Delft University of Technology

Delft Center for Systems and Control

Technical report 03-017

Adaptive model predictive control for max-plus-linear discrete event input-output systems*

T.J.J. van den Boom, B. De Schutter, G. Schullerus, and V. Krebs

If you want to cite this report, please use the following reference instead:
T.J.J. van den Boom, B. De Schutter, G. Schullerus, and V. Krebs, "Adaptive model predictive control for max-plus-linear discrete event input-output systems," *IEE Proceedings – Control Theory and Applications*, vol. 151, no. 3, pp. 339–346, May 2004. doi:10.1049/ip-cta:20040440

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.bartdeschutter.org/abs/03_017.html

Adaptive Model Predictive Control for max-plus-linear discrete event input-output systems

T.J.J. van den Boom[‡], B. De Schutter[‡], G. Schullerus[§] and V. Krebs[§]

[‡]Delft Center for Systems and Control,

Delft University of Technology, Mekelweg 2, 2628 CD Delft, The Netherlands

[§]Institut für Reglungs- und Steuerungssysteme,

Universität Karlsruhe (TH), Kaiserstr. 12, D-76131 Karlsruhe, Germany

email: {t.j.j.vandenboom,b.deschutter}@dcsc.tudelft.nl,

{schullerus,krebs}@irs.uni-karlsruhe.de

Abstract

Model predictive control (MPC) is a popular controller design technique in the process industry. Conventional MPC uses linear or nonlinear discrete-time models. Recently, we have extended MPC to a class of discrete event systems that can be described by a model that is "linear" in the max-plus algebra. In our previous work we have considered MPC for the time-invariant case. In this paper we consider an adaptive scheme for the time-varying case, based on parameter estimation of input-output models. In a simulation example we show that the combined parameter-estimation/MPC algorithm gives a good closed-loop behavior.

1 Introduction

Clarke *et al.* [5] and Mosca [21] demonstrate how predictive control can provide adaptive controllers. The predictive technique is seen as a tool to go beyond the conventional single-step-ahead adaptive control strategies. Model predictive control (MPC) [11, 16] is a proven technology for the control of multivariable systems in the presence of input and output constraints and is capable of tracking pre-scheduled reference signals. At each time instant the process model is updated, based on measured input and output data. On the basis of this model, predictions of the process signals over a specified horizon are made. A cost-criterion is formulated, reflecting the reference tracking error and the control effort. An optimization algorithm will be applied to compute a sequence of future control signals that minimizes the performance index subject to the given constraints. Predictive control uses the receding horizon principle. This means that after computation of the optimal control sequence, only the first control sample will be implemented, subsequently the horizon is shifted one sample and the parameter estimation and input optimization is restarted with new information of the measurements. The above derived controller is called an adaptive model predictive controller.

Usually adaptive MPC uses linear or nonlinear discrete-time models. However, the attractive features mentioned above have led us to extend the adaptive MPC scheme to discrete event systems. Typical examples of discrete event systems (DES) are flexible manufacturing systems, telecommunication networks, parallel processing systems, traffic control systems, and logistic systems. The class of DES essentially consists of man-made systems that contain a finite number of resources (such as machines, communications channels, or processors) that are shared by several users (such as product types, information packets, or jobs) all of which contribute to the achievement of some common goal (the assembly of products, the end-to-end transmission of a set of information packets, or a parallel computation) [1]. There exist many different modeling and analysis frameworks for DES such as Petri nets, finite state machines, automata, languages, process algebra, computer models, etc. [4, 13]. In this paper we consider the class of DES with synchronization but no concurrency or choice. Such systems can be modeled using the operations maximization (corresponding to synchronization: a new operation starts as soon as all preceding operations have been finished) and addition (corresponding to durations: the finishing time of an operation equals the starting time plus the duration). This leads to a description that is "linear" in the max-plus algebra [1, 7]. Such DES are therefore called max-plus-linear (MPL) DES. So typical examples are serial production lines, production systems with a fixed routing schedule, queuing systems, telecommunication networks, and railway networks.

Note that although the class of MPL DES is a small (but relevant) subclass of the general DES, one of its main advantages is that having an analytic MPL model allows us to derive some properties of the system (such as the steady state behavior) fairly easily, and to develop efficient model-based control design methods for the MPL DES [1, 2, 6, 8, 12, 14, 19, 20]. More specifically, in [8] we have derived an MPC controller for this framework and we have also shown that under quite general conditions the resulting MPC optimization problem is a convex optimization problem. This paper describes an adaptive MPC methodology for slowly time-varying MPL systems using an input-output model. An input-output setting is used because in many applications only input and output measurements are available. In this paper we consider the noise-free case.

Note that the supervisory control framework for DES[1] introduced by Wonham and Ramadge [22, 23] also provides a feedback control structure. However, the main difference between the supervisory control framework and MPC is that in supervisory control the control actions consist in *blocking* certain events (i.e., preventing them from occurring), whereas in MPC the control actions consist in *delaying* certain events (i.e., shifting their occurrence time with a finite amount of time).

[1]This supervisory control framework has been developed for untimed automata and later on also extended to timed automata, which are a superclass of the MPL DES.

The two main ingredients of the adaptive predictive controller are the identification module and optimal control law module. We will discuss these modules in the Sections 3 and 4, respectively, and we give the final adaptive MPC algorithm in Section 5. Finally, Section 6 gives a worked example and a comparison with conventional methods. We start with the introduction of the max-plus algebra and the concept of MPL input-output systems in Section 2.

2 Max-plus-linear input-output systems

In this section we define the class of MPL input-output systems. For this purpose we will first give the basic definition of the max-plus algebra and min-plus algebra, and we present some results for max-plus polynomials.

Max-plus algebra

Define $\varepsilon = -\infty$ and $\mathbb{R}_\varepsilon = \mathbb{R} \cup \{\varepsilon\}$. The max-plus-algebraic addition ($\oplus$) and multiplication ($\otimes$) are defined as follows [1, 7]:

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

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

$$[A \oplus B]_{ij} = a_{ij} \oplus b_{ij} = \max(a_{ij}, b_{ij})$$
$$[A \otimes C]_{ij} = \bigoplus_{k=1}^{n} a_{ik} \otimes c_{kj} = \max_{k=1,\ldots,n} (a_{ik} + c_{kj})$$

for matrices $A, B \in \mathbb{R}_\varepsilon^{m \times n}$ and $C \in \mathbb{R}_\varepsilon^{n \times p}$.

