Delft University of Technology

Delft Center for Systems and Control

Technical report 13-040

A modeling framework for model predictive scheduling using switching max-plus linear models*

T.J.J. van den Boom, G.D. Lopes, and B. De Schutter

If you want to cite this report, please use the following reference instead:
T.J.J. van den Boom, G.D. Lopes, and B. De Schutter, "A modeling framework for model predictive scheduling using switching max-plus linear models," *Proceedings of the 52nd IEEE Conference on Decision and Control*, Florence, Italy, pp. 5456–5461, Dec. 2013.

Delft Center for Systems and Control
Delft University of Technology
Mekelweg 2, 2628 CD Delft
The Netherlands
phone: +31-15-278.24.73 (secretary)
URL: `https://www.dcsc.tudelft.nl`

*This report can also be downloaded via `https://pub.deschutter.info/abs/13_040.html`

A Modeling Framework for Model Predictive Scheduling Using Switching Max-Plus Linear Models

Ton J.J. van den Boom[1], Gabriel Delgado Lopes[1], and Bart De Schutter[1]

***Abstract*— In this paper we discuss a modeling framework for model predictive scheduling of a class of semi-cyclic discrete event systems that can be described by switching max-plus linear models. We study the structure of the system matrices and derive how routing, ordering, and synchronization can be manipulated by a set of control variables. In addition, we show that this leads to a system matrix that is linear in the control variables. We define the model predictive scheduling design problem to optimize the schedule, and we show that the problem can be recast as a mixed integer linear programming (MILP) problem.**

I. INTRODUCTION

Global competition in industry has compelled engineers to improve the quality of production systems and to reduce the costs. The operation of these systems requires a supervisory controller that schedules the jobs in order to optimize certain criteria, e.g., high productivity or first-product-out time. The main goal of this paper is to derive a design procedure for optimal scheduling of discrete event systems with a semi-cyclic behavior. A cyclic behavior is specified by a cycle periodically repeated set of operations behind the other. In the case of semi-cyclic behavior the set of operations may vary over a limited set of possible sequences of operations. The scheduling of semi-cyclic discrete event systems is crucial in many applications, such as railroad and urban traffic networks [13], [28]; production systems [23], [31]; paper handling in printers [2]; legged locomotion [17], [20]; queuing systems and array processors [14]; and genomics [27].

Scheduling is the process of deciding how to allocate a set of jobs to limited resources over time in such a way that one or more objectives are optimized. We can use a model of the system to predict the future behavior while searching for an optimal schedule for the future. If the model is perfect, the optimal schedule can be executed without feedback and the system will behave as predicted. However, in the case of disturbances or model uncertainty the schedule has to be adapted on-line in response to the unexpected events. This is called *operational scheduling* or *rescheduling*.

Since the preliminary work of Johnson [16] on scheduling problems, several books have presented general surveys [5], [9], [25]. However, [6], [7], [18], [19], [24], [30] already showed that for many scheduling problems, above all for semi-cyclic discrete event systems, max-plus algebra is better suited for solving sequencing problems than the classical algebra. An overview of the cyclic scheduling problem is available in [10], [12]. In this paper we continue this work on scheduling semi-cyclic discrete event systems using switching max-plus linear models, and extend this work so that we can handle rerouting, reordering, and resynchronization in a systematic way. We like to emphasize the importance of the use of switching max-plus linear systems in the scheduling procedure:

1) By using max-plus linear systems for modeling dynamical discrete event systems, we are able to analyze the evolution of the system by computing the system trajectories. In this way we can obtain valuable information about its behavior and use this information to explain observed phenomena.
2) There are many system-theoretical results for max-plus linear systems in literature. We can use them for finding bottlenecks in the scheduling process as well as good initial scheduling values by using system properties, based on the max-plus eigenvalue and eigenvectors (see [17]).
3) There is a close relation between max-plus linear models and the graph representation of the system. Graph-based methods can be used in the scheduling procedure (see [21]).
4) Max-plus linear models are often written implicitly: routing, ordering, and synchronization occur within the same cycle resulting in expressions where the state variables appear on both sides of the constraint equations for the same cycle k. Using max-plus theory we can easily transform such a model into an explicit form from which the optimization of the schedule may benefit.

The model predictive scheduler contains four basic modules: the switching max-plus-linear model, the schedule optimization module, the observer, and the actuator. Figure 1 shows the interconnection structure of these four basic modules and their environment.

Max-plus algebra

First we give the basic definition of the max-plus algebra [4], [11].
Define $\varepsilon = -\infty$ and $\mathbb{R}_\varepsilon = \mathbb{R} \cup \{\varepsilon\}$. The max-plus-algebraic addition ($\oplus$) and multiplication ($\otimes$) are defined as follows:

$$x \oplus y = \max(x, y) \qquad , \qquad x \otimes y = x + y$$

for any $x, y \in \mathbb{R}_\varepsilon$, and

$$[A \oplus B]_{i,j} = a_{i,j} \oplus b_{i,j} = \max(a_{i,j}, b_{i,j})$$

[1]Delft University of Technology, Delft Center for Systems and Control, Delft, The Netherlands (e-mail: {a.j.j.vandenboom, g.a.delgadolopes, b.deschutter}@tudelft.nl).

