Dynamic delay management at railways: a Semi-Markovian Decision approach

Al Ibrahim, A.

Publication date
2010

Chapter 2

Model preliminaries

In this chapter we will introduce some model preliminaries that will be useful for the remainder of the thesis. Also, some important modelling choices that are of influence on the remainder of the thesis are the subject of this chapter. As we have pointed out in the previous chapter, the goal of the research is optimizing the situation at junctions. In order to model railway junctions correctly we first need to understand what the typical junction looks like. This will be done in Section 2.1. Afterwards we will introduce the SMD-railway framework which will form the basis for the models of subsequent chapters. Here, some important ideas are presented and some concepts are outlined.

2.1 A typical junction

At a typical junction several railway routes converge. This implies a physical connection between the tracks of these routes. Trains that run on these converging routes share at least a small portion of the track with trains from other routes. We will call the tracks, from which the trains arrive at the junction, *arrival tracks* and the track which is shared with trains from other routes the *destination track*. Sometimes the destination track is used in both directions. In that case the track is called a *bi-directional destination track*.

In the real world signals are found towards the end of every arrival track. These signals regulate the train traffic through the junction and ensure that only one train can cross the junction at a time. This way collisions are prevented. Thus, at most one signal at a time can show green, the rest of the signals at the same junction will show red. The notion of signals is part of the safety system currently being used. It is possible that in the future these signals will be replaced by other safety measures. In our model we will use a so-called *speed indicator* that can be regarded as the signal color but in our model provides us with more detailed information: the train speed on a track. More on this will be presented in Section 2.2.

In reality, the railway tracks are divided into smaller pieces, which are called *blocks*. Within each block only one train is allowed to be at the same time. The length, the position and the number of the blocks are defined by the safety system being used. Current safety systems use so-called *fixed blocks* which are physically separated by signals or other hardware units. The block length is set in such a manner as to be sure that all trains have enough time to come to a stand still at the end of the block if this is necessary. In the future, new safety systems are likely to be implemented. These systems allow for a better usage of the available capacity. One such system is the so-called *Moving block safety system* (e.g. [98], [38]). This system does not make use of signals, which are actually inherent to the current safety system, but instead keeps track of the position of every train on the railways. Each train reserves some space in front of itself. This reserved space moves along with the train. Hence the name: Moving block safety system.

When a train crosses the junction, the signal changes to red for all other arrival tracks. The trains on these tracks will need to adjust their speeds or even come to a stand still while waiting for the junction to be cleared. When a train gets the green signal, the train has to accelerate to its desired speed again. Thus trains get delayed by the amount of time they wait for the signal to turn green and by the amount of time they lose when accelerating to their desired speeds again. This acceleration time depends on the speed the train decelerated to, the type of the train, the number of carriages and, particularly in case of freight trains, the type of the locomotive and the mass of the train.

As trains can not stop immediately, the trains should claim the junction beforehand. Thus the junction is blocked for other trains for the time the claiming train needs to get to the junction, cross it and clear the junction.

Trains can overtake each other only at a limited number of locations. So when a train order is set at the junction and the trains run behind each other, the trains will move in that order for quite some time. If a train catches up with its predecessor on the destination track, the train has to adjust its speed to that of the predecessor and to respect the minimal distance of one block. This continues until a double track is reached where trains may overtake each other, or until the predecessor train branches off into another direction.

2.2 The SMD railway framework

To facilitate our research on dynamic delay management at railways, a framework will be built which is based on the Semi-Markovian decision technique. This framework will be used in later chapters to construct SMD models which meet characteristics of specific railway junctions. In this section we will introduce this framework and will describe the different aspects of it.

As stated, a junction will be divided into two parts, the arrival tracks and the destination track. We will begin by addressing the arrival tracks in Section 2.2.1. We will explain how these tracks are modelled. Then, in Section 2.2.2 we will explain the arrival process which is used to model train arrivals to the arrival tracks. Next in Section 2.2.3 we will explain how the trains cross the arrival track to enter the destination track and why we can jump in time to the next decision moment. During this time jump the state on the destination track changes. To reflect this change we will first explain the different alternatives to model the destination track (Section 2.2.4) and will explain their advantages and disadvantages. Then we will explain how the destination track is discretized and divided into a number of blocks. In Section 2.2.5 we will sum up the different functions of these blocks and will explain in Section 2.2.7 how the movement of trains on the destination track is modelled. In Section 2.2.6 we will discuss how the headway, which is the minimal safety time between trains, is incorporated into the model. Finally the concept of Externality costs is explained in Section 2.2.8.