Min-plus algebra

Define $\top = \infty$ and $\bar{\mathbb{R}} = \mathbb{R}_\varepsilon \cup \{\top\} = \mathbb{R} \cup \{\varepsilon, \top\}$. The min-plus-algebraic addition ($\oplus'$) and multiplication ($\otimes'$) are defined as follows [1, 7]:

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

for numbers $x, y \in \bar{\mathbb{R}}$. By definition $\varepsilon \otimes \top = \top \otimes \varepsilon = \varepsilon$ and $\varepsilon \otimes' \top = \top \otimes' \varepsilon = \top$. For matrices $A, B \in \bar{\mathbb{R}}^{m \times n}$ and $C \in \bar{\mathbb{R}}^{n \times p}$ we have

$$[A \oplus' B]_{ij} = a_{ij} \oplus' b_{ij} = \min(a_{ij}, b_{ij})$$
$$[A \otimes' C]_{ij} = {\bigoplus_{k=1}^{n}}' \, a_{ik} \otimes' c_{kj} = \min_{k=1,\ldots,n} (a_{ik} + c_{kj})$$

Max-plus polynomials

This section is based on Baccelli *et al.* [1]. Define the delay operator γ as

$$\gamma\, z(k) = z(k-1)$$

Now we can define the max-plus polynomial

$$P(\gamma) = p_0 \otimes \gamma^0 \oplus p_1 \otimes \gamma^1 \oplus \ldots \oplus p_n \otimes \gamma^n$$

then we obtain

$$\begin{aligned} P(\gamma)\, z(k) &= \Big(p_0 \otimes \gamma^0 \oplus p_1 \otimes \gamma^1 \oplus \ldots \oplus p_n \otimes \gamma^n\Big) z(k) \\ &= p_0 \otimes \gamma^0\, z(k) \oplus p_1 \otimes \gamma^1\, z(k) \oplus \ldots \oplus p_n \otimes \gamma^n\, z(k) \\ &= p_0 \otimes z(k) \oplus p_1 \otimes z(k-1) \oplus \ldots \oplus p_n \otimes z(k-n) \end{aligned}$$

Let P, Q and R be three max-plus polynomials:

$$\begin{aligned} P(\gamma) &= p_0 \otimes \gamma^0 \oplus p_1 \otimes \gamma^1 \oplus \ldots \oplus p_n \otimes \gamma^n \\ Q(\gamma) &= q_0 \otimes \gamma^0 \oplus q_1 \otimes \gamma^1 \oplus \ldots \oplus q_n \otimes \gamma^n \\ R(\gamma) &= r_0 \otimes \gamma^0 \oplus r_1 \otimes \gamma^1 \oplus \ldots \oplus r_m \otimes \gamma^m \end{aligned}$$

(if some monomial γ^i is missing in P or Q, this means that the corresponding coefficient is 'zero', that is, it is equal to ε). The max-plus product and max-plus sum for polynomials are defined as follows:

$$\begin{aligned} P(\gamma) \oplus Q(\gamma) &= p_0 \otimes \gamma^0 \oplus p_1 \otimes \gamma^1 \oplus \ldots \oplus p_n \otimes \gamma^n \oplus q_0 \otimes \gamma^0 \oplus q_1 \otimes \gamma^1 \oplus \ldots \oplus q_n \otimes \gamma^n \\ &= \bigoplus_{i=0}^{n} (p_i \oplus q_i) \otimes \gamma^i \\ P(\gamma) \otimes R(\gamma) &= \Big(p_0 \otimes \gamma^0 \oplus p_1 \otimes \gamma^1 \oplus \ldots \oplus p_n \otimes \gamma^n\Big) \otimes \Big(r_0 \otimes \gamma^0 \oplus r_1 \otimes \gamma^1 \oplus \ldots \oplus r_m \otimes \gamma^m\Big) \\ &= \bigoplus_{i=0}^{n} \bigoplus_{j=0}^{m} (p_i \otimes r_j) \otimes \gamma^{i+j} \end{aligned}$$

Let P, Q and R be three max-plus polynomials and z and w two signals, then we can observe the following properties of the max-plus polynomial expressions:

$$\begin{aligned} P(\gamma)\, z(k) \oplus Q(\gamma)\, z(k) &= \Big(P(\gamma) \oplus Q(\gamma)\Big)\, z(k) \\ P(\gamma)\, z(k) \oplus P(\gamma)\, w(k) &= P(\gamma)\Big(z(k) \oplus w(k)\Big) \\ P(\gamma)\Big(R(\gamma) z(k)\Big) &= \Big(P(\gamma) \otimes R(\gamma)\Big)\, z(k) \end{aligned}$$

Max-plus-linear Input-Output systems

In [8, 9] we have used a state-space setting to study DES in which there is synchronization but no concurrency. In this paper we will consider these systems in an input-output setting. Our motivation behind this is that in practice only input and output signals are available, and the input-output form gives a compact description of the system. Consider systems that can be described by the input-output relation

$$\begin{aligned} y(k) &= a_1 \otimes y(k-1) \oplus a_2 \otimes y(k-2) \oplus \ldots \oplus a_n \otimes y(k-n) \oplus \\ & \quad b_0 \otimes u(k) \oplus b_1 \otimes u(k-1) \oplus \ldots \oplus b_m \otimes u(k-m) \end{aligned}$$

This can be rewritten in polynomial form as

$$y(k) = A(\gamma)y(k) \oplus B(\gamma)u(k) \tag{1}$$

where $A(\gamma)$ and $B(\gamma)$ are polynomial operators

$$\begin{aligned} A(\gamma) &= a_1 \otimes \gamma^1 \oplus a_2 \otimes \gamma^2 \oplus \ldots \oplus a_n \otimes \gamma^n \\ B(\gamma) &= b_0 \otimes \gamma^0 \oplus b_1 \otimes \gamma^1 \oplus \ldots \oplus b_m \otimes \gamma^m \end{aligned} \tag{2}$$

DES that can be described by this model will be called max-plus-linear input-output (MPLIO) systems. The index k is called the event counter. The input $u(k)$ contains the time instants at which the input events occur for the kth time, and the output $y(k)$ contains the time instants at which the output events occur for the kth time[2]. The entries of system polynomials $A(\gamma)$ and $B(\gamma)$ are varying in time due to slow changes in the system.

3 Identification of MPLIO systems

Consider the SISO[3] MPLIO model, described by the input-output relation (1) and (2). We assume that the "real" system is in the model set, and we denote the estimates of the input-output polynomials from (1) by $\hat{A}(\gamma)$ and $\hat{B}(\gamma)$. The prediction error $\xi(k)$ after the

[2]More specifically, for a manufacturing system, $u(k)$ contains the time instants at which the kth batch of raw material is fed to the system, and $y(k)$ the time instants at which the kth batch of finished product leaves the system.

[3]For sake of simplicity SISO systems are considered in this paper. However, most of the results are easily extended to the MIMO case. It will be important to know the ε-structure of the system, which is related to the layout and the internal connection between different subparts of the system (see, e.g., [1]). The problem is comparable to system identification of time-driven systems (appendix 4A, [15]), where we need to know the black-box multivariable model structure.

measurements of the kth event is then defined as

$$\begin{aligned}
\xi(k) &= y(k) - \Big(\hat{A}(\gamma)y(k) \oplus \hat{B}(\gamma)u(k)\Big) \\
&= y(k) - \Bigg(\underbrace{\left[\begin{array}{cccccc} \hat{a}_1 & \cdots & \hat{a}_n & \hat{b}_0 & \cdots & \hat{b}_m \end{array}\right]}_{\hat{\theta}} \otimes \underbrace{\left[\begin{array}{c} y(k-1) \\ \vdots \\ y(k-n) \\ u(k) \\ \vdots \\ u(k-m) \end{array}\right]}_{p(k)}\Bigg) \\
&= y(k) - \hat{\theta} \otimes p(k) \qquad (3)
\end{aligned}$$