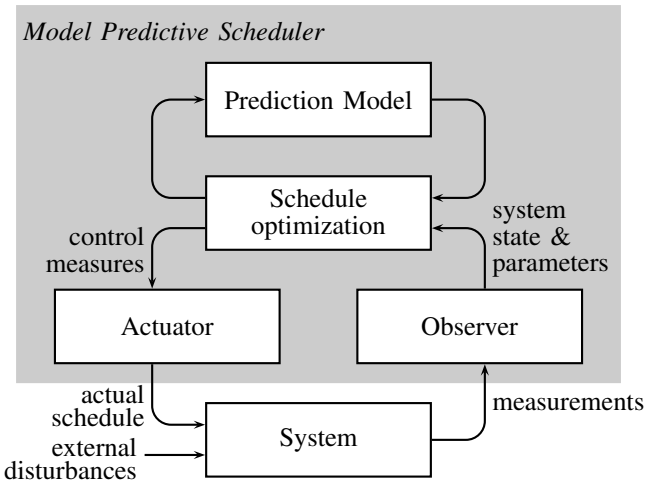

Fig. 1. The model predictive scheduler and its environment

$$[A \otimes C]_{i,j} = \bigoplus_{k=1}^{n} a_{i,k} \otimes c_{k,j} = \max_{k=1,\ldots,n} (a_{i,k} + c_{k,j})$$
$$[A \odot B]_{ij} = a_{ij} + b_{ij}$$

for matrices $A, B \in \mathbb{R}_{\varepsilon}^{m\times n}$ and $C \in \mathbb{R}_{\varepsilon}^{n\times p}$. The last operation ($\odot$) is the max-plus Schur product. The matrix $\mathcal{E}$ is the max-plus-algebraic zero matrix: $[\mathcal{E}]_{i,j} = \varepsilon$ for all i, j.

Let $u \in \mathbb{B}_{\varepsilon} = \{0, \varepsilon\}$ be a max-plus binary variable, then the adjoint variable $\bar{u} \in \mathbb{B}_{\varepsilon}$ is defined as follows:

$$\bar{u} = \begin{cases} 0 & \text{if } u = \varepsilon \\ \varepsilon & \text{if } u = 0 \end{cases}$$

II. SWITCHING MAX-PLUS-LINEAR MODELS

The core of the model predictive scheduling approach is the semi-cyclic model for which a *max-plus linear (MPL) system* description is proposed. An MPL system is a discrete-event dynamic system characterized by synchronization constraints that excellently fits the modeling of large-scale discrete event systems.

Three basic types of control decisions that play a major role in scheduling are routing, ordering, and synchronization. *Routing* decides how a job follows a sequence of resources. A job typically consists of a number of operations that have to be performed using different resources. Each job has to follow a route through the system [16]. Often alternative routes are also possible and the routing for each job has to be determined. Once the routing of the jobs has been done, conflicts may occur while jobs need to be operated at the same resource. The second step in the scheduling is to determine the *ordering* of concurring jobs in resources. Finally some jobs running on different resources may need *synchronization*. An operation of a job can only start when a specific operation of another job has finished. This be the third and final step in the scheduling.

Routing in MPL systems

We aim to derive models that describe the behavior of the semi-cyclic discrete event systems, and that contain the three basic types of control decisions variables for routing, synchronization, and ordering.

Consider a system that has to operate M jobs. For each job a specific route through the system has to be scheduled and resources have to be ordered accordingly. Let job $j \in \{j = 1, \ldots, M\}$ consist of p_j operations on the sequence of resources $\mathcal{R}_j = (r_{j,1}, \ldots, r_{j,p_j})$ in processing order, and let $\mathcal{T}_j(k) = (\tau_{j,1}(k), \ldots, \tau_{j,p_j}(k))$ be the corresponding processing times in cycle k with $\tau_{j,i}(k) \geq 0$ for all i, j. Each operation is assigned to a unique machine and is not interruptible.

Finally, let $\check{x}(k) = \begin{bmatrix} x_{j,1}(k) & \ldots & x_{j,p_j}(k) \end{bmatrix}^T$ be the vector with all starting times of the operations of job j. This will give us the following inequalities for all $j = 1, \ldots, M$:

$$x_{j,m}(k) \geq x_{j,l}(k) + \tau_{j,l}(k), \text{ with } m > l,\ m, l \in \{1, \ldots, p_j\}.$$

In max-plus matrix notation this can be written as

$$\begin{bmatrix} x_{j,1}(k) \\ x_{j,2}(k) \\ \vdots \\ x_{j,p_j}(k) \end{bmatrix} \geq \begin{bmatrix} \varepsilon & \varepsilon & \ldots & \varepsilon \\ \tau_{j,1}(k) & \varepsilon & & \varepsilon \\ \vdots & \ddots & \ddots & \vdots \\ \varepsilon & \ldots & \tau_{j,p_j-1}(k) & \varepsilon \end{bmatrix} \otimes \begin{bmatrix} x_{j,1}(k) \\ x_{j,2}(k) \\ \vdots \\ x_{j,p_j}(k) \end{bmatrix}$$

or in short notation

$$\check{x}_j(k) \geq \check{A}_{\text{job},j}(k) \otimes \check{x}_j(k)$$

If we have M jobs, we can collect all starting times in one state vector $x(k)$ and we obtain:

