Chapter 6

The SMD track speed model

In the SMD model a number of modelling choices were made. One of these choices involved the modelling of the destination track. It has been decided to model the track in a fairly detailed way. This detailed way of modelling lead to a large state space description which needed to be reduced by the state space reduction technique introduced in Section 3.6. Moreover, the model raised an issue involving a conflict between the headway concept and the block lengths on the destination track.

In this chapter we discuss a different way to model the destination track which will resolve these issues and thus, in a sense, will be an enhancement for the original model. We start by discussing in Section 6.1 the reasons for this alternative approach. The changes to the model are discussed in Section 6.2. We then elaborate on the model complexity in Section 6.3 and present the results of the new model in Section 6.4. The chapter will be closed with conclusions.

6.1 Reasons for an alternative approach

In previous chapters we have introduced the SMD model and applied it to the Fork_R and bidirectional junctions. The model proved to have good results but at the same time suffered from the state description, which tends to lead to a large state space, and from the conflict between the concept of headway and the block lengths on the destination track. The latter issue has been discussed in detail in Sections 2.2.4 and 2.2.6. The conflict has been settled by respecting the headway time at the entrance and departure from the tracks and by allowing for violations of this rule in between.

The second issue, the state space inefficiency in the SMD model, is induced by the way the destination track is modelled. Since each block of the destination track can be occupied by a train of type s or be empty, the total number of possible states is $(S+1)^n$

where S is the number of train types and n is the number of blocks at the destination track. To tackle this issue, in Section 3.6 a state space reduction technique has been introduced which works well for short destination tracks and a small number of train types. However, in case of a large number of train types visiting a long destination track the technique will still result in a large state space; The state space reduction technique consists of two steps. In the first step, it is calculated whether a train on the destination track forms a hindrance to the trains behind it. The train is removed if no other train suffers from this train. Then in the second step the remaining trains are inspected. If these trains can not hinder any train that may enter the track next, then these trains may be removed from the description too. From the above one easily can construct a number of examples where the reduction technique will be unable to simplify the states: A slow train on the last block followed by any mix of faster trains behind it will not be simplified. A long destination track with a large number of trains on it will in this case result in a large number of states which can not be simplified.

Moreover, in case of the bidirectional junction, only the first step of the reduction technique may be applied. The second step will result in a loss of information and thus may not be executed. This will lead to less states being reduced.

Another inefficiency as a result of the destination track being modelled as a number of blocks comes from the modelling of continuous train movements within the discretized environment. Since the trains are running with different speeds, the number of blocks they cross within some time τ can be non-integer. This is modelled by means of probabilities. This way, one state of the destination track will often lead to more than one new states. This number is positively correlated with the number of train types.

In this chapter we introduce a new way of modelling for the destination track. This new approach does not suffer from the above disadvantages. The state transition of the destination track will lead to only one possible new state depending on the decision being taken. Moreover, the number of possible states of the destination track is limited. In the worst case, this number can exceed the number of states of the SMD model but in most of the cases this number is much lower. Moreover, by aggregating states, this number can be reduced even further. The aggregation process is straightforward and intuitive which is not the case with the state space reduction technique of the SMD model.

Furthermore, in the new model, the description of the destination track will respect the headway concept at all times and not only at the entrance and departure from the tracks.

6.2 Modelling changes

In the new model, the destination track will be modelled in a different way. Instead of keeping track of the location of the trains and their speed, the so-called track speed will be monitored. We will explain this new concept in Section 6.2.1. This new concept will change the way the state space is defined. In Section 6.2.2 we will elaborate on this. The change in state space leads to the change in the transition process. This is discussed in Section 6.2.3 after which the changes to the calculation of the costs are explained in Section 6.2.4.