The elements of the vector $\hat{\theta}$ are estimates of the system parameters. Considering k consecutive events, i.e. the measurement data of k process cycles, one obtains the prediction error matrix

$$\underbrace{\left[\begin{array}{ccc} \xi(k) & \cdots & \xi(1) \end{array}\right]}_{\Xi(k,1)} = \underbrace{\left[\begin{array}{ccc} y(k) & \cdots & y(1) \end{array}\right]}_{Y(k,1)} - \hat{\theta} \otimes \underbrace{\left[\begin{array}{ccc} p(k) & \cdots & p(1) \end{array}\right]}_{P(k,1)} \qquad (4)$$

or

$$\Xi(k,1) = Y(k,1) - \hat{\theta} \otimes P(k,1) \qquad (5)$$

As shown in [10] the solution that minimizes the prediction error $\Xi(k,1)$ corresponds to the greatest solution of the inequality

$$Y(k,1) \geq \hat{\theta} \otimes P(k,1) \qquad (6)$$

and can be computed using the min–plus–algebraic operators "$\oplus'$" and "$\otimes'$":

$$\begin{aligned}
\hat{\theta}_i &= \bigoplus_{j=1}^{k}{}' \, Y_j(k,1) \otimes' \big(-P_{ij}(k,1)\big) \qquad (7) \\
&= \bigoplus_{j=1}^{k}{}' \big(y(j) - p_i(j)\big) \qquad (8) \\
&= \min_{j=1,\ldots,k} \big(y(j) - p_i(j)\big) \qquad (9)
\end{aligned}$$

where $Y_j(k,1)$ denotes the j-th column of $Y(k,1)$. For this solution, the following properties hold [18]:

$$\begin{aligned}
\hat{\theta}_i &\geq \theta_i \qquad (10) \\
\hat{\theta} \otimes P(k) &= \theta \otimes P(k) \qquad (11)
\end{aligned}$$

such that the prediction error $\xi(j) = 0$, for $j = 1, \ldots, k$ due to (11). On the other hand, property (10) shows that in general, the parameters will be overestimated by this approach. This issue has been investigated in [24] and a condition for convergence of the estimated parameters to their true values was given. It essentially states that $\theta_i = \hat{\theta}_i$ if there exist $y(j)$ and $p(j)$ such that

$$y(j) = \theta_i \otimes p_i(j) \tag{12}$$

holds. Obviously, this condition can in general not be satisfied for MPLIO systems since the required trajectories cannot be achieved for all parameters using only one input signal. However, if no event trajectory that satisfies (12) for $\hat{\theta}_i$ exists, then the original system and the estimated system are equivalent with respect to θ_i since both systems will always lead to the same input–output behavior.

Hence, an initial estimate for the system parameters can be obtained based on k data points using (8). To track changing system parameters, an update of the estimates after each update of the output is necessary. A first possibility is the recursive evaluation of (8) as first proposed in [17] for the estimation of the system's impulse response. Thus,

$$\begin{aligned}
\hat{\theta}_i(k) &= \bigoplus_{j=1}^{k}{}' \big(y(j) - p_i(j)\big) && (13)\\
&= \bigoplus_{j=1}^{k-1}{}' \big(y(j) - p_i(j)\big) \oplus' \big(y(k) - p_i(k)\big) && (14)\\
&= \hat{\theta}_i(k-1) \oplus' \big(y(k) - p_i(k)\big) && (15)\\
&= \min\Big(\hat{\theta}_i(k-1), \big(y(k) - p_i(k)\big)\Big) && (16)
\end{aligned}$$

However, since $\oplus'$ corresponds to minimization, an update where $y(k) - p_i(k) > \hat{\theta}_i(k-1)$ will not have any influence on $\hat{\theta}_i(k)$. Thus, increasing parameter values will not be detected by this approach. As a possible solution to this problem the estimation can be carried out considering only the most recent N_e data points, and choosing

$$\underbrace{\begin{bmatrix} \xi(k) & \cdots & \xi(k-N_\mathrm{e}) \end{bmatrix}}_{\Xi(k,k-N_\mathrm{e})} = \underbrace{\begin{bmatrix} y(k) & \cdots & y(k-N_\mathrm{e}) \end{bmatrix}}_{Y(k,k-N_\mathrm{e})} - \hat{\theta} \otimes \underbrace{\begin{bmatrix} p(k) & \cdots & p(k-N_\mathrm{e}) \end{bmatrix}}_{P(k,k-N_\mathrm{e})} \tag{17}$$

However, using the reasoning above, it can be concluded that a change in a parameter θ_i that leads to measurements with $y(j) - p_i(j) > \hat{\theta}_i(k)$ may be detected only when all N_e data points considered in the estimation are influenced by this new parameter value.

Therefore, the algorithm used in the sequel is based on a different strategy. Assume, that the initial estimation $\hat{\theta}(0)$ was determined from the first N_e data points by (8). Similar to the conventional recursive estimation algorithms, the new estimate can be computed by adding the (weighted) difference between the new measurement and the measurement predicted by the model. This principle was used in [18] (though the similarity to the conventional recursive estimation was not pointed out) and will be applied for adaptive MPC with some modifications. Let $\hat{\theta}(k-1)$ be the estimate at the end of the $(k-1)$th cycle. If $\hat{\theta}(k-1)$ satisfies $y(k) = \hat{\theta}(k-1) \otimes p(k)$, we choose $\hat{\theta}(k) = \hat{\theta}(k-1)$. If not, then $\hat{\theta}(k)$ is obtained by the series

$$\begin{cases} \hat{\theta}^{(0)}(k) &= \hat{\theta}(k-1) \\ \hat{\theta}^{(\ell)}(k) &= \hat{\theta}^{(\ell-1)}(k) + \alpha \Delta^{(\ell-1)}(k) \quad \ell > 0 \end{cases} \tag{18}$$

where $0 < \alpha \leq 2$ is a weighting parameter and

$$\begin{aligned} \Delta^{(\ell-1)}(k) \;=\; & \Big[\Big(Y(k,k-N_\mathrm{e})) \otimes' \big(-P^T(k,k-N_\mathrm{e})\big)\Big) \\ & - \Big(\big(\hat{\theta}^{(l-1)}(k) \otimes P(k,k-N_\mathrm{e})\big) \otimes' (-P^T(k,k-N_\mathrm{e}))\Big)\Big] \end{aligned} \tag{19}$$

In [18] it is proven that for $\alpha = 1$ and $(Y(k,k-N_e) \otimes' (-P^T(k,k-N_e))) \otimes P(k,k-N_e) = Y(k,k-N_e)$ the iteration (18)-(19) will converge to a value that satisfies $y(k) = \hat{\theta}^{(\ell)}(k) \otimes p(k)$. In appendix A we show that we expect convergence for all $\alpha \in (0,2)$. The choice $\alpha \neq 1$ will slow down the convergence of the iterative procedure, and so we choose tuning parameter $\alpha = 1$.

Note that in contrast to [18], in this paper we use an MPLIO model rather than an impulse response model. The MPLIO description is more compact and so the estimation can be done using less information. Furthermore we have two new parameters: N_e, the number of past values of input and outputs, and the parameter α, which can be used to tune the convergence rate of the recursive estimation algorithm.

