

An efficient evaluation scheme for KPIs in regulated urban train systems

Bruno Adeline¹, Pierre Dersin¹, Éric Fabre², Loïc Hérouët², Karim Kecir^{1,2}

¹ALSTOM and ²INRIA Rennes, France
bruno.adeline,pierre.dersin@alstom.com,
eric.fabre,loic.helouet,karim.kecir@inria.fr

Abstract. This paper considers evaluation of Key Performance Indicators (KPIs) for urban train systems equipped with regulation algorithms. We describe an efficient simulation model that can represent a network, animate metros, and integrate existing regulation schemes as black boxes. This macroscopic model allows efficient simulation of several hours of networks operations within a few seconds. We demonstrate the capacities of this simulation scheme on a case study and show how statistics can be derived during simulation campaigns. We then discuss possible improvements to increase accuracy of models.

Keywords: Key Performance Indicators, metro networks, regulation, simulation

1 Introduction

Urban train systems are subject to performance requirements originating from customers, operators or local authorities. These requirements (or Key Performance Indicators) can focus on punctuality of metros, regularity of service, passengers comfort. . . recent indicators also address energy consumption. Usually, trains follow predetermined schedules (a.k.a *timetables*) that allow, if realized as expected, to meet quality requirements. In a perfect world, trains arrive at stations and leave at the exact dates prescribed by a timetable or by a service rate. However, in everyday life, perturbations arise, and schedules are rarely satisfied.

Indeed, urban train systems are subject to random perturbations originating from weather conditions, passengers misbehavior, or failures. To recover from small delays, metro systems are equipped with *regulation* mechanisms, that give advices to train drivers (or to automated systems embedded in trains, if the line is driverless). Advices can be, for example, to reduce/increase dwell time or change commercial speed for a while to resume to the original timetable or to meet a regular service rate.

Regulation mechanisms are hence a key element for metros performance. They should be seen as an important part of the design of a metro line, and considered at early design stages. Several standard regulation techniques appear in the literature: the simplest ones try to stick to a prescribed timetable, but complex

proprietary regulation algorithms are also in use. One can however notice that there is no consensual mechanism considered as the best regulation technique: efficiency of a regulation scheme depends on frequency of delays, passengers behaviors, metro lines topology, and many other contextual features. Considering regulation and evaluating its performance at early design stages has several advantages. First, it allows to decide which regulation technique is adapted to particularities of the line under construction. Second, it allows to build timetables and to obtain estimates of achievable performances.

Several tools have been used to evaluate performance of mainline railway systems. Following the classification in [7], one can define these tools as macroscopic or microscopic simulation tools. Macroscopic approaches abstract away details and do not usually consider trains particularities for simulation; they usually lead to optimistic results. An example of such macroscopic model is the NEMO tool [5]. This tool uses abstract network graphs to compute timetables and detect possible bottlenecks. Microscopic approaches consider trains details, and many parameters such as weather conditions and up to passenger flows. Usually, these approaches consider how trains influence one another at runtime. They use synchronous techniques, i.e., repeatedly evaluate evolution of a network during user-defined time steps (for instance one second). OpenTrack is an example of such simulation framework (see for instance a description in [7]). Synchronous simulation is time consuming, and many steps simulated by the tools are simply useless, as no interaction between trains (forcing one of them to brake, for instance) nor change to a train's behavior (excepted for their positions) occur during most of time steps. OpenTrack and NEMO, as well as commercial softwares such as RailSys target main lines, where delays between departures and arrivals are quite long, and where small local perturbations have little influence on service performance. Challenges for these models are to design timetables, that are quite stable, and in case of failure in a network, find alternative paths for trains (see [2] for an introduction to the timetabling problem and associated tools). Computation of best alternative routes can take a few minutes without affecting too much traffic. In metro networks, paradigms change: trains are really close, minor disturbances may affect service quality, and advices have to be computed as fast as possible to be usable. Hence, corrective mechanisms are quite reactive, and the computed solutions to recover from a delay are applied as soon as possible. Models such as those proposed in the SimMETRO tool [6] address performance of metro systems in a microscopic (and stochastic) setting.

In this paper, we propose a macroscopic performance evaluation scheme for regulated metro systems, that can be used at early design stages. Metro networks are modeled as a variant of Stochastic Time Petri Nets [4]. Dwell and trip times are modeled as sojourn times in places, perturbations are modeled as random variations for these durations. In addition to the network dynamics, the system integrates a timetable and a regulation algorithm. The regulation algorithm is used as a black box that sends departure orders to trains and recomputes the timetable. We consider a fixed block policy: the metro network is divided into zones that can be entered by a single train. The distributions governing

trip and dwell durations are defined using expolynomial distributions. Indeed, as delays are more likely than advances the repartition of trip durations have particular asymmetric shapes that cannot be captured by standard uniform, exponential or Gaussian distributions. Distributions are hence defined on an interval in which durations with the highest probabilities are concentrated around several nominal values (nominal dwell or running times). Simulation of traffic is performed using an efficient technique that advances time to the date of the next event(s) (departures and arrivals), hence avoiding useless steps of standard synchronous simulation approaches. The proposed model is abstract enough to allow efficient simulation (many characteristics of trains, tracks and so on are abstracted away), but yet accurate enough to derive useful performance measures. We show that KPIs can be easily evaluated from our model, and demonstrate its practical interest on a real case study, namely line 1 on Santiago’s metro. The paper is organized as follows: Section 2 introduces our simulation model. Section 3 introduces KPIs and shows statistics obtained from a simulation of 4 hours of exploitation on our case study. Section 4 discusses our design choices, and possible improvements of the model, before conclusion.

