Delft University of Technology

Delft Center for Systems and Control

Technical report 06-035

Model Predictive Control for Uncertain Max-Min-Plus-Scaling Systems*

I. Necoara, B. De Schutter, T. van den Boom, and H. Hellendoorn

To cite this work, please refer to the published version:

I. Necoara, B. De Schutter, T. van den Boom, and H. Hellendoorn, "Model predictive control for uncertain max-min-plus-scaling systems," *International Journal of Control*, vol. 81, no. 5, pp. 701–713, May 2008. doi:10.1080/00207170601094404

Delft Center for Systems and Control
Delft University of Technology
Mekelweg 2, 2628 CD Delft
The Netherlands
phone: +31-15-278.24.73 (secretary)
URL: https://www.dcsc.tudelft.nl

Model predictive control for uncertain max-min-plus-scaling systems

I. Necoara*, B. De Schutter, T. van den Boom and H. Hellendoorn

Delft Center for Systems and Control, Delft University of Technology, Mekelweg 2, 2628 CD Delft, The Netherlands

Abstract: In this paper we extend the classical min-max model predictive control framework to a class of uncertain discrete event systems that can be modeled using the operations maximization, minimization, addition and scalar multiplication, and that we call max-min-plus-scaling (MMPS) systems. Provided that the stage cost is an MMPS expression and considering only linear input constraints then the open-loop min-max model predictive control problem for MMPS systems can be transformed into a sequence of linear programming problems. Hence, the min-max model predictive control problem for MMPS systems can be solved efficiently, despite the fact that the system is nonlinear. A min-max feedback model predictive control approach using disturbance feedback policies is also presented, which leads to improved performance compared to the open-loop approach.

1 Introduction

Discrete event systems (DES) are event-driven dynamical systems, i.e. the state transitions are initiated by events rather than a clock. An important class of DES is the class of max-min-plus-scaling (MMPS) systems, the evolution equations of which can be described using the operations maximization, minimization, addition and scalar multiplication. We show that this class encompasses several other classes of DES such as max-plus-linear systems, bilinear max-plus systems and max-min-plus systems. Using the results of Heemels *et al.* (2001), we can prove that MMPS systems are also equivalent with a particular class of hybrid systems, called continuous piecewise (PWA) systems. PWA systems are defined by partitioning the state space of the system in a finite number of polyhedral regions and associating to each region a different affine dynamic. The relation between PWA and MMPS systems is useful for the investigation of structural properties of PWA systems such as observability and controllability but also in designing controller schemes like model predictive control (MPC) (Bemporad *et al.*, 1999; Johansson, 2003).

MPC (Maciejowski, 2002; Mayne *et al.*, 2000) is a very popular control methodology in the process industry. MPC provides many attractive features: it is an easy-to-tune method, it is applicable to multi-variable systems, it can handle constraints in a systematic way, and it is capable of tracking pre-scheduled reference signals. In MPC at each sample step the optimal control inputs that minimize a given performance criterion over a given prediction horizon are computed, and applied using a receding horizon approach until new measurements become available. Feedback is incorporated by using these measurements to update the optimization problem for the next step.

Several authors have developed control design methods (e.g. MPC) for some specific subclasses of DES or hybrid systems (Cassandras *et al.*, 2001; Bemporad *et al.*, 1999; Necoara , 2006), in particular for max-plus-linear systems (Baccelli *et al.*, 1992; Menguy *et al.*, 1998; Cottenceau *et al.*, 2001; Necoara *et al.*, 2005; De Schutter *et al.*, 2001) or PWA systems (Kerrigan *et al.*, 2002; Rakovic *et al.*, 2004; De Schutter *et al.*, 2004; Necoara *et al.*, 2004). Using the work of De Schutter *et al.* (2004) in which MPC for MMPS (or equivalently for continuous PWA) systems for the deterministic case without disturbances and modeling errors is proposed, we further extend in this paper the conventional min-max MPC approach for the cases with bounded disturbances and modeling errors. An important difference between MPC and some other control methods is the explicit use of a prediction model. Because the

*Corresponding author. Email: i.necoara@yahoo.com

models play an important role in MPC, we must take the disturbance and modeling errors into account when using MPC. Both features can perturb the system by introducing uncertainty in the system equations and ignoring them may lead to a bad tracking or even to unstable closed-loop behavior. In this paper we model disturbances and modeling errors in one single framework by including extra terms in the equations of MMPS system.

Note that there are some results in the literature on specific classes of uncertain DES and hybrid systems (Kerrigan *et al.*, 2002; Rakovic *et al.*, 2004; Necoara *et al.*, 2005) but to the authors' best knowledge this is the first time that such an approach is used for the MMPS framework. Some papers (Kerrigan *et al.*, 2002; Rakovic *et al.*, 2004; Necoara *et al.*, 2005) focus on worst-case problems, which basically involves finding the maximum of the cost criterion over some bounded disturbance set and then minimization over the feasible input set. In Kerrigan *et al.* (2002); Rakovic *et al.* (2004) dynamic programming was used to solve the min-max feedback MPC problem for continuous PWA systems with bounded disturbances. The core difficulty with the dynamic programming approach is that optimizing over feedback policies with arbitrary nonlinear functions is in general a computationally hard problem. Moreover, in the dynamic programming approach it is difficult to take into consideration variable input constraints, which is typically the case in DES (e.g. bounded rate variation $m \leq u(k+1) - u(k) \leq M$). In Necoara *et al.* (2005) it is proved that the min-max feedback MPC for max-plus systems is a convex problem if some assumptions about the cost function and constraints are fulfilled. The main difficulty in this case is represented by the - in the worst case - exponential number of constraints, that result from the transformation of max constraints in linear constraints. The approach proposed in this paper addresses some of these issues.

In this paper we consider uncertain MMPS systems, and thus also uncertain continuous PWA systems. The paper is organized as follows. As an introduction to our discussion and to make the paper self-contained, a brief overview of MMPS and PWA systems is given, and MPC for them as it was developed in De Schutter *et al.* (2004) is presented in Section 2. We also discuss some results on multi-parametric linear programming. In Section 3 we discuss open-loop MPC for uncertain MMPS systems. We obtain an efficient MPC method that is based on minimizing the worst-case cost criterion. One of the key results of this paper is showing that the optimization problem at each MPC step can be transformed into a sequence of linear programming problems, for which efficient solution methods exist. It is well-known (Mayne *et al.*, 2000) that in the presence of disturbance, a feedback controller performs better than an open-loop controller. Therefore, in Section 4 we introduce feedback in the worst-case MPC optimization problem, optimizing over disturbance feedback policies. Although this approach was applied successfully to linear systems (Goulart *et al.*, 2006), the extension to DES has not been done yet, this paper being the first attempt. In Section 5 we discuss the complexity of the proposed and existing algorithms. We conclude with a worked example in Section 6 where these two approaches are compared.

2 Preliminaries

2.1 *Equivalence between MMPS and continuous PWA systems*

Definition 2.1 A scalar-valued MMPS function $f : \mathrm{R}^n \to \mathrm{R}$ is defined by the recursive relation:

$$f(x) = x_i | \alpha | \max(f_k(x), f_l(x)) | \min(f_k(x), f_l(x)) | f_k(x) + f_l(x) | \beta f_k(x),$$

where $i \in \{1, ..., n\}$, $\alpha, \beta \in \mathrm{R}$ and $f_k, f_l : \mathrm{R}^n \to \mathrm{R}$ are again MMPS functions, and the symbol $|$ stands for *"or"*. For vector-valued MMPS functions the above statements hold component-wise.

An MMPS system is written in the following form:

$$x(k+1) = \mathcal{M}_x(x(k), u(k)) \quad (1)$$

$$y(k) = \mathcal{M}_y(x(k), u(k)), \quad (2)$$

where $\mathcal{M}_x$, $\mathcal{M}_y$ are vector-valued MMPS functions with input $u \in \mathrm{R}^m$, output $y \in \mathrm{R}^p$ and state $x \in \mathrm{R}^n$ and k is an integer. In the context of DES, k is an event counter, and thus x, u and y correspond to time instants, while in the context of hybrid systems k is a discrete time and x, u and y represent physical variables.

Definition 2.2 A vector-valued function $f : \mathrm{R}^n \to \mathrm{R}^m$ is said to be a *continuous* PWA function if there exists a finite family $\mathcal{C}_1, ..., \mathcal{C}_N$ of closed polyhedral regions that covers R^n and for each $i \in \{1, ..., N\}$, $j \in \{1, ..., m\}$, the component f_j of f can be expressed as $f_j(x) = \alpha_{i,j}^T\, x + \beta_{i,j}$, for any $x \in \mathcal{C}_i$ with $\alpha_{i,j} \in \mathrm{R}^n$, $\beta_{i,j} \in \mathrm{R}$ and if f is continuous on the boundary between any two regions.

A *continuous* PWA system in state space representation is a system of the form:

$$x(k+1) = \mathcal{P}_x(x(k), u(k)) \tag{3}$$

$$y(k) = \mathcal{P}_y(x(k), u(k)), \tag{4}$$

where $\mathcal{P}_x$ and $\mathcal{P}_y$ are continuous PWA functions.

PROPOSITION 2.3 *(De Schutter et al., 2004) Any scalar-valued MMPS function $f : \mathrm{R}^n \to \mathrm{R}$ can be written into min-max canonical form*

$$f(x) = \min_{j \in \{1,...,\hat{l}\}} \max_{i \in T_j} (\alpha_{i,j}^T x + \beta_{i,j}), \tag{5}$$

or into max-min canonical form

$$f(x) = \max_{j \in \{1,...,l\}} \min_{i \in S_j} (\gamma_{i,j}^T x + \delta_{i,j}), \tag{6}$$

for some integers $\hat{l}$, l, N; $\{S_j\}_{j=1}^{l}$ and $\{T_j\}_{j=1}^{\hat{l}}$ each are a family of subsets of $\{1, ..., N\}$ and $\alpha_{i,j}, \gamma_{i,j} \in \mathrm{R}^n$, $\beta_{i,j}, \delta_{i,j} \in \mathrm{R}$.

