

Technical report bds:00-18

Model predictive control 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, "Model predictive control for perturbed max-plus-linear systems," *Systems & Control Letters*, vol. 45, no. 1, pp. 21–33, Jan. 2002.

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>

*This report can also be downloaded via http://pub.deschutter.info/abs/00_18.html

Model Predictive Control for Perturbed Max-Plus-Linear Systems

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

Control Lab, Faculty of Information Technology and Systems
Delft University of Technology, P.O.Box 5031, 2600 GA Delft, The Netherlands
email: {t.j.j.vandenboom,b.deschutter}@its.tudelft.nl

Abstract

Model predictive control (MPC) is a popular controller design technique in the process industry. Conventional MPC uses linear or nonlinear discrete-time models. Recently, we have extended MPC to a class of discrete event systems that can be described by a model that is “linear” in the $(\max,+)$ algebra. In our previous work we have only considered MPC for the deterministic noise-free case without modeling errors. In this paper we extend our previous results on MPC for max-plus-linear systems to cases with noise and/or modeling errors. We show that under quite general conditions the resulting optimization problems can be solved very efficiently.

Keywords: discrete event systems, model predictive control, max-plus-linear systems, noise and modeling errors.

1 Introduction

Model predictive control (MPC) [1, 3, 5, 10] 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 multi-input multi-output systems, it can handle constraints 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 [2, 6]. Loosely speaking, 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 [2, 6] (see also Section 2). Max-plus-linear discrete event systems usually arise in the context of manufacturing systems, telecommunication networks, railway networks, and parallel computing.

In [8, 9] we have extended MPC to MPL systems, and in [17] we have presented some results in connection with the closed-loop behavior (including stability) and tuning rules for MPL-MPC. However, in those papers we have only considered the deterministic noise-free

*Corresponding author.

case without modeling errors. In this paper we will extend these 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 (i.e., the noise is considered to be additive), 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 will be perturbed and as a consequence the system properties will change. Ignoring the noise can lead to a bad tracking behavior or even to an unstable closed loop. A second important feature is modeling errors. Uncertainty in the modeling or identification phase leads to errors in the system matrices. It is clear that both modeling errors, and noise and disturbances perturb the system by introducing uncertainty in the system matrices. Sometimes it is difficult to distinguish the two from one another, but 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 and the characterization of the perturbation will determine whether it describes model mismatch or disturbance. To the authors' best knowledge this is the first time that such an approach is used in the MPL context. We will also show that under quite general restrictions the resulting MPC optimization problem can be solved very efficiently.

Note that 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, 11, 15, 19] and the references therein.

This paper is organized as follows. In Section 2 we give a concise introduction to MPL systems and MPC for MPL systems (without noise or modeling errors). Next, we present a noise and uncertainty model for MPL systems. In Section 4 we describe the prediction model and in Section 5 we show how the worst-case MPC controller can be designed. We conclude with a worked example.

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 [2, 6]:

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

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

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

for matrices $A, B \in \mathbb{R}_\varepsilon^{m \times n}$ 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 [2, 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

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

$$y(k) = C \otimes x(k) . \tag{2}$$

Systems that can be described by this model are called time-invariant max-plus-linear (MPL) systems. The index k is called the event counter. For discrete event systems the state $x(k)$ typically contains the time instants at which the internal events occur for the k th time, the input $u(k)$ contains the time instants at which the input events occur for the k th time, and the output $y(k)$ contains the time instants at which the output events occur for the k th time¹.

In [8, 9] we have extended the MPC framework to time-invariant MPL models (1)–(2) as follows. Just as in conventional MPC [5, 10] we define a cost criterion J that reflects the reference tracking error (J_{out}) and the control effort (J_{in}) in the event period $[k, k + N_p - 1]$:

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

where N_p is the prediction horizon and λ is a weighting parameter. Possible choices for J_{out} and J_{in} are given in [8, 9] (see also Section 5). The aim is now to compute an optimal input sequence $u(k), \dots, u(k+N_p-1)$ 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 additional condition $\Delta u(k+j) = u(k+j) - u(k+j-1) \geq 0$ for $j = 0, \dots, N_p - 1$. Furthermore, in order to reduce the number of decision variables and the corresponding computational complexity we introduce a control horizon $N_c (\leq N_p)$ and we impose the additional condition that the input rate should be constant from the point $k+N_c-1$ on: $\Delta u(k+j) = \Delta u(k+N_c-1)$ for $j = N_c, \dots, N_p - 1$, or equivalently $\Delta^2 u(k+j) = \Delta u(k+j) - \Delta u(k+j-1) = 0$ 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, if necessary the model and the state estimate are updated using new information of the measurements, and the optimization is restarted.