6.2.1 TrackSpeed variable ts

The key aspect of our new model is the completely new modelling of the destination track. Previously, the state of the destination track was described by the variable x_0 which holds information about the position of every train on the destination track. This information is then used by a train which enters the destination track when computing its stay time there. The trains can not overtake each other and need to respect the minimal headway time upon departing from the track. The idea is that the departure time of the train depends on the departure time of its predecessor. But since this is the case, then one may consider monitoring the departure time of the last train instead of keeping track of all the trains on the destination track. Alternatively, one may monitor the speed of the last train. The latter option gets the preference since it will result in a strategy which is easier to comprehend, as we will see in this chapter.

So, in this chapter we introduce a notion of track speed. Instead of keeping track of the train types that are found on the destination track along with their positions on it, we keep track of the speed of the 'flow' at the destination track. This speed depends on the trains running on the destination track and thus depends on the previous decisions that have been made. We will call this speed the *Track Speed*.

The idea behind this is that keeping track of the *Track Speed*, if done correctly, gives as much information as keeping track of train types and their exact position on the destination track. The major advantage of using track speeds instead, is the significant cut down on the size of the problem, particularly for longer destination tracks and/or a larger number of train types.

Before moving on to the formal definitions and equations, let us first think of what values this variable should have and how these values should evolve in time in order to mimic the underlying continuous train movement.

First of all, the variable should give information about the potential speed of the next
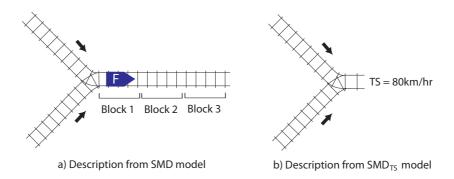
a) Description from SMD model   b) Description from SMD$_{TS}$ model

Figure 6.1: Illustration of the *track speed* concept

arriving train to the destination track, so if the destination track is empty this variable should not be restrictive to the next arriving train. Thus, the default value of ts should be sufficiently high. We will set this default ts value to the speed of the fastest train type.

Next, the lowest value of the ts should correspond to the situation where the slowest train type is found on the first block of the destination track and should be equal to the speed of that train type.

Finally, the following two correlations should exist: the more distance the trains have traversed on the destination track, the higher the ts value. And on the other hand, the longer the time jump, the higher the ts value, since the previous decision has a lower influence on the next one.

6.2.2   Changes to the state space

In Chapter 3 we have defined the state space by vectors x and y. The situation on the destination track was defined by vector x_0 which denoted the types and positions of every train on the destination track. In the new formulation, introduced in this chapter, the state space will be denoted by (x, y, ts) where x is a vector $(x_1, \cdots, x_R)$ and thus does not include x_0 any more.

6.2.3   Changes to transitions

The introduction of the new variable ts has a large influence on the transitions. Also, the computation of the new value ts which depends on several factors needs a special attention. The transition process can be divided into the following three phases:

1. The junction crossing phase: First of all, the train that received permission to cross the junction, leaves the arrival track and crosses the junction. The speed indicators of the arrival tracks are updated. Next,

2. The Track speed update phase: Given the train type, the time jump and the current track speed, the new track speed value is calculated. And finally,

3. The new arrivals phase: new trains arrive at the arrival tracks. This phase is the same as in the SMD model of Chapter 3 and thus will not be explained in this chapter.

6.2.3.1 The junction crossing phase

Since no train movement on the destination track needs to be monitored, the junction crossing phase becomes the first phase of the transition. The authorized train is removed from the first position of its arrival track. The only difference with the SMD model is that the train is not placed on the destination track. If no train is authorized to enter the destination track (decision $a = 0$) then no train leaves the arrival tracks.

The speed indicator of the arrival tracks changes in the same manner as has been previously explained in section 3.4.2.

6.2.3.2 The Track speed update phase

Given the current state (x, y, ts) and the decision a the time is advanced with $\tau(x, y, ts)$. If decision $a = 0$ then $\tau(x, y, ts) = h$ the minimal headway time.

