

# MODELLING AND PERFORMANCE EVALUATION OF DES — A MAX-PLUS ALGEBRA TOOLBOX FOR MATLAB

Jarosław Stańczyk

*Lehrstuhl für Systemtheorie technischer Prozesse, Otto-von-Guericke University Magdeburg  
Postfach 41 20, D-39016 Magdeburg, Germany*

Eckart Mayer

*Max Planck Institute Dynamics of Complex Technical Systems, Sandtorstr. 1, D-39106 Magdeburg, Germany*

Jörg Raisch

*Lehrstuhl für Systemtheorie technischer Prozesse, Otto-von-Guericke University Magdeburg  
Postfach 41 20, D-39016 Magdeburg, Germany  
and Max Planck Institute Dynamics of Complex Technical Systems, Sandtorstr. 1, D-39106 Magdeburg, Germany*

**Keywords:** Discrete event systems, max-plus algebra, performance evaluation.

**Abstract:** This paper discusses the usefulness of  $(\max, +)$  algebra as a mathematical modelling framework for discrete event systems (DESs). A Max-Plus Algebra Toolbox developed at Lehrstuhl für Systemtheorie technischer Prozesse is presented. This software package is a set of functions to take advantage of the  $(\max, +)$  algebra in the Matlab environment for rapid prototyping, design, and analysis of DESs. An overview of the modelling and analysis concepts of the  $(\max, +)$  algebra approach for DES is given. Application examples are provided in the final part of the paper to illustrate the potential of this approach and the toolbox.

## 1 INTRODUCTION

Many phenomena from manufacturing systems, telecommunication networks and transportation systems can be described as so-called discrete event systems (DES), or discrete event dynamic systems. A DES is a dynamic asynchronous system where the state transitions are initiated by events that occur at discrete instants of time. An event corresponds to the start or the end of an activity. A common property of such examples is that the start of an activity depends on termination of several other activities. Such systems cannot conveniently be described by differential or difference equations, and naturally exhibit a periodic behaviour.

An introduction to DES has been given, e.g. in (Cassandras and Lafortune, 1999). Many frameworks exist to study DES. Examples are queuing theory, e.g. (Gross and Harris, 1997), Petri nets, e.g. (Banaszak et al., 1991), the  $(\max, +)$  algebra (Baccelli et al., 1992) and many others. The most widely used technique to analyze DES is computer simulation. An important drawback of simulation is that it often does not give a real understanding of how parameter changes affect important system properties such

as stability, robustness and optimality of system performance. Analytical techniques can provide a much better insight in this respect. Therefore, formal methods are to be preferred as tools for modelling, analysis and control of DES.

This paper presents a software tool for rapid prototyping, design and analysis of DESs: a Max-Plus Algebra Toolbox for Matlab (Stańczyk, 2003). This is a set of functions implementing major aspects of the  $(\max, +)$  algebra in the Matlab environment.

The  $(\max, +)$  algebra was first introduced in (Cunningham-Green, 1979). A standard reference is (Baccelli et al., 1992), a brief survey of methods and applications of this algebra is given in (Cohen et al., 1999) and (Gaubert and Max-Plus, 1997). In certain aspects, the  $(\max, +)$  algebra is comparable to the conventional algebra. In the  $(\max, +)$  algebra the addition  $(+)$  and multiplication  $(\times)$  operators from the conventional algebra are replaced by the maximization  $(\max)$  and addition  $(+)$  operators, respectively. Using these operators, a linear description (in the  $(\max, +)$  algebra sense) of certain non-linear systems (in the conventional algebra) is achieved.

There are other tools available in the Internet for computation in  $(\max, +)$  algebra:

Stańczyk J., Mayer E. and Raisch J. (2004).

MODELLING AND PERFORMANCE EVALUATION OF DES — A MAX-PLUS ALGEBRA TOOLBOX FOR MATLAB.

In *Proceedings of the First International Conference on Informatics in Control, Automation and Robotics*, pages 270-275

Copyright © SciTePress

- *the MaxPlus Toolbox for Scilab* (INRIA Max-Plus Working Group, 1998);
- *MAX*: a Maple package (Gaubert, 1992).