Define the vectors

$$\tilde{u}(k) = \begin{bmatrix} u(k) \\ \vdots \\ u(k+N_p-1) \end{bmatrix}, \quad \tilde{y}(k) = \begin{bmatrix} y(k) \\ \vdots \\ y(k+N_p-1) \end{bmatrix}.$$

Now the (noise-free) MPL-MPC problem for event step k can be defined as:

$$\min_{\tilde{u}(k)} J_{\text{out}}(k) + \lambda J_{\text{in}}(k)$$

$$\text{subject to}^2 \quad x(k+j) = A \otimes x(k+j-1) \oplus B \otimes u(k+j) \quad \text{for } j = 0, \dots, N_p - 1 \quad (3)$$

$$y(k+j) = C \otimes x(k+j) \quad \text{for } j = 0, \dots, N_p - 1 \quad (4)$$

$$\Delta u(k+j) \geq 0 \quad \text{for } j = 0, \dots, N_p - 1 \quad (5)$$

$$\Delta^2 u(k+j) = 0 \quad \text{for } j = N_c, \dots, N_p - 1 \quad (6)$$

$$A_c(k)\tilde{u}(k) + B_c(k)\tilde{y}(k) \leq c_c(k), \quad (7)$$

where (7) represents the linear constraints on the inputs and the outputs.

¹More specifically, for a manufacturing system, $x(k)$ contains the time instants at which the processing units start working for the k th time, $u(k)$ the time instants at which the k th batch of raw material is fed to the system, and $y(k)$ the time instants at which the k th batch of finished product leaves the system.

²In Section 4 we will see that it is possible to rewrite (3)–(4) as an expression of the form $\tilde{y}(k) = \tilde{C} \otimes x(k-1) \oplus \tilde{D} \otimes \tilde{u}(k)$ for properly defined matrices \tilde{C} and \tilde{D} .

We conclude this section with some results on a class of $(\max,+)$ functions. Let $\mathcal{S}_{\text{mpns}}$ be the set of max-plus-nonnegative-scaling functions³, i.e., functions f of the form $f(x) = \max_i(\alpha_{i,1}x_1 + \dots + \alpha_{i,n}x_n + \beta_i)$ with $x \in \mathbb{R}_\varepsilon^n$ and $\alpha_{i,j} \in \mathbb{R}^+$ and $\beta_i \in \mathbb{R}$, where \mathbb{R}^+ is the set of the nonnegative real numbers. If we want to stress that f is a function of x we will denote this by $f \in \mathcal{S}_{\text{mpns}}(x)$.

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

Proof: This is a consequence of the fact that for $x, y, z, v \in \mathbb{R}_\varepsilon$ and $\rho \in \mathbb{R}^+$ we have $\max(x, y) \oplus \max(z, v) = \max(\max(x, y), \max(z, v)) = \max(x, y, z, v)$, $\max(x, y) \otimes \max(z, v) = \max(x, y) + \max(z, v) = \max(x + z, x + v, y + z, y + v)$ and $\rho \max(x, y) = \max(\rho x, \rho y)$. \diamond

Lemma 2 *If $f \in \mathcal{S}_{\text{mpns}}$ 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 $f(\tilde{x}) \leq f(\hat{x})$. \diamond

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 MPL system:

$$x(k) = A(k) \otimes x(k-1) \oplus B(k) \otimes u(k) \quad (8)$$

$$y(k) = C(k) \otimes x(k) \quad (9)$$

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 $e(k)$. In this paper 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. Furthermore, the system matrices of an MPL model usually consist of sums or maximizations of internal process times, transportation times, etc. (see, e.g., [2] or 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 uncertain system matrices belong to $\mathcal{S}_{\text{mpns}}$:

$$A(k) \in \mathcal{S}_{\text{mpns}}^{n \times n}(e(k)), \quad B(k) \in \mathcal{S}_{\text{mpns}}^{n \times m}(e(k)), \quad C(k) \in \mathcal{S}_{\text{mpns}}^{l \times n}(e(k)) . \quad (10)$$

4 Prediction model

We collect the uncertainty over the interval $[k, k + N_p - 1]$ in one vector

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

³It is easy to verify that max-plus-nonnegative-scaling functions are convex and piecewise affine.

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 $\tilde{e}(k)$. Now it is easy to verify that the prediction model, i.e., the prediction of the future outputs for the system (8)–(9) is given by

$$\tilde{y}(k) = \tilde{C}(\tilde{e}(k)) \otimes x(k-1) \oplus \tilde{D}(\tilde{e}(k)) \otimes \tilde{u}(k) , \quad (11)$$

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

$$\tilde{C}(\tilde{e}(k)) = \begin{bmatrix} \tilde{C}_1(\tilde{e}(k)) \\ \vdots \\ \tilde{C}_{N_p}(\tilde{e}(k)) \end{bmatrix}, \quad \tilde{D}(\tilde{e}(k)) = \begin{bmatrix} \tilde{D}_{11}(\tilde{e}(k)) & \cdots & \tilde{D}_{1N_p}(\tilde{e}(k)) \\ \vdots & \ddots & \vdots \\ \tilde{D}_{N_p 1}(\tilde{e}(k)) & \cdots & \tilde{D}_{N_p N_p}(\tilde{e}(k)) \end{bmatrix}$$

where

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

and

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

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

Proof: This is a direct consequence of the definition of $\tilde{C}(\tilde{e}(k))$, $\tilde{D}(\tilde{e}(k))$ and (11) in combination with (10) and Lemma 1. \diamond