$$\begin{aligned} x(k) &= \begin{bmatrix} \check{x}_1(k) \\ \check{x}_2(k) \\ \vdots \\ \check{x}_M(k) \end{bmatrix} \\ &\geq \begin{bmatrix} \check{A}_{\text{job},1}(k) & \mathcal{E} & \ldots & \mathcal{E} \\ \mathcal{E} & \check{A}_{\text{job},2}(k) & & \mathcal{E} \\ \vdots & & \ddots & \vdots \\ \mathcal{E} & \ldots & \ldots & \check{A}_{\text{job},M}(k) \end{bmatrix} \otimes \begin{bmatrix} \check{x}_1(k) \\ \check{x}_2(k) \\ \vdots \\ \check{x}_M(k) \end{bmatrix} \\ &\geq A_{\text{job}}(k) \otimes x(k) \end{aligned}$$

In many applications jobs are not finished within one cycle, but need multiple cycles to complete. The state equation is then given by

$$x(k) \geq A_{\text{job}}^{(0)}(k) \otimes x(k) \oplus A_{\text{job}}^{(-1)}(k) \otimes x(k-1) \qquad (1)$$

Often there are alternative routes available for the jobs. Alternative routes may result in the same 'product' (e.g. various machines in production line may execute the same operation) and sometimes the route may be changed to make another 'product'.

Let there be L alternative sets of routes for this system, then for each set of routes we can define the matrices $A_{\text{job},\ell}^{(0)}$ and $A_{\text{job},\ell}^{(-1)}$ for $\ell = 1, \ldots, L$. Let us now define a set of max-plus binary variables $(w_1(k), \ldots, w_L(k))$ such that if we have the ℓth set of alternative routes for the system in

cycle k, we find $w_\ell(k) = 0$ and $w_j(k) = \varepsilon$ for all $j \neq \ell$. Now the job-system matrices can be written as

$$A^{(\mu)}_{\text{job}}(w(k), k) \geq \bigoplus_{\ell=1}^{L} w_\ell \otimes A^{(\mu)}_{\text{job},\ell}(k), \tag{2}$$

Now let $2^{m-1} < L \leq 2^m$. Then we can parameterize the variables $(w_1, \ldots, w_L)$ by m max-plus binary variables $(\omega_1, \ldots, \omega_m)$ and its adjoint values $(\bar{\omega}_1, \ldots, \bar{\omega}_m)$. For example if $L = 8$ we can parameterize $w_1, \ldots, w_8$ as follows:

$$\left[\begin{array}{l} w_1(k) = \bar{\omega}_1(k) \otimes \bar{\omega}_2(k) \otimes \bar{\omega}_3(k) \\ w_2(k) = \bar{\omega}_1(k) \otimes \bar{\omega}_2(k) \otimes \omega_3(k) \\ w_3(k) = \bar{\omega}_1(k) \otimes \omega_2(k) \otimes \bar{\omega}_3(k) \\ w_4(k) = \bar{\omega}_1(k) \otimes \omega_2(k) \otimes \omega_3(k) \\ w_5(k) = \omega_1(k) \otimes \bar{\omega}_2(k) \otimes \bar{\omega}_3(k) \\ w_6(k) = \omega_1(k) \otimes \bar{\omega}_2(k) \otimes \omega_3(k) \\ w_7(k) = \omega_1(k) \otimes \omega_2(k) \otimes \bar{\omega}_3(k) \\ w_8(k) = \omega_1(k) \otimes \omega_2(k) \otimes \omega_3(k) \end{array}\right. \tag{3}$$

In general we can define a max-plus multiplicative function f such that $w_i(k) = f_i(\omega(k))$. By substitution of $w(k) = f(\omega(k))$ in (2) the number of variables can be reduced. Instead of $A^{(\mu)}_{\text{job}}(w(k), k)$ we will use the notation $A^{(\mu)}_{\text{job}}(\omega(k), k)$.

Remark: If L is not an exact power of 2, there will be more permutations of $\omega_i(k)$ and $\bar{\omega}_i(k)$ than necessary. In that case we can either introduce constraints on $\omega_i(k)$ and $\bar{\omega}_i(k)$ to describe the allowed set.

Ordering operations on resources in MPL systems

Consider a system with n operations, divided over N resources. Further let the system allow L alternative routes, and let $P_\ell \in {\mathbb{B}_\varepsilon}^{n \times n}$, $\ell \in \{1, \ldots, L\}$, be a matrix with max-plus binary entries, where $[P_\ell]_{i,j} = 0$ if operation i and operation j are executed on the same resource, and $[P_\ell]_{i,j} = \varepsilon$ if operation i and operation j are executed on different resources. The matrix $P(\omega(k))$ for selection of the resources can now be expressed as follows:

$$P(\omega(k)) = \bigoplus_{\ell=1}^{L} f_\ell(\omega(k)) \otimes P_\ell$$

Let $H(k)$ be a separation time matrix, where $H_{i,j}(k) \neq \varepsilon$ is the separation time between operations i and j if they may be scheduled on the same resource and $H_{i,j}(k) = \varepsilon$ if operations i and j can never be scheduled on the same resource. Finally let $V^{(\mu)}(k)$ be order decision matrices with max-plus binary entries, where $[V^{(\mu)}(k)]_{i,j} = 0$ if operation i in cycle k is scheduled after operation j in cycle $k+\mu$, and $[V^{(\mu)}(k)]_{i,j} = \varepsilon$ if operation i in cycle k is scheduled before operation j in cycle $k+\mu$. Define $v^{(\mu)}(k) = \text{vec}(V^{(\mu)}(k)$, then we use the notation $\text{vec}(V^{(\mu)}(k)) = V(v^{(\mu)}(k))$.

