FP1 MOTIONAL
European Rail Network and Mobility Management

Deliverable D8.2
Developed methods and models for evaluating feedback loops between planning and operations

Project acronym:	MOTIONAL
Starting date:	01/04/2023
Duration (in months):	12
Call (part) identifier:	HORIZON-ER-JU-2022-01
Grant agreement no:	101101973
Due date of deliverable:	30/11/2024
Actual submission date:	27/11/2024
Code	FP1-WP08-D-TRV-001-04
Responsible/Author:	TRV
Dissemination level:	PU
Status:	External review version

Reviewed: (yes/no)
yes

This project has received funding from the European Union's Horizon Europe research and innovation programme under Grant Agreement No 101101973.

Document history		
Revision	*Date*	*Description*
1	29-02-2024	First issue: 50% check version
2	19-07-2024	Second issue: 90% check version for WP8 internal review
3	07-10-2024	Third issue: internal review version
4	27-11-2024	Fourth issue: external review version

Report contributors		
Name	Beneficiary Short Name	Details of contribution
Camiel Simons	NS	Main editor. Task 8.3.1 lead and responsible for Section 5, chapter of demonstration 9.2
Emma Solinen	TRV	Responsible for summary, background, objective. Contributions connected to demonstration 9.1.
Axel Valentin	SNCF	Task 8.3.2 lead and responsible for Section 6
Emma Ferrand	SNCF	Contributor to task 8.3.2
Hans Sipilä	TRV A.E. KTH	Chapter of demonstration 9.1
Carl-William Palmqvist	TRV A.E. LU	Chapter of demonstration 9.1
Isabel Meseguer Hijós	CAF	Chapter of demonstration 9.8
Carmen Ramos Prieto	INDRA	Chapter of demonstration 9.4
Enrique Gómez González	INDRA	Chapter of demonstration 9.4
Ricardo Saldanha	NS A.E. SISCOG	Chapter of demonstration 9.2
Rafael Mendes Borges	NS	Chapter of demonstration 9.2
Thomas van den Berg	ProRail	General sections in Chapter 5
Pablo Ciaurriz	CEIT	WP8 Reviewer
Mariano Martínez	ADIF	WP8 Internal Reviewer
Marco Nanni	Alstom	FA1 Internal Reviewer

Disclaimer

The information in this document is provided "as is", and no guarantee or warranty is given that the information is fit for any particular purpose. The content of this document reflects only the author's view – the Joint Undertaking is not responsible for any use that may be made of the information it contains. The users use the information at their sole risk and liability.

Co-funded by the European Union. Views and opinion expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or [name of the granting authority]. Neither the European Union nor the granting authority can be held responsible for them.

Table of Contents

List of Figures

List of Tables

1. Executive Summary

The present document constitutes the Deliverable D8.2 "Developed methods and models for evaluating feedback loops between planning and operations" in the framework of the Flagship Project FP1-MOTIONAL as described in the EU-RAIL MAWP. The overall objective of this task is to develop and improve railway traffic simulation methods and models to improve feedback loops between planning and operations. With better modelling of the traffic, we can achieve more reliable and effective evaluation of capacity and punctuality, which can be used to increase the quality of the plan.

In this deliverable, development is presented related to four demonstrations, i.e., demonstrations 9.1–9.4. Also, demonstration 9.8 is presented briefly, even though the main developments there are presented in Deliverable D8.3 (FP1–MOTIONAL,2025).

Results of the Deliverable D8.2 are:

- Presentation of PROTON, a simulation tool usable for large networks. PROTON will be demonstrated for a large Swedish case and calibrated and validated with real-world disturbance data.
- Method for calibration and validation of simulation models. These results will be used to calibrate and validate PROTON.
- Improved method for processing historical data to create delay distributions, including small disturbances and calibration of background noise.
- Introduction of a simulation tool for improved feedback between crew plan and operation. With this tool, it is possible to simulate the combination of a timetable, rolling stock and crew to evaluate the robustness of a crew plan.
- A simulation tool that can be used to evaluate different infrastructure layouts and timetables. The goal is to improve a timetable based on feedback loop between C-DAS data and simulation tool/TMS.

Main conclusions are that there is a need for further developments within railway traffic simulation modelling. The more we know about setup of simulations, the faster we can do efficient analyses. With the development presented in this deliverable, we are able to perform better traffic and capacity analyses and give more relevant feedback to planning. Improved input data and stochastic disturbances that are presented in the deliverable will make the results more reliable.

2. Abbreviations and acronyms

Abbreviation / Acronym	Description
ASFA	“Signal Announcement and Automatic Braking”. Railway signalling system used primarily in Spain designed to improve safety and efficiency in railway operations, which monitors and controls the adherence to signals and train speeds. Its main purpose is to prevent accidents such as collisions or derailments in case the driver fails to respond correctly to the signals.
ATO	Automatic Train Operation
ATP	Automatic Train Protection
C-DAS	Connected Driver Advisory System
CCC	Central Composite Circumscribed
EBI	Emergency Brake Intervention supervision limit
ERTMS	Signalling and operation management system encompassing ETCS for the Control Command, ATO for the Automatic Train Operation and FRMCS and/or GSM-R for voice and data communication. FRMCS and/or GSM-R are/is used as radio bearer for ETCS and ATO. ERTMS is aimed at harmonizing railway operations across Europe. It is a more advanced and comprehensive system that allows for greater automation and interoperability between different countries.
ETCS L1, L2, L3	The different ERTMS/ETCS application levels are a way to express the possible operating relationships between track and train. Level definitions are related to the trackside equipment used, to the way the trackside information reaches the on-board units and to which functions are processed in the trackside and in the on-board equipment respectively.
HL3	Hybrid Level 3
IM	Infrastructure Manager
GUI	Graphical User Interface
SQL	Structured Query Language, a language used for database querying
RMSE	Root Mean Square Error
TE	Technical Enabler
TMS	Traffic Management System
TPS	Train Planning System
WP	Work Package

3. Background

The present document constitutes the Deliverable D8.2 “Developed methods and models for evaluating feedback loops between planning and operations” in the framework of the Flagship Project FP1-MOTIONAL as described in the EU-RAIL Multi Annual Working Plan.

Rail traffic simulation is a powerful tool for analysing and optimising rail transportation systems. It involves the use of software programs that simulate the simultaneous movement of trains and other rail vehicles through a network, considering various factors such as train characteristics, train schedules, track capacities, signalling systems, traffic control and other variables. The tools can be used to simulate certain processes, subsystems of the environment or a complete use case. The functionality provides the ability to simulate the interaction of trains amongst each other, with some kind of traffic control.

Simulation models are essential to be able to estimate the capacity of a given infrastructure and the feasibility and robustness of a timetable. They offer powerful means for analysing and optimising rail transportation systems. By using simulation models to test different scenarios and strategies, railway operators and planners can improve the reliability and efficiency of their networks. In order to achieve these simulation models, continuous data improvement by feedback of historical information available for analysis is needed. This will increase the reliability of the railway network simulation both in optimal and degraded states, while always focusing on improving the simulation models.

The Deliverable D8.2 is related to the Technical Enabler 5 “Improved rail traffic simulation models”. In the Deliverable D3.1 “Mapping against scope, specification of technical enablers, high-level use cases, high-level requirements, high-level design for demonstrators in WPs 4-9” (see, (FP1–MOTIONAL, 2024a), the following high-level requirement are listed for TE5:

- Models that can simulate traffic in a large network.
- Support to improve feedback loops between planning and operation and timetable evaluation with historical data and improved delay distributions.
- Improved stochastic simulation models to increase the precision in punctuality prediction.
- Models with integrated crew scheduling.

In this deliverable these high-level requirements are assessed.

4. Objective/Aim

The overall objective of this task (Task 8.3) is to develop and improve railway traffic simulation methods and models to improve feedback loops between planning and operations. The improvements are in line with the high-level requirements listed in the Deliverable D3.1 (see, FP1–MOTIONAL, 2024a) for Technical Enabler 5. With better modelling of the traffic, we can achieve more reliable and effective capacity and punctuality evaluations, which can be used to increase the quality of the timetable and resource planning.

5. Development of simulation environment for micro and macro simulation

Simulation models are important in providing feedback to the operations. Deliverable D8.1 (see, FP1–MOTIONAL, 2024b) gives background information on different feedback loops and the different topics in the loops. There, the role of simulation has been explained generally. In this chapter, we zoom in on the simulation models, and how they are used for the feedback loops. We give an introduction to each model and describe what developments are being done in WP8 and WP9.

Broadly, one can divide simulations into two aggregation levels (microscopic and macroscopic), two levels of internal synergy between scheduling process and running operation (synchronous and asynchronous), and two levels of randomisation of train runs and delays (stochastic and deterministic) (Hansen & Pachl, 2014; Botte & D'Acierno, 2018; Pachl, 2024).

The difference between the microscopic and macroscopic level is as follows. At the microscopic level, one models each signal, track and switch, as well as the signalling and train protection systems, at an accuracy level representative of reality. One then simulates exact train paths through the network. By contrast, macroscopic simulation models define infrastructure at a higher aggregation level. They may specify stations as singular points without specifying a precise track plan or signalling system within or between them, and simulate trains solely defined by the state they are in (e.g., waiting to depart, departed, occupying a line between stations). We illustrate the difference between microscopic and macroscopic models in Figure 1. While microscopic models more closely represent reality, they require extensive preparation work and are more computationally complex.

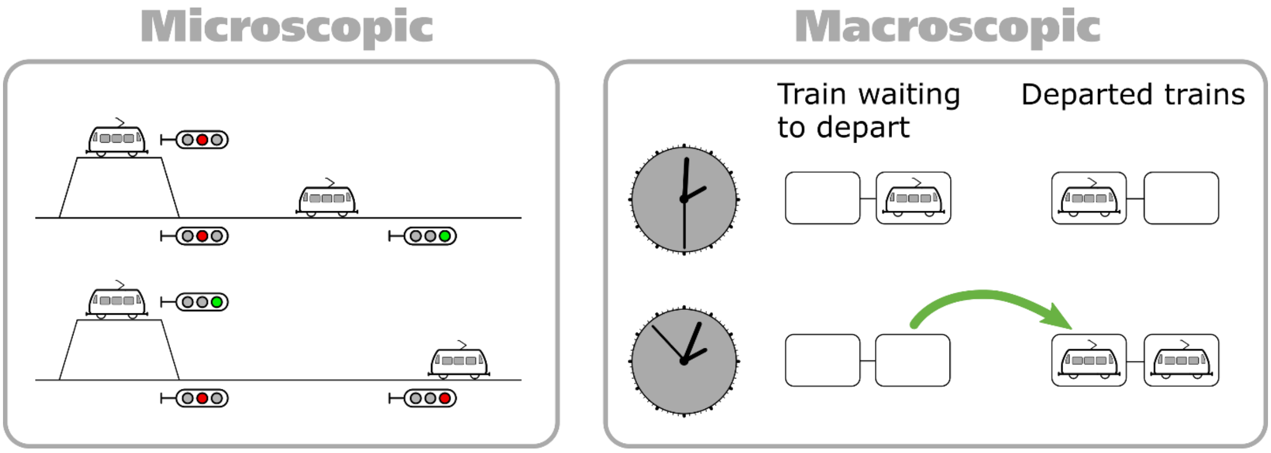


Figure 1 Microscopic and macroscopic simulation: differences in infrastructure and running operation modelling

Synchronous and asynchronous (or event-based) simulations differ as follows (Hansen & Pachl, 2014). In asynchronous simulations, the scheduling process and the train running operation are simulated separately. When scheduled train paths converge, the simulation automatically resolves conflicts based on train priorities and other predefined scheduling principles. This process generates a train running order that prevents any conflicts. Subsequently, the running operation simulation runs the trains in this order. By contrast, a synchronous simulation combines scheduling and running operation in real-time, without a parallel scheduling simulation that resolves conflicts. Trains are run according to their initial schedule, after which conflicts may occur naturally. Hence, synchronous models are vulnerable to deadlocks and asynchronous models are not. However, it also follows, that synchronous models are a better approximation of reality.

Capacity simulations can be deterministic or stochastic (Hansen & Pachl, 2014). In deterministic simulations the user specifies exact train paths and initial delays manually. Oppositely, in stochastic simulations, train paths and delays follow some predetermined, often empirically motivated probability distribution. Since the values pulled from this probability distribution are random, one needs to simulate the network multiple times to get a reliable estimate of railway performance. While stochastic simulations approximate reality better, they also require more computational power and are more vulnerable to deadlocks, especially when there is a microscopic aggregation level.

A sizeable number of railway simulation software packages exists nowadays, as outlined in Table 6-1 of D8.1 (FP1–MOTIONAL, 2024b). Each software package has their own strengths and weaknesses, in accordance with the multitude of simulation choices described above. The operator or planner chooses between simulation categories and corresponding software tools based on which of them are most appropriate to answer their research question. Borndörfer et al. (2018) recommend synchronous, microscopic, stochastic simulation to most accurately represent reality. However, following D8.1 Figure 6-1, this may not always be computationally feasible, and certain research questions may still be satisfyingly answered when choosing asynchronous, macroscopic or deterministic simulation methods. In microscopic tools, the level-of detail is high and therefore microscopic tools are often used for detailed timetable and traffic analyses and evaluations. Macroscopic models make use of less details, aggregated data and simplifications and they can be used for a variety of purposes, still traffic and timetable evaluations are the main applications. Due to shorter calculation times and less need for information, they can be used for larger networks or situations in the future.

We will explain the use and development of both types of models in the following paragraphs. By means of the demonstrations, see Table 1, all developments are described.

Table 1. Demonstrations that are presented in this deliverable.

Demonstration	Short description	Leading partner
Demo 9.1	Simulate large networks, calibration and validation methodology of simulation model, mainly regarding finding primary delay distribution input (from historical data)	TRV A.E. KTH TRV A.E. LU
Demo 9.2	Demonstrate a method to evaluate the robustness of a crew plan by a new simulation tool. The simulation focuses on delay propagation between trains by shared crew members.	NSR A.E. SISCOG
Demo 9.3	Demonstrate a method for processing the historical data and implement the delay distribution into RailSys for stochastic models	SNCF
Demo 9.4	Simulate how the timetable behaves with different topology networks	INDRA
Demo 9.8	Create mixed operational plans taking into consideration the hour of the day or the area where the track is placed	CAF

5.1. Demonstration 9.1: Feedback loop from simulation to planning for large scale networks

Railway traffic creates dependencies between trains running for long times and at different lines. To get a complete evaluation of the traffic, IMs need to be able to capture all these dependencies and perform stochastic traffic simulations in large networks. Microscopic models are at a high level-of-detail, which makes it complicated and time consuming to simulate traffic in large networks. There is a need for a model with lower level-of-detail, that can handle large networks/datasets but still give reliable results.

Microscopic models include a detailed description of the track layout. They describe tracks, switches, track speeds, signals and signal system parameters. Train movements are modelled regarding acceleration and braking characteristics. In other words, a microscopic model has the potential to model at least some parts of a railway system in a realistic way. Naturally, microscopic models require detailed data about the infrastructure, to some extent the timetable and additional simulation parameters and settings.

Synchronous train simulation models can simulate all events in a network in short time steps (time sequences). Asynchronous models initiate trains successively according to priority and work quite similar to scheduling (Pachl, 2011). A general drawback with synchronous microscopic railway simulation tools is the possible occurrence of deadlocks. This means that a situation occurs in which a few trains cannot continue along their path at all, because they are blocked by another train. Even though the dispatching functionality (logic) in microscopic simulation tools can avoid simple types of deadlocks, e.g., sending two trains into a single-track line from opposite direction, complex infrastructure areas (such as large stations) and bidirectional operation (such as single-track lines) are prone to deadlocks. The probability of deadlocks thus depends on the simulated network (area), timetable and imposed perturbations (delays) and their effects in the simulation. To some extent, the probability of deadlocks can be reduced by, for example, restricting alternative routing through stations. However, this may mean that capabilities used in real situations are not possible in the simulation. Restrictions in possible routing alternatives need to be balanced between the desire for realistic behaviours in simulations and at the same time reduce the probability of deadlocks.

In a macroscopic simulation model, the level of detail is reduced to a relatively large extent, i.e., the level of aggregation is increased, to provide possibilities to simulate larger networks and in short time. The idea is not to replace microsimulation but to use macrosimulation in applications where microsimulation is practically difficult, time-consuming or even impossible to use. TRV together with KTH will demonstrate the use of the existing macroscopic railway simulation tool PROTON for a large network in a Swedish use case.

The macroscopic tool that we work with is PROTON. The development of PROTON started within Shift2Rail by DB Analytics (Deutsche Bahn). The original objective was to develop a macroscopic simulation tool that could, in a computationally efficient way, handle large networks (Zinser et al. 2019). DB has continued developing the tool after Shift2Rail ended, and it has been extended to handle microscopic modelling of infrastructure and train movements.

