Delft University of Technology

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

Technical report bds:00-14

MPC for perturbed max-plus-linear systems*

T.J.J. van den Boom and B. De Schutter

If you want to cite this report, please use the following reference instead:
T.J.J. van den Boom and B. De Schutter, "MPC for perturbed max-plus-linear systems," *Proceedings of the European Control Conference 2001 (ECC'01)*, Porto, Portugal, pp. 3783–3788, Sept. 2001.

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.bartdeschutter.org/abs/00_14

MPC FOR PERTURBED MAX-PLUS-LINEAR SYSTEMS

Ton J.J. van den Boom, Bart De Schutter
Control Lab, Fac. ITS, Delft University of Technology
P.O.Box 5031, 2600 GA Delft, The Netherlands
Phone: +31-15-2784052/5113, Fax: +31-15-2786679
Email: t.j.j.vandenboom/b.deschutter@its.tudelft.nl.

Keywords: predictive control, control and optimization, control of discrete event systems, max-plus-linear systems, manufacturing systems.

Abstract

Model predictive control (MPC) is a popular controller design technique in the process industry. Conventional MPC uses (non)linear 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,+) algebra. Up to now we have only considered the deterministic noise-free case without modeling errors. In this paper we extend our previous results to cases with noise and/or modeling errors. We show that under quite general conditions the resulting optimization problem can be solved very efficiently.

1 Introduction

Model predictive control (MPC) [2, 3, 5, 9] is currently one of the most widely used *advanced* control design methods in the process industry. MPC provides many attractive features: it is applicable to MIMO systems, it can handle constrains on inputs and outputs in a systematic way, it is capable of tracking pre-scheduled reference signals, and it is an easy-to-tune method. Usually MPC uses linear or nonlinear discrete-time models. However, the attractive features mentioned above have led us to extend MPC to a class of discrete event systems: the max-plus-linear (MPL) systems. Loosely speaking we could say that this class corresponds to the class of discrete event systems in which there is synchronization but no concurrency. 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, 6] (see also Section 2).
In [7, 8] we have extended MPC to MPL systems, and in [14] we have presented some results in connection with the closed-loop behavior and tuning rules for MPL-MPC. However, in those papers we have only considered the deterministic noise-free case without modeling errors. In this paper we will extend our previous results to cases with noise and/or modeling errors.
In contrast to conventional linear systems, where noise and disturbances are usually modeled by including an extra term in the system equations, the influence of noise and disturbances in MPL discrete event systems is not max-plus-additive, but max-plus-multiplicative. This means that the system matrices are perturbed and the system properties change. Ignoring the noise can lead to bad tracking behavior or even an unstable closed loop. A second important feature is modeling error. Uncertainty in the modeling or identification phase leads to errors in the system matrices. It is clear that modeling errors, and noise and disturbances both perturb the system by introducing uncertainty in the system matrices. Sometimes it is difficult to distinguish the two from one another, and usually fast changes in the system matrices are considered as noise and disturbances, whereas slow changes or permanent errors are considered as model mismatch. In this paper both features are treated in one single framework and the characterization of the perturbation determines whether it describes model mismatch or disturbance. We also show that under quite general restrictions the resulting MPC optimization problem can be solved very efficiently.
There are few results in the literature on noise and modeling errors in an MPL context. However, for other classes of discrete event systems uncertainty results can be found in [4, 10, 13, 15] and the references therein.

2 Max-plus-linear systems and MPC

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, 6]:

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

for $x, y \in \mathbb{R}_\varepsilon$ 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 $A, B \in \mathbb{R}_\varepsilon^{m\times n}$ and $C \in \mathbb{R}_\varepsilon^{n\times p}$. The matrix $\mathcal{E}$ is the max-plus-algebraic zero matrix: $[\mathcal{E}]_{ij} = \varepsilon$ for all i, j.
In [1, 6] it has been shown that (time-invariant) discrete event systems in which there is synchronization but no concurrency can be described by a model of the form

$$x(k) = A \otimes x(k-1) \oplus B \otimes u(k) \qquad (1)$$
$$y(k) = C \otimes x(k) \ . \qquad (2)$$

