FH AACHEN
UNIVERSITY OF APPLIED SCIENCES

FH Aachen

Fachbereich
Elektrotechnik und Informationstechnik

Masterarbeit

Der Titel der Arbeit ist zweizeilig

Vorname Nachname
Matr.-Nr.: 123456

Referent:	**Prof. Dr-Ing. ...**
Korreferent:	**Prof. Dr.-Ing. ...**

April 6, 2020

Erklärung

Ich versichere hiermit, dass ich die vorliegende Arbeit selbstständig verfasst und keine anderen als die im Literaturverzeichnis angegebenen Quellen benutzt habe.

Stellen, die wörtlich oder sinngemäß aus veröffentlichten oder noch nicht veröffentlichten Quellen entnommen sind, sind als solche kenntlich gemacht.

Die Zeichnungen oder Abbildungen in dieser Arbeit sind von mir selbst erstellt worden oder mit einem entsprechenden Quellennachweis versehen.

Diese Arbeit ist in gleicher oder ähnlicher Form noch bei keiner anderen Prüfungsbehörde eingereicht worden.

Aachen, April 6, 2020

Danksagung

Danke.

Contents

Abstract Modern day railway system operations require automated train control mechanisms, e.g. European Train Control System ETCS, to maximize efficiency, which is often times limited by outdated infrastructure, as well as safety of operations. One way to achieve this is by lowering the required distance between two trains on the same track, which in turn demands a reliable method of predicting the braking distance at any given moment.

While determination of the necessary braking curves is feasible for a limited number of train formations, the large diversity of vehicles in freight operations poses an issue. One approach for a solution would be using Big Data, which would be able to process the required amounts of data to calculate reliable braking curves even for freight operations.

The problem here is there is simply not enough data available since freight trains usually don't have the sensory equipment needed. To circumvent that obstacle, this work proposes generation of artificial data via white box modeling to be then used in further big data operations.

Introduction

Introduction This section describes the background and motivation of the research (Sect. 1.1), the problem to be addressed (Sect. 1.2) and the proposed solution (Sect. 1.3)

1.1 Background

1.1.1 Railway Vehicle Operations

1.1.2 Train Protection Systems

1.1.3 Braking Curves

1.2 Problem

As has been shown, to predict the braking behavior of trains, readings of wagons and locomotives are needed. Unfortunately, freight vehicles do not currently posses the sensory equipment that would be necessary to obtain such data in an adequate quantity and quality, especially in regards to *big data processing*. Although it has been proposed to equip freight wagons accordingly **<TODO: ref zu wagon4.0>**, it will be years before enough rolling stock has been retrofitted as to make it possible to obtain the desired data.

1.3 Solution

This work proposes to circumvent the problem described above by creation of an artificial data set. The set must replicate the actual distribution of braking behavior as close as possible. It is therefore necessary to first create a model encompassing the braking process of a freight train. This model will be discussed in depth in chapter 3. It can then, once finished, be also used to generate the data set by simulation. This process will be discussed in chapter 4.

As real life operations would yield very high quantities of data, the simulation output must be stored in a data structure which is suitable for big data processing. This structure will also be discussed in chapter 4.

Fundamentals of Railway Vehicle Engineering

Modeling of Train Operations

Introduction As has been noted in 1.3, it is necessary to model the braking process of freight trains. All modeling work has been performed with Matlab Simulink.

3.1 Initial Model

The initial model to be expanded upon describes a single braking process. It's sole input, apart from some constants, is pressure over time, meaning a distinct value ranging between 5 and 3.5 bar for every timestamp. For visualization, please refer to B. Let's take a look at the whole model first.

