Modeling and Simulation of Train Networks using MaxPlus Algebra

Hazem Al-Bermanei1   Jari M. Böling2   Göran Högnäs3

1Faculty of Business ICT and Life Sciences, Turku University of Applied Sciences, Turku, Finland, hazem.al-bermanei@turkuamk.fi
2Department of Chemical Engineering, Åbo Akademi University, Turku, Finland, jboling@abo.fi
3Department of Mathematics and Statistics, Åbo Akademi University, Turku, Finland

Abstract

Max-plus algebra provides mathematical methods for solving nonlinear problems that can be given the form of linear problems. Problems of this type, sometimes of an administrative nature, arise in areas such as manufacturing, transportation, allocation of resources, and information processing technology. Train networks can be modelled as a directed graph, in which nodes correspond to arrivals and departures at stations, and arcs to travelling times. A particular difficulty is represented by meeting conditions in a single-track railway system. Compared to earlier work which typically include numerical optimization, max-plus formalism is used throughout this paper. The stability and sensitivity of the timetable is analyzed, and different types of delays and delay behavior are discussed and simulated. Interpretation of the recovery matrix is also done. A simple train network with real world background is used for illustration.

Keywords: train schedules, meeting conditions, max-plus algebra, discrete-event systems, delay sensitivity, recovery matrix

1. Introduction

The increasingly saturated European railway infrastructure has, among other concerns, drawn attention to the stability of train schedules as they may cause of domino effect delays across the entire network. A train timetable must be insensitive with regard to small disturbances so that recovery from such disturbances can occur without external control. After a break of self-regulation, this behavior schedule requires the distribution of accurate recovery times and buffer times to reduce delays and prevent the propagation of delay, respectively. Schedule models for railways are usually based on deterministic process times (running times, and transfer times). Moreover, running times are rounded and train tracks are modified to fit the schedule or constraints. The validity of these decisions and streamline schedules must be evaluated to ensure the viability and stability and durability, with respect to network mutual relations and differences in process times. Train networks can be modeled using max-plus algebra (D’Ariano et al., 2007). Stability can be evaluated by calculating the eigenvalue of the matrix in max-plus algebra (Baccelli et al., 1992; van den Boom and De Schutter, 2004; van den Boom et al., 2012; Corman et al., 2012). This eigenvalue is the minimum cycle time required to satisfy all of the schedule and progress constraints, where the timetable operating with this eigenvalue time is given by the associated eigenvector (Baccelli et al., 1992; De Schutter and van den Boom, 2008). Thus, if the eigenvalue λ is more than the intended length of the schedule T, then the schedule is unstable. If λ<T the schedule will be stable, and critical if λ=T (van den Boom et al., 2012; Corman et al., 2012).

If individual trains are delayed, the effect on the whole network is quite difficult to predict. Smaller delays can typically be absorbed by speeding up the trains, and this can be handled by using max-plus algebra. Larger delays are often handled by rescheduling, typically using optimization, see for example De Schutter et al., (2002); D’Ariano et al., (2007); Corman et al., (2012); and van den Boom and De Schutter, (2004).

In this paper we study the impact of both permanent and dynamic delays in a train network, but restrict ourselves to using max-plus algebra, and thus we do not consider rescheduling. So in practice our study is limited to delays up to half of the cycle time. Meeting conditions, including those introduced by having single tracks, are also fully handled using max-plus state-space formalism, by extending the state with delayed states. When constructing a recovery matrix (van den Boom et al., 2012). Of this extended system, it naturally results in redundancy, as the same physical state appears many times. This redundant recovery information can however be incorrect, due to that no constraints are specified for the delayed states, which are only shifted copies of the most recent state. The parts of the recovery matrix corresponding to the most recent states are still valid.
2. Max-plus algebra

In max-plus algebra we work with the max-plus semi-ring which is the $\mathbb{R}_{\text{max}} = \mathbb{R} \cup \{-\infty\}$ and the two binary operations addition $\oplus$ and multiplication $\otimes$, which are defined by:

$$a \oplus b = \max(a, b), \quad a \otimes b = a + b, \quad \text{and} \quad (-\infty) + a = -\infty.$$ 