This contribution is organized as follows. In Section 2, the  $(\max, +)$  algebra formalism and the  $(\max, +)$  state space description of DESs are introduced. The Max-Plus Algebra Toolbox for Matlab is shortly described in Section 3. Section 4 is devoted to illustrate the potential and abilities of the presented tool. In the last section, we summarize the paper and indicate some directions for future research.

## 2 MAX-PLUS ALGEBRA

In this section, we give an introduction to the  $(\max, +)$  algebra. Most of the material presented in this section is taken from (Baccelli et al., 1992) and (Cuningham-Green, 1979), where a complete overview of the  $(\max, +)$  algebra can be found. We also present the state space description of DES.

### 2.1 Max-plus algebra

$(\max, +)$  algebra is defined as follows:

- $\mathbb{R}_\varepsilon = \mathbb{R} \cup \{\varepsilon\}$ , where  $\mathbb{R}$  is the field of real numbers and  $\varepsilon = -\infty$ ;
- $\forall a, b \in \mathbb{R}_\varepsilon : a \oplus b = \max(a, b)$ ;
- $\forall a, b \in \mathbb{R}_\varepsilon : a \otimes b = a + b$ .

The algebraic structure  $\mathbb{R}_{\max} = (\mathbb{R}_\varepsilon, \oplus, \otimes, \varepsilon, e)$ ,  $e = 0$ , is called *the max-plus algebra*. More specifically, the algebraic structure  $\mathbb{R}_{\max}$  is an idempotent, commutative semiring (or dioid). This structure satisfies all the semiring axioms, i.e.:

- commutativity of operation  $\oplus$ :  
 $\forall a, b \in \mathbb{R}_\varepsilon : a \oplus b = b \oplus a$ ;
- associativity of operation  $\oplus$ :  
 $\forall a, b, c \in \mathbb{R}_\varepsilon : (a \oplus b) \oplus c = a \oplus (b \oplus c)$ ;
- existence of a neutral element ( $\varepsilon$ ) for operation  $\oplus$ :  
 $\forall a \in \mathbb{R}_\varepsilon : a \oplus \varepsilon = \varepsilon \oplus a = a$ ;
- associativity of operation  $\otimes$ :  
 $\forall a, b, c \in \mathbb{R}_\varepsilon : (a \otimes b) \otimes c = a \otimes (b \otimes c)$ ;
- existence of a neutral element ( $e$ ) for operation  $\otimes$ :  
 $\forall a \in \mathbb{R}_\varepsilon : e \otimes a = a \otimes e = a$ ;
- $\varepsilon$  is absorbing for  $\otimes$ :  
 $\forall a \in \mathbb{R}_\varepsilon : a \otimes \varepsilon = \varepsilon$ ;
- distributivity of  $\otimes$  with respect to  $\oplus$ :  
 $\forall a, b, c \in \mathbb{R}_\varepsilon : a \otimes (b \oplus c) = (a \otimes b) \oplus (a \otimes c)$ ,  
 $(b \oplus c) \otimes a = (b \otimes a) \oplus (c \otimes a)$ .

This semiring is:

- commutative with regard to  $\otimes$ :  
 $\forall a, b \in \mathbb{R}_\varepsilon : a \otimes b = b \otimes a$ ;
- idempotent:  
 $\forall a \in \mathbb{R}_\varepsilon : a \oplus a = a$ ;

Note that non-zero elements do not have an inverse for  $\oplus$ , because  $a \oplus x = \varepsilon$  does not have a solution for  $a \neq \varepsilon$ .

We will write  $ab$  for  $a \otimes b$  whenever there is no possible confusion. Now, we extend the max-plus algebra operations to vectors and matrices in the following way.

