Simulations with PROTON and RailSys

–

Use of a macroscopic and microscopic railway simulation tool in Swedish applications

KAJT project SIMPOR

Hans Sipilä

Stockholm 2023
1 Introduction

Within the European Shift2Rail research project, a macroscopic simulation tool was developed in the sub-projects Plasa and Plasa2. Development was done by DB Analytics, which is part of DB (Deutsche Bahn) and one of the main goals was that the tool should be able to simulate large networks in short computation time. Both DB and the Swedish Transport Administration (Trafikverket), along with several others participated in the Plasa and Plasa2 projects. The tool is named PROTON (Punctuality and Railway Operation Simulation), it was formerly known under the name PRISM (Plasa Railway Interaction Simulation Model).

Trafikverket has an intention of introducing PROTON as an in-house tool and to increase the use of simulation as a method for analysing for example proposed future timetables. Trafikverket is currently using RailSys which is a microscopic timetable and simulation software. The intention is to use macrosimulation in applications where microsimulation is impractical or infeasible to use, typically in large area or even network wide simulations.

SIMPOR is a project with the aim of using PROTON in different types of Swedish applications and it is carried out within KAJT (Kapacitet i järnvägstrafiken), Capacity in the Railway Traffic System which is a research program for improved railway system performance financed by Trafikverket. Most of the applications where PROTON has been used has been carried out in combination with other projects, such as FR8Rail2 and FR8Rail3, in which the use cases have been formulated. This report describes briefly PROTON and the input data needed for running simulations, and summarizes results from the different other projects where PROTON was used.

DB has further developed PROTON into a microscopic simulation tool, and it is still under development. However, no microscopic application with PROTON has yet been performed or tested in Sweden. Consequently, this report deals only with the macroscopic part of PROTON. The microscopic development of PROTON is briefly mentioned in section 5.

2 PROTON

Development of PROTON started in Shift2Rail. Infrastructure is represented by nodes and edges. Nodes represent operational control points, commonly referred to as stations, but they can also be subareas of larger stations. Arrivals and departures of trains are handled in the nodes. The nodes doesn’t include any kind of track layout representation, but it is possible to enable overtakings, i.e., a change in train sequence on double track lines or enable meetings and overtakings on single track lines in a specific direction. The nodes handle arrival and departure events for trains. Edges represent line track(s) between nodes and they have attributes such as, line identification, number of tracks, length, representative block section length, type of signalling and if electrified or not.

The timetable specifies the sequence of nodes and edges for each train, along with associated arrival and departure times including scheduled dwell times. Additionally there is
information about train ID (train number), train type (timing load) and available time supplements on edges which can be used for reducing delays. Technical minimum running times are needed for all relevant combinations of train types and edges. Each train type and edge has four basic drive modes given by the combinations of pass and stop on the respective nodes. In simulations, in case trains are required to do unscheduled stops on nodes the available technical driving times for the relevant drive modes set lower limits for the driving times used in the simulations.

Train conflicts are modelled based on minimum headway times, which are computed from block occupation times taking into account infrastructure edge properties and the running times for trains. In a microscopic tool the actual train movements are modelled and here the minimum headway is controlled by trains releasing routes (block sections), where after new routes can be set for subsequent trains. In the macroscopic PROTON, this can more be viewed as trains being moved from one state to another, thus observing the minimum headway time between the particular trains and events. Figure 1 shows the principle for headway.

![Figure 1: Example of departure headway in a microscopic and macroscopic tool.](image)

A set of different dispatching modes are implemented. One of the modes is based on train type priorities, another mode prioritizes punctual trains over non punctual trains where the limit for punctuality is 6 minutes, a third mode prioritizes trains on time within 60 seconds from scheduled time over trains outside of this interval. The functionality of PROTON is described in Zinser et al. (2019) and to some extent in Zinser et al. (2018) in which a Swedish use case is simulated both in macroscopic PROTON and in microscopic RailSys.

The necessary input data for PROTON is described in more detail in section 3, but figure 2 gives an overview of the different data sources that have been used in the Swedish applications so far. Timetables can be converted from either TrainPlan or RailSys, and in all simulations empirical (historical) delay data, commonly referred to as LUPP data, has been widely used to create the entry delay distributions as well as primary dwell and run time extensions.
3 Input data overview

When simulating with PROTON, different types of input data are required, the most obvious being an infrastructure and a timetable together with basic information about the train types used in the timetable. Figure 3 gives an overview of the different input (files) that can be given. For a deterministic simulation, i.e., without introducing any stochastic and/or systematic delay input, it will run only with infrastructure, timetable and train information. This requires that relevant options (flags) are set to off. If, for example, the option of using dwell time distributions is active, it requires the relevant input data for this. The following sections give a brief overview of different input data used in PROTON (macroscopic). The version used so far in Sweden is from September 2019 (0.12), therefore the information relates to this version. The node delay functionality was not implemented yet (see figure 3), but at least dummy input files are still needed if the option for modelling edge or node disruptions is active.
3.1 Infrastructure

The macroscopic infrastructure consists of nodes and edges. Depending on the granularity of the timetable, nodes will represent stations and other locations where trains stop or have a scheduling point (timing point). The infrastructure must contain all nodes and edges that appear in the timetable. Two example infrastructure parts are used to explain the macroscopic infrastructure with respect to its microscopic counterpart. Figure 4 shows schematically a double track section with four stations. Station A has side tracks on both sides of the main tracks, Station B is merely a place where passenger trains can have scheduled stops on main tracks, Station C has one side track and Station D only turnouts between the main tracks.