5 Worst-case criterion MPC

Recall that in MPL-MPC we want to minimize the criterion

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

where J_{out} represents the tracking error and J_{in} is related to the input dates. We aim to find the optimal $(\tilde{u}(k), \tilde{y}(k))$ that minimizes $J(\tilde{y}(k), \tilde{u}(k))$, where $\tilde{y}(k)$ and $\tilde{u}(k)$ are related by (11). Note that, in contrast to the noise-free case, the relation between $\tilde{u}(k)$ and $\tilde{y}(k)$ is not unique anymore in the perturbed case because of the (bounded) perturbation $\tilde{e}(k)$. Instead of considering general linear constraints (7) on the inputs and outputs as was done in [8, 9], we will only consider linear constraints $A_c(k)\tilde{u}(k) \leq c_c(k)$ on the input for the perturbed case. A typical example of such a constraint is an upper and lower bound for the input rate:

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

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

$$\begin{aligned} & \min_{\tilde{u}(k)} \max_{\tilde{e}(k) \in \mathcal{E}_{\tilde{e}}} J(\tilde{y}(k), \tilde{u}(k)) \\ & \text{subject to } \tilde{y}(k) = \tilde{C}(\tilde{e}(k)) \otimes x(k-1) \oplus \tilde{D}(\tilde{e}(k)) \otimes \tilde{u}(k) \end{aligned} \quad (12)$$

$$\Delta u(k+j) \geq 0 \quad \text{for } j = 0, \dots, N_p - 1 \quad (13)$$

$$\Delta^2 u(k+j) = 0 \quad \text{for } j = N_c, \dots, N_p - 1 \quad (14)$$

$$A_c(k)\tilde{u}(k) \leq c_c(k) . \quad (15)$$

We now will eliminate (12) by substituting it in the cost criterion and by maximizing the result over all possible $\tilde{e}(k)$. For a fixed $\tilde{u}(k)$ 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 $\tilde{u}(k)$, we let⁴

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

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

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

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

$$\text{subject to } \Delta u(k+j) \geq 0 \quad \text{for } j = 0, \dots, N_p - 1 \quad (16)$$

$$\Delta^2 u(k+j) = 0 \quad \text{for } j = N_c, \dots, N_p - 1 \quad (17)$$

$$A_c(k)\tilde{u}(k) \leq c_c(k) . \quad (18)$$

Now we make the following assumptions:

Assumption A1: J_{out} is a nondecreasing⁵, 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 systems context. As a consequence, they are not overly restrictive:

- If the due dates r for the outputs of the systems are known and if we have to pay a penalty for every delay, a possible output cost criterion is the tardiness:

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

where \tilde{r} is defined in a similar way as \tilde{y} . Clearly, $J_{\text{out},1}$ satisfies Assumption A1. Another objective function that satisfies Assumption A1 is the maximal output delay:

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

- For the input cost criterion we could take [8, 9]:

$$J_{\text{in},0}(\tilde{u}(k)) = \tilde{u}^T(k)\tilde{u}(k) \quad \text{or} \quad J_{\text{in},1}(\tilde{u}(k)) = \sum_i \tilde{u}_i(k)$$

which minimize the input time instants, or

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

⁴Note that $J_{\text{in}}(k)$ does not depend on $\tilde{e}(k)$.

⁵The function $f : \mathbb{R}^n \rightarrow \mathbb{R}$ is nondecreasing if for any $x, y \in \mathbb{R}^n$ with $x_i \leq y_i$ for $i = 1, \dots, n$, we have $f(x) \leq f(y)$.

which maximizes the input time instants⁶. Clearly, $J_{\text{in},0}$, $J_{\text{in},1}$ and $J_{\text{in},2}$ 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: The function J_{in} is convex in \tilde{u} by Assumption A2. Furthermore, the constraints (16)–(18) only depend on \tilde{u} and they are convex in \tilde{u} . So we only have to prove that $J_{\text{out}}^\#$ is convex in \tilde{u} . Define for $0 \leq \rho \leq 1$:

$$\begin{aligned}\tilde{u}_3(k) &= \rho\tilde{u}_1(k) + (1 - \rho)\tilde{u}_2(k) & \tilde{e}_1^\#(k) &= \arg \max_{\tilde{e}(k) \in \mathcal{E}_\varepsilon} J_{\text{out}}(\tilde{y}(\tilde{e}(k), \tilde{u}_1(k))) \\ \tilde{e}_2^\#(k) &= \arg \max_{\tilde{e}(k) \in \mathcal{E}_\varepsilon} J_{\text{out}}(\tilde{y}(\tilde{e}(k), \tilde{u}_2(k))) & \tilde{e}_3^\#(k) &= \arg \max_{\tilde{e}(k) \in \mathcal{E}_\varepsilon} J_{\text{out}}(\tilde{y}(\tilde{e}(k), \tilde{u}_3(k))).\end{aligned}$$