PROTON has been used in Sweden since 2019 for case specific applications related to research projects, for example in Shift2Rail projects PLASA2 and FR8RAIL II/III. A comparative case study is presented in Johansson et al. 2022. Simulations were performed with both macroscopic PROTON and microscopic RailSys, and the focus was to study the effect on punctuality by letting freight trains depart ahead of schedule compared to not allowing this. Another focus was also to compare the simulated punctualities from both tools, given that the same input delay distributions were used in both setups. In FR8RAIL II D3.4 (2021) scenarios involving restrictions in infrastructure, speed reductions, closed line tracks etc., were simulated in RailSys and PROTON. Simulation results from RailSys were used as a means of verifying that the selected use cases could also be represented in PROTON.

A methodological framework that, among other things, consists of different scripts have been developed to produce the necessary input data from various sources for PROTON simulations and process output data for result evaluations. For example, timetable input is produced either from exported timetables from RailSys projects or from TRV's previous timetable planning tool, TrainPlan. In principle, timetable data from other tools can be used, if the necessary data exists, it is only a matter of converting the data into a format that PROTON reads. Other data can also be converted, like infrastructure, delay distributions, train type information and other related data required by the simulation.

The intention is to use the macroscopic part of PROTON in WP8 and WP9, not the microscopic.

Work is ongoing to establish an IT framework within TRV for the use of PROTON, with possibilities for external organisations outside of TRV IT (such as KTH) to access relevant areas and run simulations on TRV servers. In addition to the work that will take place within FP1, TRV and KTH have regular meetings regarding the IT framework with respect to PROTON.

The aim, with Demonstration 9.1 is to be able to simulate a large network, preferrable the whole Swedish network. This task is closely related to the description in Section 6.1 *Demonstration 9.1: Analysing historical data to create primary delay distributions,* which deals with developing methodology for generating and calibration of input delay distributions from historical data for use in simulations. In connection with this, the framework for using PROTON in Sweden will be further developed and structured to enable and facilitate the use of the tool in different relevant applications for TRV.

There is no inherent value in simulating a large network if, for a certain question at issue, it is sufficient to simulate a significantly smaller part of the network. On the other hand, one can think of questions where it can be an advantage to be able to simulate a large network at once and, for example, not have to divide simulations into sub-networks. One example is to simulate an annual timetable or selected periods of an annual timetable for the whole network. Preferably, there would be at least two timetable alternatives. Simulation can then be used to get an idea of any differences between these regarding expected operational performance, for example punctuality. However, available timetables are needed to be able to simulate the traffic. PROTON, like other simulation tools, needs a timetable as input, it is not designed to generate timetables.

Simulation of networks, as opposed to one or a few lines, provides opportunities to evaluate networks effects. For example, how disturbances propagate between parts of a network via the trains that run longer distances. Another example is if there are different reductions in the track infrastructure, longer speed reductions, single-track operation on double-track sections due to maintenance or other reasons etc. These types of reductions can take place simultaneously in different parts of a network and can create significant delays, that will propagate in a network.

Figure 2 describes the framework for PROTON use in Sweden. Once again, PROTON does not create input data by itself, nor can it be created with help of PROTON. In some other simulation tools, such as in RailSys, a user can design a timetable if all necessary data exists (infrastructure, vehicle data etc.) in the tool, by creating train runs and adjusting them both in tabular form and graphically. PROTON is a pure simulator which requires consistent input data in correct formats to run. Besides PROTON itself, a central function is the data conversion of various data and formats into formats required by PROTON. So far, timetables have been converted from either RailSys data or TrainPlan which is the former timetable planning tool used in Sweden. TRV has traditionally imported annual timetables into their RailSys model, both for doing timetable feasibility evaluations, but also for being able to distribute the RailSys model including the annual timetable(s) to other entities doing work either for or related to TRV in various projects. The direct conversion from TrainPlan to PROTON (without using RailSys as an intermediate step) was created to speed up the process of creating input data to PROTON for various geographical areas, days or periods.

The macroscopic infrastructure has previously been defined manually. A script package is currently under development within WP8, which will generate the necessary macroscopic infrastructure attributes by processing microscopic infrastructure from RailSys. This means that the whole Swedish network, in the national RailSys model, can be converted into PROTON macroscopic infrastructure in one step. Taking into consideration that infrastructure will be generated from RailSys, it will also be easier to use RailSys timetables since, coming from the same model, they will map well to the infrastructure and the need of making additional adjustments, either manually or by script handling, can be minimized.

The boxes which relate to preparing sets of delay distributions are essential, realistic delay modelling is required for results to be credible. When it comes to demonstration 9.1, the methodological framework and development for preparing these is described in Section 6.1 *Demonstration 9.1: Analysing historical data to create primary delay distributions*.

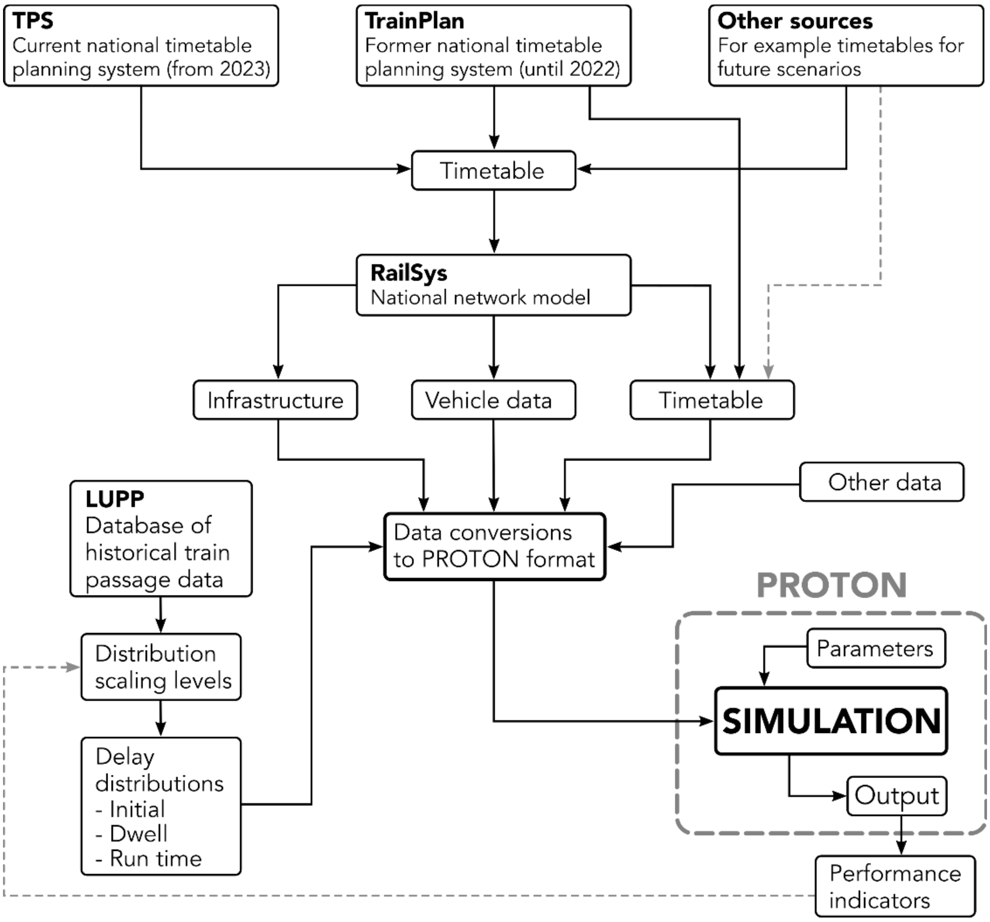
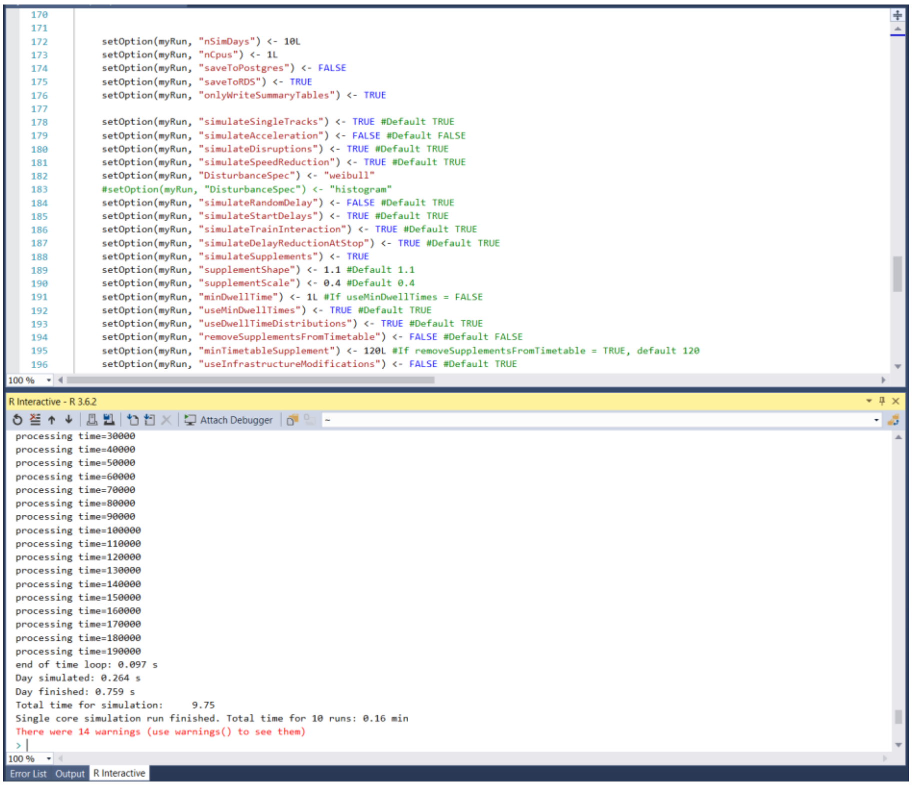

TPS
Current national timetable planning system (from 2023)
TrainPlan
Former national timetable planning system (until 2022)
Other sources
For example timetables for future scenarios
Timetable
RailSys
National network model
Infrastructure
Vehicle data
Timetable
Other data
LUPP
Database of historical train passage data
Data conversions to PROTON format
PROTON
Distribution scaling levels
Parameters
SIMULATION
Delay distributions
- Initial
- Dwell
- Run time
Output
Performance indicators

Figure 2. Framework for PROTON use in Sweden. The actual tool PROTON is indicated inside the dashed line. All other processes, e.g., data preparation and computing performance indicators, take place outside of the tool.

The Swedish setup of PROTON does not have any sort of GUI or other visualization features. Input data needs to be generated, and after a simulation, output data needs to be processed for computing relevant measures (punctuality, average delays, deviations, etc.). The current Swedish PROTON setup is configured within a Microsoft Visual Studio environment. PROTON is started by running R-scripts, although the simulation core is programmed in C++. Figure 3 shows an example of how it looks like.



```


    setOption(myRun, "nSimDays") <- 10L
    setOption(myRun, "nCpus") <- 1L
    setOption(myRun, "saveToPostgres") <- FALSE
    setOption(myRun, "saveToRDS") <- TRUE
    setOption(myRun, "onlyWriteSummaryTables") <- TRUE

    setOption(myRun, "simulateSingleTracks") <- TRUE #Default TRUE
    setOption(myRun, "simulateAcceleration") <- FALSE #Default FALSE
    setOption(myRun, "simulateDisruptions") <- TRUE #Default TRUE
    setOption(myRun, "simulateSpeedReduction") <- TRUE #Default TRUE
    setOption(myRun, "DisturbanceSpec") <- "weibull"
    #setOption(myRun, "DisturbanceSpec") <- "histogram"
    setOption(myRun, "simulateRandomDelay") <- FALSE #Default TRUE
    setOption(myRun, "simulateStartDelays") <- TRUE #Default TRUE
    setOption(myRun, "simulateTrainInteraction") <- TRUE #Default TRUE
    setOption(myRun, "simulateDelayReductionAtStop") <- TRUE #Default TRUE
    setOption(myRun, "simulateSupplements") <- TRUE
    setOption(myRun, "supplementShape") <- 1.1 #Default 1.1
    setOption(myRun, "supplementScale") <- 0.4 #Default 0.4
    setOption(myRun, "minDwellTime") <- 1L #If useMinDwellTimes = FALSE
    setOption(myRun, "useMinDwellTimes") <- TRUE #Default TRUE
    setOption(myRun, "useDwellTimeDistributions") <- TRUE #Default TRUE
    setOption(myRun, "removeSupplementsFromTimetable") <- FALSE #Default FALSE
    setOption(myRun, "minTimetableSupplement") <- 120L #If removeSupplementsFromTimetable = TRUE, default 120
    setOption(myRun, "useInfrastructureModifications") <- FALSE #Default TRUE
```

```
processing time=30000
processing time=40000
processing time=50000
processing time=60000
processing time=70000
processing time=80000
processing time=90000
processing time=100000
processing time=110000
processing time=120000
processing time=130000
processing time=140000
processing time=150000
processing time=160000
processing time=170000
processing time=180000
processing time=190000
end of time loop: 0.097 s
Day simulated: 0.264 s
Day finished: 0.759 s
Total time for simulation:     9.75
Single core simulation run finished. Total time for 10 runs: 0.16 min
There were 14 warnings (use warnings() to see them)
> 
```

Figure 3. Example of a finished PROTON run.

Besides computing numbers, it is many times useful to be able to produce a traditional graphical timetable which may show both scheduled and simulated train paths. This illustrates how the simulation tool handles trains in different situations and can be used to check how a change in simulation parameters will affect the behaviour by comparing different simulations. Figure 4 shows an example of a graphical timetable generated from PROTON data.

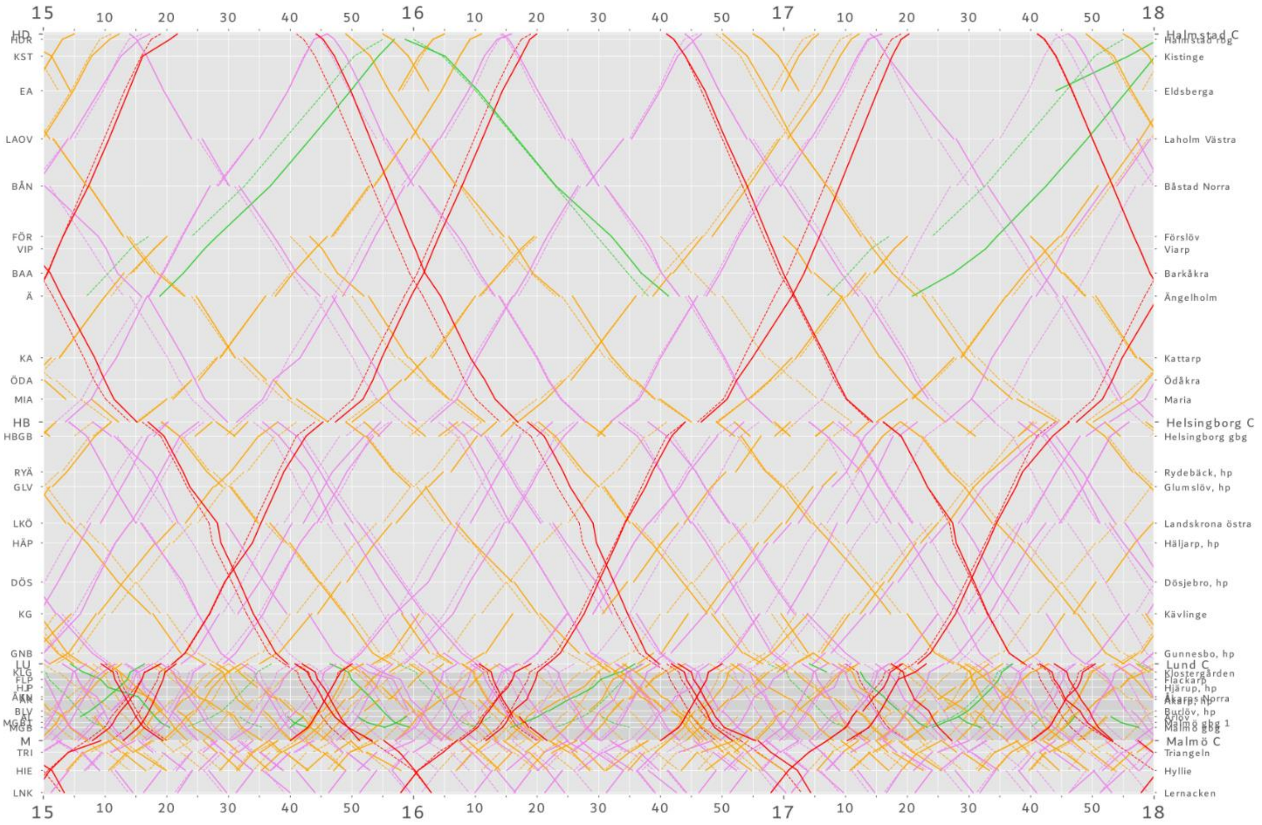


Figure 4. Example of plotting scheduled and PROTON simulated train paths in a graphical timetable. Dashed lines show scheduled paths and solid lines show simulated paths for one cycle. Colours represent different train categories.

