IFAC

# Max-Plus Algebraic Modeling and Control of High-Throughput Screening Systems

**T. Brunsch** * **J. Raisch** *,**

\* *Fachgebiet Regelungssysteme, Technische Universität Berlin, 10587 Berlin, Germany (e-mail: brunsch@control.tu-berlin.de)*
\*\* *Fachgruppe System- und Regelungstheorie, Max-Planck-Institut für Dynamik komplexer technischer Systeme, 39106 Magdeburg, Germany*

**Abstract:** In this paper, we present a max-plus algebraic model for cyclically operated high-throughput screening plants. A max-plus algebraic representation of the system, derived directly from a discrete-event systems model of the predetermined globally optimal solution, contains negative order arcs, forcing certain events in previous cycles to occur after events in the current cycle. With respect to the cycle index, though of course not in terms of time, the model is acausal. However, the model can be transformed into a system representation without negative order arcs. The obtained max-plus algebraic model can then be applied as a controller to handle unexpected deviations from the predetermined cyclic operation during runtime.

*Keywords:* Cyclic systems, discrete-event systems, max-plus algebra, high-throughput screening

## 1. INTRODUCTION

Until the early 90's the search for new pharmaceutical ingredients was performed manually. This was an extremely time-consuming procedure lasting for months or even years. Through advances in robotics and high-speed computer technology it was possible to develop systems that are able to automatically screen thousands of substances in a very short time. The procedure of automatically analyzing biochemical compounds is called *high-throughput screening* (HTS). Nowadays HTS systems play an important role in the pharmaceutical industries but they are also relevant to other fields of biology and chemistry.

A batch subsumes all worksteps that are necessary to analyze one set of substances. The set of substances is aggregated on one microplate. Additional microplates may be included in the batch to convey reagents or waste material. The plates are automatically moved between the resources of the HTS system, which include readers, incubators, and pipettors. To be able to compare many different batches of an experiment, each batch has to follow an identical pattern within the system, in terms of timing as well as ordering of resources. Thus, the system has to be operated cyclically. The aim of maximizing the throughput of the system results in an optimal scheduling problem.

A method to determine the globally optimal schedules for cyclic systems, such as HTS systems, has been introduced by Mayer and Raisch (2004). This approach is based on *discrete-event systems* modeling, i.e., the system is characterized by the occurrence of discrete changes or events. More specifically, the model is given as a time window precedence network. Using standard graph reduction methods, the complexity of this network can then be reduced. The procedure ensures that the globally optimal

solution of the scheduling system is not cut off. Another important step in the proposed method is the transformation of the resulting mixed integer non-linear program (MINLP) into a mixed integer linear program (MILP). Although these steps decrease the complexity of the system significantly, the scheduling problem is still too complex to be performed online. Therefore, the algorithm is carried out offline before the execution of the HTS systems starts, i.e., it determines a static schedule. Static schedules, however, do not perform well when deviations from the predetermined cyclic scheme occur during runtime.

To handle such deviations, we propose a supervisory control scheme using a *max-plus algebraic* model of the HTS system. The model is based on the specific operation the user wants to run as well as on the globally optimal cyclic schedule determined offline. In case of a deviation from the cyclic scheme, the supervisor generates possible actions to be taken, i.e., the controller updates the schedule of the HTS plant and thus ensures continuous operation.

This paper is structured as follows. Section 2 gives a brief overview of the basic ideas of graph theory and max-plus algebra. The different constraints for high-throughput screening systems are explained in Section 3. It is described how the constraints are merged into a max-plus algebraic model of the HTS operation. In Section 4, a max-plus algebraic control scheme introduced for cyclic systems by Li et al. (2007) is adapted for HTS systems. Conclusions and suggestions for future work are given in Section 5.

## 2. GRAPH THEORY AND MAX-PLUS ALGEBRA

### 2.1 Fundamentals of Graph Theory

A directed graph (or digraph) is a pair $(\mathcal{V}, \mathcal{E})$ where $\mathcal{V}$ is the set of nodes or vertices, and $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ is a set

of ordered pairs of nodes, called edges or arcs. A weighted graph is a digraph with a real number (the weight) $w_{ji} \in \mathbb{R}$ assigned to each arc $(v_i, v_j) \in \mathcal{E}$. It can be represented by a precedence matrix $W \in \mathbb{R}_{max}^{n \times n}$, with $\mathbb{R}_{max} = \mathbb{R} \cup \{-\infty\}$ and $n$ being the total number of nodes in the graph. The entries of the matrix $W$ represent the weights of arcs. If no arc exists from node $v_i$ to node $v_j$ a weight of $-\infty$ is assigned to $w_{ji}$. If the weights $w_{ji} \in \mathbb{R}_{max}$ represent times, the corresponding weighted digraph will also be referred to as a time window precedence network. Then, nodes represent events and arcs represent minimum time offsets between the occurrence of events.