Now we have⁷:

$$\begin{aligned}[\tilde{y}_3(\tilde{e}_3^\#, \tilde{u}_3)]_i &= \left[\tilde{C}(\tilde{e}_3^\#) \quad \tilde{D}(\tilde{e}_3^\#) \right]_i \otimes \begin{bmatrix} x(k-1) \\ \tilde{u}_3(k) \end{bmatrix} && \text{(by (11))} \\ &= \max_\ell \left(\left[\tilde{C}(\tilde{e}_3^\#) \quad \tilde{D}(\tilde{e}_3^\#) \right]_{i,\ell} + \begin{bmatrix} x(k-1) \\ \tilde{u}_3(k) \end{bmatrix}_\ell \right) \\ &= \max_\ell \left(\left[\tilde{C}(\tilde{e}_3^\#) \quad \tilde{D}(\tilde{e}_3^\#) \right]_{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(\left[\begin{array}{cc} \rho\tilde{C}(\tilde{e}_3^\#) + (1 - \rho)\tilde{C}(\tilde{e}_3^\#) & \rho\tilde{D}(\tilde{e}_3^\#) + (1 - \rho)\tilde{D}(\tilde{e}_3^\#) \end{array} \right]_{i,\ell} + \right. \\ &\quad \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) \\ &\leq \max_\ell \left(\left[\begin{array}{cc} \rho\tilde{C}(\tilde{e}_3^\#) & \rho\tilde{D}(\tilde{e}_3^\#) \end{array} \right]_{i,\ell} + \begin{bmatrix} \rho x(k-1) \\ \rho\tilde{u}_1(k) \end{bmatrix}_\ell \right) \\ &\quad + \max_\ell \left(\left[\begin{array}{cc} (1 - \rho)\tilde{C}(\tilde{e}_3^\#) & (1 - \rho)\tilde{D}(\tilde{e}_3^\#) \end{array} \right]_{i,\ell} + \begin{bmatrix} (1 - \rho)x(k-1) \\ (1 - \rho)\tilde{u}_2(k) \end{bmatrix}_\ell \right) \\ &\quad \text{(since } \max_i(v_i, w_i) \leq \max_i(v_i) + \max_i(w_i) \text{ for } v_i, w_i \in \mathbb{R}_\varepsilon) \\ &\leq \rho \max_\ell \left(\left[\begin{array}{cc} \tilde{C}(\tilde{e}_3^\#) & \tilde{D}(\tilde{e}_3^\#) \end{array} \right]_{i,\ell} + \begin{bmatrix} x(k-1) \\ \tilde{u}_1(k) \end{bmatrix}_\ell \right) \\ &\quad + (1 - \rho) \max_\ell \left(\left[\begin{array}{cc} \tilde{C}(\tilde{e}_3^\#) & \tilde{D}(\tilde{e}_3^\#) \end{array} \right]_{i,\ell} + \begin{bmatrix} x(k-1) \\ \tilde{u}_2(k) \end{bmatrix}_\ell \right) \\ &\leq \rho[\tilde{y}(\tilde{e}_3^\#, \tilde{u}_1)]_i + (1 - \rho)[\tilde{y}(\tilde{e}_3^\#, \tilde{u}_2)]_i\end{aligned}$$

and thus

$$J_{\text{out}}(\tilde{y}_3) \leq J_{\text{out}}(\rho\tilde{y}(\tilde{e}_3^\#, \tilde{u}_1) + (1 - \rho)\tilde{y}(\tilde{e}_3^\#, \tilde{u}_2)) , \quad (19)$$

⁶For a manufacturing system, this would correspond to a production scheme in which raw material is fed to the system as late as possible.

⁷We use $[M]_i$ ($[m]_i$) to denote the i th row (component) of a matrix M (column vector m).

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

$$\begin{aligned}
J_{\text{out}}^{\#}(\rho\tilde{u}_1 + (1 - \rho)\tilde{u}_2) &= J_{\text{out}}^{\#}(\tilde{u}_3) = J_{\text{out}}(\tilde{y}_3(\tilde{e}_3^{\#}, \tilde{u}_3)) \\
&\leq J_{\text{out}}(\rho\tilde{y}(\tilde{e}_3^{\#}, \tilde{u}_1) + (1 - \rho)\tilde{y}(\tilde{e}_3^{\#}, \tilde{u}_2)) && \text{(by 19)} \\
&\leq \rho J_{\text{out}}(\tilde{y}(\tilde{e}_3^{\#}, \tilde{u}_1)) + (1 - \rho)J_{\text{out}}(\tilde{y}(\tilde{e}_3^{\#}, \tilde{u}_2)) \\
&\quad \text{(since } J_{\text{out}} \text{ is convex in } \tilde{y} \text{ by Assumption A1)} \\
&\leq \rho J_{\text{out}}(\tilde{y}(\tilde{e}_1^{\#}, \tilde{u}_1)) + (1 - \rho)J_{\text{out}}(\tilde{y}(\tilde{e}_2^{\#}, \tilde{u}_2)) \\
&\quad \text{(by the definition of } \tilde{e}_1^{\#} \text{ and } \tilde{e}_2^{\#}) \\
&\leq \rho J_{\text{out}}^{\#}(\tilde{u}_1) + (1 - \rho)J_{\text{out}}^{\#}(\tilde{u}_2) .
\end{aligned}$$