4 Model predictive control for MPLIO systems

In [8, 26] we have extended the MPC framework to MPL state-space models. Following the strategy for conventional discrete-time systems in an input-output setting [3, 5] we define a cost criterion $J(k)$ that reflects the output and input cost functions ($J_\mathrm{out}(k)$ and $J_\mathrm{in}(k)$, respectively) in the event period $[k, k+N_\mathrm{p}-1]$:

$$J(k) \;=\; J_\mathrm{out}(k) + \lambda J_\mathrm{in}(k) \tag{20}$$

in which

$$\begin{aligned} J_\mathrm{out}(k) &= \sum_{j=0}^{N_\mathrm{p}-1} \max\Big(\, \hat{y}(k+j|k) - r(k+j)\,,\, 0\Big) \\ J_\mathrm{in}(k) &= -\sum_{j=0}^{N_\mathrm{p}-1} u(k+j) \end{aligned}$$

where N_p is the prediction horizon and λ is a weighting parameter, $\hat{y}(k+j|k)$ is the prediction of the output signal $y(k+j)$, based on the knowledge at event step k, and $r(k)$

is the due date signal. The function $J_{\rm out}(k)$ reflects the due date-error and $J_{\rm in}(k)$ is used to penalize a large input-buffer. Other choices for cost function J are given in [8, 9].

In order to compute the optimal MPC input signal, we need to make predictions of the output signal.

Lemma 1 *Consider an MPLIO system (1)-(2). For any non-negative integer j, there exist polynomials*

$$
\begin{aligned}
C_j(\gamma) &= c_{1,j} \otimes \gamma^1 \oplus c_{2,j} \otimes \gamma^2 \oplus \ldots \oplus c_{n,j} \otimes \gamma^n & (21)\\
D_j(\gamma) &= d_{0,j} \otimes \gamma^0 \oplus d_{1,j} \otimes \gamma^1 \oplus \ldots \oplus d_{m-1,j} \otimes \gamma^{m-1} & (22)\\
F_j(\gamma) &= f_{0,j} \otimes \gamma^0 \oplus f_{1,j} \otimes \gamma^1 \oplus \ldots \oplus f_{j,j} \otimes \gamma^j & (23)
\end{aligned}
$$

such that

$$\hat{y}(k+j|k) = C_j(\gamma)y(k) \oplus D_j(\gamma)u(k-1) \oplus F_j(\gamma)u(k+j) \tag{24}$$

Proof:
We will use a proof by induction. Define

$$C_0(\gamma) = A(\gamma)\ , \quad D_0(\gamma) = b_1 \otimes \gamma^0 \oplus b_2 \otimes \gamma^1 \oplus \ldots \oplus b_m \otimes \gamma^{m-1}\ , \quad F_0(\gamma) = b_0 \otimes \gamma^0$$

and for $j < 0$

$$C_j(\gamma) = \gamma^{-j}\ , \quad D_j(\gamma) = \varepsilon\ , \quad F_j(\gamma) = \varepsilon$$

then (24) is satisfied for $j = 0$, because

$$y(k) = A(\gamma)y(k) \oplus B(\gamma)u(k) = C_0(\gamma)y(k) \oplus D_0(\gamma)u(k-1) \oplus F_0(\gamma)u(k)$$

and for $j < 0$, because with $i = -j > 0$ we find

$$y(k-i) = \gamma^i y(k)$$

Let for $j \in \mathbb{Z}$, $j > 0$, the polynomials $C_{j-\ell}(\gamma)$, $D_{j-\ell}(\gamma)$ and $F_{j-\ell}(\gamma)$ for all $\ell \in \mathbb{Z}$, $\ell > 0$ be such that

$$\hat{y}(k+j-\ell|k) = C_{j-\ell}(\gamma)y(k) \oplus D_{j-\ell}(\gamma)u(k-1) \oplus F_{j-\ell}(\gamma)u(k+j-\ell)$$

then

$$\begin{aligned}
\hat{y}(k+j|k) &= A(\gamma)\hat{y}(k+j|k) \oplus B(\gamma)u(k+j) \\
&= (a_1 \otimes \hat{y}(k+j-1|k) \oplus a_2 \otimes \hat{y}(k+j-2|k) \oplus \ldots \oplus a_n \otimes \hat{y}(k+j-n|k)) \\
&\quad \oplus B(\gamma)u(k+j) \\
&= \bigoplus_{\ell=1}^{n} a_\ell \otimes \Big(C_{j-\ell}(\gamma)y(k) \oplus D_{j-\ell}(\gamma)u(k-1) \oplus F_{j-\ell}(\gamma)u(k+j-\ell) \Big) \\
&\quad \oplus B(\gamma)u(k+j) \\
&= \bigoplus_{\ell=1}^{n} (a_\ell \otimes C_{j-\ell}(\gamma))y(k) \oplus \bigoplus_{\ell=1}^{n} (a_\ell \otimes D_{j-\ell}(\gamma))u(k-1) \\
&\quad \oplus \bigoplus_{\ell=1}^{n} (a_\ell \otimes F_{j-\ell}(\gamma) \otimes \gamma^\ell)u(k+j) \oplus B(\gamma)u(k+j) \\
&= \bigoplus_{\ell=1}^{n} (a_\ell \otimes C_{j-\ell}(\gamma))y(k) \oplus \bigoplus_{\ell=1}^{n} (a_\ell \otimes D_{j-\ell}(\gamma))u(k-1) \\
&\quad \oplus \left(\bigoplus_{\ell=1}^{n} (a_\ell \otimes F_{j-\ell}(\gamma) \otimes \gamma^\ell) \oplus B(\gamma) \right) u(k+j)
\end{aligned}$$

Now define two polynomials $B_j^{\rm fut}(\gamma)$ and $B_j^{\rm past}(\gamma)$ for $j < m$:

$$\begin{aligned}
B_j^{\rm fut}(\gamma) &= b_0 \otimes \gamma^0 \oplus b_1 \otimes \gamma^1 \oplus \ldots \oplus b_j \otimes \gamma^j \\
B_j^{\rm past}(\gamma) &= b_{j+1} \otimes \gamma^0 \oplus b_{j+2} \otimes \gamma^1 \oplus \ldots \oplus b_m \otimes \gamma^{m-i-1}
\end{aligned}$$

and for $j \geq m$:

$$B_j^{\rm fut}(\gamma) = B(\gamma) \qquad B_j^{\rm past}(\gamma) = \varepsilon$$

Then we find for all $j \in \mathbb{Z}$, $j > 0$:

$$\begin{aligned}
B(\gamma)u(k+j) &= B_j^{\rm fut}(\gamma)u(k+j) \oplus B_j^{\rm past}(\gamma) \otimes \gamma^{j+1}u(k+j) \\
&= B_j^{\rm fut}(\gamma)u(k+j) \oplus B_j^{\rm past}(\gamma)u(k-1)
\end{aligned}$$

and so

$$\begin{aligned}
\hat{y}(k+j|k) &= \bigoplus_{\ell=1}^{n} (a_\ell \otimes C_{j-\ell}(\gamma))y(k) \oplus \bigoplus_{\ell=1}^{n} (a_\ell \otimes D_{j-\ell}(\gamma))u(k-1) \\
&\quad \oplus \left(\bigoplus_{\ell=1}^{n} (a_\ell \otimes F_{j-\ell}(\gamma) \otimes \gamma^\ell) \oplus B_j^{\rm fut}(\gamma) \right) u(k+j) \oplus B_j^{\rm past}(\gamma)u(k-1) \\
&= C_j(\gamma)y(k) \oplus D_j(\gamma)u(k-1) \oplus F_j(\gamma)u(k+j)
\end{aligned}$$

where

$$
\begin{aligned}
C_j(\gamma) &= \bigoplus_{\ell=1}^{n} (a_\ell \otimes C_{j-\ell}(\gamma)) \\
D_j(\gamma) &= B_j^{\mathrm{past}}(\gamma) \oplus \bigoplus_{\ell=1}^{n} (a_\ell \otimes D_{j-\ell}(\gamma)) \\
F_j(\gamma) &= B_j^{\mathrm{fut}}(\gamma) \oplus \bigoplus_{\ell=1}^{n} (a_\ell \otimes F_{j-\ell}(\gamma) \otimes \gamma^\ell)
\end{aligned}
$$

This concludes the proof. ◇

Note that in (24) the first part of the expression, $C_j(\gamma)y(k) \oplus D_j(\gamma)u(k-1)$, only depends on values of previous event steps and the second part of the expression, $F_j(\gamma)u(k+j)$, only on present and future values of the input signal.