Development work needed for demonstration 9.1 includes both to further extend the framework around PROTON simulations in Sweden, preparing data and processing simulation output. There is also a specific need to develop the methodology around using historical data for creating primary delays (see Section 6.1 *Demonstration 9.1: Analysing historical data to create primary delay distributions*), in connection to extending the simulation areas to larger and larger parts of the Swedish network. A large share of the Swedish railway network will be simulated in demonstration 9.1. The idea is that sub-networks should primarily consist of TRV's national railway dispatching areas and simulations can be setup by selecting one or more of these areas. Figure 5 gives an overview of the eight dispatching areas in Sweden, the development to a full network simulation will be built up by adding more and more of these dispatching areas.

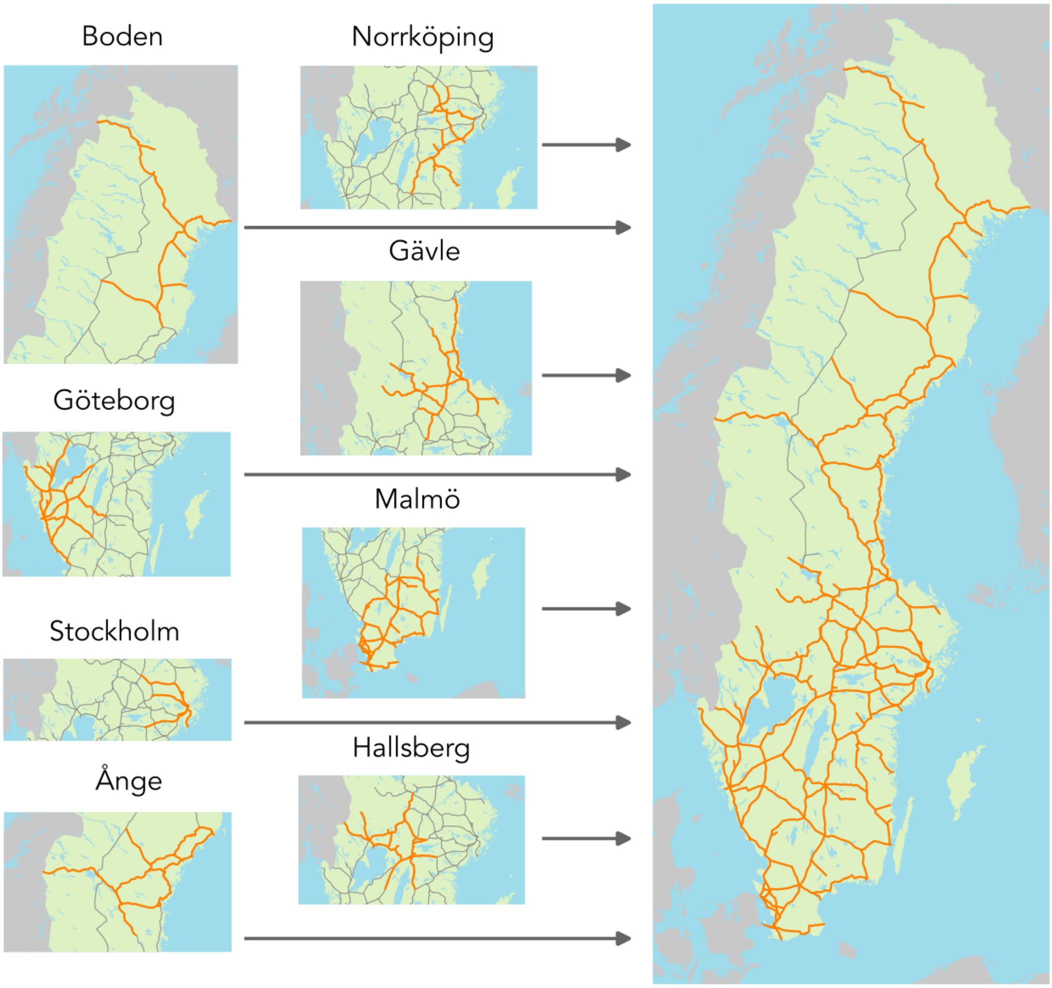


Figure 5. National railway dispatching areas in Sweden.

5.2. Demonstration 9.2: Feedback loops between crew plan and operation

The assessment of the robustness of a crew plan is novel to train operating companies. An increasingly complex timetable with more trains on the same infrastructure poses challenges to the network punctuality. In disrupted situations, the rescheduling of crew is time-critical. We want to develop a crew plan that is as robust as possible, so that it takes away some pressure of the dispatchers.

New timetables are introduced in the coming years, introducing challenges to crew rescheduling. For certain lines, there are less possibilities to change crew, which means that connections and meal breaks move to other stations. NS wants to check whether there are new vulnerabilities in the plan, and whether dependencies by crew are propagating delay from one corridor to another.

First the question arises: what makes a crew plan robust? By what characteristics do we judge the robustness, and how can it be quantified? We start by measuring the delay propagation because of crew dependencies: if an inbound train and an outbound train at a given station share the same train driver, how much delay of the inbound train will be propagated to the outbound train?

In the current crew plan all consecutive train trips that share train driver but not rolling stock are planned with at least 20 minutes connection time (except for the case where the train driver travels as a passenger on at least one of the trips). A part of those 20 minutes is necessary for operations: i.e., getting off the train, walking time, toilet break, starting up the next train. Another part is a buffer for absorbing delays (and thus mitigate propagation). The research goal is to better distribute this buffer, to times, places and type of service where the benefit is the largest. The total amount of buffer must stay the same. We want to achieve this by developing a decision support tool that combines optimisation and simulation. As shown in Figure 6, this tool involves a controller, data interfaces, a duty optimiser, and a simulator.

Buffer times are also convenient when a train trip is cancelled, because it provides some time before and after the cancelled trip for the dispatcher to manage the disruption. So, depending on the cancellation risk, more or less buffer time should be added to the crew plan. The question is how much.

In order to address this challenge NS joined forced with SISCOG, the developer of a state-of-the-art railway crew scheduling optimiser, one of the few optimisers used in practice mentioned in the survey by Heil et al. (2020).

5.2.1. Controller

The controller is developed by SISCOG in the context of this project. It controls the way the optimiser and the simulator cooperate with each other towards obtaining robust crew plans.

So far, the following capabilities have been developed:

- Launch the duty optimiser.
- When the duty optimiser finishes, extract relevant information to send to the simulator taking in consideration the optimiser-simulator interface described below.
- The simulator-optimiser data interface described in the feedback loop further below.

5.2.2. Development of optimiser-simulator interface

This interface is developed by SISCOG and NS. It is shown in the upper part of the figure and establishes the communication between the optimiser and the simulator. This interface contains:

- Crew dependencies: a link between train A and B at station X where a driver changes trains. This dependency has a certain buffer time.
- Crew duties: for each duty a list of tasks with information about the times and trains that are driven. It also contains duty start, meal breaks and duty end.
- General information: list of home bases etc.

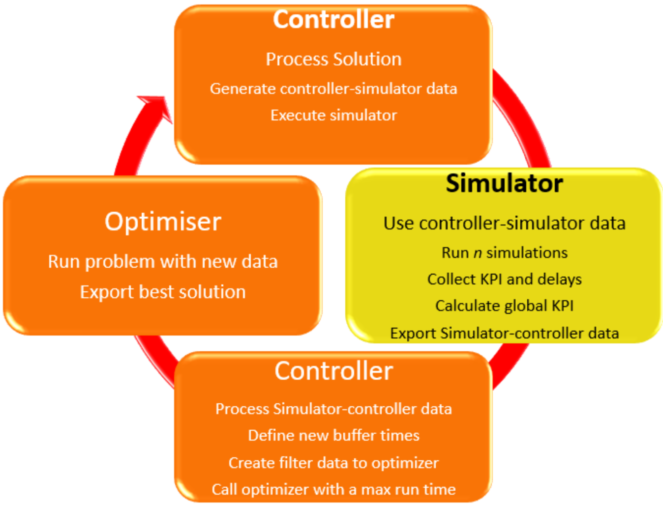

Controller
Process Solution
Generate controller-simulator data
Execute simulator
Simulator
Use controller-simulator data
Run *n* simulations
Collect KPI and delays
Calculate global KPI
Export Simulator-controller data
Controller
Process Simulator-controller data
Define new buffer times
Create filter data to optimizer
Call optimizer with a max run time
Optimiser
Run problem with new data
Export best solution

Figure 6. Diagram representing the solution architecture.

5.2.3. Development of simulation software for assessing robustness of crew plan

NS is developing a new simulation model that simulates the combination of a timetable, rolling stock and crew. This macroscopic model is built in AnyLogic, a Java-based generic simulation software. Below is an outline of the working of the model.

Input:

- Timetable (currently 2023)
- Rolling stock plan from the same period
- Crew dependencies and/or duties
- Delay distributions: dwell time, driving time and departure at big stations.

Agents:

- Crew duty
- Train (rolling stock unit)
- Tasks (mapping train on timetable)
- Infrastructure between stations
- Station

Output:

- Average delay
- Delay per time of day/location/train series
- Delay propagation because of crew dependency

For this simulation model, a new way of macroscopic infrastructure modelling is developed. This enables a nation-wide simulation model, with the possibility of also simulating trains from other operators. Figure 7 and Figure 8 illustrate the network layout and agent modelling.

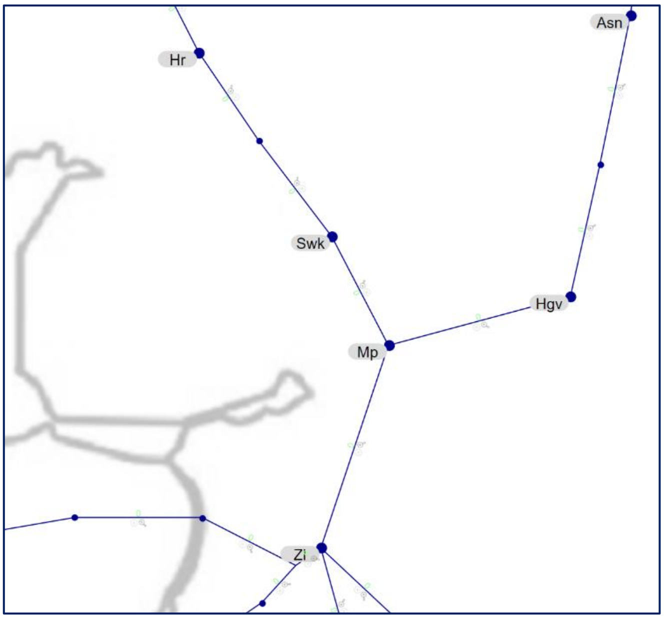


Figure 7. Macroscopic network view in the simulation model.

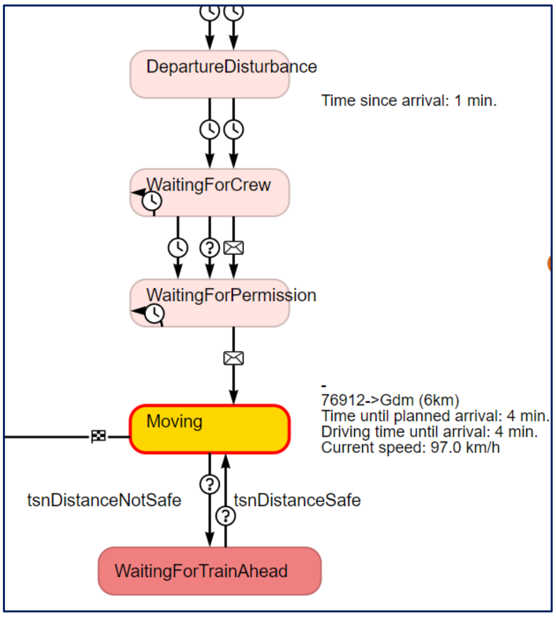


Figure 8. Agent-based modelling with a state chart, in this case for a train unit.

5.2.4. Development of feedback loop from simulation to optimiser

Two feedback loops have been developed, both providing input to the optimiser, see Table 2. In both cases, the metrics are aggregated on train series and direction levels. Otherwise, the input to the duty optimiser would be too specific. By combining all trains of a series but making a distinction per direction, the results become more generic thus avoiding solutions that would become sensitive to every particular train.

The simulation output is aggregated over the day, because the delays in the simulation model are from a single distribution, derived from the average over the day. To prevent inaccurate and meaningless information, we chose to only output aggregated delays.

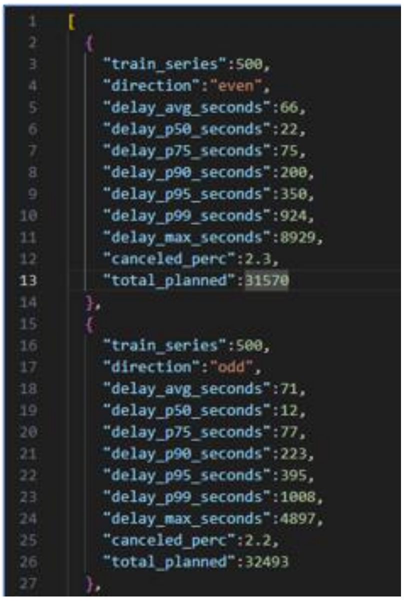
Because of the intrinsic randomness of the model (stochastic delay distributions), the simulation output is based on multiple runs for each input. For a representative outcome it is necessary to perform a Monte Carlo simulation with multiple runs.

Table 2. Feedback loops from realisation data and simulation to optimiser.

<table>
<tr>
<td>

1) Realisation data from data warehouse, includes the following data per train series and direction (see also Figure 9):

- Delay percentiles (P50, P75, P90, P95, P99, P100) and average delay
- Percentage of cancelations
- Number of observations (i.e., the number of trains the metric is based on)



```
[
  {
    "train_series":500,
    "direction":"even",
    "delay_avg_seconds":66,
    "delay_p50_seconds":22,
    "delay_p75_seconds":75,
    "delay_p90_seconds":200,
    "delay_p95_seconds":350,
    "delay_p99_seconds":924,
    "delay_max_seconds":8929,
    "canceled_perc":2.3,
    "total_planned":31570
  },
  {
    "train_series":500,
    "direction":"odd",
    "delay_avg_seconds":71,
    "delay_p50_seconds":12,
    "delay_p75_seconds":77,
    "delay_p90_seconds":223,
    "delay_p95_seconds":395,
    "delay_p99_seconds":1008,
    "delay_max_seconds":4897,
    "canceled_perc":2.2,
    "total_planned":32493
  },
```

Figure 9. Extract of the data interface for the feedback loop that contains realisation data.

</td>
<td>

2) Simulation output, includes:

- Delay percentiles (P50, P75, P90, P95, P99, P100) and average delay per train series and direction
- Number of observations

The simulation has additional visualisations during runtime for verification:

- Number of active trains
- Average delay of all trains in the model

</td>
</tr>
</table>

5.2.5. Validation and technical readiness level

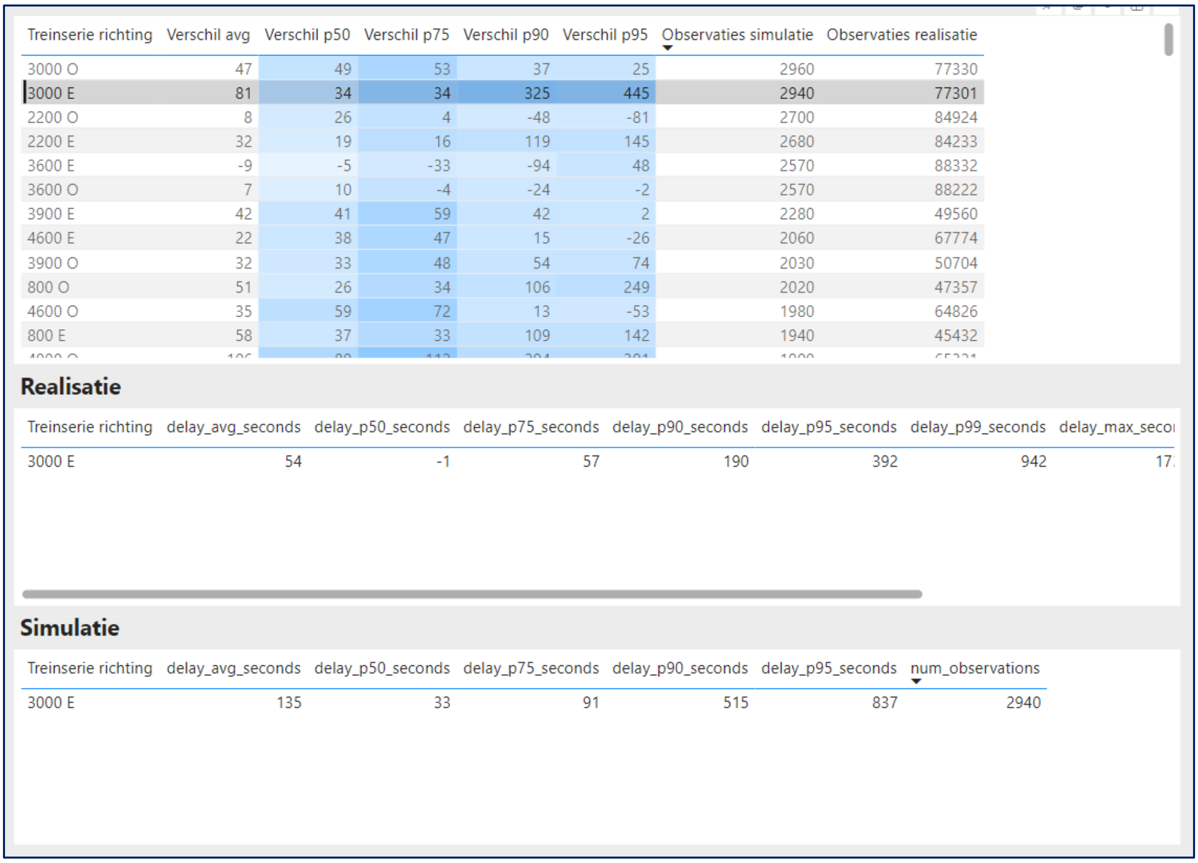
The assessment of the model correctness starts with face verification: we looked at the graphical interface of the model to see if all trains were running correctly according to their timetable. The case study is based on the February 2023 timetable. At the start of the development, that was the most recent timetable for which a full crew plan and realised data from operations were available.