Hence, $J_{\text{out}}^{\#}$ is a convex function of \tilde{u} . \diamond

So the outer worst-case MPC problem is a convex problem, which can be solved very efficiently using, e.g., an interior-point algorithm [14, 18].

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

$$\max_{\tilde{e}(k) \in \mathcal{E}_{\tilde{e}}} J_{\text{out}}(\tilde{y}(\tilde{e}, \tilde{u})) \quad (20)$$

$$\text{subject to } \tilde{y}(\tilde{e}, \tilde{u}) = \tilde{C}(\tilde{e}) \otimes x(k-1) \oplus \tilde{D}(\tilde{e}) \otimes \tilde{u} . \quad (21)$$

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

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

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

$$\max_{\tilde{e}(k) \in \mathcal{E}_{\tilde{e}, \text{max}}^{\text{v}}} J_{\text{out}}(\tilde{C}(\tilde{e}) \otimes x(k-1) \oplus \tilde{D}(\tilde{e}) \otimes \tilde{u}) . \quad (22)$$

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

Proof: If 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 [16, Theorem 5.1]. Functions that belong to $\mathcal{S}_{\text{mpns}}$ are convex. Since for a given \tilde{u} we have $\tilde{y}(\tilde{e}, \tilde{u}) \in \mathcal{S}_{\text{mpns}}$ by Lemma 3, \tilde{y} is convex as a function of \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} . \diamond

Proposition 6 *If Assumption A1 holds, then an optimal solution of the reduced inner worst-case MPC problem (22) is also an optimal solution of the (full) inner worst-case MPC problem (20)–(21).*

Proof: First we prove that the maximum in the (full) inner worst-case MPC problem (20)–(21) 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 we have $\nexists \tilde{e} \in \mathcal{E}_{\tilde{e}} \text{ with } \tilde{e} \neq \tilde{e}_{\text{max}} \text{ and } \tilde{e}_{\text{max}} \leq \tilde{e}$. Indeed, from Lemmas 2 and 3 it follows that if $\tilde{e}_1 \leq \tilde{e}_2$ then we have $\tilde{y}(\tilde{e}_1) \leq \tilde{y}(\tilde{e}_2)$ 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 $\tilde{e}_{\text{max}}^{\text{v},j}$ of the face of $\mathcal{E}_{\tilde{e}}$ to which \tilde{e}_{max} belongs. Since for a given $x(k-1)$ and $\tilde{u}(k)$ the function J_{out} is convex in \tilde{e} by Lemma 5 and thus also quasi-convex, we have $J_{\text{out}}(\tilde{y}(\tilde{e}_{\text{max}}, \tilde{u})) \leq \max_j J_{\text{out}}(\tilde{y}(\tilde{e}_{\text{max}}^{\text{v},j}, \tilde{u}))$. Hence, an optimal solution of the reduced inner problem is also an optimal solution of the full inner problem. \diamond

The set $\mathcal{E}_{\tilde{e},\text{max}}^{\text{v}}$ 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 [12, 13]⁸. Note that the computation can be made more efficient by already discarding the vertex points that cannot result in vertex points that will belong to $\mathcal{E}_{\tilde{e},\text{max}}^{\text{v}}$ during the computation (cf. [7]). 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

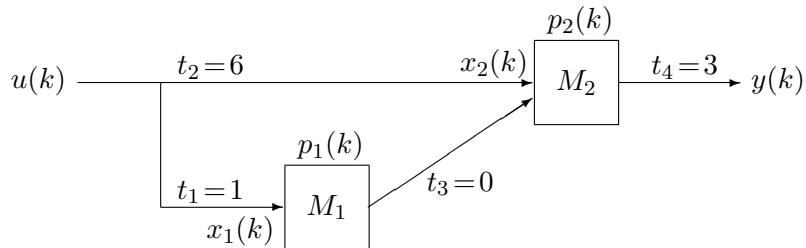


Figure 1: A production system.

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_2 (with a certain transportation delay), whereas the other part of the batch first goes to machine M_1 for pre-processing. Afterward, assembly takes place on machine M_2 . 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). The values of the transportation times t_j are specified in the figure.

Define $u(k)$: time instant at which the system is fed for the k th time
 $y(k)$: time instant at which the k th product leaves the system
 $x_i(k)$: time instant at which machine i starts for the k th time
 $p_i(k)$: processing time on machine i for the k th 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 $p_1(k) \in [2, 6]$ and $p_2(k) \in [1, 6]$ and $p_1(k) + p_2(k) \leq 9$.