Define $\varepsilon = -\infty$ and $e = 0$. The additive and multiplicative identities are thus $\varepsilon$ and $e$ respectively and the operations are associative, commutative and distributive as in conventional algebra. Furthermore the pair of operations $(\oplus, \otimes)$ can be extended to matrices and vectors similarly as in conventional linear algebra:

- For all $A, B \in \mathbb{R}_{\text{max}}^{n \times n}$, $(A \oplus B)_{ij} = a_{ij} \oplus b_{ij} = \max(a_{ij}, b_{ij})$
- For $A \in \mathbb{R}_{\text{max}}^{m \times n}$ and $B \in \mathbb{R}_{\text{max}}^{n \times p}$ define their product by

$$AB = \sum_{k=1}^{n} (a_{ik} \otimes b_{kj}) = \max_{l=1, \ldots, k} (a_{il} + b_{lj})$$

1 ≤ l ≤ m, 1 ≤ l ≤ p

The $n \times n$ identity matrix $I_n$ in max-plus is defined as:

$$I_n = \begin{cases} 1 & \text{if } i = j \\ \varepsilon & \text{if } i \neq j \end{cases}$$

For $A \in \mathbb{R}_{\text{max}}^{m \times n}$, $I_n \otimes A = A \otimes I_n = A$.

- For a square matrix $A$ and positive integer $n$ the $n^{\text{th}}$ power of $A$ is written as: $A^\otimes n$ and it is defined by

$$A^\otimes n = A \otimes A \otimes \ldots \otimes A \quad \text{with} \quad n \text{ times}$$

See also Heidegrott et al. (2006), De Schutter and van den Boom (2004), and Baccelli et al. (1992).

3. Scheduled max-plus linear systems

Consider the train network in Figure 1 (vr.fi, 2014). This is a simple network consisting of four stations, Helsinki (H), Karjaa (K), Salo (S) and Turku (T). The end stations are modeled with nodes for both arrival (A in front of the city first letter) and departure (D). The stops at the intermediate stations are short, and thus only the departures are modeled. The weights $d_i$ on the arcs corresponds to the traveling times, while $d_i$ and $d_s$ are service times at end stations. The stations between Helsinki and Karjaa are connected by double tracks, and the other connections are single tracks that introduce meeting time conditions. There are five trains available for this, which also introduce some constraints:

![Figure 1](image)

Table 1 provides the schedule (vr.fi, 2014) of five trains running regularly between Helsinki and Turku, and gives the information in connection with the nominal travelling times and the departures.

**Table 1**: Train time table for trains 1, …, 5 between Turku and Helsinki in hours: minutes. Abbreviations: D=departure, A=arrival, T=Turku, and H=Helsinki.

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>DH</td>
<td>8:02</td>
<td>9:02</td>
<td>10:02</td>
<td>12:02</td>
<td>11:02</td>
</tr>
<tr>
<td>AT</td>
<td>10:00</td>
<td>11:00</td>
<td>12:00</td>
<td>14:00</td>
<td>13:00</td>
</tr>
<tr>
<td>DT</td>
<td>11:00</td>
<td>12:00</td>
<td>13:00</td>
<td>15:00</td>
<td>14:00</td>
</tr>
<tr>
<td>DH</td>
<td>13:02</td>
<td>14:02</td>
<td>15:02</td>
<td>17:02</td>
<td>16:02</td>
</tr>
<tr>
<td>AT</td>
<td>15:00</td>
<td>16:00</td>
<td>17:00</td>
<td>19:00</td>
<td>18:00</td>
</tr>
<tr>
<td>DT</td>
<td>16:00</td>
<td>17:00</td>
<td>18:00</td>
<td>20:00</td>
<td>19:00</td>
</tr>
</tbody>
</table>

Now, in order to define the train network as a discrete event system (DES), a state vector is defined as $x = (x_{DH}, x_{KS}, x_{ST}, x_{DT}, x_{SK}, x_{KH}, x_{AH})^T$ with descriptive subscripts:

- **DH** = departure from Helsinki
- **KS** = departure from Karjaa to Salo
- **ST** = Salo to Turku
- **KH** = Karjaa to Helsinki
- **AH** = arrival to Helsinki
- **DT** = departure from Turku
- **AH** = arrival to Helsinki

The argument $k$ on the states denote the kth departure or when indicated arrival to the end stations. Furthermore $k$ also indicates the train number, so that $x_{DH}(k)$ is the departure time from Helsinki for train $k$, and $x_{AT}(k)$ is the arrival of the same train to Turku.