During this verification, the developers discovered minor anomalies in the matching between the timetable and rolling stock plan. Most issues could be traced back to edge cases in the input data (i.e., trips for testing trains, empty stock movements). These issues have been solved in consequent versions of the data preparation. The current version of the simulation model runs without any issues in the timetable of the trains, rolling stock or crew.

With the correct input, the simulation developers ran iterations of the model to verify whether the input delay distributions had the desired effect. The train delays in the model must be realistic, while avoiding situations where in reality a dispatcher would take action (e.g., trains that leave with more than 30 minutes delay would in reality be cancelled if the next is train on time).

While validating this, it was found that the number of delays larger than five minutes was higher than expected. The cause was found in the start-up delays, which we introduce when a train starts a new trip, after a long stop at a major station or after turning around. These were applied unintendedly to trains that did not have a passenger stop at stations. We adjusted the filtering and the simulated delays were much closer to reality after this correction.

In the next step the developers compared differences between the delays in the simulation and observed delays in reality. The level was on train series per direction. This indicated if there are structural differences between the simulation and reality, which might be geographical or due to specific situations in the timetable. Train series with the highest number of observations in the simulation were the starting point, these are long train series running through the entire country.



Treinserie richting	Verschil avg	Verschil p50	Verschil p75	Verschil p90	Verschil p95	Observaties simulatie	Observaties realisatie
3000 O	47	49	53	37	25	2960	77330
3000 E	81	34	34	325	445	2940	77301
2200 O	8	26	4	-48	-81	2700	84924
2200 E	32	19	16	119	145	2680	84233
3600 E	-9	-5	-33	-94	48	2570	88332
3600 O	7	10	-4	-24	-2	2570	88222
3900 E	42	41	59	42	2	2280	49560
4600 E	22	38	47	15	-26	2060	67774
3900 O	32	33	48	54	74	2030	50704
800 O	51	26	34	106	249	2020	47357
4600 O	35	59	72	13	-53	1980	64826
800 E	58	37	33	109	142	1940	45432

Realisatie

Treinserie richting	delay_avg_seconds	delay_p50_seconds	delay_p75_seconds	delay_p90_seconds	delay_p95_seconds	delay_p99_seconds	delay_max_seco
3000 E	54	-1	57	190	392	942	17

Simulatie

Treinserie richting	delay_avg_seconds	delay_p50_seconds	delay_p75_seconds	delay_p90_seconds	delay_p95_seconds	num_observations
3000 E	135	33	91	515	837	2940

Figure 10. Validation of delay percentiles, comparing simulation output with realisation data, per train series, all weekdays.

Figure 10 shows the train series 3000 (with directions **O**dd and **E**ven), which has a high number of observations. It compares the percentile values of the delays (median, P75, P90, P95) between simulation results and reality. In this early phase, the simulation output is based on a single run, not a Monte Carlo run with replications. We observe that in the ‘odd’-direction the differences of the P90 and P95 are much higher than in the P50/P75 and higher than the ‘even’-direction. This indicates that there is more delay in the simulation than in reality, possibly caused by a deadlock situation in the simulation or a structural issue in the timetable or infrastructure of the model. This requires further investigation. Also running multiple stochastic replications is an important step for further validation.

In conclusion, the new simulation model *AnyCrew* has been tested and validated according to TRL4. The correct working of the model is proven under development conditions, comparable to laboratorial environment.

5.2.6. Verification of crew plan from optimiser

One of the research questions is to find out whether we can make a more robust crew plan by taking into account the chance that trains are cancelled. This cancellation chance differs significantly between train series, as some provide the basis of the service and are less frequently cancelled than services added only in peak hours.

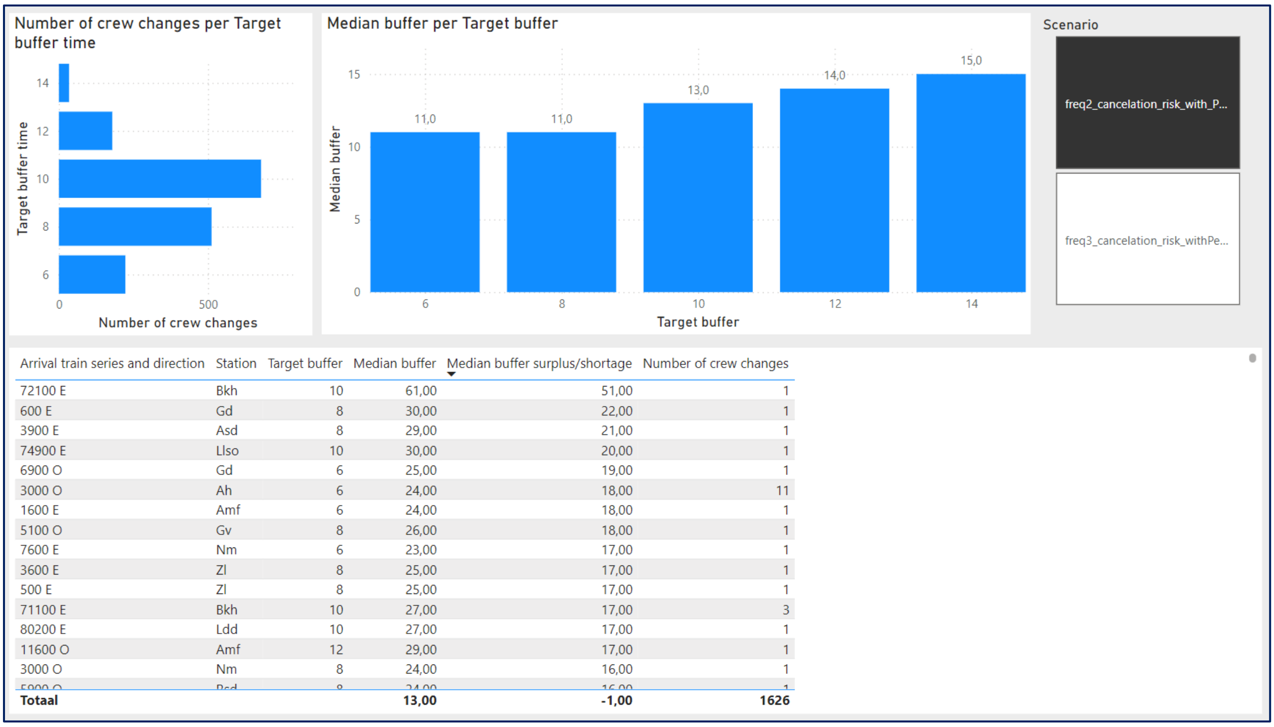
We define a basis minimum connection time of 10 minutes, with added buffer depending on the chance of the trains being cancelled, before and after the task, as shown in Table 3.

Table 3. Additional buffer times for different categories of cancellation risk.

Cancellation risk category	Cancellation percentage	Additional buffer time
Low risk task	<2% risk of cancellation	+3 min buffer time both sides of task
Medium risk task	2-4% risk of cancellation	+5 min buffer time both sides of task
High risk task	>4% risk of cancellation	+7 min buffer time both sides of task
Fixed connection time to perform operations: 10 minutes		

This results in target connection times between 16 and 24 minutes, of which 10 minutes are minimum, the rest is buffer time. The targets are implemented in the optimiser as soft constraints, penalised in the cost function. They are not absolute, which means that the buffer time can be lower than the target. To check whether the buffer times in the plan meet the targets, we visualised the target buffer and median buffer (see upper graph in Figure 11).

In Figure 11 we see that the buffer times for all targets have a higher median (between 11 and 15) than the target (6 to 14). We can conclude that even when the buffer time is allowed to be less than 10 minutes, the duty solution does not exploit this, due to the nature of the timetable and other constraints (e.g., a longer lay-over is required after a certain driving time).


Number of crew changes per Target buffer time
Target buffer time
14
12
10
8
6
0
500
Number of crew changes
Median buffer per Target buffer
Median buffer
15
10
5
0
11,0
11,0
13,0
14,0
15,0
6
8
10
12
14
Target buffer
Scenario
freq2_cancelation_risk_with_P...
freq3_cancelation_risk_withPe...

Arrival train series and direction	Station	Target buffer	Median buffer	Median buffer surplus/shortage	Number of crew changes
72100 E	Bkh	10	61,00	51,00	1
600 E	Gd	8	30,00	22,00	1
3900 E	Asd	8	29,00	21,00	1
74900 E	Llso	10	30,00	20,00	1
6900 O	Gd	6	25,00	19,00	1
3000 O	Ah	6	24,00	18,00	11
1600 E	Amf	6	24,00	18,00	1
5100 O	Gv	8	26,00	18,00	1
7600 E	Nm	6	23,00	17,00	1
3600 E	Zl	8	25,00	17,00	1
500 E	Zl	8	25,00	17,00	1
71100 E	Bkh	10	27,00	17,00	3
80200 E	Ldd	10	27,00	17,00	1
11600 O	Amf	12	29,00	17,00	1
3000 O	Nm	8	24,00	16,00	1
5900 O	Rsd	8	24,00	16,00	1
Totaal			**13,00**	**-1,00**	**1626**

Figure 11. Check of crew connections made by the optimiser, to see whether they meet the expectations.

Concluding the validations, we see that the model and plans from the optimiser are fit for purpose: the delays in the model are representative enough for the current applications of the model (comparing crew plans). We investigate the differences in delays between simulation and realisation and make improvements to get these more in line with each other. We will keep validating the model during the demonstration phase.

5.2.7. Tuning the duty optimiser for robust scheduling

SISCOG is making small adjustments in the duty optimiser aiming at improving the performance when solving robust scheduling problems. The duty optimiser is a component of CREWS, the product of the SISCOG Suite related with planning and managing the work of crew members, devoted to automatically produce an optimised crew schedule. More detail about the optimiser can be found in Abbink et al. (2011).

So far, SISCOG replaced the standard pricing procedure of the duty optimiser by a procedure more suitable for robust scheduling.

In order to assess the performance of this tuning, tests were done with two train driver scheduling real-world problem instances from NS, shown in Table 4:

Table 4. Problem instances used to tune the optimiser.

Problem instance	Number of bases	Number of tasks
P1	30	9249
P2	30	8726

Based on these two problems instances several runs were made aiming at achieving different levels of robustness against delay propagation. This was achieved by running the optimiser with different sets of target buffer time soft constraints. These sets of soft constraints define different minimum buffer times for connections where the driver has to transfer between a normal train and any kind of train and between a high-speed train and any kind of train. Table 5 shows the number of duties and total buffer time obtained in each run.

Table 5. Numeric results of optimiser for different levels of robustness against delay propagation

Problem instance	Target buffer time normal trains	Target buffer time HS trains	Number of duties	Total buffer time (hours)
P1	0	0	844	1107.36
	8	12	849	1261.43
	10	10	862	1351.38
P2	0	0	805	1057.15
	8	12	812	1204.35
	10	10	818	1255.85

As shown in Table 5, the optimiser behaves consistently in the sense that it obtains solutions with increasing number of duties as more buffer time (i.e., as more robustness) is added to the solution. The second and third scenarios in each problem instance are particularly interesting because they confirm the benefits obtained by going from a uniform buffer time distribution to a non-uniform one.

Also based on the two problem instances, several runs were made aiming at achieving different levels of robustness against cancelation risk. This was achieved by running the optimiser with and without the soft constraints presented in Table 5. Table 6 shows the number of duties and total target buffer time violation obtained in each run.

Table 6. Numeric results of optimiser for different levels of robustness against cancelation.

Problem instance	Cancelation risk soft constraints?	Number of duties	Total target buffer time violation (hours)
P1	No	844	157.93
	Yes	857	8.70
P2	No	805	144.13
	Yes	814	8.15

As shown in Table 6, the optimiser behaves consistently in the sense that it obtains solutions with increasing number of duties as less target buffer time violation (i.e., as more robustness) is achieved in the solution.

5.3. Demonstration 9.4: Assess the feasibility of a change in the topology network

The aim is to evaluate the infrastructure layout developing new functions in the Infrastructure Capacity Analysis tool (TMS_CAP). This helps to improve the timetabling process demonstrating the performance of a capacity analysis tool with several topologies. In case of poor performance by simulating with a specific topology, the capacity analysis tool is used to accurate identify bottlenecks that can be solved by topology changes. Iterative simulations with different topologies facilitate timetabling.

Currently, operation feedback is provided using the post-analysis module that provides reports of the operation. Indra's simulation environment allows to simulate train performance following a scheduled plan (timetable). The goal is to improve the timetable based on feedback loop between C-DAS data and simulation tool/TMS run with different topologies.

The plan management is focused on providing a feasible and conflict-free timetable based on, on the one hand an accurate infrastructure modelling as the enabler of realistic traction and route calculation. On the other hand, facilities to create and edit services, temporary entities (Temporary Speed Restrictions, possessions, isolations, and station closures), links and connections.

The simulation environment allows to:

- Define train physics (train formation, acceleration, etc)
- Driver management (behaviour of the train driver)
- Interactive train management (adding/removing trains, etc) by a simulator operator
- Replan train paths by a dispatcher

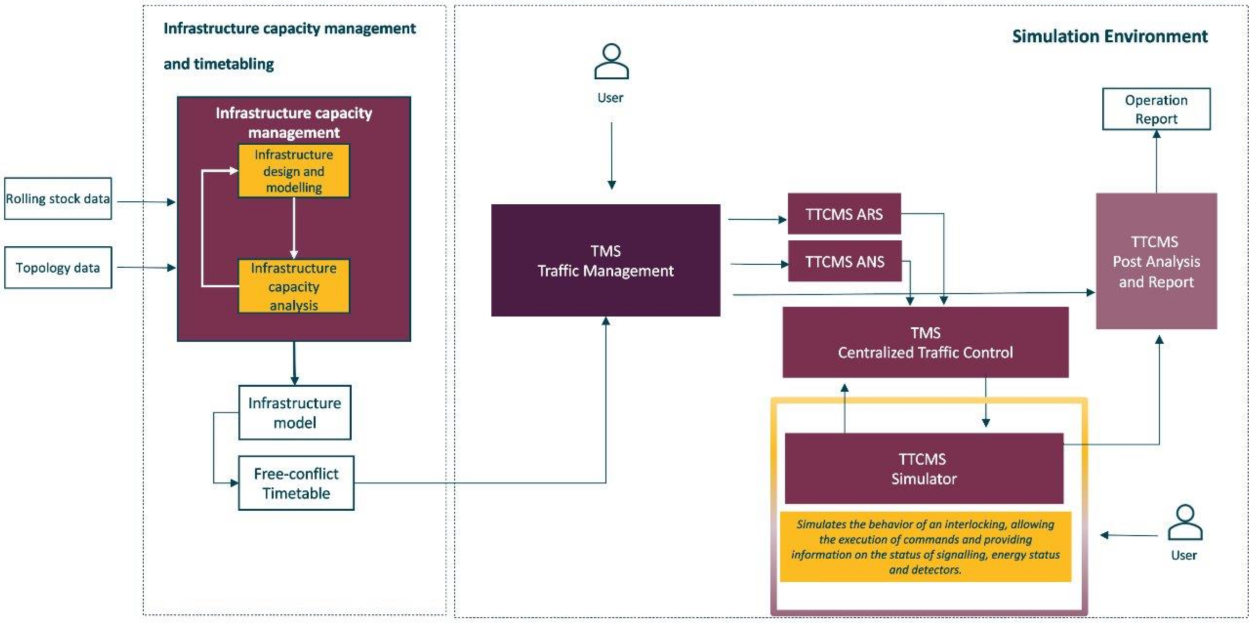

Infrastructure capacity management and timetabling
Infrastructure capacity management
Infrastructure design and modelling
Infrastructure capacity analysis
Rolling stock data
Topology data
Infrastructure model
Free-conflict Timetable
Simulation Environment
User
TMS Traffic Management
TTCMS ARS
TTCMS ANS
TMS Centralized Traffic Control
TTCMS Post Analysis and Report
Operation Report
TTCMS Simulator
Simulates the behavior of an interlocking, allowing the execution of commands and providing information on the status of signalling, energy status and detectors.
User

Figure 12. Simulation environment.