PROPOSITION 2.4 *(De Schutter et al., 2004) Any continuous PWA function having domain R^n can be written as an MMPS function and vice versa.*

COROLLARY 2.5 *Continuous PWA systems and MMPS systems are equivalent in the sense that for a given continuous PWA model there exists an MMPS model (and vice versa) such that the input-output behavior of both models coincides.*

Note that the above propositions imply that any continuous PWA system (3)–(4) can be written in the form (1)–(2), with each component of $\mathcal{M}_x$ and $\mathcal{M}_y$ in min-max canonical form (5) or max-min canonical form (6).

2.2 *MMPS systems and other classes of DES*

In this section we will show that the model (1)–(2) can be considered as a generalized framework that encompasses several subclasses of DES such as: max-plus-linear systems, max-plus-bilinear systems, max-min-plus systems.

2.2.1 ***Max-plus-linear systems.*** Max-plus-linear systems (Baccelli *et al.*, 1992; Cuninghame-Green, 1979; Heidergott *et al.*, 2005) are DES that can be described by a state space model of the following form:

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

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

where the operations $\oplus$ and $\otimes$ are defined by

$$(U \oplus V)_{ij} = u_{ij} \oplus v_{ij} = \max(u_{ij}, v_{ij}) \tag{9}$$

$$(U \otimes W)_{ij} = \bigoplus_{k=1}^{q} u_{ik} \otimes w_{kj} = \max_{k=\{1,...,q\}} (u_{ik} + w_{kj}) \tag{10}$$

for matrices $U, V \in \mathbb{R}_{-\infty}^{p\times q}$, and $W \in \mathbb{R}_{-\infty}^{q\times r}$ with $\mathbb{R}_{-\infty} = \mathbb{R}\cup\{-\infty\}$. Regarding the order of evaluation, the operation $\otimes$ has precedence over $\oplus$. We call $\oplus$ the max-plus-algebraic addition and $\otimes$ the max-plus-algebraic multiplication. This also explains why the model (7)–(8) is called max-plus-linear, i.e. linear in the max-plus-algebraic sense. Loosely speaking, this class corresponds to the class of DES in which there is synchronization (corresponding to maximization) but no concurrency. Max-plus linear DES often arise in the context of manufacturing systems, railway networks, parallel computing, etc.

The model (7)–(8) can be rewritten as

$$\begin{aligned} x_i(k+1) &= \max\big(\max_j(a_{ij}+x_j(k)),\, \max_j(b_{ij}+u_j(k))\big) && \text{for } i=1,2,\dots,n,\\ y_i(k) &= \max_j(c_{ij}+x_j(k)), && \text{for } i=1,2,\dots,l, \end{aligned}$$

which is clearly a special case of an MMPS system.

2.2.2 *Max-plus-bilinear systems.*

Max-plus-bilinear systems are DES that can be described by a state space model of the following form:

$$x(k+1) = A\otimes x(k)\oplus B\otimes u(k)\oplus \bigoplus_{p=1}^{m} L_p\otimes u_p(k)\otimes x(k) \tag{11}$$

$$y(k) = C\otimes x(k)\oplus D\otimes u(k), \tag{12}$$

where $L_p \in \mathbb{R}_{-\infty}^{n\times n}$ for $p = 1,2,\dots,m$. This description is the max-plus-algebraic equivalent of conventional bilinear discrete-time systems. Max-plus-bilinear systems could arise when some of the inputs of a max-plus-linear system of the form (7)–(8) are used as a switch to control the entries of the system matrix A, i.e. the constant system matrix A is replaced by the input-dependent system matrix $A\oplus L_1\otimes u_1(k)\oplus\cdots\oplus L_m\otimes u_m(k)$. Clearly, max-plus-bilinear systems are also a subclass of the MMPS systems.

2.2.3 *Max-plus-polynomial systems.*

The rth max-plus-algebraic power of the scalar variable v is defined by $v^{\otimes^r} = rv$. A max-plus-polynomial p of the scalar variables $v_1, v_2, \dots, v_n$ can be written as

$$p(v_1,v_2,\dots,v_n) = \bigoplus_{i=1}^{q} c_i\otimes {v_1}^{\otimes^{r_{i,1}}}\otimes {v_2}^{\otimes^{r_{i,2}}}\otimes\cdots\otimes {v_n}^{\otimes^{r_{i,n}}}, \tag{13}$$

where c_i and $r_{i,j}$ are scalars.

Max-plus-polynomial systems are a further extension of max-plus-linear and max-plus-bilinear DES. They can be described by a state space model of the following form:

$$x(k) = p_x(x(k-1), u(k)) \tag{14}$$

$$y(k) = p_y(x(k), u(k)), \tag{15}$$

where p_x and p_y are max-plus-polynomials. In van Egmond *et al.* (1999) a subclass of max-plus-polynomial systems has been used in the design of traffic signal switching schemes.

Since (13) can be rewritten as

$$p(v_1,v_2,\dots,v_n) = \max_{i=1,\dots,q}(c_i + r_{i,1}v_1 + r_{i,2}v_2 + \cdots + r_{i,n}v_n), \tag{16}$$

which is an MMPS expression, the system (14)–(15) is also an MMPS system.

2.2.4 ***Max-min-plus systems.*** Max-min-plus systems (or max-min systems as they are called in Olsder (1994)) are described by the model

$$x(k+1) = \mathcal{M}_{\mathrm{mm}x}(x(k), u(k)) \tag{17}$$

$$y(k) = \mathcal{M}_{\mathrm{mm}y}(x(k), u(k)), \tag{18}$$

where $\mathcal{M}_{\mathrm{mm}x}, \mathcal{M}_{\mathrm{mm}y}$ are max-min-plus expressions, i.e. expressions defined recursively by

$$f := x_i | f_k + \alpha | \max(f_k, f_l) | \min(f_k, f_l),$$

where α is a scalar, and f_k and f_l are again max-min-plus expressions. So max-min-plus expressions are special cases of MMPS expressions. This implies that max-min-plus systems are also a subclass of the MMPS systems.

2.3 *MPC for MMPS systems*

In this section we give a short description of MPC for MMPS systems of the form (1)-(2) (see De Schutter *et al.* (2004) for more details). Note that De Schutter *et al.* (2004) does not consider disturbances in the model. In MMPS-MPC we define for each step k a cost criterion

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

over the period $[k, k+N_{\mathrm{p}}-1]$, where N_{p} is the prediction horizon and $\lambda > 0$ is a weighting factor. By optimizing this cost criterion we obtain an optimal input sequence $u^*(k), ..., u^*(k+N_{\mathrm{p}}-1)$, but we apply to our system only the first input sample $u^*(k)$ according to a receding horizon strategy. At the next sample step the whole procedure is repeated.

Now we explain in more detail how MPC for MMPS systems can be implemented efficiently in the case when the cost criterion $J(k)$ is an MMPS function of the input. Assuming that at each step k, the state $x(k)$ can be measured or predicted, we can make an estimation of the output of the model (1)–(2):

$$\hat{y}(k+j|k) = \mathcal{M}_j(x(k), u(k), ..., u(k+j)) \tag{19}$$

at sample step $k+j$, for $j = 0, ..., N_{\mathrm{p}}-1$ using the information available up to sample step k. It is easy to verify that $\mathcal{M}_j$ is an MMPS function of $x(k), u(k), ..., u(k+j)$. Our goal is to track a reference (due dates) signal r. We define the vectors $\tilde{u}(k) = [u^T(k), ..., u^T(k+N_{\mathrm{p}}-1)]^T$, $\tilde{y}(k) = [\hat{y}^T(k|k), ..., \hat{y}^T(k+N_{\mathrm{p}}-1|k)]^T$, and $\tilde{r}(k) = [r^T(k), ..., r^T(k+N_{\mathrm{p}}-1)]^T$.

We consider only linear constraints on the input[1]

$$P(k)\tilde{u}(k) + q(k) \leq 0. \tag{20}$$

In practical situations, such constraints occur when we have to guarantee that the input signal or the rate of variation of the input signal must stay within certain bounds, e.g. $m(k+j) \leq u(k+j) - u(k+j-1) \leq M(k+j)$, where $m(\cdot)$ and $M(\cdot)$ are the lower and upper bounds respectively. As output cost functions one could take:

$$J_{\mathrm{out},1}(k) = \|\tilde{y}(k) - \tilde{r}(k)\|_1, \quad J_{\mathrm{out},\infty}(k) = \|\tilde{y}(k) - \tilde{r}(k)\|_\infty$$
$$J_{\mathrm{out,t}}(k) = \max\{\tilde{y}(k) - \tilde{r}(k), 0\}, \tag{21}$$

which reflect the tracking error or tardiness, and which are MMPS functions of $x(k), \tilde{u}(k), \tilde{r}(k)$. As input cost

[1] We can take into account also constraints on states, but in this case the number of optimization problems that must be solved increases.

function one could take:

$$J_{\rm in,1}(k) = \|\tilde{u}(k)\|_1, \quad J_{\rm in,\infty}(k) = \|\tilde{u}(k)\|_\infty, \quad J_{\rm in,t}(k) = -\sum_i \tilde{u}_i(k), \tag{22}$$

which are also MMPS functions of $\tilde{u}(k)$. Or we can use any other output or input cost criterion that can be expressed as an MMPS function of $\tilde{u}(k)$. We introduce a control horizon $N_{\rm c}$ such that

$$u(k+j) = u(k+N_{\rm c}-1) \quad \text{for} \quad j = N_{\rm c}, ..., N_{\rm p}-1, \tag{23}$$

to decrease the number of degrees of freedom for $\tilde{u}(k)$. As a consequence, we obtain a reduction in computational effort but this also makes the control signal more smooth and the controller more robust. Note that (23) can also be expressed in the form (20).

