**Delft University of Technology**

**Fac. of Information Technology and Systems** **Control Systems Engineering**

Technical report bds:99-10

# Model predictive control for max-plus-linear discrete event systems*

B. De Schutter and T. van den Boom

*If you want to cite this report, please use the following reference instead:*
B. De Schutter and T. van den Boom, "Model predictive control for max-plus-linear discrete event systems," *Automatica*, vol. 37, no. 7, pp. 1049–1056, July 2001.

Control Systems Engineering
Faculty of Information Technology and Systems
Delft University of Technology
Delft, The Netherlands
phone: +31-15-278.51.19 (secretary)
fax: +31-15-278.66.79
Current URL: `http://www.dcsc.tudelft.nl`

# Model predictive control for max-plus-linear discrete event systems ⋆

Bart De Schutter [a] Ton van den Boom [a]

[a] *Control Lab, Faculty of Information Technology and Systems, Delft University of Technology, P.O. Box 5031, 2600 GA Delft, The Netherlands*

**The conventional model predictive control framework is extended and adapted to max-plus-linear systems, i.e., discrete-event systems that can be described by models that are "linear" in the (max,+) algebra.**

**Abstract**

Model predictive control (MPC) is a very popular controller design method in the process industry. A key advantage of MPC is that it can accommodate constraints on the inputs and outputs. Usually MPC uses linear discrete-time models. In this paper we extend MPC to a class of discrete-event systems that can be described by models that are "linear" in the max-plus algebra, which has maximization and addition as basic operations. In general the resulting optimization problem are non-linear and nonconvex. However, if the control objective and the constraints depend monotonically on the outputs of the system, the model predictive control problem can be recast as problem with a convex feasible set. If in addition the objective function is convex, this leads to a convex optimization problem, which can be solved very efficiently.

*Key words:* Discrete-event systems; Predictive control; Model-based control; Generalized predictive control; Max-plus-linear systems; Max-plus algebra.

⋆ This paper was not presented at any IFAC meeting. Corresponding author B. De Schutter. Tel. +31-15-278.51.13. Fax +31-15-278.66.79. E-mail b.deschutter@its.tudelft.nl.

*Preprint submitted to Elsevier*

## 1 Introduction

Process industry is characterized by always tighter product quality specifications, increasing productivity demands, new environmental regulations and fast changes in the economical market. In the last decades Model Predictive Control (MPC), has shown to respond in an effective way to these demands in many practical process control applications and is therefore widely accepted in the process industry. Control design techniques such as pole placement, LQG, $H_2$, $H_\infty$, etc. yield optimal controllers or control input sequences for the entire future evolution of the system. However, extending these methods to include additional constraints on the inputs and outputs is not easy. An important advantage of MPC is that the use of a finite horizon allows the inclusion of such additional constraints. Furthermore, MPC can handle structural changes, such as sensor or actuator failures and changes in system parameters or system structure, by adapting the model.

Traditionally MPC uses linear discrete-time models for the process to be controlled. In this paper we extend and adapt the MPC framework to a class of discrete-event systems. In general, models that describe the behavior of a discrete-event system are nonlinear in conventional algebra. However, there is a class of discrete-event systems – the max-plus-linear discrete-event systems – that can be described by a model that is "linear" in the max-plus algebra (Baccelli *et al.*, 1992). The max-plus-linear discrete-event systems can be characterized as the class of discrete-event systems in which only synchronization and no concurrency or choice occurs. So typical examples are serial production lines, production systems with a fixed routing schedule, and railway networks.

We will develop an MPC framework for max-plus-linear discrete-event systems. Several other authors have already developed methods to compute optimal control sequences for max-plus-linear discrete-event systems (Baccelli *et al.*, 1992; Boimond and Ferrier, 1996; Libeaut and Loiseau, 1995; Menguy *et al.*, 1997; Menguy *et al.*, 1998*a*; Menguy *et al.*, 1998*b*). The main advantage of our approach is that it allows to include general linear inequality constraints on the inputs and outputs of the system.

## 2 Model predictive control

In this section we give a short introduction to MPC. Since we will only consider the deterministic, i.e. noiseless, case for max-plus-linear systems (cf. Remark 1), we will also omit the noise terms in this brief introduction to MPC for linear systems. More extensive information on MPC can be found in (Camacho and Bordons, 1995; García *et al.*, 1989).

Consider a plant with $m$ inputs and $l$ outputs that can be modeled by a state space description of the form

$$x(k+1) = A\,x(k) + B\,u(k) \tag{1}$$
$$y(k) = C\,x(k)\ . \tag{2}$$

The vector $x$ represents the state, $u$ the input, and $y$ the output. In order to distinguish systems that can be described by a model of the form (1) – (2) from the max-plus-linear systems that will be considered later on, a system that can be modeled by (1) – (2) will be called a *plus-times-linear* (PTL) system.