2 Modeling

Urban train networks are composed of tracks, trains, safety and regulation mechanisms. Tracks can be decomposed in stations, rails and platforms, depots and turnback areas. Trains follow paths expressed as a succession of trips from departure to arrival terminuses and turnback maneuvers. The trip plans are usually detailed in a preconstructed timetable for a day or part of a day of exploitation. Timetables give a desired ideal schedule of trains departures and arrivals. They are an idealized representation of behaviors of trains, that is never perfectly met because of random delays due to incidents, weather conditions, etc. To leverage the effects of these disturbances, urban train systems are equipped with traffic regulation mechanisms that observe delays and compute orders and reschedulings to help the system get back to the ideal timetable.

We propose to model urban train systems with a variant of Stochastic Time Petri Nets as defined by Horváth et al. [4]. As we will show later in this section, this graphical model is particularly adapted to represent a network topology, and to manipulate durations subject to random perturbations. In the rest of the section, we only give an informal presentation of the model and refer to [3] for a complete presentation of the model and of its semantics.

Definition 1 (Stochastic Time Petri Net). *A Stochastic Time Petri Net (STPN for short) is a tuple $\mathcal{N} = \langle P, T, \bullet(), ()^\bullet, m_0, \text{eft}, \text{lft}, \mathcal{F} \rangle$ where P is a finite set of places; T is a finite set of transitions; $\bullet(): T \rightarrow 2^P$ and $()^\bullet: T \rightarrow 2^P$ are pre and post conditions depicting from which places transitions consume tokens, and to which places they output produced tokens; $m_0: P \rightarrow \{0, 1\}$ is the initial marking of the net; $\text{eft}: T \rightarrow \mathbb{Q}_{\geq 0}$ and $\text{lft}: T \rightarrow \mathbb{Q}_{\geq 0} \cup \{+\infty\}$ respectively specify the minimum and maximum time-to-fire that can be sampled for each transition; and $\mathcal{F}: T \rightarrow \Sigma_{\text{pdf}}$ associates a probability distribution to each transition.*

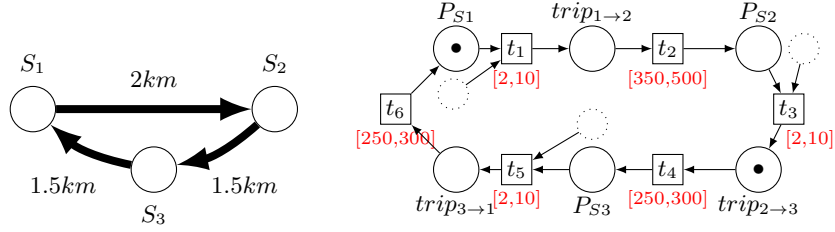


Fig. 1. An example STPN, symbolizing a simple ring topology with two trains.

Intuitively, places of a net represent either a track segment, a station, or a boolean condition allowing departure of trains. Transitions represent actions, i.e., departures or arrivals of trains. Intervals associated to transitions symbolize the range of possible dwell and trip times, and the distributions attached to transitions the probability distribution for each of these durations.

We denote by f_t the distribution $\mathcal{F}(t)$. To be consistent, we assume that for every $t \in T$, the support of f_t is $[\text{eft}(t), \text{lft}(t)]$. For a given transition $t \in P \cup T$, $\bullet t$ will be called the preset of t , and $t \bullet$ the postset of t . Transitions represent departures or arrivals of trains. The preset of an arrival transition has a single place representing the track portion arriving to the station. The preset of a departure transition has two places in its preset: a place representing a station, and a place representing an order from the regulation system allowing departures (we will come back to these places later in this section). Consider for instance the drawing at the left of Figure 1. This is a toy ring topology with 3 stations (S_1, S_2, S_3). The distance between S_1 and S_2 is 2 kms, and the distance between S_2 and S_3 is equal to the distance between S_3 and S_1 and is equal to 1.5 km. The commercial speed of train is 20 km/h, and the ring contains 2 trains. This simple topology can be depicted by the STPN at the right of the figure. The places labeled by P_{S_i} symbolize station S_i , and places labeled by $\text{trip}_{i \rightarrow j}$ the track portion between station S_i and station S_j . Transitions t_1, t_3, t_5 symbolize departures, and t_2, t_4, t_6 arrivals of trains. The intervals associated with transitions represent possible ranges of dwell and trip times. In the represented net, places P_{S_1} and $\text{trip}_{2 \rightarrow 3}$ contain a token, which represents a situation where a train is stopped at station S_1 and another one is moving from station S_2 to station S_3 .

This syntax of STPNs is similar to the one in [4], but we need to adapt their semantics to represent metro systems: for safety reasons, trains in a metro network have to preserve a *safety headway*. A way to address this safety requirement is to decompose a network into *blocks*, and allow a train to enter a block only when no other train uses it. This policy is called *fixed block* policy. Standard semantics of transitions firing in Petri nets consume tokens from the preset of a transition and produce tokens in its postset regardless of the contents of a place. In the standard setting, places can contain more than one token. To implement a fixed block policy, we define a blocking semantics that requires, in addition to standard rules, that places in $\bullet t$ that receive tokens when firing a transition t are empty.

The semantics of STPNs is defined in terms of sequences of discrete transition firings, and timed moves. We will say that a transition t is *enabled* by a marking m iff $\forall p \in \bullet t, m(p) = 1$. For a given marking m and a set of places P' , we will denote by $m - P'$ the marking that assigns $m(p)$ tokens to each place $p \in P \setminus P'$, and $m(p) - 1$ tokens to each place $p \in P'$. Similarly, we will denote by $m + P'$ the marking that assigns $m(p)$ tokens to each place $p \in P \setminus P'$, and $m(p) + 1$ tokens to each place $p \in P'$. Firing a transition t is done in two steps and consists in: (1) consuming tokens from $\bullet t$, leading to a *temporary* marking $m_{\text{tmp}} = m - \bullet t$, then (2) producing tokens in $t \bullet$, leading to a marking $m' = m_{\text{tmp}} + t \bullet$.