2.2.1 Arrival track

The arrival tracks are the tracks where the trains enter the junction. The trains enter the arrival track and thus enter the scope of the model whenever they approach the junction and the request is sent to the train dispatcher to claim the junction. In practice, this request can be granted or dismissed by a train dispatcher depending on the specified rules being used.

The arrival tracks will be modelled as queues, i.e. the positions of the trains on the track will not be part of the state description. Only the number of trains in front of the junction, their order in the queue and their type are registered. This way the state space (the size of the problem) remains manageable while the most relevant information about the trains is still available.

The signals, that in real life are found at the end of the arrival tracks, are not part of the model. Instead, the speed indicators are used. The speed indicator is a code which provides us with the information about the speeds of the trains on that track. In the most basic case the code has only two values, 0 and 1 with:

- 1 meaning that the trains on that track are running according to their speed profile and do not experience any hindrance from trains of other directions.

- 0 meaning that the trains on that track are standing still in order to let one or more trains from other directions cross the junction first.

The speed indicator can be extended by other values indicating that the trains on that track are experiencing some hindrance and thus are running slowly. In fact, the number of speed indicator values is free to specify, but of course, this number does influence the size of the state space and by this the complexity of the problem to be solved. In real life the speed indicator values can be related to the color of the signal being it green, yellow, red or some blinking variations that indicate that the train may proceed but with some limited speed.

2.2.2 The arrival process of trains

Nowadays, using timetables is the common practice at railways. Trains are scheduled very accurately so that if there would be no disturbances at all, there would be no delays and no conflicts. However, most trains are not exactly on schedule and this results in fairly frequent conflicts. It is expected that the traffic intensity will be increased in the future. In an already dense network as that of The Netherlands, preserving the timetables may become unsustainable. In Section 1.2 we have discussed the ambitions of the Dutch government to head in the direction of the timetable-free operation and mentioned the pilot study which has already been conducted where on a busy corridor the number of trains has been raised to study the feasibility of such a concept.

As has been stated in Section 1.2, the operation will not entirely be timetable-free since the trains will likely be scheduled to be separated in time with more or less equal intervals in-between. But due to the high frequency and delays one may speak of a timetable-free operation. As a direct consequence of the new situation, the train arrival times will be more random than is the case nowadays.

In our approach we will assume train arrivals to follow a Poisson process, however the actual process will be slightly less random due to the incorporation of the minimal allowable time between two subsequent trains. The assumption of Poisson arrivals is an approximation for the timetable-free case sketched above but due to the high frequency of operation and the delays, we believe that this approximation will be quite well. Nevertheless, it is possible to incorporate a less chaotic random process into the model by using the *phase type arrival process*. We will discuss the required changes to the model in Chapter 9 but for now we will stick to the Poisson process.

The arrival process, that we use in our research, is the one that we call the $\mathcal{HP}$-process. This process combines the idea that, by regulation, the trains are separated in time from each other with the idea of Poisson arrivals. This way, the $\mathcal{HP}$-process will be a better model for the actual arrival process than the standard Poisson process. The process is constructed in the following way.

Let random variable I be Poisson distributed with mean λ, and let it represent the number of train arrivals within a given time interval τ. Then, the probability of exactly i arrivals within time interval τ is denoted by

$$P^{\tau}[I = i] = P_{\lambda}(i) = \frac{\lambda^i}{i!} e^{-\lambda} \tag{2.2.1}$$

