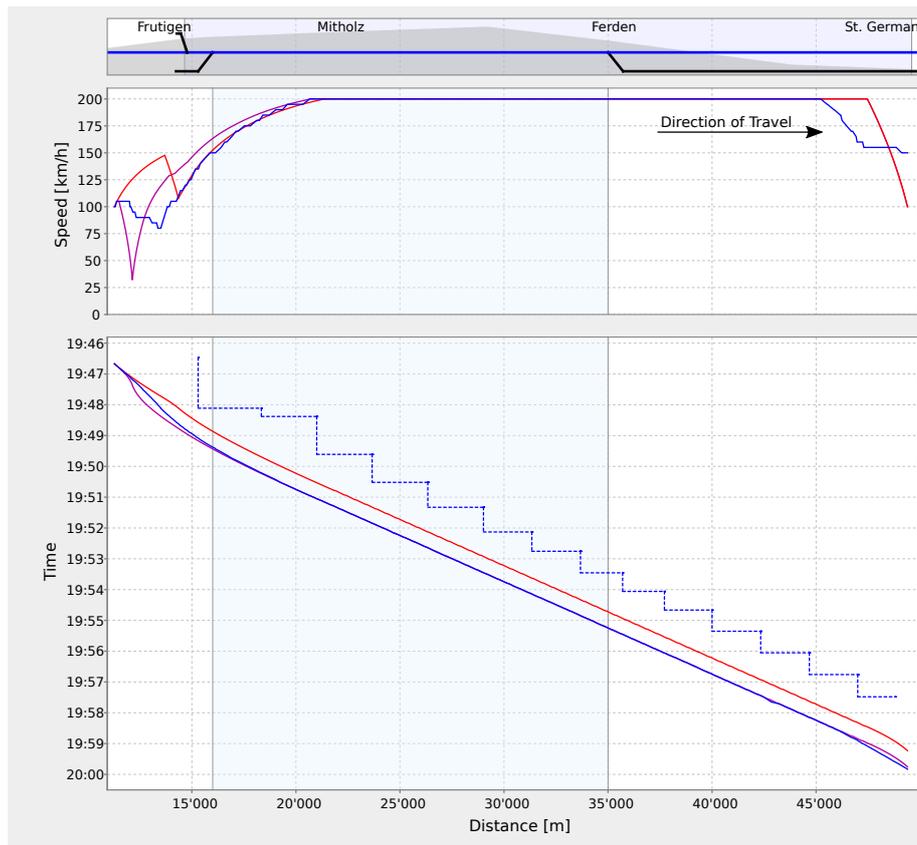
Figure 5.2: Trajectories of a conflict situation at the northern end of the tunnel. The „Non Optimized" scenario (red) performs better than the reality scenario (blue) and the „Proposed AF" scenario (purple). For short distances to the conflict point, it is very unlikely, that drivers would actually behave as greedy as assumed in the „Non Optimized" scenario.

5.2.3 Impact on Overall Capacity and Energy

Table 5.3 shows the capacity indicators and mechanical energy consumption for three weeks analyzed. Our capacity analysis does only take into account the first level of conflicts, as it is an aggregation of the simulation results for each conflict identified in the log file (section 2.2.8). The values are therefore a lower bound. In practice, resolving a conflict in the present has a cascading effect of preventing future conflicts, which in turn avoids even further conflicts.

The „AF optimized" scenario mainly focuses on preventing unplanned stops inside the tunnel and thereby saving train running time. In contrast the „Proposed AF" aims at additional capacity benefits.

Scenario	Number of Trains	Average Speed [km/h]	Stability Range [0; 1]	Heterogenity Range [0; 1]	Mechanical Energy Consumption[2] [kWh]
Non Optimized	1835	149	0.556	0.891	1'496'000
Reality	1835	150	0.554	0.892	1'475'000
AF Optimized	1835	150	0.566	0.890	1'478'000
Proposed AF	1835	150	0.561	0.890	1'486'000

Table 5.3: Capacity Indicators and mechanical energy consumption for the duration analyzed.