The  $(\max, +)$  sum of a scalar  $a \in \mathbb{R}_\varepsilon$  and a vector  $\mathbf{b} = (b_i) \in \mathbb{R}_\varepsilon^n$  is defined by:

$$(a \oplus \mathbf{b})_i = a \oplus b_i, \quad i = 1, \dots, n. \quad (1)$$

The sum  $\oplus$  of matrices  $\mathbf{A} = (a_{ij}), \mathbf{B} = (b_{ij}) \in \mathbb{R}_\varepsilon^{m \times n}$  is defined to be the  $m \times n$  matrix  $\mathbf{A} \oplus \mathbf{B}$  obtained by adding corresponding entries. That is,

$$(\mathbf{A} \oplus \mathbf{B})_{ij} = a_{ij} \oplus b_{ij}, \quad i = 1, \dots, m; j = 1, \dots, n. \quad (2)$$

We define the product of a scalar  $a \in \mathbb{R}_\varepsilon$  and a vector  $\mathbf{b} = (b_i) \in \mathbb{R}_\varepsilon^n$  by:

$$(a \otimes \mathbf{b})_i = a \otimes b_i, \quad i = 1, \dots, n. \quad (3)$$

The product  $\otimes$  of matrices  $\mathbf{A} = (a_{ik}) \in \mathbb{R}_\varepsilon^{m \times p}$  and  $\mathbf{B} = (b_{kj}) \in \mathbb{R}_\varepsilon^{p \times n}$  is defined to be the  $m \times n$  matrix whose  $(i, j)$ -entry is the inner product of the  $i^{\text{th}}$  row of  $\mathbf{A}$  with the  $j^{\text{th}}$  column in  $\mathbf{B}$ . That is,

$$(\mathbf{A} \otimes \mathbf{B})_{ij} = \bigoplus_{k=1}^p a_{ik} \otimes b_{kj} = \max_k (a_{ik} + b_{kj}), \quad i = 1, \dots, m; j = 1, \dots, n, \quad (4)$$

where:  $\bigoplus_{j=1}^m a_j$  is short-hand for  $a_1 \oplus \dots \oplus a_m$ .

The matrix  $\mathbf{I}_n = (e_{ij}) \in \mathbb{R}_\varepsilon^{n \times n}$  with  $e$ 's on the main diagonal and  $\varepsilon$ 's elsewhere is called the identity matrix of order  $n$ .

The matrix  $\mathbf{E} = (\varepsilon_{ij}) \in \mathbb{R}_\varepsilon^{m \times n}$  with  $\varepsilon_{ij} = \varepsilon$  for all  $i, j$ , is the zero-matrix.

The operator  $*$  for square matrices  $\mathbf{A} \in \mathbb{R}_\varepsilon^{n \times n}$  is defined by:

$$\mathbf{A}^* = \bigoplus_{k \in \mathbb{N}_0} \mathbf{A}^k, \quad (5)$$

where:  $\mathbf{A}^0 = \mathbf{I}_n$ ,  $\mathbf{A}^k = \mathbf{A} \otimes \mathbf{A}^{k-1}$  and  $\mathbb{N}_0$  is the set of nonnegative integers. (5) is only meaningful if the right hand side converges.

### 2.2 State space description of timed event graph

Timed event graphs are (timed) Petri-nets, which are convenient to model (timed) synchronization problems. They are characterized by the fact that every

place has exactly one predecessor transition and one successor transition. Time constraints are modelled by so-called holding times, representing the minimum amount of time a token has to “spend” in a place before it can contribute to enable a “downstream” transition.

Let  $x_i(k)$  denote the time instant, when an “internal” transition  $i$  can fire for the  $k^{\text{th}}$  time, and  $\mathbf{x}(k) = (x_i(k))$  the corresponding vector of firing times. Similarly, let  $u_i(k)$  denote the firing times of “input transitions” which can be triggered by the outside world, and  $y_i(k)$  the firing times of “output transitions” which carry information to the outside world. It is then straightforward, to read the following  $(\max, +)$  equations from the timed event graph:

$$\mathbf{x}(k+1) = \bigoplus_{i=0}^M \mathbf{A}_i \mathbf{x}(k+1-i) \oplus \bigoplus_{j=0}^N \mathbf{B}_j \mathbf{u}(k+1-j), \quad (6)$$

$$\mathbf{y}(k) = \bigoplus_{i=0}^M \mathbf{C}_i \mathbf{x}(k-i) \oplus \bigoplus_{j=0}^N \mathbf{D}_j \mathbf{u}(k-j). \quad (7)$$