Since after substitution of $\tilde{y}(k)$ using (19), the cost function $J(k)$ is an MMPS function of $\tilde{u}(k)$ which can be written in min-max canonical form, it follows that at each step k we have to solve an optimization problem of the following form

$$\min_{\tilde{u}(k)} \min_{j\in\{1,...,\hat{l}\}} \max_{i\in T_j} (\alpha_{i,j}^T \tilde{u}(k) + \beta_{i,j}(k)) \tag{24}$$
$$\text{subject to: } P(k)\tilde{u}(k) + q(k) \leq 0,$$

and thus for any $j \in \{1, ..., \hat{l}\}$ we obtain a linear programming problem:

$$\min_{\tilde{u}(k),t(k)} t(k) \tag{25}$$
$$\text{subject to: } \begin{cases} P(k)\tilde{u}(k) + q(k) \leq 0 \\ t(k) \geq \alpha_{i,j}^T \tilde{u}(k) + \beta_{i,j}(k)), \text{ for all } i \in T_j. \end{cases}$$

The linear programming problems are easy to solve using the simplex method or an interior point algorithm (Pardalos *et al.*, 2002). Let $[t^*(k)\ \tilde{u}_{(j)}^{*T}(k)]^T$ be the optimal solution of (25). To obtain the solution of (24), we solve (25) for $j \in \{1, ..., \hat{l}\}$ and afterward we select the $\tilde{u}_{(j)}^*(k)$ for which $\max_{i\in T_j}(\alpha_{i,j}^T(k)\tilde{u}_{(j)}^*(k) + \beta_{i,j}(k))$ is the smallest.

2.4 *Multi-parametric linear programming*

A multi-parametric linear programming (MP-LP) problem is defined (Gal, 1995; Borrelli *et al.*, 2003) as:

$$\max\ c^T x \tag{26}$$
$$\text{subject to} \quad Sx \leq q + U\theta, \tag{27}$$

where $x \in \mathrm{R}^n$ is the optimization variable, $\theta \in \Theta \subseteq \mathrm{R}^s$ is a vector of parameters, $S \in \mathrm{R}^{m\times n}$, $c \in \mathrm{R}^n$, $q \in \mathrm{R}^m$, and $U \in \mathrm{R}^{m\times s}$. We assume that Θ is a polytope given by $\Theta = \{\theta \in \mathrm{R}^s : W\theta \leq \omega\}$. For simplicity, we assume that for any $\theta \in \Theta$ (recall that Θ is bounded), the problem (26)–(27) has a finite optimal solution. Let $V^*(\theta)$ denote the maximum value of the objective function in problem (26)–(27) and $x^*(\theta)$ the optimizer[1] related to $V^*(\theta)$ for any $\theta \in \Theta$. The following proposition which is a slight adaptation of a result of Gal (1995) characterizes the solution of an MP-LP problem:

PROPOSITION 2.6 *With the above notations, the function $V^* : \Theta \to \mathrm{R}$ is a concave MMPS function (i.e. only a min-plus-scaling expression).*

[1] In general, $x^*(\theta)$ is set-valued.

Proof In Gal (1995) it is proved that $V^* : \Theta \to \mathrm{R}$ is a continuous concave PWA function, which implies that V^* is also an MMPS function (according to Proposition 1.4). Actually we know (Pardalos *et al.*, 2002) that any concave, PWA function can be written as:

$$V^*(\theta) = \min_{i \in \{1,\ldots,k\}} (\alpha_i^T \theta + \beta_i).$$

□

The reader is referred to (Borrelli *et al.*, 2003; Kvasnica *et al.*, 2005) for a geometric algorithm for computing the solution to an MP-LP. Furthermore, a method that allow us to compute a continuous minimizer $X^* : \Theta \to \mathrm{R}^n$ such that $X^*(\theta) \in x^*(\theta)$ for all $\theta \in \Theta$ is given in Spjøtvol *et al.* (2005).

The following lemma deals with the special case of a multi-parametric program having as cost function an MMPS function (for a less restrictive version of this lemma see Lemma 5.1.2 in Necoara (2006)). Note that similar results were obtained in Kerrigan *et al.* (2002) for continuous PWA functions, but our proof is somewhat more intuitive and easier and moreover we obtain that $V^*(\theta)$ is also continuous and thus an MMPS function (a property that is crucial in Section 3.2).

LEMMA 2.7 *Let* $V : \mathrm{R}^n \times \mathrm{R}^{n_\theta} \to \mathrm{R},\ (x, \theta) \mapsto V(x, \theta)$ *be an MMPS function and consider the following multi-parametric optimization problem:*

$$\max\ V(x, \theta) \tag{28}$$

$$\textit{subject to} \quad Sx \le q + U\theta. \tag{29}$$

Assuming that for any parameter $\theta \in \Theta$ *the optimization problem* (28)–(29) *has a finite solution, then the solution of the multi-parametric optimization problem can be obtained by solving a set of MP-LPs. Moreover,* $V^*(\theta)$ *is an MMPS function.*

Proof Since $V(x, \theta)$ is an MMPS function and thus continuous we can use the max-min canonical representation of it. We have $V(x, \theta) = \max_{i \in \{1,\ldots,p\}} \min_{j \in \{1,\ldots,q\}} (\alpha_{ij}^T x + \beta_{ij}^T \theta + \gamma_{ij})$. Therefore, for each $i \in \{1, \ldots, p\}$ we must solve: $\max_x \min_{j \in \{1,\ldots,q\}} (\alpha_{ij}^T x + \beta_{ij}^T \theta + \gamma_{ij})$ subject to (29). The last multi-parametric program is in fact an MP-LP problem: $V_i^*(\theta) = \max_{x, \mu_i} \mu_i$ subject to $\alpha_{ij}^T x + \beta_{ij}^T \theta + \gamma_{ij} \ge \mu_i$, for $j = 1, \ldots, q$ and (29). From Proposition 2.6, we know that $V_i^*(\theta)$ is a *min* expression of affine terms in θ. In conclusion, we have to solve p MP-LPs and then $V^*(\theta) = \max_{i \in \{1,\ldots,p\}} V_i^*(\theta)$, i.e. V^* is an MMPS function. □

3 Open-loop MPC for uncertain MMPS or continuous PWA systems

3.1 *Uncertain MMPS or continuous PWA systems*

In this section we extend the MMPS (or equivalently the continuous PWA) deterministic model (1)–(2) or (3)–(4), to take also the disturbances and modeling errors into account. As in conventional linear systems, we model the disturbances and modeling errors by including an extra term in the system equations for MMPS systems. Hence, we consider the *uncertain MMPS* model:

$$x(k+1) = \mathcal{M}_x(x(k), u(k), e(k)) \tag{30}$$

$$y(k) = \mathcal{M}_y(x(k), u(k), e(k)), \tag{31}$$

where $\mathcal{M}_x$, $\mathcal{M}_y$ are vector-valued MMPS functions. The uncertainty caused by disturbances and modeling errors in the estimation of the real system is gathered in the uncertainty vector $e(k)$. We assume that this uncertainty is included in a bounded polyhedral set $\mathcal{E} = \{e \in \mathrm{R}^s : Se \le q\}$ and if consecutive samples $e(k)$,...,$e(k+j)$ are related (which is typically the case in the context of DES), we assume that this relation is linear (e.g. a system of linear equalities or inequalities).

Using the link between MMPS and continuous PWA systems, the uncertain MMPS system (30)–(31) can be also written as a *uncertain continuous PWA* system:

$$x(k+1) = \mathcal{P}_x(x(k), u(k), e(k)) \tag{32}$$

$$y(k) = \mathcal{P}_y(x(k), u(k), e(k)), \tag{33}$$

where $\mathcal{P}_x$ and $\mathcal{P}_y$ are continuous vector-valued PWA functions. Therefore, the algorithms derived in this paper can be also applied to uncertain continuous PWA systems. Note that in conventional uncertain PWA systems (Kerrigan *et al.*, 2002; Rakovic *et al.*, 2004; Johansson, 2003) the partition that generates the system is independent on the disturbance e. In our definition of an uncertain MMPS and uncertain *continuous* PWA system the partition will in general also depend on the disturbance (note that this is necessary to guarantee continuity of the system). Therefore, our modeling approach is more general than the one used in (Kerrigan *et al.*, 2002; Rakovic *et al.*, 2004; Johansson, 2003) where the partition is fixed.

We assume that at each step k of MPC, the state $x(k)$ is available[1] and we gather the uncertainty over the interval $[k, k+N_\mathrm{p}-1]$ in the vector $\tilde{e}(k) = [e^T(k), ..., e^T(k+N_\mathrm{p}-1)]^T \in \tilde{\mathcal{E}}$, where $\tilde{\mathcal{E}}$, according to our assumption, is a bounded polyhedral set. Then it is easy to see that the prediction $\hat{y}(k+j|k)$ of the future output for the system (30)–(31) can be written in MMPS form, for $j = 0, ..., N_\mathrm{p}-1$.