Finally define the ordering matrices

$$A^{(\mu)}_{\text{ord}}(\omega(k), v^{(\mu)}(k), k) = P(\omega(k)) \odot V(v^{(\mu)}(k)) \odot H(k) \tag{4}$$

Now the operation ordering constraints in the system can be formulated as follows:

$$x(k) \geq \bigoplus_{\mu=\mu_{\min}}^{\mu_{\max}} A^{(\mu)}_{\text{ord}}(\omega(k), v^{(\mu)}(k), k) \otimes x(k-\mu) \tag{5}$$

where $\mu_{\min}$ and $\mu_{\max}$ are the minimum and maximum value for μ, respectively. The terms with $\mu > 0$ are due to the fact that in some circumstances it may happen that we want an operation in cycle k to be scheduled behind an operation in cycle $k + \mu$ (see e.g. [28]).

Note that certain values of $v^{(\mu)}(k)$ will lead to an infeasible schedule because of cycles in the ordering[1]. To avoid these infeasible schedules we can define the set $\mathcal{V}$ with all feasible values $v^{(\mu)}(k)$. To reduce the number of parameters needed we can make use of a binary decision tree [1]. For scheduling p operations on one resource we have $p!$ possible permutations. This is also the maximum number of elements in the set $\mathcal{V}$. Let $m = \lceil \log_2 p! \rceil$ then we need m binary parameters $\gamma^{(\mu)}(k) = \left[\begin{array}{ccc} \gamma^{(\mu)}_1(k) & \ldots & \gamma^{(\mu)}_m(k) \end{array}\right]^T$ to model all possible allowed values $v^{(\mu)}(k)$. Using a binary decision tree we can define max-plus function $f : (\mathbb{B}_\varepsilon)^{p^2} \times (\mathbb{B}_\varepsilon)^m$ such that $v^{(\mu)}_i(k) = f^{(\mu)}_i((\gamma^{(\mu)}(k))$.

Example 1: Consider an example for $p = 3$, where we have a matrix

$$V(v^{(0)}(k)) = \left[\begin{array}{ccc} v^{(\mu)}_1 & v^{(0)}_2 & v^{(0)}_3 \\ v^{(0)}_4 & v^{(0)}_5 & v^{(0)}_6 \\ v^{(0)}_7 & v^{(0)}_8 & v^{(0)}_9 \end{array}\right]$$

with 9 variables. First note that all diagonal elements are redundant[2] and so $v^{(0)}_1 = v^{(0)}_5 = v^{(0)}_9 = \varepsilon$ are chosen fixed. We have $p! = 6$ feasible combinations of $(v^{(0)}_2, v^{(0)}_3, v^{(0)}_4, v^{(0)}_6, v^{(0)}_7, v^{(0)}_8)$ and so $m = \lceil \log_2 3! \rceil = 3$.

TABLE I

MAX-PLUS TRUTH TABLE AND CORRESPONDING PERMUTATION

	$\gamma^{(0)}_1$	$\gamma^{(0)}_2$	$\gamma^{(0)}_3$	$v^{(0)}_2$	$v^{(0)}_3$	$v^{(0)}_4$	$v^{(0)}_6$	$v^{(0)}_7$	$v^{(0)}_8$	perm
0	ε	ε	ε	ε	ε	0	ε	0	0	1 2 3
1	ε	ε	0	ε	ε	0	0	0	ε	1 3 2
2	ε	0	ε	0	ε	ε	ε	0	0	2 1 3
3	ε	0	0	0	0	ε	ε	ε	0	2 3 1
4	0	ε	ε	0	0	ε	0	ε	ε	3 1 2
5	0	ε	0	ε	0	0	0	ε	ε	3 2 1
6	0	0	ε	0	0	0	0	0	0	dummy
7	0	0	0	0	0	0	0	0	0	dummy

Table I shows the permutations (last column) with the corresponding values of the entries $v^{(0)}$ of the matrix $V^{(0)}$. The column 'perm' gives the ordering of the operations. From the table we can see that

$$v^{(0)}_2 = \gamma^{(0)}_2 \oplus (\gamma^{(0)}_1 \otimes \bar{\gamma}^{(0)}_3)$$
$$v^{(0)}_3 = \gamma^{(0)}_1 \oplus (\gamma^{(0)}_2 \otimes \gamma^{(0)}_3)$$

[1]An infeasible ordering is for example in the case of three starting times of operations x_1, x_2, x_3 we choose an ordering $x_1 > x_2$, $x_2 > x_3$, and $x_3 > x_1$.