Using the simulation´s environment (see Figure 12) and testing with a specific topology and timetable, we can detect a poor planning behaviour. Our tool (TMS_CAP) is specifically designed for the capacity evaluation of railway networks. It provides rich functionality in order to assess capacity of the network with different topologies. It helps operators make informed decisions regarding signalling adjustments or even relocating railway elements to optimize network performance.

Some of the key features of the tool are:

- **Train Movement Simulation**: Simulates train positions across different networks, providing detailed tracking and interaction analysis.
- **Block Time Analysis**: Calculates how long trains occupy track segments, optimizing track usage and efficiency.
- **Critical Block Identification**: Highlights key sections of the network that impact capacity, guiding improvement efforts.
- **Conflict Detection**: Identifies bottlenecks or scheduling conflicts that may require changes in signalling or infrastructure.

We intend to enhance the existing tool providing microscopic simulations of train movements, including block times and critical block information using real-world infrastructure data and new ATP systems (ETCS Level 1 and ETCS Level 2) applying their behaviours to the calculation and signalling procedures to achieve optimal performance.

It employs deterministic travel times in its capacity calculations, ensuring consistent and reproducible results. Expected results are a high level of adaptability to the actual operating conditions of the railway infrastructure and assess the feasibility of the timetable.

The software uses an event-based simulation approach. This allows for a thorough analysis of train movements and interactions within the network, crucial for identifying and resolving conflicts. Key functionalities include detailed reports on train movements, block times analysis, critical block identification, conflict detection, and enhanced capacity calculations integrating multiple ATP systems.

5.3.1. Method of development for demo 9.4

We will demonstrate the performance of a capacity analysis tool with several topologies and timetables. In case of poor performance by simulating with a specific topology, the capacity tool is used to accurate identify bottlenecks that can be solve by topology changes.

As shown in Figure 13, the simulation is run in medium-complexity stations on a single-track layout. The tool uses real data for simulation purposes, but does not connect to a real TMS.

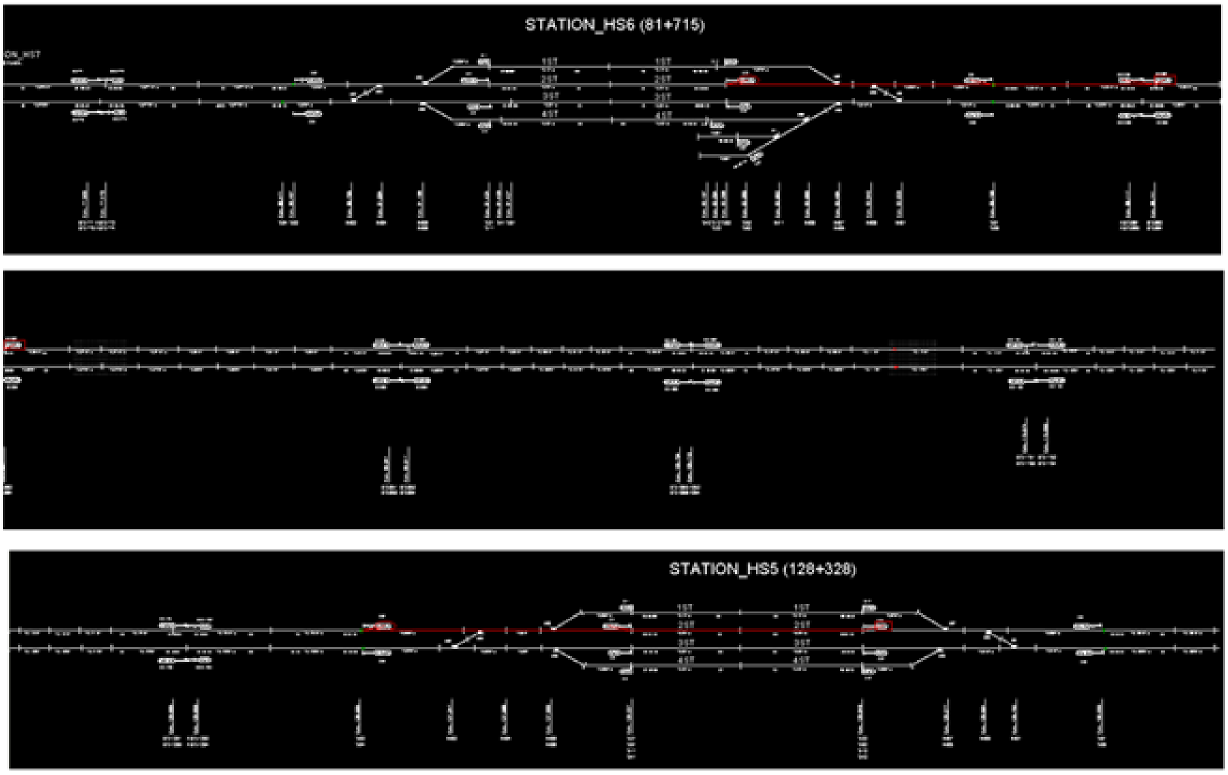


Figure 13. Simulation station.

This level of detail is crucial for accurately simulating how different signaling systems affect (e.g., ERTMS L0, ERTMS L1) train speeds and, consequently, capacity. We analyse train movements, block section occupations, and the performance of signalling systems.

5.3.2. Expected results for demo 9.4

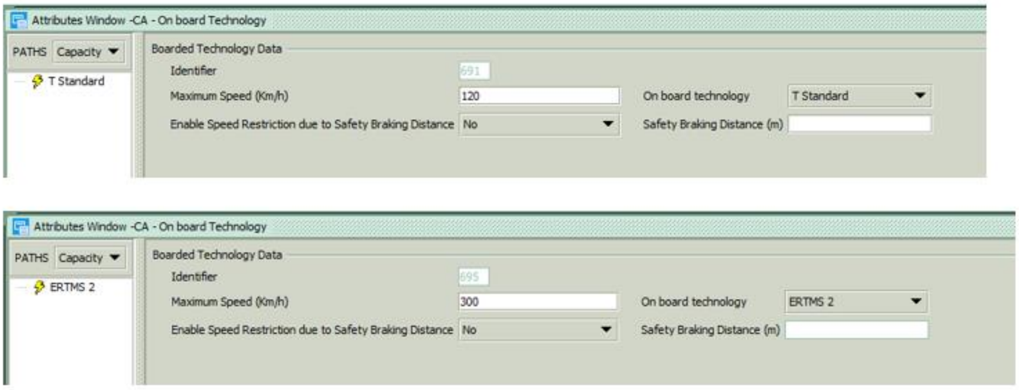
The expected results from the simulation tool include comprehensive reports on train movements, precise block time analyses, and identification of critical blocks essential for network efficiency. The tool will detect capacity restrictions, enabling prompt resolution and enhancing overall capacity calculations by integrating new ATP systems, such as ETCS Level 1 and Level 2. Additionally, the tool is anticipated to demonstrate high adaptability to actual railway operating conditions, allowing us to assess timetable feasibility and suggest necessary infrastructure optimizations for improved performance.

The tool provides deterministic, event-based rescheduling to accurately reflect real-world scenarios and conflicts and generates empirical data that can be used for further capacity analysis. The simulation and comparison are based on a topology designed to support both conventional and ETCS Level 2 signalling systems, ensuring compatibility across these technologies.

We can see how the system behaves with both conventional and new technologies, which can be selected for corresponding capacity analysis studies. Considering trains at their maximum allowable speed, the final screenshot graphically illustrates the system's output, showing that with the new ETCS Level 2 technology, train speeds increase and more trains can be accommodated, thereby significantly improving capacity.

Different Technologies that the selected train can use for the capacity study. Attributes window is shown in Figure 14 and Figure 15. Other important data:

- Conventional Technology has a Maximum speed = 120 Km/h
- ERTMS2: Technology has a Maximum speed = 300Km/h


Attributes Window -CA - On board Technology

PATHS Capacity

T Standard

Boarded Technology Data

Identifier: 691

Maximum Speed (Km/h): 120

Enable Speed Restriction due to Safety Braking Distance: No

On board technology: T Standard

Safety Braking Distance (m):

Attributes Window -CA - On board Technology

PATHS Capacity

ERTMS 2

Boarded Technology Data

Identifier: 695

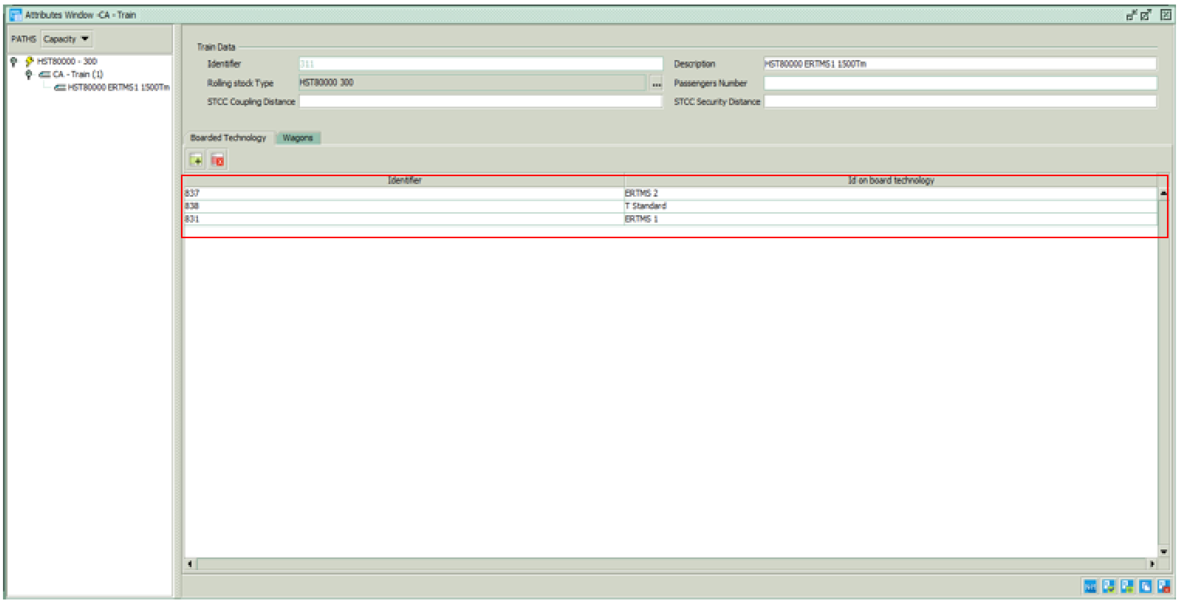
Maximum Speed (Km/h): 300

Enable Speed Restriction due to Safety Braking Distance: No

On board technology: ERTMS 2

Safety Braking Distance (m):

Figure 14. Attributes window. On board technology.


Attributes Window -CA - Train

PATHS Capacity

HST80000 - 300
CA - Train (1)
HST80000 ERTMS1 1500Tm

Train Data

Identifier: 311

Description: HST80000 ERTMS1 1500Tm

Rolling stock Type: HST80000 300

Passengers Number:

STCC Coupling Distance:

STCC Security Distance:

Boarded Technology | Wagons

Identifier	Id on board technology
837	ERTMS 2
838	T Standard
831	ERTMS 1

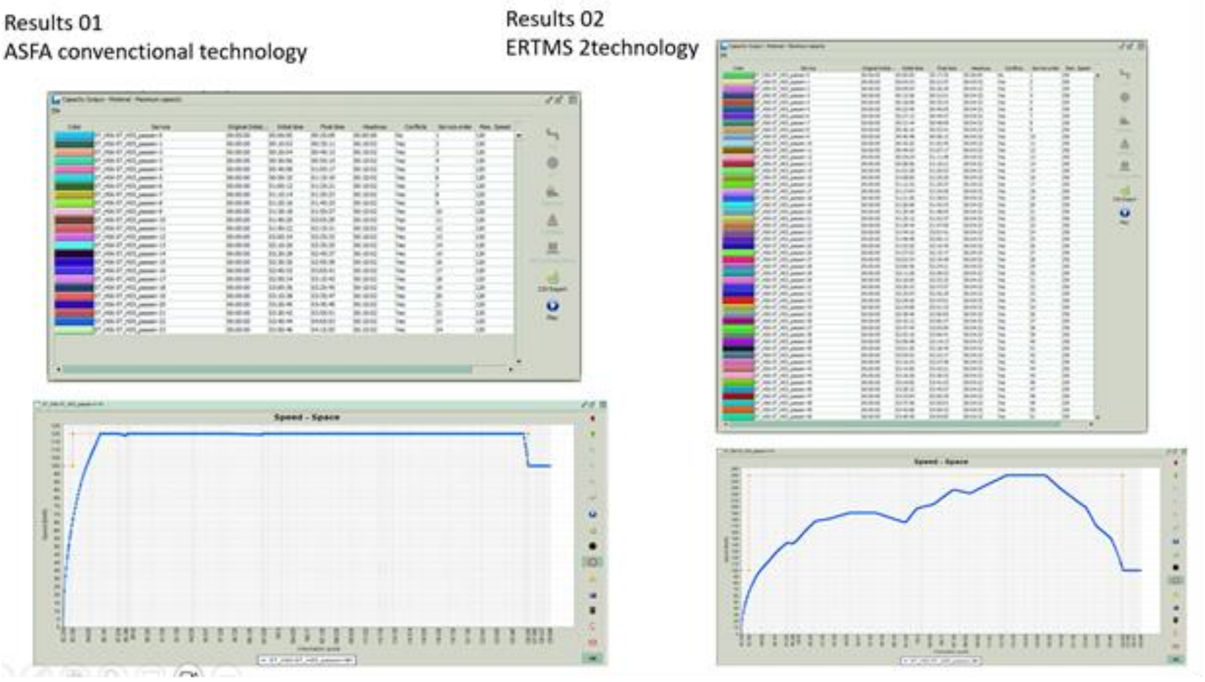
Figure 15. Attributes window. Train Data.

Below in Figure 16 are the results provided by the tool, taking into account the scope of work in WP8, including enhancing the tool by incorporating ETCS 1 and ETCS 2, which are components of the broader ERTMS framework.

We can observe that in Results 01, the exercise outcome for a train with standard/conventional technology, there are 23 services at a maximum speed of 120 km/h.

On the other hand, in Results 02, the exercise outcome for a train with ETCS L2 technology, there are up to 50 services at 250 km/h.

Significant differences can be seen between the technologies used.


Results 01
ASFA convencional technology

Results 02
ERTMS 2technology

Figure 16. Results comparative. ASFA vs ERTMS technologies.

Additionally, the tool is anticipated to demonstrate high adaptability to actual railway operating conditions, allowing us to assess timetable feasibility and suggest necessary infrastructure optimizations for improved performance.

5.4. Demonstration 9.8: Generate planification through different inputs

5.4.1. State of the art

The CAF TOOL provides data on the line under analysis, with respect to the theoretical minimum interval and cantonment sizing under the assumption of a line equipped with ERTMS Level 2 although you can make analysis of line equipped with ERTMS Level 1. CAF TOOL allows you to:

- Capacity analysis: obtaining the theoretical minimum interval and the train capacity of the line.
- Dimensioning analysis: block section design; Calculation of the maximum length of block section design in tracking sections (constant or variable) to allow operation with a given minimum interval.

In order to carry out the functionalities described; the tool includes the following internal functions:

- Gait simulation, either calculated internally or provided by the user.
- Calculation of ERTMS braking distances: Parameterizable to simulate different curves and behaviours (driving by driver in Level 2, driving with ATO).

For the global analysis of the line, it is necessary to carry out the specific analysis of each of the nodes/sections by which it is composed. By knowing the minimum interval of each node, the theoretical minimum interval of the line can be known. This theoretical minimum interval assumes that the ideal conditions of the exploitation pattern that has been assumed in each of them are occurring in each node, this is the theoretical maximum capacity.

The tool allows you to define two trains with different characteristics (persecutor and pursued services).

5.4.2. Method of development for demo 9.8

Concerning the development that has been carried out, we have analysed the different needs that exist in the market, seeing that we needed to adapt the tool in order to be able to carry out capacity analysis with HL3.

All the details of how we have defined the parameters that we needed new ones as well as the modifications, if necessary, of the ones we already had defined can be seen in D8.3: Developed simulation methods and models for capacity evaluation of ETCS and C-DAS/ATO.

