Available online at www.sciencedirect.com
SCIENCE DIRECT®
Systems & Control Letters 55 (2006) 266–274
SYSTEMS & CONTROL LETTERS
www.elsevier.com/locate/sysconle

Explicit solution of min–max MPC with additive uncertainties and quadratic criterion

D. Muñoz de la Peña*, D.R. Ramírez, E.F. Camacho, T. Alamo

Dep. Ingeniería de Sistemas y Automática, Universidad de Sevilla, Escuela Superior de Ingenieros, Camino de los Descubrimientos s/n, 41092 Sevilla, Spain

Received 23 June 2003; received in revised form 29 July 2005; accepted 11 August 2005
Available online 5 October 2005

Abstract

Min–max model predictive control (MMMPC) is one of the strategies proposed to control plants subject to bounded uncertainties. This technique is very difficult to implement in real time because of the computation time required. Recently, the piecewise affine nature of this control law has been proved for unconstrained linear systems with quadratic performance criterion. However, no algorithm to compute the explicit form of the control law was given. This paper shows how to obtain this explicit form by means of a constructive algorithm. An approximation to MMMPC in the presence of constraints is presented based on this algorithm.

Keywords: Predictive control; Robust control; Uncertain systems

1. Introduction

Model predictive control (MPC) is one of the few control techniques able to cope with model uncertainties in an explicit way [7]. One approach used in MPC when uncertainties are present, is to minimize the objective function for the worst possible case. This strategy is known as min–max model predictive control (MMMPC) and was originally proposed in [27] in the context of robust receding horizon control and in [8] in the context of robust MPC. All MMMPC techniques for constrained and unconstrained linear uncertain systems have a great computational burden in common (see [14,21,25]) which limits the range of processes to which they can be applied. Few applications can be found in literature even for slow dynamics or complex simulated models (see [11,18]). In order to overcome the computational burden, several works have been proposed in the literature (see for example [12,13,23]). Even though, the implementation of robust MPC on real systems remain an open question.

It was shown in [5] that constrained MPC could be solved using multiparametric linear or quadratic programming (depending on the objective function). In this way an easily implemented explicit solution can be obtained. These types of results were extended to min–max controllers for linear uncertain systems with l_1 or l_∞ norms in [4,10]. The piecewise affine nature for quadratic cost functions has also been proved by other means in [19,20]. However, these works do not include an algorithm to obtain the explicit form of the control law.

This paper presents an algorithm that computes the explicit form of an unconstrained MMMPC controller with a quadratic cost function. The range of processes to which, in practice, these controllers can be applied is thus considerably broadened. Moreover, the constrained formulation is taken into account in the paper. An approximated min–max controller based on the explicit solution of the unconstrained formulation is presented. This controller minimizes an upper bound of the cost function and the optimization problem to solve is a quadratic programming problem.

The paper is organized as follows: Section 2 introduces the controller and its related optimization problem. Some properties of the min–max problem are shown in Section 3. The characterization of the regions in which the state space

* Corresponding author. Tel.: +34 95 448 7343; fax: +34 95 448 7340.
E-mail addresses: davidmps@cartuja.us.es (D. Muñoz de la Peña), danirr@cartuja.us.es (D.R. Ramírez), eduardo@cartuja.us.es (E.F. Camacho), alamo@cartuja.us.es (T. Alamo).

0167-6911/$ - see front matter
doi:10.1016/j.sysconle.2005.08.006

can be partitioned is presented in Section 4. In Section 5 the algorithm for exploring the state space and computing the explicit controller is presented. In Section 6 constraint handling is addressed. Section 7 illustrates the results presented in the paper by means of some simulated examples. Finally, we present concluding remarks in Section 8.

2. Min–max MPC with additive bounded uncertainties

Consider the discrete invariant time linear system with bounded uncertainties

$$x_{k+1} = Ax_k + Bu_k + Dw_k, \tag{1}$$

where $x_k \in \mathbb{R}^{n_x}$ is the state, $u_k \in \mathbb{R}^{n_u}$ is the control input, and $w_k \in \mathbb{R}^{n_w}$ is the uncertainty. The uncertainty is supposed to be bounded; i.e. $\|w_k\|_\infty \leqslant \varepsilon$.

Open loop min–max MPC obtains a single control input sequence that minimizes the worst case cost (see [8,17,26]) in which the predictions are computed in an open-loop manner (although the resulting controller is a feedback controller). These controllers are based on the solution of a single min–max problem optimizing a single control sequence for all possible values of the uncertainty. This formulation is known to be conservative because it underestimates the set of feasible input trajectories [21]. One solution proposed in the literature is to minimize a sequence of control corrections efforts to a given linear feedback stabilizing control law for the nominal plant. In this way, some kind of feedback is introduced in the prediction without increasing the computational effort (see [3,15]). The control input is given by $u_k = Kx_k + v_k$, where K is chosen in order to achieve some desired property for the non-constrained problem such as stability or LQR optimality. The MPC controller will compute the optimal sequence of correction control inputs v_k. The dynamics of the system can be rewritten as

$$x_{k+1} = A_K x_k + Bv_k + Dw_k,$$

where $A_K = (A + BK)$.

Consider a sequence $\mathbf{v} = \{v_0, v_1, \ldots, v_{N-1}\}$ of correction control inputs and $\mathbf{w} = \{w_0, w_1, \ldots, w_{N-1}\}$ a possible sequence of input disturbances to the system over a prediction horizon N. The objective function is defined as a quadratic performance index of the form

$$J(\mathbf{v}, \mathbf{w}, x) = \sum_{j=0}^{N-1} [x_j^{\mathrm{T}} Q x_j + u_j^{\mathrm{T}} R u_j] + x_N^{\mathrm{T}} P x_N,$$

where x_j and u_j are the predicted state and input of time j taking into account the uncertainty $\mathbf{w}$. The initial state is $x_0 = x$. Weighting matrices $Q = Q^{\mathrm{T}} \geqslant 0$ and $P = P^{\mathrm{T}} \geqslant 0$ are positive semi-definite, and $R = R^{\mathrm{T}} > 0$ is positive definite.

