# Control of cyclically operated High-Throughput Screening Systems [⋆]

**T. Brunsch** [*,**] **L. Hardouin** [**] **J. Raisch** [*,***]

[*] *Fachgebiet Regelungssysteme, Technische Universität Berlin,*
*10587 Berlin, Germany (email: brunsch@control.tu-berlin.de)*
[**] *Laboratoire d'Ingénierie des Systèmes Automatisés,*
*Université d'Angers, 49000 Angers, France*
[***] *Fachgruppe System- und Regelungstheorie, Max-Planck-Institut für*
*Dynamik komplexer technischer Systeme, 39106 Magdeburg, Germany*

**Abstract:** In previous work we have shown how $(\max, +)$-algebra can be used to model cyclically operated high-throughput screening systems. In this paper the system is modeled in a two-dimensional dioid $\mathcal{M}_{in}^{ax}[\![\gamma, \delta]\!]$. A controller is determined using residuation theory. The resulting control guarantees *just-in-time* operation of the plant. A small example is used to demonstrate the approach to model and control HTS systems. To apply the determined controller, it is rewritten in terms of *counter*-functions. A simulation of the system with and without controller is given and results are discussed.

*Keywords:* Cyclic discrete-event systems, dioid algebra, residuation theory, high-throughput screening, scheduling

## 1. INTRODUCTION

In recent years high-throughput screening (HTS) has become an important technology to quickly test thousands of biochemical substances. Especially in the pharmaceutical industries, HTS is used for a first screening in the process of drug discovery, to reduce the almost unlimited number of possible combinations of active ingredients to a reasonable number of compounds, on which further screening methods are applied.

In general, high-throughput screening plants are fully automated systems containing a fixed set of devices performing liquid handling, storage, reading, plate handling, and incubation steps. All operations which have to be conducted to analyze one set of substances are combined in a batch. A set of substances consists of up to 1536 compounds which are aggregated on one microplate. Additional microplates may be included in a batch to convey reagents or waste material. To compare the screening results of different compound sets, the single batch time scheme, i.e., the sequence and the timing of activities for one batch, needs to be identical for all batches. Thus, it is desired to run HTS plants in a cyclic manner.

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. The optimization problem results in a mixed integer non-linear program (MINLP), which can be transformed into a mixed integer linear program (MILP). This transformation significantly decreases the complexity of the problem but it is guaranteed that a globally optimal solution of the original scheduling problem is retained. However, the process of solving MILPs is still computationally too expensive to be carried out online. Therefore, the globally optimal solution is determined off-line resulting in a static schedule. Of course static schedules do not perform well when deviations from the predetermined cyclic scheme occur during runtime (Murray and Anderson, 1996).

However, using the result from the optimization process, i.e., the information on the globally optimal solution for the scheduling problem, it is possible to determine a model of the HTS system in a $(\max, +)$-algebraic setting (Brunsch and Raisch, 2009b). This approach has been extended to HTS systems containing multi-capacity resources, i.e., resources that can handle more than one activity at the same time (Brunsch and Raisch, 2009a).

In this paper, a feedback controller is computed based on a $\mathcal{M}_{in}^{ax}[\![\gamma, \delta]\!]$-model of the HTS system. To apply the determined controller to a real system, e.g., in a PLC, it is easier to rewrite it in a $(\min, +)$-setting. The advantage of a controller description in $(\min, +)$-algebra is its *counter* characteristic, i.e., the events that have occurred up to a specific time are counted. Thus, it is dependent on time which can then be sampled by the PLC. If any deviations from the predetermined cyclic scheme occur during runtime the controller generates possible actions to return as fast as possible to the optimal schedule and thus ensures its continuous operation.

This paper is structured as follows. Section 2 briefly describes the mathematical theories used in our approach. The modeling process of high-throughput screening systems is outlined in Section 3. An illustrative example is introduced to explain the modeling process in detail.

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

In Section 4 the feedback control scheme introduced by Lhommeau et al. (2002) is adapted to the requirements of HTS systems. The calculated controller for our example is applied to the model and a simulation is presented in Section 5. Conclusions are given in Section 6.

## 2. MATHEMATICAL BACKGROUND

### *2.1 Dioids*

A dioid is an idempotent semiring, i.e. an algebraic structure $(\mathcal{D}, \oplus, \otimes)$ containing two binary operations $\oplus$ and $\otimes$ defined on $\mathcal{D}$ (Baccelli et al., 2001).The addition $\oplus$ of the dioid is associative, commutative, and idempotent, i.e., $a \oplus a = a$, $\forall a \in \mathcal{D}$. The multiplication $\otimes$ is associative but not necessarily commutative. Multiplication is left- and right-distributive with respect to $\oplus$. The neutral elements of $\oplus$ and $\otimes$ are denoted by $\varepsilon$ and $e$, respectively.