5.4.3. Expected results for demo 9.8

The expected result that we want to obtain from the simulations done with this tool, in the framework of this package, is to be able to obtain the minimum interval of all lines regardless of whether they are equipped with ERTMS Level 1, Level 2 or HTD.

In the same way, if we look at the simulations, we will do on lines equipped with HL3 it will also allow us to make calculations of the cost reduction that using this level would bring us since we could reduce the amount of material that is being used in the field.

6. Develop methods for analysing historical data from operations to improve capacity evaluations

In stochastic simulations, primary delays are inserted to capture timetable performance indexes such as punctuality and arrival delay. To get accurate output from the simulations, it is important that the input is correct and corresponds to real world historical data. Therefore, we 1) need to perform analysis of historical data and 2) calibrate primary delay distributions, to turn the data into realistic disturbance distributions which can be used in operational railway simulations.

6.1. Demonstration 9.1: Analysing historical data to create primary delay distributions

An important part in calibrating and validating simulation models is to test the model behaviour with realistic disturbances. To do so, we first need to analyse historical data and create primary delay distributions that can be used in the simulation model. The aim is to create a method for processing and analysing historical data to produce realistic and relevant disturbance scenarios. This data should then be used in use case FP1-DEMO-9.1-UC-1 to validate the macroscopic simulation model PROTON see Section 5.1 *Demonstration 9.1: Feedback loop from simulation to planning for large scale network*.

6.1.1. Data description

The data we start with are operational records of train movements in Sweden, which include any delays that happen, along with attribution data. These data do not cover cancelled trains, and the planned simulations do not cover the simulation of train cancellations. We have these data for more than twenty years, going back to 2001, with the same structure every year. However, we typically use only a subset of these years, often one year, when we create the primary delay distributions. This is because conditions change from year to year, in particular the number of trains operated generally increases by 1-3% per year, and this can be expected to affect the proportion of primary and secondary delays, so that calibrating distributions of several years of data at once can be problematic.

The data is structured so that one row represents the arrival or departure time at one station. This corresponds to a macroscopic simulation level, with data registered at stations rather than signals. In the background, these data are generated by signal-level data, at the entry and exit signals to stations, with timestamps being adjusted by a constant which is intended to approximate the running time between the signal in question and the middle of the platform (on the main track, if there are multiple tracks). The timestamps are then truncated to whole minutes, losing detail on the level of seconds.

In essence, we thus use top-down data from a macroscopic level. Another approach would be to model delays from the bottom up, starting with a listing of all components and delay causes, along with frequencies and size distributions of these delays, and then aggregating up to a holistic distribution. Rather, we start with the timestamps of trains moving into and out of stations.

A detail in the structure of the data is that timestamps of arrival times only happen at stations where the trains are *scheduled* to stop. At stations where trains do *not* stop, only the departure times are registered. We can and do adjust this by creating arrival time observations, assuming that the arrival time is equal to the departure time, so that we can then more easily generate run times in the same way for all stations. For dwell time delay distributions, we only use observations where trains are scheduled to stop, where we have "real" observations of the dwell time.

From the sets of arrival and departure timestamps, we derive delay distributions which can be input into simulation models. In practical terms, we store both sets of data in a database and process the data using SQL. What we need is distributions for entry delays, runtime delays, and dwell time delays. Each of these are made up of size intervals for the deviation around the scheduled process time, typically in one-minute intervals, along with a frequency expressed as a fraction, covering both positive and negative deviations, and no deviations. An example of these distributions is shown in Figure 17 below.

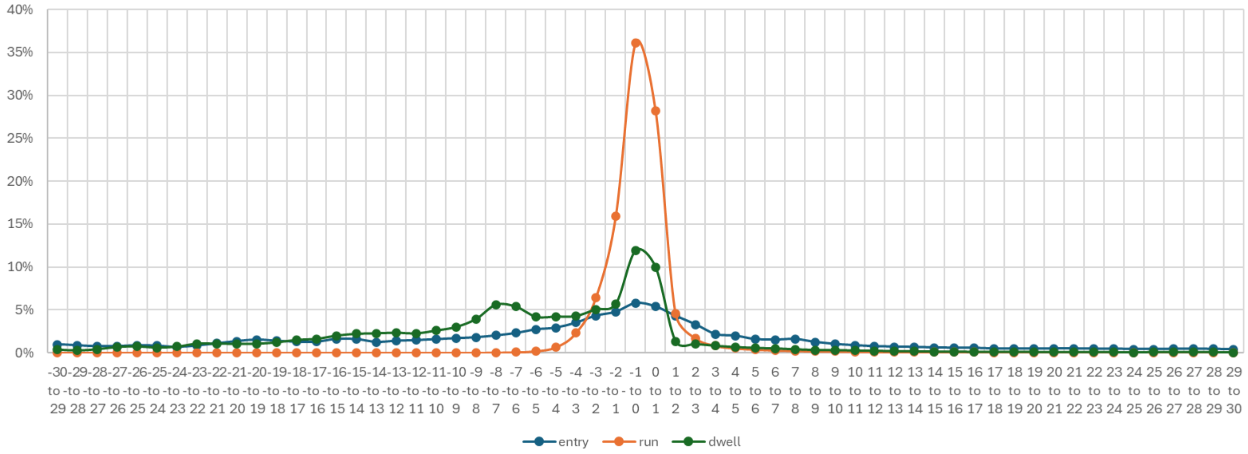


Figure 17. Example distribution of entry, run, and dwell time delays. This example is for freight trains.

When we estimate entry and runtime delays, both positive and negative deviations occur. The same is the case for dwell times, although passenger trains that arrive early to a station (perhaps due to a negative run time deviation) need to be handled in a special way. If a train arrives at a station two minutes ahead of schedule, and thus has a dwell time of four minutes instead of two, this should not be considered as a dwell time delay of two minutes, but rather a deviation of zero minutes. If the train arrives two minutes early, and departs one minute late, we would count that as a dwell time of one minute, not three minutes. This is because passenger trains are not allowed to depart ahead of schedule, after a scheduled stop. This is typically not the case for freight trains, and we do not adjust the dwell time delays in the same way for them – treating them instead in the more straightforward way as the difference in process time between actual and scheduled dwell time. Negative delays can and do still occur sometimes, and are included in the distributions. To simplify simulation, we typically truncate the delay distributions so that entry, run and dwell delays are within a plus minus 60-minute interval. In the empirical data, larger deviations do occur, but they are very rare, and the instability and low probability of them occurring in the simulation model make us prefer to truncate the distributions. This is unlikely to make a major difference to the outputs, however.

Similarly, entry delays can be and often are handled in a special way. Often simulations are done assuming that trains are not allowed to depart early, but in practice we see some deviations around this. In particular, the data suggest that some 60% of freight trains depart ahead of schedule, sometimes by a few hours. This introduces some difficulties in the simulation models, which we have investigated at some depth in Johansson et al. (2022). For this reason, we sometimes truncate the distribution of entry delays at or close to 0, saying that trains that departed early instead departed on time.

When we create delay distributions, it is possible to have separate distributions for different train types (such as passenger and freight trains), but also for different stations, parts of the network, or time windows (such as peak and off-peak hours, for instance). This is an important and useful possibility, but it introduces many complications and difficulties both interpreting, calibrating, and validating the distributions and outputs. We will come back to this later.

An important fact and limitation to consider is that simulation models draw from the delay distributions, assuming that each draw is independent. In practice, disturbances and delays are not necessarily independent of one another. If there is a snowstorm, for instance, many trains are typically delayed to a much greater extent than normal – in a way they draw from a different and worse delay distribution than when the weather is good. Simulation models do not currently handle this in a convenient way – instead both the good and bad weather influences what goes into the distribution, and the draws are then made from a joint distribution. One movement might be drawn from what corresponds to very bad conditions, and the next one from much better ones. This causes the output delay distributions from the simulations to be much more uniform and clustered than what is the case in real operations and empirical distributions. The simulations underestimate the variance in the delays, both between trains during a given day or simulation cycle, and between the days and cycles. The averages might be the same, but the variance is smaller in the simulated than real data.

There are also choices to be made regarding the output measure from simulations, the key metrics which are first calibrated and then used to interpret the results. The most common metrics include punctuality, which can be given as an aggregate or per train type (i.e., the punctuality for freight trains is X, for high-speed passenger trains Y, and for commuter trains Z). A more detailed and expansive alternative would be to target the arrival delay distributions, which would be much more precise. However, this is complicated by the above-mentioned independent draws from the delay distributions. To achieve a good fit between empirical and simulated arrival delay distributions, correlations between delays have to be handled in a different way. For now, the more aggregated punctuality measure is better suited, and suffers less from the underestimated variance in delays.

6.1.2. Methodology

Our approach to calibrating delays is based on assuming that a certain fraction of empirical delays is primary, and the rest secondary. To re-create a distribution of primary delays, we thus apply a reduction factor, F (with a value between 0 and 1), to the empirical distribution of delays. The calibration process then becomes an issue of iteratively determining the size of this reduction factor.

The current state of the art in the experiments (Palmqvist, Johansson & Sipilä, 2023; Minbashi et al. 2022; Johansson et al. 2022) has been to assume the same reduction factor for run and dwell time delays (entry delays have so far been left untouched). In reality, it is quite possible that the factors should be different for the different types of delays. There is not necessarily a strong theoretical reason for them to be different, but similarly there is no strong reason to believe that they must be the same. However, it has been much more computationally efficient to assume that the reduction factor, and thus the proportions between primary and secondary delays, to be the same across the different delay types.

The plan to further develop these methods includes allowing the three different types of delays (entry, run, dwell) to use different reduction factors. In this way, we can better approximate the real conditions and achieve a better fit between real and simulated operations. We will also be able to observe and learn from any differences in the reduction factors, which may help influence policy and other measures to improve the operations. If the share of primary delays is higher for dwell than run times, for instance, this suggests prioritising measures to reduce the primary dwell time delays, and vice versa.

Another assumption in our past efforts to calibrate simulation models has been that the reduction factor is assumed to be the same across all train types, the entire network, and time windows. This has also been for reasons of simplicity, rather than a belief that the factor really is constant across these dimensions. An ambition for later steps in the development of the methodology is thus to allow variation across train types, geography, and possibly time. In addition to the computational difficulties introduced by this change, one must also be cautious not to go too far in this direction, as there is an obvious risk of overfitting to the data. This would lead to a very good fit to the training data, but to a much worse fit to new data and conditions. Thus, we only aim to experiment with a rather course division of train types (perhaps remaining on the level of differentiating between freight and passenger trains, but not pursue a finer division than that), geography (calibrating separately for two or three different regions), and time windows (omitted for now, but in principle day- and night-time operations could be separated).

An implication from these planned developments is that the number of scenarios to simulate grows very substantially. In our past experiments, we have allowed the reduction factor to vary between 0% and 100% in intervals of 5%. This has meant 21 different scenarios to simulate, and then to choose the level that fit best to historical data – possibly with some interpolation along a two-dimensional curve. With this limited number, we have chosen to perform a "full factorial design", simulating every combination. If we were to extend this by allowing the reduction factor to vary across the three types of delays separately, maintaining the granularity of 5%-intervals, this would imply 21^3 = 9,261 scenarios to simulate in a full factorial design. This is already impractical, with the available hardware and software, but not entirely inconceivable.

The next steps are even more demanding. If we were to further allow for different factors for two different train types (freight and passenger), the number of combinations would go to $21^{(2*3)}$ = 85,766,121. This is clearly impossible. Extending further by also allowing variation across three geographies in a combined simulation would yield $21^{(2*3*3)} = 21^{18} = 6.3 * 10^{23}$ scenarios. This is also obviously impossible to perform.

To cope with the rapid expansion of the number of scenarios to simulate as the reduction factor is allowed to vary across more dimensions, we need a different approach. Fortunately, there is an established field called "design of experiments", that is based on dealing with problems like these. By choosing an appropriate design of the experiments, it is possible to choose a much smaller subset of scenarios to simulate. There are many types of experimental designs, and part of the method developments here will involve studying and choosing the appropriate designs. Essentially, however, it will be possible to choose a number of scenarios in the 10s or low 100s that need to be simulated. It will still be a larger number than has been done previously, but it will be within the realm of reason.

Based on the simulated subset of scenarios, one will then perform a linear regression to estimate a model that describes how the different parameters interact and make up the space of outcomes. It is then possible to analytically or numerically solve this model, so that the estimated punctuality is equal to the observed punctuality. The solution then tells us the values of the reduction factors for each of the delay distributions (i.e., X for passenger train dwell delays in region A, Y for freight train run delays in region B, etc.). To validate the results, the idea is then to run a simulation with this particular combination of parameters, to see that the simulated punctuality does indeed fit closely to the empirically observed one.

In principle, we believe that the approach outlined above can be gradually extended. One natural extension would be to repeat the cycle again, with more granular steps and a refined starting position for the parameter values. This would enable a higher precision. However, there is a clear danger of overfitting to the training data, so one should be cautious with this. Similarly, one can imagine including a more granular division of train types, geographies, and/or time windows, but one still needs to be mindful of the costs of adding more dimensions and granularity. It is a fine balance, and we believe that the developments outlined here will both yield a new standard approach that can be used in a variety of countries and case studies, and a path forward for even further developments.

6.1.3. Results – experimental design

This section outlines the results of the experimental design that we have chosen for the next few steps of calibration and validation. This includes allowing different scaling factors for entry, run, and dwell delays. The distributions are generated and reduced separately for the different types of trains, but using the same parameter settings. Any geographical differences are handled by separate simulations and calibration procedures, rather than allowing the parameters to be different and calibrated jointly inside one simulation. Time windows are not considered.

We land on what is called a central composite circumscribed (CCC) design. This uses five different settings, which are combined in different ways, outlined in Table 7 below.

Table 7. A Central Composite Circumscribed (CCC) design for an experiment with three factors. From NIST (2012).

Repetitions	X_1	X_2	X_3
1	-1	-1	-1
1	1	-1	-1
1	-1	1	-1
1	1	1	-1
1	-1	-1	1
1	1	-1	1
1	-1	1	1
1	1	1	1
1	-1.682	0	0
1	1.682	0	0
1	0	-1.682	0
1	0	1.682	0
1	0	0	-1.682
1	0	0	1.682
6	0	0	0

The settings are based on a central point, indicated by 0, two intermediate levels indicated by 1 and –1, and then two extreme settings indicated by 1.682 and –1.682. In the experiments, we set that central point at 30% primary delays, the intermediate distance to 10%, and the extreme thus 16.82%. This means that the five parameter settings we use are: 13.18%, 20%, 30%, 40%, and 46.82%. Our choice here has been informed by our earlier experiments in this area, but with time and further experience different settings can of course be used. The experimental design with parameter settings is summarised in Table 8 below.

Table 8. Parameter settings in the experimental design.

Repetitions	Entry	Run	Dwell
1	20%	20%	20%
1	40%	20%	20%
1	20%	40%	20%
1	40%	40%	20%
1	20%	20%	40%
1	40%	20%	40%
1	20%	40%	40%
1	40%	40%	40%
1	13.18%	30%	30%
1	46.82%	30%	30%
1	30%	13.18%	30%
1	30%	46.82%	30%
1	30%	30%	13.18%
1	30%	30%	46.82%
6	30%	30%	30%

After the simulations are performed, a table like the one in Table 9 is filled out. E indicates the deviation between the simulated and empirical punctuality for train category C1, C2, C3, etc, (such as passenger or freight trains). The error is raised to the power of two, the average of these values is taken, and the root of that average is taken to get the RMSE. The parameter values and RMSE are then used together in the next stage.

Table 9. Data entry table after simulations are performed. Yellow fields are results of the simulations that are used to calculate the values in the green fields.

Simulation	Entry	Run	Dwell	E_{C1}	E_{C2}	E_{C3}	RMSE
1	20%	20%	20%				
2	40%	20%	20%				
3	20%	40%	20%				
4	40%	40%	20%				
5	20%	20%	40%				
6	40%	20%	40%				
7	20%	40%	40%				
8	40%	40%	40%				
9	13.18%	30%	30%				
10	46.82%	30%	30%				
11	30%	13.18%	30%				
12	30%	46.82%	30%				
13	30%	30%	13.18%				
14	30%	30%	46.82%				
15	30%	30%	30%				
16	30%	30%	30%				
17	30%	30%	30%				
18	30%	30%	30%				
19	30%	30%	30%				
20	30%	30%	30%				

Once Table 9 has been filled out, a linear regression will be trained on it with the following specification:

$$Y = c + \beta1*\text{Entry} + \beta2*\text{Run} + \beta3*\text{Dwell} + \beta4*\text{Entry}*\text{Entry} + \beta5*\text{Entry}*\text{Run} + \beta6*\text{Entry}*\text{Dwell} + \beta7*\text{Run}*\text{Run} + \beta8*\text{Run}*\text{Dwell} + \beta9*\text{Dwell}*\text{Dwell} + \varepsilon$$

At first, we will try to do this with the aggregated data from all simulation cycles within each simulation. In this way, there will be 20 observations. Given that the model has 10 parameters, there are enough degrees of freedom to estimate it. However, it is unlikely that the regression will be able to show whether the different parameters are statistically significant or not.