The blocking semantics of an STPN can be informally described as follows. A variable τ_t is attached to each transition t of the STPN. If a transition t represents an arrival at a station S , when $\bullet t$ is marked, this means that there a train on its way to station S . If t represents a departure from a station S , when $t \bullet$ is marked this means that a train is stationed at station S , and has received an authorization to leave. As soon as a transition t is enabled, τ_t is set to a random value ζ_t (called the *time-to-fire* of t , or TTF for short) sampled from $[\text{eft}(t), \text{lft}(t)]$ according to f_t . Intuitively, this TTF represents a duration that *must* elapse before firing t once t is enabled. The value of τ_t then decreases as time elapses but cannot reach negative values. When the TTF of a transition t reaches 0, then if $t \bullet$ is empty t becomes *urgent* and has to fire unless another transition with TTF 0 and empty postset fires; otherwise (if $t \bullet$ is not empty), t becomes *blocked*: its TTF stops decreasing and keeps value 0, and its firing is delayed until the postset of t becomes empty; in the meantime, t can be disabled by the firing of another transition. The semantics of STPNs is *urgent*: time can elapse by durations that do not exceed the minimal remaining TTF of enabled transitions that are not blocked. At a given moment, one can consider all remaining time to fire of enabled transitions, and compute the delay that has to elapse before some transition firing will occur. This allows to avoid synchronous approaches and perform *macro* time steps between two discrete events.

Let us say a few words about distributions attached to transitions. In our model, transitions symbolize departures and arrivals of trains. Places symbolize a station, or a track portion between two stations. A departure occurs a certain amount of time after arrival of the train at the considered station, and similarly, going from one station to another one takes time. Distributions describe the probability of durations for dwell and trip times. If one wants to obtain realistic models and accurate enough performance measures, these distributions have to be realistic enough. Distributions can be discrete (i.e., a list of possible values with associated weight), but for precision reasons, it is preferable to use continuous distributions. An usual way to model continuous distributions is to use Gaussian distributions, i.e., of the form $f(x) = \frac{1}{\sqrt{2\pi\sigma^2}} \cdot e^{-\frac{(x-\mu)^2}{2\sigma^2}}$, where μ and σ are parameters of the distribution. Such distributions describe a bell shaped curve, centered around the most probable value. In the setting of durations for dwell times or trips, delays are more likely than advance, and in general our distributions are not that symmetric. We hence use asymmetric distributions, modeled with expolynomial distributions.

Definition 2. A truncated expolynomial function over domain $[u, v]$ is a function of the form $f(x) = \begin{cases} \sum_0^K c_k \cdot x^{a_k} e^{-\lambda_k \cdot x} & \text{if } x \in [u, v] \\ 0 & \text{otherwise} \end{cases}$ where u, v and c_k, a_k, λ_k for every $k \in \{0, 1, \dots, K\}$ are rational values.
 $f(x)$ is an expolynomial distribution iff $\int_u^v f(x) = 1$.

During simulations of our Petri net model, dwell and trip durations are sampled according to distributions attached to transitions. Sampling from continuous distributions can be done using inverse transform techniques (see for instance [10]). Let us denote by F_t the cumulative distribution function (CDF) associated with f_t , i.e., $F_t(x) = \int_0^x f_t$. We will assume that every CDF F_t is strictly increasing on $[\text{eft}(t), \text{lft}(t)]$, which allows *inverse transform sampling*. Then sampling a value for a distribution defined by f_t amounts to sampling a value v from the standard uniform distribution in the interval $[0, 1]$, compute the value x such that $F_t(x) = v$, and take x as the random duration sampled from law f_t .

For efficiency reasons, one can also approximate truncated expolynomial functions with areas defined by zones, which greatly simplifies sampling for an acceptable precision loss. Figure 2 shows a Gaussian distribution $g(x) = \frac{1}{\sqrt{2\pi}} \cdot e^{-\frac{(x-4)^2}{2}}$ (i.e., with parameters $\mu = 4$ and $\sigma = 1$), an expolynomial distribution $f(x) = 0.58 \cdot x^2 \cdot e^{-1.7x} + 0.29 \cdot x^3 \cdot e^{-1.2x}$ defined over $[0.5, 6]$, and an approximation of this function on the same domain by an area delimited by two affine functions. On this figure, one can notice that a Gaussian distribution is centered around a pivot value: $g(x)$ describes a distribution in which the most probable values lay around 4 time units, but where the probability density of values before and after 4 is exactly 0.5. Conversely, the expolynomial distribution $f(x)$ has its most probable values centered around 2 time units, but the probability mass of values greater than 2 (0.67) is larger than that of values smaller than 2 (0.33). This can be interpreted as: the normal value for a delay is 2, but the probability to be delayed is higher than the probability to be in advance.

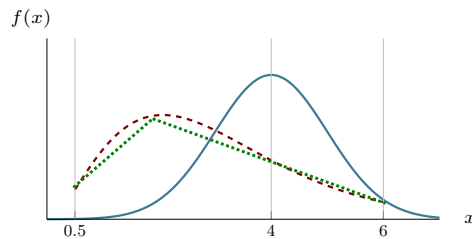


Fig. 2. A Gaussian distribution (plain line), an expolynomial function (dashed line), and its area approximation (dotted line).

Now, STPNs only describe the dynamics of trains, i.e. how they move from one track portion to another, and the time needed to move from one part of the network to another. As already mentioned, unwanted delays are recovered using regulation techniques that should be hence be considered when evaluating

the overall dynamics and performance on an urban train network. The overall behavior of our model is hence provided by a combination of a Petri net, and of a regulation algorithm. The Petri net part of our simulator simulates train moves, dwell and trip times and random delays for these durations. Firing a transition in this net means a departure or an arrival of a train at a given date. The regulation part of the model reads arrival and departure dates of trains (i.e., firing dates of transitions of the Petri net), and allows departures at dates prescribed by a timetable. Upon delay, the regulation algorithm recomputes a new timetable according to a regulation policy. Regulation algorithms usually recompute future departure or arrival dates of trains, which amounts to change dwell time or commercial speed (through the reduction of running and dwell times) upon observation of a delay. These techniques usually allow to catch up delays within a few stations. However, more involved regulation algorithms can redefine trains paths, allow overtaking of trains, insert/extract trains, etc.