[2]Note that for positive values of the diagonal entries we obtain a contradiction $x_i(k) > x_i(k)$, $i = 1, 2, 3$, and for zero or negative values of the diagonal entries we obtain a triviality $x_i(k) = x_i(k)$, $i = 1, 2, 3$.

$$
\begin{aligned}
v_4^{(0)} &= (\bar{\gamma}_1^{(0)} \otimes \bar{\gamma}_2^{(0)}) \oplus (\gamma_1^{(0)} \otimes \gamma_3^{(0)}) \oplus (\gamma_1^{(0)} \otimes \gamma_2^{(0)}) \\
v_6^{(0)} &= \gamma_1^{(0)} \oplus (\bar{\gamma}_2^{(0)} \otimes \gamma_3^{(0)}) \\
v_7^{(0)} &= (\bar{\gamma}_1^{(0)} \otimes \bar{\gamma}_2^{(0)}) \oplus (\bar{\gamma}_1^{(0)} \otimes \bar{\gamma}_3^{(0)}) \oplus (\gamma_1^{(0)} \otimes \gamma_2^{(0)}) \\
v_8^{(0)} &= \gamma_2^{(0)} \oplus (\bar{\gamma}_1^{(0)} \otimes \bar{\gamma}_3^{(0)})
\end{aligned}
$$

A way to derive concise description of the function f can be found in [1].

Subsequently we can substitute $v^{(\mu)}(k) = f^{(\mu)}((\gamma^{(\mu)}(k)))$ into (4) and we obtain

$$
A_{\text{ord}}^{(\mu)}(\omega(k), \gamma^{(\mu)}(k), k) = P(\omega(k)) \odot V(f(\gamma^{(\mu)}(k))) \odot H(k) \tag{6}
$$

Note that for higher values of p the difference between p^2 and m grows very rapidly. Using the variables $\gamma^{(\mu)}(k)$ has two important advantages: we need less parameters and there are no infeasible choices for $v^{(\mu)}(k)$. This is very beneficial if we want to optimize the schedule later on.

Synchronization of operations in MPL systems

Synchronization occurs when a specific operation can only start when a specific operation of another job has finished. In general we can define a number of synchronization modes $\ell = 1, \ldots, L_{\text{syn}}$, where for every mode we obtain a system matrix

$$
[A_{\text{syn},\ell}^{(\mu)}(k)]_{ij} = \begin{cases} \\ 0 \\ \\ \varepsilon \end{cases} \begin{array}{l} \text{if operation } j \text{ in cycle } k \text{ is to be} \\ \text{scheduled behind operation } i \text{ in cycle} \\ k+\mu. \\ \text{elsewhere} \end{array}
$$

Now the operation synchronization constraints in the system can be formulated as follows:

$$
x(k) \geq \bigoplus_{\mu=1}^{\mu_{\max}} A_{\text{syn}}^{(\mu)}(s(k), k) \otimes x(k-\mu), \tag{7}
$$

where

$$
A_{\text{syn}}^{(\mu)}(s(k), k) = \bigoplus_{\ell=0}^{L_{\text{syn}}} s_\ell(k) \otimes A_{\text{syn},\ell}^{(\mu)}(k), \tag{8}
$$

where $s(k) \in \mathbb{B}_\varepsilon^{L_{\text{syn}}}$ are max-plus binary variables for scheduling the synchronizations. Synchronizations may be coupled and appear in groups (e.g. the synchronization of legs in a legged robot [17]), but can also be an isolated phenomenon (e.g. the synchronization of two trains on a platform to give passengers the chance to change trains [8]). If there is a coupling between the synchronization variables, this coupling can be parameterized in a way similar to the ones in (3).

Reference signal

Some discrete event systems work with a predefined schedule that gives a lower bound for the starting time of the operations in the system (e.g. in a railway system we have a timetable with the departure time of trains). Let $r_j(k)$ be the starting time for operation i according to the given time schedule. To guarantees a lower bound $r_i(k)$ on operation i we introduce the constraint

$$
x(k) \geq r(k). \tag{9}
$$

Overall MPL system

We have derived four conditions (1), (5), (7), and (9) for $x(k)$. We also have a set of scheduling decision variables from

- Routing: $\omega(k)$.
- Ordering: $\gamma^{(\mu)}(k)$, $\mu = \mu_{\min}, \ldots, \mu_{\max}$.
- Synchronization: $s(k)$, $\mu = 1, \ldots, \mu_{\max}$.

If we now stack all decision variables into one vector

$$
u(k) = \begin{bmatrix} \omega(k) \\ \gamma^{(\mu_{\min})}(k) \\ \vdots \\ \gamma^{(\mu_{\min})}(k) \\ s(k) \end{bmatrix} \in (\mathbb{B}_\varepsilon)^{L_{\text{tot}}}
$$

where L_{tot} is the total number of scheduling variables, then we can write our scheduling model as follows

$$
x(k) = \bigoplus_{\mu=\mu_{\min}}^{\mu^{\max}} A^{(\mu)}(u(k), k) \otimes x(k-\mu) \oplus r(k) \tag{10}
$$

where

$$
\begin{aligned}
&A^{(\mu)}(u(k), k) = \\
&\quad = A_{\text{job}}^{(\mu)}(\omega(k), k) \oplus A_{\text{ord}}^{(\mu)}(\omega(k), v^{(\mu)}(k), k) \oplus A_{\text{syn}}^{(\mu)}(s(k)) \\
&\quad = \bigoplus_{\ell=1}^{L_{\text{tot}}} u_\ell(k) \otimes A_{\text{tot},\ell}^{(\mu)}(k)
\end{aligned}
$$

Note that by choosing a specific control vector $u(k)$ the system switches between different modes of operation. Such a system is called a switching max-plus linear system [29].

