

# Max-Plus Algebraic Modeling and Control of High-Throughput Screening Systems with Multi-Capacity Resources

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 previous work we have shown how a max-plus algebraic model can be derived for cyclically operated high-throughput screening systems and how such a model can be used to design a controller to handle unexpected deviations from the predetermined cyclic operation during runtime. In this paper, we introduce an extension of this approach for high-throughput screening systems containing multi-capacity resources, i.e., resources that can handle more than one activity at the same time.

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

---

## 1. INTRODUCTION

High-throughput screening (HTS) has become a standard technology for drug discovery in pharmaceutical industries. HTS plants are fully automated systems that are able to analyze thousands of biochemical compounds in a very short time.

In high-throughput screening, a batch subsumes all work-steps that are necessary to analyze one set of substances. Such a set consists of up to 1536 substances, which are aggregated on one microplate. Additional microplates may be included in the batch to convey reagents or waste material. An HTS plant involves a fixed set of resources performing liquid handling, storage, reading, plate handling and incubation steps. For comparison reasons the sequence and the timing of activities that have to be performed on a batch – the single batch time scheme – has to be identical for all batches. Cyclic operation is therefore an important requirement.

A method to determine 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 at least one globally optimal solution of the scheduling problem is retained. 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 problem significantly, it is still too complex to be solved 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, though, do not perform well when deviations from the predetermined cyclic scheme occur during runtime.

However, using the predetermined static schedule it is possible to develop a max-plus algebraic model of the HTS system's operation (Brunsch and Raisch, 2009). Based on this model, a supervisor may be designed that generates possible actions to be taken in case of unexpected deviations from the cyclic scheme. Doing so, the supervisor updates the schedule of the HTS plant and thus ensures its continuous operation.

In this paper the max-plus algebraic modeling and control scheme is extended to HTS plants containing multi-capacity resources, i.e., resources that can handle more than one activity at the same time. Such resources are contained in many high-throughput screening plants. One of the most common multi-capacity resources in HTS plants is the incubator, where the biochemical substances are allowed to bind to or react (or fail to react) with each other.

This paper is structured as follows. Section 2 briefly describes the fundamentals of graph theory and max-plus algebra. The specifications for high-throughput screening systems are explained in Section 3. Using an illustrative example it is explained how the constraints are merged into a max-plus algebraic model of the HTS operation and how multi-capacity resources can be incorporated into the model. In Section 4, the max-plus algebraic control scheme introduced by Li et al. (2007) is extended to HTS systems with multi-capacity resources. Conclusions and suggestions for future work are given in Section 5.

---

\* This work has been partially supported by the European Community's Seventh Framework Programme under project DISC (Grant Agreement n. INFSO-ICT-224498).

## 2. GRAPH THEORY AND MAX-PLUS ALGEBRA

### 2.1 Fundamentals of Graph Theory

A directed graph 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 directed graph is a directed graph with a real number (the weight)  $w_{ji} \in \mathbb{R}$  assigned to each arc  $(v_i, v_j) \in \mathcal{E}$ . All weights of the graph can be written as a 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. If no arc exists from node  $v_i$  to node  $v_j$  the corresponding entry in the matrix  $W$  is set to  $-\infty$ . The pair  $(\mathcal{V}, \mathcal{E})$  is then called the precedence graph of  $W$ . If the weights  $w_{ji} \in \mathbb{R}_{max}$  represent times, the respective 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 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:  $\forall a, b \in \mathbb{R}_{max}$ :

$$\begin{aligned} a \oplus b &:= \max(a, b) \\ a \otimes b &:= a + b. \end{aligned}$$

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  $e$ .

Addition of matrices in max-plus algebra for  $A, B \in \mathbb{R}_{max}^{n \times m}$  is defined by

$$[A \oplus B]_{ji} = [A]_{ji} \oplus [B]_{ji}.$$

Multiplication of max-plus matrices  $A \in \mathbb{R}_{max}^{n \times l}$  and  $B \in \mathbb{R}_{max}^{l \times m}$  is defined by

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

Similar to conventional algebra, some standard properties such as associativity, commutativity, and distributivity of  $\otimes$  over  $\oplus$  hold for max-plus algebra.