The microscopic infrastructure contain some typical objects that may occur. The corresponding macroscopic infrastructure has only nodes and edges linking the nodes, both with a set of attributes. Edges are defined for both directions, and the number of tracks attribute tells if it is a single track or double track line. Block length is typically an average line block length. An important attribute is the overtaking attribute, this tells whether the target node offers a possibility for trains to overtake each other in this direction. On a single track line this attribute also defines if trains can meet or not at a station (node). The macroscopic edges are only designed to model either single or double tracks per each unique line ID. For parts in the infrastructure containing more than two tracks (e.g. a quadruple track section), they cannot share the same line ID in PROTON and must be split on two or more line IDs.
Microscopic representation

<table>
<thead>
<tr>
<th>Regular node</th>
<th>Signal scheduled/dispatch</th>
<th>Stop board</th>
<th>Station border</th>
<th>Switch</th>
<th>Single-slip switch</th>
</tr>
</thead>
<tbody>
<tr>
<td>Signal</td>
<td>Shunting signal</td>
<td>Timing point</td>
<td>Speed board</td>
<td>Crossing</td>
<td>Double-slip switch</td>
</tr>
</tbody>
</table>

Macroscopic representation

An example is shown in figure 5. Here the microscopic infrastructure is a quadruple track between Station E and Station G. In Sweden, this type of layout would still have one line ID and this is also possible in RailSys. Normally, however, there is a system in which tracks are planned to be used by different train categories. In the example, stopping passenger trains would use the outer tracks since these have platforms, whereas maybe non stopping (faster) passenger trains would be scheduled on the inner tracks.

In a case like this, with originally a single line ID, the timetable to be simulated must reflect the line IDs used in the macroscopic infrastructure in order for trains to use the intended edges. This also affect multiple other input data files where line IDs are used as one of the attributes.

In addition to the already mentioned attributes, edges have attributes about whether electrified or not, whether a train protection system is active and some classification information. The macroscopic nodes does not contain much information, the most important one is the node name (station short name) since this needs to match with the node names used in the timetable. Additionally, full station names, coordinates and a few more attributes can be set.
Figure 5: Example of a microscopic infrastructure and its macroscopic counterpart in PROTON for a quadruple (4) track line with three stations. Macroscopic edges need to be split in two double track edges with different line IDs.

An important prerequisite to know is that PROTON is not modelling train movements inside stations, merely arrival and departure events to and from stations. Consequently there is no information about the station track layout, except if a particular station node can facilitate train overtakings and/or meetings as mentioned before. This information is given via the edges and thus is valid in the relevant direction and line ID. Figure 6 explains how this works.

Assume that station B has one siding located next to the main track that trains normally use in direction A–B–C, if the mode is left hand traffic. Turnouts exist so that this siding can also be reached by trains in direction C–B–A, however, this requires crossing the main track used by trains in opposite direction. This means that trains using the siding in direction A–B–C do not restrict simultaneous movements on the other main track. Trains using the siding in direction C–B–A restrict movements on the other main track since they need to cross it when entering and leaving the siding.

If overtakings are enabled at station B in direction A to B in PROTON, this means that the siding capacity is in practice infinite. In figure 6 this is visualized as having an infinite number of sidings. Similarly, if overtakings are enabled at station B for trains from C to B. Since the track layout is not modelled, this can be visualized so that the infinite number of sidings are located on the other side. Trains in direction A–B–C will not be restricted by trains arriving or departing at station B in relation to overtakings in direction C–B–A.
For these types of stations there is thus a choice to be made whether allowing overtakings or not in the other direction (i.e., C–B–A in figure 6). Allowing them can lead to an increase in the overestimation of station capacity, on the other hand, not allowing can underestimate the capacity. On single track lines this attribute should normally be active in both directions to facilitate meetings, if the station is used that way in reality.

![Actual microscopic track layout](image)

Figure 6: Principle for overtakings in a PROTON macro station node. No track layout is modelled, including turnouts, thus the over crossed turnouts between main tracks since crossing movements restricting other movements do not occur.