If this becomes an issue, it would be possible to run the regression on the disaggregated output, with each simulation cycle used as an observation. With an estimated 300 cycles per simulation setting, we would then have 20 * 300 = 6,000 observations, which is certainly sufficient to estimate the model and to determine which effects are significant.

The next step is to solve the equation with the estimated coefficients. The aim is for the Y-variable in the equation, defined as the RMSE between the observed and simulated punctuality levels, to be as close to zero as possible. This is achieved by altering the parameter values for Entry, Run, and Dwell delays. Finally, a validation simulation is done using these parameters.

6.2. Demonstration 9.3: Historical data processing

6.2.1. Railsys / Denfert

The RailSys modelling tool is widely used by WP8/WP9 participants, including ProRail, NSR, SNCF and Trafikverket, as well as several affiliated research institutes.

This tool has the advantage of being very powerful and used as a common tool in microscopic railway simulations across Europe. Despite its relative complexity and the significant training required to use it effectively, RailSys has established itself as the reference tool for microscopic studies for WP8/WP9 participants.

It should be noted that SNCF uses a version of RailSys developed specifically for France, called Denfert. This version is very similar to the native version, with a few additions of functionalities linked to French signalling (permissiveness of certain block signals, and free track margin called *khi*).

On this basis, the participants in WP8/WP9 defined a number of developments to the RailSys software that would benefit them jointly, depending on their current and projected use of the tool. These RailSys development needs are defined in Deliverable D8.3 (see, FP1–MOTIONAL, 2025).

Although the contractual and funding arrangements for these developments have not yet been defined, it appears that they will facilitate the use of the tools for microscopic simulations.

Most of these developments concern the simulation module. Indeed, it is this module that requires the most refined use to obtain quality results.

The first expected development concerns the ability of trains to run ahead of their theoretical timetable in simulation. Today, the RailSys software allows simulated trains to run only according to their theoretical timetable or else late. However, in reality, trains sometimes run ahead of schedule, in particular because of the margin consumed by running faster than the calculated running speed (although still less than or equal to the speed limit). This early running can have repercussions on the running of following trains (possible restrictive indication) and can lead to effects on the line's flow or capacity. As part of the development of ATO, it is particularly important to be able to measure the impact of ATO-equipped trains on line flow/capacity. In particular, it appears that trains equipped with ATO will often run ahead of their theoretical timetable, and that they will have to continually recalculate their target speed in relation to their updated running time, based on information transmitted by the TMS.

With this in mind, the RailSys tool should include the possibility of simulating trains running ahead of their theoretical timetable to improve the accuracy of simulations.

As described in the Deliverable D8.3 (FP1–MOTIONAL, 2025), another desired functionality consists to keep the order of the trains fixed according to the schedule during simulation. This should be valid at unification tracks and at switch areas in the network. In the Netherlands timetable stability and robustness is investigated without the influence of traffic control. This means that disturbances can happen during the simulation, but dispatching is not permitted. It needs to be prevented that a train is going first, which was initially planned second, so the second train gets secondary delays from the first train while waiting. In addition, in general the interlocking system in the Netherlands keeps the order of the trains fixed according to the schedule, and that can be manually adjusted by the dispatcher if needed. Therefore, to have realistic results of the simulation runs that can be validated using realisation data, it is desirable that RailSys enables the option to keep the order of trains fixed according to the schedule.

To conclude, it appears that the desired developments of the RailSys tool (particularly in the simulation module) are useful to give more precise results in the microscopic simulation studies. These features will be particularly recommended in the context of developing new technologies for railway operations, such as ATO.

Despite these desired tool developments in RailSys, the 9.3 demonstration focuses on processes and add-ons outside the RailSys/Denfert tool. Indeed, SNCF would like to be able to envisage a generic routine for conducting stochastic tests, without being tied to a specific tool. This consideration remains central in defining the objectives of demonstration 9.3: today, RailSys/Denfert remains SNCF Réseau's reference tool, and the demonstration will be carried out with the help of this tool. However, SNCF intends to capitalise on this demonstration to envisage a more generic stochastic tests routine.

The illustration in Figure 18 represents the framework and implications of demonstration 9.3.

The aim of demonstration 9.3 is to demonstrate a method for processing the historical data and implement the delay distribution into RailSys/Denfert for stochastic models. There are three main objectives:

- Improve the quality of regularity data processing (historical data). At SNCF Réseau, time differences of less than 5 minutes are not documented in the regularity reporting tool (Bréhat/ORE). The first step is to "reconstruct" the original incidents on the basis of existing regularity data and additional data (track circuit occupancy for instance)
- Better calibration of the "background noise" to improve the performance of stochastic tests. This involves finer parameter setting, or even inputting parameters from an external routine.
- Identify development needs for simulation in RailSys: trains running in advance, regulation rules, etc.

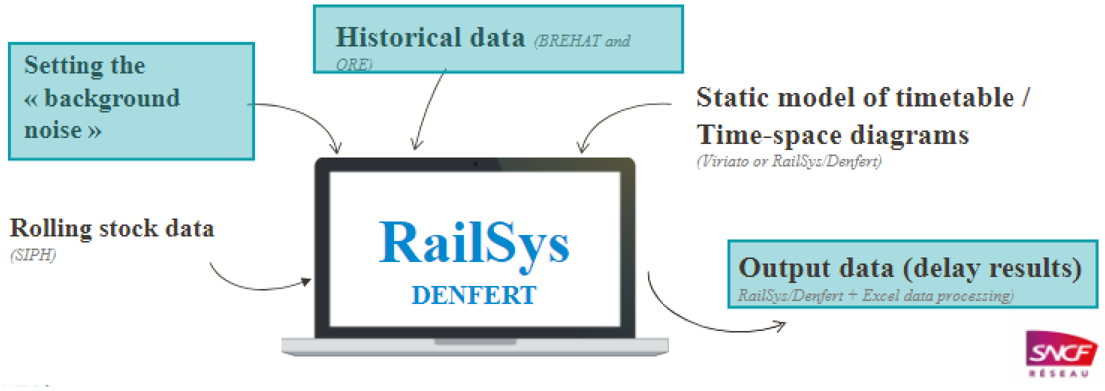

Setting the « background noise »
Historical data (BREHAT and ORE)
Static model of timetable / Time-space diagrams
(Viriato or RailSys/Denfert)
Rolling stock data
(SIPH)
RailSys
DENFERT
Output data (delay results)
RailSys/Denfert + Excel data processing)
SNCF RÉSEAU

Figure 18. Framework and implications of demonstration 9.3.

6.2.2. Main steps of demonstration 9.3

The demonstration 9.3 will be phased in 5 steps. Only the phasing is presented here, the results of these different phases will be detailed in WP9. The 5 steps are:

- Step 1: Identify and retrieve available data (ATESS, Bréhat, GPS, Mistral, …). Retrieve the data on the EOLE perimeter. Contributors: SNCF D2D ICODEV & DGEX INCA EEx
- Step 2: Interpret the input data to display regularity curves by line. Deliverable: application or Power Bi to process the data and display the results. Contributors: SNCF D2D ICODEV & DGEX INCA EEx.
- Step 3: Distinguish between Origin Incidents (OI) and Induced Incidents (II). Deliverable: algorithm that analyses the data and isolates unit disturbances according to certain criteria. Academical partnership: Lund University (Sweden). Contributors: SNCF D2D ICODEV & DGEX INCA EEx
- Step 4: Inject the primary perturbations into the background noise and calibrate it under RailSys/Denfert. Use of Denfert scripts to inject disturbances automatically. Need to develop the RailSys/Denfert software to make it easier to change parameters. Contributors: TRV (collaboration on the RailSys scripts) & SNCF DGEX INCA EEx
- Step 5: Faster reading of results and improved iterations to converge on a calibration model Internal evaluation module, looking for academic collaboration for data analysis

6.2.3. Case study of demonstration 9.3

EOLE – acronym for Est-Ouest Liaison Expresse – is the first incarnation of the future 'high performance' network and is a key piece for the development of everyday transport in Île-de-France.

The EOLE project, declared of public interest on January 31, 2013, aims to extend a suburban train line (called RER E) to the west. After its entry into service, it will benefit more than 650,000 passengers daily, within employment areas bringing together more than 2 million workers.

The new RER E will be the most interconnected line in Île-de-France, between Tournan/Chelles to the east (Seine-et-Marne) and Mantes-la-Jolie to the west (Yvelines), passing through Paris and La Défense (Hauts-de-Seine). A map of this line is shown in Figure 19. The EOLE project notably includes the modernization or creation of 55 km of tracks (including the digging of 8 km in tunnels), the modernization or creation of 14 stations (including 3 entirely new stations).

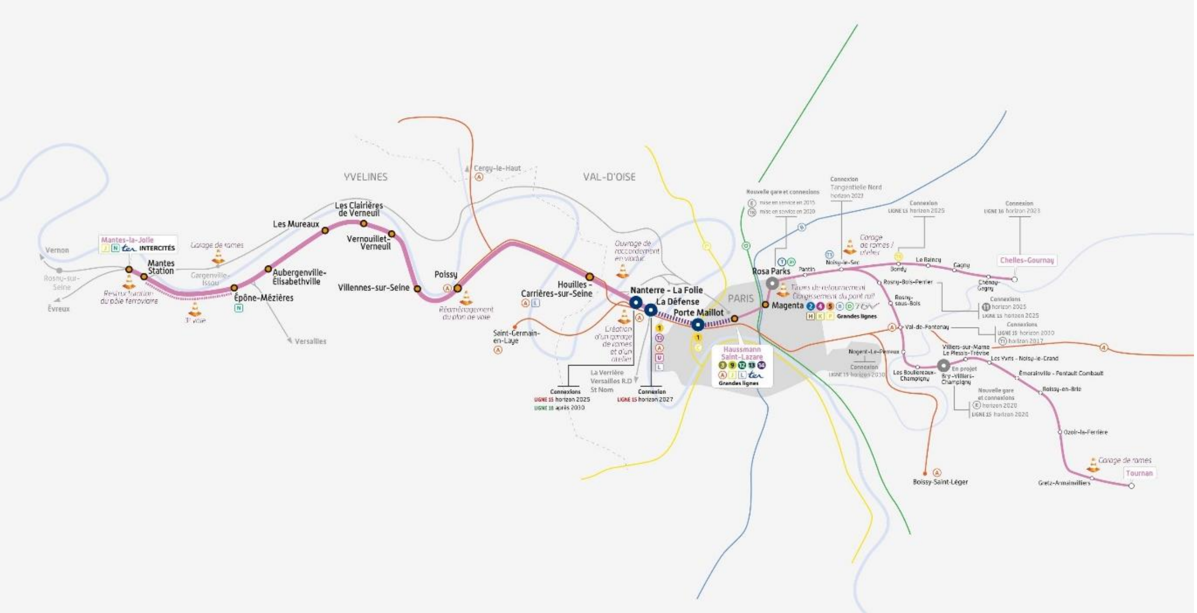


Figure 19. Map of the new Île de France east-west suburban train line (RER E).

This project introduces a whole new generation of RER rolling stock (the RER NG), with a particular operation in the central section (Rosa Parks/Nanterre) known as overlapping operation, an Eastern line (Chelles/Tournan <-> Nanterre) and a Western line (Mantes la Jolie <-> Rosa Parks) which are overlapped between Nanterre and Rosa Parks and the commissioning of NExTEO, a new train control and supervision system which offers automated assistance to driving and traffic regulation, and allows run more trains and faster.

This case study is chosen by SNCF for demonstration 9.3 because many stochastic studies are currently being carried out there. It will be easier to compare the results in order to validate the demonstration. In addition, SNCF Réseau has more accurate historical data on this perimeter, given that the signalling stations are equipped with more modern technology on average than on the rest of the French network.

6.2.4. Description of the available data

Three main sources of data will be used to carry out the previously presented demonstration steps:

- ATESS: "train black boxes", these are data from the operation of the rolling stock. They record the speed according to the position of the train as well as the appearance of the signals encountered. A specific internal SNCF tool makes it possible to consult this anonymized data and to deduce the driving behaviours applied in a particular area.
- Mistral data: from new generation signalling boxes developed by Alstom, these data are "images" of the man-machine interface visualized by signalling agents. This data makes it possible to know the state of track circuits and signals at the time of train movement. By cross-referencing them with localisation data, it is possible to reconstruct the appearance of the signals encountered by drivers.

These Mistral data were collected for the entire perimeter of the RER E east of Paris for a period of 6 months. Processing of this data will be necessary to extract the desired information.

- Bréhat regularity data: this data comes from the SNCF delay database. They contain the time differences at notable points on the network (by projection of the times measured on locators). For delays of more than 5 minutes, the time difference observed is linked to an incident, and the cause of this has been entered manually. On the other hand, for delays of less than 5 minutes, the cause is not provided by the reporting agent.

6.2.5. Context and issues

SNCF is working to improve its operational study processes, particularly its stochastic tests. To this end, it wishes to develop a data visualisation tool in the form of graphs for analysing historical regularity data. These data are then used, within the framework of stochastic studies, to predict the robustness of projected transport plans. For each study carried out, different scenarios are tested, each time generating a series of results that need to be compared.

The challenges are to:

- Save time on data processing, analysis, comparison and presentation of results in a single report capable of evolving as and when required.
- Explore new, innovative solutions for operational studies.
- Standardise and share practices with the various members of the Motional program.

The aim here is to develop a tool that can:

- Automatically extract Bréhat regularity data via a simplified interface
- Retrieve delay data (output data) from RailSys/Denfert
- Display regularity data in graphical form
- Be able to identify ‘sick’ trains (trains subject to long delays that are not consistent with the service provided)
- Be able to estimate the number of simulations required to obtain reliable results

For the initial development of the tool, SNCF will concentrate solely on the EOLE study currently underway, the temporal and spatial limits of which will be specified later.

6.2.6. Geographic scope and time horizon

The study will be carried out within the EOLE perimeter, comprising the following lines:

- Line number 340000 between Paris-Saint-Lazare and Vernon
- Line number 396000 between Nézel-Aulnay and Mantes-la-Jolie
- Line number 366000 between Mantes-la-Jolie and Bréval

On these 2 lines, only the following traffic is considered:

- RER E / J5 Les Mureaux/Mantes as a reference
- J5 Vernon

- N Mantes
- Normandy TERs running between Paris and Mantes-la-Jolie

As traffic between Houilles - Carrières-sur-Seine and Achères is tubed, traffic on lines A and L should not be taken into account.

The desired time horizon is from June to December 2023 for the recovery of historical data.

6.2.7. Functionality requirements

SNCF is seeking for a tool capable of performing 5 main functions, that are shown in Figure 20:

1) Integrate input data (Excel format):

- Recovering real data from databases (Bréhat, etc...)
- Retrieve simulation data from RailSys/Denfert

2) Integrate user data via repositories (Excel format):

- List train groups (numbers/group/parity)
- List station sequences
- List the days studied (to filter real data)

3) Process the data:

- Aggregate data in a similar format within the same database
  - Link trains from different sources
  - Associate groups of trains with a sequence of stations
- Calculate regularity data for arrivals at and departures from stations
  - Calculate delays using the arrival and departure times of the theoretical and actual delays provided by Denfert and not using the delay times already calculated (unreliable data).
  - The delay calculation methodology should be differentiated if trains stop or pass through a station.
- Filter trains from actual data according to day reference

4) Display and filter data:

- Display the evolution of the delay on the line route
  - Filter display by data (Bréhat/simulated/combined)
  - Filter display by train group
  - Filter display by time period (start time/end time)
  - Filter display by delay threshold
  - Filter display by station
  - Display limits (actual curve +X% or -X%)
  - Exclude simulations
- Display a 'scatter plot' type graph to evaluate sick trains or poorly simulated trains (median delay/standard deviation of delays)
  - Filter display by data (Bréhat/simulated/combined)
  - Filter display by time period (start time/end time)
  - Filter display by simulation
  - Filter display by train group
  - Filter display by train

- We want a display with cascading filtering: initial display of the point cloud according to simulations, from which you just need to click on the point of interest to access a display by groups of trains within the same simulation, etc...

- Display a graph showing the convergence of simulations to assess the number of simulations required (no. of simulations/average delay).