The period of the time table is T=60 minutes. Due to that we have only one track between Karjaa and Salo, and between Salo and Turku, we get the following meeting conditions:
In order to get an equation of type $x(k) = A \Theta x(k-1)$, the right hand side expressions containing $k$ or higher indices are substituted with expressions containing index $k-1$ at most:

1. $x_{DH}(k) = x_{AH}(k - 5) + d_1,$
2. $x_{XS}(k) = \max (x_{DH}(k) + d_2, x_{KH}(k - 3)),$
3. $x_{ST}(k) = \max (x_{XS}(k) + d_3, x_{SK}(k - 2)),$
4. $x_{AT}(k) = x_{ST}(k) + d_4,$
5. $x_{DT}(k) = \max (x_{AT}(k) + d_5, x_{AT}(k + 1)),$
6. $x_{SK}(k) = \max (x_{DT}(k) + d_6, x_{ST}(k + 2)),$
7. $x_{KH}(k) = x_{SK}(k) + d_7,$
8. $x_{AH}(k) = x_{KH}(k) + d_8.$

In order to get an equation of type $x(k) = A \Theta x(k-1)$, the right hand side expressions containing $k$ or higher indices are substituted with expressions containing index $k-1$ at most:

Define the augmented system $x_j(k)$ where $j = 1, 2, 3, \ldots, A0$:

\begin{align*}
    x_j(k) &= x_{DH}(k - j + 1), & j = 1, \ldots, 5 \\
    x_j(k) &= x_{XS}(k - j + 6), & j = 6, \ldots, 10 \\
    x_j(k) &= x_{ST}(k - j + 11), & j = 11, \ldots, 15 \\
    x_j(k) &= x_{AT}(k - j + 16), & j = 16, \ldots, 20 \\
    x_j(k) &= x_{DT}(k - j + 21), & j = 21, \ldots, 25 \\
    x_j(k) &= x_{SK}(k - j + 26), & j = 26, \ldots, 30 \\
    x_j(k) &= x_{KH}(k - j + 31), & j = 31, \ldots, 35 \\
    x_j(k) &= x_{AH}(k - j + 36), & j = 36, \ldots, 40 \\
\end{align*}

This means that $x_i(k) = x_{i-1}(k - 1)$ for $i = 2, 3, \ldots, 40$ except $i = 1, 6, 11, 16, 21, 26, 31$ and 36. The main equations using numbers as subscripts then become as follows:

1. $x_4(k) = x_{x40}(k - 1) + d_1,$
2. $x_6(k) = \max (x_{x40}(k - 1) + d_1, d_2, x_{x33}(k - 1)),$
3. $x_{14}(k) = \max (x_{x40}(k - 1) + d_1, d_2, d_3, x_{x27}(k - 1), x_{x33}(k - 1) + d_3),$
4. $x_{16}(k) = \max (x_{x40}(k - 1) + d_1, d_2, d_3 + d_4 + d_5, x_{x27}(k - 1) + d_4, x_{x33}(k - 1) + d_3 + d_4),$
5. $x_{21}(k) = \max (x_{x40}(k - 1) + d_1, d_2, d_3, d_4 + d_5, x_{x27}(k - 1) + d_4, x_{x33}(k - 1) + d_1 + d_2 + d_3 + d_4 + d_5, x_{x26}(k - 1) + d_1, x_{x32}(k - 1) + d_1 + d_2 + d_3 + d_4 + d_5, x_{x39}(k - 1) + d_4 + d_5 + d_6, x_{x26}(k - 1) + d_1, x_{x32}(k - 1) + d_1 + d_2 + d_3 + d_4 + d_5 + d_6, x_{x39}(k - 1) + d_1, d_2 + d_3, x_{x31}(k - 1) + d_3),$
6. $x_{31}(k) = \max (x_{x40}(k - 1) + d_1, d_2, d_3, d_4 + d_5, d_6, x_{x27}(k - 1) + d_3 + d_4 + d_5 + d_6, x_{x33}(k - 1) + d_1 + d_2 + d_3 + d_4 + d_5 + d_6, x_{x39}(k - 1) + d_4 + d_5 + d_6, x_{x26}(k - 1) + d_1, x_{x32}(k - 1) + d_1 + d_2 + d_3 + d_4 + d_5 + d_6, x_{x39}(k - 1) + d_1, d_2 + d_3, x_{x31}(k - 1) + d_3).$
If we rewrite the above evolution equations as a max-
plus-linear discrete event systems state space model of
the form
\[ x(k) = A \otimes x(k-1) \]
we obtain a square matrix \( A \) of size 40x40. For
example the 36th row in the matrix \( A \) is:
\[ [\varepsilon \varepsilon \varepsilon \varepsilon 148 208 \varepsilon \varepsilon 115 175 235 \varepsilon \varepsilon \varepsilon \varepsilon 180 240 300] \]
where the entry 148 has column index 26.