### 3.2 Timetable

Currently there are scripts developed for converting exported timetables from either RailSys or TrainPlan to PROTON timetable format. Table 1 shows an example of the format. There are some additional columns which are not shown in the example. A timetable is given for a particular date (operationDay). The original timetable can be constructed for a date sequence or a mix of different date sequences and individual dates, in a simulation only the trains with a matching operationDay are simulated for the relevant date. Each train ID needs to have an assigned train type string (trainType), which is inherited from the RailSys or TrainPlan timetable. Most of the delay distributions introduced through the different input files are allocated to trains by the train type string and location. The traffic type index partially resembles the train categories used in RailSys. These have in Sweden been divided into:
1. High speed and other long distance passenger trains
2. Regional trains
3. Commuter trains
4. Freight trains
5. Non revenue (empty) and service trains

In the simulations with Swedish use cases the input delay distributions have been broken down into either passenger and freight or according to the five traffic type indices. The \texttt{trainRunId} is merely an integer train ID which needs to be unique for the particular date in the same way as the string train ID. The station short names (\texttt{prevNodeName} and \texttt{nodeName}) together with the line ID:s (\texttt{lineIdIn} and \texttt{lineIdOut}) must of course be consistent with the infrastructure data. The supplement (seconds) is the available driving allowance which can be used for reducing delays and this must be given per edge. If the input timetable comes from RailSys, the easiest way is to set all stations to time reference nodes before exporting. The other solutions is to redistribute a supplement that spans over more than one edge to the respective edges in the conversion process.

In both TrainPlan and RailSys timetables there can be negative driving allowances, meaning that the scheduled driving time between two consecutive nodes cannot be met according to the train type performance. This should be avoided in the currently used version of PROTON in Sweden, since this will easily lead to stopping errors in a simulation. So far this has been achieved by adjusting the timetable in RailSys before export. If the timetable comes from TrainPlan there is a module in the conversion script that checks for this and redistributes allowance from previous and/or subsequent edges but avoiding having to change a scheduled arrival or departure time at a scheduled stop if possible. It can be pointed out that if using a sub part of an annual timetable in the national RailSys model, this originates from the TrainPlan system and the occurrence of negative allowances is partially inherited into RailSys.
Table 1: Example of PROTON timetable format showing most of the columns. Arrival and departure times are given in datetime format. Operation day controls which date the particular trains belongs to. The supplements can be used to reduce delays in a simulation.

<table>
<thead>
<tr>
<th>operation</th>
<th>train Type</th>
<th>train Run</th>
<th>train Run</th>
<th>traffic</th>
<th>index</th>
<th>arrival Time</th>
<th>departure Time</th>
<th>stop</th>
<th>prev Node</th>
<th>node Name</th>
<th>node Id</th>
<th>lineName</th>
<th>in Node</th>
<th>out Node</th>
<th>supple ment</th>
<th>dis tance</th>
<th>cumu lative Distance</th>
</tr>
</thead>
<tbody>
<tr>
<td>2019-03-06 PX2-2000</td>
<td>1 50 544</td>
<td>1 2019-03-06 16:04:59</td>
<td>2019-03-06 16:05:00</td>
<td>1 M</td>
<td>901aY</td>
<td>6</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2019-03-06 PX2-2000</td>
<td>1 50 544</td>
<td>2 2019-03-06 16:07:59</td>
<td>2019-03-06 16:07:59</td>
<td>0 MGB 901aY 901aY</td>
<td>23 2830</td>
<td>2830</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2019-03-06 PX2-2000</td>
<td>1 50 544</td>
<td>5 2019-03-06 16:11:19</td>
<td>2019-03-06 16:11:19</td>
<td>0 BLV SKM 522 522</td>
<td>2300</td>
<td>5130</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2019-03-06 PX2-2000</td>
<td>1 50 544</td>
<td>6 2019-03-06 16:11:36</td>
<td>2019-03-06 16:11:36</td>
<td>0 BLV ÅK 912 912</td>
<td>41 540</td>
<td>9390</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2019-03-06 PX2-2000</td>
<td>1 50 544</td>
<td>8 2019-03-06 16:14:07</td>
<td>2019-03-06 16:14:07</td>
<td>0 BLV FLP 912 912</td>
<td>3 1600</td>
<td>13110</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2019-03-06 PX2-2000</td>
<td>1 50 544</td>
<td>9 2019-03-06 16:18:00</td>
<td>2019-03-06 16:18:00</td>
<td>0 FLP LU 912 912</td>
<td>4 3330</td>
<td>16440</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2019-03-06 GR401610</td>
<td>4 1040 4613</td>
<td>1 2019-03-06 21:52:00</td>
<td>2019-03-06 21:52:00</td>
<td>0 HRBI 417 19</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2019-03-06 GR401610</td>
<td>4 1040 4613</td>
<td>2 2019-03-06 22:02:34</td>
<td>2019-03-06 22:02:34</td>
<td>0 HRBI SKMS 417 522</td>
<td>95 6770</td>
<td>6770</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2019-03-06 GR401610</td>
<td>4 1040 4613</td>
<td>5 2019-03-06 22:18:06</td>
<td>2019-03-06 22:18:06</td>
<td>0 ÂK LMO 522 522</td>
<td>8 5390</td>
<td>19560</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2019-03-06 GR401610</td>
<td>4 1040 4613</td>
<td>12 2019-03-06 22:46:00</td>
<td>2019-03-06 22:46:00</td>
<td>0 MDL ÂK 522 522</td>
<td>8 5090</td>
<td>33420</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
3.3 Dwell time variation

The modelling of dwell time variation differs between RailSys and PROTON. The same basic distributions can be used but these need to be shifted with respect to different reference times. Figure 7 gives an example of the differences. Assume that there is a distribution (histogram) describing observed dwell times (historical data). This can be for a specific station, direction and train category or an aggregation of data from multiple stations and/or direction and/or train categories. From the distribution and relating to the scheduled dwell time, a representative minimum dwell time can be observed by checking for the shortest measured dwell time or setting it on a low percentile of all measured dwell times.