Let N represent the maximum number of arrivals that can occur within the time interval τ) when the minimal headway between trains is considered. Then we can construct the truncated Poisson distribution as being

$$P_{\lambda}^{N}(i) = \frac{\lambda^i}{i!} e^{-\lambda} \quad \forall i < N \tag{2.2.2}$$

and for $i = N$ we have

$$P_{\lambda}^{N}(i) = 1 - \sum_{j=0}^{N-1} P_{\lambda}(j) \tag{2.2.3}$$

The mean of the truncated Poisson process is different from that of the original Poisson process. Nevertheless, we wish our arrival $\mathcal{HP}$ process to match the original Poisson arrival rate λ. Therefore a correction is required. Let μ represent the mean of the truncated Poisson process, that is

$$\mu = \sum_{j=0}^{N} j P_{\lambda}^{N}(j) \tag{2.2.4}$$

then we can define the $\mathcal{HP}$-process as:

$$\bar{P}_{\lambda}(i) = \frac{\lambda}{\mu} P_{\lambda}^{N}(i) \quad \forall 0 < i \leq N \tag{2.2.5}$$

and

$$\bar{P}_{\lambda}(0) = 1 - \sum_{i=1}^{N} \bar{P}_{\lambda}(i) \tag{2.2.6}$$

Note that theoretically Equation (2.2.6) could result in a negative $\bar{P}_\lambda(0)$. The reason for this is the value of μ being too low when compared to the value of λ. The latter is the case when N is chosen too low. For practical instances this is however not the case; In the model, λ depends on the time interval $\tau(\mathbf{x}, a)$ (see next Section) and given this time interval the maximum number of arrivals N is defined as $\lceil \frac{\tau(\mathbf{x},a)}{h} \rceil$[1] where h is the minimal headway due to regulation. This definition of N not only makes sense but turns out to result in a legitimate $\mathcal{HP}$-process (the one where all probabilities are between 0 and 1). If for any reason, for some specific problem instance, the above definition of N does violate the legitimacy of the $\mathcal{HP}$-process, then the value of N should be increased.

In practice h can be dependent on the pair of trains, e.g. two passenger trains may have a different safety gap than two freight trains. In this case we set h to be the smallest headway.

2.2.3 Time jumps and train movement on arrival tracks

There are different train types that arrive at the arrival tracks. Every train type has its own characteristics. Some are fast, some are slow, some have large masses making the train a slow accelerator. These train types approach and cross the junction with different speeds. When a certain train approaches the junction and receives permission to cross it (i.e. the decision is taken to let this train cross the junction),the train will need some time to cross the junction and clear it for other trains. During this time, the junction is blocked for other trains. In the mean time, no decision can take place, the time can then be advanced to the next decision moment. We will call this advance in time the *time jump*, and will denote it by $\tau(\mathbf{x}, a)$. This time depends on state $\mathbf{x}$ and on decision a. The time jump reflects the following events:

- The approach time of the train: The stretch of time between the moment the train arrives at the arrival track and the time it reaches the crossing. This time interval depends on the speed of the train and, in case of accelerating, on its acceleration rate.

- The junction clearance time: the time the train needs to clear the junction and make it available for other trains. This is typically the time needed for the rear of the train to clear the junction which again depends on the train speed and its acceleration.

[1]The notation $\lceil \frac{\tau(\mathbf{x},a)}{h} \rceil$ indicates that $\frac{\tau(\mathbf{x},a)}{h}$ is rounded up to the first integer value.

- Headway time: The safety margin between two subsequent trains, which may depend on their train type.
- Time that is needed for the first block on the destination track to be cleared by the predecessor train. Since only one train can occupy a block simultaneously, the first block of the destination track needs to be made available.

Note, that the immediate consequence of these events is that the time jump is at least equal to the value of the headway and there is also some relation between the minimum value of the time jump and the time the last train on the destination track needs to make some space for the next train.

If there are no trains to cross the junction, then the time jump in the model is taken to be the headway time. Alternatively, the time can be advanced up to the moment of the next arrival. The latter is however not suitable for practical reasons. The former modelling choice limits the number of possible different values of time jumps which in its turn limits the number of states significantly.

2.2.4 Destination track

A conflict occurs whenever two or more trains want to claim the same infrastructure at more or less the same time. Thus, there is a certain track that these trains share. We call this track the *destination track*.

There are some choices to be made when considering the modelling of this track, but let us first recall why it is so important to take this track into consideration in the first place. The reason for this, is the fact that trains affect each other not only while crossing the junction but also after that, on the track these trains share together. Thus, the length of the track should represent the length of the railway segment where the trains can hinder each other. Note, that when the trains only cross each others' path at the junction and do not share any infrastructure after that, this can be modelled as a destination track of length zero. Moreover, the trains on the track can also affect the trains that are yet to enter the track and thus may influence the conflict resolution strategy.