As in standard algebra, addition and multiplication are not restricted to scalar elements. For matrices $A, B \in \mathcal{D}^{n\times m}$ and $C \in \mathcal{D}^{m\times l}$ addition and multiplication are defined by

$$[A \oplus B]_{ji} = [A]_{ji} \oplus [B]_{ji}$$
$$[A \otimes C]_{ji} = \bigoplus_{k=1}^{m} \left([A]_{jk} \otimes [C]_{ki}\right).$$

Powers are defined by $a^i = a \otimes a^{i-1}$ with $a^0 = e$. As in conventional algebra, $\otimes$ is often omitted in equations.

Due to the idempotency property, a partial order relation can be defined for dioids by $a \preceq b \Leftrightarrow a \oplus b = b$, i.e., the sum of the elements $a$ and $b$ is the least upper bound of $a$ and $b$.

An idempotent semiring is said to be complete if it is closed for infinite sums and if $\otimes$ distributes over infinite sums. The implicit equation $x = ax \oplus b$ defined over a complete idempotent semiring $(\mathcal{D}, \oplus, \otimes)$ admits $x = a^*b$, with $a^* := \bigoplus_{i=\mathbb{N}_0} a^i$, as the least solution. Some useful properties of the star operator in idempotent semirings are the following:

$$\begin{aligned} a^*(ba^*)^* &= (a \oplus b)^* = (a^*b)^*a^* \\ (ab)^*a &= a(ba)^*. \end{aligned} \tag{1}$$

Two of the most commonly used idempotent semirings are $(\max, +)$- and $(\min, +)$-algebra. In $(\max, +)$-algebra, the addition is defined by the standard max operator, i.e., $a \oplus b := \max(a, b)$, and multiplication of two elements is defined by the standard addition of these elements, i.e., $a \otimes b := a + b$. These operations are defined on the set $\mathbb{Z}_{\max} := \mathbb{Z} \cup \{-\infty\}$ and the neutral elements for addition and multiplication are $\varepsilon = -\infty$ and $e = 0$, respectively.

Similarly, $(\min, +)$-algebra is defined on the set $\mathbb{Z}_{\min} := \mathbb{Z} \cup \{+\infty\}$, and addition and multiplication are defined by $a \oplus b := \min(a, b)$ and $a \otimes b := a + b$. The neutral elements are $\varepsilon = +\infty$ and $e = 0$. $(\max, +)$- and $(\min, +)$-algebra defined on the sets $\overline{\mathbb{Z}}_{\max} := \mathbb{Z}_{\max} \cup \{+\infty\}$ and $\overline{\mathbb{Z}}_{\min} := \mathbb{Z}_{\min} \cup \{-\infty\}$ constitute complete dioids.

The dioid considered in this paper is denoted $\mathcal{M}_{in}^{ax}[\![\gamma, \delta]\!]$ (Baccelli et al., 2001). It is defined as the quotient dioid of $\mathbb{B}[\![\gamma, \delta]\!]$, the set of formal power series in two variables $(\gamma, \delta)$ with Boolean coefficients $\mathbb{B} = \{\varepsilon, e\}$ and with exponents in $\mathbb{Z}$ with respect to the equivalence relation $x\mathcal{R}y \Leftrightarrow \gamma^*(\delta^{-1})^*x = \gamma^*(\delta^{-1})^*y$. In addition to the conventional sum and product of series, the following rules apply for the dioid $\mathcal{M}_{in}^{ax}[\![\gamma, \delta]\!]$:

$$\gamma^k\delta^t \oplus \gamma^l\delta^t = \gamma^{\min(k,l)}\delta^t$$
$$\gamma^k\delta^t \oplus \gamma^k\delta^\tau = \gamma^k\delta^{\max(t,\tau)}.$$

The neutral elements for addition and multiplication of $\mathcal{M}_{in}^{ax}[\![\gamma, \delta]\!]$ are $\varepsilon = \gamma^{+\infty}\delta^{-\infty}$ and $e = \gamma^0\delta^0$. The dioid is complete with top element $T = \gamma^{-\infty}\delta^{+\infty}$.

This algebraic structure is very efficient to model synchronization graphs, a subclass of timed Petri-nets which can be used to model deterministic discrete-event systems subject to synchronization phenomena. More precisely, the dynamical behavior of a synchronization graph can be described by a linear model in $\mathcal{M}_{in}^{ax}[\![\gamma, \delta]\!]$ (Baccelli et al., 2001). The information of the monomial $\gamma^k\delta^t \in \mathcal{M}_{in}^{ax}[\![\gamma, \delta]\!]$ in terms of synchronization graphs may be interpreted as: *the k-th event occurs at the latest at time t* or *at time t at least k events have occurred* (Cottenceau et al., 1999).