⁸The paper [12] also provides more information on the complexity of computing the set of vertices of $\mathcal{E}_{\tilde{e}}$ and a (crude) upper bound for the number of elements of $\mathcal{E}_{\tilde{e}}$ (and thus also of $\mathcal{E}_{\tilde{e},\text{max}}^{\text{v}}$).

Note that this implies that if $p_1(k) < 5$ then the direct path from the input to machine M_2 is the longest, whereas if $p_1(k) > 5$ the path from the input via machine M_1 to machine M_2 is the longest. The constraint $p_1(k) + p_2(k) \leq 9$ indicates that the processing times are somehow related. For example, if the processing time $p_1(k)$ is larger than 3, the upper bound for $p_2(k)$ will decrease. So, if $p_1(k)$ is large, then $p_2(k)$ will be small, and vice versa.

From the system equations

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

we derive:

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

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

$$A(k) = \begin{bmatrix} e_1(k) & \varepsilon \\ e_1(k) + e_3(k) & e_2(k) \end{bmatrix}, B(k) = \begin{bmatrix} 1 \\ \max(6, e_3(k) + 1) \end{bmatrix}, C(k) = \begin{bmatrix} \varepsilon & e_4(k) + 3 \end{bmatrix} .$$

Note that the entries of $A(k)$, $B(k)$ and $C(k)$ belong to $\mathcal{S}_{\text{mpns}}(e(k))$.

For this perturbed MPL system we solve the worst-case MPC problem with $N_p = 4$, $N_c = 2$ and the cost criterion⁹

$$\begin{aligned} J(\tilde{y}(k), \tilde{u}(k)) &= J_{\text{out},1}(\tilde{y}(k)) + \lambda J_{\text{in},2}(\tilde{u}(k)) \\ &= \sum_{i=0}^{N_p-1} \max(y(k+i) - r(k+i), 0) - \lambda \sum_{i=0}^{N_p-1} u(k+1) \end{aligned} \quad (23)$$

with $\lambda = 0.01$. The initial state is equal to $x(0) = [5 \ 10]^T$. The due date signal r is an monotonically increasing function with initial value $r(0) = 18$ and a random increment within the bounds $6.1 \leq \Delta r(k) \leq 6.5$. With the above choice of the cost criterion, we can recast the worst-case MPC problem as a linear programming problem (see Appendix), which we have solved using the linear programming function `linprog` of Matlab.

The top points \tilde{e}_{max}^v correspond to the critical points $(p_1(k+j), p_2(k+j)) = (3, 6)$ and $(p_1(k+j), p_2(k+j)) = (6, 3)$. The set $\mathcal{E}_{\tilde{e}_{\text{max}}^v}^v$ consists of $2^{N_p+1} = 32$ top points¹⁰, corresponding to all 2^5 combinations of the two critical points $(p_1(k+j), p_2(k+j))$ for $j = -1, 0, \dots, N_p - 1$.

The optimal MPC input sequence is computed for $k = 1, \dots, 100$, 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

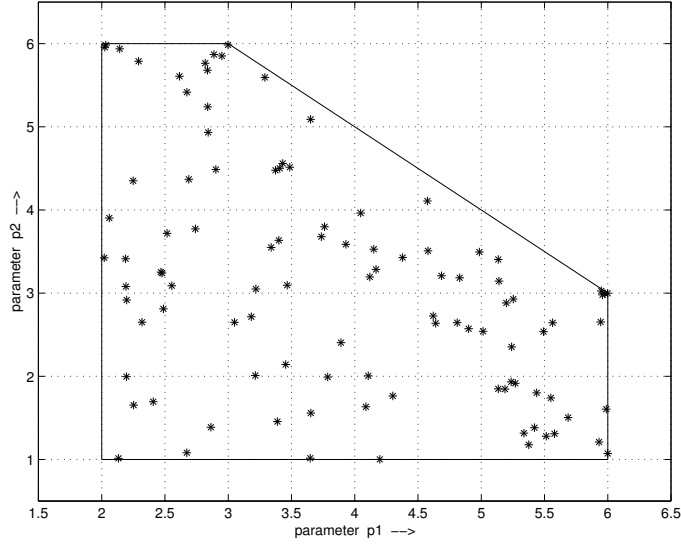


Figure 2: The parameters $\{(p_1^{\text{true}}(k), p_2^{\text{true}}(k))\}_{k=0, \dots, 100}$ for the true system.