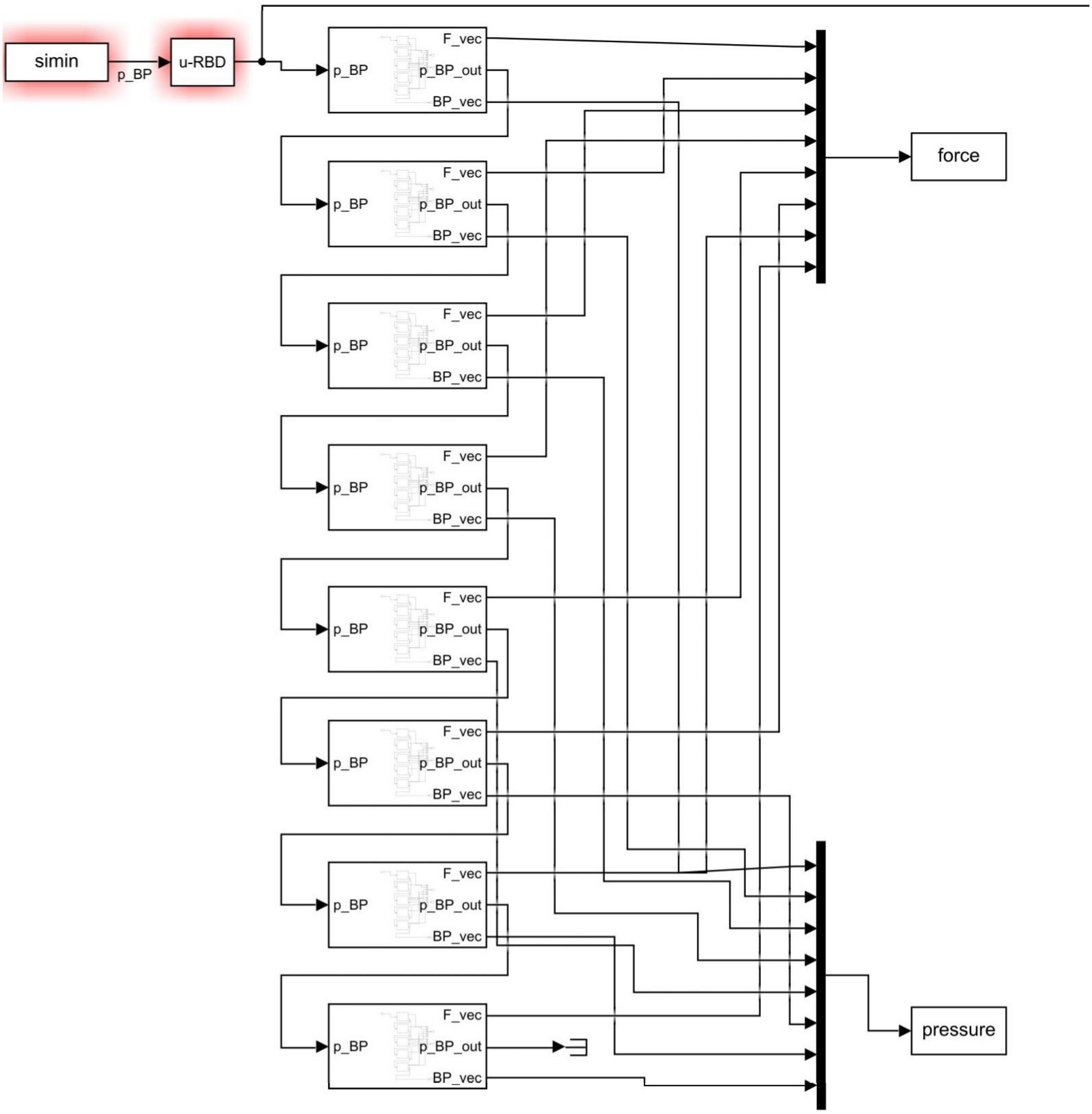

Figure 3.1: Initial Model

Here we see a model of a freight train of fixed length, consisting of 40 wagons, which are, for better readability, further condensed to subsystems of five wagons each, so there are eight of these subsystems. They are interconnected via braking pipe, which is also the sole input to each system. Outputs are braking pressure and braking force. We will take a look at the actual wagon model next.