In MPC a performance index or cost criterion $J$ is formulated that reflects the reference tracking error ($J_{\text{out}}$) and the control effort ($J_{\text{in}}$):

$$\begin{aligned} J &= J_{\text{out}} + \lambda J_{\text{in}} \\ &= \sum_{j=1}^{N_{\text{p}}} \|(\hat{y}(k+j|k) - r(k+j)\|^2 \ + \ \lambda \sum_{j=1}^{N_{\text{p}}} \|u(k+j-1)\|^2 \end{aligned} \tag{3}$$

where $\hat{y}(k+j|k)$ is the estimate of the output at time step $k+j$ based on the information available at time step $k$, $r$ is a reference signal, $\lambda$ is a nonnegative scalar, and $N_{\text{p}}$ is the prediction horizon.

In MPC the input is taken to be constant from a certain point on: $u(k+j) = u(k+N_{\text{c}}-1)$ for $j = N_{\text{c}}, \ldots, N_{\text{p}}-1$ where $N_{\text{c}}$ is the control horizon. The use of a control horizon leads to a reduction of the number of optimization variables. This results in a decrease of the computational burden, a smoother controller signal (because of the emphasis on the average behavior rather than on aggressive noise reduction), and a stabilizing effect (since the output signal is forced to its steady-state value).

MPC uses a receding horizon principle. At time step $k$ the future control sequence $u(k), \ldots, u(k+N_{\text{c}}-1)$ is determined such that the cost criterion is minimized subject to the constraints. At time step $k$ the first element of the optimal sequence ($u(k)$) is applied to the process. At the next time instant the horizon is shifted, the model is updated with new information of the measurements, and a new optimization at time step $k+1$ is performed.

By successive substitution of (1) in (2), estimates of the future values of the output can be computed (Camacho and Bordons, 1995). In matrix notation we obtain:

$$\tilde{y}(k) = H\tilde{u}(k) + g(k)$$

with

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

$$H = \begin{bmatrix} CB & 0 & \dots & 0 \\ CAB & CB & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ CA^{N_\mathrm{p}-1}B & CA^{N_\mathrm{p}-2}B & \dots & CB \end{bmatrix}, \quad g(k) = \begin{bmatrix} CA \\ CA^2 \\ \vdots \\ CA^{N_\mathrm{p}} \end{bmatrix} x(k). \tag{4}$$

The MPC problem at time step $k$ for PTL systems is defined as follows:

> Find the input sequence $u(k), \dots, u(k+N_\mathrm{c}-1)$ that minimizes the performance index $J$ subject to the linear constraint
>
> $$E(k)\tilde{u}(k) + F(k)\tilde{y} \leqslant h(k) \tag{5}$$
>
> with $E(k) \in \mathbb{R}^{p \times mN_\mathrm{p}}$, $F(k) \in \mathbb{R}^{p \times lN_\mathrm{p}}$, $h(k) \in \mathbb{R}^p$ for some integer $p$, subject to the control horizon constraint
>
> $$u(k+j) = u(k+N_\mathrm{c}-1) \qquad \text{for } j = N_\mathrm{c}, N_\mathrm{c}+1, \dots \tag{6}$$

Note that minimizing $J$ subject to (5) and (6), boils down to a convex quadratic programming problem, which can be solved very efficiently.

The parameters $N_\mathrm{p}$, $N_\mathrm{c}$ and $\lambda$ are the three basic MPC tuning parameters: The prediction horizon $N_\mathrm{p}$ is related to the length of the step response of the process, and the time interval $(1, N_\mathrm{p})$ should contain the crucial dynamics of the process. The control horizon $N_\mathrm{c} \leqslant N_\mathrm{p}$ is usually taken equal to the system order. The parameter $\lambda \geqslant 0$ makes a trade-off between the tracking error and the control effort, and is usually chosen as small as possible (while still getting a stabilizing controller).

## 3 Max-plus algebra and max-plus-linear systems

### *3.1 Max-plus algebra*

The basic operations of the max-plus algebra are maximization and addition, which will be represented by $\oplus$ and $\otimes$ respectively:

$$x \oplus y = \max(x, y) \quad \text{and} \quad x \otimes y = x + y$$

for $x, y \in \mathbb{R}_\varepsilon \stackrel{\text{def}}{=} \mathbb{R} \cup \{-\infty\}$. Define $\varepsilon = -\infty$. The structure $(\mathbb{R}_\varepsilon, \oplus, \otimes)$ is called the max-plus algebra (Baccelli *et al.*, 1992). The operations $\oplus$ and $\otimes$ are called the max-plus-algebraic addition and max-plus-algebraic multiplication respectively since many properties and concepts from linear algebra can be translated to the max-plus algebra by replacing $+$ by $\oplus$ and $\times$ by $\otimes$.

The matrix $\mathcal{E}_{m\times n}$ is the $m \times n$ max-plus-algebraic zero matrix: $(\mathcal{E}_{m\times n})_{ij} = \varepsilon$ for all $i, j$; and $E_n$ is the $n \times n$ max-plus-algebraic identity matrix: $(E_n)_{ii} = 0$ for all $i$ and $(E_n)_{ij} = \varepsilon$ for all $i, j$ with $i \neq j$. If $A, B \in \mathbb{R}_\varepsilon^{m\times n}$, $C \in \mathbb{R}_\varepsilon^{n\times p}$ then

$$\begin{aligned}
(A \oplus B)_{ij} &= a_{ij} \oplus b_{ij} = \max(a_{ij}, b_{ij}) \\
(A \otimes C)_{ij} &= \bigoplus_{k=1}^{n} a_{ik} \otimes c_{kj} = \max_k (a_{ik} + c_{kj})
\end{aligned}$$

for all $i, j$. Note the analogy with the conventional definitions of matrix sum and product. The max-plus-algebraic matrix power of $A \in \mathbb{R}_\varepsilon^{n\times n}$ is defined as follows: $A^{\otimes^0} = E_n$ and $A^{\otimes^k} = A \otimes A^{\otimes^{k-1}}$ for $k = 1, 2, \ldots$

### *3.2 Max-plus-linear systems*

Discrete-event systems with only synchronization and no concurrency can be modeled by a max-plus-algebraic model of the following form (Baccelli *et al.*, 1992):

$$x(k+1) = A \otimes x(k) \ \oplus \ B \otimes u(k) \tag{7}$$

$$y(k) = C \otimes x(k) \tag{8}$$

with $A \in \mathbb{R}_\varepsilon^{n\times n}$, $B \in \mathbb{R}_\varepsilon^{n\times m}$ and $C \in \mathbb{R}_\varepsilon^{l\times n}$ where $m$ is the number of inputs and $l$ the number of outputs. Note the analogy of the description (7) – (8) with the state space model (1) – (2) for PTL systems. An important difference with the description (1) – (2) is that now the components of the input, the output and the state are event times, and that the counter $k$ in (7) – (8) is an event counter (and event occurrence instants are in general not equidistant), whereas in (1) – (2) $k$ increases each clock cycle. A discrete-event system that can be modeled by (7) – (8) will be called a max-plus-linear time-invariant discrete-event system or *max-plus-linear* (MPL) system for short.

**Remark 1** *For PTL systems the influence of noise is usually modeled by adding an extra noise term to the state and/or output equation. For MPL models the entries of the system matrices correspond to production times or transportation times. So instead of modeling noise, (i.e. variation in the processing times), by adding an extra max-plus-algebraic term in (7) or (8), noise should rather be modeled as an additive term to these system matrices. However, this would not lead to a nice model structure. Therefore, we will use the max-plus-linear model (7) – (8) as an approximation of a discrete-event system with uncertainty and/or modeling errors when we extend the MPC framework to MPL systems. This also motivates the use of a receding horizon strategy when we define MPC for MPL systems, since then we can regularly update our model of the system as new measurements become available.*

## 4 Model predictive control for MPL systems

### *4.1 Evolution of the system*

We assume that $x(k)$, the state at event step $k$, can be measured or estimated using previous measurements. We can then use (7) – (8) to estimate the evolution of the output of the system for the input sequence $u(k), \ldots, u(k+N_{\mathrm{p}}-1)$:

$$\hat{y}(k+j|k) = C \otimes A^{\otimes^j} \otimes x(k) \ \oplus \ \bigoplus_{i=0}^{j-1} C \otimes A^{\otimes^{j-i}} \otimes B \otimes u(k+i) \ ,$$

or, in matrix notation, $\tilde{y}(k) = H \otimes \tilde{u}(k) \oplus g(k)$ with

$$H = \begin{bmatrix} C \otimes B & \varepsilon & \dots & \varepsilon \\ C \otimes A \otimes B & C \otimes B & \dots & \varepsilon \\ \vdots & \vdots & \ddots & \vdots \\ C \otimes A^{\otimes^{N_\mathrm{p}-1}} \otimes B & C \otimes A^{\otimes^{N_\mathrm{p}-2}} \otimes B & \dots & C \otimes B \end{bmatrix},$$

$$g(k) = \begin{bmatrix} C \otimes A \\ C \otimes A^{\otimes^2} \\ \vdots \\ C \otimes A^{\otimes^{N_\mathrm{p}}} \end{bmatrix} \otimes x(k)\,.$$

Note the analogy between these expressions and the corresponding expressions (4) for PTL systems.

### *4.2 Cost criterion*

#### *4.2.1 Tracking error or output cost criterion $J_\mathrm{out}$*

If the due dates $r$ for the finished products are known and if we have to pay a penalty for every delay, a well-suited cost criterion is the tardiness:

$$J_{\mathrm{out},1} = \sum_{j=1}^{N_\mathrm{p}} \sum_{i=1}^{l} \max(\hat{y}_i(k+j|k) - r_i(k+j), 0) \ . \tag{9}$$

If we have perishable goods, then we could want to minimize the differences between the due dates and the actual output time instants. This leads to

$$J_{\mathrm{out},2} = \sum_{j=1}^{N_\mathrm{p}} \sum_{i=1}^{l} |\hat{y}_i(k+j|k) - r_i(k+j)| \ . \tag{10}$$

If we want to balance the output rates, we could consider

$$J_{\mathrm{out},3} = \sum_{j=2}^{N_\mathrm{p}} \sum_{i=1}^{l} |\Delta^2 \hat{y}_i(k+j|k)| \tag{11}$$

where $\Delta^2 s(k) = \Delta s(k) - \Delta s(k-1) = s(k) - 2s(k-1) + s(k-2)$.

#### *4.2.2 Input cost criterion $J_\mathrm{in}$*

A straightforward translation of the input cost criterion $\tilde{u}^T(k)\tilde{u}(k)$ would lead to a minimization of the input time instants. Since this could result in input buffer overflows, a better objective is to *maximize* the input time instants. For a manufacturing system, this would correspond to a scheme in which raw material is fed to the system as late as possible. As a consequence, the internal buffer levels are kept as low as possible. This also leads to a notion of stability if we let instability for the manufacturing system correspond to internal buffer overflows. So for MPL systems an appropriate cost criterion is $J_{\mathrm{in},0} = -\tilde{u}^T(k)\tilde{u}(k)$. Note that this is exactly the opposite of the input effort cost criterion for PTL systems. Another objective function that leads to a maximization of the input time instants is

$$J_{\mathrm{in},1} = -\sum_{j=1}^{N_\mathrm{p}} \sum_{i=1}^{m} u_i(k+j-1) \ . \tag{12}$$

If we want to balance the input rates we could take

$$J_{\mathrm{in},2} = \sum_{j=1}^{N_\mathrm{p}-1} \sum_{i=1}^{l} |\Delta^2 u_i(ik+j| \ . \tag{13}$$

We could replace the summations in (9) – (13) by max-plus-algebraic summations, or consider weighted mixtures of several cost criteria.

### *4.3 Constraints*

Just as in MPC for PTL systems we can consider the linear constraint

$$E(k)\tilde{u}(k) + F(k)\tilde{y}(k) \leqslant h(k) \ . \tag{14}$$

Furthermore, it is easy to verify that typical constraints for discrete-event systems are minimum or maximum separation between input and output events:

$$a_1(k+j) \leqslant \Delta u(k+j-1) \leqslant b_1(k+j) \qquad \text{for } j = 1, \ldots, N_\mathrm{c} \tag{15}$$

$$a_2(k+j) \leqslant \Delta \hat{y}(k+j|k) \leqslant b_2(k+j) \qquad \text{for } j = 1, \ldots, N_\mathrm{p} \,, \tag{16}$$

or maximum due dates for the output events:

$$\hat{y}(k+j|k) \leqslant r(k+j) \qquad \text{for } j = 1, \ldots, N_\mathrm{p} \,, \tag{17}$$

can also be recast as a linear constraint of the form (14).

Since for MPL systems the input and output sequences correspond to occurrence times of consecutive events, they should be nondecreasing. Therefore, we should always add the condition $\Delta u(k+j) \geqslant 0$ for $j = 0, \ldots, N_\mathrm{p} - 1$ to guarantee that the input sequences are nondecreasing.

### 4.4 The evolution of the input beyond the control horizon

A straightforward translation of the conventional control horizon constraint would imply that the input should stay constant from event step $k + N_\mathrm{c}$ on, which is not very useful for MPL systems since there the input sequences should normally be increasing. Therefore, we change this condition as follows: the feeding rate should stay constant beyond event step $k + N_\mathrm{c}$, i.e.

$$\Delta u(k+j) = \Delta u(k+N_\mathrm{c}-1) \qquad \text{for } j = N_\mathrm{c}, \ldots, N_\mathrm{p}-1\,, \tag{18}$$

or $\Delta^2 u(k+j) = 0$ for $j = N_\mathrm{c}, \ldots, N_\mathrm{p}-1$. This condition introduces regularity in the input sequence and it prevents the buffer overflow problems that could arise when all resources are fed to the system at the same time instant as would be implied by the conventional control horizon constraint (6).

### 4.5 The standard MPC problem for MPL systems

If we combine the material of previous subsections, we finally obtain the following problem:

$$\min_{\tilde{u}(k)} J = \min_{\tilde{u}(k)} J_{\mathrm{out},p_1} + \lambda J_{\mathrm{in},p_2} \tag{19}$$

for some $J_{\mathrm{out},p_1}, J_{\mathrm{in},p_2}$ subject to

$$\tilde{y}(k) = H \otimes \tilde{u}(k) \oplus g(k) \tag{20}$$

$$E(k)\tilde{u}(k) + F(k)\tilde{y}(k) \leqslant h(k) \tag{21}$$

$$\Delta u(k+j) \geqslant 0 \qquad \text{for } j = 0, \ldots, N_\mathrm{p}-1 \tag{22}$$

$$\Delta^2 u(k+j) = 0 \qquad \text{for } j = N_\mathrm{c}, \ldots, N_\mathrm{p}-1 \tag{23}$$

This problem will be called the MPL-MPC problem for event step $k$. MPL-MPC also uses a receding horizon principle.

Other design control design methods for MPL systems are discussed in (Baccelli *et al.*, 1992; Boimond and Ferrier, 1996; Libeaut and Loiseau, 1995; Menguy *et al.*, 1997; Menguy *et al.*, 1998*a*; Menguy *et al.*, 1998*b*). However, these methods do not allow the inclusion of general linear constraints of the form (21) or even simple constraints of the form (15) or (16).

## 5 Algorithms to solve the MPL-MPC problem

### *5.1 Nonlinear optimization*

In general the problem (19) – (23) is a nonlinear nonconvex optimization problem: although the constraints (21) – (23) are convex in $\tilde{u}$ and $\tilde{y}$, the constraint (20) is in general not convex. So we could use standard multi-start nonlinear nonconvex local optimization methods to compute the optimal control policy.

The feasibility of the MPC-MPL problem can be verified by solving the system of (in)equalities (20) – (23) [1]. If the problem is found to be infeasible we can use the same techniques as in conventional MPC and use constraint relaxation (Camacho and Bordons, 1995). Additional information on these topics can be found in (De Schutter and van den Boom, 2000).

### *5.2 The ELCP approach*

Now we discuss an alternative approach which is based on the Extended Linear Complementarity problem (ELCP) (De Schutter and De Moor, 1995). Consider the $i$th row of (20) and define $\mathcal{J}_i = \{ j \mid h_{ij} \neq \varepsilon \}$. We have $\tilde{y}_i(k) = \max_{j \in \mathcal{J}_i} (h_{ij} + \tilde{u}_j(k), g_i(k))$ or equivalently

$$\begin{aligned} &\tilde{y}_i(k) \geqslant h_{ij} + \tilde{u}_j(k) \qquad \text{for } j \in \mathcal{J}_i \\ &\tilde{y}_i(k) \geqslant g_i(k) \end{aligned}$$

with the extra condition that at least one inequality should hold with equality (i.e. at least one residue should be equal to 0):

$$(\tilde{y}_i(k) - g_i(k)) \cdot \prod_{j \in \mathcal{J}_i} (\tilde{y}_i(k) - h_{ij} - \tilde{u}_j(k)) = 0 \ . \tag{24}$$

Hence, equation (20) can be rewritten as a system of equations of the form

$$A_\mathrm{e}\tilde{y}(k) + B_\mathrm{e}\tilde{u}(k) + c_\mathrm{e}(k) \geqslant 0 \tag{25}$$

$$\prod_{j \in \phi_i} (A_\mathrm{e}\tilde{y}(k) + B_\mathrm{e}\tilde{u}(k) + c_\mathrm{e}(k))_j = 0 \qquad \text{for } i = 1, \ldots, lN_\mathrm{p} \tag{26}$$

---

[1] In general this is a nonlinear system of equations but if the constraints depend monotonically on the output, the feasibility problem can be recast as a linear programming problem (cf. Theorem 2).

for appropriately defined matrices and vectors $A_{\mathrm{e}}, B_{\mathrm{e}}, c_{\mathrm{e}}$ and index sets $\phi_i$. We can rewrite the linear constraints (21) – (23) as

$$D_{\mathrm{e}}(k)\tilde{y}(k) + E_{\mathrm{e}}(k)\tilde{u}(k) + f_{\mathrm{e}}(k) \geqslant 0 \tag{27}$$

$$G_{\mathrm{e}}\tilde{u}(k) + h_{\mathrm{e}} = 0 \ . \tag{28}$$

So the feasible set of the MPC problem (i.e. the set of feasible system trajectories) coincides with the set of solutions of the system (25) – (28), which is a special case of an Extended Linear Complementarity Problem (ELCP) (De Schutter and De Moor, 1995). In (De Schutter and De Moor, 1995) we have also developed an algorithm to compute a compact parametric description of the solution set of an ELCP. In order to determine the optimal MPC policy we can use nonlinear optimization algorithms to determine for which values of the parameters the objective function $J$ over the solution set of the ELCP (25) – (28) reaches its global minimum. The algorithm of (De Schutter and De Moor, 1995) to compute the solution set of a general ELCP requires exponential execution times, which that the ELCP approach is not feasible if $N_{\mathrm{c}}$ is large.

### 5.3 *Monotonically nondecreasing objective functions*

Now consider the *relaxed* MPC problem which is also defined by (19) – (23) but with the =-sign in (20) replaced by a $\geqslant$-sign. Note that whereas in the original problem $\tilde{u}(k)$ is the only independent variable since $\tilde{y}(k)$ can be eliminated using (20), the relaxed problem has both $\tilde{u}(k)$ and $\tilde{y}(k)$ as independent variables. It is easy to verify that the set of feasible solutions of the relaxed problem coincides with the set of solutions of the system of linear inequalities (25), (27), (28). So the feasible set of the relaxed MPC problem is convex. Hence, the relaxed problem is much easier to solve numerically.

A function $F : y \to F(y)$ is a monotonically nondecreasing function if $\bar{y} \leqslant \check{y}$ implies that $F(\bar{y}) \leqslant F(\check{y})$. Now we show that if the objective function $J$ and the linear constraints are monotonically nondecreasing as a function of $\tilde{y}$ (this is the case for $J = J_{\mathrm{out},1}$, $J_{\mathrm{in},0}$, $J_{\mathrm{in},1}$, or $J_{\mathrm{in},2}$, and e.g. $F_{ij} \geqslant 0$ for all $i, j$), then the optimal solution of the relaxed problem can be transformed into an optimal solution of the original MPC problem. So in that case the optimal MPC policy can be computed very efficiently. If in addition the objective function is convex (e.g. $J = J_{\mathrm{out},1}$ or $J_{\mathrm{in},1}$), we finally get a convex optimization problem. Note that $J_{\mathrm{in},1}$ is a linear function. So for $J = J_{\mathrm{in},1}$ the problem even reduces to a linear programming problem, which can be solved very efficiently [2].

[2] It easy to verify that for $J = J_{\mathrm{out},1}$, $J_{\mathrm{out},11}$, $J_{\mathrm{out},12}$ the problem can also be reduced to a linear programming problem by introducing some additional dummy variables.

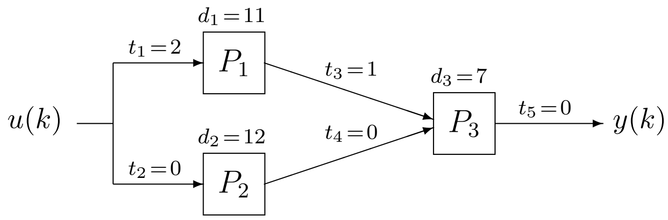


Fig. 1. A simple manufacturing system.

**Theorem 2** *Let the objective function $J$ and mapping $\tilde{y} \to F(k)\tilde{y}$ be monotonically nondecreasing functions of $\tilde{y}$. Let $(\tilde{u}^*, \tilde{y}^*)$ be an optimal solution of the relaxed MPC problem. If we define $\tilde{y}^\sharp = H \otimes \tilde{u}^* \oplus g(k)$ then $(\tilde{u}^*, \tilde{y}^\sharp)$ is an optimal solution of the original MPC problem.*

**PROOF.** First we show that $(\tilde{u}^*, \tilde{y}^\sharp)$ is a feasible solution of the original problem. Clearly, $(\tilde{u}^*, \tilde{y}^\sharp)$ satisfies (20), (22) and (23). Since $\tilde{y}^* \geqslant H \otimes \tilde{u}^* \oplus g(k) = \tilde{y}^\sharp$ and since $\tilde{y} \to F(k)\tilde{y}$ is monotonically nondecreasing, we have

$$E(k)\tilde{u}^* + F(k)\tilde{y}^\sharp \leqslant E(k)\tilde{u}^* + F(k)\tilde{y}^* \leqslant h(k) \ .$$

So $(\tilde{u}^*, \tilde{y}^\sharp)$ also satisfies the constraint (21). Hence, $(\tilde{u}^*, \tilde{y}^\sharp)$ is a feasible solution of the original problem. Since the set of feasible solutions of the original problem is a subset of the set of feasible solutions of the relaxed problem, we have $J(\tilde{u}, \tilde{y}) \geqslant J(\tilde{u}^*, \tilde{y}^*)$ for any feasible solution $(\tilde{u}, \tilde{y})$ of the original problem. Hence, $J(\tilde{u}^*, \tilde{y}^\sharp) \geqslant J(\tilde{u}^*, \tilde{y}^*)$. On the other hand, we have $J(\tilde{u}^*, \tilde{y}^\sharp) \leqslant J(\tilde{u}^*, \tilde{y}^*)$ since $\tilde{y}^\sharp \leqslant \tilde{y}^*$ and since $J$ is a monotonically nondecreasing function of $\tilde{y}$. As a consequence, we have $J(\tilde{u}^*, \tilde{y}^\sharp) = J(\tilde{u}^*, \tilde{y}^*)$, which implies that $(\tilde{u}^*, \tilde{y}^\sharp)$ is an optimal solution of the original MPC problem. $\square$

## 6 Example

Consider the production system of Fig. 1. This manufacturing system consists of three processing units: $P_1$, $P_2$ and $P_3$, and works in batches (one batch for each finished product). Raw material is fed to $P_1$ and $P_2$, processed and sent to $P_3$ where assembly takes place. The processing times for $P_1$, $P_2$ and $P_3$ are respectively $d_1 = 11$, $d_2 = 12$ and $d_3 = 7$ time units. It takes $t_1 = 2$ time units for the raw material to get from the input source to $P_1$, and $t_3 = 1$ time unit for a finished product of $P_1$ to get to $P_3$. The other transportation times and the set-up times are assumed to be negligible. A processing unit can only start working on a new product if it has finished processing the previous product. Each processing unit starts working as soon as all parts are available.

The system is described by the following state space model (De Schutter and van den Boom, 2000):

$$x(k+1) = \begin{bmatrix} 11 & \varepsilon & \varepsilon \\ \varepsilon & 12 & \varepsilon \\ 23 & 24 & 7 \end{bmatrix} \otimes x(k) \ \oplus \ \begin{bmatrix} 2 \\ 0 \\ 14 \end{bmatrix} \otimes u(k) \tag{29}$$

$$y(k) = \begin{bmatrix} \varepsilon & \varepsilon & 7 \end{bmatrix} \otimes x(k) \tag{30}$$

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 $k$th time, and $y(k)$ the time at which the $k$th finished product leaves the system.

Let us now compare the efficiency of the methods discussed in Section 5 when solving one step of the MPC problem for the objective function $J = J_{\text{out},1} + J_{\text{in},1}$ (so $\lambda = 1$) with the additional constraints $2 \leqslant \Delta u(k+j) \leqslant 12$ for $j = 0, \ldots, N_{\text{c}} - 1$. We take $N_{\text{c}} = 5$ and $N_{\text{p}} = 8$. Assume that $k = 0$, $x(0) = [\ 0 \ \ 0 \ \ 10\ ]^T$, $u(-1) = 0$, and $\tilde{r}(k) = [\ 40 \ \ 45 \ \ 55 \ \ 66 \ \ 75 \ \ 85 \ \ 90 \ \ 100\ ]^T$.

The objective function $J$ and the linear constraints are monotonically nondecreasing as a function of $\tilde{y}$ so that we can apply Theorem 2. We have computed a solution $\tilde{u}_{\text{elcp}}$ obtained using the ELCP method and the ELCP algorithm of (De Schutter and De Moor, 1995), a solution $\tilde{u}_{\text{nlcon}}$ using nonlinear constrained optimization, a solution $\tilde{u}_{\text{penalty}}$ using linearly constrained optimization with a penalty function for the nonlinear constraints, a solution $\tilde{u}_{\text{relaxed}}$ for the relaxed MPC problem, and a linear programming solution $\tilde{u}_{\text{linear}}$ (cf. footnote 2). For the nonlinear constrained optimization we have used a sequential quadratic programming algorithm, and for the linear optimization a variant of the simplex algorithm. All these methods result in the same optimal input sequence:

$$\{u_{\text{opt}}\}_{k=0}^{7} = 12, 24, 35, 46, 58, 70, 82, 94.$$

The corresponding output sequence is $\{y_{\text{opt}}(k)\}_{k=1}^{8} = 33, 45, 56, 67, 79, 91, 103, 115$ and the corresponding value of the objective function is $J = -381$.

In Table 1 we have listed the CPU time needed to compute the various input sequence vectors $\tilde{u}$ for $N_{\text{c}} = 4, 5, 6, 7$ on a Pentium II 300 MHz PC with the optimization routines called from MATLAB and implemented in C (average values over 10 experiments). For the input sequence vectors that have to be determined using a nonlinear optimization algorithm selecting different (feasible) initial points always leads to the same numerical value of the final objective function (within a certain tolerance). Therefore, we have only performed one run with a random feasible initial point for each of these cases.

The CPU time for the ELCP algorithm of (De Schutter and De Moor, 1995) increases exponentially as the number of variables increases (see also Table 1).

Table 1
The CPU time needed to compute the optimal input sequence vectors for the example of Section 6 for $N_c = 4, 5, 6, 7$. For $N_c = 7$ we have not computed the ELCP solution since it requires too much CPU time.

| $\tilde{u}_{\text{opt}}$ | CPU time | | | |
|---|---|---|---|---|
| | $N_c = 4$ | $N_c = 5$ | $N_c = 6$ | $N_c = 7$ |
| $\tilde{u}_{\text{elcp}}$ | 5.525 | 106.3 | 287789 | — |
| $\tilde{u}_{\text{nlcon}}$ | 0.870 | 1.056 | 1.319 | 1.470 |
| $\tilde{u}_{\text{penalty}}$ | 0.826 | 0.988 | 1.264 | 1.352 |
| $\tilde{u}_{\text{relaxed}}$ | 0.431 | 0.500 | 0.562 | 0.634 |
| $\tilde{u}_{\text{linear}}$ | 0.029 | 0.030 | 0.031 | 0.032 |

So in practice the ELCP approach cannot be used for on-line computations if the control horizon or the number of inputs or outputs are large. In that case one of the other methods should be used instead. If we look at Table 1 then we see that the $\tilde{u}_{\text{linear}}$ solution — which is based on Theorem 2 and a linear programming approach — is clearly the most interesting.

Let us now compare the MPC-MPL method with the other control design methods mentioned in Section 4.5. In (De Schutter and van den Boom, 2000) we have used results from (Baccelli *et al.*, 1992) to derive an analytic solution for two special cases of the MPL-MPC problem. If we use these analytic solutions we obtain $\{u^1(k)\}_{k=0}^{7} = -3, 9, 21, 33, 45, 57, 68, 79$ and $\{u^2(k)\}_{k=0}^{7} = 8, 20, 32, 44, 56, 68, 79, 90$. The first solution is not feasible since for this solution we have $u(0) = -3 < 0 = u(-1)$. This infeasibility is caused by the fact that the solution aims to fulfill the constraint $\tilde{y}(k) \leqslant \tilde{r}(k)$, which cannot be met using a nondecreasing input sequence. So other control design methods that also include this constraint such as (Libeaut and Loiseau, 1995; Menguy *et al.*, 1997; Menguy *et al.*, 1998*b*) would also yield a nondecreasing — and thus infeasible — input sequence. The second solution is feasible and the corresponding value of the objective function is $J = -358$. The control design method of (Boimond and Ferrier, 1996) leads to $\{u^3(k)\}_{k=0}^{7} = 19, 30, 41, 53, 65, 77, 89, 101$, but this input sequence does not satisfy the constraints since $\Delta u(0) = u(0) - u(-1) = 19 \nleqslant 12$. The method of (Menguy *et al.*, 1998*a*) results in $\{u^4(k)\}_{k=0}^{7} = 12, 23, 34, 46, 58, 70, 82, 94$ with $J = -380$.

So for this particular case the MPC method and the method of (Menguy *et al.*, 1998*a*) outperform the other methods to compute (optimal) input time sequences for MPL systems. However, the method of (Menguy *et al.*, 1998*a*) does not take the input constraint $2 \leqslant \Delta u(k+j) \leqslant 12$ for $j = 0, 1, \ldots, 9$ into account so that in general this method will not always lead to a feasible solution. So in general the MPC-MPL method is the only method among the methods considered in this paper that is guaranteed to yield a feasible solution (provided that one exists).

## 7 Conclusions

We have extended the popular MPC framework from linear discrete-time systems to max-plus-linear discrete event systems. The reason for using an MPC approach for max-plus-linear systems is the same as for conventional linear systems: MPC allows the inclusion of constraints on the inputs and outputs, it is an easy-to-tune method, and it is flexible for structure changes (since the optimal strategy is recomputed every time or event step so that model changes can be taken into account as soon as they are identified). Although in general the optimization may be complex and time-consuming and should be performed each event step, the inter-event times are usually sufficiently long so that the calculation can be performed on-line (especially if the objective function and the constraints are monotonically nondecreasing and if the objective function is convex, since then the resulting (relaxed) optimization problem is convex). We have also presented some methods to solve the max-plus-algebraic MPC problem. In general this leads to a nonlinear nonconvex optimization problem. If the objective function and the constraints are nondecreasing functions of the output, then we can relax the MPC problem to problem with a convex set of feasible solutions. If in addition the objective function is convex, this leads to a problem that can be solved very efficiently.

Topics for future research include: investigation of issues (such as prediction) that arise when we consider MPC for nondeterministic max-plus-algebraic systems, investigation of the effects of the tuning parameters on the stability of the controlled system, and determination of rules of thumb for the selection of appropriate values for the tuning parameters.

## Acknowledgements

Research partially sponsored by TMR project ALAPEDES and FWO-ICCoS.

## References

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

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

Camacho, E.F. and C. Bordons (1995). *Model Predictive Control in the Process Industry.* Springer-Verlag. Berlin, Germany.

De Schutter, B. and B. De Moor (1995). The extended linear complementarity problem. *Mathematical Programming* **71**(3), 289–325.

De Schutter, B. and T. van den Boom (2000). Model predictive control for max-plus-linear discrete-event systems: Extended report & Addendum. Technical Report bds:99-10a. Systems and Control Engineering, Fac. of Information Technology and Systems, Delft University of Technology. Delft, The Netherlands. A short version of this report has been accepted for publication in *Automatica*, vol. 37, no. 7, pp. 1049–1056, July 2001.

García, C.E., D.M. Prett and M. Morari (1989). Model predictive control: Theory and practice — A survey. *Automatica* **25**(3), 335–348.

Libeaut, L. and J.J. Loiseau (1995). Admissible initial conditions and control of timed event graphs. In: *Proceedings of the 34th IEEE Conference on Decision and Control.* New Orleans, Louisiana. pp. 2011–2016.

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

Menguy, E., J.L. Boimond and L. Hardouin (1998*a*). Adaptive control for linear systems in max-algebra. In: *Proceedings of the International Workshop on Discrete Event Systems (WODES'98).* Cagliari, Italy. pp. 481–488.

Menguy, E., J.L. Boimond and L. Hardouin (1998*b*). Optimal control of discrete event systems in case of updated reference input. In: *Proceedings of the IFAC Conference on System Structure and Control (SSC'98).* Nantes, France. pp. 601–607.