### *2.2 Trajectories in Discrete-Event Systems*

Naturally, cyclic discrete-event systems evolve in a periodic manner. Periodic series in the dioid $\mathcal{M}_{in}^{ax}[\![\gamma, \delta]\!]$ are usually represented in the form $s = p \oplus q \otimes r^*$. In such series, the term $p$ is a polynomial referring to a transient phase, e.g., a start-up of the system. The term $q$ is then a polynomial which represents the periodical behavior, i.e., the pattern which will be repeated with a periodicity given by $r = \gamma^\nu\delta^\tau$. Then the ratio $\nu/\tau$ is the throughput of the series, i.e., an event occurs $\nu$ times every $\tau$ time units, once the periodic regime is reached.

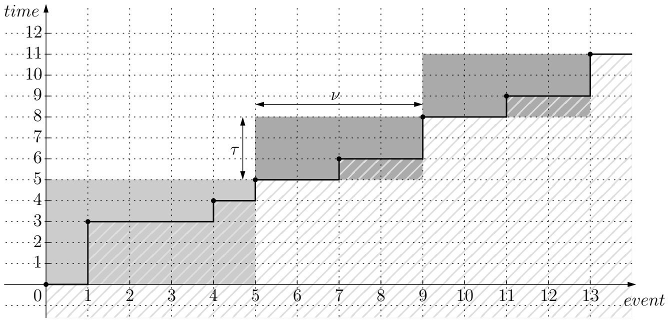

Fig. 1. Graphical representation of the trajectory $s = e \oplus \gamma\delta^3 \oplus \gamma^4\delta^4 \oplus \left(\gamma^5\delta^5 \oplus \gamma^7\delta^6\right)\left(\gamma^4\delta^3\right)^*$.

For example, consider the series $s \in \mathcal{M}_{in}^{ax}[\![\gamma, \delta]\!]$:

$$s = \underbrace{e \oplus \gamma\delta^3 \oplus \gamma^4\delta^4}_{p} \oplus \underbrace{\left(\gamma^5\delta^5 \oplus \gamma^7\delta^6\right)}_{q}\underbrace{\left(\gamma^4\delta^3\right)^*}_{r^*}.$$

The graphical representation of this series is given in Fig. 1. Note that in $\mathbb{B}[\![\gamma, \delta]\!]$, the monomial $\gamma^k\delta^\tau$ is represented as the point $(k, \tau)$ in the event-time-domain. Because we work in $\mathcal{M}_{in}^{ax}[\![\gamma, \delta]\!]$, this monomial is identified with all points in the south-east cone of $(k, \tau)$ (shaded area in Fig. 1).

### 2.3 Residuation Theory

An isotone, or order preserving, mapping $f$ from one partially ordered set $\mathcal{S}$ to another partially ordered set $\mathcal{T}$ is a mapping that satisfies $a \preceq b \Rightarrow f(a) \preceq f(b),\ \forall a, b \in \mathcal{S}$. It is called residuated if there exists a unique mapping $f^{\#} : \mathcal{T} \to \mathcal{S}$ such that $f \circ f^{\#} \preceq \mathrm{Id}_{\mathcal{T}}$ and $f^{\#} \circ f \succeq \mathrm{Id}_{\mathcal{S}}$, where $\mathrm{Id}_{\mathcal{T}}$ and $\mathrm{Id}_{\mathcal{S}}$ refer to the identity functions on $\mathcal{T}$ and $\mathcal{S}$, respectively (Blyth and Janowitz, 1972). If $f$ is residuated, $f^{\#}$ is called the residual of $f$. It can be shown that $f^{\#}(y)$ is the least upper bound of the subset $\{x \mid f(x) \preceq y\}$.

Two of the most frequently used isotone mappings are the left and right multiplication over a complete dioid, i.e., $L_a : x \mapsto a \otimes x$ and $R_a : x \mapsto x \otimes a$. Both mappings are residuated and their residuals are denoted by $L_a^{\#}(x) = a \circ\!\!\!\!\backslash x$ and $R_a^{\#} = x \circ\!\!\!/ a$, respectively.

## 3. MODELING OF HTS SYSTEMS

Depending on the specific operations the user wants to run on a set of compounds, one can determine certain requirements for the single batch time scheme. It consists of $\mu$ activities, which are executed on $\varrho$ resources. Each activity is therefore assigned to a specific resource $J_i \in \{1, \ldots, \varrho\}$. While activity $i$ is executed, the assigned resource $J_i$ is said to be occupied. In general, it is possible that one batch occupies two (or more) resources at the same time. For example, this is the case when a microplate is moved from one resource to another one. During the transfer both resources are occupied by the same batch concurrently. It may also be possible that a batch consists of more than one microplate (e.g., there are additional microplates carrying reagents) and the set of microplates of a single batch may occupy different resources at the same time.