Systems that can be described by this model will be called time-invariant max-plus-linear (MPL) systems. The index k is called the event counter. For a discrete event system $x(k)$ would typically contain the time instants at which the internal events occur for the kth time, $u(k)$ would contain the time instants at which the input events occur for the kth time, and $y(k)$ would contain the time instants at which the output events occur for the kth time.
In [7, 8] we have extended the MPC framework to time-invariant MPL models (1)–(2) as follows. Just as in conventional MPC we define a cost criterion J that reflects the reference tracking error ($J_{\rm out}$) and the control effort ($J_{\rm in}$) in the event period $[k, k+N_{\rm p}-1]$:

$$J(k) = J_{\rm out}(k) + \lambda J_{\rm in}(k)$$

where $N_{\rm p}$ is the prediction horizon and λ a weighting parameter. Possible choices for $J_{\rm out}$ and $J_{\rm in}$ are given in [7, 8] (see also Section 5). The aim is to compute an optimal input sequence $u(k), \ldots, u(k+N_{\rm p}-1)$ that minimizes $J(k)$ subject to linear constraints on the inputs and outputs. Since the $u(k)$'s correspond to consecutive event occurrence times, we have the condition $\Delta u(k+j) \geq 0$ for $j = 0, \ldots, N_{\rm p}-1$. Furthermore, in order to reduce the number of decision variables and the corresponding computational complexity we introduce a control horizon $N_{\rm c}$ ($\leq N_{\rm p}$) and we impose the additional condition that the input rate should be constant from $k + N_{\rm c} - 1$ on: $\Delta u(k+j) = \Delta u(k+N_{\rm c}-1)$ for $j = N_{\rm c}, \ldots, N_{\rm p}-1$. MPC uses a receding horizon principle. After computation of the optimal control sequence $u(k), \ldots, u(k+N_{\rm c}-1)$, 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.
Let $\mathcal{S}_{\rm mps}$ be the set of functions f defined as:

$$f(x) = \max_i(\alpha_{i,1}x_1 + \ldots + \alpha_{i,n}x_n + \beta_i)$$

with $x \in \mathbb{R}_\varepsilon^n$ and $\alpha_{i,j} \in \mathbb{R}^+$ and $\beta_i \in \mathbb{R}$. We write $f \in \mathcal{S}_{\rm mps}(x)$, if we want to stress that f is a function of x.

Lemma 1 *The set $\mathcal{S}_{\rm mps}$ is closed under the operations $\oplus$, $\otimes$, and scalar multiplication by a nonnegative scalar.*

Proof: For $x, y, z, v \in \mathbb{R}_\varepsilon$ and $\rho \in \mathbb{R}^+$ we have $\max(\max(x,y), \max(z,v)) = \max(x,y,z,v)$, $\max(x,y) + \max(z,v) = \max(x+z, x+v, y+z, y+v)$ and $\rho \max(x,y) = \max(\rho x, \rho y)$. ∎

Lemma 2 *If $f \in \mathcal{S}_{\rm mps}$ then f is a nondecreasing function of its arguments.*

Proof: If $\tilde{x}, \hat{x} \in \mathbb{R}_\varepsilon^n$ and $\tilde{x} \leq \hat{x}$ then we have $\sum_j \alpha_{ij}\tilde{x}_j + \beta_i \leq \sum_j \alpha_{ij}\hat{x}_j + \beta_i$ since $\alpha_{i,j} \geq 0$ for all i, j. As a consequence, we have $f(\tilde{x}) \leq f(\hat{x})$. ∎

3 Noise and uncertainty model

In this section we extend the noise-free deterministic model (1)–(2) to include uncertainty. So we now consider the following max-plus-linear system:

$$x(k) = A(k) \otimes x(k-1) \oplus B(k) \otimes u(k) \qquad (3)$$
$$y(k) = C(k) \otimes x(k) \qquad (4)$$

where $A(k)$, $B(k)$ and $C(k)$ represent uncertain system matrices due to modeling errors or disturbances. Usually fast changes in the system matrices will be considered as noise and disturbances, whereas slow changes or permanent errors are considered as model mismatch. In this paper both features will be treated in one single framework. The uncertainty caused by disturbances and errors in the estimation of physical variables, is gathered in the uncertainty vector $e(k)$. We assume that the uncertainty is bounded. Furthermore, $e(k)$ and $e(k-1)$ may be related, e.g., by assuming the change $\Delta e(k) = e(k) - e(k-1)$ to be bounded.
We assume that the uncertainty vector $e(k)$ captures the complete time-varying aspect of the system. The system matrices of an MPL model usually consist of sums or maximizations of internal process times, transportation times, etc. (see also Section 6). Since the entries of $e(k)$ directly correspond to the uncertainties in these duration times, it follows from Lemma 1 that the entries of the system matrices belong to $\mathcal{S}_{\rm mps}$:

$$A(k) \in \mathcal{S}_{\rm mps}^{n\times n}(e(k)) \qquad B(k) \in \mathcal{S}_{\rm mps}^{n\times m}(e(k)) \qquad C(k) \in \mathcal{S}_{\rm mps}^{l\times n}(e(k)) \qquad (5)$$