Using as cost criterion a combination of the output and input cost criterion as defined in (21) and (22):

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

and keeping in mind that all these cost criteria are MMPS expressions[2], we can write J as a max-min canonical expression:

$$J(\tilde{e}(k), \tilde{u}(k), x(k)) = \max_{j\in\{1,\dots,l\}} \min_{i\in S_j} (\bar{\alpha}_{i,j}^T x(k) + \bar{\beta}_{i,j}^T \tilde{u}(k) + \bar{\gamma}_{i,j}^T \tilde{e}(k) + \bar{\delta}_{i,j}). \tag{34}$$

Note that if the reference signal r depends on k then $\bar{\delta}_{i,j}$ will depend also on k (i.e. $\bar{\delta}_{i,j}$ are affine expressions in $\tilde{r}$).

3.2 *Worst-case MMPS-MPC*

In this section we study open-loop MPC for an uncertain MMPS system when $e(k)$ is a bounded uncertainty. We want to minimize an MMPS cost criterion $J(k) = J_\mathrm{out}(k) + \lambda J_\mathrm{in}(k)$ subject to some constraints. As we said before, we consider only linear constraints on the input, i.e. constraints of the form (20). The *worst-case MMPS-MPC problem* at step k is then defined as:

$$J^*(x(k)) = \min_{\tilde{u}(k)} \max_{\tilde{e}(k)\in\tilde{\mathcal{E}}} J(\tilde{e}(k), \tilde{u}(k), x(k)) \tag{35}$$

$$\text{subject to: } P(k)\tilde{u}(k) + q(k) \leq 0, \tag{36}$$

where $J(\cdot)$ is given by (34).

For a given $\tilde{u}(k), x(k)$ we define the *inner* worst-case MMPS-MPC problem

$$\max_{\tilde{e}(k)\in\tilde{\mathcal{E}}} J(\tilde{e}(k), \tilde{u}(k), x(k)). \tag{37}$$

[1] In the case of DES this assumption is not restrictive since the components of $x(k)$ correspond to event times and thus they are in general easy to measure.
[2] Recall that $|x| = \max(x, -x)$ for $x \in$ R.

We denote[3]

$$\tilde{e}^*(\tilde{u}(k), x(k)) = \arg \max_{\tilde{e}(k) \in \tilde{\mathcal{E}}} J(\tilde{e}(k), \tilde{u}(k), x(k)) \tag{38}$$

$$J^*(\tilde{u}(k), x(k)) = J(\tilde{e}^*(\tilde{u}(k), x(k)), \tilde{u}(k), x(k)). \tag{39}$$

PROPOSITION 3.1 *For a given $\tilde{u}(k)$ and $x(k)$, $\tilde{e}^*(\tilde{u}(k), x(k))$ given by (38) can be computed using a sequence of linear programming problems.*

Proof Because the uncertainty $e(k)$ is in a bounded polyhedral set $\mathcal{E}$ and if $e(k), \cdots, e(k+j)$ are related then this relation is linear we conclude that $\tilde{e}(k)$ will also be in a bounded polyhedral set: $\tilde{\mathcal{E}} = \{\tilde{e}(k) : \tilde{S}\tilde{e}(k) \leq \tilde{q}\}$. We determine for any fixed $[\tilde{u}^T(k)\ x^T(k)]^T$ the optimal $\tilde{e}^*(\tilde{u}(k), x(k))$, using the max-min canonical form (34) of $J(\cdot)$, by solving the following optimization problem:

$$\begin{aligned} &\max_{\tilde{e}(k)} \max_{j \in \{1,\dots,l\}} \min_{i \in S_j} (\bar{\alpha}_{i,j}^T x(k) + \bar{\beta}_{i,j}^T \tilde{u}(k) + \bar{\gamma}_{i,j}^T \tilde{e}(k) + \bar{\delta}_{i,j}) \\ &\text{subject to: } \tilde{S}\tilde{e}(k) \leq \tilde{q}, \end{aligned} \tag{40}$$

which is equivalent with:

$$\begin{aligned} &\max_{j \in \{1,\dots,l\}} \max_{\tilde{e}(k)} \min_{i \in S_j} (\bar{\alpha}_{i,j}^T x(k) + \bar{\beta}_{i,j}^T \tilde{u}(k) + \bar{\gamma}_{i,j}^T \tilde{e}(k) + \bar{\delta}_{i,j}) \\ &\text{subject to: } \tilde{S}\tilde{e}(k) \leq \tilde{q}. \end{aligned} \tag{41}$$

Now for each $j \in \{1, ..., l\}$ we have to solve the following optimization problem:

$$\begin{aligned} &\max_{\tilde{e}(k)} \min_{i \in S_j} (\bar{\alpha}_{i,j}^T x(k) + \bar{\beta}_{i,j}^T \tilde{u}(k) + \bar{\gamma}_{i,j}^T \tilde{e}(k) + \bar{\delta}_{i,j}) \\ &\text{subject to: } \tilde{S}\tilde{e}(k) \leq \tilde{q}, \end{aligned}$$

which is equivalent with the following linear programming problem:

$$\max_{\tilde{e}(k), t_{(j)}(k)} t_{(j)}(k) \tag{42}$$

subject to:

$$\begin{cases} t_{(j)}(k) \leq \bar{\alpha}_{i,j}^T x(k) + \bar{\beta}_{i,j}^T \tilde{u}(k) + \bar{\gamma}_{i,j}^T \tilde{e}(k) + \bar{\delta}_{i,j} & \text{for each } i \in S_j \\ \tilde{S}\tilde{e}(k) \leq \tilde{q}. \end{cases} \tag{43}$$

To obtain the solution of (40) we solve (42)–(43) for each $j \in \{1, ..., l\}$, with the optimal solution $[t_{(j)}^*(\tilde{u}(k), x(k))\ \ \tilde{e}_{(j)}^{*T}(\tilde{u}(k), x(k))]^T$ and then we select as $\tilde{e}^*(\tilde{u}(k), x(k))$, the optimal solution $\tilde{e}_{(j)}^*(\tilde{u}(k), x(k))$ for which $\min_{i \in S_j}(\bar{\alpha}_{i,j}^T x(k) + \bar{\beta}_{i,j}^T \tilde{u}(k) + \bar{\gamma}_{i,j}^T\ \tilde{e}_{(j)}^*(\tilde{u}(k), x(k)) + \bar{\delta}_{i,j})$ is the largest. □

Now, we define $U = \{\tilde{u}(k) : P(k)\tilde{u}(k) + q(k) \leq 0\}$ and we assume U to be bounded. Note that this assumption is not restrictive, because in practice the input $\tilde{u}(k)$ will always be bounded. Furthermore, the feasible set of the states X is also assumed to be a bounded polyhedron (since for physical systems operational range is usually known). For simplicity we assume that the multi-parametric program (37) has a finite optimal solution for any parameter $[\tilde{u}^T(k)\ x^T(k)] \in U \times X$.

PROPOSITION 3.2 *With the notations (38)–(39), $J^* : U \times X \to \mathrm{R}$ is an MMPS function and $\tilde{e}^* : U \times X \to \mathrm{R}^s$ is a PWA function.*

[3]Note that in general $\tilde{e}^*(\tilde{u}(k), x(k))$ may be set-valued, but as we will use Proposition 2.6, this is not an issue.

Proof For each $j \in \{1, ..., l\}$ we denote with $[t^*_{(j)}(\tilde{u}(k), x(k)) \;\; \tilde{e}^{*T}_{(j)}(\tilde{u}(k), x(k))]^T$ the optimal solution of the MP-LP problem (42)–(43), with the parameter $\theta = [\tilde{u}^T(k) \; x^T(k)]^T \in \Theta$ with $\Theta = U \times X$ a bounded polyhedral set. Using similar arguments as in Lemma 2.7 we know that $t^*_{(j)}(\cdot, \cdot)$ is a min expression of the form $t^*_{(j)}(x(k), \tilde{u}(k)) = \min_i(\bar{\mu}^T_{i,j} x(k) + \bar{\nu}^T_{i,j}\tilde{u}(k) + \bar{\xi}_{i,j})$ and

$$J^*(\tilde{u}(k),x(k)) = \max_{j \in \{1,...,l\}} (t^*_{(j)}(\tilde{u}(k), x(k))) = \max_{j \in \{1,...,l\}} \min_i (\bar{\mu}^T_{i,j} x(k) + \bar{\nu}^T_{i,j}\tilde{u}(k) + \bar{\xi}_{i,j}).$$

We thus obtain directly the max-min expression of $J^*(\cdot, \cdot)$.

Furthermore $\tilde{e}^*(\tilde{u}(k), x(k)) = \tilde{e}^*_{(j)}(\tilde{u}(k), x(k))$ if $t^*_{(j)}(\tilde{u}(k), x(k)) \geq t^*_{(i)}(\tilde{u}(k), x(k))$ for each $i \in \{1, ..., l\} \setminus \{j\}$. But each $\tilde{e}^*_{(j)}(\cdot, \cdot)$ is an MMPS function, and therefore a continuous PWA function. This implies that $\tilde{e}^*(\cdot, \cdot)$ is a PWA function. Note that $\tilde{e}^*(\cdot, \cdot)$ is not necessarily continuous. □

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

$$\min_{\tilde{u}(k)} J^*(\tilde{u}(k), x(k)) \tag{44}$$

$$\text{subject to } P(k)\tilde{u}(k) + q(k) \leq 0, \tag{45}$$

where we assume that at step k, the state $x(k)$ is given (or can be estimated).