The minimal requirements for the single batch time scheme can be modeled with synchronization graphs. For better understanding we introduce a simple example of an HTS operation. This example consists of three activities which are executed on two different resources. The first activity is the filling of some biochemical substance $A$ into the wells of a microplate. This activity is executed on the resource *Pipettor*. After that, the microplate is moved to a storage and has to stay there for at least seven time units. Since we assume the storage to be of infinite capacity, this resource is not explicitly modeled. Once the waiting time has elapsed, the microplate enters the resource *Pipettor* again. At the same time a second microplate, carrying a biochemical substance $B$, is moved by a *Robot* to the *Pipettor* as well. There, substance $B$ is moved to the microplate carrying substance $A$. For real HTS systems further activities, such as incubation steps or reading operations, would be executed on the compound $AB$. However, for simplicity reasons, we restrict our example to the three activities $act_1$, $act_2$, and $act_3$, executed on the two resources *Pipettor* and *Robot*.

According to our desired operation, we define a synchronization graph, which is shown in Fig. 2. In this graph the timing information is given by the numbers attached to the places. Note that these numbers denote minimal required times, i.e., every activity must be delayed at least by the time given but may be delayed even further. The resulting degrees of freedom can be used to obtain an optimal schedule for the operation of the HTS plant.

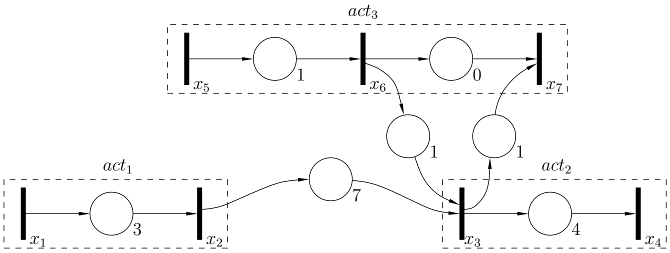

Fig. 2. Single batch time scheme.

In the Petri-net graph in Fig. 2 the dashed boxes indicate the three different activities. While activities $act_1$ and $act_2$ are executed on *Pipettor*, activity $act_3$ occupies *Robot*. It can be clearly seen how $act_3$ and the start of $act_2$ are synchronized. In the $(\max, +)$-algebraic setting $x(k)$ denotes the earliest possible time instant of the occurrence of transition $x$. Thus, the event evolution point of view is considered, i.e., the state variables are dater functions which link event number $k$ to and occurrence date $x(k)$. In the $(\min, +)$-algebraic setting, the time domain is considered, i.e., the state variables considered are counter functions, counting the number of events $x(t)$ that have occurred up to time $t$.

Often, schedules are graphically represented by so called Gantt charts. The Gantt chart of the shortest single batch time scheme of our example is illustrated in Fig. 3.

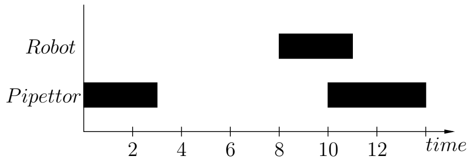

Fig. 3. Gantt chart of a single batch.

After the single batch time scheme has been defined by the user, a globally optimal schedule for the processing of a large number of batches can be determined (Mayer and Raisch, 2004). Additional information can be included in the optimization process, e.g., capacity of each resource, minimal time duration between two activities executed on the same resource, etc. For our example, each resource has a capacity of one, i.e., every resource can execute only one activity at a time, and between the release event of one activity and the start event of the following activity executed on the same resource a time of at least one time unit has to elapse. Using all this information the optimization algorithm determines a globally optimal schedule which is given as a Gantt chart in Fig. 4.

Note that in the optimal schedule the delay between $act_1$ and $act_2$ of a single batch has been increased when compared to the shortest single batch time scheme, i.e., in the optimal schedule of our example a total of ten time units elapse between these two activities (instead of the required minimum of seven time units). Thus, the throughput of the overall system, i.e., the number of batches processed per time unit, is increased by increasing the execution time of a single batch.

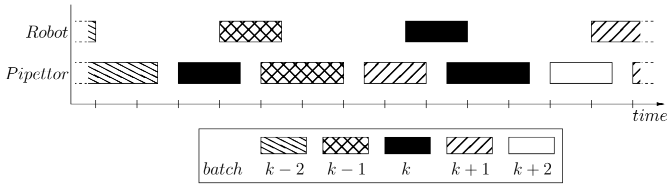

Fig. 4. Gantt chart of the globally optimal schedule.

Clearly, one can see how activities of different cycles, i.e., batches, are nested in the globally optimal schedule. More precisely, activities of the $(k-1)$-st and $(k+1)$-st batch are scheduled on *Pipettor* in between the execution of activities $act_1$ and $act_2$ of the $k$-th batch. Also the additional requirement of at least one time unit between the execution of two activities on the same resource is included in the optimal schedule.

To determine a controller for the system, we first have to include the information obtained from the optimization process in our model given in Fig. 2. The extended synchronization graph is illustrated in Fig. 5.