If  $\mathbf{A}_0^*$  exists, by defining a state vector

$$\tilde{\mathbf{x}}(k) = [ \mathbf{x}'(k) \quad \mathbf{x}'(k-1) \quad \dots \quad \mathbf{x}'(k-M) ]', \quad (8)$$

an augmented input vector

$$\tilde{\mathbf{u}}(k) = [ \mathbf{u}'(k) \quad \mathbf{u}'(k-1) \quad \dots \quad \mathbf{u}'(k-N) ]', \quad (9)$$

and matrices

$$\tilde{\mathbf{A}} = \begin{bmatrix} \mathbf{A}_0^* \mathbf{A}_1 & \mathbf{A}_0^* \mathbf{A}_2 & \dots & \dots & \mathbf{A}_0^* \mathbf{A}_M \\ \mathbf{I} & \boldsymbol{\varepsilon} & \dots & \dots & \boldsymbol{\varepsilon} \\ \boldsymbol{\varepsilon} & & & & \\ \vdots & & & \ddots & \\ \boldsymbol{\varepsilon} & \dots & \boldsymbol{\varepsilon} & \mathbf{I} & \boldsymbol{\varepsilon} \end{bmatrix}, \quad (10)$$

$$\tilde{\mathbf{B}} = \begin{bmatrix} \mathbf{A}_0^* \mathbf{B}_0 & \dots & \mathbf{A}_0^* \mathbf{B}_N \\ \boldsymbol{\varepsilon} & \dots & \boldsymbol{\varepsilon} \\ \vdots & & \vdots \\ \boldsymbol{\varepsilon} & \dots & \boldsymbol{\varepsilon} \end{bmatrix}, \quad (11)$$

$$\tilde{\mathbf{C}} = [\mathbf{C}_0 \quad \dots \quad \mathbf{C}_M], \quad (12)$$

$$\tilde{\mathbf{D}} = [\mathbf{D}_0 \quad \dots \quad \mathbf{D}_N], \quad (13)$$

where  $\mathbf{I}$  and  $\boldsymbol{\varepsilon}$  are appropriately sized  $(\max, +)$ -algebraic identity and zero matrices, respectively, Eqns. (6), (7) can be written as

$$\tilde{\mathbf{x}}(k+1) = \tilde{\mathbf{A}} \tilde{\mathbf{x}}(k) \oplus \tilde{\mathbf{B}} \tilde{\mathbf{u}}(k+1), \quad (14)$$

$$\tilde{\mathbf{y}}(k) = \tilde{\mathbf{C}} \tilde{\mathbf{x}}(k) \oplus \tilde{\mathbf{D}} \tilde{\mathbf{u}}(k). \quad (15)$$

### 3 THE TOOLBOX

The Max-Plus Algebra Toolbox is a set of functions, written as M-files, to take advantage of the  $(\max, +)$  algebra in the Matlab environment. This package has been developed for fast and easy modelling and analysis of DESs amenable to  $(\max, +)$  algebra. Basic elements of this toolbox are:

- elementary functions: the  $(\max, +)$  addition for scalars, vectors and matrices, multiplication, rise to a power, etc.;
- solving an equation of type  $\mathbf{x} = \mathbf{A}\mathbf{x} \oplus \mathbf{b}$ ;
- solving an equation of type  $\mathbf{A}\mathbf{x} = \mathbf{b}$ ;
- solving the spectral problem  $\mathbf{A}\mathbf{x} = \lambda\mathbf{x}$ , i.e. determining the eigenvalue and eigenvectors of  $\mathbf{A}$ ;
- DESs analysis:
  - graphically by providing Gantt charts (see Fig. 1),
  - calculate performance indices (flow time, resource utilizations, etc.),
  - calculate cycle time and length of transient state.