PROPOSITION 3.3 *Given $x(k)$, the outer worst-case MMPS-MPC problem can be solved using a sequence of linear programming problems.*

Proof From Proposition 3.2 we know that $J^* : U \times X \to \mathrm{R}$ is an MMPS function. Therefore it can be written in the following min-max canonical form

$$J^*(\tilde{u}(k), x(k)) = \min_{j \in \{1,...,\hat{l}\}} \max_{i \in T_j} (\mu^T_{i,j} x(k) + \nu^T_{i,j}\tilde{u}(k) + \xi_{i,j}).$$

Then, the outer worst-case MMPS-MPC problem (44)–(45) can be written as

$$\min_{\tilde{u}(k)} \min_{j \in \{1,..,\hat{l}\}} \max_{i \in T_j} (\mu^T_{i,j} x(k) + \nu^T_{i,j}\tilde{u}(k) + \xi_{i,j})$$
$$\text{subject to: } P(k)\tilde{u}(k) + q(k) \leq 0.$$

For each $j \in \{1, ..., \hat{l}\}$ we must thus solve the following linear programming problem:

$$\min_{\tilde{u}(k), t_{(j)}} t_{(j)}$$
$$\text{subject to: } \begin{cases} t_{(j)} \geq \mu^T_{i,j} x(k) + \nu^T_{i,j}\tilde{u}(k) + \xi_{i,j}, \text{ for each } i \in T_j \\ P(k)\tilde{u}(k) + q(k) \leq 0. \end{cases} \tag{46}$$

In order to obtain the solution of (44)–(45), we solve (46), obtaining the optimal solution $[t^*_{(j)}(x(k)) \; \tilde{u}^{*T}_{(j)}(x(k))]^T$, for each $j \in \{1, ..., \hat{l}\}$ and then we select the optimal $\tilde{u}^*(x(k))$ as the optimal solution $\tilde{u}^*_{(j)}(x(k))$ for which $\max_{i \in T_j}(\mu^T_{i,j} x(k) + \nu^T_{i,j}\tilde{u}^*_{(j)}(x(k)) + \xi_{i,j})$ is the smallest. □

Based on the results discussed above we now present an algorithm to solve the worst-case MMPS-MPC problem.

Algorithm 1

Step 1: Compute the max-min expression of $J(\cdot)$. Solve *off-line* the inner worst-case MMPS-MPC problem (37) using MP-LP. According to Proposition 3.2 $J^*(x, u)$ is an MMPS function. Compute also off-line the min-max canonical form of this function.

Step 2: Compute *on-line* (at each step k) the solution of the outer worst-case MMPS-MPC problem (44)-(45) according to Proposition 3.3.

COROLLARY 3.4 *According to Algorithm 1, the outer worst-case MMPS-MPC problem can be solved using a sequence of linear programming problems. Moreover the associated controller is a PWA function of the argument* $x(k)$.

Proof In order to solve the problem (35)–(36), first we look for the worst-case uncertainty $\tilde{e}(k)$ as a function of $\tilde{u}(k), x(k)$ (Proposition 3.1) while in the second step of the algorithm we want to find the optimal input $\tilde{u}(k)$ corresponding to the worst-case uncertainty (Proposition 3.3). The first step is computed off-line. The second step can be solved using a sequence of linear programming problems according to Proposition 3.3.

For the second part of the corollary, we consider the MP-LP problem (46), with the parameter $x(k) \in X$ with X a polyhedral set. Then the optimal solution $[t^*_{(j)}(\cdot)\ u^{*T}_{(j)}(\cdot)]^T$ is an MMPS function of the argument $x(k)$ (according to Proposition 2.6). Therefore $\tilde{u}^*_{(j)}(\cdot)$ is a PWA function. But

$$\tilde{u}^*(k) = \tilde{u}^*_{(j)}(x(k)) \quad \text{if} \quad t^*_{(j)}(x(k)) \leq t^*_{(i)}(x(k))$$

for $i \in \{1, ..., \hat{l}\} \setminus \{j\}$. In conclusion, the worst-case MMPS-MPC controller $u^*(k)$ is a PWA function of the argument $x(k)$. □

Remark 1 It is clear from Proposition 3.2 that the outer worst-case MMPS-MPC problem can also be solved off-line, using again Lemma 2.7. Then, step 2 of Algorithm *1* consists in solving off-line the outer worst-case MMPS-MPC problem and then on-line at each step k we need only to evaluate a PWA function corresponding to the controller.

3.3 *Solution of the inner worst-case problem based on duality*

In Algorithm *1* we have to solve off-line the inner-worst case MMPS-MPC using MP-LP. In the case when the reference signal r is a non-zero sequence we have to include $r(k), ..., r(k + N_\mathrm{p} - 1)$ as additive parameters in the MP-LP program when we want to solve the inner-worst case problem off-line, using MP-LP, because the cost function depends also on r. Of course, the computational complexity increases in that case because the dimension of the vector of parameters ($[x(k)^T\ \tilde{u}(k)^T\ \tilde{r}(k)^T]^T$) is much larger than $\theta = [x(k)^T\ \tilde{u}(k)^T]^T$, corresponding to the case $r = 0$. An alternative method is to use the duality theory of linear programming (Pardalos *et al.*, 2002). For each $j \in \{1, \cdots, l\}$ the primal problem (42)-(43) can be written (for simplicity we drop the index k):

$$\text{(P):} \begin{cases} \max_{\tilde{e}, t_{(j)}} t_{(j)} \\ \text{subject to:} \begin{cases} t_{(j)} - \bar{\gamma}^T_{i,j}\tilde{e} \leq \bar{\alpha}^T_{i,j}x + \bar{\beta}^T_{i,j}\tilde{u} + \bar{\delta}_{i,j}, \ \text{for each } i \in S_j \\ \tilde{S}\tilde{e} \leq \tilde{q}. \end{cases} \end{cases}$$

We denote with $c_{i,j}(x, \tilde{u}) = \bar{\alpha}^T_{i,j}x + \bar{\beta}^T_{i,j}\tilde{u} + \bar{\delta}_{i,j}$, which is an affine expression in $(x, \tilde{u}) \in X \times U$, where $\bar{\delta}_{i,j}$ depends on $\tilde{r}$, which varies with k. In matrix notation the primal problem becomes:

$$\text{(P):} \begin{cases} \max_{\tilde{e}, t_{(j)}} t_{(j)} \\ \text{subject to:} \begin{bmatrix} 1 & -\bar{\gamma}^T_{i,j} \\ 0 & \tilde{S} \end{bmatrix} \begin{bmatrix} t_{(j)} \\ \tilde{e} \end{bmatrix} \leq \begin{bmatrix} c_{i,j}(x, \tilde{u}) \\ \tilde{q} \end{bmatrix}, \ \text{for each } i \in S_j. \end{cases}$$

Note that in primal problem (P) the variables $t_{(j)}, \tilde{e}$ are free. The dual problem then has the following form:

$$\text{(D)}: \begin{cases} \min_{y_j} [c_{1,j}(x,\tilde{u}), ..., c_{\#S_j,j}(x,\tilde{u}), \tilde{q}_1, ..., \tilde{q}_{n_{\tilde{S}}}]^T y_j \\ \text{subject to: } \begin{bmatrix} 1 & 0 \\ -\bar{\gamma}_{i,j} & \tilde{S}^T \end{bmatrix} y_j = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \text{ for each } i \in S_j \\ y_j \geq 0, \end{cases}$$

where $\#S_j$ denotes the cardinality of the set S_j and $n_{\tilde{S}}$ denotes the number of rows of the matrix $\tilde{S}$.

There are algorithms (e.g. the double description method of Motzkin *et al.* (1953)) to compute a compact explicit description of the elements of the polyhedral set:

$$K_j = \left\{ y_j \geq 0 : \begin{bmatrix} 1 & 0 \\ -\bar{\gamma}_{i,j} & \tilde{S}^T \end{bmatrix} y_j = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \text{ for each } i \in S_j \right\}.$$

These elements can be expressed as follows (according to *the finite basis theorem*):

$$y_j = \sum_{i=1}^{N_j} \alpha_{ij} y_j^i + \sum_{i=1}^{M_j} \beta_{ij} z_j^i,$$

with $\sum_i \alpha_{ij} = 1, \alpha_{ij} \geq 0$ and $\beta_{ij} \geq 0$. The y_j^i are called vertexes and the z_j^i are called extremal rays (using the definitions of Pardalos *et al.* (2002)). Because we assume that the primal problem (P) has a finite optimum, we are interested only in the vertexes (as extremal rays give rise to infinite solutions):

$$\{y_j^1, ..., y_j^{N_j}\}.$$

Note that the finite vertexes $y_j^1, ..., y_j^{N_j}$ do not depend on the reference signal $\tilde{r}(k)$, since $\tilde{r}(k)$ appears linearly in the $\bar{\delta}_{i,j}$'s which are present in the expressions of $c_{i,j}$ but not in the expression of the polyhedral set K_j. According to strong duality theorem for linear programming[1] we have:

$$t_{(j)}^*(x,\tilde{u}) = \min(c_j^T(x,\tilde{u}) y_j^1, ..., c_j^T(x,\tilde{u}) y_j^{N_j}), \quad (47)$$

where $c_j(x,\tilde{u}) = [c_{1,j}(x,\tilde{u}), ..., c_{\#S_j,j}(x,\tilde{u}), \tilde{q}_1, ..., \tilde{q}_{n_{\tilde{S}}}]^T$. Then,

$$J^*(x,\tilde{u}) = \max_{j \in \{1,\cdots,l\}} (t_{(j)}^*(x,\tilde{u})) = \max_{j \in \{1,\cdots,l\}} \min(c_j(x,\tilde{u}) y_j^1, ..., c_j(x,\tilde{u}) y_j^{N_j}). \quad (48)$$

Therefore we obtain directly the max-min canonical form of J^*. Algorithm *1* of the previous section can be applied also for this case (i.e. when $\tilde{r}$ is a non-zero sequence). Note that after we eliminate the redundant terms the max-min expression of $J^*(x(k), \tilde{u}(k))$ obtained applying duality coincides with the max-min expression of $J^*(x(k), \tilde{u}(k))$ obtained considering $\tilde{r}(k)$ as an extra parameter in the MP-LPs.