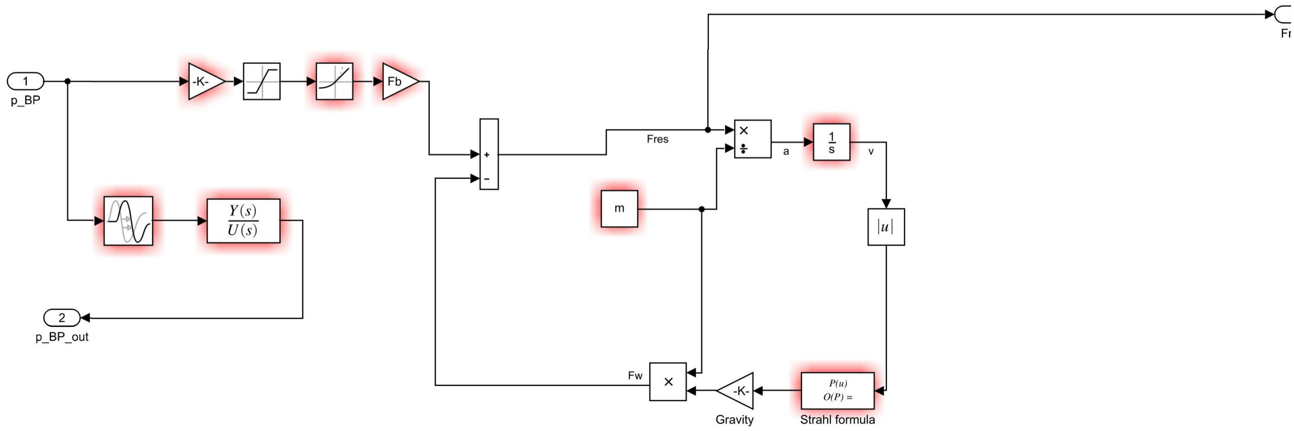

Figure 3.2: Initial Model - Wagon

Above is the initial wagon model. NOTE: All 40 wagon models are identical here. This will be addressed in section 3.2. It consists of three main components.

In the upper left corner is the input, which is the current pressure in the braking pipe. In the lower left corner, the propagation delay of the braking pipe is calculated. This is done by **<TODO: >**. Top center describes the calculation of the actual braking force, which is achieved by **<TODO: >**. Finally, the **<TODO: Formulieren: Fahrzeugwiderstand>**.

3.2 Model Expansion

This initial model is however not of sufficient detail. Where it merely describes one single braking process, we need to simulate a whole ride, with alternating phases of braking and accelerating. For that purpose, the simulation input has to be adjusted accordingly. Where previously it was only one braking process, using braking pressure as input was the obvious choice, whereas now the idea is to use a kind of track profile, which shall describe the maximum allowed velocity over time, of a notional track. For visualization, please refer to B. The simulation then only needs to brake or accelerate depending on train velocity versus maximum velocity at the current time.

Accordingly, the first expansion step is to create a mechanism to control the train so to speak. For this purpose, the system simply checks for each timestamp whether the current velocity of the train is greater than the maximum allowed velocity at the current time, according to simulation input. If this is the case, a braking pressure is applied to the pipe, scaling with the difference between v_{max} and v_{real}, v_{dif}. This means the higher v_{dif} is, the more braking pressure gets applied. This more or less covers the braking part of the system.

The model however also needs a component for acceleration. To simplify things, the logic here is that if the train is not braking, it is accelerating, which actually works out pretty well. To accelerate, a traction force is applied, which also scales with v_{dif}, so the higher v_{dif}, the higher the applied traction force.