Our simulation framework integrates regulation as follows: places of the Petri net represent stations or track portions. Transitions of the net represent departures or arrivals of trains. Some places in the preset of a departure transition (dotted places in Figure 3) represent orders given by the regulation algorithm. When all places of $\bullet t$ are filled and in particular the dotted place, the departure is allowed. This way, regulation algorithms can allow departures at a precise date, or impose a direction to a train leaving a station, in order to follow a plan. Consider the example of Figure 3. Place P_{S_1} contains a token. This token was put in the place by transition t_1 . The occurrence date of t_1 can be recorded and compared by the regulation algorithm to detect whether this event (a train arrival) was late. If this is the case, then the time table attached to the system can be updated. As soon as the regulation part fills place C_1 , a value from $[12, 20]$ can be sampled, and the train will leave as soon as this TTF reaches 0. Place P_{S_2} also contains a token, but the place has several transitions consuming tokens from it. According to the mission of the next train leaving station S_2 , the regulation module will fill either place C_2 , allowing firing of t_4 between 20 and 25 seconds later, or C_3 , allowing firing of t_5 between 18 and 25 seconds later.

This way, our simulator is an abstract representation of trains moves, but integrates a real regulation policy. Regulation algorithms are written as a set of rules applicable following a triggering event such as the delayed arrival of a train. They can be simple rules of the form "if a train arrives late by more than x time units then reduce dwell time to minimum allowed dwell time for the station". They can also be intricate rules choosing a decision to perform according to a set of thresholds... The framework proposed above has the advantage of integrating a real regulation policy. The same network and train fleets can be tested with different regulation algorithms without changing the whole model. However, this modularity and the expressiveness allowed in regulation has a cost: it is very hard to formalize and analyze the effect of regulation on the overall behavior of the model, that is hence more adapted to statistical analysis of performance via simulation.

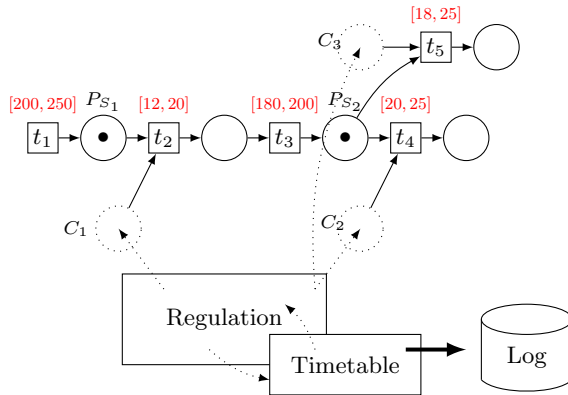


Fig. 3. The SIMSTORS simulation framework

In the modeled setting, we consider that regulation is a deterministic process: for a given delay detected at departure or arrival of a train, the changes to the schedule computed by the regulation algorithm is always the same. Though our simulation framework uses regulation as a black box, i.e., imposes no constraint of the type of regulation used by the system, this assumption seems sensible. An advantage of this assumption is that randomness comes exclusively from the part of the system represented by the stochastic time Petri net. An access to the current state of the schedule and to the times to fire of all transitions suffices to know the date of the next event that will occur in the system (a departure order given by the regulation, an effective departure, or an arrival of a train). As a consequence, one needs not discretize time according to a fixed sampling rate, and can consider only dates at which events occur. This is called *event-based simulation*, and allows for fast simulation of long runs of metro systems.

We have used this model to represent Line 1 of Santiago’s metro [?]. This line is a complex ring topology: two imbricated rings connecting 24 stations. The Petri net built for this line was a a net with 102 dwell places, 147 trip places, 147 control places, 147 departure transitions and 147 arrival transitions. The model contains depots and turnback zones in addition to stations and their interconnections. With this Petri net, we can simulate 4 hours of operations of Santiago’s metro with 50 trains and random perturbations in 40 seconds on an average laptop.

3 KPIs and simulation campaign results

Urban train networks are driven by operators that have commitments with local authorities to meet quality criteria. These criteria are standardized by the UITP [8], and known as Key Performance Indicators (KPIs for short). Failing to meet fixed performance objectives can result in financial penalties for operators. KPIs address several criteria: punctuality, regularity of service, number of failures, ratios of successful missions completed, but also more subjective measures such as

passengers comfort,... Usually, KPI measures are obtained by computing statistics from logs of train operations. Of course, these statistics make sense only if logs are recorded for a sufficiently long duration (day, week or month). Statistics are derived from a set of complete trips (travel from one terminus of a line to another endpoint of a line). In what follows, we assume that a sufficiently large set of effectively realized trips T is recorded. For each trip, departure and arrival dates at all stations have been recorded. We also assume that reference timetables are provided, indicating expected dates of departures and arrivals of trains when no perturbation arises.

Given this set T , the **punctuality** KPI is defined as the ratio of train trips delayed by less than x minutes over the total number of trips [8]. As formalized by UITP, this KPI only considers ending dates of trips. Formally speaking, this KPI is defined as

$$P_{\text{KPI}} \triangleq |\{t \in T \mid \dot{d}(t) - d(t) < x\}| / |T|$$

$\dot{d}(t)$ is the occurrence date of the last event of trip t , $d(t)$ the scheduled date of the last event of t , and x is a given threshold (in minutes).