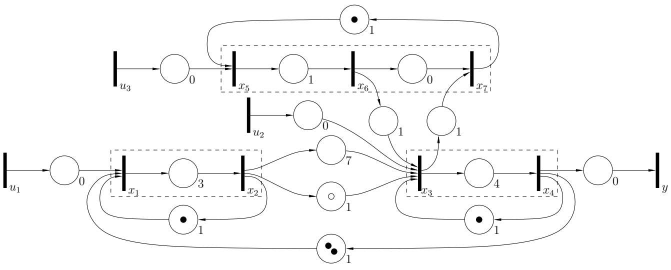

Fig. 5. Extended synchronization graph of the globally optimal schedule with additional input and output events

In this graph, each activity's release event is connected with its start event. The connecting place has a delay of one time unit and contains one token. This means that each activity in a batch can start at the earliest one time unit after the same activity of the previous batch has ended. Furthermore, a connection is introduced between the release event of $act_2$, i.e., $x_4$, and the start event of $act_1$, i.e., $x_1$. This connection represents the repetitive nature of the determined optimal cyclic behavior. The corresponding place contains two tokens due to the sequence of the optimized schedule, i.e., after $act_2$ of the $(k-2)$-nd batch ends, $act_1$ of the $k$-th batch starts. Similarly, a connection is added between the release event of $act_1$ and the start event of $act_2$. The new place, however, contains a negative token, represented as a white circle, indicating that the second activity of the $k$-th batch starts after the first activity of the $(k+1)$-st batch has been finished. This can be interpreted as acausality with respect to the cyclic index $k$. Regarding time, however, the system is of course causal.

Additionally, input- and output events are introduced. Generally, one can delay the start events of every activity of the HTS system, i.e., every start event is directly connected to (at least one) control input. The output of HTS systems is defined to be the release event of the last activity of a batch (comp. Fig. 5).

The dependencies between events in the synchronization graph can be written as $\mathcal{M}_{in}^{ax}[\![\gamma,\delta]\!]$-algebraic equations. According to Fig. 5, the dependencies for event $x_1$ of the $k$-th batch are

$$x_1(k) = \max\left(1 + x_2(k-1), 1 + x_4(k-2), u_1(k)\right),$$

which can be rewritten in $\mathcal{M}_{in}^{ax}[\![\gamma,\delta]\!]$-algebra

$$x_1(\gamma\delta) = \gamma\delta x_2(\gamma\delta) \oplus \gamma^2\delta x_4(\gamma\delta) \oplus u_1(\gamma\delta).$$

Similarly, the dependencies of all other events, i.e., $x_2$ to $x_7$, and the output event $y$ can be determined. In general, this information can also be summarized in matrix form

$$\begin{aligned} x(\gamma\delta) &= A(\gamma\delta) \otimes x(\gamma\delta) \oplus B(\gamma\delta) \otimes u(\gamma\delta), \\ y(\gamma\delta) &= C(\gamma\delta) \otimes x(\gamma\delta) \end{aligned} \tag{2}$$

with $A \in \mathcal{M}_{in}^{ax}[\![\gamma,\delta]\!]^{n\times n}$, $B \in \mathcal{M}_{in}^{ax}[\![\gamma,\delta]\!]^{n\times\pi}$, and $C \in \mathcal{M}_{in}^{ax}[\![\gamma,\delta]\!]^{m\times n}$. The vectors $x \in \mathcal{M}_{in}^{ax}[\![\gamma,\delta]\!]^{n\times 1}$, $u \in \mathcal{M}_{in}^{ax}[\![\gamma,\delta]\!]^{\pi}$, and $y \in \mathcal{M}_{in}^{ax}[\![\gamma,\delta]\!]^{m}$ represent the trajectories of the system's internal state, its input and output, respectively. For our example, we get

$$A = \begin{bmatrix} \varepsilon & \gamma\delta & \varepsilon & \gamma^2\delta & \varepsilon & \varepsilon & \varepsilon \\ \delta^3 & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \delta^7 \oplus \gamma^{-1}\delta & \varepsilon & \gamma\delta & \varepsilon & \delta & \varepsilon \\ \varepsilon & \varepsilon & \delta^4 & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \gamma\delta \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & \delta & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \delta & \varepsilon & \varepsilon & e & \varepsilon \end{bmatrix}$$

$$B = \begin{bmatrix} e & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & e & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & e \\ \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon \end{bmatrix} \quad \text{and} \quad C = \begin{bmatrix} \varepsilon & \varepsilon & \varepsilon & e & \varepsilon & \varepsilon & \varepsilon \end{bmatrix}.$$

Using the star operator in dioids, (2) can be rewritten as

$$\begin{aligned} x &= A^*Bu \\ y &= \underbrace{CA^*B}_{G}u, \end{aligned} \tag{3}$$

where $G$ is a matrix representing the input/output relation of the system. The elements of matrix $G$ are periodic series in the dioid $\mathcal{M}_{in}^{ax}[\![\gamma,\delta]\!]$.