Using the results of lemma 1, we can construct matrices that relate the future output signal with past values of the output and future values of the input. By defining the vector

$$
\tilde{y}_0(k) = \begin{bmatrix} C_0(\gamma)y(k) \oplus D_0(\gamma)u(k-1) \\ \vdots \\ C_{N_\mathrm{p}-1}(\gamma)y(k) \oplus D_{N_\mathrm{p}-1}(\gamma)u(k-1) \end{bmatrix}, \tag{25}
$$

and the constant matrix

$$
\tilde{F} = \begin{bmatrix} f_{0,0} & \varepsilon & \cdots & \varepsilon \\ f_{0,1} & f_{1,1} & & \vdots \\ \vdots & & \ddots & \\ f_{0,N_\mathrm{p}-1} & \cdots & & f_{N_\mathrm{p}-1,N_\mathrm{p}-1} \end{bmatrix}, \tag{26}
$$

we obtain the relation

$$
\tilde{y}(k) = \tilde{y}_0(k) \oplus \tilde{F} \otimes \tilde{u}(k)
$$

where

$$
\tilde{y}(k) = \begin{bmatrix} \hat{y}(k|k) \\ \vdots \\ \hat{y}(k+N_\mathrm{p}-1|k) \end{bmatrix}, \qquad \tilde{u}(k) = \begin{bmatrix} u(k) \\ \vdots \\ u(k+N_\mathrm{p}-1) \end{bmatrix}
$$

The aim is now to compute an optimal input sequence $\tilde{u}(k)$ that minimizes $J(k)$ subject to constraints on the inputs and outputs. These constraints are due to limits on the input and output event separation times or due to maximum due dates for the output events. Since the elements of $u(k)$ correspond to consecutive event occurrence times, we have the additional condition $\Delta u(k+j) = u(k+j) - u(k+j-1) \geq 0$ for $j = 0, \ldots, N_\mathrm{p}-1$. Furthermore, in order to reduce the number of decision variables and the corresponding computational complexity we introduce a control horizon N_c ($\leq N_\mathrm{p}$) and we impose the additional condition that the input rate[4] should be constant from event step $k + N_\mathrm{c} - 1$ on:

$$\Delta u(k+j) = \Delta u(k+N_\mathrm{c}-1) \;\text{ for }\; j = N_\mathrm{c}, \ldots, N_\mathrm{p}-1,$$

or equivalently

$$\Delta^2 u(k+j) = \Delta u(k+j) - \Delta u(k+j-1) = 0 \;\text{ for }\; j = N_\mathrm{c}, \ldots, N_\mathrm{p}-1.$$

MPC uses a receding horizon principle. This means that after computation of the optimal control sequence $u(k), \ldots, u(k+N_\mathrm{c}-1)$, only the first control sample $u(k)$ will be implemented, subsequently the horizon is shifted one event step, and the optimization is restarted with new information of the measurements. The MPC problem for MPL systems for event step k is formulated as follows (compare with [8] for the state-space case):

$$\min_{\tilde{u}(k),\tilde{y}(k)} J(\tilde{u}(k), \tilde{y}(k)) = \min_{\tilde{u}(k),\tilde{y}(k)} J_\mathrm{out}(\tilde{y}(k)) + \lambda J_\mathrm{in}(\tilde{u}(k)) \tag{27}$$

subject to

$$\tilde{y}(k) = \tilde{y}_0(k) \oplus \tilde{F} \otimes \tilde{u}(k) \tag{28}$$
$$A_\mathrm{c}(k)\tilde{u}(k) + B_\mathrm{c}(k)\tilde{y}(k) \leqslant c_\mathrm{c}(k) \tag{29}$$
$$\Delta u(k+j) \geqslant 0 \qquad \text{for } j = 0, \ldots, N_\mathrm{p}-1 \tag{30}$$
$$\Delta^2 u(k+j) = 0 \qquad \text{for } j = N_\mathrm{c}, \ldots, N_\mathrm{p}-1, \tag{31}$$

where equation (29) reflects the constraints on the inputs and outputs. If we replace (28) by the following inequality:

$$\tilde{y}(k) \geq \tilde{y}_0(k) \oplus \tilde{F} \otimes \tilde{u}(k) \tag{32}$$

we obtain the relaxed MPL-MPC problem, which is defined by the optimization of (27) subject to (32), (29), (30) and (31).

Theorem 2 *Let the mapping $\tilde{y} \to B_\mathrm{c}(k)\tilde{y}$ be a monotonically non-decreasing function of $\tilde{y}$. Let $(\tilde{u}^*, \tilde{y}^*)$ be an optimal solution of the relaxed MPL-MPC problem. If we define $\tilde{y}^\sharp(k) = \tilde{y}_0(k) \oplus \tilde{F} \otimes \tilde{u}^*(k)$, then $\tilde{y}^\sharp(k)$ is an optimal solution of the original MPL-MPC problem.*

Proof: Similar to [8]. ◇

So if the linear constraints are monotonically non-decreasing as a function of $\tilde{y}(k)$, the MPL-MPC problem can be recast as a convex problem. Moreover, by introducing some additional dummy variables the problem can even be reduced to a linear programming problem (see [8]).

[4]For a manufacturing system the input rate corresponds to the rate at which raw material or external resources are fed to the system

5 The adaptive MPC algorithm

The two important ingredients of the adaptive controller, identification and control law, have been discussed in the previous sections. This leads to the final adaptive MPC algorithm, which consists of the following 5 steps.

step 1 (initial identification): The model is initialized by computing $\hat{\theta}_0$ using equation (8).

step 2 (measurement): Obtain new measurement $y(k)$ at event step k.

step 3 (adaptation): Make a recursive estimation of $\hat{\theta}^{(k)}$ using equation (18)-(19).

step 4 (control law): Compute new control sequence $\tilde{u}^*(k)$ by solving the relaxed MPL-MPC problem, which is defined by the optimization of (27) subject to (29), (30), (31) and (32). The first element $u(k)$ of $\tilde{u}^*(k)$ is fed to the system.

step 5 (receding horizon): The horizon is shifted one step $k \to k+1$. Return to step 2.

As was pointed out in [26], MPC for MPL systems is different from conventional MPC in the sense that the event counter k is not directly related to a specific time. The best time $t(k)$ to start the estimation of $\hat{\theta}(k)$ and subsequently to start the optimization to compute the optimal control sequence $\tilde{u}(k)$ with elements $u(k|k), u(k+1|k), \ldots, u(k+N_c-1|k)$, is the moment that a new measurement $y(k)$ becomes available, so $t(k) = y(k)$.