In RailSys, trains can be assigned with a minimum dwell time at scheduled stops. Dwell time variation is modelled by adding a value drawn from a distribution to the minimum dwell time set in the timetable. This gives a new value and this will be the actual minimum dwell time for a specific train at a specific station and in the current simulation cycle. Other parameters and settings in the timetable together with the current dispatching situation will then affect what the actual dwell time will be, i.e., the resulting departure time. For example, normally passenger trains are not allowed to depart ahead of scheduled time while freight trains may sometimes be allowed to do that. The shifting of a historical distribution, originally configured so that zero would mean 0 seconds of dwell, is affected by the scheduled minimum dwell. If, for whatever reason, a lower minimum dwell is used in the RailSys timetable than what has been observed from a distribution, this needs to be accounted for in the shifted distribution.

![Dwell time variation diagram](image)

Figure 7: Handling of dwell time variation in RailSys and PROTON.
In PROTON, dwell time variation is modelled by adding a value drawn from a distribution to the scheduled dwell time, meaning that the distributions should relate the variation in dwell time relating to the scheduled dwell time. Starting with the same distribution of observed values (historical data), this needs to be shifted so that the scheduled dwell time ends up on the zero point. Thus, it will then be a histogram that can give both negative and positive values. A negative value means that the simulated dwell time can be shorter than scheduled, a positive value means longer than scheduled and zero means scheduled dwell time. The possible minimum dwell time in a simulation is decided by taking the maximum between the sum of scheduled dwell time and value drawn from distribution, and assigned minimum dwell time. As in RailSys, the actual simulated dwell time is also affected by the dispatching situation.

3.4 Other input data

This section gives a brief overview of the other input data (files), in addition to those already mentioned, that can be used depending on the application (figure 3). As also mentioned, the allocation of delay distributions on trains goes mostly through the train type parameter (string). This means that even if there would be just a few different distributions used, there is a lot of repetitive data in the files if many different train types are used in a timetable. In the simulation applications, the generation of input files is done by scripts so this poses no real problem.

Time window specification

This makes it possible to define of different time windows during a 24 hour traffic day, which enables that different delay distributions can be used depending on the time of day. For example, trains arriving into the simulation model which in reality start outside of the model area may have different entry delay characteristics depending on if it is peak, off peak etc. So far, this functionality has not been used in the Swedish simulation applications, all input delay distributions apply irrespective of time of day.

Initial/entry delays

All train types used in the timetable can be assigned an entry delay distribution impacting trains when they are initiated. This can be both at a true origin location or at the border of the simulation area for modelling trains entering the network from outside. In contrast to RailSys, which only handles perturbation values $\geq 0$, entry delay distributions in PROTON can have both negative and positive values. This means that trains can be initiated ahead of scheduled time. Entry delay distributions are assigned to trains by time windows, train types and locations. If there are trains initiated at a location using the same train type and there is a need to assign different distributions to these, then a copy or copies of this train type can be created and assigned alternative train type designations. Naturally, this change also needs to match with other relevant input data.

Dwell time delays

The difference between how RailSys and PROTON model dwell time variation is explained in section 3.3. Dwell time distributions are assigned to trains by time windows, train types and locations. Additionally, it can be defined for which scheduled dwell time interval a
particular distribution is valid. This enables that trains using the same train type can still be assigned different delay distributions if their scheduled dwell times differ. As with entry delays, there may be a need to create copies of train types with alternative designations if there is a need to, for example, use different distributions for different directions.

An associated input file is also needed for the dwell time variation, namely dwell time standard deviations. These are assigned only by train types, time windows and locations are not used. These are originally used so that if a train at a location with a scheduled stop draws a value from a dwell time distribution which would lead to a pre-plan departure, then a new value is sampled from a normal distribution with mean 0 and standard deviation according to file input and this value is used instead.

Minimum dwell times can either be set globally, one value for all train types and locations, or specified through file input in which case values are assigned by train types, locations and time windows. The values set lower limits for the possible dwell time reduction at stops.

**Edge and node delays**
Primary delays on edges can be assigned in two ways, either by pointing on an edge but without regarding train types (probability edge infrastructure) or by pointing on both an edge and a train type (probability edge train), both methods can be used simultaneously. Direction is controlled by giving from and to node (location) information, line ID is also needed. Time windows can be used for giving different probabilities and delay distributions depending on time of day. In the 'probability' files the probabilities for a delay and no delay are given. If a delay will occur, the associated delay distribution defined in the 'delay' file (delay edge) is used for sampling values. In the PROTON version used so far in Sweden (0.12), the distributions must be given as Weibull distributions. The principle is the same for assigning delays on nodes, but this was not completely developed in the currently used version, hence the red crosses in figure 3.