Let us first consider the simplest case where decision $a = 0$. No train enters the destination track. The trains (if any) that are on the destination track move to a new block. In our formulation this means that the value of the track speed will rise (or stay as is, in case of an empty track or trains being too far to hinder others). Now, given the current value of ts, the last train on the destination track will leave that track after $t = \frac{l}{ts} \cdot 3600$ seconds where l is the length of the destination track in kilometres. Since no train enters the destination track, the time advances with h seconds so that the time until the last train leaves the track becomes $t - h$. Now there can be two possibilities: $t - h \leq 0$ or $t - h > 0$. If the former possibility is the case, then we know that the trains on the destination track, if any, will never be able to hinder any future train since the future train can enter the destination track no sooner than after h seconds (minimal time to next decision). So the new track speed value will be the default track speed value $\tau_{default}$, i.e. the speed of the fastest train. When $t - h > 0$ then there are still trains on the destination track which can hinder next arrivals. The new track speed will then increase to $\frac{l}{t-h} \cdot 3600$. However, the value of ts_{new} should be bounded by $ts_{default}$ to limit the number of different track speed values.

Summarizing, if the decision is $a = 0$ then the new track speed value is equal to

$$
ts_{new} = \begin{cases} min(ts_{default}, \frac{l}{t-h} \cdot 3600) & \text{if } t - h \geq 0 \\ ts_{default} & \text{if } t - h < 0 \end{cases} \tag{6.2.1}
$$

The computation of the new ts value becomes more complicated when decision a involves a train that is authorized to enter the destination track. To calculate the new track speed value, we first need to calculate the time this train will leave the destination track, which depends on the speed of the train and on the track speed.

Let us denote the desired speed of the train which is authorized to enter the destination track by sp. Then the train will traverse the destination track in $t^{desired}$ seconds where

$$
t^{desired} = \frac{l}{sp} \cdot 3600 \tag{6.2.2}
$$

On the other hand, the track speed is ts which means that the trains on the destination track will leave the track in $t^{flow} = \frac{l}{ts} \cdot 3600$ seconds. Since the authorized train will enter the destination track only after $\tau(x, y, ts)$ seconds, t^{flow} becomes

$$
t^{flow} = \frac{l}{ts} \cdot 3600 - \tau(x, y, ts) \tag{6.2.3}
$$

The authorized train can then leave the destination track at time:

$$
t = max(t^{desired}, t^{flow} + h^*) \tag{6.2.4}
$$

The new ts value is then set to

$$
ts_{new} = \frac{l}{t} \cdot 3600 \tag{6.2.5}
$$

The h^* within equation (6.2.4) can refer to the minimal headway which separates trains in time. However, in our case, we will set the value of h^* in such a way as to mimic the departure process of the SMD model of the previous chapters. When examining the behaviour of trains at the destination track in case of the SMD model, one can see that whenever the slow train has left the track, all information about the train is lost. A faster train which run directly behind the slow train is now the first train at the track and will run with maximal speed the last part of the track. To mimic this behaviour and make results of the SMD$_{ts}$ model as close as possible to the SMD model, we will set the value of

h^* to the time the train will run the last block of the destination track if it were the first train on the track. So $h^* = \frac{bl}{sp} \cdot 3600$ where bl is the length of one block of the destination track and sp the desired speed of the train, so in this case h^* will depend on the type of the train which is authorised to cross the junction.

Finally, consider the case where the train, that enters the destination track, leaves the track somewhere halfway through it. Since the track speed is an indicator for the speed of the 'flow' on the whole track, such a train should have only a limited influence on the track speed. Therefore, in the Equation 6.2.2 one should consider the desired speed of the train sp only for a distance d that the train will spend on the track and applying the maximal track speed value, which is $ts_{default}$ on the rest of the track. The Equation 6.2.2 changes to

$$t^{desired} = \frac{d}{sp} \cdot 3600 + \frac{l - d}{ts_{default}} \cdot 3600 \tag{6.2.6}$$

Of course, by modelling in this way, the actual situation is simplified, since the single variable ts indicates that the speed of the flow is constant on the whole track while the flow can be slower at the beginning of the track than on the end of it. In order to incorporate this, one could use multiple track speeds, each representing the speed of the flow on a portion of the track. In this chapter we would like to keep things simple and will not consider multiple track speeds. In Chapter 9 it is discussed how the model can be extended.