## 2.2 Max-Plus Algebra

Max-plus algebra (e.g., Baccelli et al. (2001), Heidergott et al. (2006)) is a powerful tool for the analysis and simulation of a certain class of discrete-event systems and provides a compact representation of such systems. It consists of two operations, $\oplus$ and $\otimes$ on the set $\mathbb{R}_{max} = \mathbb{R} \cup \{-\infty\}$. The operations are defined by: $\forall a, b \in \mathbb{R}_{max}$:

$$a \oplus b = \max(a, b)$$
$$a \otimes b = a + b.$$

The operation $\oplus$ is called *addition* of the max-plus algebra, the operation $\otimes$ is called *multiplication* of the max-plus algebra. The neutral element of max-plus addition is $-\infty$, also denoted as $\varepsilon$. The neutral element of multiplication is $0$, also denoted as $e$.

For matrices $A \in \mathbb{R}_{max}^{n \times l}$ and $B \in \mathbb{R}_{max}^{l \times m}$, the matrix product $A \otimes B$ is defined by

$$[A \otimes B]_{ji} = \bigoplus_{k=1}^{l} ([A]_{jk} \otimes [B]_{ki}) = \max_{k=1,\ldots,l} \{[A]_{jk} + [B]_{ki}\}.$$

Systems without repetition of events in max-plus algebra are represented by:

$$x = A_0 \otimes x \oplus B \otimes u,$$
$$y = C \otimes x,$$

where the vectors $u$ and $y$ contain the earliest time instants for the occurrence of certain input and output events of the system. The elements of matrix $A_0$ represent the minimum time offsets between events. If the corresponding graph does not contain any circuits, the matrix $A_0$ is said to be acyclic. In this case the matrix $A_0^* = I \oplus A_0 \oplus A_0^2 \oplus \ldots$ can be determined as $A_0^* = I \oplus A_0 \oplus A_0^2 \oplus \cdots \oplus A_0^{n-1}$, where $I$ is the identity matrix with respect to max-plus algebra. For acyclic system matrices $A_0$, the implicit representation of the system can be rewritten in an explicit form:

$$x = A_0^* \otimes B \otimes u,$$
$$y = C \otimes x.$$

For cyclically repeated systems, the max-plus model has to be extended such that dependencies of events belonging to different cycles can be included. For systems that are causal with respect to the cycle index, an event of cycle $k$ can only depend on events of the same cycle or previous cycles. Thus, the recurrence relation for such cyclic systems can formally be written

$$x(k) = \bigoplus_q (A_q \otimes x(k-q)) \oplus B \otimes u(k)$$
$$y(k) = C \otimes x(k),$$

with $k \in \mathbb{Z}$ and $q \in \{\mathbb{N} \cup 0\}$. This implicit recurrence relation can be rewritten in its explicit form if the matrix $A_0$ is acyclic:

$$x(k) = \bigoplus_q (A_0^* A_q \otimes x(k-q)) \oplus A_0^* B \otimes u(k)$$
$$y(k) = C \otimes x(k),$$

with $k \in \mathbb{Z}$ and $q \in \mathbb{N}$.

## 3. MAX-PLUS MODEL OF HTS SYSTEMS

A HTS plant is assumed to consist of $m$ resources. According to the operation the user wants to run, the sequence of activities for a single batch is given. It consists of $i_{max}$ activities and each activity $i$ is assigned to one of the resources, denoted by $J_i \in \{1, \ldots, m\}$, where it is executed. During the execution of activity $i$ the respective resource $J_i$ is said to be occupied. Different activities of a batch may overlap in time. Thus, a batch may occupy two resources at the same time, e.g., during the transfer from one resource to another one. However, we assume all resources to have capacity one, i.e., no batch can allocate a resource while this resource is occupied by another batch.

One possibility to model temporal dependencies between events is through a time window precedence network. To model HTS systems as a time window precedence network three different events have to be considered: start events $o_i$ denoting the start of activity $i$, release events $r_i$ referring to the end of activity $i$, and transfer events which model the transfer of a batch between two resources. A transfer event always occurs simultaneously with a corresponding transfer event associated with another resource. In a corresponding graph, events are represented by nodes which are connected by weighted arcs. The arcs represent the temporal interdependencies. A label or weight assigned to an arc denotes the minimum time which has to elapse between the events connected by the arc.

In general, dependencies between activities belonging to a single batch are called conjunctive constraints, while dependencies between activities on each single resource are referred to as disjunctive constraints.