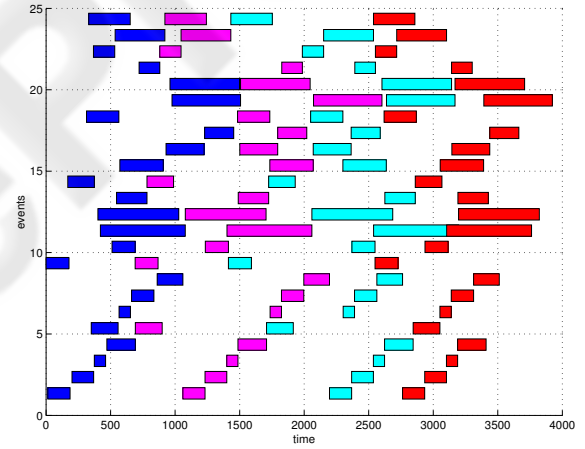


Figure 1: An exemplary Gantt chart generated by the `mp_gantt()` function.

### 4 APPLICATION

In order to illustrate our software package let us examine two examples: a simple manufacturing system and a transportation system, namely the suburban train network of Stuttgart, Germany.

## 4.1 Multi-product manufacturing system

The idea for this example has been taken from (Baccelli et al., 1992). Consider a manufacturing system that consists of three machines ( $M_1$ ,  $M_2$  and  $M_3$ ). In this manufacturing system three different types of parts ( $P_1$ ,  $P_2$  and  $P_3$ ) are produced according to a certain product mix. The routes followed by the various types of parts are depicted in Fig. 2.

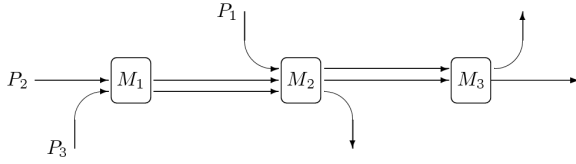


Figure 2: The routing of the various types of parts along the machines.

Parts of type  $P_1$  first visit machine  $M_2$  and then go to  $M_3$ . Parts of type  $P_2$  enter the system via machine  $M_1$ , then they go to machine  $M_2$  and finally leave the system through machine  $M_3$ . Parts of type  $P_3$  first visit machine  $M_1$  and then go to  $M_2$ . It is assumed that:

- Parts are carried around on pallets. There is one pallet available for each type of part.
- It is assumed that the transportation times are negligible and that there are no set-up times on the machines when they switch from one part type to another.
- The sequencing of the various parts on the machines is known: on machine  $M_1$  it is  $(P_2, P_3)$ , i.e. the machine first processes a part of type  $P_2$  and then a part of type  $P_3$ , on machine  $M_2$  the sequence is  $(P_1, P_2, P_3)$ , and  $(P_1, P_2)$  on machine  $M_3$ . We will call these sequences *local dispatching rules* and we will describe them as  $\sigma$  (i.e.  $\sigma_1$  for the sequence on  $M_1$ ,  $\sigma_2$  for the sequence on  $M_2$ , and  $\sigma_3$  for  $M_3$ ).

The information about the sequencing and the duration of the various activities (processing times) is shown in Fig. 3. In this figure, the activities are represented by ordered pairs of the form  $(P_i, M_j)$  meaning that a part of type  $P_i$  is processed on machine  $M_j$ . The arcs represent the precedence constraints between activities. At the bottom right of each activity we have indicated its duration, e.g.  $(P_1, M_2)$  (activity 3) has duration  $d_3 = 3$ .

In order to simplify the process of deriving the evolution equations of this system, we shall first look at what happens in one cycle of the production process. We define:

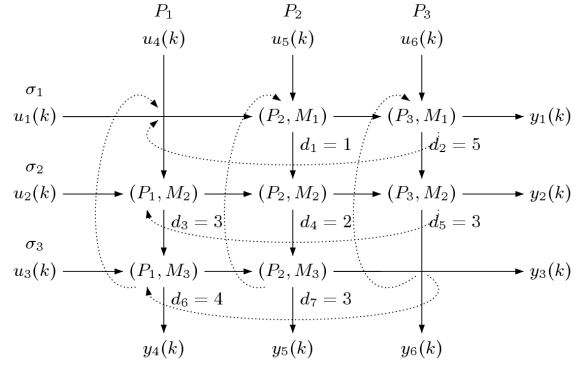


Figure 3: The sequence and the duration of the various activities.