6.2.4 Changes to costs

Our goal is again to minimize the total traverse time in the system. As has been explained previously in Section 3.5 there are three cost components that together form the total costs. These are the arrival track costs, the destination track costs and the train rejection costs. The arrival track costs refer to the stay times of the trains on the arrival tracks. Since there are no changes to the arrival tracks, the computation of these costs is not altered. The destination track costs are replaced by the track speed costs c_{ts} which are computed in the following way.

Every time a train makes a transition from the arrival track to the destination track the total stay time of the train on the destination track is calculated. The stay time depends on the track speed ts and on the speed sp of the train which is authorised to enter the track. Let us denote the distance the train will cross on the destination track by d. Let us further denote the time the last train on the track, if any, will leave the track by t^{flow}. This value can be computed also for the situation where no train is on the track

since the computation involves the track speed value which will be equal to $ts_{default}$ in case of an empty track. The t^{flow} is calculated in the following way:

$$t^{flow} = \frac{d}{ts} \cdot 3600 \tag{6.2.7}$$

Note, that t^{flow} is calculated over the distance d which thus indicates the time the 'flow' crosses the first d kilometres of the track. The train that crosses the junction, will cross the same distance in $t^{desired}$ seconds, where

$$t^{desired} = \frac{d}{sp} \cdot 3600 \tag{6.2.8}$$

The track speed costs are then:

$$c_{ts} = max(t^{flow} - \tau(x, y, ts) + h^*, t^{desired}) \tag{6.2.9}$$

6.3   Computational complexity

The computational complexity is determined by the number of states, the number of transitions per state, the number of possible decisions and the convergence rate of the iteration process.

The state space of the track speed model is defined either by the triple (x, y, ts) or in case of the bidirectional destination track by (x, y, z, ts). The variable x holds information about the situation on each of the arrival tracks. The number of possible states of variable x depends on the number of arrival tracks R, the number of train types S_r that make use of each of these tracks and the maximum capacity of these tracks n_r. Since all arrival tracks are modelled as queues, the number of states of variable x_r is equal to $1 + S_r + S_r{}^2 + \cdots + S^{n_r} = \frac{S_r{}^{n_r+1}-1}{S_r-1}$.

The variable y depends on the number of speed indicators J and the number of arrival tracks R which accounts for J^R states. Variable z is a binary variable and thus has two states. Let further T represent the number of different value of ts. Then, the total number of states is equal to $2 \cdot T \cdot \prod_r \frac{S_r{}^{n_r+1}-1}{S_r-1} \cdot J^R$.

The number of transitions does no longer depend on the number of transitions on the destination track and is only determined by the arrival process $\mathcal{HP}$.

In the remainder of this section we will show how the different values of track speeds are found to get some feeling about the order of magnitude of this variable. The following steps are executed in order to compute all possible values of the track speed variable.

1) **Initialization of list** L**:** For every possible decision a we store the value of the train type s which is authorized to cross the junction together with the corresponding time jump τ. Note, that s can also be 0 which indicates that no train is authorized. This information is stored in list L.

2) **Definition of recursive function:** Define function $FindSpeeds(ts)$ which for a given track speed ts calculates all possible new track speeds. The working of the function is as follows: Consider all possible combinations (s, τ) of list L and let train type s enter the destination track. The track speed changes to ts'. If ts' is not within the list TS, add speed ts' to the list[1] and call function $FindSpeeds(ts')$.

3) **Start of recursion:** Start with an empty destination track. The current track speed is then the default track speed $ts = default$. Initialize the array of found track speeds to $TS = \{ts\}$. Call now function $FindSpeeds(ts)$.