Taking into account (1), variables x_j and u_j are given by linear functions of x, $\mathbf{v}$ and $\mathbf{w}$, namely

$$\begin{aligned} x_j &= A_K^j x + \sum_{i=1}^{j} A_K^{i-1} B v_{j-i} + \sum_{i=1}^{j} A_K^{i-1} D w_{j-i}, \\ u_j &= K x_j + v_j. \end{aligned} \tag{2}$$

Min–max MPC [8] is based on finding the control sequence $\mathbf{v}$ that minimizes $J(\mathbf{v}, \mathbf{w}, x)$ for the worst possible case of the predicted future evolution of the process state or output signal. This is accomplished by the solution of a min–max problem denoted $P(x)$

$$J^*(x) = \min_{\mathbf{v}} \max_{\mathbf{w} \in W_N} J(\mathbf{v}, \mathbf{w}, x), \tag{3}$$

where W_N denotes the set of possible disturbance sequences of length N:

$$W_N = \{\mathbf{w} | \ \|w_i\|_\infty \leqslant \varepsilon, \ i = 0, \ldots, N-1\}.$$

This optimization problem is solved at each sampling time and the solution $\mathbf{v}^*(x)$ is applied using the well known receding horizon approach [7]; i.e., only the first component of $\mathbf{v}^*(x)$ is used and the control input applied to the system is given by $u_0 = Kx + v_0^* = K_{\mathrm{MPC}}(x)$.

Taking into account (2), matrices H_x, H_u and H_w can be found (see [5,7]) in such a way that

$$J(\mathbf{v}, \mathbf{w}, x) = \|H_x x + H_u \mathbf{v} + H_w \mathbf{w}\|_2^2. \tag{4}$$

The cost function is a convex quadratic function on $\mathbf{v}$, x and $\mathbf{w}$ because it is the square of the Euclidean norm of a vector which depends linearly on these parameters (see [2]).

Function $J(\mathbf{v}, \mathbf{w}, x)$ is convex in $\mathbf{w}$, thus the maximum will be attained at least at one of the vertices $\mathbf{w}_i$ of the polyhedron W_N (see [2, Theorem 3.4.6]). The maximizer is not unique and the maximum can also be attained at another vector $\mathbf{w} \notin \mathrm{ver}(W_N)$, where $\mathrm{ver}(W_N)$ is the set of vertices of W_N. However, the maximum is unique and that is what is needed to solve the inner maximization problem (the maximizer is indeed irrelevant). The maximum of $J(\mathbf{v}, \mathbf{w}, x)$ can therefore be obtained evaluating the cost function at the set of vertices of the hypercube W_N. The min–max problem can be rewritten as

$$J^*(x) = \min_{\mathbf{v}} \max_{\mathbf{w} \in W_N} J(\mathbf{v}, \mathbf{w}, x) = \min_{\mathbf{v}} J_{\max}(\mathbf{v}, x),$$

with

$$J_{\max}(\mathbf{v}, x) = \max_{\mathbf{w} \in W_N} J(\mathbf{v}, \mathbf{w}, x) = \max_{\mathbf{w}_i \in \mathrm{ver}(W_N)} J(\mathbf{v}, \mathbf{w}_i, x). \tag{5}$$

Function $J(\mathbf{v}, \mathbf{w}, x)$ is convex on $\mathbf{v}$, and as $R > 0$, it holds that $H_u^{\mathrm{T}} H_u > 0$. Thus $J(\mathbf{v}, \mathbf{w}, x)$ is indeed strictly convex on $\mathbf{v}$. On the other hand, $J_{\max}(\mathbf{v}, x)$ is the pointwise maximum of a set of strictly convex functions of $\mathbf{v}$. Therefore $J_{\max}(\mathbf{v}, x)$ is also strictly convex on $\mathbf{v}$ [6]. This implies that the solution to the min–max problem is unique because strictly convex functions have an unique minimizer (see [2, Theorem 3.4.2]).

3. Some properties of the min–max problem

In this section some properties and definitions are introduced.

Definition 1. For a given state vector x, the set of active vertices is the subset $I = \{\mathbf{w}_i\}$ at which the maximum is attained for the solution of the min–max problem, i.e. $J(\mathbf{v}^*(x), \mathbf{w}_i, x) = J^*(x)$.

The following definition introduces the concept of critical region of a set of vertices.

Definition 2. The critical region CR_I corresponding to the set of vertices $I = \{\mathbf{w}_i\}$ is the region of the state space where this set is the set of active vertices.

The set of active vertices for a given x characterizes the solution of (3). Its definition implies that

$$J(\mathbf{v}^*(x), \mathbf{w}_i, x) > J(\mathbf{v}^*(x), \mathbf{w}_j, x), \quad \mathbf{w}_i \in I, \quad \mathbf{w}_j \notin I.$$

As illustrated in Fig. 1, the solution of (3) for x is the same as that of

$$J^*(x) = \min_{\mathbf{v}} \max_{\mathbf{w}_i \in I} J(\mathbf{v}, \mathbf{w}_i, x) \quad \forall x \in \mathrm{CR}_I. \tag{6}$$

Moreover, it can be associated to each vertices set a minimization problem whose solution is an affine function of x as it is shown in the following.

Definition 3. Given a set of vertices $I = \{\mathbf{w}_i\}$, the candidate solution $\mathbf{v}_I(x)$ is the optimizer of

$$\begin{aligned} J_I(x) = \; & \min_{\mathbf{v}} J(\mathbf{v}, \mathbf{w}_i, x) \\ & \text{s.t. } J(\mathbf{v}, \mathbf{w}_i, x) = J(\mathbf{v}, \mathbf{w}_j, x), \ \mathbf{w}_i \in I \ \ \forall \mathbf{w}_j \in I. \end{aligned} \tag{7}$$

Proposition 1. *The optimizer $\mathbf{v}_I(x)$ of (7) is an affine function of x.*

$$\mathbf{v}_I(x) = K_I x + q_I. \tag{8}$$

Proof. From (4) it is inferred that (7) is equivalent to a quadratic program with linear equality constraints, namely

$$\begin{aligned} \mathbf{v}_I(x) = \; & \arg\min_{\mathbf{v}} \|H_x x + H_u \mathbf{v} + H_w \mathbf{w}_i\|_2^2 \\ & \text{s.t. } (\mathbf{w}_i^{\mathrm{T}} - \mathbf{w}_j^{\mathrm{T}}) H_w^{\mathrm{T}} (H_x x + H_u \mathbf{v}) \\ & \qquad = \mathbf{w}_j^{\mathrm{T}} H_w^{\mathrm{T}} H_w \mathbf{w}_j - \mathbf{w}_i^{\mathrm{T}} H_w^{\mathrm{T}} H_w \mathbf{w}_i, \\ & \qquad \mathbf{w}_i \in I \ \ \forall \mathbf{w}_j \in I. \end{aligned}$$

It is well known that the solution to this problem is an affine function of x. □

In the following, based on the results presented in [19,20], it will be shown that the solution of the min–max problem is an affine function of x.