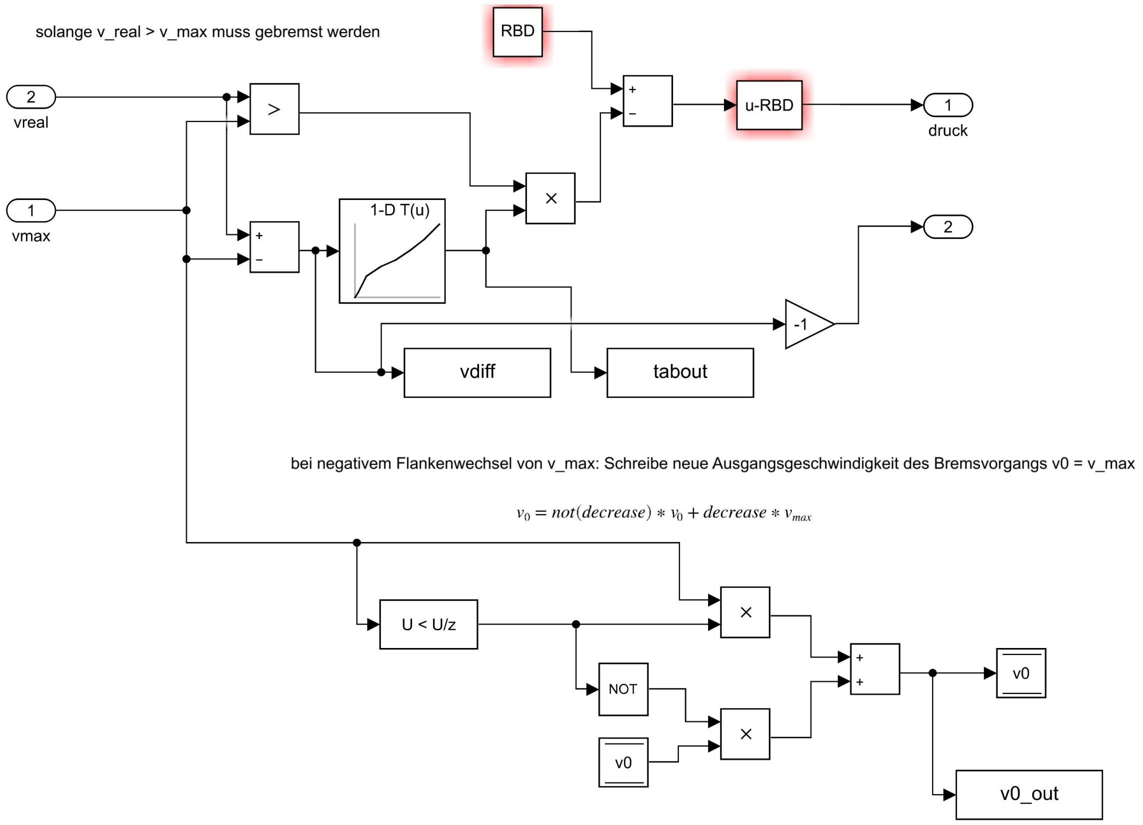

Figure 3.3: Expanded Model - Pressure Calculation

Depicted above is the system which determines braking pressure to apply. It calculates v_{dif} by subtracting v_{max} from v_{real}, which is then fed into a one-dimensional lookup table. The table is a sampled representation of a function with fixed breakpoints, mapping one function value to each breakpoint, like so

$$H(n) = \begin{cases} 0.1 & \text{if } n = 1 \\ 0.7 & \text{if } n = 15 \\ 0.8 & \text{if } n = 20 \\ .. \end{cases} \tag{3.1}$$

where n is the breakpoints of v_{dif}. Since the pressure should only be applied if v_{real} is greater than v_{max}, the ultimate result follows the logic of the following equation

$$P(n,t) = H(n) * (v_{real}(t) > v_{max}(t)) \tag{3.2}$$

where $v_{real}(t)$ is train velocity over time, $v_{max}(t)$ is maximum velocity over time, and $v_{real}(t) > v_{max}(t)$ is either 1 or 0.