Thus, somehow, we need to keep track of the trains that have entered the destination track and memorize their location and speed. From this data, the amount of hindrance can be computed whenever needed. However, both speed and location of the trains are continuous in nature and can not be modelled directly. Instead, some other approaches need to be searched for. In reality, the speed of the trains depends for the largest part on the train type and the amount of delay the train experiences. Long distance trains run faster than local trains and delayed trains usually run faster (or at least are prepared to run faster) than trains that are on time. If we would construct a number of train categories (e.g. Intercity train being on time, Intercity train being slightly delayed, Intercity train being heavily delayed, Freight train being on time etc.) then it will be enough to keep track of the location of such train category in order to have all the needed information to be able to compute the hindrance at any given time. Alternatively, we can state within our model that all trains always run with maximum allowable speed eliminating the need of incorporation of the level of delay the trains experience. Then the speed that the trains run with on the destination track depends solely on the type of the train. The advantage of this approach is that it requires less variables and thus will lead to a more compact state space. A disadvantage of course is that the resulting model will be less accurate when comparing it to the real world. Either way, this choice does not affect the model since in both cases, the continuous speed that a train runs with, is substituted by the type (or the category) of that train which will be an integer variable that can easily be incorporated within the model.

Modelling the location of the trains on the destination track is a much more challenging issue. When thinking of the way to model this, two approaches come to mind:

One might choose to keep track of the time the trains enter the destination track. This, together with the type of these trains should give enough information to be able to calculate at any given time instance the amount of hindrance these trains may cause to other trains. To model this, one might think of using relative times where the 'entrance time' of the trains to the track are relative to the time when the decision must be made about the next train that may enter the track. Although, at the first glance, these relative times seem to be continuous in nature and thus unsuitable to model, this approach is still feasible. The reason for this, is the fact that the trains may enter the destination track only if this does not violate the headway (the minimal time between trains) and thus only at a number of prescribed decision moments. This limits the number of possible relative entrance times making it a finite set. So with some skilful modelling choices this approach can be used, however, this leads to a large state space since per train on the arrival track one needs to keep track of its type, the rank (order number on the track) and the time it entered the track. So the state will be characterized by the vector $x = (x(t_{min}, 1), x(t_{min+1}, 2), \cdots, x(t_{max}, r))$ where t stands for entrance time, r for the rank and where the $x(t, r)$ gives the train type.

A different approach is to divide the destination track into a number of blocks of equal length. Then the block number where the train is located is directly correlated with the distance, the train has travelled on the track. This results in a much more compact state space where the state is characterized by vector $x = (x(1), \cdots, x(K))$ where K is the number of blocks at the destination track and $x(i)$ the type of train that occupies block i. If block i is empty then $x(i)$ is zero. A drawback of this modelling choice is however that the transitions become more complicated. Particularly when the length of the block does not correspond to the distance the train crosses given its speed and the time jump. An even bigger challenge is to incorporate the headway in such a description. The headway is the minimal time that the trains should keep between each other while the destination track is characterized by blocks of a certain length which do not relate to the headway.

In this thesis we will use the latter approach to model the destination track. We will demonstrate the resulting model in Chapters 3, 4 and 5. Then in Chapter 6 we will show how the modelling of the destination track can be improved. In the improved model the headway will be incorporated into the destination track in a very natural way resolving the conflict between the headway and the block length of the original model.

2.2.5 Function of blocks in discretized destination track

As stated above, the destination track has a certain length and is divided into a number of blocks. On each block only one train can be found simultaneously. In our model, the blocks have the following functions:

- The blocks provide us with information about how far the trains have travelled on the destination track.
- Current decisions are influenced by past decisions through the information that blocks provide: The type of the train, found on a certain block, together with the position of the block on the destination track influence the train that enters the destination track next.
- The blocks separate trains from each other in space since only one train can be on a block at the same time.

The length of the destination track can be regarded as the area where the trains can hinder each other. Past the end of it, the trains either move in separate directions or enter some railway hub with high capacity. Either way, we regard the railway capacity behind the destination track to be sufficiently large.