**Running time random deviations**
This gives the possibility to model smaller variations to the running times in simulations, e.g. driver variation, by giving parameters to normal distributions. These are assigned by edge, train type and line ID. This feature has not been used in the Swedish simulations, instead the distributions applied on edges described in the previous paragraph are intended to capture also this kind of variation.
Technical driving times
Technical driving times are provided to enable delay reductions in cases where the scheduled driving time exceeds the corresponding technical driving time, meaning that the scheduled driving time includes a supplement. Since train movements are not modelled by computing train speed profiles, as in a microscopic tool, this information needs to be provided as input. Technical driving times are typically given for each train type and edge (including direction) with four times representing the modes: pass-pass, stop-stop, stop-pass and pass-stop. In the Swedish simulations, the driving times have been provided to PROTON either by using exported driving time data from a system at Swedish Transport Administration or by using scripts and RailSys in combination to generate and compute all necessary times.

Train information
This file provides some basic parameters about the train types. All train types that occur in the timetable must be listed here in order for a simulation to run. The classification of train types to main train categories, translated into traffic type index integers, is specified here. Perhaps the most important information is the dispatching rank (priority), given as an integer, the lower the number the higher the rank. In RailSys, the reverse is used, the higher the number the higher the priority. If there is a need to assign different delay distributions on trains that are originally using the same train type, then copies of that train type can be created and given differing train type designations. Train information, as well as all other relevant files, need to reflect this so that all assignments and other data is matching.

Infrastructure modifications
Infrastructure edge modifications can be used to model restrictions in the infrastructure, typically related to maintenance activities. For edges (including the direction) there is for example possibility to limit a double track to one available track for both directions, i.e., temporary single track operation. Additionally, reduced speeds can be defined on edges, either partly or for the whole edge. Start and end of the restrictions are provided by setting dates and times (datetimes) for these.

4 Applications
This section describes briefly the Swedish applications where PROTON has been used. The idea and purpose for the applications is formulated within the respective projects. The work with the simulations can therefore be seen as a combination between SIMPOR and the other projects. In some of the applications comparisons are done between PROTON and RailSys, mainly by setting up as similar setups as possible in both tools including the delay distributions and checking simulation results. There are fundamental differences between the tools, mainly that one is macroscopic and the other microscopic, which should be considered when comparing simulation results. Despite this, it is still interesting to compare and, e.g., to investigate whether the same delay distributions can generally be used in both tools or whether they should be individually adapted (calibrated) for the respective tool.
4.1 PLASA2 – Macroscopic and microscopic simulations with and without freight train initiations ahead of schedule

The main aim in this study was to investigate the modelling of early freight train departures from origin stations in PROTON and RailSys (Johansson et al., 2022a). Results from both tools were also compared to empirical delay data. Generally in Sweden, freight trains show a large variance when it comes to deviations from scheduled times. Freight trains are also frequently departing ahead of their scheduled times, empirical data show that around 60% of the freight train departures are ahead of scheduled departure time from some yards.

In the microscopic tool RailSys, trains cannot be initiated ahead of scheduled time in a simulation, only on time or delayed is possible. Common practice in Swedish RailSys applications has therefore been to consider all empirical early departures to be on time when creating initial (entry) distributions for freight trains. Freight trains in simulations therefore have a smaller variance compared to real operations, largely as a consequence of this. Unlike RailSys, PROTON has the possibility of initiating trains ahead of schedule, meaning that the empirical departure distributions can pretty much be used straightforward and without modifications.

In this study, early freight trains in RailSys are modelled by time shifting all freight train schedules with 60 minutes and adjusting the empirical initial distributions so that they cover 120 minutes and the actual scheduled time is on 60 minutes. Figure 8 shows the principle for this in RailSys and also how it differs to the handling in PROTON. Basically the same initial distributions can now be used in PROTON and RailSys, the difference is that in PROTON they are given with a span from -60 to +60 minutes with 0 as scheduled time and in RailSys from 0 to 120 minutes with 60 as scheduled time. In RailSys, an adjustment of the dispatching priorities is also needed for freight trains due to the "faked" timetable, so that they will have their highest priority when they in the RailSys scope are around 60 minutes delayed, i.e., on time according to the actual timetable. The simulated network was between Hallsberg and Malmö, most of which is part of the Swedish Southern Main Line. The chosen timetable day was a normal Thursday in October 2016.
Empirical initial delay distribution (can be used straightforward in PROTON).

Deviation from scheduled time

Initial delay distribution adjusted for RailSys. Shares for departures ahead of schedule considered to be on time (added to 0).

Empirical initial delay distribution but shifted to work with the adjusted trains.

Figure 8: Principle for modelling early freight train initiations in PROTON and RailSys.