Systems with (cyclic) repetition of events can be represented in max-plus algebra by:

$$\begin{aligned} x(k) &= \bigoplus_q (A_q \otimes x(k-q)) \oplus B \otimes u(k) \\ y(k) &= C \otimes x(k), \end{aligned}$$

with  $k \in \mathbb{Z}$  and  $q \in \{\mathbb{N} \cup 0\}$ , where the vectors  $u(k)$  and  $y(k)$  contain the earliest time instants for the occurrence of certain input and output events in the  $k$ -th cycle. The elements of the matrix  $A_0$  represent the minimum time offsets between events occurring in the same cycle, while the matrices  $A_q$  with  $q > 0$  refer to minimum time offsets between events in previous cycles and events in the current cycle. If matrix  $A_0$  is acyclic, i.e., its precedence graph does

not contain any circuits, the matrix  $A_0^* = I \oplus A_0 \oplus A_0^2 \oplus \dots$  can be determined as the finite sum  $A_0^* = I \oplus A_0 \oplus A_0^2 \oplus \dots \oplus A_0^{n-1}$ , where  $I$  is the identity matrix with respect to max-plus algebra. In this case the implicit recurrence relation can be rewritten in an explicit form:

$$\begin{aligned} x(k) &= \bigoplus_q (A_0^* \otimes A_q \otimes x(k-q)) \oplus A_0^* \otimes B \otimes u(k) \\ y(k) &= C \otimes x(k), \end{aligned}$$

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

## 3. MAX-PLUS MODEL OF HTS SYSTEMS

The specific operation the user wants to run determines requirements for the single batch time scheme. It consists of  $i_{max}$  activities which are executed on  $m$  resources. Thus, each activity  $i$  is assigned to a specific resource  $J_i \in \{1, \dots, m\}$ . During the execution of activity  $i$  the respective resource  $J_i$  is said to be occupied. As activities of a batch may overlap in time, it is possible that a batch occupies two resources at the same time. This is, for example, the case during transfer of a microplate from one resource to another one.

The minimal requirements for the single batch time scheme can be modeled using time window precedence networks. To do so, each activity  $i$  is described by three different kinds of events, i.e., start events denoted by  $o_i$ , release events denoted by  $r_i$ , and transfer events. The time window precedence network of a simple example is given in Fig. 1.

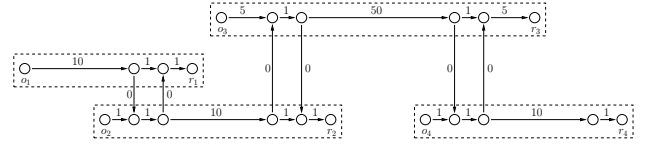


Fig. 1. Time window precedence network to describe requirements for a single batch time scheme.

Often scheduling problems are illustrated by Gantt charts. The Gantt chart of our example is given in Fig. 2. It can be seen that the operation contains four activities, which are executed on a total of three resources. While *Reader* and *Pipettor* are single-capacity resources, *Incubator* is of capacity three. That means that the resource *Incubator*



Fig. 2. Gantt chart of the single batch time scheme

can mount a total of three microplates at the same time. Usually *Incubators* are of a much higher capacity. For simplicity and illustrative reasons, however, we will assume the rather small capacity of three in our running example.

In general, a multi-capacity resource with capacity  $cap = \xi$  can handle  $\xi$  activities concurrently. These concurrent operations are executed independently from one another.

Thus, it is possible to model a multi-capacity resource as  $\xi$  resources with capacity one, which are also called slots and denoted by  $\sigma_i$ . However, when modeling a multi-capacity resource as multiple single-capacity resources, it is a-priori not possible to uniquely assign one specific resource to every activity of every batch. Nonetheless, it is still possible to write and solve a HTS scheduling problem with multi-capacity resources as an MILP. Once a globally optimal solution of the scheduling problem has been obtained, the mapping of every activity to one specific resource is defined.

In HTS it is of utmost importance that every single batch is executed in an identical time scheme. This constraint cannot be satisfied, if a multi-capacity resource is split into multiple single-capacity resources, because the same activities of different batches may now be executed on different resources (slots). It is possible, however, to solve this problem by combining several batches into one *super-batch*. If a HTS plant contains a multi-capacity resource with  $cap = \xi$ , such a super-batch consists of  $\xi$  consecutive single-batch time schemes. The resulting super-batch time scheme can then be repeated without violating the constraints imposed on HTS systems.