The following conclusions can be drawn:

- The comparison of the overall capacity indicators based on the „capacity spider" model only shows a slight increase. This is due to the fact, that the vast majority (approx. 90%) of trains is not affected by a crossing conflict (see Table 5.1) and does therefore not influence the capacity.

- The highest stability indexes occur in the „AF Optimized" and „AF Proposed" scenarios. This indicates, that the advisory speeds result in less occupation time of the single track section.

  The amount of time gained is only a few minutes per day. The impact between the „Non Optimized" and the „Proposed AF" scenario is approximately 2.5 hours over a time period of three weeks.

- „AF Optimized" achieves better stability results than „Proposed AF" due to the optimistic acceleration curves assumed. The faster entry speeds result in a stability gain.

---

[2]Measured from the ETCS entry point to the end of the single track section.

- A clear pattern on the „mechanical energy consumption" can be recognized. The consumption increases in the following order: 1. „Reality" 2. „AF Optimized " 3. „AF Proposed" 4. „Non Optimized". The following conclusions can be drawn:

  – The good result of the „Reality" scenario in terms of the energy consumption can be explained by the energy-efficient driving behavior performed by most drivers in conflict situations. It can be observed that drivers often decelerate to the advisory speed by coasting, whereas all other scenarios include braking curves that require regenerative or even dissipative braking. The larger deceleration distance during coasting will result in less propulsion work required to haul the train at constant speed on the 10‰ ramp in front of point W60. A sample is given in Figure 5.3.
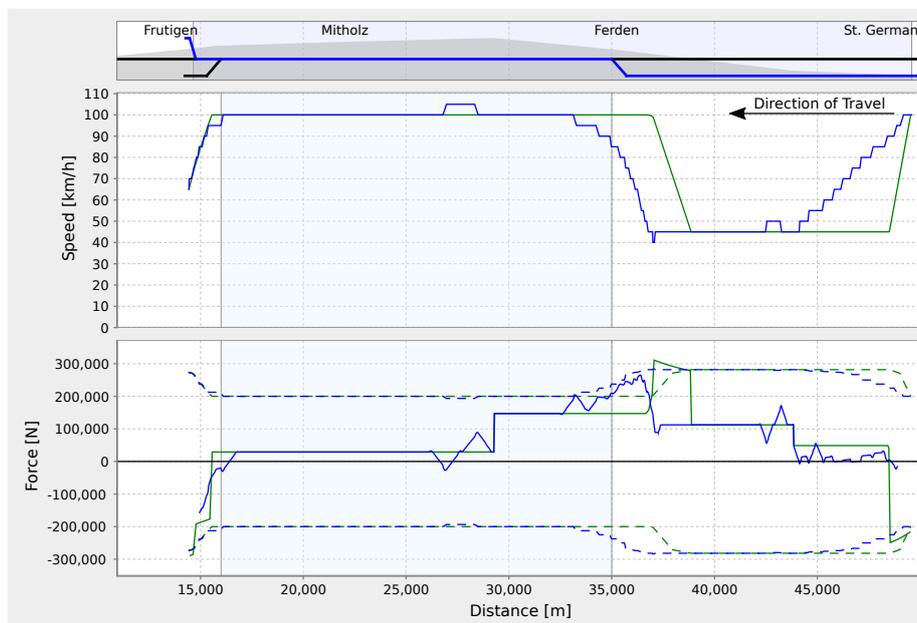
Figure 5.3: Speed and force plot of a freight train (RoLa) hauled by an Re 485 locomotive. The tractive force of approximately zero during the deceleration phase in the „Reality" (blue) case indicates, that the driver used coasting to reduce his speed to the advised 45 km/h. The trajectory proposed by AF (green) would require the driver to apply a braking force and then perform propulsion work to travel the same distance, which most likely is less energy-efficient.

Chapter 6

Conclusions and Results

6.1 Analysis Results

Our work has produced two major results:

- Based on the results of our analysis, we have observed the following improvements during the operation with advisory speeds from AF in comparison with to the estimated operation without advisory speeds:

  - Approximately 25% of the trains are able to leave the single track section at least 15 seconds earlier in the „Reality" scenario than in the assumed „Non Optimized" scenario. These time savings help to reduce delays.