4 Disturbance feedback MPC for uncertain MMPS systems

It is well-known (Mayne *et al.*, 2000) that in the presence of uncertainties, the MPC controller performs better if we optimize over feedback policies in the worst-case optimization problem (35)–(36). So, another approach to controlling an uncertain MMPS system different from the ones presented in Section 3 is to include feedback by

[1] Computational geometry based on epigraph theory was used in Diehl *et al.* (2004) for solving MP-LPs.

searching over the set of affine functions of the past disturbances as it was done in (Lofberg, 2003; Goulart *et al.*, 2006) for linear systems. Therefore, we consider *disturbance feedback policies* of the form:

$$u(k+i) = \sum_{j=0}^{i-1} M_{i,j} e(k+j) + v(k+i),\ \forall i \in \{0, \cdots, N_{\mathrm{p}}-1\}, \tag{49}$$

where each $M_{i,j} \in \mathrm{R}^{m \times s}$ such that $M_{0,j} = 0$ and $v(k+i) \in \mathrm{R}^m$. Let us denote with $\tilde{u} = [u^T(k)\ u^T(k+1) \cdots u^T(k+N_{\mathrm{p}}-1)]^T$, $\tilde{v} = [v^T(k)\ v^T(k+1) \cdots v^T(k+N_{\mathrm{p}}-1)]^T$ and

$$\tilde{M} = \begin{bmatrix} 0 & 0 & \cdots & 0 \\ M_{1,0} & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ M_{N_{\mathrm{p}}-1,0} & M_{N_{\mathrm{p}}-1,1} & \cdots & 0 \end{bmatrix}, \tag{50}$$

so that the disturbance feedback policy becomes

$$\tilde{u} = \tilde{M}\tilde{e} + \tilde{v}. \tag{51}$$

Note that in contrast to optimal control approach Goulart *et al.* (2006) where one has to determine $e(k), \ldots, e(k+j-1)$ in order to compute the optimal control input $u(k+j)$, in the MPC framework, where we use a receding horizon approach, we do not need to know the value of the previous disturbance since at step k we apply the control input $u(k) = v(k)$.

Under this type of policy (51), the worst-case MMPS-MPC problem becomes:

$$J^*(x(k)) = \min_{\tilde{M},\tilde{v}} \max_{\tilde{e} \in \tilde{\mathcal{E}}} J(\tilde{e}, \tilde{M}\tilde{e} + \tilde{v}, x(k)) \tag{52}$$

$$\text{subject to: } P(k)(\tilde{M}\tilde{e} + \tilde{v}) + q(k) \leq 0,\ \forall \tilde{e} \in \tilde{\mathcal{E}}. \tag{53}$$

We also split the optimization problem (52)–(53) into two subproblems, as it was done in Section 3. The inner worst-case problem is formulated as:

$$J^*(\tilde{M}, \tilde{v}, x(k)) = \max_{\tilde{e}} \max_{j \in \{1,\ldots,l\}} \min_{i \in S_j} (\bar{\alpha}_{i,j}^T x(k) + (\bar{\beta}_{i,j}^T \tilde{M} + \bar{\gamma}_{i,j}^T)\tilde{e} + \bar{\beta}_{i,j}^T \tilde{v} + \bar{\delta}_{i,j})$$

$$\text{subject to: } \tilde{S}\tilde{e}(k) \leq \tilde{q}.$$

Using similar arguments as in Proposition 3.1, we conclude that for a given $(\tilde{M}, \tilde{v})$, $J^*(\tilde{M}, \tilde{v}, x(k))$ can be computed efficiently using a sequence of linear programming problems. Note that in this particular case we cannot obtain an explicit expression for $J^*(\tilde{M}, \tilde{v}, x(k))$ as in the open-loop case since the function $(M, e) \mapsto \beta^T M e$, for some fixed β, is neither convex nor concave.

The outer worst-case problem becomes:

$$\min_{\tilde{M},\tilde{v}} J^*(\tilde{M}, \tilde{v}, x(k)) \tag{54}$$

$$\text{subject to } P(k)(\tilde{M}\tilde{e} + \tilde{v}) + q(k) \leq 0,\ \forall \tilde{e} \in \tilde{\mathcal{E}}. \tag{55}$$

Note that the constraints (55) are nonlinear in $\tilde{M}$ and $\tilde{e}$. We can write the constraints as $P(k)\tilde{M}\tilde{e} \leq -P(k)\tilde{v} - q(k)$ for all $\tilde{e} \in \tilde{\mathcal{E}}$ or

$$[\max_{\tilde{e} \in \tilde{\mathcal{E}}} (P(k)\tilde{M})_1 \tilde{e} \cdots \max_{\tilde{e} \in \tilde{\mathcal{E}}} (P(k)\tilde{M})_{n_P} \tilde{e}]^T \leq -P(k)\tilde{v} - q(k),$$

where $(P(k)\tilde{M})_i$ denotes the i^{th} row of the matrix $P(k)\tilde{M}$. Therefore, using duality for linear programs and the fact that $\tilde{\mathcal{E}} = \{\tilde{e} : \tilde{S}\tilde{e} \leq \tilde{q}\}$ is a polytope and thus compact, it follows that the constraint (55) are equivalent with:

$$P(k)\tilde{M} = Z^T\tilde{S},\ Z^T\tilde{q} + P(k)v + q(k) \leq 0,\ Z \geq 0,$$

where by $Z \geq 0$ we mean $Z_{ij} \geq 0$ for all i, j. It follows that the outer worst-case problem can be written as:

$$\min_{\tilde{M},\tilde{v},Z} J^*(\tilde{M}, \tilde{v}, x(k))$$

$$\text{subject to: } P(k)\tilde{M} = Z^T\tilde{S},\ Z^T\tilde{q} + P(k)v + q(k) \leq 0, Z \geq 0.$$

Note that now the constraints are linear in $\tilde{M}$, $\tilde{v}$ and Z. The following algorithm provides a solution to the min-max disturbance feedback MPC approach formulated in this section:

Algorithm 2

Step 1: Compute the max-min expression of $J(\cdot)$

Step 2: Solve (52)–(53) using a standard nonlinear optimization algorithm for nonlinear optimization problems with linear constraints (e.g., a gradient projection algorithm Pardalos *et al.* (2002)[1]).

Note that in each iteration step ℓ of the algorithm for the outer problem the function values of J^* (and its gradient, which can be obtained using numerical approximation) have to be computed in the current iteration point (M_ℓ, v_ℓ). This involves solving the inner problem for the given M_ℓ and v_ℓ. This can be done efficiently by solving a sequence of linear programming problems as was shown before.

It is clear that in the particular case when $M_{i,j} = 0$ for all i, j, we obtain the open-loop controller derived in previous section.

5 Computational complexity

From a computational point of view, both approaches that we have derived before (the open-loop scheme and the disturbance feedback scheme) consist in two steps. In the first step we have to solve the maximization problem corresponding to the worst-case uncertainty. This can be done off-line solving a set of MP-LP problems as in Section 3.2 (or alternatively by computing the vertexes of some polyhedral set as in Section 3.3). In the second step we have to solve on-line a set of linear programming problems or to apply an iterative procedure based on solving a set of linear programming problems in order to determine the optimal MPC input. The main advantage of the second approach is that by introducing feedback, the corresponding MPC controller will perform better than the open-loop MPC controller. This improvement in performance is obtained at the expense of introducing $\frac{N_\mathrm{p}(N_\mathrm{p}-1)}{2}\, m\, s + n_P\, n_{\tilde{S}}$ extra variables and $n_P + n_{\tilde{S}}$ extra inequalities (recall that n_P and $n_{\tilde{S}}$ denote the number of rows of the matrices P and $\tilde{S}$, respectively). Note that the number of min terms in the max-min canonical form of the cost is the same in both approaches. See also Table 1 for a comparison of computational times for different methods applied to an example.

From Table 1 we see that in the case of open-loop min-max MPC, the CPU time corresponding to the dual approach (Section 3.3) is less than the CPU time corresponding to the MP-LP approach (Section 3.2). Theoretically, it is known (Borrelli *et al.*, 2003) that the number of partitions N_r generated by an MP-LP (i.e. (42)–(43)) is less than or equal to the number of vertexes ν corresponding to the polyhedron generated by the associated dual (i.e. K_j). The complexity of algorithms (Pardalos *et al.*, 2002; Motzkin *et al.*, 1953) for enumerating the vertexes of K_j with $n_0 = n_{\tilde{S}} + 1$ rows and n_1 columns is $\mathcal{O}(n_0^2 n_1 \nu)$. An upper bound on the number of vertexes is given by

[1] Note that sequential quadratic programming is less suited due to the PWA nature of the objective function.

(Goodman *et al.*, 1979):

$$N_{\mathrm{r}} \leq \nu \leq \binom{n_0 + n_1 - \lfloor n_1/2 \rfloor}{\lfloor n_1/2 \rfloor} + \binom{n_0 + n_1 - 1 - \lfloor (n_1 - 1)/2 \rfloor}{\lfloor (n_1 - 1)/2 \rfloor},$$

where $\lfloor x \rfloor$ is the largest integer less or equal to x and $\binom{m}{n} = \frac{m!}{n!(m-n)!}$. This means that in the worst-case the number of vertexes ν can be of the order $\mathcal{O}((n_0 + n_1)^{\lfloor n_1/2 \rfloor})$ if $n_0 + n_1 \gg n_1$. Of course, we need extra computations in order to compute the optimal value in the case of MP-LP approach. Since the execution time of an MP-LP algorithm depends on many factors it is difficult to give a net characterization of the computational complexity as a function of the number of variables, parameters and inequalities. But, after elimination of the redundant terms both approaches produces the same number of affine expressions, i.e. we get the same MMPS function for $J^*(\tilde{u}(k), x(k))$. Moreover, when $\tilde{r}$ is not a constant vector, the dimension of the vector of parameters $\theta = [x^T(k)\ \tilde{u}^T(k)\ \tilde{r}^T(k)]^T$ is large, which makes the computation of an MP-LP solution difficult.