Note that if there exists more than one multi-capacity resource in the system, the number of single batches constituting the super-batch is the least common multiple of all capacities of all resources. Formally, if there are  $m$  resources in the system the number of single batches  $\xi$  forming the super-batch can be determined by

$$\xi = \text{lcm}_{i=1, \dots, m} (cap_i).$$

However, dealing with multi-capacity resources in this way may increase the complexity of the max-plus model significantly. If for example the HTS plant contains one resource of capacity  $cap_i = 50$ , one resource with  $cap_j = 3$  and one multi-capacity resource with  $cap_k = 70$  the resulting super-batch would consist of  $\xi = 1,050$  batches. Given that for each batch several hundred activities may have to be executed the (single) super-batch time scheme may easily exceed 100,000 activities.

Fortunately, it is possible to incorporate multi-capacity resources into the max-plus algebraic model without forming a super-batch. This approach is neither dependent on the number of higher capacity resources in the system nor on their capacity and will be shown in Section 3.2.

### 3.1 Conjunctive Constraints

In general, requirements for minimal time offsets between events of a single batch time scheme are called conjunctive constraints and are predefined by the user, i.e., the operation the user wants to run. Thus, the time window precedence network shown in Fig. 1 contains all conjunctive constraints necessary to describe the processing operation of a single batch. It is possible to decrease the complexity, i.e., the number of nodes, of the graph using standard graph reduction methods. The reduced graph for the example from Fig. 1, denoted by  $\mathcal{G}_c$ , is shown in Fig. 3. It contains only nodes representing start and release events of the different activities, which were renamed  $x_1$  to  $x_8$ .

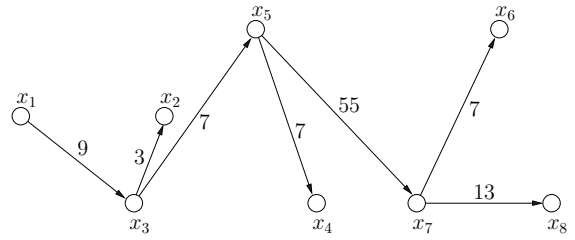


Fig. 3. Graph of conjunctive constraints  $\mathcal{G}_c$ .

The information contained in the reduced conjunctive graph  $\mathcal{G}_c$  can be summarized in a matrix  $A_c$ . For our example,  $A_c \in \mathbb{R}_{max}^{8 \times 8}$  is

$$A_c = \begin{bmatrix} \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & 3 & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ 9 & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & 7 & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & 7 & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & 7 & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & 55 & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & 13 & \varepsilon \end{bmatrix}.$$

Clearly,  $\mathcal{G}_c$  is the precedence graph of  $A_c$ . Using the precedence matrix  $A_c$ , a max-plus algebraic system of the single batch requirements can be written as

$$x = A_c \otimes x, \quad (1)$$

where  $x$  refers to the earliest possible time instants of each activity's start and end events. By definition, the single batch time scheme does not contain any circuits, thus, matrix  $A_c$  is acyclic.

### 3.2 Disjunctive Constraints

Disjunctive constraints describe the temporal dependencies between activities (in all batches) which are executed on a specific resource. In particular, these constraints impose restrictions such that activities are prohibited to start as long as the corresponding resource is fully loaded. They also ensure that deadlocks will not occur during runtime. Note that the sequence of activities on a resource, which is part of the disjunctive constraints, is a result of the offline optimization procedure.

In general, every resource  $\kappa$  has its own set of disjunctive constraints which can be modeled as so called disjunctive graphs. Given that the system consists of  $m$  resources, we will obtain  $m$  disjunctive graphs  $\mathcal{G}_{d_\kappa}, \kappa = 1, \dots, m$ . To denote dependencies between events in different batches standard precedence graphs are extended by marked arcs. A marked arc connecting an event to an event in a subsequent batch is indicated by a "/". Similarly, a marked arc connecting an event to an event in a previous batch is indicated by a "o" on the arc. The number of "/" or "o" symbols, respectively, on a marked arc represents its order  $q$ , i.e., the difference in batch numbers for the corresponding events.

Depending on the capacity of the resource the disjunctive constraints may differ. Therefore, the approach for single-capacity resources is discussed first before we will deal with the disjunctive constraints for multi-capacity resources.