The **regularity** KPI is defined by UITP as “the ratio of train departures at specified stations complying with planned headways within x minutes over the total number of departures from the specified stations”. More formally, we assume a selection of stations $S = \{s_1, s_2, \dots, s_m\}$ of interest where regularity is important. For each station $s_j \in S$, we denote by $Ed_j = \{ed_{1,j}, ed_{2,j}, \dots, ed_{k,j}\}$ the ordered set of departures from station s_j . We also denote by $\dot{h}(ed_{i+1,j})$ the effective headway between departure event ed_{i+1} and its predecessor ed_i , and by $h(ed_{i+1,j})$ the reference headway (for instance the headway planned for these trips in a reference timetable). The regularity KPI is then formally defined as:

$$R_{\text{KPI}} \triangleq \sum_{j=1}^m |\{ed_{i+1,j} \in Ed_j \mid 1 < i \leq k \wedge \dot{h}(ed_{i+1,j}) - h(ed_{i+1,j}) < x\}| / \sum_{j=1}^m |Ed_j|$$

x is a given threshold.

We have performed a simulation campaign for Santiago’s metro based on the model of line 1 mentioned in section 2. We have simulated the first 4 hours of operation of the line, with 50 trains operating on the line¹. The system was equipped with a regulation algorithm trying to stick as much as possible to a precomputed ideal timetable TT^{id} . The regulation plays on dwell times to recover from unexpected delays, and maintains a feasible timetable that associates to departures and arrivals their earliest possible occurrence date. This 4 hours simulation has been performed 100 times to record arrival and departure dates at all stations. During each simulation, dwell and running times for each event are randomly sampled from their respective distributions. The distributions attached to transitions were discretization of asymmetric bell shaped curves (i.e., close to a discretization of an expolynomial function).

¹ Traffic is not immediately maximal but increases progressively as trains are inserted in the network.

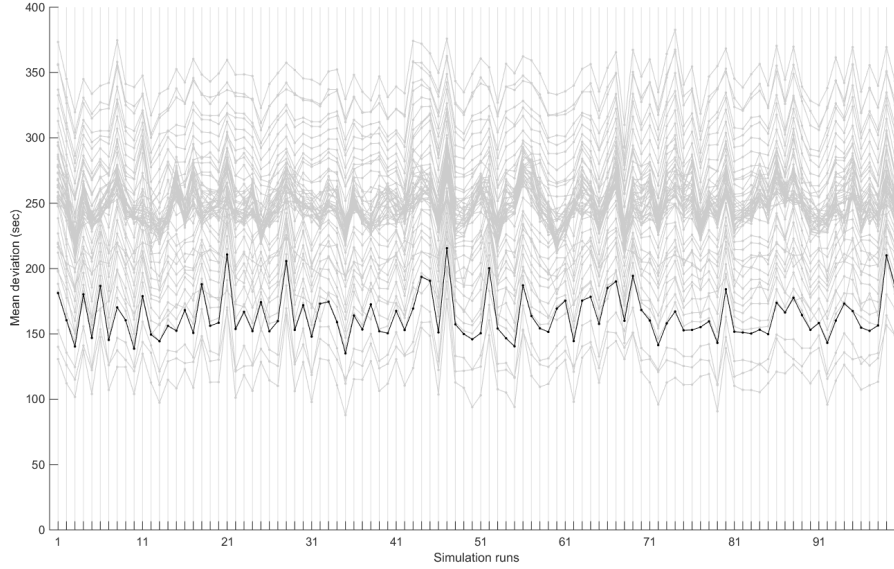


Fig. 4. Mean deviations from reference timetable for $n = 100$ runs

At the end of the simulation campaign, the obtained data were a succession of departure and arrival dates corresponding to 100 simulations, each simulation providing departure and arrival dates for all steps of realized trips. From these data, we have computed statistics and derived a KPI, namely the mean deviation w.r.t. desired departure headways. Overall, the campaign took around 1 hour.

Figure 4 depicts the mean deviations computed for each individual simulation. Abscissae indicate the simulation number (ranging from 1 to 100). The different curves on the picture represent the mean deviation with respect to the ideal timetable TT^{id} at each station (1 curve per station). The dark curve represents one particular station, namely Pajaritos, in running direction 1. Note that we slightly abuse the term “station”, as for each physical location of line 1, we have a station number for each running direction. (There are two possible directions: direction 1 from station “San Pablo” to “Los Dominicos” and direction 2 the converse way.) From these recorded mean deviations w.r.t. TT^{id} , one can observe the randomness of the simulation, as for each run, the results are different.

Let n be the number of simulations performed during a campaign (in our case $n = 100$), and let r_j with $j = 1, 2, \dots, n$ denote the j^{th} simulation (also called a *run* hereafter). Let m be the number of stations, and let us denote by s_k with $k = 1, 2, \dots, m$ the k^{th} station on the line. Events occurring at a given station s_k in a run r_j are denoted $e_{i,j,k}$ with $i = 1, 2, \dots, q_k$. Note that, during our simulation campaign, the number of events per station was the same from a run to another. Considering the idealized timetable TT^{id} , one can easily find the i^{th} event (departure or arrival) at station k , and hence obtain its planned occurrence date. We denote by $d(e_{i,j,k})$ the *reference date* for the occurrence

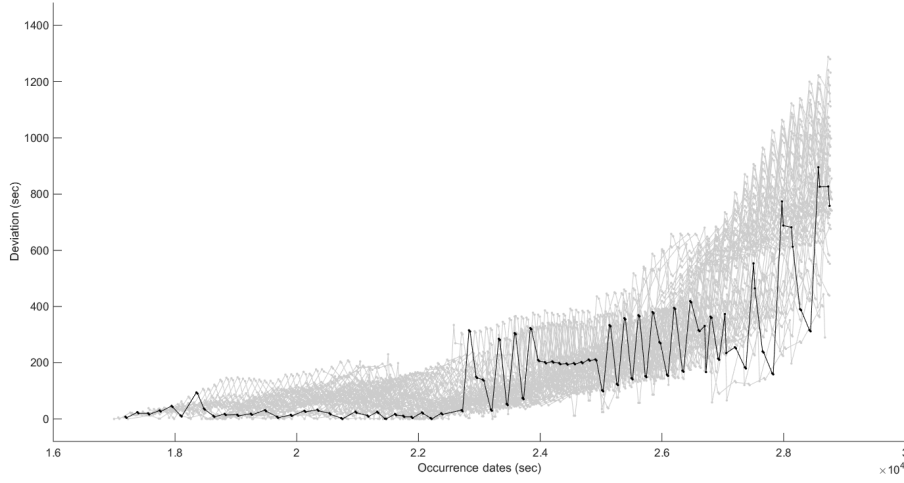


Fig. 5. Progress of deviations from reference timetable for station Pajaritos, direction 1