Proposition 2. *For all $x \in \mathrm{CR}_I$, the solution to the min–max problem $\mathbf{v}^*(x)$ is an affine expression of x equal to $\mathbf{v}_I(x)$.*

$$\forall x \in \mathrm{CR}_I, \quad \mathbf{v}^*(x) = \mathbf{v}_I(x) = K_I x + q_I.$$

Proof. Let $x \in \mathrm{CR}_I$, $\mathbf{v}^*(x)$ be the solution of (3) and $\mathbf{v}_I(x)$ computed as in (7). As $\mathbf{v}^*(x)$ is the solution of (3) and the set of active vertices is I it holds that

$$J(\mathbf{v}^*(x), \mathbf{w}_i, x) = J(\mathbf{v}^*(x), \mathbf{w}_j, x)$$

for all $\mathbf{w}_i, \mathbf{w}_j \in I$, so $\mathbf{v}^*(x)$ is feasible for (7).

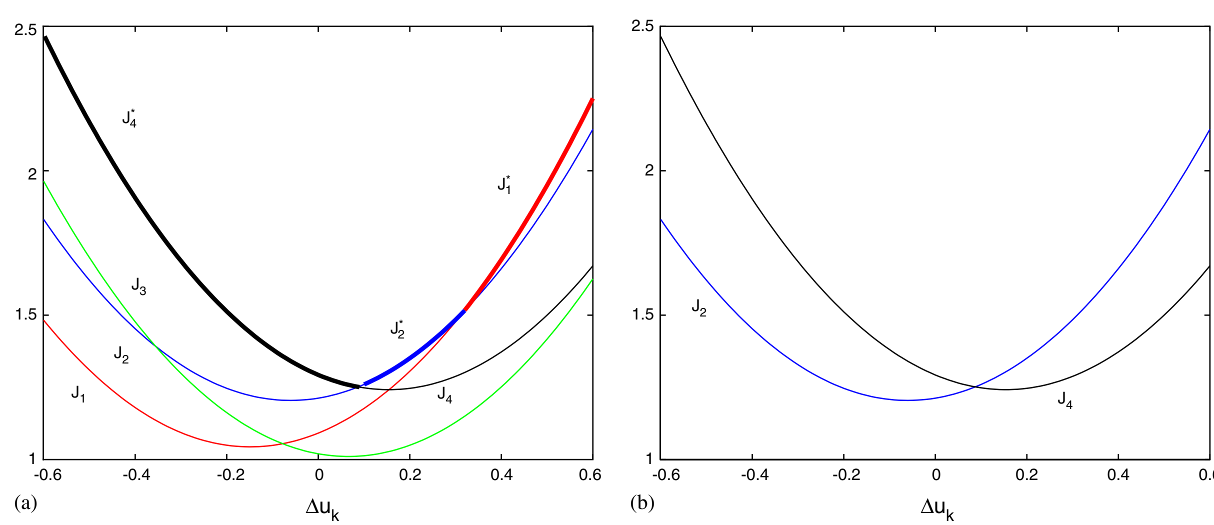

Fig. 1. Two min–max problems with the same solution: (a) full min–max with all curves and (b) reduced min–max with only curves related to active vertices.

Now consider the reduced min–max problem in which only the active vertices at x are considered. Taking into account (6) it is straightforward to show that

$$\min_{\mathbf{v}} \max_{\mathbf{w}_i \in I} J(\mathbf{v}, \mathbf{w}_i, x) \leqslant \min_{\mathbf{v}} \max_{\mathbf{w}_i \in I} J(\mathbf{v}, \mathbf{w}_i, x) \quad \text{s.t.} \quad J(\mathbf{v}, \mathbf{w}_i, x) = J(\mathbf{v}, \mathbf{w}_j, x) \quad \forall \mathbf{w}_i, \mathbf{w}_j \in I. \tag{9}$$

Note that in the min–max problem at the right side of (9) all the quadratic functions $J(\mathbf{v}, \mathbf{w}_i, x)$ considered have always the same value thus it can be replaced by

$$J^*(x) \leqslant J_I^*(x) = \min_{\mathbf{v}} J(\mathbf{v}, \mathbf{w}_i, x) \quad \text{s.t.} \quad J(\mathbf{v}, \mathbf{w}_i, x) = J(\mathbf{v}, \mathbf{w}_j, x), \quad \mathbf{w}_i \in I \;\; \forall \mathbf{w}_j \in I. \tag{10}$$

This implies that $J^*(x) \leqslant J_I^*(x)$. As $\mathbf{v}^*(x)$ is feasible for (7) it can be concluded that $\mathbf{v}^*(x)$ is a minimizer of (7) and because of strictly convexity of J on $\mathbf{v}$ the minimizer is unique, thus $\mathbf{v}^*(x) = \mathbf{v}_I(x) = K_I x + q_I, \; \forall x \in \mathrm{CR}_I$, i.e. $\mathbf{v}^*(x)$ is an affine function of x. □

To characterize the explicit solution to the min–max problem, it is necessary to characterize the critical regions CR_I.

The following proposition can be stated from the results presented in [19,20] and determines which conditions must be fulfilled by a vertices set in order to be active in a given state.

Proposition 3. *If the set of vertices $I = \{\mathbf{w}_i\}$ satisfies both the following conditions*:

- C1: $J(\mathbf{v}_I(x), \mathbf{w}_i, x) > J(\mathbf{v}_I(x), \mathbf{w}_j, x) \; \forall \mathbf{w}_i \in I, \forall \mathbf{w}_j \notin I$,
- C2: $\mathbf{v}_I(x)$ *is a local minimizer of* $\max_{\mathbf{w}_i \in I} J(\mathbf{v}, \mathbf{w}_i, x)$,

then I is the active set of vertices of x.

Proof. Note that if C1 holds then

$$J(\mathbf{v}_I(x), \mathbf{w}_i, x) > J(\mathbf{v}_I(x), \mathbf{w}_j, x) \quad \forall \mathbf{w}_i \in I, \forall \mathbf{w}_j \notin I,$$

$$J(\mathbf{v}_I(x), \mathbf{w}_i, x) = J(\mathbf{v}_I(x), \mathbf{w}_j, x) \quad \forall \mathbf{w}_i, \mathbf{w}_j \in I,$$

also holds. Thus, if C1 is satisfied $J_I(\mathbf{v}_I(x), x) = J_{\max}(\mathbf{v}_I(x), x)$.