4) **Aggregation step:** Aggregate all speeds of array TS which differ by at most d km/hr from each other, where d is a predefined parameter. This step is executed in the following way: Sort the list TS. Consider the first value (the lowest track speed) and find all values which fall within the range of d km/hr. Take the average of these values and include this averaged value in the aggregated set AS. Now, proceed from the first element of set TS which fell outside the range d and repeat the same procedure until all elements of set TS have been processed.

In theory, the recursive function $FindSpeeds()$ can result in a large number of different track speeds. In fact, the number of track speeds will be at most $\lceil \frac{sp_f - sp_s}{\epsilon} \rceil$ where sp_f is the speed of the fastest train type on the destination track, sp_s is the speed of the slowest train on the destination track and ϵ being a predefined value of 0.1 km/hour. In practice though, a lot of these speeds will be equal or almost equal to each other. Moreover, by aggregating the speeds in step 4, we limit the number of different speeds even further to $\lceil \frac{sp_f - sp_s}{d} \rceil$ where d represents the range in km/hour within which the speeds are said to be equal.

---

[1]To limit the size of the list TS and rule out any round-up errors, we consider speeds within the range of $\epsilon = 0.1$ km/hour to be equal. Thus, ts' is added to the list TS only if there are no speeds in the list within the range of ϵ.

6.4   Results

The strategy of the SMD_{ts} model will be referred to as the SMD_{ts} strategy. First of all we will examine the structure of the SMD_{ts} decisions and compare this structure to that of the SMD model of Chapter 3. This will be done in Section 6.4.1. Then in Section 6.4.2 we will show the performance differences between the SMD and the SMD_{ts} for a number of scenarios.

6.4.1   Structure of the SMD_{ts} strategy

Let us consider the same basic scenario that has been studied in Section 4.3. The destination track consists of 3 blocks and the two train types that are considered have the following characteristics:

Characteristics	P	F
Speed (km/hr)	120	80
Approach time (sec)	180	270
Acceleration time loss (sec)	25	75
Arrival rate (per hour)	8	4

Table 6.1: Characteristics of the trains in the basic scenario

This information, in combination with equations (6.2.1) and (6.2.5) gives us the following set of possible track speeds: 80, 87, 94, 101, 109, 116 and 120 km/hr.

Table 6.3 shows the SMD_{ts} decision matrix which provides information about the SMD_{ts} strategy for every possible state. The explanation about how to read this matrix and how it is constructed is given in Section 4.3. The difference now, is that the values within the matrix reflect the track speeds. These values should be read as following:

Take decision $a = 1$ if the track speed is less or equal to the value found in the matrix. Otherwise, take decision $a = 2$

In Table 6.2 a number of examples with the corresponding interpretation is given.

Value	Interpretation
0	Decision is always 2
80	Decision is 1 if trackspeed $= 80$ and 2 otherwise
94	Decision is 1 if trackspeed ≤ 94 and 2 otherwise
120	Decision is always 1 (since $ts_{default} = 120$)
-	Both arrival tracks are empty. Decision is 0

Table 6.2: Interpretation of the values in the decision matrix of the SMD_{ts} strategy for the basic scenario