of event $e_{i,j,k}$, which is the ideal occurrence date. Note that as $d(e_{i,j,k})$ is the same for all runs, we can simply write it as $d_{i,k}$. We denote by $\hat{d}_{i,j,k}$ the *effective occurrence date* of event $e_{i,j,k}$ in run r_j at station s_k . The *deviation* (w.r.t. the reference timetable dates) for an event $e_{i,j,k}$ is the difference $\delta_{i,j,k} = \hat{d}_{i,j,k} - d_{i,k}$ between its effective date of occurrence and its desired occurrence date. Consider Figure 5. This graphics represent data collected during a single run of our simulation. Each curve represents the evolution of deviations for a particular station. Abscissae represent time elapsing, and ordinates give the deviations $\delta_{1,1,k}, \delta_{2,1,k}, \dots, \delta_{q_k,1,k}$. It might seem surprising that deviations grow but this is due to the chosen parameters for the simulation: we have deliberately selected high values of perturbations, to be able to observe the impact of regulation. One can see that, in the beginning of the simulation, the regulation is able to recover, more or less, from the perturbations but, as time progresses, the system becomes unstable. This is due to the fact that more and more trains are inserted into the network. As a consequence, it becomes harder for regulation algorithms to recover from consequent delays, and bunching phenomena appear.

Instead of reasoning in terms of occurrence dates and deviations, one can also consider **headways**, as they give a better measure of traffic regularity. For headways to be relevant, they have to be measured only between events of the same type (i.e., departures or arrivals). We hence denote by $e_{i,j,k}^d$ (resp. $e_{i,j,k}^a$) the i^{th} departure (resp. arrival) at station s_k in run r_j . Similarly, we denote by q_k^d (resp. q_k^a) the total number of departures (resp. arrivals) at station k (one need not differentiate between runs). We then denote by $h_{i,k}^d \triangleq d(e_{i+1,k}^d) - d(e_{i,k}^d)$ the reference headway at departure $i+1$ and by $h_{i,k}^a \triangleq d(e_{i+1,k}^a) - d(e_{i,k}^a)$ the reference headway at arrival $i+1$ at station s_k . We denote by $\hat{h}_{i,j,k}^d \triangleq \hat{d}(e_{i+1,j,k}^d) - \hat{d}(e_{i,j,k}^d)$

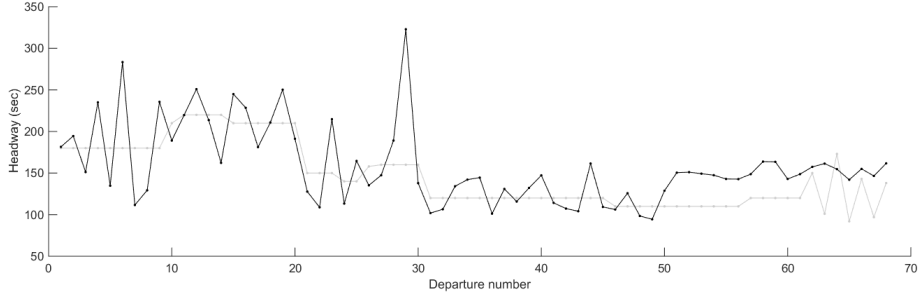


Fig. 6. Effective and reference headways for station Los Héroes, dir. 1 for one simulation

the effective headway at departure $i + 1$ in run r_j and $\hat{h}_{i,j,k}^a \triangleq d(e_{i+1,j,k}^a) - d(e_{i,j,k}^a)$ the reference headway at arrival $i + 1$ at station s_k .

We can then define $\bar{h}_k^d \triangleq \sum_{i=1}^{q_k^d-1} h_{i,k}^d / (q_k^d - 1)$ and $\bar{h}_k^a \triangleq \sum_{i=1}^{q_k^a-1} h_{i,k}^a / (q_k^a - 1)$ as the mean reference headways for departures and arrivals at station s_k respectively. Also, $\tilde{h}_{j,k}^d \triangleq \sum_{i=1}^{q_k^d-1} \hat{h}_{i,j,k}^d / (q_k^d - 1)$ and $\tilde{h}_{j,k}^a \triangleq \sum_{i=1}^{q_k^a-1} \hat{h}_{i,j,k}^a / (q_k^a - 1)$ are the *mean effective departure (resp. arrival) headway* at station s_k during run r_j , and $\bar{h}_k^d \triangleq \sum_{j=1}^n \tilde{h}_{j,k}^d / n$ and $\bar{h}_k^a \triangleq \sum_{j=1}^n \tilde{h}_{j,k}^a / n$ the *mean effective headway at station s_k* for the simulation campaign.

Figure 6 shows departure headways from Los Héroes station in running direction 1. Absissae depict events indexes, and ordinates the effective departure headways for one simulation run. Reference headways are depicted in gray and effective headways in black. One can observe that the regulation has an effect on headways. Indeed the curves of reference and effective headways are different, but their general profile remains close (there is no divergence in the effective headway curve). Now, one cannot draw conclusions from a single run of a stochastic simulation. In what follows, we give confidence intervals for means of deviations between mean effective departure headways and mean reference headways per station derived from a simulation campaign of several runs (here, 100).