*Single-Capacity Resources:* We will describe the disjunctive constraints in detail for one resource. For all other single-capacity resources the procedure can be used in the same manner.

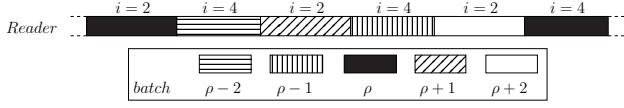


Fig. 4. Sequence of activities performed on *Reader*.

For the resource *Reader* ( $\kappa = 2$ ) of our example, the resource allocation is given as a Gantt chart in Fig. 4. It can be seen that two different activities are executed on this resource. Activity  $i = 2$  of a batch  $\rho$  is immediately followed by activity  $i = 4$  of batch  $\rho - 2$  which itself is then followed by activity  $i = 2$  of batch  $\rho + 1$ . Thus the disjunctive graph for this resource consists of one arc of order  $q = -2$ , one arc of order  $q = 3$  and two zero order arcs. The corresponding extended precedence graph  $\mathcal{G}_{d_2}$  is shown in Fig. 5.

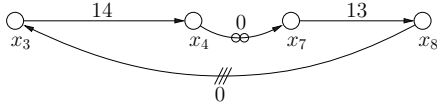


Fig. 5. Extended precedence graph of the disjunctive constraints on resource *Reader*.

Similar to conjunctive graphs, the information provided by disjunctive graphs  $\mathcal{G}_{d_\kappa}$  can be written as a max-plus algebraic system

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

where  $x$  is the vector containing the earliest time instants for start and release events of all activities and  $q$  represents the order of arcs. More precisely, the matrix  $A_{d_\kappa(q)} \in \mathbb{R}_{max}^{n \times n}$  encodes all arcs of order  $q$  within the disjunctive graph of resource  $\kappa$ .

*Multi-Capacity Resources:* For multi-capacity resources the aforementioned approach cannot be applied directly. Fig. 6 shows the Gantt chart of the multi-capacity resource *Incubator* ( $\kappa = 3$ ), which is of capacity three. As in the Gantt-chart in Fig. 4, the sequence of activities is the outcome of an offline optimization procedure. Activities can be executed independently on the slots  $\sigma_1, \sigma_2, \sigma_3$ . Due

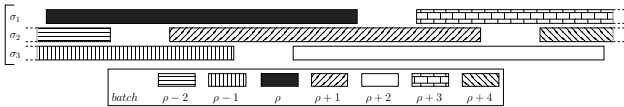


Fig. 6. Sequence of activities performed on *Incubator*.

to this independent operation it is possible to determine a set of disjunctive constraints for every slot of the resource. It turns out that at least one of the globally optimal solutions determined offline exhibits an identical sequencing of activities on each slot of a multi-capacity resource. Hence the set of disjunctive constraints is identical for every slot of a multi-capacity resource. For our example the resulting extended precedence graph of the disjunctive constraints for each slot of resource *Incubator* is shown in Fig. 7.

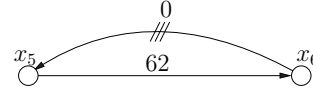


Fig. 7. Extended precedence graph of the disjunctive constraints on each slot of resource *Incubator*.

As a consequence, the disjunctive constraints for a multi-capacity resource are completely described by the disjunctive constraints for one of its slots. Note that the capacity of the resource can be easily deduced from the precedence graph describing the disjunctive constraints of a slot. More precisely, the capacity is identical to the sum of the orders of all arcs within the disjunctive graph for the slot. Thus, for the resource *Incubator* ( $\kappa = 3$ ) the sum of the orders of all arcs in Fig. 7 is three. For the resource *Reader* ( $\kappa = 2$ ) adding the orders of arcs provides  $cap_2 = 2$ .

It is obvious that the graphs for the slots can also be written in the form (2).