From the definition of local minima (see [2]), $\mathbf{v}_I(x)$ is a local minima of

$$J_{\max}(\mathbf{v}, x) \quad \text{if } \forall \mathrm{d}u \in S, \nabla J(\mathbf{v}_I(x), \mathbf{w}_i, x)\, \mathrm{d}u \geqslant 0 \quad \text{for some } \mathbf{w}_i \in \mathrm{ver}(W_N), \tag{11}$$

where S is a sufficiently small ball. In the same way, if C2 is satisfied then

$$\forall \mathrm{d}u \in S, \nabla J(\mathbf{v}_I(x), \mathbf{w}_i, x)\, \mathrm{d}u \geqslant 0 \quad \text{for some } \mathbf{w}_i \in I. \tag{12}$$

As $I \subseteq \mathrm{ver}(W_N)$, and (12) holds for $\mathbf{v}_I(x)$, (11) also holds for $\mathbf{v}_I(x)$ as it is less restrictive than (12). Thus $\mathbf{v}_I(x)$ is a local minima of $J_{\max}(\mathbf{v}, x)$. As $J_{\max}(\mathbf{v}, x)$ is strictly convex on $\mathbf{v}$, $\mathbf{v}_I(x)$ is also the global minimizer of $J_{\max}(\mathbf{v}, x)$ and thus equal to the solution of (3) $\mathbf{v}^*(x)$. Moreover, this implies that $J(\mathbf{v}_I(x), \mathbf{w}_i, x) = J^*(x)$ for all $\mathbf{w}_i \in I$ and therefore I is the set of active vertices for x. □

4. Region characterization

In this section, a constructive algorithm to obtain the critical region of a set of vertices I is presented. Each region is a polyhedron that will be characterized not only by its active vertices but also by the shared boundaries with their neighboring critical regions. Moreover, these boundaries are characterized by the set of active vertices of the neighboring regions. To compute these boundaries it is useful to introduce the definition of neighboring polyhedra.

Definition 4. Let a polyhedron $X \subset \mathbb{R}^n$ be represented by the linear inequalities $Ax \leqslant b$. Let the ith hyperplane, $a_i^{\mathrm{T}} x = b_i$ be denoted by H. If $X \cap H$ is $(n-1)$-dimensional then $F = X \cap H$ is called a facet of the polyhedron.

Definition 5. Two polyhedra are neighboring polyhedra if they have a common facet.

Definition 6. Two vertices sets are neighboring sets if they have neighbor critical regions.

The boundary between neighboring regions (i.e. a facet) satisfies the following properties that will be used to determine the inequalities of each critical region.

Proposition 4. *Consider two neighboring regions* CR_1, CR_2 *with corresponding active sets* I_1, I_2. *Let F be their common facet and H the separating hyperplane*, *then all the following statements hold*:

(a) *H is defined by the equality* $\mathbf{v}_{I_1}(x) = \mathbf{v}_{I_2}(x)$.

(b) *If $\mathbf{w}_i \in I_1$, $\mathbf{w}_j \in I_2$ and $\mathbf{w}_j \notin I_1$, then H is defined as the hyperplane corresponding to the inequality defined as*

$$J(\mathbf{v}_{I_1}(x), \mathbf{w}_i, x) \geqslant J(\mathbf{v}_{I_1}(x), \mathbf{w}_j, x).$$

(c) *If $\mathbf{w}_i \in I_1$, $\mathbf{w}_j \in I_2$ and $\mathbf{w}_i \notin I_2$, then H is defined as the hyperplane corresponding to the inequality defined as*

$$J(\mathbf{v}_{I_2}(x), \mathbf{w}_j, x) \geqslant J(\mathbf{v}_{I_2}(x), \mathbf{w}_i, x).$$

Proof. (a) Because of the uniqueness of the solution of problem (3) along the facet shared by CR_1 and CR_2 the solution of each active set I_1, I_2 (computed as in (7)) must be the same thus:

$$\forall x \in F, \quad \mathbf{v}_{I_1}(x) = \mathbf{v}_{I_2}(x) = \mathbf{v}^*(x).$$

(b) and (c) due to Proposition 3,

$$\forall x \in \mathrm{CR}_{I_1}, \quad J(\mathbf{v}_{I_1}(x), \mathbf{w}_i, x) \geqslant J(\mathbf{v}_{I_1}(x), \mathbf{w}_j, x) \quad \mathbf{w}_i \in I_1, \quad \mathbf{w}_j \notin I_1,$$

$$\forall x \in \mathrm{CR}_{I_2}, \quad J(\mathbf{v}_{I_2}(x), \mathbf{w}_j, x) \geqslant J(\mathbf{v}_{I_2}(x), \mathbf{w}_i, x) \quad \mathbf{w}_j \in I_2, \quad \mathbf{w}_i \notin I_2$$

thus taking into account that $\forall x \in F$, $\mathbf{v}_{I_1}(x) = \mathbf{v}_{I_2}(x)$ then,

$$\begin{aligned} J(\mathbf{v}_{I_1}(x), \mathbf{w}_i, x) &\geqslant J(\mathbf{v}_{I_1}(x), \mathbf{w}_j, x) \\ &= J(\mathbf{v}_{I_2}(x), \mathbf{w}_j, x) \\ &\geqslant J(\mathbf{v}_{I_2}(x), \mathbf{w}_i, x), \end{aligned}$$

so both hyperplanes are coincident. □

Taking (4) into account, it can be seen that the inequalities on the previous proposition, are linear inequalities. In the following, a normalized definition of these inequalities is presented.

Proposition 5. *Given I, $\mathbf{w}_i \in I$ and $\mathbf{w}_j \notin I$, the inequality $J(\mathbf{v}_I(x), \mathbf{w}_i, x) \geqslant J(\mathbf{v}_I(x), \mathbf{w}_j, x)$ is equivalent to $a_j^{\mathrm{T}}(I)x \leqslant b_j(I)$, where $a_j(I)$ and $b_j(I)$ can be obtained as follows*

$$[a_j(I)\ b_j(I)] = \begin{cases} \left[\dfrac{\hat{a}_j(I)}{\|\hat{a}_j(I)\|_2}\ \dfrac{\hat{b}_j(I)}{\|\hat{a}_j(I)\|_2}\right] & \text{if } \|\hat{a}_j(I)\|_2 \neq 0, \\ [\hat{a}_j(I)\ \hat{b}_j(I)] & \text{otherwise}, \end{cases}$$

where

$$\hat{a}_j(I) = -2(\mathbf{w}_i - \mathbf{w}_j)^{\mathrm{T}}(H_w^{\mathrm{T}} H_u K_I + H_w^{\mathrm{T}} H_x),$$

$$\hat{b}_j(I) = 2(\mathbf{w}_i - \mathbf{w}_j)^{\mathrm{T}} H_w^{\mathrm{T}} H_u q_I + \mathbf{w}_i^{\mathrm{T}} H_w^{\mathrm{T}} H_w \mathbf{w}_i - \mathbf{w}_j^{\mathrm{T}} H_w^{\mathrm{T}} H_w \mathbf{w}_j.$$

This proposition follows directly from the definition of $J(\mathbf{v}, \mathbf{w}_i, x)$ and $\mathbf{v}_I(x)$ (Proposition 3).