The power method (Baccelli et al., 1992; van den
Boom and De Schutter, 2004; De Schutter and van
den Boom, 2008) is used for finding the eigenvalue \( \lambda \) of
the matrix \( A \). The method means repetitive multiplications
\( x(0) = A \otimes x(0) \), and it stops when
there are integers \( i > j \geq 0 \) and a real number \( c \) for
which
\[ x(i) = x(j) \otimes c \]. The eigenvalue is then given by
\( \lambda(A) = \frac{\epsilon}{i-j} \). In this case, using \( x(0) = \mathbf{0} \), iteration
according Equation 2 gives
\[ x(12) = A \otimes x(11) = [664 604 544 484 424 725
665 605 545 485 52 692 632 572 512 782
722 662 602 542 842 782 722 662 602 872
812 752 692 632 900 840 780 720 660 960 900
840 780 720]^T \]
\[ x(13) = A \otimes x(12) = [724 664 604 544 484 785 725 665 605 545
812 752 692 632 572 842 782 722 662 602
902 842 782 722 662 932 872 812 752 692
960 900 840 780 720 1020 960 900 840 780]^T \]
\[ x(13) = x(12) \otimes 60 \]
Thus the eigenvalue is \( \lambda(A) = 60/(13 \times 12) = 60 \). The
eigenvalue represents the cycle of the schedule which
means that the trains start from each station every 60
minutes.

This also means that \( x(13) - c \), where \( c \) is any constant, is also an
eigenvector. One eigenvector of \( A \) is \( v \) where
\[ v = [10 \ 90 \ 1 \ 88 \ 28 \ 8 \ 52 \ 32 \ -92 \ -152 \ -118 \ 58 \ -2] \]
This eigenvector \( v \) includes the schedule of the trains,
relative to the last departure from Helsinki (the first
element of \( v \)). So the element -240 means that five
departures back a train from Helsinki left 240 minutes
ago, and the element 296 means that it takes 296
minutes for a train to come back to Helsinki.

4. Timetable stability
4.1 Delay sensitivity analysis
All the travel times \( d_i \) introduced in the Section 3,
consist of a minimal travel time and a slack time. Here
it is assumed that the minimal travel time is 90% of
the nominal time, and the slack is thus 10%. For the
small waiting time \( d_i \) in Helsinki it is assumed that there is
no slack.

Handling delays is a relevant and common problem in
train networks, and the sensitivity of delays can be
analyzed using max-plus models. A permanent delay
means that the nominal travel times is increased, which
is compensated for by decreasing the other travel times
to their minimal values. This gives a slightly different
system, for which a new eigenvalue can be calculated.
The relative and absolute limits for increasing the
different travelling times individually without violation of
the roundtrip time (i.e. \( \lambda > T \)) are presented in Table 2.

Table 2: Delay sensitivity of the different traveling
times.

<table>
<thead>
<tr>
<th>Traveling time with delay</th>
<th>( d_1 )</th>
<th>( d_2 )</th>
<th>( d_3 )</th>
<th>( d_4 )</th>
<th>( d_5 )</th>
<th>( d_6 )</th>
<th>( d_7 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Relative limit ( (% ))</td>
<td>440</td>
<td>18</td>
<td>28</td>
<td>10</td>
<td>10</td>
<td>27.5</td>
<td>19.3</td>
</tr>
<tr>
<td>Absolute limit ( (\text{min}))</td>
<td>17.6</td>
<td>11.5</td>
<td>7.8</td>
<td>3</td>
<td>3</td>
<td>7.7</td>
<td>11.6</td>
</tr>
</tbody>
</table>

Table 2 show the maximal value that a single travelling
time \( d_i \) can be increased, and still get the nominal
roundtrip time (given by the eigenvalue of the
modified matrix) by decreasing all the other travelling
times to their minimal values. For example if we
increase \( d_7 \) by 27.5\% which is equal to 7.7 minutes,
and reduce all the other travelling times to their
minimal values, we will still get the eigenvalue \( \lambda=60 \).