### 3.3 Model of HTS Systems

Finally, all the obtained graphs, i.e., one conjunctive graph and  $m$  disjunctive graphs (one for each resource independent of its capacity) can be merged. This will result in an extended precedence graph  $\mathcal{G}_{sys}$ . For our example this graph is shown in Fig. 8. The max-plus representation

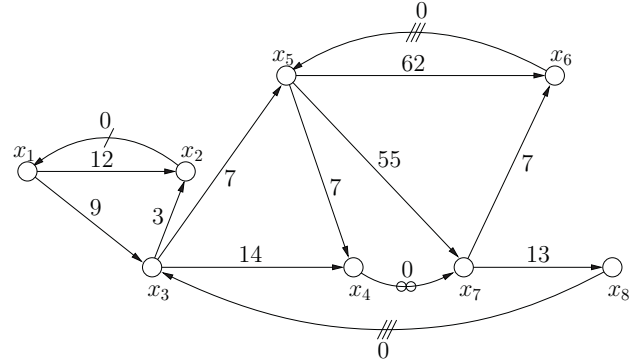


Fig. 8. Extended precedence graph  $\mathcal{G}_{sys}$ .

of this graph can be directly derived by merging the max-plus representations of the conjunctive and disjunctive constraints, i.e., Eq. (1) and Eq. (2). Furthermore, it is possible to include a control input to the system, modeling the possibility to delay the occurrence of events. This is important, if disturbances cause deviations from the previously determined optimal policy and online control becomes necessary to quickly recover the desired cyclic scheme.

If all events can be delayed independently, introducing control is equivalent to adding (in the max-plus sense) a control vector  $u(k)$  to give

$$\begin{aligned}
x(k) = & \underbrace{\left( A_c \oplus \left( \bigoplus_{\kappa=1}^m A_{d_{\kappa}(0)} \right) \right)}_{A_{(0)}} x(k) \\
& \oplus \left( \bigoplus_{q \neq 0} \underbrace{\left( \bigoplus_{\kappa=1}^m A_{d_{\kappa}(q)} \right)}_{A_{(q)}} x(k-q) \right) \oplus u(k), \quad (3)
\end{aligned}$$

with  $q, k \in \mathbb{Z}$ .

In general Equation (3) contains matrices  $A_{d_{\kappa}(q)}$  with  $q < 0$ . These matrices refer to negative order dependencies in the precedence graph of the complete system. In Fig. 8 the arc connecting node  $x_4$  to node  $x_7$  is of order  $q = -2$ . Such negative order arcs represent acausalities with respect to the cycle index  $k$  (though, of course, not with respect to absolute time). For further analysis it is advantageous to reformulate the system such that it does not contain negative order arcs.

It has been shown in (Bruns and Raisch, 2009) that this can be done using the  $\gamma$ -transform combined with a “relabeling procedure”. The resulting transformation is not unique but, according to Geyer (2004), it is always possible to find a transformation matrix which provides the desired result for implementable specifications.

For our example, one possible transformed system graph  $\tilde{\mathcal{G}}_{sys}$  is shown in Fig. 9. The resulting transformed system

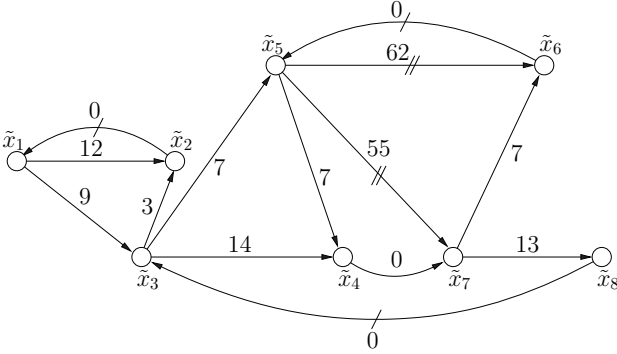


Fig. 9. Transformed extended precedence graph  $\tilde{\mathcal{G}}_{sys}$ .

vector  $\tilde{x}(k) \in \mathbb{R}_{max}^n$  contains events which do not necessarily belong to the same batch.