Propositions 4 and 5 characterize the boundaries of a critical region if the sets of the neighboring critical regions are known. Each facet between each neighbor contributes with a linear inequality to the description of the critical region. Note that if $a^{\mathrm{T}}x = b$ defines the boundary hyperplane between two regions, $ax \leqslant b$ characterizes one of them and $ax \geqslant b$ characterizes the other.

The next proposition gives a necessary condition that two sets of active vertices must satisfy in order to be neighbors.

Proposition 6. *Consider two neighboring regions CR_{I_1}, CR_{I_2}, then it holds*

$$\text{rank } (M) = 1, \tag{13}$$

with $M = [(K_{I_1} - K_{I_2})(q_{I_2} - q_{I_1})]$, where K_{I_1}, K_{I_2}, q_{I_2} and q_{I_1} are the matrices and vectors which define the optimal solution on CR_{I_1} and CR_{I_2}.

Proof. By Proposition 4, in the boundary between two critical regions it holds $\mathbf{v}_{I_1}(x) = \mathbf{v}_{I_2}(x)$. This equality results in a system of *Nu* linear equations, namely $(K_{I_1} - K_{I_2})x = q_{I_2} - q_{I_1}$. These equalities define a region in the state space. Note that the boundary is an hyperplane (i.e., a facet) thus (13) has to be fulfilled by the candidate boundary to be a facet. □

Therefore, the possible neighbors of a set I can be defined using Proposition 6. Then, using Proposition 4, the facet between two critical neighboring regions can be obtained. With these results it is possible to build an algorithm that obtains the critical region corresponding to a given set and a list of its neighboring sets.

Algorithm 1. Algorithm to define the critical region CR_I of a set I

- Build all possible vertex sets I_i
- For each I_i if (13) is satisfied, characterize the boundary by Proposition 4
- Eliminate redundant inequalities.

Algorithm 1 makes an exhaustive search of all the possible sets so it is assured that the real neighboring sets are explored, thus the critical region is computed correctly. Note that the neighboring sets are those for which (13) holds and contribute with non-redundant inequalities to the description of CR_I. This exploration is not efficient as the number of active sets that can be obtained with $2^{N \cdot n_w}$ vertices is very large. In the following, an alternative algorithm that can be implemented in an efficient manner is presented in Section 4.1.

4.1. *Efficient algorithm*

In this section, an efficient algorithm that computes the critical region for a given set of vertices I is presented. This algorithm is based on building a set of possible neighboring vertices sets $\Phi(I)$, and then explore each of them as in Algorithm 1.

Proposition 7. *Let I and I_n be sets of vertices and CR_I, CR_{I_n} their related critical regions. Let I_r be a non-empty set of vertices such that $I_r \cap I = \emptyset$. If regions CR_I and CR_{I_n} are neighbors and satisfy that $I_n \subseteq I \cup I_r$ then*

$$\begin{bmatrix} a_j(I) \\ b_j(I) \end{bmatrix} = \begin{bmatrix} a_i(I) \\ b_i(I) \end{bmatrix} \quad \forall \mathbf{w}_i, \mathbf{w}_j \in I_r. \tag{14}$$

Proof. Proposition 5 states that the separating hyperplanes between CR_I and CR_{I_n} are given by

$$a_j(I)^{\mathrm{T}} x = b_j(I) \quad \forall \mathbf{w}_j \in I_r.$$

As a facet must be $n_x - 1$ dimensional, these separating hyperplanes must be the same. □

Definition 7. Let $I = \{\mathbf{w}_i\}$ be a set of vertices then $\Gamma(I)$ is defined as the collection of all the possible non-empty subsets that can be built with all the vertices $\mathbf{w}_i$ of I.

Definition 8. Let Ψ and Γ be collections of sets of vertices such that $\Psi = \{\bigcup_{i \in A} I_i\} \cup \emptyset$ and $\Gamma = \{\bigcup_{j \in B} I_j\} \cup \emptyset$, where A and B are the sets of indices of the sets of each collection. Then the collection $\Psi \otimes \Gamma$ is defined as

$$\Psi \otimes \Gamma = \{I \neq \emptyset | I = I_1 \cup I_2, I_1 \in \Psi, I_2 \in \Gamma\},$$

where $\emptyset$ denotes the empty set.

Definition 9. Let $I = \{\mathbf{w}_i\}$ be a set of vertices. The max region $\mathrm{CR}_I^{\max}$ is defined as the region of the state space where C1 (Proposition 3) is satisfied, i.e.,

$$\mathrm{CR}_I^{\max} = \{x | J(\mathbf{v}_I(x), \mathbf{w}_i, x) \geqslant J(\mathbf{v}_I(x), \mathbf{w}_j, x), \ \ \mathbf{w}_i \in I \ \forall \mathbf{w}_j \notin I\}.$$

By definition $\mathrm{CR}_I \subseteq \mathrm{CR}_I^{\max}$.

Algorithm 2. Algorithm to compute the collection of possible neighboring sets $\Phi(I)$ of the set of vertices I. Let $A(I)$ be an auxiliary collection of candidate neighboring sets.

- $A(I) = \emptyset$
- Build $\mathrm{CR}_I^{\max}$.
- For each vertex $\mathbf{w}_j \notin I$:
  - If $a_j^{\mathrm{T}}(I)x \leqslant b_j(I)$ is not redundant in $\mathrm{CR}_I^{\max}$ then $A(I) = A(I) \cup \mathbf{w}_j$.
- $\Psi(I) = \{I_r | I_r \subseteq A(I)$ and satisfies Eq. (14)$\}$
- $\Phi(I) = \Psi(I) \otimes \Gamma(I)$.

Theorem 1. *All the neighbors of a given set I are included in the collection of sets $\Phi(I)$ built with Algorithm* 2.

Proof. All possible vertices rejected due to a redundant inequality with the max region $\mathrm{CR}_I^{\max}$ cannot be neighbors to the critical region CR_I because $\mathrm{CR}_I \subseteq \mathrm{CR}_I^{\max}$.

All subsets of I are taken into account in $\Gamma(I)$, therefore the only possible sets I_n that could have been rejected are those that satisfy that $I_n \subseteq I \cup I_r$, where $I_r \neq \emptyset$ and $I_r \cap I = \emptyset$.