The worst-case complexity of the approaches presented above is largely determined by the number of linear terms in the equivalent max-min canonical forms. In the worst case scenario this number increases rapidly as the prediction horizon, the number of states of the MMPS systems, or the number of min-max nestings in the state equations or the objective function increases. However, although the number of terms in the full max-min canonical expression may be very large, it can sometimes be reduced significantly (in De Schutter *et al.* (2004) the authors provide an example where the full canonical form contains 216 max-terms, of which only 4 are necessary). Although to the authors' best knowledge there are currently not yet any efficient algorithms for the simplification and reduction to a minimal canonical form (i.e., the canonical form with the minimal number of terms), some ad-hoc methods can be used (Heidergott *et al.*, 2005; De Schutter *et al.*, 2004) to reduce the number of min-terms significantly. Furthermore, the complexity of the reduction process can also be reduced by already eliminating redundant terms during the intermediate steps of the transformations. In conclusion, although the reduction to canonical form is computationally intensive, it can be done off-line (for both the inner and the outer worst-case MMPS-MPC problems).

If we consider reference tracking (the reference signal $r \neq 0$) or if consecutive disturbances are related, using dynamic programming approach (Kerrigan *et al.*, 2002) we must include $\tilde{r}$ or $\tilde{e}$ as parameters in the multi-parametric program, which increases the computational complexity. Note that these issues can be easily handled with our approaches (open-loop or disturbance feedback MPC). From the above we can conclude however that also our algorithms (Algorithm 1 and 2) are not well suited for large problems with many states, inputs and inequalities. This is not surprising since the computation of optimal control laws for PWA systems reduces to mixed-integer linear/quadratic optimization problems, which are difficult to solve (Bemporad *et al.*, 1999).

6 Example

6.1 *Set-up and the model of the plant*

In this section we present an example for which we apply the above method. Consider a room with a basic heat source and an additional controlled heat source (see Figure 1). Let u be the contribution to the increase in room temperature per time unit caused by the controlled heat source (so $u \geq 0$). For the basic heat source, this value is assumed to be constant and equal to 1. The temperature in the room is assumed to be uniform and obeys the first-order differential equation

$$\dot{T}(t) = \alpha(T(t))T(t) + u(t) + 1 + e_1(t)\ ,$$

the disturbance being gathered in the scalar variable e_1. We assume that the temperature coefficient has the following piecewise constant form:

$$\alpha(T) = \begin{cases} -1/2 & \text{if} \quad T < 0 \\ -1 & \text{if} \quad T \geq 0. \end{cases}$$

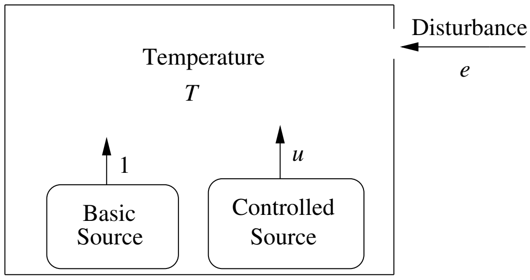

Figure 1. Temperature in a room.

We assume that the temperature is measured, but the measurement is noisy: $y(t) = T(t) + e_2(t)$. Using the Euler discretization scheme, with a sample time of 1 time unit and denoting the state $x(k) = T(k)$, we get the following continuous discrete-time PWA system:

$$x(k+1) = \begin{cases} 1/2x(k) + u(k) + e_1(k) + 1 & \text{if } x(k) < 0 \\ u(k) + e_1(k) + 1 & \text{if } x(k) \geq 0 \end{cases} \tag{56}$$

$$y(k) = x(k) + e_2(k). \tag{57}$$

Assume that we have $-2 \leq e_1(k), e_2(k) \leq 2$, $e_1(k) + e_2(k) \leq 1$, i.e. the uncertainty is given by the bounded polyhedron

$$\mathcal{E} = \left\{ [e_1 \; e_2]^T : -2 \leq e_1(k), e_2(k) \leq 2, \; e_1(k) + e_2(k) \leq 1 \right\}.$$

The equivalent MMPS representation of (56)–(57) is the following:

$$x(k+1) = \min(1/2x(k) + u(k) + e_1(k) + 1, u(k) + e_1(k) + 1), \tag{58}$$

$$y(k) = x(k) + e_2(k). \tag{59}$$

6.2 *MPC and simulations*

Because at sample step k the input $u(k)$ has no influence on $y(k)$, we take $N_\mathrm{p} = 3, N_\mathrm{c} = 2$, $\tilde{y}(k) = [\hat{y}(k+1|k) \; \hat{y}(k+2|k)]^T$, $\tilde{r}(k) = [r(k+1) \; r(k+2)]^T$, $\tilde{u}(k) = [u(k) \; u(k+1)]^T$. Let the uncertainty vector $e(k)$ be $e(k) = [e_1(k) \; e_2(k+1)]^T$. Therefore, $\tilde{e}(k) = [e^T(k) \; e^T(k+1)]^T$. We consider the following constraints on the input[1]:

$$-4 \leq \Delta u(k) = u(k+1) - u(k) \leq 4 \quad \text{and} \quad u(k) \geq 0 \;\; \text{for all } k.$$

As cost criterion we take

$$J(k) = J_{\mathrm{out},\infty}(k) + \lambda J_{\mathrm{in},1}(k) = \|\tilde{y}(k) - \tilde{r}(k)\|_\infty + \lambda \|\tilde{u}(k)\|_1. \tag{60}$$

The first term of $J(k)$ expresses the fact that we penalize the maximum difference between the reference and the output signal, while the second term penalizes the absolute value of the control effort. Because $u(k) \geq 0$, we have $\|u(k)\|_1 = u(k)$ and therefore we get the following max-min expression for $J(k)$:

$$J(k) = \max\{\min\{t_1, t_2\}, t_3, t_4, \min\{t_5, t_6, t_7\}, t_8, t_9, t_{10}\},$$

[1] Because we have only heating, a physical constraint on input is $u(k) \geq 0$. Furthermore we assume that the rate of heating is bounded.

	off-line			on-line		
N_{p}	2	3	4	2	3	4
Nr. of MP-LPs/LPs	7	12	18	4	8	16
Time MP-LP saved (s)	10.2	45	130	0.12	0.35	0.79
Time Dual (s)	0.35	0.9	2	0.06	0.08	0.1
Time Dist. feedback (s)	0.65	2.3	7.5	0.09	0.3	0.95
Time MP-LP ref. (s)	2	6.8	32	0.06	0.08	0.1

Table 1. The CPU time for $N_{\mathrm{p}} \in \{2,\ 3,\ 4\}$ with different methods: MP-LP saved=computes off-line and stores the controller for different values of r; Dual=computes off-line the controller based on Section 3.3; Dist. feedback=computes the controller based on Section 4; MP-LP ref= the reference signal is considered as an extra parameter (Kerrigan *et al.*, 2002).

where t_j are appropriately defined affine functions of $x(k), u(k), u(k+1), e(k),\ e(k+1), r(k+1), r(k+2)$. We compute now the closed-loop MPC controller over a simulation period $[1, 20]$, with $\lambda = 0.1$, initial state $x(0) = -6,\ u(-1) = 0$ and the reference signal $\{r(k)\}_{k=1}^{20} = -5, -5, -5, -5, -5, -3, -3, 1, 3, 3, 8, 8, 8, 8, 10$, 10, 10, 7, 7, 7, 4, 3, 1, 1, 6, 7, 8, 9, 11, 11 using the methods given in Sections 3 and 4.

After we compute off-line the max-min canonical form of $J^*(x, \cdot)$ and after elimination of the redundant terms we obtain a min-max canonical form of $J^*(x(k), \cdot)$ that gives rise to only 4 LPs that must be solved on-line at each sample step k in the open-loop approach.

In Table 1 we provide the CPU time[1] for different steps of the algorithms and for different methods, where the values for the prediction horizon N_{p} are $2,\ 3$ and 4. Note that the number of MP-LP or LP problems increases with N_{p} (see the third row). Note that in this example the computational time for the approach from Section 3.3 is less than the computational time for the MP-LP approach from Section 3.2. Since the reference signal is not constant, we have to include $\tilde{r}$ as an extra parameter when we apply the approach of (Kerrigan *et al.*, 2002), which results in a large CPU time.

In Figure 2, the top plot represents the reference signal (dashed line) and the output of disturbance feedback approach (full line) and the open-loop approach (star line). We see that the MPC controller obtained using disturbance feedback policies performs the tracking better than the open-loop MPC controller. In the second plot we show the optimal input: we can see that always $u(k) \geq 0$. The third plot shows the absolute value of the tracking error. Note that the error from the open-loop approach is substantially above the error from disturbance feedback approach. Finally, we plot $\Delta u^*(k) = u^*(k+1) - u^*(k)$ and the vector of uncertainty. We can see that also the constraint $|u^*(k+1) - u^*(k)| \leq 4$ is fulfilled, and that at some moments this constraint is indeed active.

