Available online at www.sciencedirect.com
SCIENCE DIRECT®

Control Engineering Practice 13 (2005) 1463–1471

CONTROL ENGINEERING PRACTICE

www.elsevier.com/locate/conengprac

# Application of an explicit min-max MPC to a scaled laboratory process

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

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

Received 24 July 2003; accepted 23 December 2004
Available online 8 February 2005

**Abstract**

Min-max model predictive control (MMMPC) requires the on-line solution of a min-max problem, which can be computationally demanding. The piecewise affine nature of MMMPC has been proved for linear systems with quadratic performance criterion. This paper shows how to move most computations off-line obtaining the explicit form of this control law by means of a heuristic algorithm. These results are illustrated with an application to a scaled laboratory process with dynamics fast enough to preclude the use of numerical solvers.


*Keywords:* Predictive control; Minimax techniques; Multi-parametric programming; Piecewise linear controllers; Uncertainty

## 1. Introduction

Model predictive control (MPC) is a popular strategy originated in the late seventies for the automation of large multivariable industrial processes. The basic idea of MPC is the explicit use of a model of the process to predict the process output at future time instants, and to obtain the control signal by minimizing a cost function that depends on such predictions.

One approach used in robust MPC when uncertainties are taken into account in the model, is to minimize the objective function for the worst possible realization of the uncertainty. This strategy is known as min-max model predictive control (MMMPC). Min-max control was originally proposed in Witsenhausen (1968) in the context of robust receding control. In robust MPC the problem was first tackled in Campo and Morari (1987). All min-max control techniques have in common a high computational burden which limits the range of processes to which they can be applied (see Veres & Norton, 1993; Lee & Yu, 1997; Scokaert & Mayne, 1998 and the references therein). Few applications of min-max MPC can be found in literature (see Porfirio, Almeida, & Odloak, 2002; Kim & Kwon, 1998).

Recently multi-parametric programming has been applied with success to deterministic MPC to solve the optimization problems off-line in order to obtain an explicit description of the control law (see Bemporad, Morari, Dua, & Pistikopoulos, 2002; Seron, Goodwin, & De Doná, 2002). For cost functions based on 1-norms or $\infty$-norms, robust MPC controllers have also been obtained in explicit form (see Bemporad, Borrelli, & Morari, 2003; Kerrigan & Maciejowski, 2004). The piecewise affine nature of these controllers for unconstrained linear systems with quadratic cost functions was shown in Ramírez and Camacho (2001), but these results did not include an algorithm for determining the explicit solution of these controllers. This paper presents a heuristic algorithm for obtaining the different regions of a quadratic unconstrained MMMPC and the

*Corresponding author. Tel.: +34 954487347; fax: +34 954487340.
*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).

0967-0661/$ - see front matter 
doi:10.1016/j.conengprac.2004.12.008

corresponding affine solutions. An application of an explicit controller to a well known scaled laboratory process (Ljung, 1987; Li, Tsang, & Ho, 1998; Bandyopadhyay & Patranabis, 2001; Horácek, 2000) is shown. The process has simple dynamics, but is fast enough to preclude the use of any nonexplicit implementation. Experimental results are discussed in the paper.

The rest of the paper is organized as follows: In Section 2 we introduce the controller and the corresponding optimization problem. In Section 3 the piecewise linear nature of the controller is shown. In Section 4 an algorithm for building the regions of the explicit formulation is developed. In Section 5 the algorithm for exploring the state space and determining the explicit controller is presented. In Section 6 experimental results are shown. Finally, in Section 7 we present concluding remarks.

## 2. Problem formulation

Consider the discrete time CARIMA model with bounded integrated uncertainties of a SISO system

$$\Delta A(z^{-1})y_{k+1} = z^{-d}B(z^{-1})\Delta u_k + \theta_k, \tag{1}$$

where $y_k \in \mathbb{R}$ is the output, $u_k \in \mathbb{R}$ is the input, $\Delta = 1 - z^{-1}$ and $\theta_k \in \mathbb{R}$ is the uncertainty. The uncertainty $\theta_k$ is supposed to be bounded, that is $\|\theta_k\|_\infty \leqslant \varepsilon$ with $\varepsilon > 0$, and globalizes all modeling errors. This is a general way of describing uncertainty (see Camacho & Bordóns, 1999).