Suppose that $I_r \notin \Psi(I)$ and that I_n is neighbor of I. Then I_r satisfies Proposition 7 and by construction $I_r \in \Psi(I)$. Therefore $I_n \in \Phi(I)$. □

The following efficient algorithm uses Algorithm 2.

Algorithm 3. Algorithm to define the critical region CR_I of a set I

- Build $\Phi(I)$ as in Algorithm 2
- For each $I_i \in \Phi(I)$ if (13) is satisfied, characterize the boundary by Proposition 4
- Eliminate redundant inequalities.

5. Characterization of the partition

The explicit solution of the min–max problem could be obtained exploring all the possible active sets. This is the strategy followed by the reverse transformation method (see [16,22]). However, there is a combinatorial explosion of the amount of possible sets. Using the previous results, the explicit piecewise affine solution of a min–max problem can be obtained using the following algorithm which does not explore all the possible sets, but only those which are solution to the problem in a region of the state space. Exploring all these sets assures that the whole state space partition is obtained.

Algorithm 4. Algorithm to compute the explicit solution of the min–max problem (3). Let S_c be the collection of candidate active sets, S_e the collection of explored sets and x_0 an initial state.

- Find a valid active vertices set I_0 solving problem (3) for x_0 using numerical methods.
- $S_c = I_0$
- $S_e = \emptyset$
- (a): Extract a set of vertices I from S_c
- $S_e = S_e \cup I$
- Build $\mathbf{v}_I$ as in Proposition 3
- Build CR_I as in Algorithm 3
- For each facet of CR_I
  - Calculate neighboring set I_a
  - If I_a is not in $S_c \cup S_e$ then $S_c = S_c \cup I_a$
- If S_c is not empty go to step (a) else stop.

This algorithm is based on the ideas for partitioning the state space presented in [24] used for exploring the state space of a mpQP problem. Any full dimensional region must have at least a neighboring region. The algorithm explores a given set from a list of candidate sets of active vertices. For that given set, its critical region is computed and all of its neighboring sets determined. Then, the algorithm adds to the list of candidates those neighboring sets that have not been previously explored or are already in the collection of candidates. This ensures that each set is only explored once and that all possible sets are explored. The algorithm finishes when the list of candidates is empty.

5.1. Complexity

The complexity of the state partition depends on the particular system. The number of vertices grows exponentially with the prediction horizon. This fact makes finding the explicit solution a demanding problem. However, as pointed out in [1], not all possible vertices are implied in the solution. In that work, a vertex rejection algorithm is presented.

It defines a set of vertices $\text{red}(W_N) \subseteq \text{ver}(W_N)$ which satisfies

$$\max_{\mathbf{w}_i \in \text{ver}(W_N)} J(\mathbf{v}, \mathbf{w}_i, x) = \max_{\mathbf{w}_i \in \text{red}(W_N)} J(\mathbf{v}, \mathbf{w}_i, x).$$

Although vertex rejection efficiency depends greatly on the problem parameters, from the simulation results observed, the number of vertices can be manageable for a wide family of problems [1].

Finally, although the memory storage requirements are high, the online computational burden of explicit controllers is low because efficient search methods can be used to find the critical region for each x (see [9]).

6. Constraint handling

This paper deals with the explicit solution of an unconstrained min–max problem. The optimization problem structure is different for the constrained formulation and the concepts used for defining the explicit solution cannot be applied. However, both problems are strongly related.

Consider system (1) subject to state and input constraints $x_k \in X$ and $u_k \in U$ where X and U are polyhedral sets. The optimization problem for constrained MMMPC is posed as

$$J_c^*(x) = \min_{\mathbf{v}} \max_{\mathbf{w} \in W_N} J(\mathbf{v}, \mathbf{w}, x),$$

s.t.

$$x_j \in X \quad \forall \mathbf{w} \in W_N, \quad j = 0 \ldots N,$$

$$u_j \in U \quad \forall \mathbf{w} \in W_N, \quad j = 0 \ldots N-1.$$

Note that in this formulation is also used the linear feedback law $u_j = Kx_j + v_j$ to introduce some kind of feedback in the predictions.

Taking into account (2), when X and U are polyhedral regions, matrices F, G, m and M can be found such that the feasible set S_F can be expressed as

$$S_F = \{(x, \mathbf{v}) | Fx + G\mathbf{v} \leqslant m + M\mathbf{w}, \forall \mathbf{w} \in W_N\}. \tag{15}$$

For linear systems with additive uncertainties it is possible to reduce the number of constraints that define the feasible set S_F. It can be seen that (15) is equivalent to

$$S_F = \{(x, \mathbf{v}) |\ Fx + G\mathbf{v} \leqslant d\},$$

where d is a vector such that its ith entry satisfies

$$d_i = m_i + \min_{\mathbf{w} \in W_N} M_i \mathbf{w},$$

and m_i and M_i are the ith element and row of vector m and matrix M, respectively.

The optimum solution for the unconstrained problem can be used to propose a modified constrained problem with guaranteed performance. This controller is based on evaluating the control correction effort that makes the optimal solution for the unconstrained problem feasible and minimizes an upper bound of the min–max cost function. The future control inputs are defined as

$$u_k = Kx_k + v_k^*(x) + z_k,$$

where $v_k^*(x)$ is the optimum control effort at time step k of the unconstrained problem and z_k is the correction term to assure constraint satisfaction.

Taking into account (4) it is possible to find matrices H_{zz}, F_x, F_u and F_w such that

$$\begin{aligned} J(\mathbf{v}^*(x) + \mathbf{z}, \mathbf{w}, x) &= J(\mathbf{v}^*(x), \mathbf{w}, x) + \mathbf{z}^{\mathrm{T}} H_{zz} \mathbf{z} \\ &\quad + 2\mathbf{z}^{\mathrm{T}} (F_x x + F_u \mathbf{v}^*(x) + F_w \mathbf{w}), \end{aligned}$$

where $\mathbf{v}^*(x)$ is the solution of the unconstrained problem and $\mathbf{z} = [z_k \cdots z_{k+N-1}]^{\mathrm{T}}$ are the future correction terms.