The tuning rules of a predictive controller for max-plus-linear systems, as derived in [26] are still valid. Of course one should keep in mind that the prediction horizon N_p is related to the length of the step response of the open-loop system: the time interval $[1, N_p]$ should contain the crucial dynamics of the process. Therefore, N_p should be larger than the worst-case step response length. The trade-off constant λ should satisfy $0 < \lambda < 1$ and it is usually chosen as small as possible without causing instability or numerical problems in the optimization. The parameter N_c, called control horizon, can be chosen between 1 and N_p. We usually take it equal to the upper bound of the minimal system order, which is equal to n (=order of the A polynomial) in the time-varying case.

Note that the identification of the MPL system will be done in closed-loop. As in system identification of time-driven systems, we have to take care that the input signal will be 'rich' enough to be able to estimate all parameters. In [25] we have derived constraints for signals to be persistently exciting. If in step 3 we find that the input signal is not persistently exciting, we can add additional requirements on the input signal in step 4 to make sure that the future input signal will become 'rich' enough to do an accurate parameter estimation.[5]

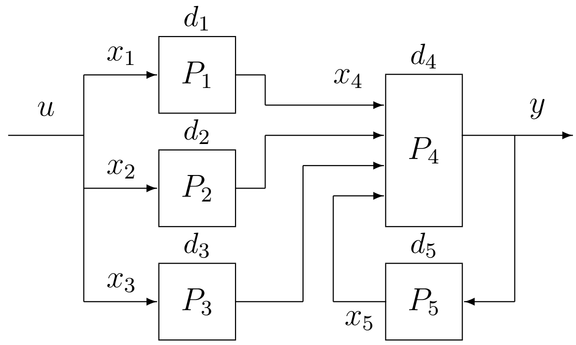

Figure 1: A simple manufacturing system.

6 Example

Consider the production system of Fig. 1. This manufacturing system consists of five processing units, P_1 to P_5. Raw material is fed to P_1, P_2 and P_3, processed and sent to P_4 where assembly takes place. Unit P_4 works with pallets, on which the assembly takes place. Each production cycle one pallet is used in unit P_4, while at the same time a second pallet is recycled through unit P_5. The units P_1, P_2 and P_3 work continuously, and may work on more products at the same time[6]. The units P_4 and P_5 work in batches (one batch for each finished product), and can only start working on a new product if they have finished processing the previous product. Each processing unit starts working as soon as all parts are available. The preprocessing in P_2 and P_3 takes so much time that the output is delayed one cycle in P_2 and two cycles in P_3. The processing time for P_i, $i = 1, \ldots, 5$ is denoted by d_i. It takes t_4 time units for the pallet to get from P_4 to P_5. The other transportation times and the set-up times are assumed to be negligible.

The system is described by the following state space model:

$$\begin{aligned}
x_1(k) &= x_2(k) \;=\; x_3(k) \;=\; u(k) \\
x_4(k) &= \max\Big(x_1(k) + d_1, x_2(k-1) + d_2, x_3(k-2) + d_3, x_4(k-1) + d_4, x_5(k-1) + d_5\Big) \\
&= \max\Big(u(k) + d_1, u(k-1) + d_2, u(k-2) + d_3, x_4(k-1) + d_4, x_5(k-1) + d_5\Big) \\
x_5(k) &= \max\Big(x_4(k-1) + d_4 + t_4, x_5(k-1) + d_5\Big)
\end{aligned}$$

with $u(k)$ the time at which a batch of raw material is fed to the system for the $(k+1)$th time, $x_i(k)$ the time at which P_i starts working for the kth time, and $y(k)$ the time at which the kth finished product leaves the system.

Define the state space parameter vector

$$\theta_{\text{ss}} = \begin{bmatrix} d_1 & d_2 & d_3 & d_4 & d_5 & t_4 \end{bmatrix}$$

[5]This implies that system (27)–(31) will be extended with some extra constraints.

[6]e.g. a conveyor belt with a heating step.

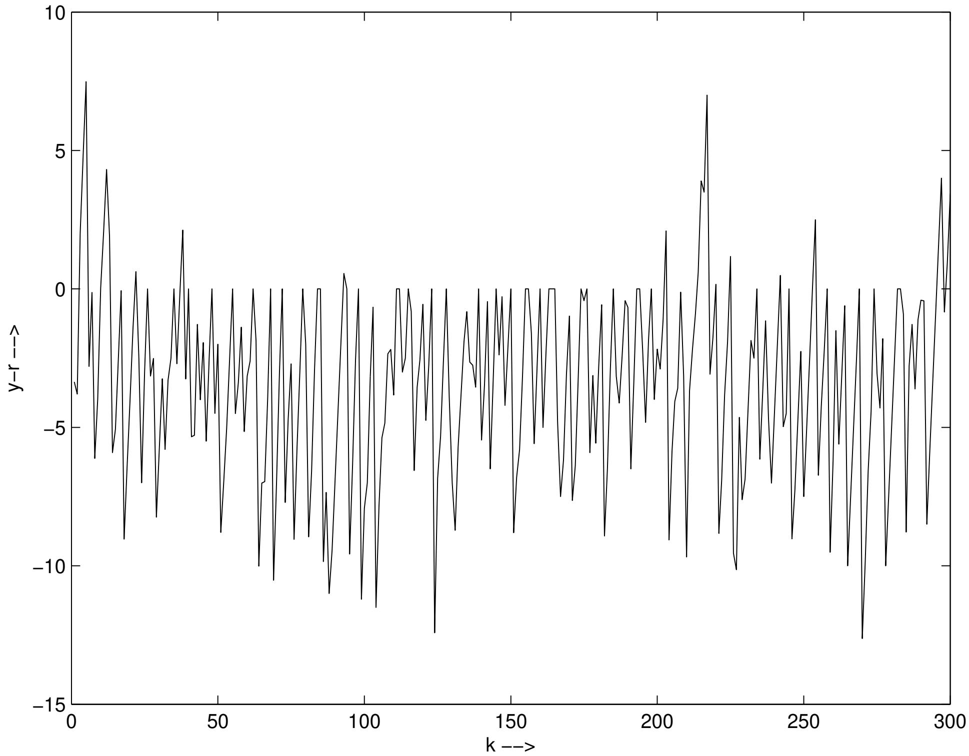

Figure 2: Due date error $y(k){-}r(k)$

We simulate the system for $k = 1, \ldots, 300$ where

$$\theta_{\text{ss}} = \begin{bmatrix} 3 & 6 & 10 & 2 & 1 & 2 \end{bmatrix} \quad \text{for} \quad k = 1, \ldots, 100,$$

$$\theta_{\text{ss}} = \begin{bmatrix} 3 & 5 & 8 & 1 & 1 & 1 \end{bmatrix} \quad \text{for} \quad k = 101, \ldots, 200,$$

$$\theta_{\text{ss}} = \begin{bmatrix} 1 & 3 & 9 & 1.5 & 1 & 2.5 \end{bmatrix} \quad \text{for} \quad k = 201, \ldots, 300.$$

We can translate the MPL state space system into an MPLIO system[7], described by the input-output relation

$$y(k) = A(\gamma)y(k) \oplus B(\gamma)u(k)$$

where $A(\gamma)$ and $B(\gamma)$ are polynomial operators

$$A(\gamma) = a_1 \otimes \gamma^1 \oplus a_2 \otimes \gamma^2 \quad , \quad B(\gamma) = b_0 \otimes \gamma^0 \oplus b_1 \otimes \gamma^1 \oplus b_2 \otimes \gamma^2$$

The input-output parameter vector

$$\theta = \begin{bmatrix} a_1 & a_2 & b_0 & b_1 & b_2 \end{bmatrix}$$

[7]The similarity is proven by showing that the impulse responses of both systems are equivalent. Loosely speaking, the impulse response of the systems can be computed by successive substitution with $u(k) = 0$, for $k = 0$ and $u(k) = \varepsilon$ elsewhere, with the initial conditions all set to ε.