The objective is to compute the future control sequence in such a way that the output predictions are driven close to the set point sequence $r_k, r_{k+1}, \ldots, r_{k+N-1}$ for the prediction horizon $N$. The way in which the system will approach the desired trajectory is indicated by a cost function which depends on present and future control signals and uncertainties.

The cost function is a quadratic criterion, namely,

$$J = \sum_{j=1}^{N}(y_{k+j} - r_{k+j})^2 + \lambda \sum_{j=0}^{N-1}(\Delta u_{k+j})^2, \tag{2}$$

with $\lambda > 0$.

Taking into account the linearity of system (1), the set of $j$-ahead optimal predictions for $j = 1, \ldots, N$, can be written (see Camacho & Bordóns, 1999; Bemporad et al., 2002) in condensed form as

$$\mathbf{y} = G_u\mathbf{u} + G_\theta\boldsymbol{\theta} + G_x\mathbf{x}, \tag{3}$$

where

$$\mathbf{y} = [y_{k+1} \cdots y_{k+N}]^{\mathrm{T}} \in \mathbb{R}^N,$$

$$\mathbf{u} = [\Delta u_k \cdots \Delta u_{k+N-1}]^{\mathrm{T}} \in \mathbb{R}^N,$$

$$\boldsymbol{\theta} = [\theta_k \cdots \theta_{k+N-1}]^{\mathrm{T}} \in \mathbb{R}^N,$$

$$\mathbf{x} = [y_k \cdots y_{k-n_a} \Delta u_{k-1} \cdots \Delta u_{k-n_b}]^{\mathrm{T}} \in \mathbb{R}^{nx}.$$

The vector $\mathbf{x}$ is the state of the system.

Without loss of generality, consider a constant set point of $r_{k+j} = 0$ (see Camacho & Bordóns, 1999; Bemporad et al., 2002 for the case $r_{k+j} \neq 0$). Then, the cost function $J(\mathbf{u}, \boldsymbol{\theta}, \mathbf{x})$ becomes

$$\begin{aligned} J(\mathbf{u}, \boldsymbol{\theta}, \mathbf{x}) &= \mathbf{y}^{\mathrm{T}}\mathbf{y} + \lambda\mathbf{u}^{\mathrm{T}}\mathbf{u} \\ &= \|G_u\mathbf{u} + G_\theta\boldsymbol{\theta} + G_x\mathbf{x}\|_2^2 + \lambda\|\mathbf{u}\|_2^2, \end{aligned} \tag{4}$$

which is a quadratic convex function on $\mathbf{u}$, $\boldsymbol{\theta}$ and $\mathbf{x}$ for positive values of $\lambda$, see Camacho and Bordóns (1999); Bazaraa and Shetty (1979).

The objective of MMMPC is to minimize the objective function for the worst possible realization of the uncertainty as first proposed in Witsenhausen (1968). We define the min-max problem $P_N(\mathbf{x})$ as

$$J^*(\mathbf{x}) = \min_{\mathbf{u}} \max_{\theta \in \Theta} J(\mathbf{u}, \boldsymbol{\theta}, \mathbf{x}), \tag{5}$$

with $\Theta = \{\boldsymbol{\theta} \in \mathbb{R}^N |\ \|\boldsymbol{\theta}\|_\infty \leqslant \varepsilon\}$ the set of all possible future uncertainty trajectories of length $N$.

The control is applied in a receding horizon scheme. The solution of the optimization problem at time step $k$ provides an optimizer $\mathbf{u}^*(\mathbf{x}_k)$ which defines the future inputs along the whole prediction horizon. However, only the control input of the current time step $\Delta u_k^*$ is applied, the optimization is repeated in the following sample time with the new real state of the system $\mathbf{x}_{k+1}$.

Problem (5) is of high complexity because the evaluation of the maximum of the cost function $J(\mathbf{u}, \boldsymbol{\theta}, \mathbf{x})$ is a NP-hard problem. The maximum of a convex function is found in the boundary of the feasible region. As $J(\mathbf{u}, \boldsymbol{\theta}, \mathbf{x})$ is convex on $\boldsymbol{\theta}$, the maximum is found in at least one of the $2^N$ vertices $\boldsymbol{\theta}_i$ of $\Theta$. This way, $2^N$ evaluations of $J(\mathbf{u}, \boldsymbol{\theta}, \mathbf{x})$ are required (Bazaraa & Shetty, 1979). This implies a complexity that grows exponentially with $N$, thus the computation of the optimal control sequence $\mathbf{u}^*(\mathbf{x})$ is a difficult task. In this paper we present an algorithm to obtain an explicit piecewise affine form of $\mathbf{u}^*(\mathbf{x})$, so the evaluation of the optimum control input can be made in an efficient way.