An upper bound for the max function can then be found as

$$\begin{aligned} &\max_{\mathbf{w} \in W_N} J(\mathbf{v}^*(x) + \mathbf{z}, \mathbf{w}, x) \\ &\quad \leqslant \max_{\mathbf{w} \in W_N} J(\mathbf{v}^*(x), \mathbf{w}, x) + \max_{\mathbf{w} \in W_N} \mathbf{z}^{\mathrm{T}} H_{zz} \mathbf{z} \\ &\qquad + 2\mathbf{z}^{\mathrm{T}} (F_x x + F_u \mathbf{v}^*(x) + F_w \mathbf{w}). \end{aligned}$$

Using the explicit solution of the unconstrained problem and taking into account that $\|\mathbf{w}\|_\infty \leqslant \varepsilon$,

$$\begin{aligned} &\max_{\mathbf{w} \in W_N} J(\mathbf{v}^*(x) + \mathbf{z}, \mathbf{w}, x) \\ &\quad \leqslant J^*(x) + \mathbf{z}^{\mathrm{T}} H_{zz} \mathbf{z} + 2\mathbf{z}^{\mathrm{T}} (F_x x + F_u \mathbf{v}^*(x)) \\ &\qquad + 2\varepsilon \| F_w^{\mathrm{T}} \mathbf{z} \|_1. \end{aligned}$$

Therefore, the proposed optimization problem is easily converted into a QP problem which can be solved efficiently, namely

$$\begin{aligned} \tilde{J}^*(x) = \min_{\mathbf{z}} &\ J^*(x) + \mathbf{z}^{\mathrm{T}} H_{zz} \mathbf{z} + 2\mathbf{z}^{\mathrm{T}} F_x x \\ &+ 2\mathbf{z}^{\mathrm{T}} F_u \mathbf{v}^*(x) + 2\varepsilon \| F_w^{\mathrm{T}} \mathbf{z} \|_1 \\ &\text{s.t.} \quad Fx + G(\mathbf{v}^*(x) + \mathbf{z}) \leqslant d. \end{aligned} \tag{16}$$

In this way, the control input of the proposed implementation is defined as

$$u_0 = Kx + v_0^* + z_0^* = K_{\text{MPC}}(x),$$

where v_0^* is the solution of the unconstrained MMMPC and z_0^* is the first correction term of the proposed implementation.

In this formulation, an upper bound of the optimum value of the cost function is minimized, that is $J_c^*(x) \leqslant \tilde{J}^*(x)$. It is important to note that the controller gives the optimum solution when no constraints are active at the solution of (16).

The main benefit is that the computational burden of this implementation is much lower than solving the constrained min–max problem. In this case, first the explicit form of the unconstrained MMMPC is evaluated, and the a QP is solved. The constrained MMMPC is a NP-Hard problem that in general has a much higher computational burden.

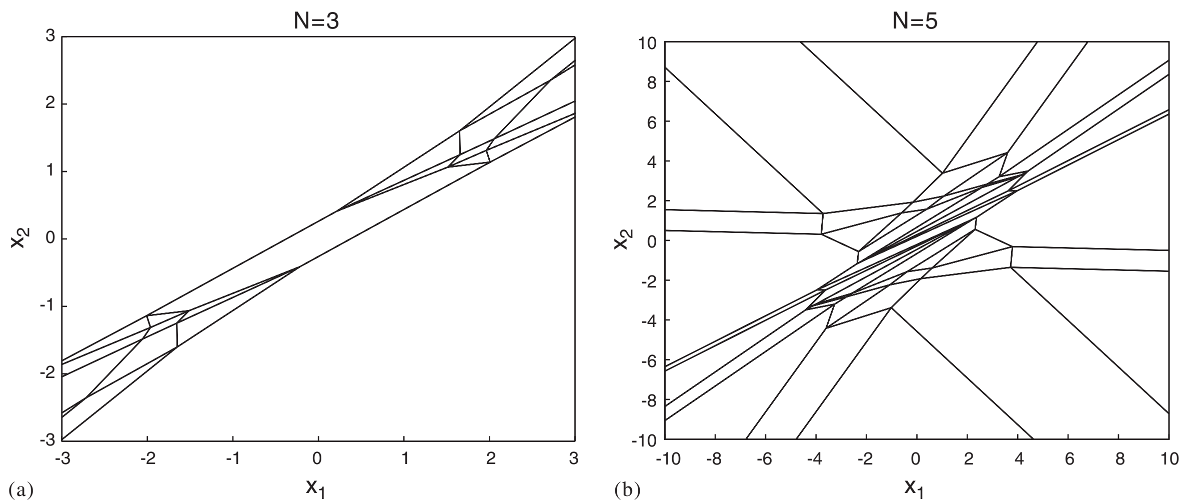

Fig. 2. Partition of the state space for the example with $N=3$ (a) and $N=5$ (b).

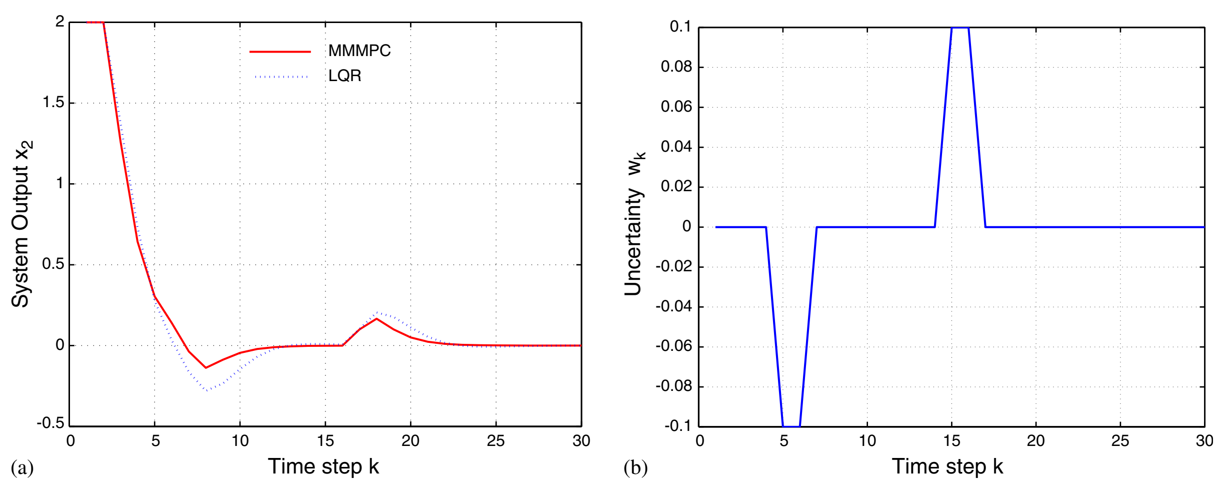

Fig. 3. Closed loop simulation for an MMMPC with $N=3$ and an LQR controller (a) for an uncertainty realization (b).