x_2	y_1	y_2	-	**P**	**P P**	**F P**	**F**	**P F**	**F F**
-	0	0	0	120	120	120	120	120	120
-	1	0	0	120	120	120	120	120	120
-	0	1	0	120	120	120	120	120	120
-	1	1	0	120	120	120	120	120	120
P	0	0	0	120	120	0	0	0	0
P	1	0	0	120	120	120	120	120	120
P	0	1	0	0	0	0	0	0	0
P	1	1	0	120	120	80	120	120	120
P P	0	0	0	0	120	0	0	0	0
P P	1	0	0	120	120	0	101	120	120
P P	0	1	0	0	0	0	0	0	0
P P	1	1	0	0	120	0	80	109	80
F P	0	0	0	120	120	120	0	0	0
F P	1	0	0	120	120	120	120	120	120
F P	0	1	0	0	120	0	0	0	0
F P	1	1	0	116	120	120	94	120	120
F	0	0	0	120	120	120	120	120	120
F	1	0	0	120	120	120	120	120	120
F	0	1	0	0	94	0	0	0	0
F	1	1	0	0	116	101	120	120	120
P F	0	0	0	120	120	120	0	120	0
P F	1	0	0	120	120	120	120	120	120
P F	0	1	0	0	0	0	0	0	0
P F	1	1	0	0	87	0	0	120	0
F F	0	0	0	120	120	120	0	120	120
F F	1	0	0	120	120	120	120	120	120
F F	0	1	0	0	0	0	0	0	0
F F	1	1	0	0	116	0	0	120	120

Table 6.3: SMD_{ts} decision matrix of the basic scenario. For each combination of (x_1, x_2, y_1, y_2) an upper bound for the value of the track speed (in km/hr) is given for which it holds that according to the SMD rule a train from arrival track 1 should cross the junction first. I.e. Take decision $a = 1$ if the track speed is less or equal to the value found in the matrix. Otherwise, take decision $a = 2$

Let us compare these decisions with the SMD decisions for the same scenario that can be found in table 4.3 (Section 4.3). A close examination of these strategies reveals that there are only a few differences. These differences are presented in table 6.4.

State				SMD Decision	SMD$_{ts}$ Decision
x_1	x_2	y_1	y_2		
F P	P	0	0	$a = 2$ unless $x_0 =$ (F - -)	$a = 2$ always
F P	P	1	1	$a = 2$ unless $x_0 =$ (F - -), (P F -)	$a = 1$ if $ts \leq 80$
F P	P P	1	0	$a = 2$ unless $x_0 =$ (F - -)	$a = 2$ always
F	P P	1	0	$a = 2$ unless $x_0 =$ (F - -), (P F -), (P P F)	$a = 1$ if $ts \leq 94$
F F	P P	1	0	$a = 2$ unless $x_0 =$ (F - -), (P F -), (P P F)	$a = 1$ always
F	P P	1	1	$a = 2$ unless $x_0 =$ (F - -)	$a = 1$ if $ts \leq 80$
P F	P P	1	1	$a = 2$ unless $x_0 =$ (F - -), (P F -), (P P F)	$a = 1$ if $ts \leq 101$
F F	P P	1	1	$a = 2$ unless $x_0 =$ (F - -)	$a = 1$ if $ts \leq 80$
F	F P	1	1	$a = 2$ unless $x_0 =$ (F - -), (P F -), (P P F)	$a = 1$ if $ts \leq 87$

Table 6.4: Overview of states of the basic scenario for which SMD decision differs from the SMD$_{ts}$

The table should be read as follows: when in state $x_1 = (F\ P)$, $x_2 = (P)$, $y_1 = 0$ and $y_2 = 0$ (see row 1 of the table), then the SMD decision prescribes that track 1 gets the right of way only when a freight train is found on the first block of the destination track (state $x_0 = $ (F - -)) while the corresponding SMD$_{ts}$ decision will never give trains from track 1 the right of way.

While from the second row of the table we learn that the SMD decision will give trains of track 1 the right of way when either the freight train is found on the first block (state $x_0 = $ (F - -)) or the freight train is on the second block and followed by the passenger train (state $x_0 = $ (P F -)). The corresponding SMD$_{ts}$ decision will authorize trains from track 1 to cross the junction only if the trackspeed is 80 km/hr ($ts = 80$).

From table 6.4 we see that, as expected, the differences between the two strategies are very limited so the performance of the two strategies should be very similar as well.