## 4. CONTROL OF HTS SYSTEMS

As mentioned before, control for HTS systems refers to the start events of the system's activities. Note that control is restricted to delaying the occurrence of these events, i.e., the firing of the corresponding transitions.

The model equations are given in (2). We can apply a state feedback (comp. Fig. 6)

$$u = Fx \oplus v. \tag{4}$$

Clearly, the closed loop equations are

$$\begin{aligned} x &= (A \oplus BF)\, x \oplus Bv \\ &= (A \oplus BF)^*\, Bv \end{aligned} \tag{5}$$

$$\begin{aligned} y &= Cx \\ &= C\,(A \oplus BF)^*\, Bv \end{aligned} \tag{6}$$

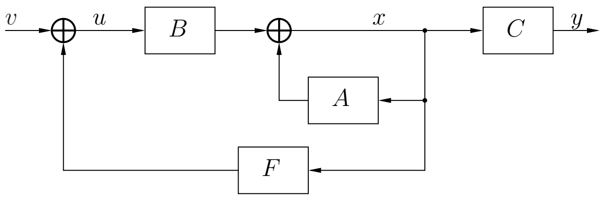

Fig. 6. Control Structure.

Using standard properties of the star operator in idempotent semirings (1), (6) is equivalent to

$$y = \underbrace{CA^*B(FA^*B)^*}_{H} v. \tag{7}$$

Therefore, $H \in (\mathcal{M}_{in}^{ax}[\![\gamma,\delta]\!])^{m\times\pi}$ can be seen as the transfer matrix describing the input-output relation of the controlled system.

The aim of control is to delay the start of activities as long as possible without affecting the throughput of the system. Recall, due to the off-line optimization, the system is designed to achieve the highest throughput possible. In other words, all activities in the closed loop are required to start *just-in-time* to maintain the optimal throughput of the system (Lhommeau et al., 2002). Furthermore, it is desired to operate the system in a strictly cyclic manner at all times, including start-up and shut-down procedures, i.e., the time schemes of all batches need to be identical to compare the results of the biochemical reactions. Formally, the greatest $F$ has to be determined such that

$$CA^*B(FA^*B)^*v \preceq CA^*Bv \quad \forall v$$

which is equivalent to

$$CA^*B(FA^*B)^* \preceq CA^*B. \tag{8}$$

Applying residuation theory provides

$$(FA^*B)^* \preceq (CA^*B)\backslash\!\!\!\circ(CA^*B). \tag{9}$$

Using a standard property of left division in complete idempotent semirings, i.e., $a\backslash\!\!\!\circ a = (a\backslash\!\!\!\circ a)^*$, (9) can be rewritten

$$(FA^*B)^* \preceq ((CA^*B)\backslash\!\!\!\circ(CA^*B))^*, \tag{10}$$

which is equivalent to

$$(FA^*B) \preceq (CA^*B)\backslash\!\!\!\circ(CA^*B). \tag{11}$$

Thus, a controller $F$ is feasible for the considered system, if

$$F \preceq (CA^*B)\backslash\!\!\!\circ(CA^*B)/\!\!\!\circ(A^*B), \tag{12}$$

and the desired greatest controller

$$F = (CA^*B)\backslash\!\!\!\circ(CA^*B)/\!\!\!\circ(A^*B), \tag{13}$$

leads to equality in (9).

For our example, there are seven events, including three start events which can be delayed. Hence, the controller $F$ is a 3×7-matrix in $\mathcal{M}_{in}^{ax}[\![\gamma,\delta]\!]$. The optimal (greatest) $F$ has been determined using existing software for manipulating periodic series (Hardouin et al., 2001):

$$F^T = \begin{bmatrix} (\gamma\delta^9)^* & \delta^{13}(\gamma\delta^9)^* & \delta^{11}(\gamma\delta^9)^* \\ \gamma\delta^6(\gamma\delta^9)^* & \delta^{10}(\gamma\delta^9)^* & \delta^8(\gamma\delta^9)^* \\ \gamma^2\delta^5(\gamma\delta^9)^* & (\gamma\delta^9)^* & \gamma\delta^7(\gamma\delta^9)^* \\ \gamma^2\delta(\gamma\delta^9)^* & \gamma\delta^5(\gamma\delta^9)^* & \gamma\delta^3(\gamma\delta^9)^* \\ \gamma^2\delta^7(\gamma\delta^9)^* & \delta^2(\gamma\delta^9)^* & (\gamma\delta^9)^* \\ \gamma^2\delta^6(\gamma\delta^9)^* & \delta(\gamma\delta^9)^* & \gamma\delta^8(\gamma\delta^9)^* \\ \gamma^2\delta^4(\gamma\delta^9)^* & \gamma\delta^8(\gamma\delta^9)^* & \gamma\delta^6(\gamma\delta^9)^* \end{bmatrix} \tag{14}$$

The resulting transfer matrix $H$ for the example is

$$\begin{aligned} H &= CA^*B(FA^*B)^* \\ &= \left[\, \delta^{17}(\gamma\delta^9)^* \quad \delta^4(\gamma\delta^9)^* \quad \delta^6(\gamma\delta^9)^* \,\right]. \end{aligned} \tag{15}$$

Note that every element in $F$ as well as in $H$ contains $r^* = (\gamma\delta^9)^*$. This represents the throughput of the system: Once the periodic behavior has been reached (after a transient start-up procedure) every event occurs once every 9 time units. Thus, one batch will be released every 9 time units.