III. MODEL PREDICTIVE SCHEDULING

In this paper we will use a *model predictive scheduling* strategy. With a receding horizon principle the schedule for the complete task is not calculated at once, but in several iterations. In every iteration the schedule is calculated for only the jobs in the nearest future, where only these few future jobs and the necessary past jobs are taken into account, instead of all jobs in the scheduling task. Two reasons to use the model predictive scheduling method:

- The scheduling task may contain many jobs. The computation time of the optimal solution increases as the number of scheduling variables increases. A too long computation time can cancel out the time gained by optimizing the schedule, or even deteriorate the total solution. The negative impact of the computation time can be avoided by using the receding horizon principle, which is one of the main characteristics of MPC.
- We aim for reactive operational scheduling, which means that based on observations of the system's behavior we can reschedule (reroute, resynchronize, and

reorder) the jobs of the system to optimize the performance. This means that we have to perform the optimization in real-time based on measurements of the actual state and knowledge of delayed operations (possibly with estimation of the remaining processing times).

Let t be the present time instant, and we like to compute the optimal future control actions. Further, define the cycle $k-1$ as the last cycle at which the state $x(k-1)$ is completely known, so

$$k = \arg\max_{\kappa} \Big\{ \kappa | x_i(\kappa - \ell) \leq t, \forall \ell \geq 1,\ i \in \{1, \ldots, n\} \Big\} \quad (11)$$

Hence, all states $x(k-\ell)$ for $\ell \geq 1$ with the starting and finished times for the cycle $k-\ell$ are completely known at time t, and thus the control actions $u(k-\ell)$ are also available.

This means that some components of the states $x(k+j)$, $j \geq 0$ may be known and some parts are unknown at time t. At each time t we can define matrices $S_x(k+j,t)$ and $S_u(k+j,t)$ such

$$\begin{aligned} S_x(k+j,t) \otimes x(k+j) &= x_{\text{past}}(k+j,t) \leq t \\ S_u(k+j,t) \otimes u(k+j) &= u_{\text{past}}(k+j,t) \end{aligned}$$

where $x_{\text{past}}(k+j,t)$ and $u_{\text{past}}(k+j,t)$ are the parts of x and u that are known at time t. Further measurements of processing times of past operations and information about future delays leads to time-varying system matrices $A^{(\mu)}(u(k+j,t),t)$.

We can now formulate the *Model Predictive Scheduling problem*. To select the optimal set of possible future control actions, we define the following optimal control problem at cycle k and time instant t:

$$\min_{u(k+j,t),\ j=0,\ldots,N_{\text{p}}-1} J(k,t) \quad (12)$$

$$x(k+j,k+j,t) = \bigoplus_{\mu=\mu_{\min}}^{\max} A^{(\mu)}(u(k+j,t),k+j,t) \otimes x(k+j-\mu,t) \oplus r(k+j) \quad (13)$$

$$A^{(\mu)}(u(k+j,t),k+j,t) = \bigoplus_{\ell=1}^{L_{\text{tot}}} u_\ell(k+j,t) \otimes A^{(\mu)}_{\text{tot},\ell}(,k+j,t) \quad (14)$$

$$S_x(k+j,t) \otimes x(k+j) = x_{\text{past}}(k+j,t) \quad (15)$$

$$S_u(k+j,t) \otimes u(k+j) = u_{\text{past}}(k+j,t) \quad (16)$$

where N_{p} is the prediction horizon, and $J(k,t)$ is the performance index in cycle k at time t.

The performance index $J(k,t)$ is usually given by

$$J(k,t) = \max_{i=1,\ldots,n} \alpha\, x_i(k+N_{\text{p}},t) + \sum_{j=0}^{N_{\text{p}}-1} \sum_{i=1}^{n} \sigma_{j,i}\, x_i(k+j,t) + \sum_{l=1}^{L_{\text{tot}}} \rho_{j,l}\, \eta_l(k+j,t). \quad (17)$$

where

$$\eta_l(k+j,t) = ((u_l(k+j,t) \otimes 1) \oplus 0) = \begin{cases} 0 & \text{for } u_l(k+j,t) = \varepsilon \\ 1 & \text{for } u_l(k+j,t) = 0 \end{cases} \quad (18)$$

is a conventional binary variable. Further $\sigma_{j,i}$, $\rho_{j,l}$ are weighting scalars. The first term of (17) is the makespan over the prediction horizon (that is the total production length over the next N_{p} jobs), the second term is related to the weighted sum of all predicted starting times, and the third term denotes the penalty for all changes in ordering or synchronization during cycle $k+j$.