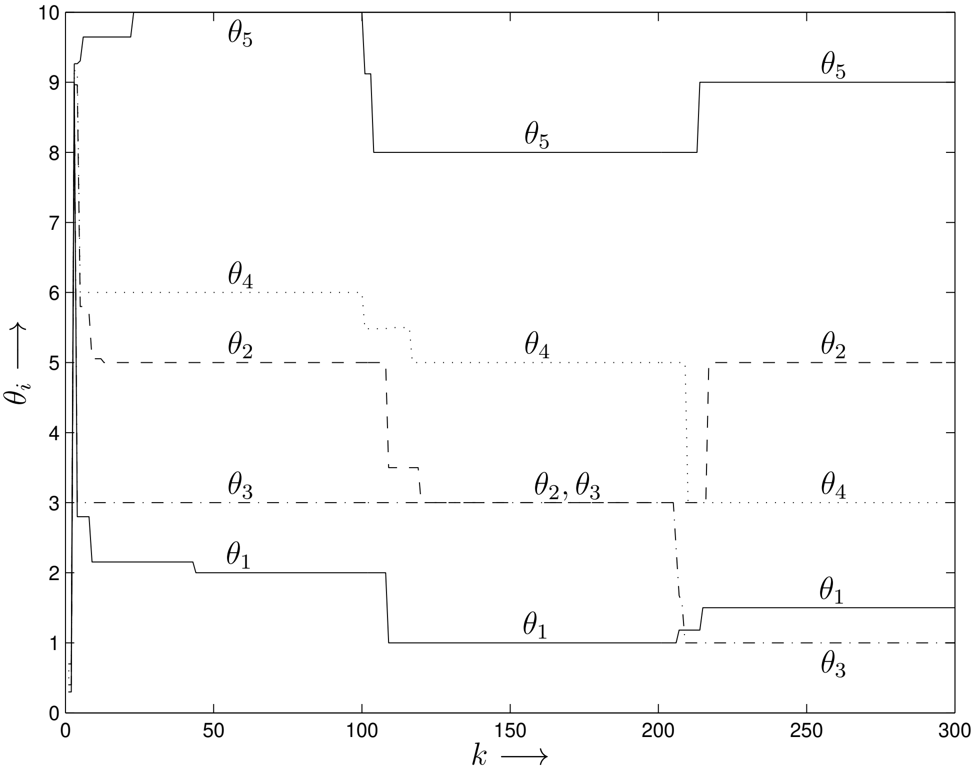

Figure 3: Estimated parameters $\hat{\theta}$

is now given by

$$\theta = \begin{bmatrix} 2 & 5 & 3 & 6 & 10 \end{bmatrix} \quad \text{for} \quad k = 1, \ldots, 100,$$

$$\theta = \begin{bmatrix} 1 & 3 & 3 & 5 & 8 \end{bmatrix} \quad \text{for} \quad k = 101, \ldots, 200,$$

$$\theta = \begin{bmatrix} 1.5 & 5 & 1 & 3 & 9 \end{bmatrix} \quad \text{for} \quad k = 201, \ldots, 300.$$

An adaptive model predictive controller strategy is applied following section 5. The due date signal $r(k)$ is a non-decreasing random[8] signal with an average slope of 3.013 and variance 19.4. The initial state is set to $p(0) = [\,0\,0\,0\,0\,0\,]^T$ and the criterion function is given by (27) for $N_\mathrm{p} = 10$, $N_\mathrm{c} = 2$ and $\lambda = 0.01$. For each k, the model is updated using an update interval with $N_\mathrm{e} = 15$ and $\alpha = 1$, and (with the updated model) the optimal input sequence is computed, and finally the first element $u(k)$ of the sequence $\tilde{u}(k)$ is applied to the system (due to the receding horizon strategy).

Figure 2 gives the due date error, i.e. the difference between the due date signal and the output signal $y(k)$. Note that when the due date error is positive, we have a due date violation. Most of the time this happens near the jumps of the parameters. Figure 3 shows the model parameters, as estimated by the identification algorithm. Note that after a transient interval, the estimated parameters converge to their true values.

[8]The due date signal is chosen random to express the varying customer demand.

7 Discussion

In this paper we have derived a technique for adaptive MPC of MPL systems, given an input-output description. We have included the identification and estimation update into the algorithm. If the linear constraints are a non-decreasing function of the output the computation of the MPC control law can be done using a linear programming algorithm. An simulation example has shown that the algorithm gives a good closed-loop behavior in the case of a MPLIO models with time-varying parameters.

Acknowledgments

Research partially funded by the Dutch Technology Foundation STW project “Model predictive control for hybrid systems” (DMR.5675), by the European IST project “Modelling, Simulation and Control of Nonsmooth Dynamical Systems (SICONOS)” (IST-2001-37172), by the NWO/STW VIDI project “Multi-agent control of large-scale hybrid systems” (DWV.6188), and by the Deutsche Forschungsgemeinschaft (DFG) under contract number Kr 949/9.

References

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

[2] J.L. Boimond and J.L. Ferrier. Internal model control and max-algebra: Controller design. *IEEE Transactions on Automatic Control*, 41(3):457–461, March 1996.

[3] E.F. Camacho and C. Bordons. *Model Predictive Control in the Process Industry, Advances in Industrial Control.* Springer, London, 1995.

[4] C.G. Cassandras and S. Lafortune. *Introduction to Discrete Event Systems.* Kluwer Academic Publishers, Boston, 1999.

[5] D.W. Clarke, C. Mohtadi, and P.S. Tuffs. Generalized predictive control – Part I. The basic algorithm. *Automatica*, 23(2):137–148, March 1987.

[6] G. Cohen, S. Gaubert, and J.P. Quadrat. Max-plus algebra and system theory: Where we are and where to go now. *Annual Reviews in Control*, 23:207–219, 1999.

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

[8] B. De Schutter and T. van den Boom. Model predictive control for max-plus-linear discrete event systems. *Automatica*, 37(7):1049–1056, July 2001.

[9] B. De Schutter and T.J.J. van den Boom. Model predictive control for max-plus-linear systems. In *American Control Conference 2000,Chicago USA*, pages 4046–4050, June, 2000.

[10] F. Gallot, J. L. Boimond, and L. Hardouin. Identification of simple elements in max–algebra: Application to SISO discrete event systems modelisation. In *Proceedings of the 4th ECC*, Brussels, 1997.

[11] C.E. Garcia, D.M. Prett, and M. Morari. Model predictive control: Theory and practice - a survey. *Automatica*, 25(3):335–348, 1989.

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

[13] Y.C. Ho. *Discrete Event Dynamic Systems: Analyzing Complexity and Performance in the Modern World*. IEEE Press, Piscataway, New Jersey, 1992.

[14] L. Libeaut and J.J. Loiseau. Admissible initial conditions and control of timed event graphs. In *Proceedings of the 34th IEEE Conference on Decision and Control*, pages 2011–2016, New Orleans, Louisiana, December 1995.

[15] L. Ljung. *System Identification: Theory for the User*. Prentice Hall, Englewood Cliffs, NJ, 1987.