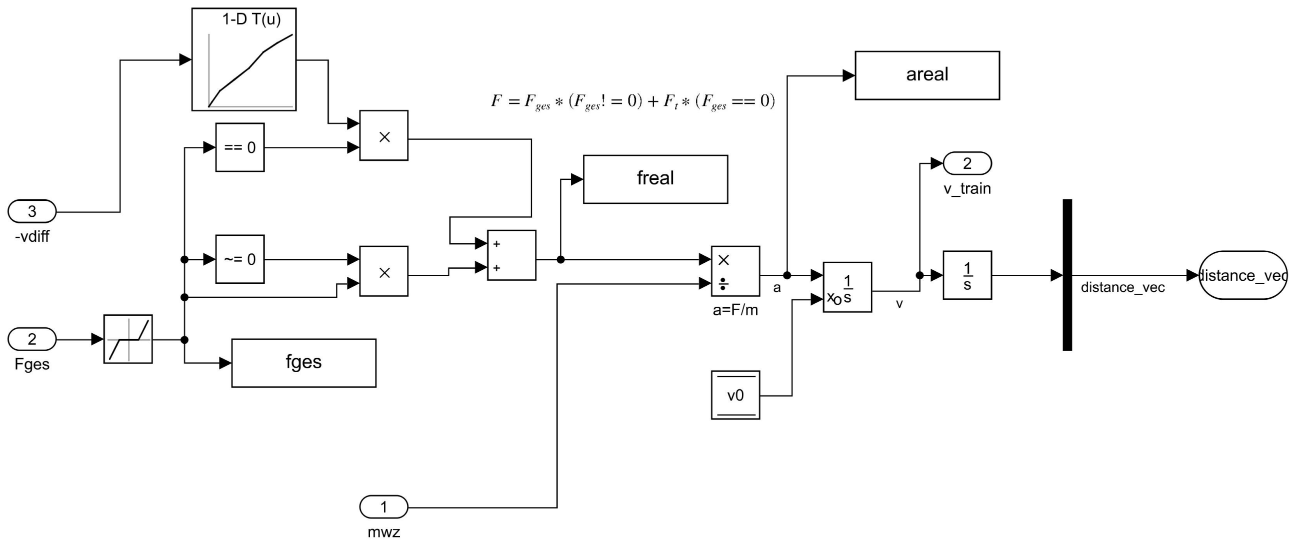

Figure 3.4: Expanded Model - Traction Force Calculation

The above system determines the traction force to apply. As has been discussed earlier, this works like a simple bang-bang controller. The design is very similar to the braking pressure system: v_{dif} is again fed into a one-dimensional lookup table, which outputs different values for traction force accordingly. The higher v_{dif} is, the higher the traction force to apply. It is then added to the current braking force, however only either traction or braking force is at any given time positive while the other is zero, which is achieved by the equations

$$f(n, t) = H(n) * (F_B(t) == 0) \tag{3.3}$$

where n is v_{dif}, $H(n)$ is the lookup table function (see equation 3.1), $F_B(t)$ is the braking force over time, and

$$g(t) = F_B(t) * (F_B(t) \neq 0) \tag{3.4}$$

where $F_B(t)$ is the braking force over time, so we have

$$F(n, t) = f(n, t) + g(t) \tag{3.5}$$

where F is the actual force over time, either braking or traction.

F is then used to calculate acceleration. According to Newton's Second Law,

$$F = m * a \tag{3.6}$$

Accordingly, acceleration is

$$a = F(n,t)/m \tag{3.7}$$

where m is the accumulated mass of all wagons and $F(n,t)$ relates to equation 3.5. The acceleration is then used to calculate the velocity by integrating a in relation to v_0, which is the initial velocity of the current braking or acceleration process **<TODO: überprüfen..>**. Integration of v in turn allows calculation of the traveled distance.