4 Prediction model

Define the vectors

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

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

We assume that $\tilde{e}(k)$ is in a bounded polyhedral set $\mathcal{E}_{\tilde{e}}$. Note that for ease of notation we will sometimes drop the index k from $\tilde{u}(k)$, $\tilde{y}(k)$ and $\tilde{e}(k)$.
The prediction model for (3)–(4) is given by:

$$\tilde{y}(k) = \tilde{C}(\tilde{e}(k)) \otimes x(k-1) \oplus \tilde{D}(\tilde{e}(k)) \otimes \tilde{u}(k) \qquad (6)$$

in which $\tilde{C}(\tilde{e}(k))$ and $\tilde{D}(\tilde{e}(k))$ are given by

$$\tilde{C}(\tilde{e}(k)) = \begin{bmatrix} \tilde{C}_1(e(k)) \\ \vdots \\ \tilde{C}_{N_\mathrm{p}}(e(k)) \end{bmatrix}$$

$$\tilde{D}(\tilde{e}(k)) = \begin{bmatrix} \tilde{D}_{11}(e(k)) & \cdots & \tilde{D}_{1N_\mathrm{p}}(e(k)) \\ \vdots & \ddots & \vdots \\ \tilde{D}_{N_\mathrm{p}1}(e(k)) & \cdots & \tilde{D}_{N_\mathrm{p}N_\mathrm{p}}(e(k)) \end{bmatrix}$$

where

$$\tilde{C}_m(\tilde{e}(k)) = C(k+m-1) \otimes A(k+m-1) \otimes \ldots \otimes A(k)$$

and

$$\tilde{D}_{mn}(\tilde{e}) = \begin{cases} C(k+m-1) \otimes A(k+m-1) \otimes \ldots \\ \quad \ldots \otimes A(k+n) \otimes B(k+n-1) & \text{if } m > n \\ C(k+m-1) \otimes B(k+m-1) & \text{if } m = n \\ \mathcal{E} & \text{if } m < n \end{cases}$$

Lemma 3 *The entries of $\tilde{C}(e(k))$ and $\tilde{D}(e(k))$ belong to $\mathcal{S}_\mathrm{mps}(e(k))$. For a given $x(k-1)$ and $\tilde{u}(k)$ the entries of $\tilde{y}(k)$ belong to $\mathcal{S}_\mathrm{mps}(e(k))$.*

Proof: This is a direct consequence of the definition of $\tilde{C}(e(k))$, $\tilde{D}(e(k))$ and (6) in combination with (5) and Lemma 1. ∎

5 Worst-case criterion MPC

In MPL-MPC we want to minimize the criterion

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

where J_out represents the tracking error and J_in is related to the input dates. We aim to find the optimal $(\tilde{u}, \tilde{y})$ that minimizes $J(\tilde{y}, \tilde{u})$, where $\tilde{y}$ and $\tilde{u}$ are related by (6). Note that the relation between $\tilde{u}$ and $\tilde{y}$ is not unique because of (bounded) perturbation $\tilde{e}(k)$. Instead of considering general linear constraints on the inputs and outputs as in [7, 8] we will only consider linear constraints on the input in this paper $A_\mathrm{c}(k)\tilde{u}(k) \le b_\mathrm{c}(k)$. A typical example of such a constraint is an upper and lower bound for the input rate:

$$d_\mathrm{min}(k+j) \le \Delta u(k+j) \le d_\mathrm{max}(k+j) .$$

The *worst-case MPC problem* at event step k is now defined as follows:

$$J_\mathrm{wc}(k) = \min_{\tilde{u}(k)} \max_{\tilde{e}(k) \in \mathcal{E}_{\tilde{e}}} J(\tilde{y}(k), \tilde{u}(k)) \qquad (7)$$

subject to

$$\tilde{y}(k) = \tilde{C}(\tilde{e}(k)) \otimes x(k-1) \oplus \tilde{D}(\tilde{e}(k)) \otimes \tilde{u}(k) \qquad (8)$$
$$\Delta u(k+j) \ge 0 \quad , \quad j = 0, \ldots, N_\mathrm{p}-1 \qquad (9)$$
$$\Delta^2 u(k+j) = 0 \ , \ j \ge N_\mathrm{c} \qquad (10)$$
$$A_\mathrm{c}(k)\tilde{u}(k) \le b_\mathrm{c}(k) \ . \qquad (11)$$

Now we eliminate (8) by substituting this equation in $J_\mathrm{wc}(k)$ and by maximizing the result over all $\tilde{e}(k)$. For a given $\tilde{u}(k)$ the worst-case $\tilde{e}(k)$ will be denoted by $\tilde{e}^{\#}(\tilde{u}(k))$, or by $\tilde{e}^{\#}(k)$ or $\tilde{e}^{\#}$ for short if no confusion is possible. So for any $\tilde{u}$, we let

$$\tilde{e}^{\#}(k) = \arg \max_{\tilde{e}(k) \in \mathcal{E}_{\tilde{e}}} J_\mathrm{out}(\tilde{y}(\tilde{e}(k), \tilde{u}))$$

$$J^{\#}_\mathrm{out}(\tilde{u}) = J_\mathrm{out}(\tilde{y}(\tilde{e}^{\#}(k), \tilde{u})) \ .$$