A limitation with the analysis is that it assumes a
permanent change in the delays, and results concerns
only steady state. It does not give information about
dynamic delay propagation, which is the theme of the
following section.
4.2 Dynamic Delay Propagation

The delay sensitivity analysis in Section 4.1 assumed that we had permanent changes in the travelling times. A more normal situation is that the delay only concerns one single travel time, which means that the corresponding max-plus system matrix becomes time varying, due to that the travel times \(d_i\) become time varying (indicated by an index \(k\)). This is so due to the meeting conditions, that is Equations (1) and (2), where varying (indicated by an index \(k\)). This is so due to the expanded to

\[
2) + d_2(k + 2) + d_3(k + 2) + d_5(k + 2) + d_6(k + 2) + d_7(k + 2) + d_8(k + 2) + d_9(k + 2)
\]

respectively. As indicated with iteration indices newer versions of travel times are needed in these equations. Speeding up can also only be done after the delay has appeared, which in our case means that after a delay in \(d_i(k)\) only the traveling times \(d_i(j)\) with \(j > i\), can be decreased in the same iteration \(k\). In the next iteration all the traveling times can be decreased.

In Table 3 it has been tested how long it takes for a delay of 10, 20 and 30 minutes respectively in a certain travel time, to disappear from the system.

Table 3: Times expressed in minutes that it takes for a delay in a certain travel time to disappear from the system.

<table>
<thead>
<tr>
<th>Travel Time</th>
<th>Delay 10 min</th>
<th>Delay 20 min</th>
<th>Delay 30 min</th>
</tr>
</thead>
<tbody>
<tr>
<td>(d_1)</td>
<td>89.2</td>
<td>182.4</td>
<td>301.3</td>
</tr>
<tr>
<td>(d_2)</td>
<td>88.3</td>
<td>182.4</td>
<td>300.4</td>
</tr>
<tr>
<td>(d_3)</td>
<td>93.2</td>
<td>182.4</td>
<td>300.4</td>
</tr>
<tr>
<td>(d_4)</td>
<td>91</td>
<td>185.1</td>
<td>303.1</td>
</tr>
<tr>
<td>(d_5)</td>
<td>91</td>
<td>185.1</td>
<td>303.1</td>
</tr>
<tr>
<td>(d_6)</td>
<td>93.2</td>
<td>182.4</td>
<td>300.4</td>
</tr>
<tr>
<td>(d_7)</td>
<td>68</td>
<td>184.2</td>
<td>305.3</td>
</tr>
<tr>
<td>(d_8)</td>
<td>89.2</td>
<td>182.4</td>
<td>301.3</td>
</tr>
</tbody>
</table>

The calculation of the disappearance of a delay can be done as follows. Let \(M_d\) denote a matrix with the nominal timetables, that is \(M_d = [v, v\otimes 7, v\otimes 7\otimes \mathbb{R}, ...]\), and \(M_d\) is a matrix with the delayed arrival and departure times at corresponding times. The part of the time tables that can be used for selecting the part of the time table that is affected by a delay using the logical expression \((M_d - M_d) > 0\). This means that the time instant of the last delay \(t_d\) can be found using

\[
t_d = \max[M_d((M_d - M_d) > 0) - M_d(i,j)],
\]

where \(i, j\) are the timetable indices when actual first delay take place. For example 88.3 in on second row second column in Table 3 means that if the single travelling time \(d_2\) is increased by 10 minutes, and the travelling times \(d_3, d_4, d_5, d_6, d_7\), and \(d_8\) are speeded up to their minimal values, then the time instant of the last deviation from the time table is 88.3 minutes after the delay.