### 3.1 Conjunctive Constraints

As mentioned before, the single batch scheme is defined by the operation the user wants to run. The single batch time scheme includes specifications on the sequence of activities of a single batch as well as information on the minimal time needed for or between activities.

To illustrate the modeling of HTS systems, we will introduce a simple example. The considered process of a single batch consists of four activities executed on a total of three resources. The specific operation defined by the user is given as a time window precedence network consisting of start events, release events and transfer events (Fig. 1). For graphical representation the system can also be illustrated as a Gantt chart (Fig. 2), in which it can easily be seen
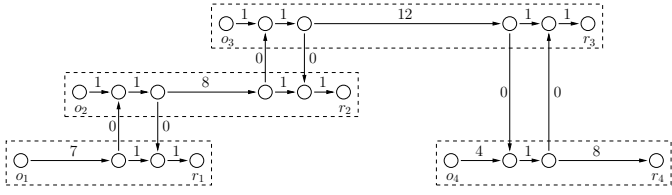
Fig. 1. Time window precedence network of single batch time scheme.

that resource 1 is revisited by each batch, in particular, the activities $i = 1$ and $i = 4$ of each batch are executed on resource 1. The time window precedence network of the
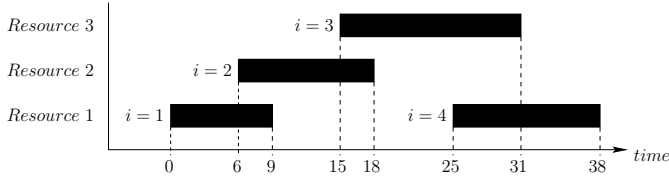


Fig. 2. Gantt chart of a single batch example.

system specification is quite large, however, it is possible to reduce the complexity of the model, i.e., the number of nodes, by introducing transfer times, omitting redundant arcs and merging events which occur concurrently.

The resulting graph, shown in Fig. 3, is referred to as the conjunctive graph and denoted by $\mathcal{G}_c$. As can be seen in
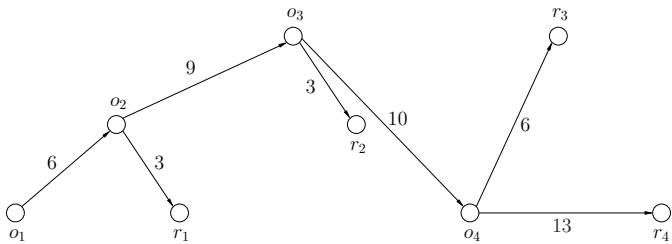


Fig. 3. Conjunctive graph $\mathcal{G}_c$.

Fig. 3, the reduced graph consists only of the start and release events of every activity. The precedence matrix corresponding to $\mathcal{G}_c$ is denoted by $A_c$. Furthermore, the conjunctive graph corresponds to the (max-plus) algebraic system:

$$x(k) = A_c \otimes x(k), \qquad (1)$$

where the vector $x(k)$ contains the earliest possible times for the $k$-th occurrence of each activity's start and release events. In our example, the vector is defined by $x = [o_1, r_1, o_2, r_2, o_3, r_3, o_4, r_4]^T$. By definition, the conjunctive graph does not contain any circuits and thus the matrix $A_c$ is acyclic.

### 3.2 Disjunctive Constraints

The disjunctive constraints describe the sequences of activities on each resource. They can be modeled as disjunctive graphs $\mathcal{G}_{d_\kappa}, \kappa = 1 \ldots, m$, where $m$ represents the total number of resources in the system. Disjunctive constraints may contain dependencies between activities which do not necessarily have to belong to the same batch. To denote dependencies between events of different batches, marked

arcs are added to the conventional notation of precedence graphs.

An arc marked with a single "/" is said to be of first order. A first order arc connecting event $i$ with event $j$ denotes the dependencies between the two events belonging to subsequent batches. More precisely, a first order arc with weight $w_{ji}$ is to be interpreted as $x_j(k) \geq x_i(k-1) + w_{ji}$. Higher order arcs may also be necessary to represent the sequence of activities. Additionally, since some activities of a batch may be executed prior to some activities of a previous batch, arcs of negative order may be necessary. The symbols for marked arcs are given in Table 1. Formally, any arc of order $q$, connecting node $i$ with $j$, can

Table 1. Notation of marked arcs.

| MARKING | ORDER OF ARC | ALGEBRAIC EXPRESSION |
|---------|--------------|----------------------|
| none | 0 | $x_j(k) \geq x_i(k) + w_{ji}$ |
| / | 1 | $x_j(k) \geq x_i(k-1) + w_{ji}$ |
| // | 2 | $x_j(k) \geq x_i(k-2) + w_{ji}$ |
| ⋮ | ⋮ | ⋮ |
| ∘ | −1 | $x_j(k) \geq x_i(k+1) + w_{ji}$ |
| ∞ | −2 | $x_j(k) \geq x_i(k+2) + w_{ji}$ |
| ⋮ | ⋮ | ⋮ |

be described by $(j, i)^{(q)}, q \in \mathbb{Z}$. The algebraic expression of this arc results in $x_j(k) \geq x_i(k-q) + w_{ji}$ with $q, k \in \mathbb{Z}$.