2.2.6 Headway time and block length on the destination track

We have mentioned that the headway (the minimal time between two subsequent trains) is taken into consideration upon the arrival of trains at the arrival tracks and upon the crossing of the junction i.e. departure from the arrival track and arrival at the destination track. The first is achieved by incorporating the headway into the $\mathcal{HP}$ arrival process while the fact that the minimal time jump is equal to the headway time, ensures that the trains can not enter the destination track without respecting the headway time. But what about the headway at the destination track? We have mentioned that the destination track is divided into a number of blocks of equal length, but how can we incorporate the concept of headway into the discretized destination track? On the destination track, the trains are not allowed to occupy the same block at the same time but two adjacent trains may violate the headway when the length of the block is shorter than the distance the trains move within the headway time. Of course we can set the length of the blocks such as to ensure that for all types the headway is always respected. But such large blocks will lead to unnecessary large distances between other train types. It is difficult to combine the concept of headway and the block length together. Instead we will accept the violation of the headway rule within the track but will enforce the rule upon the departure from the track i.e. no train is allowed to depart from the track before the headway rule is respected.

In Chapter 6 we will examine another way to model the destination track where the conflict between the headway concept and the blocks on the destination track is resolved in a natural way. However, the model of Chapter 6 is rather abstract, since the speed and position of multiple trains on the destination track is modelled by one single variable. From this perspective, it is wiser to begin by explaining a more intuitive model where the movement and the speed of the trains are modelled in a more detailed way but where, as has been said earlier, the headway is respected at the entrance to and departure from each track and may be violated on the destination track.

So, we have some freedom to choose the length of the blocks on the destination track. The choice that we will make is the one that will simplify the transition process. The length of each block will represent the distance the slowest train on the destination track travels in exactly the headway time. Since the time jump is at least equal to the value of the headway time, the obtained block length ensures that the trains on the destination track will move at least one block forward, no matter the value of the time jump.

Note that this choice can lead to the situation that the sum of the block lengths does not equal the original length of the destination track. We will solve this by extending the length of the destination track but take the original length into consideration when computing the amount of hindrance the trains have on each other. This will be explained later.

2.2.7 Train movement on the destination track

In this section we will discuss the train movements in this discretized environment. Defining these movements (transitions) has to be done very carefully. The following questions should be kept in mind while modelling these transitions. How can the movement of trains with different speeds be reflected within such an environment? How to model the transitions of trains that run at exactly the same speed? How can we ensure that fast trains will not be able to overtake the slow ones on the destination track?

On the destination track the trains are constantly in motion. Giving some time interval τ, the position of the trains on the track changes according to their speed profile. Recall that the track is divided into blocks. Such a block has a certain length. Given the speed of a train and the time interval τ, the train moves a number of blocks forward. If the train moves a non-integer number of blocks z, then a translation is needed. In the model the number of blocks the train moves will be either $\lfloor z \rfloor$ or $\lceil z \rceil$, with mean z ($\lfloor z \rfloor$ is equal to z rounded down, $\lceil z \rceil$ is z rounded up).

Further the movement of the trains will be 'coupled'. This means that the movement probabilities of one train will be coupled to the movement probabilities of other trains. The function of the coupling is twofold. First of all, it will insure that the trains that move with the same speed will stay at the same distance from each other. And second, through the coupling, the slower trains will never catch up with the faster trains. Moreover, if the fast train catches up with the slower train, its speed will be lowered so that it will stay directly behind the slower train since overtaking is not possible on the destination track.

Formally, the coupling of the train movements is realized as follows. Let us number the trains on the destination track from the end of the track down to the beginning of the track. I.e. the first train is the train which has travelled the furthest while the last train is the last train to enter the track. Let further x_i be the block number where the i-th train is currently located and s_i to denote the type of that train. Then given the transition time τ, and if it is not slowed down, this train will move to the blocks $k_L(\tau, i, s_i)$ and $k_H(\tau, i, s_i) = k_L(\tau, i, s_i) + 1$ with probabilities $p_L(\tau, i, s_i)$ and $p_H(\tau, i, s_i)$ respectively (with L for low and H for high). These blocks and probabilities follow from the train speeds and the length of the blocks in the way we discussed before.