4.3 Recovery Matrix

In Goverde (2007) max-plus linear systems are written in polynomial form,

\[
x(k) = A_0\otimes x(k)\otimes A\otimes x(k - 1)\otimes w(k)
\]

where \(A\) is defined as in Equation (4). \(A_0\) is the matrix describing the direct connections from \(x(k)\) to \(x(k)\), and \(w(k)\) is the nominal departure times in period \(k\). \(A_0\) is in this case given by all the direct travelling times \(d_i\), including all delayed states, such that

\[
A_0(m + 5, m) = d_i, \text{ for } m = (i - 1)5 + n, \text{ for all } n = 1, 2, ... 5, \text{ and for all } i = 2, 3, ... 8.
\]

All the other elements of \(A_0\) are \(\epsilon\), as there are no direct connections. The departure times are given by the eigenvector \(\mathbf{v}\) in Equation 5, and the period \(T\) according to \(w(k) = T\otimes \mathbb{R}\otimes \mathbf{v}\). The polynomial equation can be written using a single matrix \(A_p\), according

\[
x(k) = A_p\otimes x(k - 1)\otimes w(k)
\]

where \(A_p = A_0\otimes A\otimes (T\otimes 1)^{-1}\).

Definition: Consider the max-plus linear system in Equation (7). The entry \(r_{ij}\) of the recovery matrix \(R\) is defined as the maximum delay of \(x_i(m)\) such that \(x_i(k)\) is not delayed for any \(k > m\) (Goverde 2007). The following equation (Baccelli et al., 1992; Goverde 2007) defines the elements of the recovery matrix,

\[
r_{ij} = w_i - w_j - [A^+_p]_{ij}
\]

where the \(w_i\) and \(w_j\) are element of vector \(w\), \(A^+_p = \otimes A^+_p\), and the notation \([A^+_p]_{ij}\) refers to the \(ij\)th element of the matrix \(A^+_p\). If in the graph of \(A^+_p\) no path exists from node \(j\) to node \(i\) then \(r_{ij} = \infty\). The recovery matrix thus takes values from the extended set \(\mathbb{R}_\infty = \mathbb{R}_\max \cup \{\infty\}\).

In the studied train network between Helsinki and Turku, constructed from Table 1 presented in Figure 1, the recovery matrix \(R\) is of size \(40 \times 40\), with \(T = 60\). A \(20 \times 20\) submatrix of that matrix is given in Table 4.

According to Goverde, (2007), the \(ij\)th column of the recovery matrix \(R\) gives the recovery time from event \(j\) to all other events in the timetable and thus represents the impact a delay of event \(j\) has on future train events, and the \(ij\)th row of the recovery matrix \(R\) gives the recovery time from event \(i\) from all other events in the
timetable and thus represents the sensitivity of event $i$ on delays of preceding events. The diagonal elements of $R$ again represent recovery times to the event itself. In our example, most of our states are delayed versions of previous states. As can be noted in Table 4, not all diagonal elements representing the same departure at different times are the same. For example, $r_{16,16} = 12$, $r_{18,18} = 22.5$ and $r_{19,19} = 29.6$, although these elements all correspond to the event “arrival in Turku” at times $k$, $k-2$ and $k-3$ respectively. As $k$ is arbitrary, all these recovery elements should logically be the same. This is not so because the delayed versions are just memory variables, for which no other constraints than the back shifting according Equation 3 is present, and thus the recovery matrix is not correct for these. Thus in our example only every fifth row in the recovery matrix show true recovery times, and these are shown in Table 5.

The recovery matrix take in the consideration only one train not the whole system and it gives all the information for the delay of one train only. A 0 in the recovery matrix means a tight schedule, with no slack. For example the first row in the reduced recovery matrix is easy to interpret; the first value is 29.6, which is the total slack for a single train. After that the slack is reduced by the slack in corresponding travel time, up to the final value 0, which corresponds to that no slack is present in the 4 minute waiting time in Helsinki ($d_1$). All the other travelling times are assumed to have 10% slack. The other zero (row 11, columns 26) is due to a meeting condition (in Salo).

The results shown in Table 2 can also be calculated using recovery matrix calculations. In Table 2 it was assumed that we have a permanent delay in one travel time. The maximum tolerance for a permanent delay in one travel time can be obtained by increasing the corresponding travel time in the recovery matrix, until we start getting negative entries on the relevant diagonal elements in the recovery matrix (the ones indicated by green in Table 5).