Often we like to minimize the global makespan, that is the total length of the schedule. Let N_{tot} be the number of job cycles to be scheduled. Then the aim will be to minimize $\max_i x_i(k+N_{tot},t)$. If N_{tot} is very big it is usually better to choose a prediction horizon $N_{\text{p}} \ll N_{\text{tot}}$, and the criterion will be to minimize (17) where $\alpha = 1$ and $0 \leq \sigma_i \ll 1$, $i = 1, \ldots, N_{\text{p}} - 1$. A major advantage of a small prediction horizon N_{p} is that the computational complexity of the optimization problem is drastically reduced. In other cases we like to minimize the sum of delays with respect to a reference signal (i.e. $\Big(x_i(k+j,t) - r_i(k+j)\Big)$). We then have $\alpha = 0$ and $\sigma_i = 1,\ \forall\, i$. (Note that skipping the reference signal from the performance index does not change the optimal schedule and so we can apply (17) for this case).

In principle we have all elements to solve the optimal control problem (12)-(16).

IV. THE MIXED-INTEGER LINEAR PROGRAMMING PROBLEM

The model predictive scheduling problem (12)-(16) can be recast into a mixed-integer linear programming problem as follows. The scheduling parameters in the model predictive scheduling problem are either zero or infinity. For the actual numerical implementation the infinite value ε cannot be used. The first step will therefore be to replace the max-plus binary variables by conventional binary variables. We use the following approximation

$$u_i(k,t) = \beta\,(1 - \eta_i(k,t))$$

where $\beta \ll 0$ is a very large (in absolute value) negative number and $\eta(k,t) \in \{0,1\}^{L_{\text{tot}}}$ is a conventional binary variable. The adjoint of $u_i(k,t)$ can be approximated by

$$\bar{u}_i(k,t) = \beta\, \eta_i(k,t)$$

Consider constraint (13). This can be written as a set of constraints:

$$[x(k+j,t)]_i \geq [A_\mu(\beta(1-\eta(k+j,t)),t)]_{il} + [x(k+j-\mu,t)]_l \quad (19)$$

$$[x(k+j,t)]_i \geq [r(k+j)]_i \quad (20)$$

for $i = 1, \ldots, n$, $l = 1, \ldots, n$, $j = 0, \ldots, N_{\text{p}} - 1$, $\mu = \mu_{\min}, \ldots, \mu_{\max}$. In the absence of a reference signal, we can

drop constraint (20). Define the vectors

$$\tilde{x}(k,t)=\begin{bmatrix} x(k,t) \\ \vdots \\ x(k+N_{\mathrm{p}}-1,t) \end{bmatrix}, \quad \tilde{\eta}(k,t)=\begin{bmatrix} \eta(k,t) \\ \vdots \\ \eta(k+N_{\mathrm{p}}-1,t) \end{bmatrix}$$

$$\tilde{r}(k)=\begin{bmatrix} r(k) \\ \vdots \\ r(k+N_{\mathrm{p}}-1) \end{bmatrix}$$

The cost criterion $J(k,t)$ is linear in $\tilde{x}(k,t)$, $\tilde{\eta}(k,t)$, $\tilde{r}(k)$, and so there exists a vector $c = \begin{bmatrix} c_1^T & c_2^T & c_3^T \end{bmatrix}$ such that

$$J(k) = c_1^T \tilde{x}(k,t) + c_2^T \tilde{\eta}(k,t) - c_3^T \tilde{r}(k,t) \qquad (21)$$

Now the model predictive scheduling problem is recast into an Mixed integer programming problem of minimizing criterion (21) subject to constraints (19)-(20). In general, mixed integer linear programming problems are NP-hard [26]. In practice, this means that the computational complexity of a mixed integer linear programming problem grows exponentially with the number of integer values in the problem. Nevertheless, there exist fast and reliable solvers (e.g. CPLEX, Xpres, [3]) for these problems.

V. DISCUSSION

In this paper we have discussed a general framework for model predictive scheduling of semi-cyclic discrete event systems. We have introduced a systematic way to model the main scheduling steps: routing, ordering, and synchronization. A max-plus linear model has been derived with scheduling parameters for each scheduling step. The system matrix is max-plus linear in the scheduling parameters and a model predictive scheduling problem has been formulated. This problem can be recast into a mixed integer linear programming problem.

In future research we also like to consider large-scale examples where MILP algorithms might fail. In that case we will look at heuristic optimization techniques such as optimistic optimization [22] or ordinal optimization [15].

REFERENCES

[1] S.B. Akers. Binary decision diagrams. *IEEE Transactions on Computers*, C-27(6):509–516, 1978.

[2] M. Alirezaei, T.J.J. van den Boom, and R. Babuška. Max-plus algebra for optimal scheduling of multiple sheets in a printer. In *American Control Conference 2012*, pages 1973–1978, Montreal, Canada, June 2012.

[3] A. Atamtürk and M.W.P. Savelsbergh. Integer-programming software systems. *Annals of Operations Research*, 140(1):67–124, November 2005.

[4] F. Baccelli, G. Cohen, G.J. Olsder, and J.P. Quadrat. *Synchronization and Linearity*. Wiley, New York, 1992.

[5] J. Blazewicz, K.H. Ecker, E. Pesch, G. Schmidt, and J. Weglarz. *Scheduling Computer and Manufacturing Processes, second ed.* Springer, Berlin, 2001.

[6] J.L. Bouquard and C. Lenté. Two-machine flow shop scheduling problems with minimal and maximal delays. *4OR: A Quarterly Journal of Operations Research*, 4:15–28, 2006.