  - The average mechanical work saving for a train affected by a crossing conflict is 12.5% which corresponds to roughly 100 kWh of energy. This result correlates to the previous calculations performed by Mehta et al. [10].

  - During the three weeks analyzed, at least in total 21 megawatt hours of energy[1] were saved. This corresponds to 1.4% of the total traction energy consumed for the tunnel operation within the same period.

  - Train drivers can efficiently avoid stops in the tunnel, achieve a smooth traffic pattern.

- According to our analysis, our proposed enhancements provide the following benefits compared to the current solution:

  - Approximately 65% of the trains affected by conflicts would be able to leave the single track section at least 15 seconds earlier. This helps to even further minimize delays.

  - The trajectories resulting from our proposed algorithm require more energy than current speed profiles performed by drivers due to the steep deceleration curves assumed. However, still 0.7% could be saved in comparison to the non-optimized case.

---

[1]21 megawatt hours of mechanical energy. This estimation does not take into account the energy conversion efficiency ratios for propulsion and regenerative braking.

6.2   Analysis Tool Developed

The analysis tool developed during this thesis features the following highlights:

- More scenarios can be simulated than before. If required, the tool can be extended in the future with additional driving behavior or optimization algorithms to study their impact.

- The usability has been increased thanks to a simplified workflow, an overview screen that lists all trains for a given time period, and the filtering and sorting functions integrated in various places.

- The expressiveness of the trajectories has been greatly increased due to the incorporation of the new force-distance plot and the inclusion of the tunnel's topology and track's gradient.

- The force-distance plot allows interpretation of the driver behavior, as the engine and braking forces involved can be read from the diagram. Moreover, it can be used to check the feasibility of a trajectory and is used to estimate the energy consumption.

- The capacity balance spider can be calculated and visualized for any given period of time.

6.3   Conclusions

This sections summarizes the conclusions from the simulations performed during this projects.

6.3.1   Benefits of Advisory Speeds

- The performed simulation shows that thanks to advisory speeds, trains can clear the single track section in the Lötschberg Base Tunnel earlier, as the added information allows drivers to enter the single track section faster. Speed recommendations can help to reduce delays for individual trains. (section 5.2.2).

- Thanks to additional information available to train drivers in the form of advisory speeds and in combination with the drivers' ecological driving behavior, efficient energy usage can be observed in the „Reality" scenario. (section 5.2.3).

- The simulation showed clear improvements for individual trains affected by conflicts, but the overall capacity gain based on the „capacity spider model" is relatively small. This is due to the fact, that the vast majority of trains (90%) is not affected by a crossing conflict (see Table 5.3) in this case study and does therefore not influence the capacity much (section 5.2.3). However, it is to foresee that the train traffic system will have more potential in capacity gain for dense railway networks.

6.3.2 Suggested Improvements

- We conclude that a detailed vehicle dynamics model is particularly beneficial for the crossing conflict optimization algorithm in terms of increasing capacity. In many conflict cases this can reduce the time spent in the single track section. The algorithm developed during this thesis demonstrates this added optimization benefits. It can be seen that the resulting acceleration curves match the engines' limitations better. (section 5.1).

- We recommend that the optimization software should advise train drivers to decelerate to the advised speed by coasting. This is especially interesting in the Lötschberg Base Tunnel, as the track in front of points before the single track section is going uphill. We anticipate the following benefits:

  - The trajectories proposed to the train drivers will better match their ecological driving behavior.
  - Coasting will result in less energy consumption, as no energy is lost during either regenerative or dissipative braking.
  - This behavior is most likely beneficial in terms of vehicle and infrastructure wear minimization.
  - The travel comfort is increased due to a gentle deceleration.

6.4 Future Work

This section lists possible enhancements to the developed software and ideas for future areas of research.

6.4.1 General

- It is desirable, that the speed and position information is logged with higher precision. For details, see section 3.5.4.