- $u_i(k)$  — time instant at which machine  $M_i$  is available for the first activity that should be performed on it in the  $k^{th}$  production cycle for  $i = 1, 2, 3$ ;
- $u_j(k)$  — time instant at which the raw material for a part of type  $P_{j-3}$  is available in the  $k^{th}$  production cycle for  $j = 4, 5, 6$ ;
- $x_i(k)$  — time instant at which activity  $i$  starts in the  $k^{th}$  production cycle for  $i = 1, 2, \dots, 7$ ;
- $y_i(k)$  — time instant at which machine  $M_i$  has finished processing the last part of the  $k^{th}$  production cycle that should be processed on it for  $i = 1, 2, 3$ .
- $y_j(k)$  — time instant at which the finished product of type  $P_{j-3}$  of the  $k^{th}$  production cycle has been completed for  $j = 4, 5, 6$ ;

We have the following evolution equations:

$$\begin{aligned} x_1(k+1) &= 5x_2(k) \oplus 3x_7(k) \\ &\quad \oplus u_1(k+1) \oplus u_5(k+1), \\ x_2(k+1) &= 1x_1(k+1) \oplus 3x_5(k) \\ &\quad \oplus u_6(k+1), \\ &\vdots \end{aligned} \quad (16)$$

or, more compactly:

$$\begin{aligned} \mathbf{x}(k+1) &= \mathbf{A}_0 \mathbf{x}(k+1) \oplus \mathbf{A}_1 \mathbf{x}(k) \\ &\quad \oplus \mathbf{B}_0 \mathbf{u}(k+1), \\ &= \mathbf{A} \mathbf{x}(k) \oplus \mathbf{B} \mathbf{u}(k+1), \end{aligned} \quad (17)$$

where  $\mathbf{A} = \mathbf{A}_0^* \mathbf{A}_1$ , and  $\mathbf{B} = \mathbf{A}_0^* \mathbf{B}_0$ .

An exemplary analysis of the considered system based on the Max-Plus Algebra Toolbox is shown below. It provides cycle time and consecutive states  $\mathbf{x}(k)$  of the system. It is assumed that  $\mathbf{x}(0) = [\varepsilon]^7 \times 1$ , i.e. all machines are idle in the beginning, and  $\mathbf{u}(1) = [e]^{6 \times 1}$ , i.e. all machines can be started without delay.

```

% definitions of matrices:
A0 = [
  -inf -inf -inf -inf -inf -inf -inf
    1 -inf -inf -inf -inf -inf -inf
  -inf -inf -inf -inf -inf -inf -inf
    1 -inf 3 -inf -inf -inf -inf
  -inf 5 -inf 2 -inf -inf -inf
  -inf -inf 3 -inf -inf -inf -inf
  -inf -inf -inf 2 -inf 4 -inf ]

A1 = [
  -inf 5 -inf -inf -inf -inf 3
  -inf -inf -inf -inf 3 -inf -inf
  -inf -inf -inf -inf 3 4 -inf
  -inf -inf -inf -inf -inf -inf -inf
  -inf -inf -inf -inf -inf -inf -inf
  -inf -inf -inf -inf -inf -inf 3
  -inf -inf -inf -inf -inf -inf -inf ]

B0 = [ 0 -inf -inf -inf 0 -inf
  -inf -inf -inf -inf -inf 0
  -inf 0 -inf 0 -inf -inf
  -inf -inf -inf -inf -inf -inf
  -inf -inf -inf -inf -inf -inf
  -inf -inf 0 -inf -inf -inf
  -inf -inf -inf -inf -inf -inf ]

% create the A matrix
>> A = mp_multi(mp_star(A0),A1);
% calculate a cycle time
>> lambda = mp_mcm(A)
lambda =
    9.5000

% create the B matrix
>> B = mp_multi(mp_star(A0),B0);

% determine initial conditions
>> x = mp_zeros(7,1);
>> u = mp_ones(6,1);

% calculate a sequence of a state vector
>> X(:,1) =
  mp_sum(mp_multi(A,x),mp_multi(B,u));
>> for i=2:6
X(:,i) = mp_sum(mp_multi(A,X(:,i-1)),
  mp_multi(B,u));
end
>> X
X =
    0    10    19    29    38    48
    1    11    20    30    39    49
    0     9    19    28    38    47
    3    12    22    31    41    50
    6    16    25    35    44    54
    3    12    22    31    41    50
    7    16    26    35    45    54

```

## 4.2 Transportation system

Max-plus algebra has been intensely used in modelling and analysis of train traffic networks, i.e.