Using marked arcs, it is possible to model the sequence and timing of activities on each resource (Geyer (2004)). The sequence and timing are determined by the globally optimal solution, obtained with the offline algorithm developed by Mayer and Raisch (2004). Figure 4 shows the optimal sequence on the first resource for our example. It
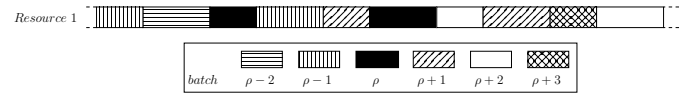


Fig. 4. Gantt chart of activities executed on resource 1.

can be seen that the first activity of batch $\rho$ is followed by the fourth activity of batch $\rho - 1$ and this activity is then followed by the first activity of batch $\rho + 1$. Thus the disjunctive graph for resource one consists of an arc of order $q = -1$, an arc of order $q = 2$, and two zero order arcs. The resulting disjunctive graph $\mathcal{G}_{d_1}$ is shown in Figure 5. Note that in disjunctive graphs only start and
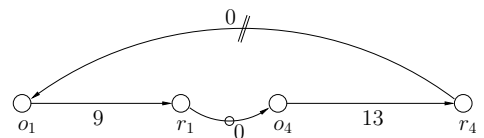


Fig. 5. Disjunctive graph $\mathcal{G}_{d_1}$

release events of activities are considered. Transfer events are not relevant for disjunctive constraints, i.e., transfer events are only necessary to model dependencies between activities executed on different resources.

Similarly, the disjunctive graphs for resource two and three can be determined. Since only one activity of a batch is executed on each of these resources their corresponding disjunctive graphs contain only one first order arc and one

arc of zeroth order between start and release event of the activity.

After determining the disjunctive graphs they can be written as a max-plus algebraic expression. Considering the different orders of arcs, each disjunctive graph $\mathcal{G}_{d_\kappa}, \kappa = 1, \ldots, m$, can be represented by

$$x(k) = \bigoplus_q \left( A_{d_\kappa(q)} \otimes x(k-q) \right), \ q, k \in \mathbb{Z}, \qquad (2)$$

where $x$ is the vector of the start and release events of all activities and $q$ denotes the order of arcs. Thus, the matrix $A_{d_\kappa(q)}$ encodes all arcs of $q$-th order within the disjunctive graph of resource $\kappa$.

### 3.3 Complete Model

The overall HTS system operated in a predetermined time-optimal cyclic way can then be described by merging the conjunctive and all disjunctive graphs into one extended precedence graph $\mathcal{G}_{sys}$. The extended precedence graph for our HTS example is shown in Figure 6. In max-plus
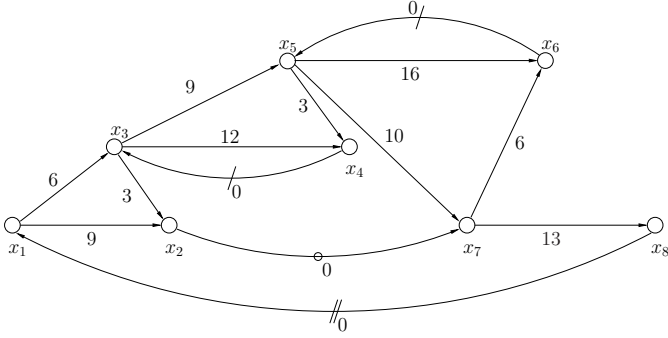


Fig. 6. Extended precedence graph $\mathcal{G}_{sys}$

algebra, this can then be written as

$$x(k) = \underbrace{\left( A_c \oplus \left( \bigoplus_{\kappa=1}^m A_{d_\kappa(0)} \right) \right)}_{A_{(0)}} x(k)$$
$$\oplus \left( \bigoplus_{q \neq 0} \underbrace{\left( \bigoplus_{\kappa=1}^m A_{d_\kappa(q)} \right)}_{A_{(q)}} x(k-q) \right) \oplus B \otimes u(k), \qquad (3)$$

with $q, k \in \mathbb{Z}$. The matrix $B$ is assumed to be the identity matrix $I$ and the input vector $u$ provides the earliest possible times for the input events.