6.4.2 Crossing Conflict Advisory Speed Algorithm

- It would be interesting to compare the performance and results of the algorithm when reimplemented using symbolic computation in Mathematica or similar software. This step would include reimplementing the vehicle dynamics functions.

- The impact on the energy consumption, when drivers are explicitly advised to brake to only brake to the advisory speed by using regenerative braking only, could be studied. However, such an algorithm would lead to lower advisory speeds.

- As an alternative to the point above, driver could be advised to decelerate to the advisory speed by coasting.

6.4.3 Driving Simulation

- As described in section 3.4 the simulation of the braking process could be more detailed.

- Currently, the maximum speed allowed of 180 km/h on point W60 in diverging position is ignored in the simulation.

- Temporary speed restrictions („Langsamfahrstellen") are ignored in the simulation.

- In the current simulation, the speed outside of the tunnel is set to 100 km/h for all trains and every tunnel exits. The maximum speeds are listed in the „R 30131 Streckentabellen RADN" document. To look up the speed applicable, a train's „Bremsreihe" must be known.

6.4.4 Energy Calculation

- To enhance the accuracy of the energy calculation, it would be desirable to use exact effort curves for the regenerative braking instead of the approximation described in section 3.5.1 / Equation 3.4.

- Calculating the electrical energy consumption using the energy conversion efficiency ratios as described in section 3.5.3.

Appendix A

Walkthrough

In this chapter we present the main screens of the analysis tool developed during this thesis. Some key features are highlighted.

The data shown here is fictional.
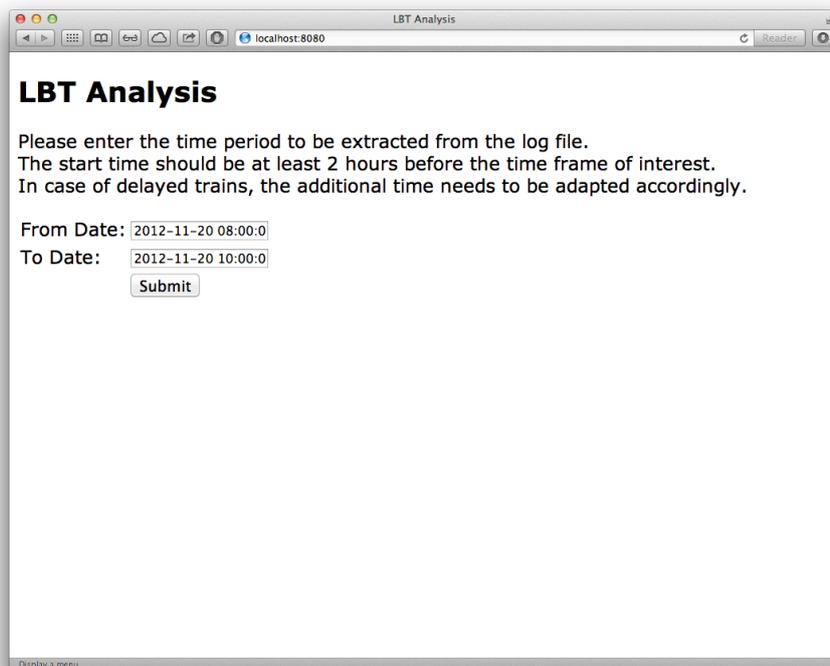
Figure A.1: This is the first screen presented to the user after opening the web application in a browser. The user is prompted to enter a time frame for analyzation.

Figure A.2: After the log extraction process has finished, the user sees an overview of all trains found in the analyzed time frame. The table can be sorted and filtered. In this example the user has entered „SIM" into the search field to search for Simplon-Inter-Modal trains.