A controller for a real HTS system may be implemented on a PLC, which works with a specific sample time. Thus, for the implementation of a controller, a representation in terms of *counters*, counting the number of every event that has occurred until the current sampling instant, would be easier to handle. The equation $\tilde{u}(\gamma\delta) = \gamma^n\delta^d \otimes \tilde{x}(\gamma\delta)$ in $\mathcal{M}_{in}^{ax}[\![\gamma,\delta]\!]$ can be interpreted in standard algebra as $\tilde{u}(t) = n + \tilde{x}(t-d)$, where $\tilde{u}(t)$ and $\tilde{x}(t)$ are counters. The latter can be easily realized in a PLC. Indeed, for a signal $\tilde{u}(\gamma\delta) = (p \oplus qr^*) \otimes \tilde{x}(\gamma\delta)$, with $p = \bigoplus_{i=1}^{N_p} \gamma^{n_i}\delta^{d_i}$, $q = \bigoplus_{j=1}^{N_q} \gamma^{n_j}\delta^{d_j}$, $r = \gamma^\nu\delta^\tau$ we can obtain counter $\tilde{u}(t)$ in the following manner in $(\min,+)$-algebra

$$\begin{aligned} \xi(t) &= \nu \otimes \xi(t-\tau) \oplus \left(\bigoplus_{j=1}^{N_q} n_j \otimes \tilde{x}(t-d_j)\right) \\ \tilde{u}(t) &= \bigoplus_{i=1}^{N_p} n_i \otimes \tilde{x}(t-d_i) \oplus \xi(t), \end{aligned} \tag{16}$$

where the operator $\oplus$ is the usual min, and $\otimes$ is $+$.

If we consider the computation of the first control-input $u_1$ for our example, it can be determined (comp. (4)) by

$$u_1(\gamma\delta) = \bigoplus_{l=1}^{n} F_{1,l}x_l(\gamma\delta) \oplus v_1(\gamma\delta),$$

where $n$ is the number of state variables $x$ in the system, i.e., $n = 7$. Using the controller given in (14), this equation results in

$$\begin{aligned} u_1(\gamma\delta) =& (\gamma\delta^9)^*x_1 \oplus \gamma\delta^6(\gamma\delta^9)^*x_2 \oplus \overbrace{\gamma^2\delta^5(\gamma\delta^9)^*x_3}^{u_{1,3}(\gamma\delta)} \oplus \\ & \oplus \gamma^2\delta(\gamma\delta^9)^*x_4 \oplus \gamma^2\delta^7(\gamma\delta^9)^*x_5 \oplus \\ & \oplus \gamma^2\delta^6(\gamma\delta^9)^*x_6 \oplus \gamma^2\delta^4(\gamma\delta^9)^*x_7 \oplus v_1. \end{aligned}$$

Note that the coefficients of $x_i$ are of the form $p \oplus qr^*$ with $p = \varepsilon$. As an example, consider the third term of the sum, i.e., $u_{1,3}(\gamma\delta) = \gamma^2\delta^5(\gamma\delta^9)^*x_3(\gamma\delta)$. Using (16), this can be written as a counter in terms of $(\min,+)$-algebra

$$\begin{aligned} \xi_3(t) =& 1 \otimes \xi_3(t-9) \oplus 2 \otimes x_3(t-5) \\ u_{1,3}(t) =& \varepsilon \oplus \xi_3(t), \end{aligned}$$