7 Conclusions and future research

In this paper we have extended the MPC framework for MMPS (or equivalently for continuous PWA) systems to include also bounded disturbances. This allowed us to design a worst-case MMPS-MPC controller for such systems based on optimization over open-loop input sequences and disturbance feedback policies. We have shown that the resulting optimization problems can be computed efficiently using a two-step optimization approach that involves basically to solve a sequence of linear programming problems. In the first step we have to solve off-line an MP-LP (or alternatively, we can compute the vertexes of some polyhedral set) and next we have to write the min-max expression of the worst-case performance criterion. In the second step we solve only a sequence of linear programming problems in both approaches. As we expected and was also illustrated in an example, the disturbance feedback based MPC controller performs better that the open-loop MPC controller, at the expense of introducing some extra variables.

For future research we want to investigate stability for uncertain max-min-plus-scaling systems by building upon the results already obtained for uncertain max-plus-linear systems (see e.g. Necoara (2006)). Moreover, we want to improve the computational efficiency of the proposed algorithms and to extend them to cope efficiently with state constraints using results from parallel processing and distributed optimization.

[1] On a 1.5 GHz Pentium 4 PC with 512 MB RAM.

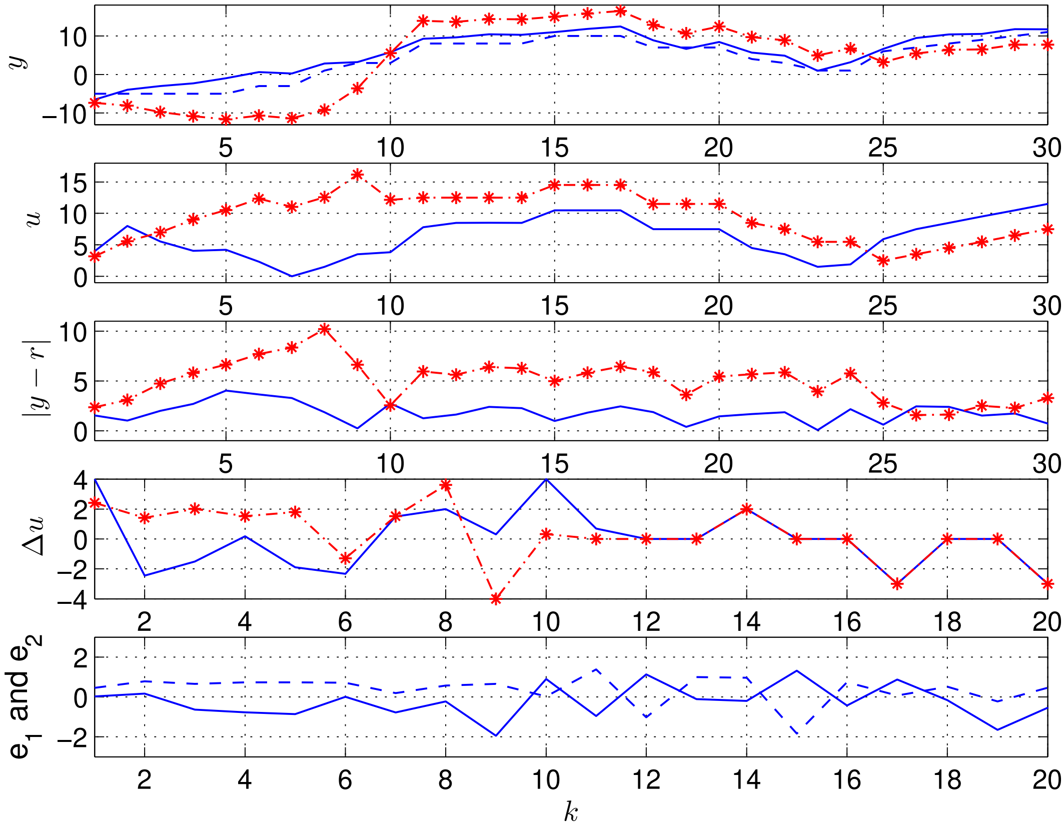

Figure 2. Illustration of the worst-case MPC for a uncertain MMPS system: full line: disturbance feedback approach, star line: open-loop approach, dashed line: reference signal r.

Acknowledgments

Research supported by the STW projects "Model predictive control for hybrid systems" (DMR. 5675), and "Multi-agent control of large-scale hybrid systems" (DWV. 6188), and by the European IST project "Modelling, Simulation and Control of Nonsmooth Dynamical Systems (SICONOS)" (IST-2001-37172) and by the European 6th Framework Network of Excellence "HYbrid CONtrol: Taming Heterogeneity and Complexity of Networked Embedded Systems (HYCON)".

References

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

A. Bemporad, M. Morari, Control of systems integrating logic, dynamics, and constraints, Automatica 35 (3) (1999) 407–427.

F. Borrelli, A. Bemporad, M. Morari, A geometric algorithm for multi-parametric linear programming, Journal of Optimization Theory and Applications 118 (3) (2003) 515–540.

C. Cassandras, D. Pepyne, Y. Wardi, Optimal control of a class of hybrid systems, IEEE Transactions on Automatic Control 46 (3) (2001) 398–415.

B. Cottenceau, L. Hardouin, J. Boimond, J. Ferrier, Model reference control for timed event graphs in dioid, Automatica 37 (8) (2001) 1451–1458.

R. Cuninghame-Green, Minimax Algebra, Vol. 166 of Lecture Notes in Economics and Mathematical Systems, Springer-Verlag, Berlin, Germany, 1979.

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

B. De Schutter, T. van den Boom, MPC for continuous piecewise-affine systems, Systems and Control Letters 52 (2004) 179–192.

M. Diehl, J. Björnberg, Robust dynamic programming for min-max model predictive control of constrained uncertain systems, IEEE Transactions on Automatic Control 49 (12) (2004) 2253–2257.

T. Gal, Postoptimal Analyses, Parametric Programming, and Related Topics, de Gruyter, Berlin, 1995.

J. Goodman, J. O'Rourke, Handbook of Discrete and Computational Geometry, Discrete Mathematics and Its Applications, Boca Raton, 1997.

P. Goulart, E. Kerrigan, J. Maciejowski, Optimization over state feedback policies for robust control with constraints, Automatica 42 (4) (2006) 523–533.

W. Heemels, B. De Schutter, A. Bemporad, Equivalence of hybrid dynamical models, Automatica 37 (7) (2001) 1085–1091.

B. Heidergott, G. Olsder, J. Woude, Max Plus at Work, Princeton University Press, Princeton, 2005.

M. Johansson, Piecewise Linear Control Systems, Vol. 284 of Lecture Notes in Control and Information Sciences, Springer, Berlin, Germany, 2003.

E. Kerrigan, D. Mayne, Optimal control of constrained, piecewise affine systems with bounded disturbances, in: Proceedings of the 2002 IEEE Conference on Decision and Control, Las Vegas, Nevada, 2002.

M. Kvasnica, P. Grieder, M. Baotić, Multi-Parametric Toolbox (MPT), http://control.ee.ethz.ch/ mpt/ (2005).

J. Lofberg, Approximations of closed-loop MPC, in: Proceedings of the 42th IEEE Conference on Decision and Control, Maui, Hawaii, USA, 2003, pp. 1438–1442.

J. Maciejowski, Predictive Control with Constraints, Prentice Hall, Harlow, England, 2002.

D. Mayne, J. Rawlings, C. Rao, P. Scokaert, Constrained model predictive control: Stability and optimality, Automatica 36 (2000) 789–814.

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

T. Motzkin, H. Raiffa, G. Thompson, R. Thrall, The double description method, in: H. Kuhn, A. Tucker (Eds.), Contributions to the Theory of Games, no. 28 in *Annals of Mathematics Studies*, Princeton University Press, Princeton, New Jersey, 1953, pp. 51–73.

I. Necoara, Model Predictive Control for Max-Plus-Linear and Piecewise Affine Systems, *PhD thesis*, Delft Center for Systems and Control, Delft University of Technology, The Netherlands, 2006.

I. Necoara, E. Kerrigan, B. De Schutter, T. van den Boom, Finite-horizon min-max control of max-plus-linear systems, Tech. Rep. 06-002, Delft Center for Systems and Control, Delft University of Technology, The Netherlands, (Provisionally accepted for IEEE Transactions on Automatic Control) 2005.

I. Necoara, B. De Schutter, T. van den Boom, J. Hellendoorn, Model predictive control for perturbed continuous piecewise affine systems with bounded disturbances, in: Proceedings of the 43rd IEEE Conference on Decision and Control, Paradise Island, The Bahamas, 2004, pp. 1848–1853.

G. Olsder, On structural properties of min-max systems, in: G. Cohen, J. Quadrat (Eds.), Proceedings of the 11th International Conference on Analysis and Optimization of Systems (Sophia-Antipolis, France, Jun. 1994), Vol. 199 of Lecture Notes in Control and Information Sciences, London: Springer-Verlag, 1994, pp. 237–246.

P. Pardalos, M. Resende, Handbook of Applied Optimization, Oxford University Press, Oxford, UK, 2002.

S. V. Rakovic, E. C. Kerrigan, D. Q. Mayne, Optimal control of constrained piecewise affine systems with state- and input-dependent disturbances, in: Proceedings 16th International Symposium on Mathematical Theory of Networks and Systems, Katholieke Universiteit Leuven, Belgium, 2004.

J. Spjøtvold, P. Tøndel, T. A. Johansen, Unique polyhedral representations to continuous selections for convex multi-parametric quadratic programs, in: Proceedings 2005 American Control Conference, Portland, Oregon USA, 2005.

R. van Egmond, G. Olsder, H. van Zuylen, A new way to optimise network traffic control using maxplus algebra, in: Proceedings of the 78th Annual Meeting of the Transportation Research Board, Washington, DC, 1999, paper 660.