Figure A.3: By clicking on a train number on the screen in Figure A.2, the user gets an overview of the train parameters such as engine types and train weight.
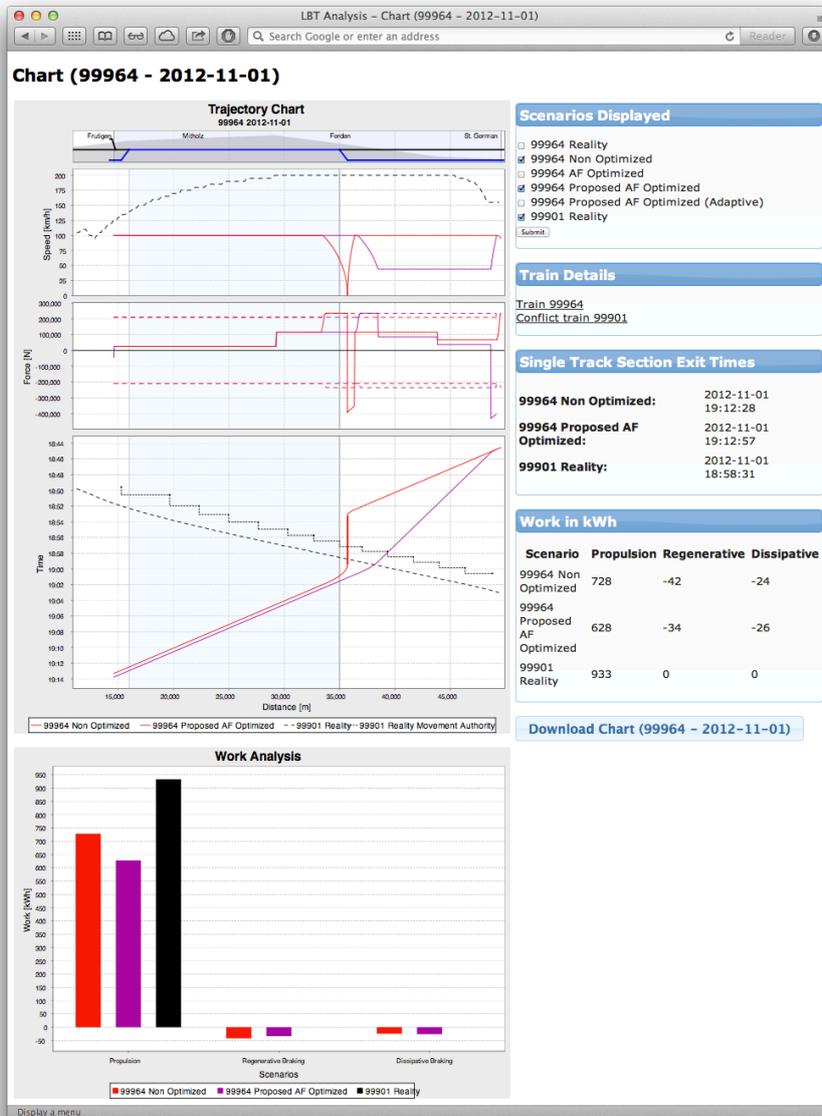
Figure A.4: The chart view for a conflict features the following highlights: 1) the combined speed-, force- and time-distance plot, 2) a check list to define which scenarios are displayed, 3) a bar chart that illustrates the propulsion, regenerative braking and dissipative braking work.
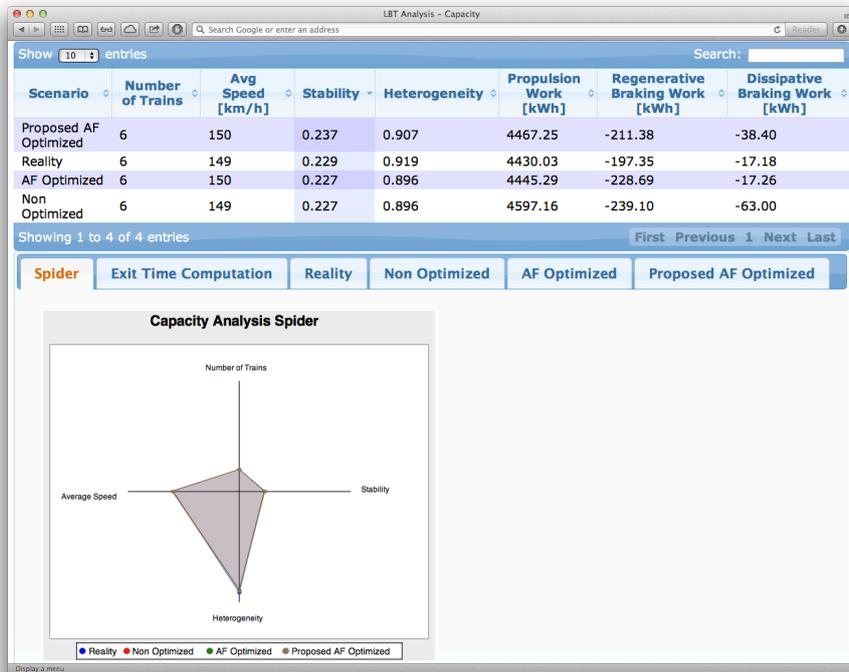
Figure A.5: This screen shows the results of the capacity analysis in a table and a spider chart for a user specified time interval. Further, the sum of the propulsion, regenerative braking and dissipative braking work is shown.
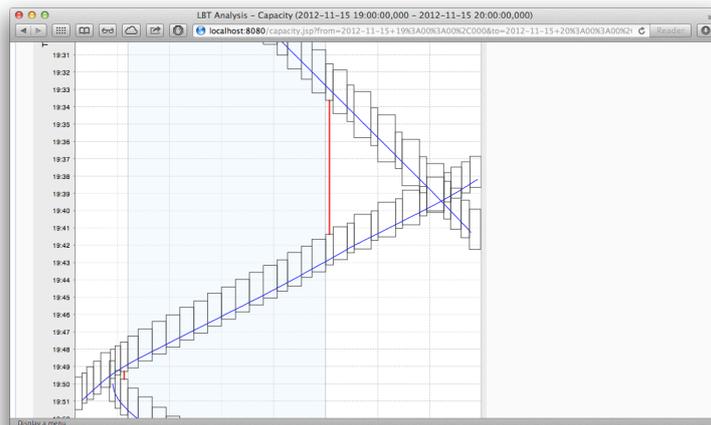
Figure A.6: Using the screen shown in Figure A.6, the blocking stair for every scenario can be displayed.