7. Simulation example

Consider the problem of robustly steering to the origin the system

$$x_{k+1} = \begin{bmatrix} 0 & 1 \\ -0.9 & 1.9 \end{bmatrix} x_k + \begin{bmatrix} 0 \\ 0.5 \end{bmatrix} u_k + \begin{bmatrix} 0 \\ 1 \end{bmatrix} w_k.$$

This model represents a first order SISO system with an integrator. The uncertainty is restricted to the set $W = \{w : \|w\|_\infty \leqslant 0.1\}$. The weighting matrices are $Q = P = [\begin{smallmatrix} 0 & 0 \\ 0 & 1 \end{smallmatrix}]$, $R = 1$. The control input is computed as $u_k = Kx_k + v_k$ where $K = [1.1021 \quad -1.7248]$ is the LQR feedback gain. Note that because of the uncertainty, the system cannot be regulated to the origin, but to a bounded set that contains the origin [15].

For a prediction horizon $N=3$ the algorithm presented in Section 5 has been applied to this system. The explicit form of the controller is defined by 27 regions. For a prediction horizon $N=5$ the explicit form of the controller is defined by 55 regions. Fig. 2 shows both state partitions.

We compare MMMPC with the LQR control law. The closed-loop system is simulated from the initial state $x_0 = [\begin{smallmatrix} 2 \\ 2 \end{smallmatrix}]$ with the disturbance profile of Fig. 3(b). Fig. 3(a) shows the output trajectories (x_2) of the SISO system. It can be seen how the MMMPC shows a better disturbance rejection behavior.

8. Conclusions

An algorithm to obtain the explicit form of the MMMPC control law for linear unconstrained systems has been presented. The algorithm does not require an exhaustive search of all possible active sets, so, although there is an exponential growth of the complexity of the algorithm with the prediction horizon, in many applications it is manageable because all the computation needed to determine the explicit form of the controller are done off-line.

For nominal MPC controllers the multi-parametric approach has been applied with success in a broad set of systems even though the number of regions of the explicit solution grow in an exponential manner with the dimension of the state and the prediction horizon. The results presented in this paper allows one to apply the same ideas to robust MPC and broaden the range of processes to which the MMMPC controller can be applied in practice.

A controller for constrained systems, has also been presented. This controller uses the explicit solution of the unconstrained problem to evaluate, in an efficient way, an upper bound of the inner maximization problem. Minimizing this upper bound in a constrained optimization problem guarantees constraint satisfaction and performance.

References

[1] T. Alamo, D. Muñoz de la Peña, E. Camacho, An offline algorithm for reducing the computational burden of a MPC min max controller, in: Proceedings of the Conference on Decision and Control, Maui, Hawaii, USA, 2003.

[2] M. Bazaraa, C. Shetty, Nonlinear Programming. Theory and Algorithms, Wiley, New York, 1979.

[3] A. Bemporad, Reducing conservativeness in predictive control of constrained systems with disturbances, in: Proceedings of the American Control Conference, Tampa, Florida, USA, 1998, pp. 1384–1389.

[4] A. Bemporad, F. Borreli, M. Morari, Min–max control of constrained uncertain discrete-time linear systems, IEEE Trans. Automat. Control 48 (9) (2003) 1600–1606.

[5] A. Bemporad, M. Morari, V. Dua, E. Pistikopoulos, The explicit linear quadratic regulator for constrained systems, Automatica 38 (2002) 3–20.

[6] S. Boyd, L. Vandenberghe, Convex Optimization, Cambridge University Press, UK, 2004.

[7] E. Camacho, C. Bordóns, Model Predictive Control in the Process Industry, second ed., Springer, Berlin, 2004.

[8] P.J. Campo, M. Morari, Robust model predictive control, in: Proceedings of the American Control Conference, 1987, pp. 1021–1026.

[9] T. Johansen, A. Grancharova, Approximate explicit constrained linear model predictive control via orthogonal search tree, Trans. Automat. Control 48 (5) (2003) 810–815.

[10] E.C. Kerrigan, J. Maciejowski, Feedback min–max model predictive control using a single linear program: robust stability and the explicit solution, Internat. J. Robust Nonlinear Control 14 (2004) 395–413.

[11] Y. Kim, W. Kwon, An application of min–max generalized predictive control to sintering processes, Control Eng. Practice 6 (1998) 999–1007.

[12] M. Kothare, V. Balakrishnan, M. Morari, Robust constrained model predictive control using linear matrix inequalities, Automatica 32 (1996) 1361–1379.

[13] W. Langson, I. Chryssochoos, S. Rakovic, D. Mayne, Robust model predictive control using tubes, Automatica 40 (2004) 125–133.

[14] J.H. Lee, Z. Yu, Worst case formulations of model predictive control for systems with bounded parameters, Automatica 3 (5) (1997) 763–781.

[15] J. Löfberg, Minimax approaches to robust model predictive control, Ph.D. Thesis, April, 2003.

[16] D. Mayne, S. Rakovic, Optimal control of constrained piecewise affine discrete time systems using reverse transformation, in: Conference on Decision and Control, Las Vegas, Nevada, USA, 2002, pp. 1546–1551.

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

[18] C. Porfirio, E. Almeida, D. Odloak, Multi-model predictive control of an industrial c3/c4 splitter, Control Eng. Practice 15 (2002) 765–779.

[19] D. Ramírez, E. Camacho, On the piecewise linear nature of min–max model predictive control with bounded uncertainties, in: Proceedings of the Conference on Decision and Control, 2001.

[20] D. Ramírez, E. Camacho, Characterization of min–max MPC with global uncertainties, in: Proceedings of the American Control Conference, ACC, 2002.

[21] P.O.M. Scokaert, D. Mayne, Min–max feedback model predictive control for constrained linear systems, IEEE Trans. Automat. Control 43 (8) (1998) 1136–1142.

[22] M. Seron, G. Goodwin, J. De Doná, Finitely parameterised implementation of receding horizon control for constrained linear systems, in: American Control Conference, Anchorage, AK, 2002, pp. 4481–4485.

[23] R. Smith., Robust model predictive control of constrained linear systems, in: Proceedings of the American Control Conference, 2004, pp. 245–250.

[24] P. Tøndel, T. Johansen, A. Bemporad, An algorithm for multi-parametric quadratic programming and explicit MPC solutions, Automatica 39 (3) (2003) 489–497.

[25] S. Veres, J. Norton, Predictive self-tuning control by parameter bounding and worst case design, Automatica 29 (4) (1993) 911–928.

[26] Y.J. Wang, J. Rawlings, A new robust model predictive control method I: theory and computation, J. Process Control 14 (3) (2003) 231–247.

[27] H.S. Witsenhausen, A minimax control problem for sampled linear systems, IEEE Trans. Automat. Control 13 (1) (1968) 5–21.