In general, Equation (3) contains disjunctive dependencies of negative orders, i.e., terms with $q < 0$. Then, (3) is acausal with respect to the cycle index $k$ (though, of course, not with respect to absolute time). To facilitate further analysis, (3) can be reformulated as a causal system specification, i.e., without negative order terms. This can be done using the $\gamma$-transform ((Baccelli et al., 2001)).

Formally, the $\gamma$-transform of $x$ is defined as

$$X(\gamma) = \bigoplus_{k \in \mathbb{Z}} x(k) \otimes \gamma^k . \qquad (4)$$

The definition of powers in max-plus algebra is extended to negative powers of $\gamma$ by

$$\gamma^a \otimes \gamma^b = \gamma^{a+b} \text{ and } \gamma^a \otimes \gamma^{-a} = \gamma^0 = e, \text{ for } a, b \in \mathbb{Z},$$

where the exponents are added using conventional algebra.

Applying (4) to (3) provides

$$X(\gamma) = \mathcal{A}(\gamma) \otimes X(\gamma) \oplus U(\gamma), \qquad (5)$$
$$\text{with} \quad \mathcal{A}(\gamma) = \bigoplus_{q \in \mathbb{Z}} A_{(q)} \gamma^q,$$
$$\text{and} \quad U(\gamma) = \bigoplus_{k \in \mathbb{Z}} u(k) \gamma^k.$$

The elements of the matrix $\mathcal{A}(\gamma)$ can therefore be determined as max-plus polynomials

$$[\mathcal{A}(\gamma)]_{ji} = \bigoplus_{q \in \mathbb{Z}} a_{ji}^{(q)} \gamma^q, \qquad (6)$$

where $a_{ji}^{(q)} = [A_{(q)}]_{ji}$. Since HTS models, by definition, do not contain multiple arcs between two nodes this equation can also be written as:

$$[\mathcal{A}(\gamma)]_{ji} = \begin{cases} a_{ji}^{(q)} \gamma^q & \text{if } a_{ji}^{(q)} \neq \varepsilon \\ \varepsilon & \text{else.} \end{cases} \qquad (7)$$

After transforming the system into the $\gamma$-domain, another transform has to be applied to remove the dependencies of negative orders. Such dependencies are represented by $\gamma$-entries with negative powers in $\mathcal{A}(\gamma)$. We introduce a transformation matrix $\mathcal{T}(\gamma)$ and a transformed vector $\tilde{X}(\gamma) = \mathcal{T}(\gamma) \otimes X(\gamma)$, i.e, $\mathcal{T}(\gamma)^{-1} \otimes \tilde{X}(\gamma) = X(\gamma)$. Substituting this into (5) results in

$$\tilde{X}(\gamma) = \underbrace{\mathcal{T}(\gamma)\mathcal{A}(\gamma)\mathcal{T}(\gamma)^{-1}}_{\tilde{\mathcal{A}}(\gamma)} \otimes \tilde{X}(\gamma) \oplus \underbrace{\mathcal{T}(\gamma) \otimes U(\gamma)}_{\tilde{U}(\gamma)}. \qquad (8)$$

The goal is to find a transformation matrix $\mathcal{T}$ such that $\tilde{\mathcal{A}}(\gamma)$ is devoid of entries with negative powers.

A possible choice is a diagonal matrix $\mathcal{T}$ where the diagonal elements are powers of $\gamma$, i.e.,

$$\mathcal{T}(\gamma) = \begin{pmatrix} \gamma^{\pi_1} & \varepsilon & \cdots & \cdots & \varepsilon \\ \varepsilon & \gamma^{\pi_2} & \varepsilon & \cdots & \varepsilon \\ \vdots & & \ddots & & \vdots \\ \vdots & & & \ddots & \varepsilon \\ \varepsilon & \cdots & \cdots & \varepsilon & \gamma^{\pi_n} \end{pmatrix},$$

with $\pi_i \in \{\mathbb{N} \cup 0\}, i = 1, \ldots, n$.