The corresponding worst-case output is then given by

$$\tilde{y}(\tilde{e}^{\#}, \tilde{u}) = \tilde{C}(\tilde{e}^{\#}) \otimes x(k-1) \oplus \tilde{D}(\tilde{e}^{\#}) \otimes \tilde{u} \ .$$

The *outer* worst-case MPC problem is defined as:

$$\min_{\tilde{u}(k)} J^{\#}_\mathrm{out}(\tilde{u}) + \lambda J_\mathrm{in}(\tilde{u})$$

subject to

$$\Delta u(k+j) \ge 0 \quad , \quad j = 0, \ldots, N_\mathrm{p}-1 \qquad (12)$$
$$\Delta^2 u(k+j) = 0 \ , \ j \ge N_\mathrm{c} \qquad (13)$$
$$A_\mathrm{c}(k)\tilde{u}(k) \le b_\mathrm{c}(k) \ . \qquad (14)$$

Now we make the following assumptions:

Assumption A1: J_out is a nondecreasing, convex function of $\tilde{y}$

Assumption A2: J_in is convex in $\tilde{u}$.

These assumptions hold for several objective functions that are frequently encountered in a discrete event context and are thus not overly restrictive.
If the due dates r for the outputs of the system are known and if we have to pay a penalty for every delay, a possible output cost criterion is the tardiness:

$$J_\mathrm{out,1}(\tilde{y}(k)) = \sum_i \max(\tilde{y}_i(k) - \tilde{r}_i(k), 0)$$

with $\tilde{r}$ defined similarly to $\tilde{y}$. Clearly, $J_\mathrm{out,1}$ satisfies Assumption A1. The maximal output delay also satisfies Assumption A1:

$$J_\mathrm{out,2}(\tilde{y}(k)) = \max_i \big(\max(\tilde{y}_i(k) - \tilde{r}_i(k), 0)\big) \ .$$

For the input cost criterion we could take [7, 8]:

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

$$J_{\mathrm{in},1}(\tilde{u}(k)) = \sum_i \tilde{u}_i(k)$$

which minimize the input time instants or

$$J_{\mathrm{in},2}(\tilde{u}(k)) = -\sum_i \tilde{u}_i(k)$$

which maximizes the input time instants[1]. Clearly, $J_{\mathrm{in},0}$, $J_{\mathrm{in},1}$ and $J_{\mathrm{in},2}$ all satisfy Assumption A2.

Proposition 4 *If Assumptions A1 and A2 hold, then the outer worst-case MPC problem is convex in $\tilde{u}$.*

Proof: J_{in} is convex in $\tilde{u}$ by Assumption A2. Furthermore, the constraints (12)–(14) only depend on $\tilde{u}$ and are convex in $\tilde{u}$. So we only have to prove that $J^{\#}_{\mathrm{out}}$ is convex in $\tilde{u}$. Define for $0 \le \rho \le 1$:

$$\tilde{u}_3(k) = \rho\tilde{u}_1(k) + (1-\rho)\tilde{u}_2(k)$$

$$\tilde{e}^{\#}_j(k) = \arg \max_{\tilde{e}(k)\in\mathcal{E}_{\tilde{e}}} J_{\mathrm{out}}(\tilde{y}(\tilde{e}(k),\tilde{u}_j)) \ \ , \ j=1,2,3$$

Define $\tilde{M}_{CD}(\tilde{e}) = \left[\ \tilde{C}(\tilde{e}) \quad \tilde{D}(\tilde{e})\ \right]$. Now we have[2]:

$$\begin{aligned}
&[\tilde{y}_3(\tilde{e}^{\#}_3,\tilde{u}_3)]_i = [\tilde{M}_{CD}(\tilde{e}^{\#}_3)]_i \otimes \begin{bmatrix} x(k-1) \\ \tilde{u}_3(k) \end{bmatrix} \\
&= \max_\ell \left([\tilde{M}_{CD}(\tilde{e}^{\#}_3)]_{i,\ell} + \begin{bmatrix} x(k-1) \\ \tilde{u}_3(k) \end{bmatrix}_\ell \right) \\
&= \max_\ell \left([\tilde{M}_{CD}(\tilde{e}^{\#}_3)]_{i,\ell} + \begin{bmatrix} x(k-1) \\ \rho\tilde{u}_1(k)+(1-\rho)\tilde{u}_2(k) \end{bmatrix}_\ell \right) \\
&= \max_\ell \left(\rho[\tilde{M}_{CD}(\tilde{e}^{\#}_3)]_{i,\ell} + (1-\rho)[\tilde{M}_{CD}(\tilde{e}^{\#}_3)]_{i,\ell} \right. \\
&\qquad \left. + \begin{bmatrix} \rho x(k-1) + (1-\rho)x(k-1) \\ \rho\tilde{u}_1(k) + (1-\rho)\tilde{u}_2(k) \end{bmatrix}_\ell \right) \\
&\le \max_\ell \left(\rho[\tilde{M}_{CD}(\tilde{e}^{\#}_3)]_{i,\ell} + \begin{bmatrix} \rho x(k-1) \\ \rho\tilde{u}_1(k) \end{bmatrix}_\ell \right) \\
&\quad + \max_\ell \left((1-\rho)[\tilde{M}_{CD}(\tilde{e}^{\#}_3)]_{i,\ell} + \begin{bmatrix} (1-\rho)x(k-1) \\ (1-\rho)\tilde{u}_2(k) \end{bmatrix}_\ell \right) \\
&\qquad (\text{since } \max_i(v_i,w_i) \le \max_i(v_i) + \max_i(w_i)) \\
&\le \rho \max_\ell \left([\tilde{M}_{CD}(\tilde{e}^{\#}_3)]_{i,\ell} + \begin{bmatrix} x(k-1) \\ \tilde{u}_1(k) \end{bmatrix}_\ell \right) \\
&\quad +(1-\rho) \max_\ell \left([\tilde{M}_{CD}(\tilde{e}^{\#}_3)]_{i,\ell} + \begin{bmatrix} x(k-1) \\ \tilde{u}_2(k) \end{bmatrix}_\ell \right) \\
&\le \rho[\tilde{y}(\tilde{e}^{\#}_3,\tilde{u}_1)]_i + (1-\rho)[\tilde{y}(\tilde{e}^{\#}_3,\tilde{u}_2)]_i
\end{aligned}$$

[1] For a manufacturing system, this would correspond to a production scheme in which raw material is fed to the system as late as possible.

[2] We use $[M]_i$ ($[m]_i$) to denote the ith row (component) of a matrix M (column vector m).

and thus

$$J_{\mathrm{out}}(\tilde{y}_3) \le J_{\mathrm{out}}\Big(\rho\tilde{y}(\tilde{e}^{\#}_3,\tilde{u}_1) + (1-\rho)\tilde{y}(\tilde{e}^{\#}_3,\tilde{u}_2)\Big) \quad (15)$$

because J_{out} is a nondecreasing function of $\tilde{y}$ by Assumption A1. This implies that

$$\begin{aligned}
J^{\#}_{\mathrm{out}}&\Big(\rho\tilde{u}_1 + (1-\rho)\tilde{u}_2\Big) = J^{\#}_{\mathrm{out}}(\tilde{u}_3) \\
=\ & J_{\mathrm{out}}(\tilde{y}_3(\tilde{e}^{\#}_3,\tilde{u}_3)) \\
\le\ & J_{\mathrm{out}}\big(\rho\tilde{y}(\tilde{e}^{\#}_3,\tilde{u}_1) + (1-\rho)\tilde{y}(\tilde{e}^{\#}_3,\tilde{u}_2)\big) \qquad (\text{by } 15) \\
\le\ & \rho J_{\mathrm{out}}(\tilde{y}(\tilde{e}^{\#}_3,\tilde{u}_1)) + (1-\rho)J_{\mathrm{out}}(\tilde{y}(\tilde{e}^{\#}_3,\tilde{u}_2)) \\
& \qquad (\text{since } J_{\mathrm{out}} \text{ is convex in } \tilde{y} \text{ by Assumption A1}) \\
\le\ & \rho J_{\mathrm{out}}(\tilde{y}(\tilde{e}^{\#}_1,\tilde{u}_1)) + (1-\rho)J_{\mathrm{out}}(\tilde{y}(\tilde{e}^{\#}_2,\tilde{u}_2)) \\
\le\ & \rho J^{\#}_{\mathrm{out}}(\tilde{u}_1) + (1-\rho)J^{\#}_{\mathrm{out}}(\tilde{u}_2) \ .
\end{aligned}$$

Hence, $J^{\#}_{\mathrm{out}}$ is a convex function of $\tilde{u}$, and as a consequence, the outer worst-case MPC problem is a convex problem. ■

Let us now consider the *inner* worst-case MPC problem:

$$\max_{\tilde{e}(k)\in\mathcal{E}_{\tilde{e}}} J_{\mathrm{out}}(\tilde{y}(\tilde{e},\tilde{u})) \quad (16)$$

$$\text{s.t. } \tilde{y}(\tilde{e},\tilde{u}) = \tilde{C}(\tilde{e}) \otimes x(k-1) \oplus \tilde{D}(\tilde{e}) \otimes \tilde{u} \ . \quad (17)$$