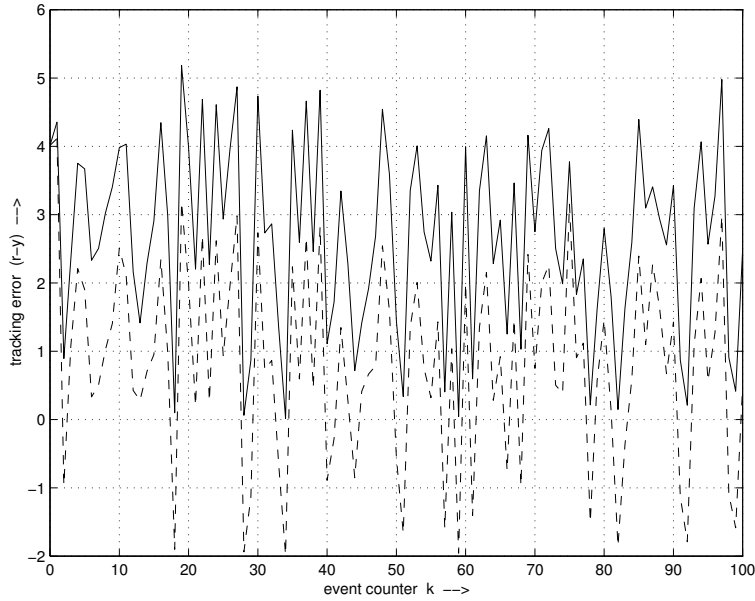


Figure 3: The difference $r(k) - y(k)$ between the due date signal r and the output date signal y (solid: worst-case algorithm, dashed: fixed model algorithm).

horizon strategy). In the experiment, the true system is simulated for a random sequence $\{(p_1^{\text{true}}(k), p_2^{\text{true}}(k))\}_{k=0, \dots, 100}$ in the allowed region (see Figure 2).

Figure 3 gives the difference between the due date signal r and the output date signal y . To see the influence of the worst-case approach, the design is also done with a fixed model

⁹Since the production system is a single-input single-output system, we have $\tilde{y}_i(k) = y(k + i - 1)$.

¹⁰In general, if $e(k)$ has 4 components, then we would expect 4^{N_p} components in $\tilde{e}(k)$, but since $e(k)$ and $e(k + 1)$ have 2 components in common, there will be 2^{N_p+1} components in $\tilde{e}(k)$.

in one of the critical points $(p_1, p_2) = (6, 3)$. The fixed model algorithm minimizes the cost criterion (23) using the algorithm of [9], so $u(k)$ is chosen as large as possible such that $r \geq y$ (then $J_{\text{out},1}$ is zero). Perturbations will often push the real value y over the due date r (so $r < y$) as can be seen in Figure 3. This implies that for the fixed model algorithm the due dates are not always met. The worst-case algorithm minimizes the cost criterion (23) for the worst-case realizations of perturbation \tilde{e} . For all possible perturbations we will find $y \leq r$, for the worst-case perturbation we find $y = r$. In Figure 3 it can be observed that for the worst-case approach, the difference signal is always larger than zero, which means that the due dates are always met. For the events $k = 35$ and $k = 60$ the solid line is close to zero, which means that the actual perturbation is close to worst-case.

To compare the closed-loop results numerically, we introduce a closed-loop criterion

$$J_{\text{cl}} = \sum_{k=1}^{100} \max(y(k) - r(k), 0) - \lambda \sum_{k=1}^{100} u(k)$$

which gives an indication of performance over the larger time window [1,100] at which we aim with the cost criterion (23) in each MPC interval $[k, k + N_p - 1]$. For the worst-case model we find $J_{\text{cl}} = -329$, while for the fixed model we obtain $J_{\text{cl}} = -302$. Obviously the worst-case approach has improved the performance in comparison with the fixed model approach.

7 Conclusions

We have further extended the MPC framework to include max-plus-linear discrete event systems with modeling errors, noise and/or disturbances. For max-plus-linear systems, the entries of the system matrices correspond to production times or transportation times. Therefore, modeling noise (i.e., variation in the processing times) can be modeled as an additive term to the system matrices. Modeling errors (caused in the modeling or identification phase) also occur as additive uncertainty on the system matrices and can be added to the noise component. In order to handle perturbations due to modeling errors and noise, 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 shown how the resulting optimization problem can be computed efficiently using a two-level optimization approach, where the outer problem is convex. For specific choices, the problem can be recast as a linear programming problem.

Topics for future include: inclusion of a state observer (based on partial information of the current and previous states), extension to infinite horizon MPC, investigation of stability issues and determination of tuning rules (for λ , N_p , N_c) in the perturbed case, characterization of the computational complexity of the worst-case MPC problem, complexity reduction and approximation to further improve the efficiency of our approach, inclusion of constraints on the outputs, and extension to a probabilistic uncertainty framework.

Acknowledgments

This research was partially sponsored by the TMR project ALAPEDES (Algebraic Approach to Performance Evaluation of Discrete Event Systems) of the European Community Training and Mobility of Researchers Program (network contract ERBFMRXCT960074), and by the FWO (Fund for Scientific Research–Flanders) Research Community ICCoS (Identification and Control of Complex Systems).