Obviously,

$$\left[ \mathcal{T}(\gamma)^{-1} \right]_{ji} = \begin{cases} \gamma^{-\pi_i} & \text{for } j = i \\ \varepsilon & \text{else.} \end{cases} \qquad (9)$$

Therefore,

$$\left[ \tilde{\mathcal{A}} \right]_{ji} = \gamma^{\pi_j} [\mathcal{A}]_{ji} \gamma^{-\pi_i}. \qquad (10)$$

Substituting Equation (7) into this equation results in

$$\left[ \tilde{\mathcal{A}} \right]_{ji} = \gamma^{\pi_j} a_{ji}^{(q)} \gamma^q \gamma^{-\pi_i}. \qquad (11)$$

Thus, to ensure that the matrix $\tilde{\mathcal{A}}$ is devoid of elements with negative powers, we have to choose the non-negative integers $\pi_i, \ i = 1, \ldots, n$, such that

$$\pi_j - \pi_i \geq -q \qquad (12)$$

for every element $[\mathcal{A}]_{ji} \neq \varepsilon$.

It can be shown, that conditions (12) can be satisfied simultaneously if the extended precedence graph $\mathcal{G}_{sys}$ does not contain negative order circuits. This is always true for implementable specifications (Geyer (2004)).

This procedure neither yields a unique transformation matrix $\mathcal{T}$, nor a unique $\tilde{\mathcal{A}}$. However, any transformation matrix obtained that way provides the desired result.

In a last step, it is necessary to apply the inverse $\gamma$-transform to the polynomial matrix

$$\tilde{\mathcal{A}}(\gamma) = \bigoplus_{q \in \mathbb{N}_0} \tilde{A}_{(q)} \gamma^q, \tag{13}$$

to provide the coefficient matrices $\tilde{A}_{(q)}, q \in \mathbb{N}_0$. It can further be shown that the elements of the corresponding state vector $\tilde{x}(k)$ and input vector $\tilde{u}(k)$ are given by:

$$\begin{aligned} \tilde{x}_i(k) &= x_i(k - \pi_i), \\ \tilde{u}_i(k) &= u_i(k - \pi_i), \end{aligned} \tag{14}$$

Therefore, the vector $\tilde{x}$ contains time instants of events which do not necessarily belong to the same batch, i.e., the events within the single batch time scheme are relabled.

Thus, we obtain an implicit recurrence relation for a time-optimal schedule of the HTS system

$$\tilde{x}(k) = \bigoplus_q \left( \tilde{A}_{(q)} \otimes \tilde{x}(k - q) \right) \oplus \tilde{u}(k), \tag{15}$$

with $k \in \mathbb{Z}$, $q \in \mathbb{N}_0$. Equation (15) can be rewritten in its explicit form as $\tilde{A}_{(0)}$ is acyclic:

$$\tilde{x}(k) = \bigoplus_q \left( \tilde{A}_{(0)}^* \tilde{A}_{(q)} \otimes \tilde{x}(k - q) \right) \oplus \tilde{A}_{(0)}^* \tilde{u}(k), \tag{16}$$

where $\tilde{A}_{(0)}^* = I \oplus \tilde{A}_{(0)} \oplus \tilde{A}_{(0)}^2 \oplus \ldots \oplus \tilde{A}_{(0)}^{n-1}$, and $k \in \mathbb{Z}$, $q \in \mathbb{N}$.

Returning to our HTS example, there are 14 arcs in the graph $\mathcal{G}_{sys}$ illustrated in Figure 6, i.e., there are 14 non-$\varepsilon$ elements in the corresponding matrix $\mathcal{A}(\gamma)$:

$$\mathcal{A}(\gamma) = \begin{pmatrix} \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \gamma^2 \\ 9 & \varepsilon & 3 & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ 6 & \varepsilon & \varepsilon & \gamma & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & 12 & \varepsilon & 3 & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & 9 & \varepsilon & \varepsilon & \gamma & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & 16 & \varepsilon & 6 & \varepsilon \\ \varepsilon & \gamma^{-1} & \varepsilon & \varepsilon & 10 & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & 13 & \varepsilon \end{pmatrix}$$

Thus, a total of 14 constraints have to hold for a valid transformation matrix $\mathcal{T}$:

$$\begin{array}{llll} \pi_1 - \pi_8 \geq -2, & \pi_2 - \pi_1 \geq 0, & \pi_2 - \pi_3 \geq 0, & \pi_3 - \pi_1 \geq 0 \\ \pi_3 - \pi_4 \geq -1, & \pi_4 - \pi_3 \geq 0, & \pi_4 - \pi_5 \geq 0, & \pi_5 - \pi_3 \geq 0 \\ \pi_5 - \pi_6 \geq -1, & \pi_6 - \pi_5 \geq 0, & \pi_6 - \pi_7 \geq 0, & \pi_7 - \pi_2 \geq 1 \\ \pi_7 - \pi_5 \geq 0, & \pi_8 - \pi_7 \geq 0. \end{array}$$