We will show how this problem can be solved efficiently. Recall that $\mathcal{E}_{\tilde{e}}$ is a bounded polyhedral set. Note that the vertices of $\mathcal{E}_{\tilde{e}}$ form a lattice w.r.t. the partial order relation $\le$. Let $\mathcal{E}^{\mathrm{v}}_{\tilde{e},\max}$ be the top points of this lattice, i.e., $\mathcal{E}^{\mathrm{v}}_{\tilde{e},\max}$ is the set of the vertex points $\tilde{e}^{\mathrm{v}}_{\max}$ of $\mathcal{E}_{\tilde{e}}$ for which we have

$$\nexists \tilde{e} \in \mathcal{E}_{\tilde{e}} \text{ with } \tilde{e} \ne \tilde{e}^{\mathrm{v}}_{\max} \text{ and } \tilde{e}^{\mathrm{v}}_{\max} \le \tilde{e} \ .$$

Now consider the *reduced inner* worst-case MPC problem:

$$\max_{\tilde{e}(k)\in\mathcal{E}^{\mathrm{v}}_{\tilde{e},\max}} J_{\mathrm{out}}\big(\tilde{C}(\tilde{e}) \otimes x(k-1) \oplus \tilde{D}(\tilde{e}) \otimes \tilde{u}\big) \ . \quad (18)$$

Lemma 5 *If Assumption A1 holds, then for a given $x(k-1)$ and $\tilde{u}(k)$ the function J_{out} is convex in $\tilde{e}(k)$.*

Proof: The function h is defined by $h(x) = f(g(x))$ and if g is convex and f is convex and nondecreasing then h is convex. Functions that belong to $\mathcal{S}_{\mathrm{mps}}$ are convex. Since for a given $\tilde{u}$ we have $\tilde{y}(\tilde{e},\tilde{u}) \in \mathcal{S}_{\mathrm{mps}}$ by Lemma 3, $\tilde{y}$ is convex as a function $\tilde{e}$. Furthermore, J_{out} is convex and nondecreasing as a function of $\tilde{y}$ by Assumption A1. Hence, J_{out} is convex in $\tilde{e}$. ■

Proposition 6 *If Assumption A1 holds, then an optimal solution of the reduced inner worst-case MPC problem (18) is also a solution of the (full) inner worst-case MPC problem (16)–(17).*

Proof: First we prove that the maximum of the (full) inner worst-case MPC problem (16)–(17) will be reached in a "maximal" point of $\mathcal{E}_{\tilde{e}}$, i.e., a point $\tilde{e}_{\max}$ (not necessarily a vertex point!) of $\mathcal{E}_{\tilde{e}}$ for which

$$\nexists \tilde{e} \in \mathcal{E}_{\tilde{e}} \text{ with } \tilde{e} \neq \tilde{e}_{\max} \text{ and } \tilde{e}_{\max} \leq \tilde{e} \ .$$

Indeed, from Lemmas 2 and 3 it follows that if $\tilde{e}_1 \leq \tilde{e}_2$ then $\tilde{y}(\tilde{e}_1) \leq \tilde{y}(\tilde{e}_2)$ and thus also $J_{\text{out}}(\tilde{y}(\tilde{e}_1, \tilde{u}) \leq J_{\text{out}}(\tilde{y}(\tilde{e}_2, \tilde{u}))$ because of Assumption A1. Hence, the maximum of the (full) inner worst-case MPC problem will be reached in a "maximal" point of $\mathcal{E}_{\tilde{e}}$.
Now we show that the maximum will be reached in a "maximal" vertex point. Suppose that the maximum would be reached in a point $\tilde{e}_{\max}$ that is not a vertex point. In that case, $\tilde{e}_{\max}$ can be written as the convex combination of the vertex points $\tilde{e}^{\mathrm{v},j}_{\max}$ of the face of $\mathcal{E}_{\tilde{e}}$ to which $\tilde{e}_{\max}$ belongs. Since for a given $x(k-1)$ and $\tilde{u}(k)$ J_{out} is convex in $\tilde{e}$ by Lemma 5 and thus also quasi-convex, we have $J_{\text{out}}(\tilde{y}(\tilde{e}_{\max}, \tilde{u})) \leq \max_j J_{\text{out}}(\tilde{y}(\tilde{e}^{\mathrm{v},j}_{\max}, \tilde{u}))$. Hence, an optimal solution of the reduced inner problem is also an optimal solution of the full inner problem. ∎

The set $\mathcal{E}^{\mathrm{v}}_{\tilde{e},\max}$ is independent of $\tilde{u}$ and can thus be pre-computed off-line. Methods to compute all vertex points of a polyhedral set can be found in [11, 12]. The computation can be made more efficient by already discarding the vertex points that cannot result in vertex points that will belong to $\mathcal{E}^{\mathrm{v}}_{\tilde{e},\max}$ during the computation. In combination with Proposition 6 this allows for an efficient solution of the inner worst-case MPC problem. Since the outer worst-case MPC problem is convex by Proposition 4 this implies that the overall worst-case MPC problem can be solved efficiently.