References

- [1] F. Allgöwer, T.A. Badgwell, J.S. Qin, J.B. Rawlings, and S.J. Wright, “Nonlinear predictive control and moving horizon estimation – An introductory overview,” in *Advances in Control: Highlights of ECC '99* (P.M. Frank, ed.), pp. 391–449, Springer, 1999.
- [2] F. Baccelli, G. Cohen, G.J. Olsder, and J.P. Quadrat, *Synchronization and Linearity*. New York: John Wiley & Sons, 1992.
- [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 B. De Moor, “The extended linear complementarity problem,” *Mathematical Programming*, vol. 71, no. 3, pp. 289–325, Dec. 1995.
- [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] 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.
- [10] 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.
- [11] F. Lin, “Robust and adaptive supervisory control of discrete event systems,” *IEEE Transactions on Automatic Control*, vol. 38, no. 12, pp. 1848–1852, Dec. 1993.
- [12] 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.
- [13] 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.
- [14] Y. Nesterov and A. Nemirovskii, *Interior-Point Polynomial Algorithms in Convex Programming*. Philadelphia, Pennsylvania: SIAM, 1994.
- [15] 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.
- [16] R.T. Rockafellar, *Convex Analysis*. Princeton, New Jersey: Princeton University Press, 1970.
- [17] T. van den Boom and B. De Schutter, “MPC for max-plus-linear systems: Closed-loop behavior and tuning,” in *Proceedings of the 2001 American Control Conference*, Arlington, Virginia, pp. 325–330, July 2001.
- [18] S.J. Wright, *Primal-Dual Interior Point Methods*. Philadelphia, Pennsylvania: SIAM, 1997.
- [19] 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.

Appendix

Consider the cost criterion (23). Let $L = 2^{N_p+1}$ and $\mathcal{E}_{\tilde{e}, \max}^v = \{\tilde{e}_{\max}^{v, \ell} | \ell = 1, 2, \dots, L\}$. Introduce the auxiliary variables $\eta_{\ell, i}$ defined as

$$\eta_{\ell, i} = \max(\tilde{y}_i(\tilde{e}_{\max}^{v, \ell}, \tilde{u}) - \tilde{r}_i, 0) \quad (24)$$

for $\ell = 1, \dots, L = 2^{N_p+1}$ and $i = 1, \dots, N_p$. An upper bound for the worst-case cost function $J_{\text{out}, 1}^{\#}(\tilde{u})$ is given by the scalar γ satisfying

$$\gamma \geq \sum_i \eta_{\ell, i} \quad \text{for all } \ell = 1, \dots, L \quad (25)$$

Using the equality

$$\tilde{y}_i(\tilde{e}_{\max}^{v, \ell}, \tilde{u}) = [\tilde{C}(\tilde{e}_{\max}^{v, \ell}) \otimes x(k-1) \oplus \tilde{D}(\tilde{e}_{\max}^{v, \ell}) \otimes \tilde{u}]_i$$

(cf. (11)), equation (24) leads to the following inequalities:

$$\eta_{\ell, i} \geq [\tilde{C}(\tilde{e}_{\max}^{v, \ell})]_{ip} + x_p(k-1) - \tilde{r}_i \quad \text{for } \ell = 1, \dots, L, \quad i = 1, \dots, N_p, \quad p = 1, \dots, n \quad (26)$$

$$\eta_{\ell, i} \geq [\tilde{D}(\tilde{e}_{\max}^{v, \ell})]_{iq} + \tilde{u}_q(k) - \tilde{r}_i \quad \text{for } \ell = 1, \dots, L, \quad i = 1, \dots, N_p, \quad q = 1, \dots, N_p \quad (27)$$

$$\eta_{\ell, i} \geq 0 \quad \text{for } \ell = 1, \dots, L, \quad i = 1, \dots, N_p \quad (28)$$

If we define the vector $z = [\gamma \quad \eta_1 \quad \dots \quad \eta_L \quad \tilde{u}^T]^T$, then it is easy to verify that the minimization of the worst-case cost criterion $J(\tilde{y}(k), \tilde{u}(k)) = J_{\text{out}, 1}(\tilde{y}(k)) + \lambda J_{\text{in}, 2}(\tilde{u}(k))$ can be recast as the optimization problem

$$\min_z \gamma - \lambda \sum_i \tilde{u}_i$$

subject to (16)–(18) and (25)–(28).

Since the cost criterion and all the constraints of this optimization problem are linear in vector z , the optimization problem is a linear programming problem, which can be solved very efficiently.

Note that the minimization of γ forces the inequality (25) to be tight. A similar statement holds for the system of inequalities (26)–(28): for each ℓ, i at least one of the inequalities in (26)–(28) is tight. Hence, we have indeed $\eta_{\ell, i} = \max(\tilde{y}_i(\tilde{e}_{\max}^{v, \ell}, \tilde{u}) - \tilde{r}_i, 0)$ and $\gamma(\tilde{u}) = \max_{\ell} J_{\text{out}, 1}(\tilde{y}(\tilde{e}_{\max}^{v, \ell}, \tilde{u}))$.