(Braker, 1993). As an application example, a model for the suburban train network of Stuttgart, Germany, is briefly described in this section. The structure of the network is pictured in Fig.4. As short followup times of trains and single line track segments (used in both directions) are critical features of this transportation system, a specifically detailed max-plus model has been used for analysis and simulation. In addition to passenger changeover times also safety constraints for track segments used consecutively by several trains are taken into account in order to get an accurate model.

Generally, the max-plus state space model structure as given in (14)–(15) is used. The state vector is composed of the event times for the following types of events:

- departure of trains at terminal stations
- arrival resp. departure of trains at changeover stations
- trains entering track segments shared with other lines
- trains entering resp. leaving single track segments

The dependencies between the events (entries in matrix  $\mathbf{A}_0$  resp.  $\mathbf{A}_1$ ) are

- minimum travelling times of trains including stopover at intermediate stations and minimum turnaround times at endpoints of lines
- minimum times passengers need for changeover between trains
- minimum followup time distance for consecutive trains on the same track
- safety constraints for single track segments shared by trains moving in opposite directions

In addition to the standard max-plus model (14), a timetable is needed for cyclically operated transportation systems. Timetable information consists of two parts: the cycle time and the timetable vector  $\mathbf{w}_0$ . For the current example, cycle time is  $T = 30$ : under normal conditions, operation is repeated every 30 minutes. The location of trains at any time  $t$  is identical to the location at time  $t - 30$  with the only difference that all trains have moved forward by one position on their line. The timetable vector  $\mathbf{w}_0$  provides the earliest time instant at which any event is allowed to happen. The  $k^{th}$  occurrence of any event is not allowed to happen before the time given by the timetable:

$$\mathbf{x}(k) \geq \mathbf{w}(k) = \mathbf{w}_0 + k \cdot T. \quad (18)$$

Thus, the timetable enters the state space model as an input  $\mathbf{u}$  with  $\mathbf{B} = \mathbf{I}$  and  $\mathbf{u}(k) = \mathbf{w}(k) = \mathbf{w}_0 \otimes T^k$ .

For the suburban train network of Stuttgart, an overall number of 73 events is necessary. The system matrix  $\mathbf{A}$  is determined from  $\mathbf{A}_0$  and  $\mathbf{A}_1$  as given in

Eqn. (17). Within  $\mathbf{A}$ , the number of non- $\varepsilon$  entries, i.e. the number of timed dependencies between the events is 426.

Hence, the overall max-plus algebra model for the cyclically operated train track network is:

$$\begin{aligned} \mathbf{x}(k+1) &= \mathbf{A}\mathbf{x}(k) \oplus \mathbf{w}(k+1) \\ \mathbf{x}(0) &= \mathbf{x}_0 \end{aligned} \quad (19)$$

with:  $\mathbf{w}(k) = \mathbf{w}_0 T^k$ ,  $\mathbf{x}(k) \in \mathbb{R}_\varepsilon^{73}$ ,  $\mathbf{A} \in \mathbb{R}_\varepsilon^{73 \times 73}$ .

The max-plus model (19) is used to analyze the system. The eigenvalue for the system matrix  $\mathbf{A}$  can be calculated using the `mp_egv1()` command of the max plus toolbox:  $\lambda = 26$ . The eigenvalue represents the minimum possible cycle time of the system. Thus, with the number of trains and the times as given, the system could be operated under a cycle time of 26 minutes. Comparing the timetable cycle time of  $T = 30$  minutes and the eigenvalue, we find a stability margin of 4 minutes per cycle, which allows for diminishing of delays.