6 Example: Simple production system

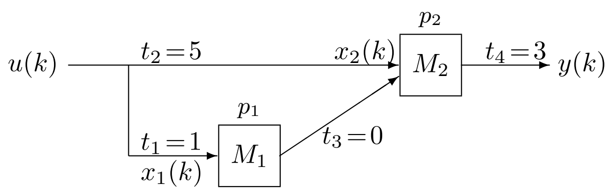

Figure 1: A production system.

Consider the production system of Figure 1. This system consists of two machines M_1 and M_2. When a batch of raw material is fed to the system, one part of the batch goes directly from the input of the system to the input of machine M_1 (with a certain transportation delay), whereas the other part of the batch first goes to machine M_2 for pre-processing. Afterwards, assembly takes places on machineM_1. We assume that each machine starts working as soon as possible on each batch, i.e., as soon as the raw material or the required intermediate products are available, and as soon as the machine is idle (i.e., the previous batch has been finished and has left the machine). Define:

$u(k)$	:	time instant at which the system is fed for the kth time
$y(k)$	:	time instant at which the kth product leaves the system
$x_i(k)$	:	time instant at which machine i starts for the kth time
t_j	:	transportation time
$p_i(k)$	:	processing time on machine i for the kth batch.

Both processing times $p_1(k)$ and $p_2(k)$ are assumed to be estimated with some modeling error, and are corrupted by noise. Suppose $p_1(k) \in [1, 5]$ and $p_2(k) \in [3, 6]$ and $p_1(k) + p_2(k) \leq 9$. Note that this implies that if $p_1(k) < 4$ then the direct path from the input to M_2 is the longest, whereas if $p_1(k) > 4$ the path from the input via M_1 to M_1 is the longest.
From the system equations

$$\begin{aligned}
x_1(k) &= \max\big(p_1(k{-}1){+}x_1(k{-}1), u(k){+}1\big) \\
x_2(k) &= \max\big(p_2(k{-}1){+}x_2(k{-}1), p_1(k){+}x_1(k), u(k){+}6\big) \\
y(k) &= x_2(k){+}p_2(k){+}3
\end{aligned}$$

we derive:

$$\begin{aligned}
x(k) &= \begin{bmatrix} p_1(k-1) & \varepsilon \\ p_1(k-1) + p_1(k) & p_2(k-1) \end{bmatrix} \otimes x(k-1) \\
&\quad \oplus \begin{bmatrix} 1 \\ \max(6, p_1(k) + 1) \end{bmatrix} \otimes u(k) \\
y(k) &= \begin{bmatrix} \varepsilon & p_2(k) + 3 \end{bmatrix} \otimes x(k) \ .
\end{aligned}$$

If we define

$$e(k) = \begin{bmatrix} e_1(k) \\ e_2(k) \\ e_3(k) \\ e_4(k) \end{bmatrix} = \begin{bmatrix} p_1(k-1) \\ p_2(k-1) \\ p_1(k) \\ p_2(k) \end{bmatrix}$$

then we obtain

$$A(k) = \begin{bmatrix} e_1(k) & \varepsilon \\ e_1(k) + e_3(k) & e_2(k) \end{bmatrix}$$

$$B(k) = \begin{bmatrix} 1 \\ \max(6, e_3(k){+}1) \end{bmatrix}, \quad C(k) = \begin{bmatrix} \varepsilon & e_4(k){+}3 \end{bmatrix}$$

For this perturbed MPL system we solve the worst-case MPC problem. For each k the critical pairs are $(p_1(k{+}j), p_2(k{+}j)) = (3, 6)$ and $(p_1(k{+}j), p_2(k{+}j)) = (5, 4)$. The set $\mathcal{E}^{\mathrm{v}}_{\tilde{e},\max}$ consists of $2^{N_{\mathrm{p}}+1} = 32$ top points, corresponding to all 2^5 combinations of the two critical points $(p_1(k+j), p_2(k+j))$ for $j = 1, \ldots, N_{\mathrm{p}} + 1$.
The reference signal is given by $r(k) = 10 + 5 \cdot k$, the initial state is equal to $x(0) = [\, 0 \ \ 10 \,]^T$, and

$$\begin{aligned}
J(\tilde{y}(k), \tilde{u}(k)) &= J_{\text{out},1}(\tilde{y}(k)) + \lambda J_{\text{in},2}(\tilde{u}(k)) \\
&= \sum_{i=1}^{N_{\mathrm{p}}} \max(\tilde{y}_i(k){-}\tilde{r}_i(k), 0){-}\lambda \sum_{i=1}^{N_{\mathrm{p}}} \tilde{u}_i(k)
\end{aligned}$$

for $N_p = 4$, $N_c = 2$ and $\lambda = 0.01$. With the above choice of the cost criterion, we can rewrite the worst-case MPC problem into a linear programming problem. The optimal input sequence is computed for $k = 1, \ldots, 100$, and for each k, the first element $u(k)$ of the sequence $\tilde{u}(k)$ is applied to the perturbed system (due to the receding horizon strategy). In the experiment, the true system is simulated for a random sequence $\big(p_{1,t}(k), p_{2,t}(k)\big)$, $k = 1, \ldots, 100$ in the allowed region.