One possible solution meeting all constraints is $\pi_1 = \pi_2 = \pi_3 = \pi_4 = \pi_5 = 0$, $\pi_6 = \pi_7 = \pi_8 = 1$. Applying the resulting transformation gives the following system without negative order dependencies:

$$\tilde{X}(\gamma) = \underbrace{\begin{pmatrix} \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \gamma \\ 9 & \varepsilon & 3 & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ 6 & \varepsilon & \varepsilon & \gamma & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & 12 & \varepsilon & 3 & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & 9 & \varepsilon & \varepsilon & 0 & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & 16\gamma & \varepsilon & 6 & \varepsilon \\ \varepsilon & 0 & \varepsilon & \varepsilon & 10\gamma & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & 13 & \varepsilon \end{pmatrix}}_{\tilde{\mathcal{A}}(\gamma)} \otimes \tilde{X}(\gamma) \oplus \tilde{U}(\gamma).$$

The corresponding extended precedence graph of the matrix $\tilde{\mathcal{A}}(\gamma)$ is shown in Figure 7. Since the extended precedence graph does not contain arcs of order higher than 1,
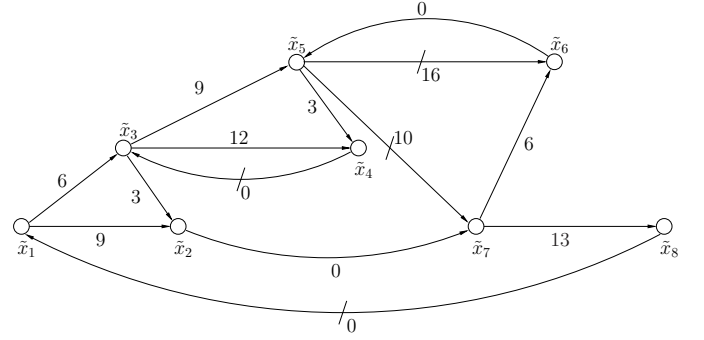


Fig. 7. Extended precedence graph $\tilde{\mathcal{G}}_{sys}$

the resulting recurrence relation of the HTS system can be written as

$$\tilde{x}(k) = \tilde{A}_{(0)} \tilde{x}(k) \oplus \tilde{A}_{(1)} \tilde{x}(k-1) \oplus \tilde{u}(k),$$

which can then be modified into its explicit representation

$$\tilde{x}(k) = \underbrace{\tilde{A}_{(0)}^* \tilde{A}_{(1)}}_{\tilde{A}} \tilde{x}(k-1) \oplus \tilde{A}_{(0)}^* \tilde{u}(k), \tag{17}$$

with $\tilde{A}_{(0)}^* = I \oplus \tilde{A}_{(0)} \oplus \tilde{A}_{(0)}^2 \oplus \ldots \oplus \tilde{A}_{(0)}^{n-1}$.

In general, higher order arcs will also be present in $\tilde{G}_{sys}$. Then by extending the state vector, the system can again be written in the form (17).

In this explicit form it is now possible to analyze the system using max-plus algebra. For example, it is possible to determine the eigenvalue $\lambda$ of $\tilde{A}$ and its corresponding eigenvector $v$. In max-plus algebra the eigenvalue is also called the asymptotic growth rate, as it represents the minimal time interval between the occurrence of the same events in subsequent batches. Thus, the eigenvalue is the minimal cycle time $T$ and represents the optimal solution of the scheduling problem.

Note that the max-plus model is a result of an offline optimization procedure, which determines the sequence and timing of activities on each resource. The max-plus model is a compact representation of the earliest possible instants of time for the occurrence of events in different batches. In the next section, it will be shown that the described max-plus model can be conveniently used for online adjustments (i.e., feedback control) in case unforeseen disturbances make the predetermined optimal policy impossible.

## 4. CONTROL OF HTS SYSTEMS

We propose a model-based feedback control system for HTS plants. The closed loop consists of the HTS plant, a controller and a signal converter. The plant sends information on its current state to the signal converter. The signal converter translates the continuous information from the plant into the discrete-event domain of the controller, which contains the max-plus model of the plant and its predetermined time-optimal schedule.