which is equivalent to

$$u_{1,3}(t) = \min\left(1 + \xi_3(t-9), 2 + x_3(t-5)\right)$$

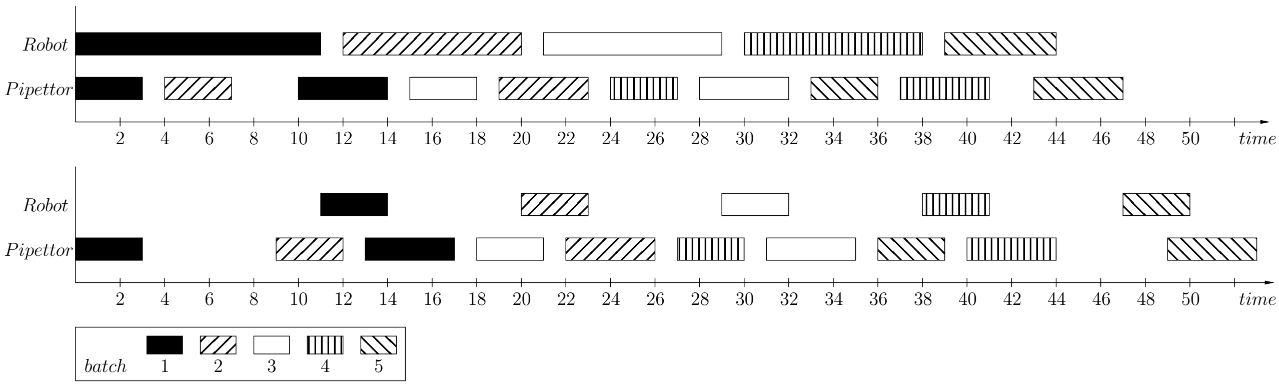

Fig. 7. Gantt chart of the simulated system without controller (upper part) and with controller (lower part). The individual batches are indicated by different colors.

in standard algebra. Similarly, all other terms in $u_1(\gamma\delta)$ can be transformed into counter-functions in $(\min,+)$-algebra, which can then be easily implemented in a PLC.

## 5. SIMULATION OF THE SYSTEM

The described HTS example and the determined controller has been implemented in *C*-language. The simulation scenario considered includes the processing of a total of five batches. The defined restrictions of the process are as given above. Namely, it is desired to obtain a strictly cyclic behavior where every single batch has an identical time scheme. Furthermore, the system shall operate by means of the *just-in-time* criterion, starting activities as late as possible without interfering with the pre-determined optimal throughput. The Gantt chart of the simulated system with and without controller is given in Fig. 7.

In this figure it can be seen that the total time to process five batches is 47 time units for the uncontrolled system and 53 time units for the controlled process. However, even though the overall time increases by implementing control, it can also be clearly seen that while the controlled system follows a strictly cyclic pattern the uncontrolled system does not. Therefore, given that all batches need to be identical to compare the results of the biochemical reactions, the first and last batch may not generate useful data for the uncontrolled system. It is important to note that the throughput $\nu/\tau$ of the periodical part of the uncontrolled case is identical to the throughput of the controlled case. Additionally, it can clearly be seen that the activity executed on *Robot* starts earlier than necessary when no controller is implemented. Applying a controller, however, will minimize the execution time of this activity by starting this activity *just-in-time*. It can be argued that the *just-in-time* execution of closed loop activities contribute to a fast recovery of the optimal schedule after disturbances in the form of delays occur.

## 6. CONCLUSION

This paper addresses the design of feedback control for high-throughput screening systems. It has been shown how a controller can be derived from a $\mathcal{M}_{in}^{ax}[\![\gamma,\delta]\!]$-algebraic representation of HTS systems using residuation theory. In a last step, the control has been rewritten in a $(\min,+)$-setting to simplify its implementation in a PLC. The resulting feedback delays the start of activities as long as possible without influencing the throughput of the system, which is optimal in the *just-in-time* context. Recall that the presented example has been chosen deliberately simple for illustration purposes. Much larger examples from HTS have been successfully solved in exactly the same manner.

## ACKNOWLEDGMENTS

The authors would like to thank Thomas Haenel from CyBio AG—a company developing and manufacturing HTS plants—for his comments and suggestions regarding the modeling and operation of HTS systems.

## 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.

Blyth, T. and Janowitz, M. (1972). *Residuation Theory.* Oxford: Pergamon Press.

Brunsch, T. and Raisch, J. (2009a). Max-plus algebraic modeling and control of high-throughput screening systems with multi-capacity resources. In *Preprints of the 3rd IFAC Conference on Analysis and Design of Hybrid Systems (ADHS'09)*, 132–137. Zaragoza, Spain.

Brunsch, T. and Raisch, J. (2009b). Max-plus algebraic modeling and control of high-throughput screeing systems. In *2nd IFAC Workshop on Dependable Control of Discrete Systems.*

Cottenceau, B., Hardouin, L., Boimond, J.L., and Ferrier, J.L. (1999). Synthesis of greatest linear feedback for timed event graphs in dioid. *IEEE Transactions on Automatic Control*, 44(6), 1258–1262.

Hardouin, L., Cottenceau, B., and Lhommeau, M. (2001). Software tools for manipulating periodic series. http://istia.univ-angers.fr/ hardouin/outils.html.

Lhommeau, M., Hardouin, L., and Cottenceau, B. (2002). Disturbance Decoupling of Timed Event Graphs by Output Feedback Controller. In *WODES'2002, Workshop on Discrete Event Systems.* Zaragoza, Spain.

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.

Murray, C. and Anderson, C. (1996). Scheduling software for high-throughput screening. *Laboratory Robotics and Automation*, 8(5), 295–305.