5) Ensure that the tool is efficient:

- Be able to display results in less than 3 minutes, including data loading

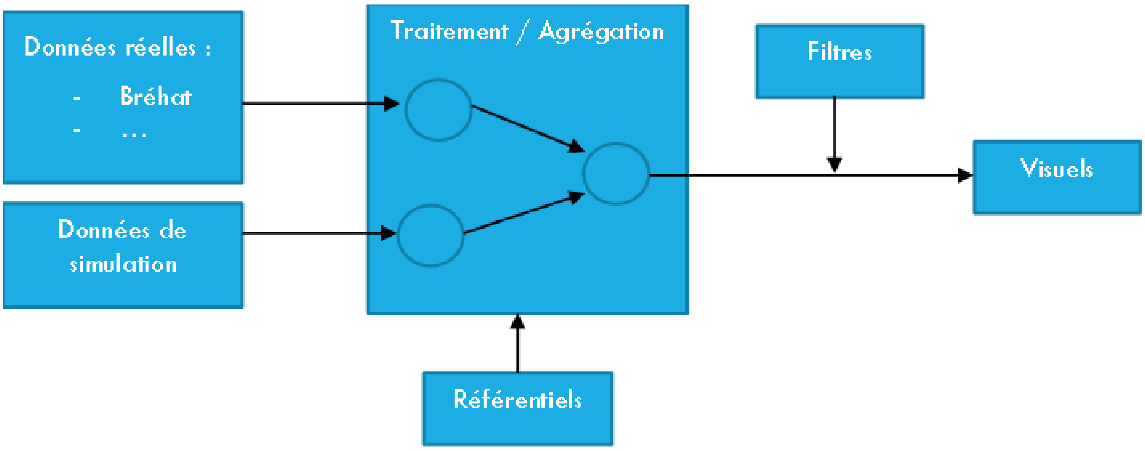


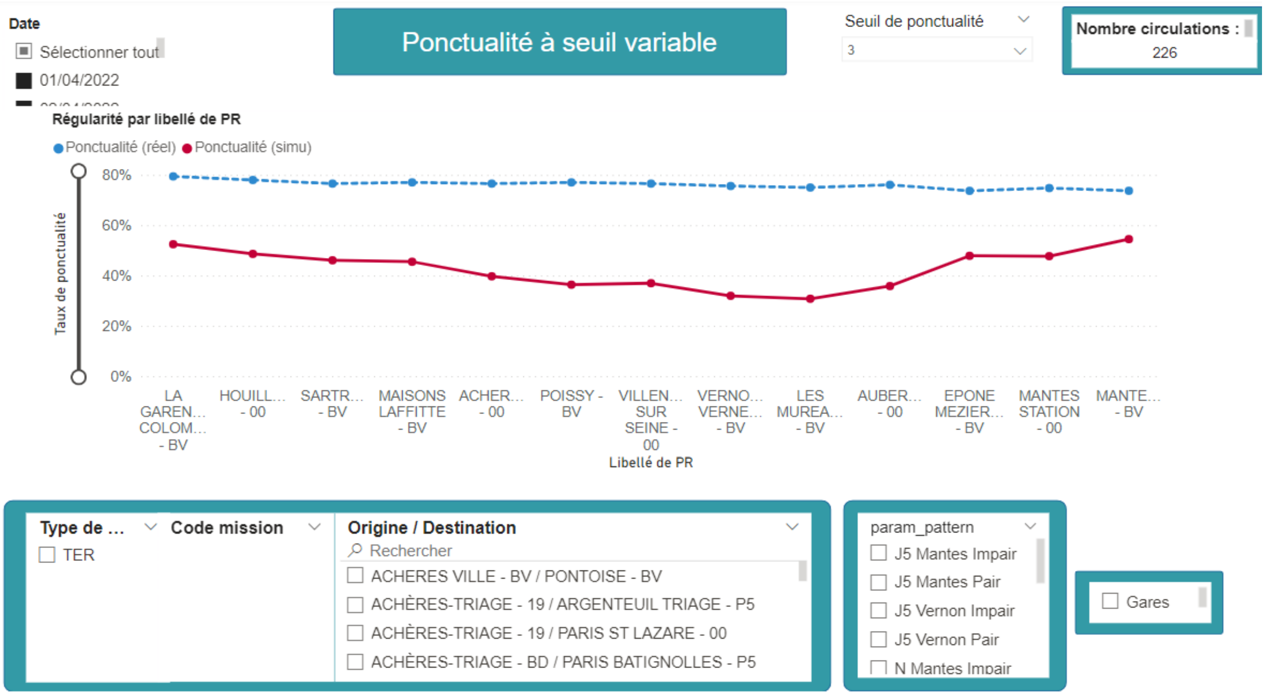
Figure 20. Mapping of functionalities for the developed tool.

Development will be carried out using a Power BI-type tool. It may be necessary to develop an Excel macro to facilitate data integration.

6.2.8. Output products

A Power BI tool meeting the specifications is required, together with instructions for use. The tool will consist of several tabs displaying all the features requested in the previous point.

The following illustrations (Figure 21, Figure 22, Figure 23) are mock-ups given as an indication.



Date

□ Sélectionner tout

■ 01/04/2022

Ponctualité à seuil variable

Seuil de ponctualité: 3

Nombre circulations : 226

Type de …

□ TER

Code mission

Origine / Destination

Rechercher

□ ACHERES VILLE - BV / PONTOISE - BV

□ ACHÈRES-TRIAGE - 19 / ARGENTEUIL TRIAGE - P5

□ ACHÈRES-TRIAGE - 19 / PARIS ST LAZARE - 00

□ ACHÈRES-TRIAGE - BD / PARIS BATIGNOLLES - P5

param_pattern

□ J5 Mantes Impair

□ J5 Mantes Pair

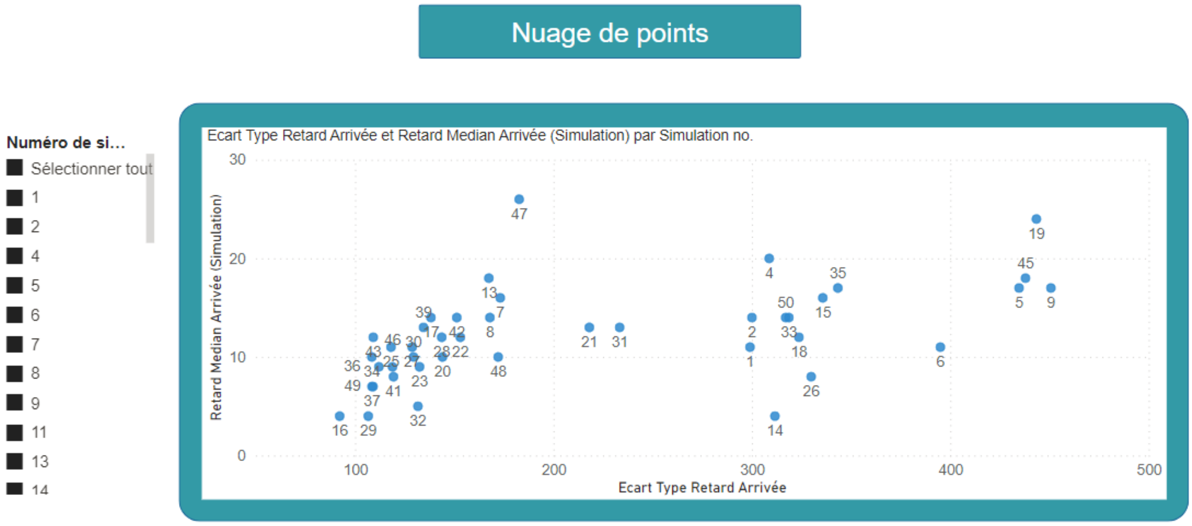
□ J5 Vernon Impair

□ J5 Vernon Pair

□ N Mantes Impair

□ Gares

Figure 21. Distribution of delays.



Nuage de points

Numéro de si…

■ Sélectionner tout

■ 1

■ 2

■ 4

■ 5

■ 6

■ 7

■ 8

■ 9

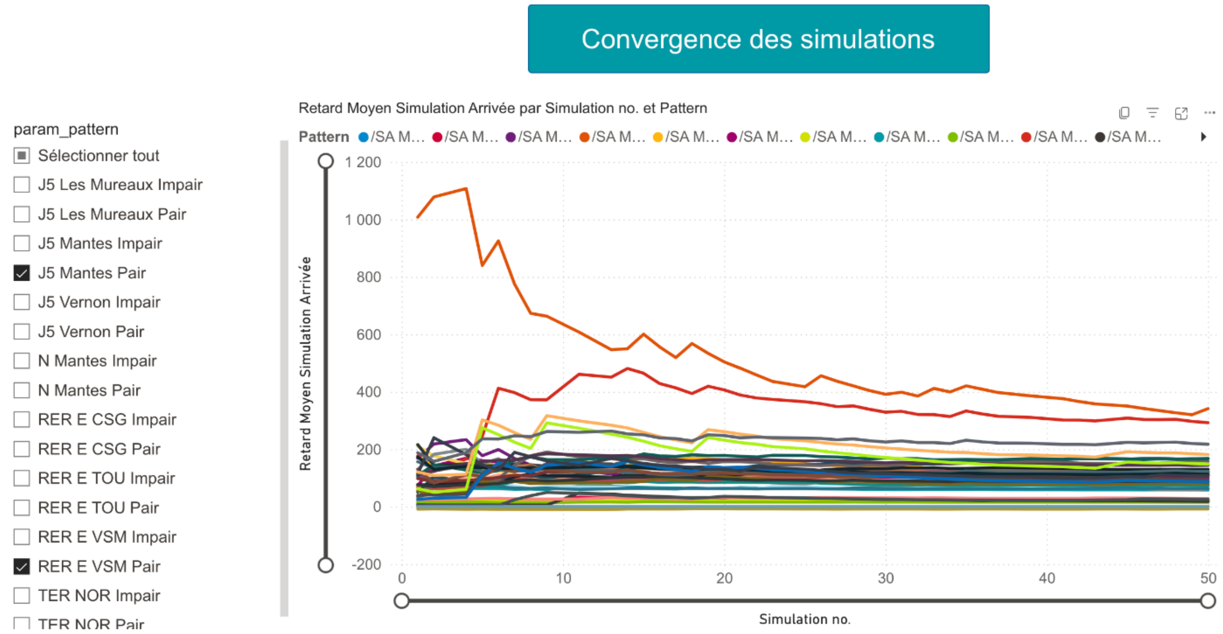
■ 11

■ 13

■ 14

Ecart Type Retard Arrivée et Retard Median Arrivée (Simulation) par Simulation no.
Retard Median Arrivée (Simulation)
30
20
10
0
100
200
300
400
500
Ecart Type Retard Arrivée
47
19
4
35
45
13
7
50
15
5
9
39
17
42
8
21
31
2
33
46
30
43
25
27
28
22
18
6
1
36
34
20
48
49
41
23
26
37
16
29
32
14

Figure 22. Identification of “sick trains”.



Convergence des simulations

param_pattern

- ▣ Sélectionner tout
- □ J5 Les Mureaux Impair
- □ J5 Les Mureaux Pair
- □ J5 Mantes Impair
- ☑ J5 Mantes Pair
- □ J5 Vernon Impair
- □ J5 Vernon Pair
- □ N Mantes Impair
- □ N Mantes Pair
- □ RER E CSG Impair
- □ RER E CSG Pair
- □ RER E TOU Impair
- □ RER E TOU Pair
- □ RER E VSM Impair
- ☑ RER E VSM Pair
- □ TER NOR Impair
- □ TER NOR Pair

Retard Moyen Simulation Arrivée par Simulation no. et Pattern
Pattern ●/SA M… ●/SA M… ●/SA M… ●/SA M… ●/SA M… ●/SA M… ●/SA M… ●/SA M… ●/SA M… ●/SA M… ●/SA M…
Retard Moyen Simulation Arrivée
1 200
1 000
800
600
400
200
0
-200
0
10
20
30
40
50
Simulation no.

Figure 23. Simulation’s convergence.

7. Conclusions

This deliverable D8.2 was developed to address the critical challenge of improving feedback loops between railway planning and operations. As rail networks become more complex and prone to operational disturbances, enhancing traffic simulation methods is essential for more accurate capacity and punctuality assessments, thereby improving overall planning quality.

The primary issue addressed is the gap between planning and operations, where inadequate feedback mechanisms lead to inefficiencies in capacity utilization and traffic management. The objective of this task was to refine simulation methods and tools, enabling more accurate feedback from operations to planning.

The methodology involved using advanced simulation tools such as PROTON, RailSys (and its French version called Denfert), SISCOG, CADIE and a CAF tool, applied across multiple real-world use cases in Sweden, the Netherlands, France and Spain. These tools were supported by methods for calibrating simulations with historical data, improving delay distribution models, and integrating timetable, rolling stock, and crew simulations to evaluate planning robustness.

The findings show the successful application of PROTON for large-scale simulations in Sweden, and the effective use of the CAF tool, RailSys/Denfert, CADIE and SISCOG in cases across Spain, the Netherlands, and France. Enhanced methods for data processing and model calibration have significantly improved the accuracy of these simulations. However, challenges remain, particularly in reducing simulation setup times and refining the handling of diverse operational disturbances.

Among the tools, PROTON and RailSys provided the best solutions for simulating large networks, while SISCOG's integration of crew and timetable simulations contributed to improved planning feedback. Despite these successes, limitations around handling more complex disturbance patterns and setup efficiency were noted.

The developments presented in this deliverable are a significant first step toward improving railway simulation tools and feedback loops between planning and operations. The developments have been tested and validated according to TRL4. However, these advancements are still in progress and will be further refined in the next phase of the project, specifically within WP9. This deliverable serves to introduce the initial developments, laying out the structure, methodology, and objectives of the demonstrations. The next deliverable, in WP9, will focus on showcasing the results of these continued improvements, providing a deeper evaluation of the tools' effectiveness and their impact on planning and operational processes. Future work will concentrate on refining these methods, improving simulation accuracy, and ensuring that the tools developed can be applied effectively across different railway networks, with continued integration and validation in real-world use cases.

8. References

Abbink E. J. W., Albino L., Dollevoet T., Huisman D., Roussado J., Saldanha R.L. *Solving Large Scale Crew Scheduling Problems in Practice*, 2011, Public Transport, 3(2): 149–164.

Borndörfer R., Klug T., Lamorgese L., Mannino C., Reuther M., Schlechte T. (Eds.). *Handbook of optimization in the railway industry* (Vol. 268), 2018, Cham: Springer.

Botte M., D'Acierno L. *Dispatching and rescheduling tasks and their interactions with travel demand and the energy domain*, 2018, Models and algorithms. Urban Rail Transit, 4(4), 163-197.

FR8RAIL II. *Deliverable D3.4 Demonstrator on improved planning – verification of demonstrators,* 2021.

Hansen I., Pachl J. *Railway timetabling and operations: Analysis, modelling, optimisation, simulation, performance, evaluation*, 2014, Eurail press.

Heil J., Hoffmann K., Buscher U. *Railway crew scheduling: models, methods and applications*, 2020, European Journal of Operational Research, 283(2): 405-425.

Johansson I., Palmqvist C.W., Sipilä H., Warg J., Bohlin M. *Microscopic and macroscopic simulation of early freight train departures*, 2022, Journal of Rail Transport Planning and Management, Volume 21, Article ID 100295.
https://doi.org/10.1016/j.jrtpm.2022.100295

Minbashi N., Sipilä H., Palmqvist C.W., Bohlin M., Kordnejad B. *Machine learning-assisted macro simulation for yard arrival prediction*, 2023, Journal of Rail Transport Planning and Management, Volume 25, March 2023, Article 100368. https://doi.org/10.1016/j.jrtpm.2022.100368<s

FP1-MOTIONAL. *Deliverable D3.1 – Mapping against scope, specification of technical enablers, high-level use cases, high-level requirements, high-level design for demonstrators in WPs 4-9*, 2024a. https://www.ct5webapi.eu/api/documents/getfile?id=19bb4dd5-6a07-4abe-b714-1038f4cfa521

FP1-MOTIONAL. *Deliverable D8.1 – The need for future development of methods and models for capacity simulations and feedback loops between planning and operations*, 2024b.

FP1-MOTIONAL. *Deliverable D8.3 – Developed simulation methods and models for capacity evaluation of ETCS and C-DAS/ATO*, 2025 (to be published).

NIST. *Engineering Statistics Handbook - 5.3.3.6.1,* 2012, Central composite designs.
https://www.itl.nist.gov/div898/handbook/pri/section3/pri3363.htm

Pachl J. *Deadlock avoidance in railroad operations simulation*, 2011, in: Transportation Research Board. 90th Annual Meeting.
http://www.digibib.tu-bs.de/?docid=00037775

Pachl J. *Railway Timetabling and Capacity, Edition 1.0*, 2024, TU Braunschweig.
https://doi.org/10.24355/dbbs.084-202407090800-0

Palmqvist C.W., Johansson I., Sipilä H. *A Method to Separate Primary and Secondary Train Delays in Past and Future Timetables Using Macroscopic Simulation,* 2023, Transportation Research Interdisciplinary Perspectives. Volume 17, January 2023, article 100747.
https://doi.org/10.1016/j.trip.2022.100747

Zinser M., Betz T., Becker M., Geilke M., Terschlüsen C., Kaluza A., Johansson I., Warg J. *PROTON: A Macroscopic Monte Carlo Railway Simulation*, 2019, Paper presented at the 12th World Congress on Railway Research (WCRR), Tokyo, Japan.