In the following, we adopt a control scheme proposed in Li et al. (2007) for a class of cyclically repeated discrete-event systems to max-plus models of HTS systems. In this context, the input vector $\tilde{u}(k)$ in (17) is updated in an online fashion to provide information on the current state of the system. We write $\tilde{u}(k, t) = \tilde{x}^{in}(k, t)$, where $\tilde{x}^{in}(k, t)$ can be interpreted as a reinitialized version of the state vector. Its elements are determined by:

- For events that have already occurred in cycle $k$, the corresponding element in $\tilde{x}^{in}(k, t)$ is exactly the time instant of their occurrence:
$$\tilde{x}_i^{in}(k, t) = \tilde{x}_i(k).$$
- For events that are next to occur on each resource, the time of their occurrence is determined by:
$$\tilde{x}_i^{in}(k, t) = \begin{cases} \tilde{x}_i^s(k), & \text{for events w/o delay,} \\ \tilde{x}_i^s(k) + \tilde{x}_i^d(k, t), & \text{for delayed events,} \end{cases}$$
where $\tilde{x}_i^s(k)$ is the scheduled time instant of the occurrence of event $i$ in cycle $k$ and $\tilde{x}_i^d(k, t)$ is its expected delay.
- For all other events, the corresponding entry in $\tilde{x}^{in}(k, t)$ will be :
$$\tilde{x}_i^{in}(k, t) = \varepsilon,$$
i.e., these events will not pose any restriction for the future evolution of the system.

We now insert the feedback scheme described above into (17). The resulting vector of event times in cycle $k$ will then of course also depend on the current update time $t$, i.e.,
$$\tilde{x}(k, t) = \tilde{A}\,\tilde{x}(k-1) \oplus \tilde{A}_{(0)}^* \tilde{x}^{in}(k, t), \tag{18}$$
with $k \in \mathbb{Z}$. This max-plus model can then be used to handle unexpected deviations from the predetermined cyclic schedule of HTS systems. Deviations considered in this work are delays of events, e.g., an activity takes longer than expected. Such a delay can be crucial for the continuous operation of the plant.

If a delay occurs at time $t_d$ during cycle $k$, it will first appear in the vector $\tilde{x}^{in}(k, t_d)$. As indicated in Equation (18), the vector $\tilde{x}(k, t_d)$ can then be determined. This vector is comprised of the time instants of the occurrences of events in the $k$-th cycle. If the delay effects any future events in cycle $k$, the scheduled time instants of their occurrence is rescheduled in $\tilde{x}(k, t_d)$. Furthermore, possible delays of events in the $k$-th cycle may be carried over into the next cycles. This effect can be handled through the first part of Equation (18), where possible deviations of previous cycles are included through the state vector $\tilde{x}(k-1)$. The newly determined (updated) information on the time instants of future events can then be used to control the plant.

Generally speaking, after the occurrence of a delay the max-plus controller postpones every future event by the minimal amount of time needed to make the schedule feasible. Doing so every constraint included in the max-plus model is enforced. In the terms of scheduling theory this is also referred to as a minimal *right shift rescheduling* (Vieira et al., 2003). Consequently, the controller ensures continuous operation of the HTS plant and the system returns to a cyclic operation as fast as possible.

## 5. CONCLUSION AND FUTURE WORK

This contribution proposes a max-plus algebraic model with negative order arcs for high-throughput screening systems operated in a predetermined time-optimal cyclic mode. It is shown how a system with negative order arcs can be transformed into a system devoid of such arcs. Furthermore, the control scheme introduced in Li et al. (2007) is adapted and it is shown how the max-plus model can be used to handle deviations of the predetermined cyclic schedule, such as delay of events.

First results show, that the proposed model can easily be extended to HTS systems with multi-capacity resources, i.e., resources that can handle more than one activity at the same time. Given a HTS system with multi-capacity resources further deviations from the predetermined cyclic scheme, such as partial break down of multi-capacity resources, have to be considered.

Another aspect that will be investigated is the possibility of using the max-plus algebra framework to change between two different cyclic schedules.

## REFERENCES

Baccelli, F., Cohen, G., Olsder, G.J., and Quadrat, J.P. (2001). *Synchronization and Linearity – An Algebra for Discrete Event Systems.* Wiley, web edition.

Geyer, F. (2004). *Analyse und Optimierung zyklischer ereignisdiskreter Systeme mit Reihenfolgealternativen.* Diploma thesis, Otto-von-Guericke-Universität Magdeburg.

Heidergott, B., Olsder, G.J., and van der Woude, J. (2006). *Max Plus at Work.* Princeton University Press, Princeton, NJ, USA, 1st edition.

Li, D., Mayer, E., and Raisch, J. (2007). A new Hierarchical Control Scheme for a Class of Cyclically Repeated Discrete-Event Systems. In *Informatics in Control, Automation and Robotics II*, 227–233. Springer Netherlands.

Mayer, E. and Raisch, J. (2004). Time-optimal scheduling for high throughput screening processes using cyclic discrete event models. *Mathematics and Computers in Simulation*, 66, 181–191.

Vieira, G.E., Herrmann, J.W., and Lin, E. (2003). Rescheduling Manufacturing Systems: A Framework of Strategies, Policies, and Methods. *Journal of Scheduling*, 6(1), 39–62.