Let u be a number between 0 and 1; think of u as being a realization of a uniformly distributed random variable U on $(0, 1]$. If the train does not have to slow down, then for (sample value) u the i-th train will then move to block $k_i^{desired}$ where

$$k_i^{desired} = \begin{cases} k_L(\tau, i, s_i) & \text{if } u \leq p_L(\tau, i, s_i) \\ k_H(\tau, i, s_i) & \text{otherwise} \end{cases} \tag{2.2.7}$$

So, as a result of this way of coupling (via u), trains of the same type will (intend to) move exactly the same number of blocks forward.

Next, given u the actual movement of the trains will be as follows. Let k_i be the new block number of train i. Since the first train can not be hindered by its predecessors, the following holds:

$$k_1 = k_1^{desired} \tag{2.2.8}$$

The subsequent trains may be hindered by their predecessors. As only one train is allowed to occupy a block, the trains need to stay at least one block behind their predecessors and thus the following holds:

$$k_i = min(k_i^{desired}, k_{i-1} + 1) \quad \forall i > 1 \tag{2.2.9}$$

Note that although the movements are formulated for realizations u, the movements are constant on a (very small) number of intervals, which immediately leads to the probabilities for the combined movement of the trains on the destination track. The effect of this is illustrated in Figure 2.1.

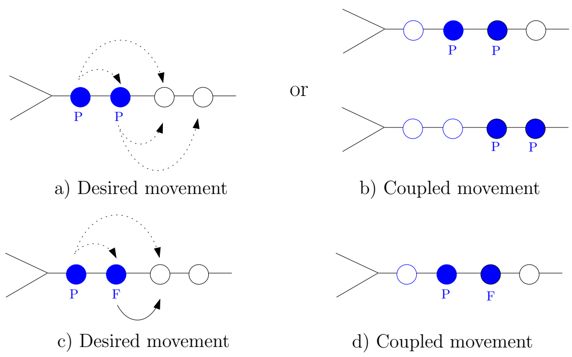

Figure 2.1: Coupling: a) + b) illustrates coupling for identical trains, c) + d) shows a fast passenger train (P) being delayed by a slow freight train (F)

2.2.8 Externality costs

With the description of the model so far, the state space would be countable since there is no limitation with respect to the number of trains on an arrival track. However, for computational purposes, one wants this number to be finite. Therefore, the length of the arrival queue has to be limited. This practice however has one drawback; if the capacity of the track is reached, new arrivals are rejected. The question is, how to handle the arriving trains that can not be 'accepted'? Since the objective of our model is to minimize the stay time of the trains, the stay times of rejected trains should somehow be incorporated. Otherwise, it could be 'profitable' to deliberately block trains from some direction to prevent trains from arriving into the model. To prevent this behaviour from occurring, we introduce the so-called *externality costs* that will represent the stay times of the rejected trains. The concept of the externality costs can not be implemented in an exact way for the complicated model we are dealing with here. Instead, we will approximate our conflict situation with a queueing system. Since the rejected train has the lowest priority, we may view our problem as an M/G/1 queueing system where the server is the junction and the jobs in the queue represent the trains of all arrival tracks waiting in front of the server. For the concept of externality costs see Haviv and Ritov [45]. Externality costs are the costs that a new arrival inflicts on the system as a whole if he would enter. An exact analysis in our situation is not possible, therefore we approximate the conflict junction as an $M|G|1$ system. For a rejected arrival the externality costs, or costs the arrival would have added to the system if he would have been accepted, are approximated as follows.

Let w be an estimate for the total amount of time needed by all trains in the system upon arrival to pass the junction. Let b be an estimate for the amount of time the rejected train would have needed and let ρ be an estimate for the load of the junction, i.e. the time needed by all arriving trains to pass relative to the total time. Let further u be the time the rejected train would have needed on the destination track, if it would not be delayed there. Then the externality costs are taken to be

$$c_{ext} = \frac{w}{1-\rho} + \frac{b}{1-\rho} + u \ . \tag{2.2.10}$$

Of course, in our case, the service time b of a train is not unique and does depend on the speed indicator of the track it arrives at. The evolution of the speed indicators over time depends however on the optimal strategy which is not known yet. To get around this, we will solve the SMD model iteratively, improving at each step the value of the estimates. This is implemented as follows. In a first step, the estimates w, b and ρ are computed based on the assumption that on the arrival track trains are never slowed down thus never incur the acceleration time loss. In a second step the results from the first optimization are used to add the acceleration time loss. (One could add one more iteration, but this turned out to be of no use.)