Figure 2 gives the tracking error between the reference signal and the output signal $y(k)$. It can be observed that $r(k)-y(k)$ becomes larger than zero. This is caused by the worst-case approach. The worst-case MPC cost criterion is based on the lower bound of the predicted tracking error, which should be larger than zero. The true value of tracking error will obviously be larger than this lower bound.

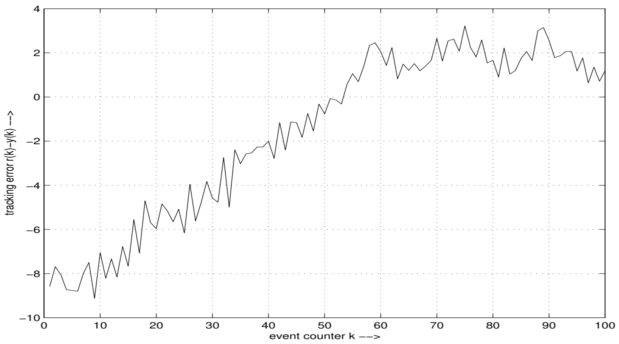

Figure 2: Tracking error $r(k)-y(k)$

7 Conclusions

We have further extended the MPC framework to include max-plus-linear discrete event systems with modeling errors, noise and/or disturbances. We have presented a unified framework to deal with bounded uncertainties for max-plus-linear discrete event systems. This allows the design of a worst-case MPC controller for such systems. We have show how the resulting optimization problem can be computed efficiently using a two-level optimization approach.

Acknowledgments

This research was partially sponsored by the European Community TMR project ALAPEDES, and by the FWO Research Community ICCoS.

References

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

[2] L. Biegler, "Efficient solution of dynamic optimization and NMPC problems," in *Nonlinear Model Predictive Control* (F. Allgöwer and A. Zheng, eds.), vol. 26 of *Progress in Systems and Control Theory*, Basel, Switzerland: Birkhäuser Verlag, 2000.

[3] E.F. Camacho and C. Bordons, *Model Predictive Control in the Process Industry*. Berlin, Germany: Springer-Verlag, 1995.

[4] J. Cardoso, R. Valette, and D. Dubois, "Possibilistic Petri nets," *IEEE Transactions on Systems, Man and Cybernetics, Part B: Cybernetics*, vol. 29, no. 5, pp. 573–582, 1999.

[5] 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.

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

[7] B. De Schutter and T. van den Boom, "Model predictive control for max-plus-linear discrete event systems," *Automatica*, vol. 37, no. 7, July 2001.

[8] B. De Schutter and T. van den Boom, "Model predictive control for max-plus-linear systems," in *Proceedings of the 2000 American Control Conference*, Chicago, Illinois, pp. 4046–4050, June 2000.

[9] 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.

[10] F. Lin, "Robust and adaptive supervisory control of discrete event systems," *IEEE Transactions on Automatic Control*, vol. 38, no. 12, pp. 1848–1852, 1993.

[11] T.H. Mattheiss and D.S. Rubin, "A survey and comparison of methods for finding all vertices of convex polyhedral sets," *Mathematics of Operations Research*, vol. 5, no. 2, pp. 167–185, May 1980.

[12] T.S. Motzkin, H. Raiffa, G.L. Thompson, and R.M. Thrall, "The double description method," in *Contributions to the theory of games* (H.W. Kuhn and A.W. Tucker, eds.), no. 28 in Annals of Mathematics Studies, pp. 51–73, Princeton, New Jersey: Princeton University Press, 1953.

[13] S.J. Park and J.T. Lim, "Fault-tolerant robust supervisor for discrete event systems with model uncertainty and its application to a workcell," *IEEE Transactions on Robotics and Automation*, vol. 15, no. 2, pp. 386–391, Apr. 1999.

[14] T. van den Boom and B. De Schutter, "MPC for max-plus-linear systems: Closed-loop behavior and tuning," in *Proceedings of the American Control Conference 2001*, Arlington, USA, June 25-27, 2001.

[15] S. Young and V.K. Garg, "Model uncertainty in discrete event systems," *SIAM Journal on Control and Optimization*, vol. 33, no. 1, pp. 208–226, Jan. 1995.