6.4.2 Performance of the SMD$_{ts}$ strategy

To compare the results of the SMD$_{ts}$ strategy with that of the SMD strategy we again use simulation. To make the comparison fair we will run identical runs, i.e. the arrival times and the types of the trains that arrive are identical in both runs. When a conflict situation arises, the conflict will be solved either by SMD or SMD$_{ts}$ rules. Regardless of the rule used, the state of the destination track within the simulation environment is modelled through the x_0 variable, i.e. for every train on the destination track the location and its type is known. So when the conflict is solved with the SMD rule, no translation is needed, the corresponding SMD decision can be found for exact same state representation. The situation is different when the conflicts are solved with the SMD$_{ts}$ rules. In this case, the x_0 representation of the destination track is first translated to the trackspeed-equivalent after which the corresponding SMD$_{ts}$ rule can be looked up.

The performance of the two strategies turns out to be almost identical. The measured mean delays differ from each other only by a couple of one tenths of a second which is not significantly large. Table 6.5 shows the differences in case of the basic scenario while Table 6.6 shows the differences for various values of acceleration time loss of freight trains.

Discipline	P	F	Mean
SMD	234.9	138.4	202.8
SMD$_{ts}$	233.4	142.9	203.3

Table 6.5: Difference in performance between the SMD and the SMD$_{ts}$ strategies for the basic scenario

Acceleration time loss (in sec.)	SMD	SMD$_{ts}$
50	194.8	194.8
75	202.8	203.3
100	205.9	206.2

Table 6.6: Difference in performance between the SMD and the SMD$_{ts}$ strategies for different values of acceleration time loss of the freight trains

By examining the values of Tables 6.7 and 6.8, which show differences between the SMD and the SMD$_{ts}$ performances for different values of the system load and different

values of the destination track length respectively, we can conclude that the two strategies are very similar indeed.

Number trains per hour	SMD	SMD$_{ts}$
0.1	0.7	0.7
1	6.9	6.9
2	14.7	14.7
3	23.3	23.3
4	33.0	33.1
5	43.8	43.8
6	56.1	56.3
7	70.2	70.4
8	86.6	86.8
9	106.1	106.1
10	130.1	130.2
11	160.4	160.4
12	202.8	203.3
13	260.6	261.0
14	352.2	352.5

Table 6.7: Difference in performance between the SMD and the SMD$_{ts}$ strategies for different values of system load

Length destination track in km	SMD	SMD$_{ts}$
4	154.3	154.3
8	176.5	176.3
12	202.8	203.3
16	229.9	230.0
20	261.2	260.6

Table 6.8: Difference in performance between the SMD and the SMD$_{ts}$ strategies for different values of destination track length

6.4.3 Aggregation of track speeds

In this section we will examine the influence of the track speed aggregation on the performance of the SMD_{ts} strategy. As a testing case for this research we will use the basic scenario and will vary the length of the destination track. The idea is, the longer the destination track, the larger the state space of the SMD model and the higher the urge for compact alternatives.

Already when the destination track consists of 5 blocks of 4 kilometre each, the number of possible states of the destination track when using the SMD model is $(S+1)^5 = 243$. When the state space reduction technique of Chapter 3 is used, the number of states reduces to 8:

(- - - - -), (F - - - -), (- F - - -), (P F - - -), (- P F - -), (P P F - -), (- P P F -) and (P P P F -).

With the track speed model, the number of states depends on the threshold value which is used to define distinctive track speeds. Table 6.9 shows that if the threshold value is 0.1 (meaning that the two speeds are considered to be different if the difference is at least 0.1km/hr), then the number of track speeds is 14. This number reduces when the threshold value is risen.

Threshold value in km/hr	Number of track speeds	Track speeds in km/hr
0.1	14	80, 86, 88, 92, 95, 99, 100, 104, 107, 109, 112, 113, 118, 120
5	7	80, 87, 94, 101, 109, 116, 120
10	4	85, 97, 109, 119
20	2	91, 112
40	1	102

Table 6.9: Relation between the threshold value and the number of track speeds

Table 6.10 shows the influence of the track speed aggregation on the performance of the SMD_{ts}-strategy. The number behind the SMD_{ts}-strategy (e.q. SMD_{ts} 5) represents the threshold value that is used. The numbers between brackets represent the number of track speeds that are distinguished.