A brief example illustrates simulation of the max-plus model. In simulation, the time instants for the events of the system, i.e.  $\mathbf{x}(1), \mathbf{x}(2), \dots$  are calculated based on the vector of initial times  $\mathbf{x}(0) = \mathbf{x}_0$ . For normal operation, the vector of initial times matches the timetable vector:  $\mathbf{x}(0) = \mathbf{w}_0$ . In order to investigate the effects of a traffic disruption, the system can instead be simulated starting from an initial vector where one or more time values are increased compared to the timetable vector. In our example, the value for the event time of a train leaving Herrenberg station is increased by 7 time units. The state sequence  $\mathbf{x}(1), \mathbf{x}(2), \dots$  can now be calculated iteratively. The deviation from the timetable,  $\mathbf{z}(k) = \mathbf{x}(k) - \mathbf{w}(k)$ , can then be determined. As a measure for the ability of the system to cope with this type of disruption, one is especially interested in the number of cycles that are necessary for the delays to vanish entirely. For the given example, this is the case after 3 cycles, i.e.  $\mathbf{x}(k) = \mathbf{w}(k), k \geq 3$ . Fig.4 shows qualitatively the effects of the disruption spreading over the network during the first 3 cycles. The affected stops are shown in gray.

## 5 CONCLUDING REMARKS

In this paper, a software package for modelling and performance evaluation of DESs in the  $(\max, +)$  algebra has been presented. The implementation at this moment is more than tentative, so every remark and suggestion is welcome. The tool is available via the Internet. A brief manual, in which all functions and algorithms used in the toolbox are described is also available. The tool is still under development and additional functions will be added.

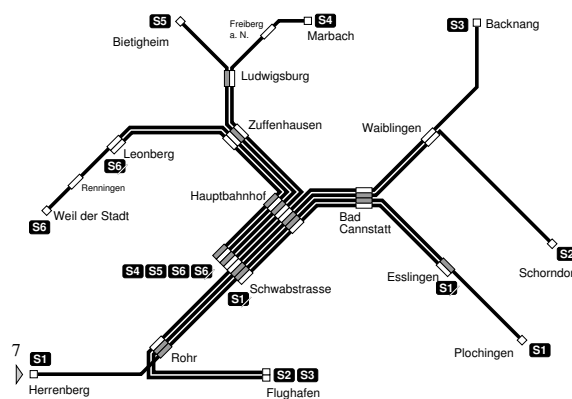


Figure 4: Railway network map with delays.

## REFERENCES

- Baccelli, F., Cohen, G., Olsder, G., and Quadrat, J.-P. (1992). *Synchronisation and Linearity: An Algebra for Discrete Event Systems*. John Wiley & Sons, Inc., Chichester, England.
- Banaszak, Z., Jampolski, L., Hasegawa, K., Krogh, B., Takahasi, K., and Borusan, A. (1991). *Modelling and Control of FMS. Petri Net Approach*. Wroclaw University of Technology, Wroclaw, Poland.
- Braker, J. G. (1993). *Algorithms and Applications in Timed Discrete Event Systems*. PhD thesis, Department of Technical Mathematics and Informatics, Delft University of Technology, Delft.
- Cassandras, C. and Lafortune, S. (1999). *Introduction to Discrete Event Systems*. Kluwer Academic Publishers, Boston, USA.
- Cohen, G., Gaubert, S., and Quadrat, J.-P. (1999). Max-plus algebra and system theory: where we are and where to go now. *Annual Reviews in Control*, 23(1):207–219.
- Cunningham-Green, R. (1979). *Minimax Algebra*, volume 166 of *Lecture Notes in Economics and Mathematical Systems*. Springer Verlag, Berlin, Germany.
- Gaubert, S. (1992). *MAX: a Maple package for the  $(\max, +)$  algebra*. Available in the Internet: <http://amadeus.inria.fr/gaubert/>.
- Gaubert, S. and Max-Plus (1997). Methods and applications of  $(\max, +)$  linear algebra. Technical Report 3088, INRIA, Rocquencourt, France.
- Gross, D. and Harris, C. (1997). *Fundamentals of Queueing Theory*. John Wiley & Sons, Inc., 3 edition.
- INRIA MaxPlus Working Group (1998). *MaxPlus Toolbox for Scilab*. Available in the Internet: <http://www.maxplus.org>.
- Stańczyk, J. (2003). *Max-Plus Algebra Toolbox for Matlab*. Available in the Internet: <http://ifatwww.et.uni-magdeburg.de/>.