Calibration of delay distributions for dwell and run time extensions was done by using empirical data from years 2012–2017 and simulating without any reduction, using 1/3 reduction and 2/3 reduction. Delay distributions pulled from empirical data consist of a mix of primary and secondary (knock-on) delays. The aim is to approach the true primary distributions by reducing the empirical distributions in steps and compare the simulation results to real outcome. A high level of aggregation was used, dwell and run time extensions as well as entry distributions were divided into passenger and freight trains, as well as southbound and northbound direction.

Figure 9 shows the difference to empirical punctuality from simulating the three delay reduction cases in both PROTON and RailSys. Since the 2/3 reduction gave the closest match, this was used in further simulations. Punctuality is calculated by using arrival values from the trains’ last stations in the simulation and in empirical data given the scope of the network.

To get an idea of the impact on both freight and passenger train punctuality by allowing early freight train departures in the simulations, compared to not allowing early departures, both cases were simulated in the respective simulation tools. The difference (impact) in punctuality is shown in figure 10. Freight benefit clearly from allowing them to depart early, which was no surprise. Passenger trains get a small decrease in punctuality.
An overall conclusions from this study was also that given the similar setups in RailSys and PROTON, both tools resulted in output that is relatively similar considering that they operate significantly different. The level of aggregation was high in handling the input delay distributions. Distinction was only made between passenger and freight, and between southbound and northbound directions. However, the main point was not to compare to empirical data but to demonstrate the different approaches with freight trains and to estimate the punctuality impact, and at the same time also to some extent compare microscopic and macroscopic simulation tools.
4.2 FR8RAIL II – Simulation of infrastructure restrictions and disruptions

The aim in this part of the project was to demonstrate how simulation can be used to model infrastructure restrictions, e.g., due to scheduled maintenance or disruptions that affect the track availability. It was also formulated so that the test scenarios would be modelled both in PROTON and RailSys and results would be compared, the idea was to do a verification of the results from PROTON. This work is reported in Gestrelius et al. (2022).

Four scenarios were used and in two of them the reductions in infrastructure were modelled by using the 'Infrastructure modifications' input in PROTON described in section 3.4). Figure 11 illustrates the four simulated scenarios which are connected to restrictions or unavailability in track resources. Simulated network is the line between Hallsberg and Malmö, same as the one used in section 4.1 and the same timetable day is used as well.

Scenario 1 models single track operation (on the double track line) and speed reductions on the open tracks during night time. Scenario 2 models speed reductions on both tracks on the same edge during the whole day. Scenario 3 models a systematic delay for all trains that pass a certain location for a time period, e.g., a signal error. This is modelled so that all trains which are scheduled to pass this location within the specified time interval are given a systematic primary delay (with some variance). Scenario 4 models a one hour disruption (in both directions), this is done by assigning systematic edge delays to two specific trains passing an edge in different directions around the same time.

There is also a fifth scenario which demonstrates a way to model a performance change for two trains (e.g. due to change of vehicle types but without changing the schedule), where the trains cannot keep the scheduled times. This scenario is not reported here.
The conclusions regarding calibration of primary delay distributions in section 4.1 are used in this study as well. A couple of calibration simulations were made since some small infrastructure changes are done compared to the setup used in section 4.1. Figure 12 shows the empirical and simulated punctuality for PROTON and RailSys, divided into passenger and freight but with both directions combined.
Results from the scenarios explained in figure 11 are shown as differences in punctuality (percentage points) compared to the reference scenario without any of the restrictions. Only the trains that are deemed to be affected, based on scheduled times and some margin before and after, are selected and it is for these groups punctuality is calculated. As before, the punctuality is calculated at each train’s final station. Figure 13 shows how punctuality is affected in both simulation tools. The number of selected trains, on which the punctuality is calculated on, is shown on the x-axis with number of freight trains first and number of passenger trains second.

![Figure 12: Aggregated punctuality on 5 minute level, empirical and simulated from calibration runs.](image1)

Figure 13: Simulation results for selected train groups in scenario 1–4. Values show drop in punctuality on 5 minute level. The number of freight and passenger trains in the respective comparison groups are shown on x-axis (freight/passenger).

Based on the results from the simulated scenarios, the assessment is that it is possible to model these types of infrastructure restrictions or disturbances in PROTON and they are in line with the corresponding simulations in RailSys. Initially a check was made against available track maintenance data in order to possibly create scenarios based on this and be able to compare simulations to empirical data. However, no appropriate scenarios were found which would also fit the timetable period, therefore no comparisons are made against empirical data.
4.3 FR8RAIL III – Interconnection of a yard prediction model with a network macrosimulation model

This study presents a concept for connecting a Yard Prediction Model with network simulation. The Yard Prediction Model (YPM), which is based on machine learning, outputs departure deviations from a yard. Network simulation is then used, which takes part of its input from the YPM, to predict the arrival deviations to the next yard. The concept with applications is reported in Lidén et al. (2022) and in Minbashi et al. (2023).