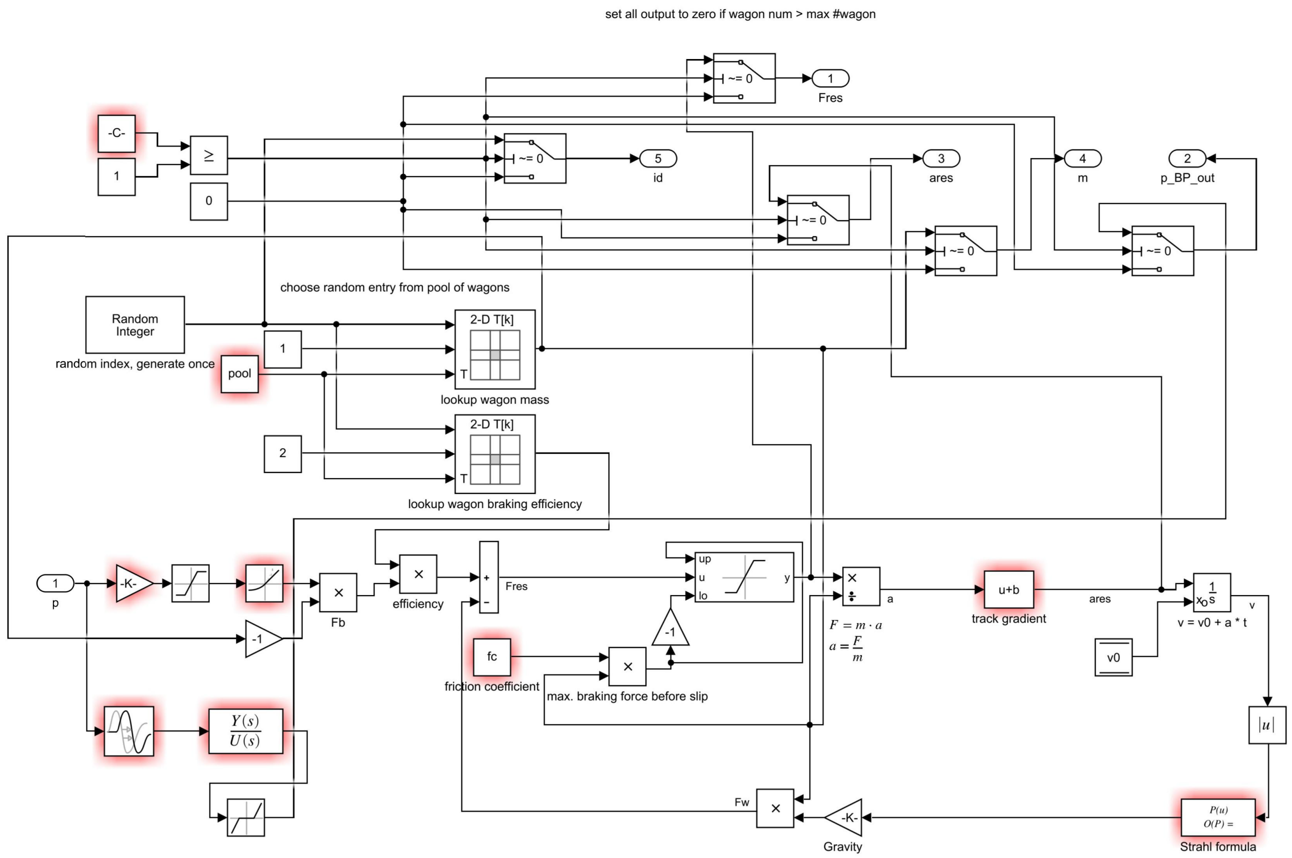

Figure 3.5: Expanded Model - Wagon

The last subsystem is the actual wagon. We will take a look at the largely unchanged elements first. The sole input is still the braking pressure on the brake pipe. Simulation of the propagation delay has also remained the same as before.

One new addition is a pool of different wagons. Whereas before all 40 were distinguishable only by their position, they are now assigned with different parameters. To that end, a pool of 500 wagons has been randomly generated via python script, where each wagon has a unique ID, as well as randomly generated mass and braking efficiency. In actual simulation, up to

40 of these 500 are, currently by generation of random indices, selected and their properties used accordingly. It would also be possible to determine the wagon ids to be used beforehand, instead of choosing randomly.

Another requirement was to make the number of wagons variable. In the initial model, the modeled train had a fixed number of 40 wagons, therefore also 40 wagon subsystems. Unfortunately, simulink offers no way to disable certain subsystems dynamically, but only by manually turning them off via model explorer, which would be unfeasible for such a large number of simulations. To circumvent this issue, output gets disabled for all unwanted wagons. For a simulation of a train of 20 wagons, the first 20 remain untouched, while the latter 20 produce no output and therefore also have no impact on the overall simulation. The turning off is achieved by simple switches; each wagon subsystem has a unique index from one to forty. If the index is greater than the specified number of wagons, all switches are turned to output zero.

3.3 Further Expansion

Data Generation

4.1 Data Structure

4.2 Analysis of generated Data

Performance Analysis

Conclusion

List of Figures

List of Tables

Quellcode

1. Source 1
2. Source 2

Data visualization

<TODO: Visualisierungen einfügen>

Initial model - simulation input Expanded model - simulation input