[7] J.L. Bouquard, C. Lenté, and J.C. Billaut. Application of an optimization problem in max-plus algebra to scheduling problems. *Discrete Applied Mathematics*, 154(15):2064–2079, 2006.

[8] J.G. Braker. Algorithms and applications in timed discrete event systems. PhD thesis, Delft University of Technology, Delft, 1993.

[9] P. Brucker. *Scheduling Algorithms, third ed.* Springer, Berlin, 2001.

[10] P. Brucker and T. Kampmeyer. A general model for cyclic machine scheduling problems. *Discrete Applied Mathematics*, 156(13):2561–2572, 2008.

[11] R.A. Cuninghame-Green. *Minimax Algebra, volume 166 Lecture Notes in Economics and Mathematical Systems*. Springer-Verlag, Berlin, 1979.

[12] C. Hanen and A. Munier. Scheduling theory and its applications. Chapter Cyclic Scheduling on Parallel Processors: An Overview. Wiley, 1995.

[13] B. Heidergott and R. de Vries. Towards a (max,+) control theory for public transportation networks. *Discrete Event Systems: Theory and Applications*, 11(4):371–398, 2001.

[14] B. Heidergott, G.J. Olsder, and J.Ṽan der Woude. *Max Plus at Work: Modeling and Analysis of Synchronized Systems*. Princeton University Press, Princeton, 2006.

[15] Y.C. Ho, R. Sreenivas, and P. Vakili. Ordinal optimization of DEDS. *Discrete Event Dynamic Systems*, 2(1):61–88, 1992.

[16] S.M. Johnson. Optimal two- and three-stage production schedule with setup times included. *Naval Research Logistics Quarterly*, 1(1):61–68, 1954.

[17] B. Kersbergen, G.A.D. Lopes, T. van den Boom, B. De Schutter, and R. Babuška. Optimal gait switching for legged locomotion. In *International Conference on Intelligent Robots and Systems (IROS-2011)*, pages 2729–2734, San Francisco, USA, September 2011.

[18] G.A.D. Lopes, B. De Schutter, and T.J.J. van den Boom. On the synchronization of cyclic discrete-event systems. In *Proceedings of the 51st IEEE Conference on Decision and Control*, pages 5810–5815, Maui, Hawaii, December 2012.

[19] G.A.D. Lopes, B. Kersbergen, T. van den Boom, B. De Schutter, and R. Babuška. On the eigenstructure of a class of max-plus linear systems. In *Proceedings of the 2011 50th IEEE Conference on Decision and Control and European Control Conference (CDC-ECC)*, pages 1823–1828, Orlando, Florida, December 2011.

[20] G.A.D. Lopes, T.J.J. van den Boom, B. De Schutter, and R. Babuška. Modeling and control of legged locomotion via switching max-plus systems. In *Proceedings of the 10th International Workshop on Discrete Event Systems (WODES 2010)*, pages 392–397, Berlin, Germany, August–September 2010.

[21] A. Mascis and D. Pacciarelli. Job shop scheduling with blocking and no-wait constraints. *European Journal of Operations Research*, 143(3):498–517, 2002.

[22] R. Munos. Optimistic optimization of a deterministic function without the knowledge of its smoothness. In J. Shawe-Taylor, R.S. Zemel, P. Bartlett, F.C.N. Pereira, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems 24*, pages 783–791. 2011.

[23] M. Mutsaers, L. Özkan, and T. Backx. Scheduling of energy flows for parallel batch processes using max-plus systems. In *Proceedings of the 8th IFAC International Symposium on Advanced Control of Chemical Processes 2012*, pages 176–181, July 2012.

[24] A. Nambiar and R. Judd. Max-plus-based mathematical formulation for cyclic permutation flow-shops. *International Journal of Mathematical Modelling and Numerical Optimisation*, 2:85–97, 2011.

[25] M. Pinedo. *Scheduling: Theory, Algorithms and Systems, second ed.* Prentice-Hall, Englewood Cliffs, NJ, 2001.

[26] A. Schrijver. *Theory of Linear and Integer Programming*. John Wiley & Sons, Chichester, UK, 1986.

[27] K. Shedden. Analysis of cell-cycle gene expression in saccharomyces cerevisiae using microarrays and multiple synchronization methods. *Nucleic Acids Research*, 30(13):2920–2929, 2002.

[28] T. van den Boom, B. Kersbergen, and B. De Schutter. Structured modeling, analysis, and control of complex railway operations. In *51st IEEE Conference on Decision and Control (CDC 2012)*, pages 7366–7371, Maui, Hawaii, USA, December 10-13 2012.

[29] T.J.J. van den Boom and B. De Schutter. Modelling and control of discrete event systems using switching max-plus-linear systems. *Control Engineering Practice*, 14(10):1199–1211, October 2006.

[30] M. Yurdakul and N.G. Odrey. Development of a new dioid algebraic model for manufacturing with the scheduling decision making capability. *Robotics and Autonomous Systems*, pages 207–218, 2004.

[31] M. Zhou, F. DiCesare, and A. Desrochers. A hybrid methodology for synthesis of petri net models for manufacturing systems. *IEEE Tran. on Robotics and Automation*, 8(3):350–361, 1992.