The setup from the PROTON perspective is similar to figure 2. The difference is that some trains will get their entry delays from YPM data, instead of empirical aggregated data. YPM is implemented by using operational yard data from one year (2019) from Malmö marshalling yard. The focus in the simulations is on freight trains running from Malmö yard to Hallsberg yard, of which some get their entry delays from YPM. The network model represents the line Hallsberg–Malmö but including some additional lines in southern Sweden for capturing freight trains that are occasionally scheduled on a different route. Calibration runs are performed for different scaling levels on empirical distributions for a set of days to find a suitable scaling level and then use that in further simulations.

Unlike before, the timetable from 2019 is now used. Since the YPM has data for almost the whole year, network simulations are done on a timetable representing in total 32 weeks of operations, but split in 16 weeks from first half year and 16 weeks in second half year of 2019. The timetable is imported and converted from TrainPlan data. Each day is simulated separately but the simulations are run in a loop, day by day, week by week. Each day is simulated for 100 cycles, most trains draw values from the aggregated distributions of different types. The exception is the freight trains departing from Malmö yard and which have a value from YPM for the corresponding date. These trains will draw entry delay values from distributions which are designed to always give the YPM deviation value ±1 minute. This doesn’t automatically mean that the trains will depart on this particular time, instead it means that the trains are initiated in the simulation and the departure (dispatching) is then handled in the simulation with respect to the current traffic situation.

Figure 14 visualizes the principle with the connection of YPM and PROTON in a graphical timetable and with one freight train highlighted as an example. In the top figure, this particular train is using the aggregated entry delay distribution for freight trains in the northbound direction which results in a certain departure spread for this train.

In the bottom figure, which shows the same train and date, but this time with entry delay originating from YPM. This gives a small spread in the departures from Malmö yard, but the spread is instead built on along the route. As the train approaches Hallsberg yard, the spread is similar in both cases. In this particular example, the YPM departure deviation value is equal or very close to the scheduled departure.
Figure 14: Simulated outcome based on 100 cycles for one highlighted freight train on a particular date where initiation deviation comes from aggregated distribution (top) and YPM (bottom). For this train, scheduled path, median and percentile intervals are shown. Other trains shown with scheduled paths.

The conclusion made from this study is that the proposed framework with machine learning applied through a random forest algorithm to implement a yard departure prediction model and network simulation performs better than using only timetable and a basic machine learning arrival prediction model.
4.4 PMR2 – Punctual train traffic in the metropolitan areas

In this project the ideas from previous projects with different scaling levels on empirical distributions are extended by using a larger set of scaling levels and divide passenger trains into subcategories high-speed/long distance, regional and commuter trains, instead of having them all in one category. The projects can be divided into two parts.

The first part includes simulating the Swedish Western Main Line between Hallsberg and Gothenburg using a representative Thursday and Saturday as timetable (Johansson et al., 2022b). Similarly to what was done in section 4.1, freight trains were modelled in two scenarios. In the first scenario they can be initiated on time or delayed and in the second scenario also ahead of scheduled time. The scaling levels range between 5–100% in steps of 5%.

Results from this study indicates that the overall ratio between primary and secondary delays is that about 30% are primary and 70% secondary, although based on the simulations the ratio seems to vary significantly between different types of trains and operational conditions. Allowing freight trains to be initiated (depart from origin) ahead of schedule accounts for a marginal improvement of +0.5 percentage points to overall punctuality.

In the second part the rail network in the Swedish region of Skåne is simulated in PROTON (Palmqvist et al., 2023). For the reference case, a representative Tuesday–Thursday from the 2019 timetable is simulated with scaling levels applied on empirical distributions ranging from 0–100% in steps of 5%. The purpose is, as before, to find the level which gives the closest match to observed punctuality. The same procedure is also done by using a draft timetable (concept) for year 2025, which is the result from an investigation done by the Swedish Transport Administration in which different timetable concepts, among other things, were designed. In 2025, some of the currently ongoing infrastructure projects will be complete which enables an increase in traffic volume and flexibility. Figure 15 highlights the major improvements.

The 2019 timetable is converted from TrainPlan data, whereas the 2025 draft timetable is converted from RailSys. As can be seen in figure 15 there is a significant expansion of the 4-track section close to Malmö, meaning that the measures describing the modelling of track sections with more than two tracks had to be used to a greater extent (section 3.1). Since the 2025 input timetable is from RailSys, trains are already allocated to respective tracks and separate script is sued to modify all necessary input data needed for PROTON to reflect this.

Simulating both timetables and applying the 21 different scaling levels resulted in the overall punctuality values in figure 16. The corresponding empirical punctuality level for 2019 was 87%. One question asked in the study was by how much the current levels of primary delays would have to be reduced in order to meet the Swedish railway industry target of 95%. This will in the simulation happen if the scaling level is around 19%, which means that about half of the primary delays would have to be removed. The 2025 timetable has
a significantly higher traffic volume than the 2019 timetable, it is therefore not surprising, despite the infrastructure improvements, that the simulated punctuality values are lower compared to corresponding 2019 values.

Figure 15: Maps showing the simulated track network with single (1 track), double (2 tracks) and quadruple (4 tracks) track sections year 2019 and 2025.