One can see, that the more speeds are aggregated, the worse the performance of the SMD_{ts} strategy. The performance approaches the level of that of the Follow strategy. This is not surprising, since when only 1 track speed is used, the SMD strategy does

not have any information about the position and the type of trains behind the junction. Follow strategy does not use this information either. And since both strategies try to maximize the capacity of the junction, the only difference between the Follow strategy and the SMD_{ts} strategy becomes the statistical information about the future arrivals which is used by SMD_{ts} and is not used by Follow strategy.

Discipline	Track length (in km)				
	12	20	28	40	60
SMD 0.1	203 (7)	261 (14)	324 (24)	423 (45)	598 (93)
SMD 5	203 (5)	261 (7)	324 (7)	423 (8)	598 (8)
SMD 10	203 (4)	261 (4)	324 (4)	423 (4)	598 (4)
SMD 20	203 (2)	267 (2)	325 (2)	424 (2)	609 (2)
SMD 40	205 (1)	267 (1)	332 (1)	432 (1)	609 (1)
Follow	206	267	332	432	608
P-F-Follow	245	306	369	469	646
F-P-Follow	229	290	355	457	635
P-F	245	306	369	469	646
F-P	229	290	355	457	635
FCFS	269	330	395	497	676

Table 6.10: The influence of the track speed aggregation on the performance of the SMD strategy

Another observation that can be made from Table 6.10 is that the longer the destination track the bigger the difference between the SMD 0.1 strategy and the Follow strategy. The threshold value of 0.1 km/hr results in a large number of track speeds. Using a threshold value of 5 km/hr seems to be a safe aggregation policy. The number of states is then low while the high level of performance is still maintained.

6.5   Conclusions

In this Chapter an alternative way of modelling of the destination track is presented. The resulting model is called SMD_{ts} model. Compared to the SMD model, both the problem size and number of transitions is reduced: The problem size is reduced by the introduction of the track speed variable which keeps track of the possible speed on the destination track and eliminates the need of including both the location and the type of trains on the destination track. Moreover, since the continuous train movement over the

discretized blocks is not modelled, the transition of the state no longer leads to multiple new states of the destination track.

Finally, the track speed model allows for an intuitive state space reduction by grouping track speeds together, which do not differ much from each other. Simulation results show that the results of the model are almost identical to those of the SMD model and thus superior as well when compared to other strategies.

Modelling the situation on the destination track by a single track speed variable has however also its disadvantages. First, it is assumed that the speed of the flow is constant on the whole track. In case a number of trains leave the track halfway through it, the speed of the flow can be higher at the end of the track than it is the case at the beginning of it. This disadvantage can be overcome by inclusion of multiple track speeds into the model, each representing the speed of the flow at a certain portion of the track. Of course this solution will increase the size of the state space. Another disadvantage is that by keeping track of only the track speed, any information about the type of the last train on the track is lost. In some cases, this information can be valuable. For instance in the case when the headway, the minimal time between two subsequent trains, at the end of the destination track, should depend on the type of these two trains (e.g. two freight trains running behind each other need larger headway in between than a freight train followed by an intercity train). Should this distinction be required, then we suggest to use a SMD model instead or construct a hybrid model which keeps track of both: track speed and the type of the last train entering the track. Either way, the purpose of this chapter was to look critically at the SMD model and present an alternative way of modelling which, when used in certain cases, would yield comparable solutions with a more compact state space. The track speed model offers a number of advantages that the original model lacks and can be freely extended to incorporate additional functionality. The reader interested in model extensions is referred to Chapter 9 where among other extensions the incorporation of the multiple track speeds is discussed.