Corresponding to the graph  $\tilde{\mathcal{G}}_{sys}$  the (transformed) max-plus recurrence relation (3) of the HTS system can be written as

$$\tilde{x}(k) = \bigoplus_q \left( \tilde{A}_{(q)} \otimes \tilde{x}(k-q) \right) \oplus \tilde{u}(k), \quad (4)$$

with  $k \in \mathbb{Z}, q \in \mathbb{N}_0$ . Acyclicity of  $A_{(0)}$  is preserved by the transformation. Therefore (4) can be written in explicit form:

$$\tilde{x}(k) = \bigoplus_q \left( \tilde{A}_{(0)}^* \tilde{A}_{(q)} \otimes \tilde{x}(k-q) \right) \oplus \tilde{A}_{(0)}^* \tilde{u}(k), \quad (5)$$

with  $\tilde{A}_{(0)}^* = I \oplus \tilde{A}_{(0)} \oplus \tilde{A}_{(0)}^2 \oplus \dots \oplus \tilde{A}_{(0)}^{n-1}$ , and  $k \in \mathbb{Z}, q \in \mathbb{N}$ . By extending the vector  $\tilde{x}$ , the higher order system (5)

can then be rewritten as an explicit recurrence relation of order one.

Returning to our example, the transformed system contains arcs of order 0, 1, and 2. Thus, Equation (5) can be written as

$$\tilde{x}(k) = \tilde{A}_{(0)}^* \tilde{A}_{(1)} \tilde{x}(k-1) \oplus \tilde{A}_{(0)}^* \tilde{A}_{(2)} \tilde{x}(k-2) \oplus \tilde{A}_{(0)}^* \tilde{u}(k).$$

To obtain a first order recurrence relation, we define  $\hat{x}(k) = [\tilde{x}^T(k) \ \tilde{x}^T(k-1)]^T$ . Then, the resulting recurrence relation of our example is

$$\hat{x}(k) = \hat{A} \otimes \hat{x}(k-1) \oplus \hat{B} \otimes \tilde{u}(k), \quad (6)$$

with

$$\hat{A} = \begin{bmatrix} \tilde{A}_{(0)}^* \tilde{A}_{(1)} & \tilde{A}_{(0)}^* \tilde{A}_{(2)} \\ I & \mathcal{E} \end{bmatrix} \quad \text{and} \quad \hat{B} = \begin{bmatrix} \tilde{A}_{(0)}^* \\ \mathcal{E} \end{bmatrix},$$

where  $\mathcal{E}$  refers to the zero matrix with respect to max-plus algebra.

Recall that our max-plus model represents a relaxation of the globally optimal offline solution: it contains the optimal sequences of activities on resources and slots; it contains minimum time requirements but does not reflect the optimal single batch time scheme. The purpose of this relaxation is to provide degrees of freedom (delay of events) to react to unforeseen disturbances. In the disturbance-free case, the optimal offline solution can be easily recovered from the max-plus model (6) by computing the eigenvalue and the eigenvector(s) of  $\hat{A}$ . The eigenvalue corresponds to the minimal cycle time and the eigenvector(s) to the optimal time scheme(s).

In the following, we will show, how the max-plus algebraic model (6) can be used as a supervisor that performs online control of the HTS plant in case of unforeseen disturbances.

#### 4. CONTROL OF HTS SYSTEMS

Real-time scheduling problems for multiple resource systems are known to be NP-hard problems (Carpenter et al., 2004). Therefore, we propose to perform the (time consuming) calculation of the optimal solution of the scheduling problem offline and to use this result as a basis for the online control of HTS systems. Online control is restricted to the delay of certain events. Its task is to make sure that the HTS system returns to the predetermined cyclic schedule at least as fast as the standard right-shift rescheduling policy (Vieira et al., 2003) without changing the order of activities on the resources.

For the control of HTS systems we propose a model-based feedback scheme. We measure the current (and possibly disturbed) state of the HTS plant and generate a vector of earliest possible event times (EPETs). We only consider disturbances that correspond to finite-time delays of events. Breakdowns of resources will not be considered because usually the “loss” of a specific resource cannot be compensated by other resources. In general, the breakdown of one or more resources will result in a stop of the complete operation of the HTS system. However, if the machine with a malfunction can be repaired in a reasonable time, the repair time may be handled as a simple delay of events.

To be able to model unforeseen delays it is necessary to expand the max-plus model (6) with a disturbance:

$$\hat{x}_d(k) = \hat{A}\hat{x}_d(k-1) \oplus \hat{B}\tilde{u}(k) \oplus S\tilde{d}(k), \quad (7)$$

where  $\hat{x}_d(k)$  refers to the (possibly) disturbed system state in cycle  $k$ . In the case of HTS systems the matrix  $S$  is assumed to be the identity matrix, since every event may be delayed individually. The entries of vector  $\tilde{d}(k)$ , i.e., the disturbances, are in general unknown.