Glossary and Abbreviations

AF *Automatic Functions* - AF *Automatikfunktionen*
  Train traffic control system that automates part of the operation in the Lötschberg base tunnel.

BEP *Brake Entrance Point* - BEP *Brems-Einsatz-Punkt*

Block - *(Strecken-) Block*
  A block is a concept used in train protection systems. A line is divided into blocks which can only be occupied exclusively by one train at a time.

Blocking Stair - *Sperrzeitentreppe*
  Graphical representation of a train's occupation times for each block along a line. For an example, see Figure 3.11.

BTP *Braking Target Point* - BZP *Brems-Ziel-Punkt*

Coasting - *Ausrollen*
  Performing a natural deceleration when the power is removed.

EOA *End Of (Movement) Authority* - *Ende der (Fahr-)Erlaubnis*
  Position where the the movement authority ends. The EOA position might be before or at the supervised location (SvL) used as braking target point for enforced brakings by ETCS's safety functions. For details see, Maschek [9].

ERTMS *European Rail Traffic Management System*
  An initiative backed by the European Union to enhance interoperability of railway equipment in Europe. ERTMS's two main components are the ETCS „European Train Control System" and GSM-R „Global System for Mobile Communications - Railway".

ETCS *European Train Control System*
  A standardized european signalling, control and train protection system aimed to replace the incompatible national safety systems.

EW IV Push–pull train - *EW IV Pendelzug*
  Push–pull train consisting of „SBB-Einheitswagen IV" passenger coaches and control car.

Headway Time
  The time difference between two trains passing a specific point.

LBT *Lötschberg Base Tunnel* - LBT *Lötschberg-Basistunnel*
The Lötschberg Base Tunnel is a 34.6 km long railway tunnel connecting Frutigen (Berne) and Raron (Valais).