A stochastic simulation campaign can be used to measure KPIs defined as mean value of some quantity ζ_i measured for each sampled run r_i . It is however interesting to know how the computed value approaches the theoretical mean μ for this KPI. Such a confidence can be quantified through **confidence intervals**. We call $M \triangleq \frac{1}{n} \sum_{i=1}^n \zeta_i$ the *sample mean* obtained from ζ_i 's, and σ the corresponding *estimated standard deviation*. According to the law of large numbers, M approaches μ only when the number of samples n is sufficiently large. To increase confidence in the computed value, a standard approach is to set a confidence level α , and compute a confidence interval I from M . I is the confidence interval for μ at confidence level $1 - \alpha$, i.e., the probability that μ belongs to I is $1 - \alpha$. Given n , M , σ and α , the confidence interval is defined as:

$$I \triangleq \left[M - \gamma_\alpha \frac{\sigma}{\sqrt{n}}, M + \gamma_\alpha \frac{\sigma}{\sqrt{n}} \right] \quad (1)$$

where γ_α is value depending only on α called the *z-score*.²

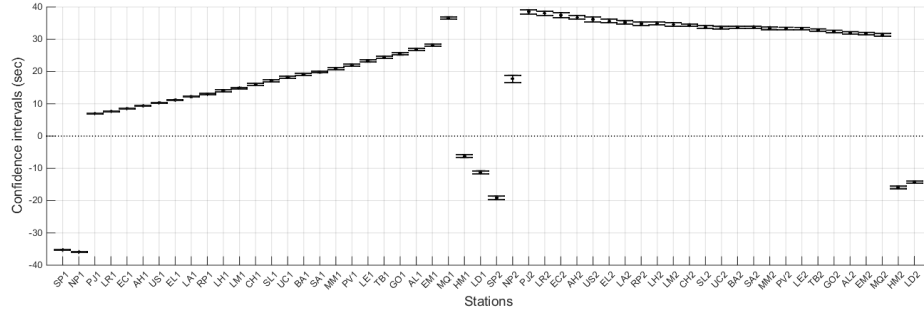


Fig. 7. 99.9% confidence intervals for means of deviations between mean effective departure headways and mean reference headways per station

Let us now consider a KPI measuring the mean deviation w.r.t. reference departure headways for a station. The *headway deviation* (difference between the effective headway and the reference headway) for event i , in run r_j at station s_k is defined as $\theta_{i,j,k} = \hat{h}_{i,j,k}^d - h_{i,k}^d$. The *mean headway deviation in a run r_j at station s_k* is given by $\bar{\theta}_{j,k} \triangleq \sum_{i=1}^{q_k^d-1} \theta_{i,j,k} / (q_k^d - 1)$. Finally, the *mean headway deviation at station s_k for a simulation campaign of n runs* is $\bar{\theta}_k \triangleq \sum_{j=1}^n \bar{\theta}_{j,k} / n$. The *standard deviation of $\bar{\theta}_k$* in a simulation campaign of n runs is $\sigma_k \triangleq \sqrt{\sum_{j=1}^n (\bar{\theta}_k - \bar{\theta}_{j,k})^2 / (n - 1)}$.

Figure 7 shows the confidence intervals computed for headway deviations at each station. The parameters of the simulation are $n = 100$ runs, and the intervals are computed for a confidence $1 - \alpha = 99.9\%$. In this Figure, the horizontal axis carries station names, and the ordinates represent values of mean deviations. For each station, the graphics contain an interval around the sample mean value computed from the simulation campaign. One can notice that headway deviations grow progressively from station Pajaritos direction 1 to Manquehue direction 1 and from Manquehue direction 2 to Pajaritos direction 2. This is explained by an accumulation of delays due to bottlenecks at both ends of the network. One can also notice that mean headway deviations at the ends of the line (stations SP1, NP1, HM1, LD1, SP2, NP2, HM2, and LD2) do not follow this general profile (they have smaller effective headways). This is due to the fact that these stations are used for train insertaion and turnback maneuvers and allow for more flexible regulation margins. Accumulated delays can be recovered at these stations (up to a certain limit) by considerably reducing sojourn time or using fast turnback

² This value is the real value such that $\mathbb{P}[|N| \leq \gamma_\alpha] = 1 - \alpha$, where N is a variable following a normal law $\mathcal{N}(0, 1)$. This value is not easily computable, but all statistical tools provide means to obtain γ_α , for instance using precalculated *z-tables*.

techniques. Last, one can see that the chosen perturbation level for this simulation is too high to allow recovery from delays by the selected regulation.

4 Discussion and improvements

The model proposed in this paper has been tested on a real case study; namely, the Line 1 of Santiago’s Metro, with a hold-on regulation policy that tries to stick as much as possible to a predetermined timetable. This first experimentation allows to obtain simulation results within a reasonable time (a few seconds for 4 hours of operation of a real network, i.e., a real topology with its actual train fleet). This shows feasibility of a simulation approach to evaluate performance of regulation algorithms.

Now, the setting of the paper can be improved along several directions. First of all, distributions for delays were designed from an a priori knowledge of normal dwell and trip times between two stations. To guarantee that these distributions are accurate enough, one could observe trains and passengers behaviors over a long enough period, and derive distributions from the collected data. A second issue regarding distributions is that the delays are modeled as Markovian noise. In this setting, every delay is sampled independently from the others. In urban train networks, latencies are correlated. For instance, if a train gets late, more passengers will enter the train, which will increase the chances of delay. Similarly, if a train is delayed due to bad weather between two stations, all trains of the network are likely to be delayed on the same part of the network. This means that sampling in our simulator should consider a context, and that distributions should be conditional distributions of the form $p(x | c_1, c_2, \dots, c_k)$ where x is a delay, and c_1, c_2, \dots, c_k are variables representing the context (station, weather, day of the week, time of the day, etc.) in which delay x is sampled. This change does not require much effort to be integrated to our simulation model. However, it does require a lot of effort from designers to evaluate the impact of an environmental factor on the distributions.