Figure 16: Simulated overall punctuality values for scaling levels from 0–100% in 5% steps. Reference timetable 2019 and draft timetable 2025.

Assuming the scaling level for the 2019 timetable which gives the closest match to the observed punctuality, the simulated punctuality for the 2025 timetable is 82%, i.e., 5 percentage points lower than in 2019. To reach the target punctuality of 95% in the 2025 simulations, requires that a scaling level of around 13% is used, which means approximately a 2/3 reduction of primary delays from the observed level in 2019. One aim was also to check the locations and magnitudes of secondary delays in 2025. These were cal-
culated by processing the output data from PROTON. Figure 17 shows an example for passenger trains. Secondary delays are expressed both as total sum per cycle and mean values for trains, split into intervals and directions.

Figure 17: Simulated secondary delays for passenger trains per direction mapped on nodes. Mean values per train (top), summed values mean per cycle (bottom).

Figure 18 gives an example of simulated train runs in a specific cycle plotted in a graphical timetable. The scheduled train paths (dashed lines) are also shown for reference. This example from the 2025 draft timetable shows part of the West Coast Line and then further Lund–Malmö. Although it is difficult to draw general conclusions by just doing spot-checks on some cycles, these kind of plots are still useful and give an indication of how the trains are actually handled in a simulation. For special cases, e.g. when modelling infrastructure restrictions, they can quickly show whether what is meant to be modeled really also happens in the simulation.
Figure 18: Example of simulation outcome (from one cycle) shown in a graphical timetable. Solid lines are simulated trains and dashed are scheduled train paths.
5 Discussion

One aim of the project has been to investigate how PROTON can be used and for which types of studies it may be used for in a Swedish context. Additionally, to develop methodology for using PROTON in Sweden. The different type of applications which have been simulated show that, given the level of aggregation used in input data and in evaluating simulation output, the agreement with corresponding simulations in microscopic RailSys is relatively good. It may be expected that the more dis-aggregated output data is presented, the more the results from the tools will deviate, given that the modelling level is substantially very different (macroscopic vs. microscopic). The applications done so far in Sweden, have mostly been oriented on aggregated punctuality, other measures of performance has not been studied yet. PROTON seems to be a suitable tool for quickly modelling the impact from restrictions in track resources (e.g. track maintenance), but this is only the case for some types of restrictions and in cases where the lack of station layout modelling is not a big issue.

However, perhaps one of the main advantages with PROTON is it’s ability to simulate very large networks in short computation time. Even if there would be high availability of computing power for RailSys simulations, it will still be hard to perform stochastic simulations of large areas in the Swedish network due to the risk of occurrence of deadlocks. This is something that is more or less a general issue for microscopic railway simulation tools, and not only for RailSys.

The short computation speeds open up the possibility of simulating a large number of different scenarios. These can relate to calibration of delay distributions and other parameters, and validation, or it can be related to simulating multiple different timetables. Getting different timetables will require that there is a tool or that can generate timetable alternatives, this would open up possibilities of simulating many alternative timetables and assess their possible weaknesses and strengths.

In the beginning of the project, time was spent for creating different converters (scripts) for creating input data of different types for PROTON to be able to test Swedish scenarios. The PROTON version in Sweden is compiled in VisualStudio environment under Windows. PROTON is not a software with a user interface, instead it is run by running a script in which information is set for project folder paths, which dates to run and simulation parameters. This script will then invoke compiled PROTON libraries, and unless the input data is incomplete or has other inconsistencies the simulation will run and output data is produced. Before running the version of PROTON (0.12) that has been used in this project, all input data needs to be converted to R Data Serialization files (RDS). At first, the different conversions scripts used for manufacturing input data and taking care of output data consisted of Matlab scripts, but later all scripts are run in Python.

When there has been errors and PROTON stops either in the preparation step before actual simulations or during simulations, the time spent to find what exactly is wrong in the input data has varied a lot. Some errors have been fixed quickly, whereas others have taken
quite some time to find the root cause of them and then implement fixes in the conversion scripts. Also, when processing TrainPlan data, inconsistencies that might not have shown up in some cases have instead appeared in other cases.

Newer versions of PROTON are designed to be run in Linux environment. Tests have been made in Sweden to compile (build) PROTON in Windows but has not succeeded. Trafikverket is currently setting up an IT solution with Linux so that a newer version of PROTON can be compiled there. For the last two years DB has implemented microscopic modelling into PROTON, most of the development of PROTON has been put on the microscopic part and this is now the main mode. Work has started to convert Swedish microscopic infrastructure for PROTON, the data source used is the Swedish national RailSys model which is maintained (kept up to date) by Trafikverket. The aim is to have the possibility of using both macroscopic and microscopic PROTON in the future for Swedish applications.

References


Palmqvist, C.W., Johansson, I., Sipilä, H., 2023. A method to separate primary and secondary train delays in past and future timetables using macroscopic simulation. Transportation Research Interdisciplinary Perspectives 17, 100747. URL: https: