

Technical report bds:99-09

# **Model predictive control for max-plus-linear 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 systems," *Proceedings of the 2000 American Control Conference*, Chicago, Illinois, pp. 4046–4050, June 2000.

Control Systems Engineering  
Faculty of Information Technology and Systems  
Delft University of Technology  
Delft, The Netherlands  
Current URL: <https://www.dcsc.tudelft.nl>

---

\*This report can also be downloaded via [https://pub.deschutter.info/abs/99\\_09.html](https://pub.deschutter.info/abs/99_09.html)

# Model predictive control for max-plus-linear systems

Bart De Schutter and Ton van den Boom

Control Lab, Fac. Information Techn. and Systems, Delft University of Technology

P.O.Box 5031, 2600 GA Delft, The Netherlands

{b.deschutter,t.j.j.vandenboom}@its.tudelft.nl

## Abstract

Model predictive control (MPC) is a very popular controller design method in the process industry. An important advantage of MPC is that it allows the inclusion of 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, i.e. we present an MPC framework for max-plus-linear systems. In general the resulting optimization problem is nonlinear and nonconvex. However, if the control objective and the constraints depend monotonically on the outputs of the system, the MPC 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.

## 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 process industry. An important advantage of MPC is that the use of a finite horizon allows the inclusion of additional constraints on the inputs and outputs.

Traditionally MPC uses linear discrete-time models for the process that has 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 [1, 5]. We will develop an MPC framework for max-plus-linear discrete event systems.

## 2 Model predictive control

In this section we give a short introduction to MPC for linear discrete-time systems. Since we will only consider the

deterministic, i.e. noiseless, case for max-plus-linear systems, we will also omit the noise terms in this introduction to MPC. More extensive information on MPC can be found in [3, 4, 8, 10].

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

$$x(k+1) = Ax(k) + Bu(k) \quad (1)$$

$$y(k) = Cx(k) \quad (2)$$

A system that can be modeled by (1)–(2) will be called a plus-times-linear (PTL) model since the basic operations in this model are addition and multiplication.

Define  $\tilde{u}(k) = [u^T(k) \dots u^T(k + N_p - 1)]^T$  and  $\tilde{y}(k) = [\hat{y}^T(k+1|k) \dots \hat{y}^T(k+N_p|k)]^T$  where  $\hat{y}(k+j|k)$  is the estimate of the output at time  $k+j$  based on the information available at time  $k$  and  $N_p$  is the prediction horizon. In MPC a performance index or cost criterion  $J$  is formulated that reflects the reference tracking error ( $J_{\text{out}}$ ) and the control effort ( $J_{\text{in}}$ ) [3]:

$$\begin{aligned} J &= J_{\text{out}} + \lambda J_{\text{in}} \\ &= \sum_{j=1}^{N_p} (\tilde{y}(k) - \tilde{r}(k))^T (\tilde{y}(k) - \tilde{r}(k)) + \lambda \tilde{u}^T(k) \tilde{u}(k) \end{aligned} \quad (3)$$

where  $\tilde{r}$  is a reference signal and  $\lambda$  is a nonnegative scalar. In MPC the input is taken to be constant from a certain point on:  $u(k+j) = u(k+N_c-1)$  for  $j = N_c, \dots, N_p-1$  where  $N_c$  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).

The MPC problem is defined as follows:

Find at each time instant  $k$  the input sequence  $\{u(k), \dots, u(k+N_c-1)\}$  that minimizes the performance index  $J$  subject to the linear constraint

$$E(k)\tilde{u}(k) + F(k)\tilde{y}(k) \leq h(k) \quad (4)$$

with  $E(k) \in \mathbb{R}^{p \times mN_p}$ ,  $F(k) \in \mathbb{R}^{p \times lN_p}$ ,  $h(k) \in \mathbb{R}^p$  for some integer  $p$ , and subject to the control horizon constraint  $u(k+j) = u(k+N_c-1)$  for  $j = N_c, \dots, N_p-1$ .

MPC uses a receding horizon principle. This means that after computation of the optimal control sequence  $\{u(k), \dots, u(k+N_c-1)\}$ , only the first control sample  $u(k)$  will be implemented, subsequently the horizon is shifted one sample and the optimization is restarted with new information of the measurements.

By successive substitution of (1), estimates of the future values of the output can be computed [3, 10], which leads to  $\tilde{y}(k) = H\tilde{u}(k) + g(k)$  with

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

The parameters  $N_p$ ,  $N_c$  and  $\lambda$  are the three basic tuning parameters of the MPC algorithm:

- The prediction horizon  $N_p$  is related to the length of the step response of the process, and the time interval  $[1, N_p]$  should contain the crucial dynamics of the process.
- The control horizon  $N_c \leq N_p$  forces the control signal to a constant value ( $u(k+j) = u(k+N_c-1)$  for  $j = N_c, \dots, N_p$ ). An important effect of a small control horizon ( $N_c \ll N_p$ ) is the smoothing of the control signal (which can become very wild if  $N_c = N_p$ ). The control signal is then rapidly forced towards its steady-state value, which is important for stability properties. Another important consequence of decreasing  $N_c$  is the reduction in computational effort, because the number of optimization parameters reduces. Typically  $N_c$  is taken equal to the model order of the system.
- The parameter  $\lambda$  makes a trade-off between the objectives of tracking error ( $J_{\text{out}}$ ) and control effort ( $J_{\text{in}}$ ). The parameter  $\lambda$  is usually chosen as small as possible, 0 in most cases. In many cases (e.g. for nonminimum phase systems), the choice  $\lambda = 0$  will lead to stability problems and so  $\lambda$  should be chosen as the smallest positive value that still results in 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 (represented by  $\oplus$ ) and addition (represented by  $\otimes$ ):

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

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. 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  $\otimes$  (see [1, 5]). The  $k$ th max-plus-algebraic power of  $x$  is denoted by  $x^{\otimes k}$  and corresponds to  $kx$  in conventional algebra.

The basic max-plus-algebraic operations are extended to matrices as follows. If  $A, B \in \mathbb{R}_\varepsilon^{m \times n}$  and  $C \in \mathbb{R}_\varepsilon^{n \times p}$  then

$$(A \oplus B)_{ij} = a_{ij} \oplus b_{ij} \quad \text{and} \quad (A \otimes C)_{ij} = \bigoplus_{k=1}^n a_{ik} \otimes c_{kj}$$

for all  $i, j$ . The max-plus-algebraic matrix power of  $A \in \mathbb{R}_\varepsilon^{n \times n}$  is defined as follows:  $A^{\otimes k} = A \otimes A \otimes \dots \otimes A$  ( $k$  times).

#### 3.2 Max-plus-linear systems

In [1, 5] it has been shown that there is a class of discrete event systems that can be modeled by a max-plus-algebraic model of the following form:

$$x(k+1) = A \otimes x(k) \oplus B \otimes u(k) \quad (5)$$

$$y(k) = C \otimes x(k) \quad (6)$$

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. The index  $k$  is called the event counter. For a manufacturing system,  $u(k)$  would typically represent the time instants at which raw material is fed to the system for the  $(k+1)$ th time,  $x(k)$  the time instants at which the machines start processing the  $k$ th batch of intermediate products, and  $y(k)$  the time instants at which the  $k$ th batch of finished products leaves the system. Note the analogy between the models (1)–(2) and (5)–(6). A discrete event system that can be modeled by (5)–(6) will be called a *max-plus-linear system* (MPL system) since the basic operations in this model are maximization and addition.

### 4 Model predictive control for MPL systems

In this section we extend and adapt the MPC framework from PTL systems to MPL systems. If possible we use analog constraints and cost criteria for both types of systems. However, as we shall see, in some cases different constraints and cost criteria are more appropriate.

#### 4.1 Evolution of the system

If we know the state of the system  $x(k)$  then for a given input sequence  $\{u(k), \dots, u(k+N_p-1)\}$  we can determine the estimates for the outputs of the system as follows. If we define  $H$  and  $g(k)$  as in Section 2 but with the conventional matrix product and zero matrix replaced by the max-plus-algebraic matrix product and the max-plus-algebraic zero matrix<sup>1</sup> respectively, we have  $\tilde{y}(k) = H \otimes \tilde{u}(k) \oplus g(k)$ .

#### 4.2 Cost criterion

Recall that the MPC cost criterion for PTL systems can be written as  $J = J_{\text{out}} + \lambda J_{\text{in}}$ , where  $J_{\text{out}}$  is related to the tracking error and  $J_{\text{in}}$  is related to the control effort. Now we discuss some possible choices for  $J_{\text{out}}$  and  $J_{\text{in}}$  for MPL systems.

<sup>1</sup>I.e., a matrix for which all entries are equal to  $\varepsilon$ .

#### 4.2.1 Tracking error or output cost criterion $J_{\text{out}}$ :

A straightforward translation of the tracking error cost criterion used in MPC for PTL systems would yield

$$\begin{aligned} J_{\text{out}} &= (\tilde{y}(k) - \tilde{r}(k))^T \otimes (\tilde{y}(k) - \tilde{r}(k)) \\ &= 2 \bigoplus_{j=1}^{N_p} \bigoplus_{i=1}^l (\hat{y}_i(k+j|k) - r_i(k+j)) . \end{aligned} \quad (7)$$

Note that this objective function does not force the difference between  $\hat{y}(k+j|k)$  and  $r(k+j)$  to be small since there is no absolute value in (7). If the due dates  $r$  for the finished products are known and if we have to pay a penalty for every delay, a better suited cost criterion is the tardiness:

$$J_{\text{out},1} = \sum_{j=1}^{N_p} \sum_{i=1}^l \max(\hat{y}_i(k+j|k) - r_i(k+j), 0) .$$

For perishable goods we want to minimize the differences between the due dates and the actual output time instants:

$$J_{\text{out},2} = \sum_{j=1}^{N_p} \sum_{i=1}^l |\hat{y}_i(k+j|k) - r_i(k+j)| .$$

If we want to balance the output rates, we could consider the following cost criterion:

$$J_{\text{out},3} = \sum_{j=2}^{N_p} \sum_{i=1}^l |\Delta^2 \hat{y}_i(k+j|k)| ,$$

where  $\Delta$  is the difference operator.

**4.2.2 Input cost criterion  $J_{\text{in}}$ :** 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 internal buffer overflows, a better objective is to *maximize* the input time instants. For a manufacturing system, this would correspond to a just-in-time production 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. Note that 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

$$J_{\text{in},0} = -\tilde{u}^T(k)\tilde{u}(k) .$$

Other objective functions that lead to a maximization of the input time instants are

$$J_{\text{in},1} = -\sum_{j=1}^{N_p} \sum_{i=1}^m u_i(k+j-1)$$

or

$$J_{\text{in},2} = \sum_{j=1}^{N_p} \sum_{i=1}^m |(\max_{i=1,\dots,l} \hat{y}_i(k+j|k)) - u_i(k+j-1)| ,$$

which minimizes the differences between the input time instants and the last output time instant of each batch. If we want to balance the input rates we could take

$$J_{\text{in},3} = \sum_{j=1}^{N_p-1} \sum_{i=1}^l |\Delta^2 u_i(k+j)| .$$

#### 4.3 Constraints

A straightforward translation of the linear constraint (4) used in MPC for PTL systems yields

$$E(k) \otimes \tilde{u}(k) \oplus F(k) \otimes \tilde{y}(k) \leq h(k) . \quad (8)$$

Typical other constraints are:

$$a_1(k+j) \leq \Delta u(k+j-1) \leq b_1(k+j) \quad (9)$$

$$a_2(k+j) \leq \Delta \hat{y}(k+j|k) \leq b_2(k+j) \quad (10)$$

$$\hat{y}(k+j|k) \leq r(k+j) . \quad (11)$$

Let us now show that all these constraints can be recast as a linear constraint of the form

$$A_c(k)\tilde{u}(k) + B_c(k)\tilde{y}(k) \leq c_c(k) \quad (12)$$

for appropriately defined matrices and vectors  $A_c(k)$ ,  $B_c(k)$ ,  $c_c(k)$ . It is easy to verify that the constraints (9)–(11) can be written in this form. If we eliminate the output estimates from (8) and if we take into account that each term in the resulting max-plus-algebraic summation has to be less than or equal to  $h(k)$ , we obtain a set of conditions of the form

$$P \otimes u(k+i) \leq h(k) . \quad (13)$$

Note that there is also a constant term involving  $x(k)$  but if this term is not less than or equal to  $h(k)$  the problem is infeasible, and otherwise this constant term can be omitted. It is easy to verify that (13) is satisfied if and only if  $p_{vw} + u_w(k+i) \leq h_v(k)$  for all  $v, w$ . Since this is a constraint of the form (12), (8) can also be recast as a linear constraint of the form (12).

Since for MPL systems the input sequence corresponds to occurrence times of consecutive events, it should always be nondecreasing. Therefore, we also have to add the condition  $\Delta u(k+j) \geq 0$  for  $j = 0, \dots, N_p - 1$ .

#### 4.4 Evolution of the input beyond the control horizon

In MPC for PTL systems the input should stay constant from the certain point  $k + N_c$  on. For MPL systems such a condition would not be very useful since the input sequences should normally be increasing. Therefore, we change this condition as follows: the feeding rate should stay constant beyond event step  $k + N_c$ , i.e.  $\Delta u(k+j) = \Delta u(k + N_c - 1)$  for  $j = N_c, \dots, N_p - 1$ , or equivalently  $\Delta^2 u(k+j) = 0$  for  $j = N_c, \dots, N_p - 1$ .

#### 4.5 The standard MPC problem for MPL systems

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

$$\min_{\tilde{u}(k)} J \quad (14)$$

$$\text{subject to } \tilde{y} = H \otimes \tilde{u} \oplus g(k) \quad (15)$$

$$A_c(k)\tilde{u}(k) + B_c(k)\tilde{y}(k) \leq c_c(k) \quad (16)$$

$$\Delta u(k+j) \geq 0 \quad \text{for } j = 0, \dots, N_p - 1 \quad (17)$$

$$\Delta^2 u(k+j) = 0 \quad \text{for } j = N_c, \dots, N_p - 1, \quad (18)$$

which will be called the max-plus-algebraic MPC problem for event step  $k$ .

Other design control design methods for MPL systems are discussed in [1, 2, 5, 9]. Note that in contrast to the max-plus-algebraic MPC method, these methods do not allow the inclusion of constraints of the form (16), (17) or (18).

## 5 Algorithms for the max-plus-algebraic MPC problem

### 5.1 The ELCP approach

In general the problem (14)–(18) is a nonlinear nonconvex optimization problem. Note that although the constraints (16)–(18) are convex in  $\tilde{u}$  and  $\tilde{y}$ , the constraint (15) is in general not convex. We could use standard multi-start nonlinear nonconvex local optimization methods to compute the optimal MPC policy. In [7] we show that the set of feasible solutions defined by (15)–(18) coincides with the solution set of an Extended Linear Complementarity problem (ELCP) [6]. In [6] 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 have to determine for which values of the parameters the objective function  $J$  over the solution set of the ELCP that corresponds to (15)–(18) reaches its global minimum. The algorithm of [6] to compute the solution set of a general ELCP requires exponential execution times. This implies that the ELCP approach sketched above is not feasible if  $N_c$ ,  $m$  or  $l$  are large.

### 5.2 Monotonically nondecreasing objective functions and constraints

Now we consider the *relaxed MPC problem* which is also defined by (14)–(18) but with the =-sign in (15) replaced by a  $\geq$ -sign. Note that the set of feasible solutions of the relaxed MPC problem is convex. As a consequence, the relaxed problem is much easier to solve numerically.

We say that a function  $F$  is a monotonically nondecreasing function of  $y$  if  $\bar{y} \leq \check{y}$  implies that  $F(\bar{y}) \leq F(\check{y})$ . If the objective function  $J$  and the linear constraints are monotonically nondecreasing as function of  $\tilde{y}$  (this is the case for  $J = J_{\text{out},1}$ ,  $J_{\text{in},0}$ ,  $J_{\text{in},1}$ ,  $J_{\text{in},3}$ , or their variants in which one or more summations are replaced by max-plus-algebraic summations, and e.g.  $(B_c)_{ij} \geq 0$  for all  $i, j$ ), then the optimal solution of the relaxed MPC problem can be transformed into a solution of the original MPC problem:

**Theorem 5.1** *Let the objective function  $J$  and mapping  $\tilde{y} \rightarrow B_c(k)\tilde{y}$  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 MPC problem. Clearly,  $(\tilde{u}^*, \tilde{y}^\sharp)$  satisfies the constraints (15), (17) and (18). Since  $\tilde{y}^* \geq H \otimes \tilde{u}^* \oplus g(k) = \tilde{y}^\sharp$  and since the mapping  $\tilde{y} \rightarrow B_c(k)\tilde{y}$  is monotonically nondecreasing, we have

$$A_c(k)\tilde{u}^* + B_c(k)\tilde{y}^\sharp \leq A_c(k)\tilde{u}^* + B_c(k)\tilde{y}^* \leq c_c(k) .$$

So  $(\tilde{u}^*, \tilde{y}^\sharp)$  also satisfies the constraint (16). Hence,  $(\tilde{u}^*, \tilde{y}^\sharp)$  is a feasible solution of the original MPC problem. Since the set of feasible solutions of the original MPC problem is a subset of the set of feasible solutions of the relaxed MPC problem, we have  $J(\tilde{u}, \tilde{y}) \geq J(\tilde{u}^*, \tilde{y}^*)$  for any feasible solution  $(\tilde{u}, \tilde{y})$  of the original problem. Hence,  $J(\tilde{u}^*, \tilde{y}^\sharp) \geq J(\tilde{u}^*, \tilde{y}^*)$ . On the other hand, we have  $J(\tilde{u}^*, \tilde{y}^\sharp) \leq J(\tilde{u}^*, \tilde{y}^*)$  since  $\tilde{y}^\sharp \leq \tilde{y}^*$  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. ■

So if Theorem 5.1 applies the optimal MPC policy can be computed very efficiently. If in addition the objective function is convex (e.g.  $J = J_{\text{out},1}$  or its variants, or  $J = J_{\text{in},1}$ ), we finally get a convex optimization problem. Since  $J_{\text{in},1}$  is a linear function, the problem even reduces to a linear programming problem for  $J = J_{\text{in},1}$ , which can be solved very efficiently.

Note that we can always obtain an objective function that is a monotonically nondecreasing by eliminating  $\tilde{y}(k)$  from the expression for  $J$  using (15) before relaxing the problem. However, some properties (such as convexity or linearity) of the original objective function may be lost in that way.

## 6 Example

We consider a production system that can be modeled by the following state space model (see [7] for more information):

$$x(k+1) = \begin{bmatrix} 9 & \varepsilon & \varepsilon \\ \varepsilon & 10 & \varepsilon \\ 18 & 20 & 7 \end{bmatrix} \otimes x(k) \oplus \begin{bmatrix} 2 \\ 0 \\ 11 \end{bmatrix} \otimes u(k)$$

$$y(k) = [ \varepsilon \quad \varepsilon \quad 7 ] \otimes x(k) .$$

Let us now compare the efficiency of the several methods discussed in Section 5 when solving one step of the MPC problem for the objective function  $J = J_{\text{out},1} + 0.1J_{\text{in},1}$  with the additional constraints  $1 \leq \Delta u(k+j) \leq 10$  for  $j = 0, \dots, N_c - 1$ . We take  $N_c = 4$  and  $N_p = 6$ . Assume that  $k = 0$ ,  $x(0) = [ 0 \quad 0 \quad 11 ]^T$ ,  $u(0) = 0$ ,  $r = [ 35 \quad 40 \quad 45 \quad 55 \quad 65 \quad 85 ]^T$ . Note that the objective function  $J$  and the linear constraints are monotonically nondecreasing as a function of  $y$  so that we can apply Theorem 5.1.

**Table 1:** The values of the objective function  $J$  (for  $N_c = 4$ ) and the CPU time (for  $N_c = 4, 5, 6$ ) needed to compute the optimal input sequence vectors for the example of Section 6.

$\tilde{u}_{\text{opt}}$	$J(\tilde{u}_{\text{opt}})$ ( $N_c = 4$ )	CPU time		
		$N_c = 4$	$N_c = 5$	$N_c = 6$
$\tilde{u}_{\text{elcp}}$	-14.600	1.765	3.320	7.535
$\tilde{u}_{\text{nlcon}}$	-14.597	8.842	10.415	11.160
$\tilde{u}_{\text{penalty}}$	-14.600	10.864	12.078	14.170
$\tilde{u}_{\text{relaxed}}$	-14.600	0.301	0.342	0.430
$\tilde{u}_{\text{lp}}$	-14.600	0.017	0.019	0.019

We have computed a solution  $\tilde{u}_{\text{elcp}}$  obtained using the ELCP method and the ELCP algorithm of [6], 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  $\tilde{u}_{\text{lp}}$ . For the nonlinear constrained optimization we have used a sequential quadratic programming algorithm, and for the linear optimization a variant of the simplex algorithm. In the second and the third column of Table 1 we have listed the value of the objective function  $J$  for the various input sequence vectors  $\tilde{u}$  and the CPU time needed to compute them on a Pentium II 300 MHz PC running Linux with the optimization routines called from MATLAB and implemented in C. We have also listed the CPU times for  $N_c = 5$  and  $N_c = 6$  (with all other variables keeping the same values as above). The CPU time values listed in the table are average values over 10 experiments. For  $\tilde{u}_{\text{nlcon}}$  and  $\tilde{u}_{\text{penalty}}$  we have listed the best solution over 10 runs<sup>2</sup> with random initial points; the indicated CPU time is the time needed for the 10 runs. For the optimization over the solution set of the ELCP and for  $\tilde{u}_{\text{relaxed}}$  different starting points always lead to more or less the same numerical value of the final objective function. Therefore, we have only performed one run with an arbitrary random initial point for these cases.

The CPU time needed to compute the optimal switching interval vector using the ELCP algorithm of [6] increases exponentially as the number of variables increases (see also Table 1). So the ELCP approach cannot be used on-line in practice 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  $\tilde{u}_{\text{lp}}$  solution — which is based on Theorem 5.1 — is clearly the most interesting.

<sup>2</sup>This choice for the number of runs is based on the typical variation in the value of the final values of objective values for the different runs.

## 7 Conclusion

In this paper 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 step or event step so that model changes can be taken into account as soon as they are identified).

We have extensively discussed the analogies and differences between the objective functions and constraints in the conventional MPC problem and in the max-plus-algebraic MPC problem. 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 monotonically 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 or linear, this leads to a problem that can be solved very efficiently.

Topics for future research include the extension of the current MPC framework to nondeterministic max-plus-algebraic models, and an elaborate determination of the influences of the tuning parameters  $\lambda$ ,  $N_p$  and  $N_c$ .

## References

- [1] F. Baccelli, G. Cohen, G.J. Olsder, and J.P. Quadrat, *Synchronization and Linearity*. New York: John Wiley & Sons, 1992.
- [2] J.L. Boimond and J.L. Ferrier, "Internal model control and max-algebra: Controller design," *IEEE Trans. on Aut. Control*, vol. 41, no. 3, pp. 457–461, Mar. 1996.
- [3] D.W. Clarke, C. Mohtadi, and P.S. Tuffs, "Generalized predictive control – Part I. The basic algorithm," *Automatica*, vol. 23, no. 2, pp. 137–148, Mar. 1987.
- [4] D.W. Clarke, C. Mohtadi, and P.S. Tuffs, "Generalized predictive control – Part II. Extensions and interpretations," *Automatica*, vol. 23, no. 2, pp. 149–160, Mar. 1987.
- [5] R.A. Cuninghame-Green, *Minimax Algebra*, vol. 166 of *Lect. Notes in Economics and Math. Systems*. Berlin, Germany: Springer-Verlag, 1979.
- [6] B. De Schutter and B. De Moor, "The extended linear complementarity problem," *Mathematical Programming*, vol. 71, no. 3, pp. 289–325, Dec. 1995.
- [7] B. De Schutter and T. van den Boom, "Model predictive control for max-plus-linear discrete event systems," Tech. rep. bds-99-10, Control Lab, Fac. ITS, Delft Univ. of Techn., Delft, The Netherlands, Sept. 1999.
- [8] C.E. García, D.M. Prett, and M. Morari, "Model predictive control: Theory and practice — A survey," *Automatica*, vol. 25, no. 3, pp. 335–348, May 1989.

- [9] M. Gazarik and Y. Wardi, "Optimal release times in a single server: An optimal control perspective," *IEEE Trans. on Aut. Control*, vol. 43, no. 7, pp. 998–1002, Jul. 1998.
- [10] M. Kinnaert, "Adaptive generalized predictive controller for MIMO systems," *Int. Journal of Control*, vol. 50, no. 1, pp. 161–172, July 1989.