The state is fed back in an online fashion, i.e., it is updated with a high sampling frequency. The input to the feedback controller is therefore  $\hat{x}_d(k, t_j)$ . The controller then determines the input signal  $\tilde{u} = \tilde{u}(k, t_j) \in \mathbb{R}_{max}^n$ . Inserting such an update into Eq. (7) will result in a max-plus system which will then of course also depend on the current update time  $t_j$ , i.e.,

$$\hat{x}_d(k, t_j) = \hat{A}\hat{x}_d(k-1) \oplus \hat{B}\tilde{u}(k, t_{j-1}) \oplus S\tilde{d}(k, t_j), \quad (8)$$

The elements of the updated input  $\tilde{u}(k, t_j)$  are determined in the following way:

- For an event that has already occurred in cycle  $k$  (or a previous cycle), the corresponding element in  $\tilde{u}(k, t_j)$  is exactly the time instant of its occurrence. For an event which is next to occur on a resource (or slot), the time of its occurrence is estimated. Thus, the input for such events is set to the value of the corresponding element in  $\hat{x}_d(k, t_j)$ , i.e.,

$$\tilde{u}_i(k, t_j) = \hat{x}_{d,i}(k, t_j),$$

where the right hand side is determined by (8).

- For all other events, the corresponding entry in  $\tilde{u}(k, t_j)$  will be:

$$\tilde{u}_i(k, t_j) = \varepsilon,$$

i.e., these events will not pose any restriction for the future evolution of the system.

The overall structure of the controlled HTS system is shown in Fig. 10.

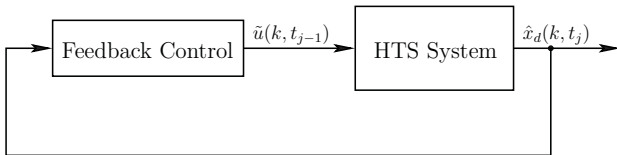


Fig. 10. Control structure.

If some (next) event  $i$  is delayed in cycle  $k$  at time  $t_j$ , the information of the delay will be contained in the corresponding element of the input vector  $\tilde{u}_i(k, t_j)$ . Using the max-plus algebraic model it is then possible to determine the vector  $\hat{x}_d(k, t_{i+1})$ . The first  $n$  elements of this vector contain the earliest possible event times for all events of the  $k$ th cycle.

Generally speaking, the feedback controller postpones the occurrence of every event affected by the delay of another event. As max-plus algebra determines the earliest possible event times for future events, the controller postpones events by the minimal amount of time needed to make the schedule feasible. Consequently the controller guarantees continuous operation of the HTS plant even if unforeseen delays of events occur during runtime.

## 5. CONCLUSION AND FUTURE WORK

In this paper we propose an extension of the previously introduced max-plus algebraic modeling and control scheme (Brunsch and Raisch, 2009) for high-throughput screening systems containing multi-capacity resources. It has been shown how multi-capacity resources can be incorporated into this model without increasing the dimension of the state space of the max-plus model.

Furthermore it has been explained how the max-plus algebraic model can be used to determine an online control scheme for the HTS system to handle deviations of the predetermined cyclic schedule.

As mentioned before, the deviations from the predetermined cyclic scheme considered in this paper are restricted to delays of events. Due to the lack of redundancy with respect to resources, the permanent breakdown of a resource will result in a disruption of the operation of the HTS system. If however a breakdown only affects parts of a resource, namely only one or more slots of a multi-capacity resource such that at least one slot remains available, the operation of the system may (partially) continue. Therefore, such a partial breakdown of multi-capacity resources has to be considered.

Furthermore, the possibility of using the max-plus algebra framework to change between two different cyclic schemes during runtime has to be investigated.

## 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.
- Brunsch, T. and Raisch, J. (2009). Max-plus algebraic modeling and control of high-throughput screening systems. In *2nd IFAC Workshop on Dependable Control of Discrete Systems*.
- Carpenter, J., Funk, S., Holman, P., Srinivasan, A., Anderson, J., and Baruah, S. (2004). A categorization of real-time multiprocessor scheduling problems and algorithms. In *Handbook on Scheduling Algorithms, Methods, and Models*. Chapman Hall/CRC, Boca.
- 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.