Table 4: The upper left quadrant of the recovery matrix, with diagonal element shaded.

| 1     | 2     | 3     | 4     | 5     | 6     | 7     | 8     | 9     | 10    | 11    | 12    | 13    | 14    | 15    | 16    | 17    | 18    | 19    | 20    |
|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|
| 29.6  | 29.6  | 29.6  | 29.6  | 29.6  | 29.6  | 29.6  | 29.6  | 29.6  | 29.6  | 29.6  | 29.6  | 29.6  | 29.6  | 29.6  | 29.6  | 29.6  | 29.6  | 29.6  | 29.6  |
| 0     | 29.6  | 29.6  | 29.6  | 29.6  | 29.6  | 29.6  | 29.6  | 29.6  | 29.6  | 29.6  | 29.6  | 29.6  | 29.6  | 29.6  | 29.6  | 29.6  | 29.6  | 29.6  | 29.6  |
| 0     | 0     | 29.6  | 29.6  | 29.6  | 29.6  | 29.6  | 29.6  | 29.6  | 29.6  | 29.6  | 29.6  | 29.6  | 29.6  | 29.6  | 29.6  | 29.6  | 29.6  | 29.6  | 29.6  |
| 0     | 0     | 0     | 29.6  | 29.6  | 29.6  | 29.6  | 29.6  | 29.6  | 29.6  | 29.6  | 29.6  | 29.6  | 29.6  | 29.6  | 29.6  | 29.6  | 29.6  | 29.6  | 29.6  |
| 0     | 0     | 0     | 0     | 29.6  | 29.6  | 29.6  | 29.6  | 29.6  | 29.6  | 29.6  | 29.6  | 29.6  | 29.6  | 29.6  | 29.6  | 29.6  | 29.6  | 29.6  | 29.6  |
| 29.6  | 29.6  | 29.6  | 29.6  | 29.6  | 29.6  | 29.6  | 29.6  | 29.6  | 29.6  | 29.6  | 29.6  | 29.6  | 29.6  | 29.6  | 29.6  | 29.6  | 29.6  | 29.6  | 29.6  |

Table 5: The relevant parts of the recovery matrix. Diagonal elements highlighted by green, and recovery times related to a full cycle is highlighted with orange.

For example the first row in the reduced recovery matrix is easy to interpret; the first value is 29.6, which is the total slack for a single train. After that the slack is reduced by the slack in corresponding travel time, up to the final value 0, which corresponds to that no slack is present in the 4 minute waiting time in Helsinki ($d_1$). All the other travelling times are assumed to have 10% slack. The other zero (row 11, columns 26) is due to a meeting condition (in Salo).

The results shown in Table 2 can also be calculated using recovery matrix calculations. In Table 2 it was assumed that we have a permanent delay in one travel time. The maximum tolerance for a permanent delay in one travel time can be obtained by increasing the corresponding travel time in the recovery matrix, until we start getting negative entries on the relevant diagonal elements in the recovery matrix (the ones indicated by green in Table 5).

The results in Table 3 can only partially be calculated using recovery matrix calculations. In Table 3 certain temporary delays (10, 20 and 30 minutes) were considered. In Table 3 it can be seen that the time it
takes for the system to catch up after delays of 30 minutes, are all slightly more than 300 minutes. This is not a coincidence, in most cases it is the delayed train itself that uses most time to catch up, and the recovery time 29.6 in positions highlighted with orange, it means that if we have a delay which is larger than 29.6, it will take more than 300 minutes (i.e. a full cycle) for the system to catch up.

5. Conclusions

This paper described how a max-plus model for a train system can be constructed. Meeting conditions caused by having a single track, and other physical constrains, have been handled by extending the state space with delayed states, which has enabled rewriting the state update equation in the form \( x(k) = A \otimes x(k - 1) \). Static and dynamic delay sensitivity of the network has been analyzed by modifying the \( A \)-matrix, and using eigenvalue calculations. The such obtained results were compared to standard recovery matrix based calculations. A recovery matrix for the chosen extended state space becomes large, and contains even irrelevant information. Guidelines for finding and interpreting the relevant information from the recovery matrix have been discussed. Max-plus formalism was used throughout this paper.

References