Train fleets: A second issue that should be considered is the impact of fleet composition on computed metrics. In the simulation that we have performed, we have considered regulation techniques that cannot change composition of fleets to meet their objectives. The number of running trains changes according to the period of the day, but follow planned insertions and removals of trains: It would be interesting to consider regulation techniques that can recommend to insert or remove trains to meet a desired KPI. In a similar way, we have considered uniform fleets. This is however not the case that all trains have the same speed, same capacities, etc. One can easily integrate to distributions (and to the context as described above) the type of each train when sampling a dwell or running duration. As for all environmental factors, this difference between trains can be defined using conditional distributions, but with an increased design cost.

Moving block: In this paper, all experiments have been conducted assuming that the line was operated with a fixed block policy, forbidding trains to enter an already occupied track section (block). However, in reality, trains can also follow

a moving block policy [9]. The moving block policy as described by Pearson states that “A train is continuously supplied with accurate information of the position of the nearest obstacle on the track ahead of it [...] it may be a preceding train, which itself may be moving or stationary. The speed of the train is constantly checked and adjusted [...] so that it is always possible for the train to be brought to rest without colliding with the obstacle.” In this setting, several trains can enter a track portion as long as they adapt their speed to their predecessors. Changing the Petri net setting to adapt to this change needs to consider trip durations as constrained delays attached to trains and not as time to fire attached to transitions. This change to the model is currently under study.

Passengers flows: the last aspect that may improve accuracy of the model is to consider how passengers transfer from one line to another. Indeed, metro networks are often composed of several interconnected lines. A flow of passenger entering a line at an endpoint is likely to transfer to another line at a junction point of the network. This flow of passengers is often captured with Origin-Destination Matrices, in which entries indicate the proportion of passengers alighting at station i that leave a train at station j , or which proportion of passengers leaving at a junction station enter the next train of another line. In its current status, our model does not integrate flows nor address the number of passengers. As already mentioned, the number of passengers impacts the distribution of delays. However, integrating passengers flows to our model is likely to increase simulation time dramatically, as it requires counting (or at least quantizing) trains population, and remembering passengers alighting histories to guarantee faithful representation of passengers flows. An inspiration for this improvement of our model and of our simulation framework is the *multiphase fluid Petri nets* proposed in [1]. Another difficulty in flows representation is that input-output matrices are not known a priori. They are not available at early design stages. They have to be built once a metro network is operational, which usually requires observation of passenger habits for long periods of time.

5 Conclusion

In this paper, we have detailed a framework for performance evaluation of regulation algorithms on a particular metro line. This framework consists of a high-level model of the network and of train moves, with random perturbations, in which a regulation algorithm is inserted to correct these delays. The overall systems allow for fast simulation, and hence for realization of simulation campaigns to obtain statistics on the efficiency of a regulation algorithm to meet KPI objectives.

The proposed framework allowed us to derive statistics for a case study, namely Line 1 of Santiago’s Metro. A key question raised by our study is the tradeoff between abstraction (allowing efficiency of simulation) and accuracy of the statistics derived. Petri nets allow for an accurate modeling of network topologies, the key ingredient for our model is hence accuracy of trip and dwell times. As explained in the paper, truncated expolynomial functions allow for precise modeling of distributions in which trains are more likely to be delayed

than advanced. When sampling for such functions is too time consuming, these functions can be approximated with areas delimited by affine functions.

Now, a major challenge is to define these distributions. Of course, at early stages of design, one can rely on expected characteristics of the network and trains to design distributions a priori. For an existing system, when the challenge is not design but rather to adapt regulation train fleets and their paths to improve KPIs, one may want to work with accurate distributions, that consider elements from context: passengers, trains, but also regulation itself. In such a situation, collected logs can help learning parameters of a distribution for dwell or trip time, but it remains a challenging task to estimate the contribution of passengers or regulation to a certain duration, as these parameters are usually not remembered logs. As a future work, we plan to use our tool to compare regulation techniques, and to improve its accuracy by learning distributions.

References

1. Santiago's metro map. <http://transitmap.net/post/18863388725/santiago>.
2. S. Haar and S. Theissing. A hybrid-dynamical model for passenger-flow in transportation systems. In *ADHS'15, IFAC-PapersOnLine 48(27)*, Elsevier Science Publishers, pages 236–241, 2015.
3. I.A. Hansen. Railway network timetabling and dynamic traffic management. *International Journal of Civil Engineering*, 8(1):19–32, 2010.
4. L. Hérouët and K. Kecir. Realizability of schedules by stochastic time Petri nets with blocking semantics. In *Proc. of PETRI NETS 2016*, volume 9698 of *Lecture Notes in Computer Science*, pages 155–175, 2016.
5. A. Horváth, M. Paolieri, L. Ridi, and E. Vicario. Transient analysis of non-markovian models using stochastic state classes. *Perform. Eval.*, 69(7-8):315–335, 2012.
6. M. Kettner, B. Sewcyk, and C. Eickmann. Integrating microscopic and macroscopic models for railway network evaluation. *Association for European Transport*, 2003.
7. H.N. Koustopoulos and Z. Wang. Simulation of urban rail operations: model and calibration methodology. In *Transport Simulation, Beyond Traditional Approaches*, pages 153–169, 2009.
8. A. Nash and D. Huerlimann. Railroad simulation using opentrack. In *Computers in Railways IX*, J. Allan, C. A. Brebbia, R. J. Hill, G. Sciutto & S. Sone (Editors), page ??, 2004.
9. UITP (International Association of Public Transports). Metro service performance indicators, a uitp information sheet. 2011.
10. L.V. Pearson. *Moving Block Railway Signalling*. PhD thesis, Wiley, 1973.
11. R. Y. Rubinstein and D.P. Kroese. *Simulation and the Monte Carlo Method*. Wiley, 2nd edition, 2008.