RADN - RADN *Zugreihe*
Trains on Switzerland's standard gauge network are categorized into the four train classes R, A, D and N. Based on a train's class and the train's braking capability, the maximum speed for each section of the network can be looked up.

Rolling highway - RoLa *Rollende Landstrasse*
Combined transport train conveying road trucks.

UIC *Union internationale des chemins de fer*
International Union of Railways

Unaccompanied combined transport - UKV *Unbegleiteter Kombinierter Verkehr*

Picture Credits

- Figure 1.2 is a modified version from Wikimedia Commons [17] which has been released into the public domain by its author Bukk.

- Figure 1.5 is a modified version from Wikimedia Commons [16] which has been made available under the Creative Commons CC0 1.0 Universal Public Domain Dedication by its author Marten de Vries.

- The Rh 1116 picture in Figure 3.5 is copyrighted by and used with permission from RAILCOLOR.net [13].

Bibliography

[1] *UIC Code 406 - Capacity.* UIC Union Internationale des Chemins de fer, 2004.

[2] *Schweizerische Fahrdienstvorschriften (FDV) - R 300.14 - Bremsen*, 2012.

[3] ETCS Standbericht 2012. Technical report, Bundesamt für Verkehr BAV, 2012.

[4] *BLS Geschäftsbericht 2011.* BLS AG, 2012.

[5] Gert Heister, Jörg Kuhnke, Carsten Lindstedt, Roswitha Pomp, Thomas Schill, Thorsten Schaer, Stephan Schmidt, Norbert Wagner, and Wolfgang Weber. *Eisenbahnbetriebstechnologie.* DB Fachbuch. Bahn Fachverlag, 2006.

[6] Daniel Hürlimann. *Objektorientierte Modellierung von Infrastrukturelementen und Betriebsvorgängen im Eisenbahnwesen.* PhD thesis, ETH Zürich, 2001.

[7] Alex Landex. *Methods to estimate railway capacity and passenger delays.* PhD thesis, Danmarks Tekniske Universitet, 2008.

[8] Marco Lüthi. *Improving the efficiency of heavily used railway networks through integrated real-time rescheduling.* PhD thesis, ETH Zürich, 2009.

[9] Ulrich Maschek. *Sicherung des Schienenverkehrs: Grundlagen und Planung der Leit- und Sicherungstechnik.* Springer Vieweg, 2012.

[10] Farhad Mehta, C Rößiger, and Markus Montigel. Latent energy savings due to the innovative use of advisory speeds to avoid occupation conflicts. *systransis Ltd.*, 2010.

[11] Markus Montigel. *Modellierung und Gewährleistung von Abhängigkeiten in Eisenbahnsicherungsanlagen.* PhD thesis, ETH Zürich, 1994.

[12] Jörn Pachl. *Systemtechnik des Schienenverkehrs: Bahnbetrieb planen, steuern und sichern.* Vieweg+Teubner, 6th edition, 2012.

[13] RAILCOLOR.net. Siemens 20922 - Locomotive visual, 2008. URL http://www.railcolor.net/index.php?nav=1405043&lang=1&id=27438&action=portrait. [Online; accessed 18-May-2013].

[14] Hans G. Wägli. *Bahnprofil Schweiz CH+.* AS Verlag, 2012.

[15] Dietrich Wende. *Fahrdynamik des Schienenverkehrs*. Vieweg+Teubner, 2003.

[16] Wikimedia Commons. ERTMS DMI.svg, 2013. URL `http://commons.wikimedia.org/wiki/File:ERTMS_DMI.svg`. [Online; accessed 01-May-2013].

[17] Wikimedia Commons. ETCS L2 w RBC de.svg, 2013. URL `http://commons.wikimedia.org/wiki/File:ETCS_L2_w_RBC_de.svg`. [Online; accessed 01-May-2013].

[18] Martin Wyss. Five years experiences of optimizing a long single track line. Presentation at IT13.rail conference, [Online; accessed 02-May-2013]. URL `http://www.it13rail.ch/download`.