[16] J.M. Maciejowski. *Predictive control with constraints*. Prentice Hall, Pearson Education Limited, Harlow, UK, 2002.

[17] E. Menguy. Contribution à la commande des systèmes linéaires dans les dioides. PhD thesis, Université d'Angers, Angers, France, 1997.

[18] E. Menguy, J. L. Boimond, L. Hardouin, and J. L. Ferrier. A first step towards adaptive control for linear systems in max algebra. *Discrete Event Dynamic Systems*, 10(4):347–368, 2000.

[19] E. Menguy, J.L. Boimond, and L. Hardouin. A feedback control in max-algebra. In *Proceedings of the European Control Conference (ECC'97)*, Brussels, Belgium, paper 487, July 1997.

[20] E. Menguy, J.L. Boimond, L. Hardouin, and J.L. Ferrier. A first step towards adaptive control for linear systems in max algebra. *Discrete Event Dynamic Systems: Theory and Applications*, 10(4):347–367, 2000.

[21] E. Mosca. *Optimal Predictive and Adaptive Control*. Prentice Hall, Englewood Cliffs, NJ, 1995.

[22] P.J. Ramadge and W.M. Wonham. Supervisory control of a class of discrete-event processes. *SIAM Journal on Control and Optimization*, 25(1):206–230, January 1987.

[23] K. Rudie and W.M. Wonham. Think globally, act locally: Decentralized supervisory control. *IEEE Transactions on Automatic Control*, 37(11):1692–1708, November 1992.

[24] G. Schullerus and V. Krebs. A method for estimating the holding times in timed event graphs. In *Proceedings of the 6th Workshop on Discrete Event Systems (WODES'02)*, pages 119–124, Zaragossa, 2002.

[25] G. Schullerus, V. Krebs, T.J.J. van den Boom, and B. De Schutter. Input signal design for max–plus–linear system identification using model predictive control. In *European Control Conference 2003, Cambridge, UK, paper 26*, 2003.

[26] T.J.J. van den Boom and B. De Schutter. Properties of MPC for max-plus-linear systems. *European Journal of Control*, 8(5), 2002.

Appendix A

In this appendix we investigate the effect of tuning parameter $\alpha \in (0,2)$. Denote

$$\hat{\theta}(k, k-N_e) \;=\; Y(k, k-N_\mathrm{e})) \otimes' \big(- P^T(k, k-N_\mathrm{e})\big) \tag{33}$$

$$\tilde{\hat{\theta}}^{(\ell-1)} \;=\; \big(\hat{\theta}^{(l-1)}(k) \otimes P(k, k-N_\mathrm{e})\big) \otimes' (-P^T(k, k-N_\mathrm{e})) \tag{34}$$

For the sake of simplicity the arguments k and $k - N_e$ will be omitted in the subsequent considerations.

$\tilde{\hat{\theta}}^{(\ell-1)}$ is the greatest solution to the inequation

$$\theta \otimes P \;\leq\; \hat{\theta}^{(\ell-1)} \otimes P\ ,$$

such that $\tilde{\hat{\theta}}^{(\ell-1)} \geq \hat{\theta}^{(\ell-1)}$ also holds. Using this property, one obtains an upper bound, from (18) for $\hat{\theta}^{(\ell)}$:

$$\begin{aligned}\hat{\theta}^{(\ell)} \;&=\; \hat{\theta}^{(\ell-1)} + \alpha(\hat{\theta} - \underbrace{\tilde{\hat{\theta}}^{(\ell-1)}}_{\geq\, \hat{\theta}^{(\ell-1)}}) \\ &\leq\; \hat{\theta}^{(\ell-1)} + \alpha(\hat{\theta} - \hat{\theta}^{(\ell-1)}) \;=\; \hat{\theta}^{(\ell-1)}(1-\alpha) + \alpha\hat{\theta} \quad \ell > 0\ .\end{aligned} \tag{35}$$

Thus, $\hat{\theta}^{(1)} \leq \hat{\theta}^{(0)}(1-\alpha) + \alpha\hat{\theta}$ also holds. Assume now that

$$\hat{\theta}^{(\ell)} \;\leq\; \hat{\theta}^{(0)}(1-\alpha)^\ell + \hat{\theta}\left(1 - (1-\alpha)^\ell\right) \tag{36}$$

holds. Then

$$\begin{aligned}\hat{\theta}^{(\ell+1)} \;&\leq\; \hat{\theta}^{(\ell)}(1-\alpha) + \alpha\hat{\theta} \;\leq\; (1-\alpha)\left(\hat{\theta}^{(0)}(1-\alpha)^\ell + \hat{\theta}\left(1-(1-\alpha)^\ell\right)\right) + \alpha\hat{\theta} \\ &\leq\; \hat{\theta}^{(0)}(1-\alpha)^{\ell+1} + \hat{\theta}\left(1-(1-\alpha)^{\ell+1}\right)\end{aligned}$$

such that by induction (36) holds for any $\ell > 0$. As $\alpha \in (0, 2)$ and therefore $|1 - \alpha| < 1$,

$$\lim_{\ell \to \infty} \hat{\theta}^{(\ell)} \leq \lim_{\ell \to \infty} \left(\hat{\theta}^{(0)} (1 - \alpha)^{\ell} + \hat{\theta} \left(1 - (1 - \alpha)^{\ell} \right) \right) = \hat{\theta} \tag{37}$$

holds, such that $\hat{\theta}$ is an upper bound for $\hat{\theta}^{(\ell)}$ for ℓ sufficiently large.

Assume now, that for all $\ell \geq \tilde{\ell}$, $\hat{\theta} \geq \hat{\theta}^{(\ell)}$ holds. Following the same reasoning than in [17] one obtains

$$\hat{\theta}^{(\ell)} \otimes P = \tilde{\hat{\theta}}^{(\ell)} \otimes P \leq \hat{\theta} \otimes P \leq Y \tag{38}$$

Since $\hat{\theta}$ is the greatest solution to $x \otimes P \leq Y$, one concludes that $\hat{\theta} \geq \tilde{\hat{\theta}}^{(\ell)}$.

Thus from (35) one obtains

$$\hat{\theta}^{(\ell)} - \hat{\theta}^{(\ell-1)} = \alpha(\hat{\theta} - \tilde{\hat{\theta}}^{(\ell-1)}) \geq 0 \qquad \ell \geq \tilde{\ell}$$

such that for sufficiently large ℓ, the series $\hat{\theta}^{(\ell)}$ is increasing and bounded from above by $\hat{\theta}$ according to (37).

Note that convergence in a strict mathematical sense can not be concluded from the above considerations. However, they provide an insight in the behavior of the series $\hat{\theta}^{(\ell)}$. As in practical applications, the tolerances are typically not too tight, the series will in practice converge for ℓ sufficiently large. The above considerations follow the same ideas than the proof given in [17] for $\alpha = 1$. Since in [17] $\hat{\theta}^{(\ell)}$ is increasing and bounded from above by $\hat{\theta}$ for $\ell > 0$, it seems that using a tuning parameter $\alpha \neq 1$ slows down the convergence of the iterative procedure.

The convergence of the series for some ℓ implies $\hat{\theta} = \tilde{\hat{\theta}}^{(\ell)}$. Provided that $Y = \hat{\theta} \otimes P$, using the reasoning from [17] one can replace the inequalities in (38) by the corresponding equalities. Then

$$Y = \hat{\theta}^{(\ell)} \otimes P$$

is verified in for ℓ sufficiently large.