This paper deals with a CARIMA model of a SISO system and a fixed cost criterion over the input and the output because this formulation is present in almost all industrial plants, however, the results presented can be applied to all linear systems with a quadratic cost function, in particular, for state space models with cost functions which include a terminal cost term for which stability has been proven in Alamo, Muñoz de la Peña, Limon, and Camacho (2003).

## 3. Some properties of the min-max problem

As seen before, the maximum of the cost function $J(\mathbf{u}, \boldsymbol{\theta}, \mathbf{x})$ is found in at least one of the vertices of $\Theta$. Each of the $2^N$ vertices of $\Theta$, denoted as $\boldsymbol{\theta}_i$, $i = 1, 2, \ldots, 2^N$, defines a quadratic function on $\mathbf{u}$ and $\mathbf{x}$:

$$J_i(\mathbf{u}, \mathbf{x}) = J(\mathbf{u}, \boldsymbol{\theta}_i, \mathbf{x}).$$

The following property can be stated from the results presented in Ramírez and Camacho (2002).

**Proposition 1.** *The solution* $\mathbf{u}^*(\mathbf{x})$ *of the min-max problem* (5) *in a given state vector* $\mathbf{x}$, *can be characterized by the set of vertices* $I$ *in the following way*:

$$\mathbf{u}^*(\mathbf{x}) = \mathbf{u}_I(\mathbf{x}) = \arg\min_{\mathbf{u}} J_i(\mathbf{u},\mathbf{x})$$
$$s.t.\quad J_i(\mathbf{u},\mathbf{x}) = J_j(\mathbf{u},\mathbf{x}) \quad \boldsymbol{\theta}_i \in I, \quad \forall \boldsymbol{\theta}_j \in I \tag{6}$$

*if and only if* $\mathbf{u}_I(\mathbf{x})$ *satisfies the following optimality conditions*:

- $J_i(\mathbf{u}_I(\mathbf{x}),\mathbf{x}) \geqslant J_j(\mathbf{u}_I(\mathbf{x}),\mathbf{x})$ $\boldsymbol{\theta}_i \in I$ $\forall \boldsymbol{\theta}_j \in \Theta$ (*max condition*)
- $\mathbf{u}_I(\mathbf{x})$ *is a local minimizer of* $J_I(\mathbf{u},\mathbf{x}) = \max_{\boldsymbol{\theta}_i \in I} J(\mathbf{u},\boldsymbol{\theta},\mathbf{x})$ (*local minimum condition*)

**Note 1.** The optimum found solving (6) is an affine function of $\mathbf{x}$ (i.e., $\mathbf{u}_I(\mathbf{x}) = K_I\mathbf{x} + q_I$) as, taking into account (4), it is a quadratic minimization problem with linear equality constraints.

**Definition 2.** We define the active vertices set for a given state vector $\mathbf{x}$ as the set that verifies the optimality conditions.

**Definition 3.** Let $I$ be a set of vertices. The critical region $CR_I$ is the region of the state space where set $I$ is the active vertices set.

For all states inside the critical region of $I$, the solution to the min-max problem is an affine expression of $\mathbf{x}$ (see Note 1):

$$\mathbf{u}^*(\mathbf{x}) = \mathbf{u}_I(\mathbf{x}) = K_I\mathbf{x} + q_I \quad \text{if } \mathbf{x} \in CR_I. \tag{7}$$

Therefore, $\mathbf{u}^*(\mathbf{x})$ is a piecewise linear function of the state. To characterize the explicit solution to the min-max problem, it is necessary to characterize all the critical regions $CR_I$ and their corresponding optimizers $\mathbf{u}_I(\mathbf{x})$.

## 4. Region characterization

In this section, an algorithm for determining the critical region $CR_I$ of a set $I$ of vertices is presented. The following definitions (Borrelli, 2002) will be used in the rest of the paper.

**Definition 4.** Let a polyhedron $X \in \mathbb{R}^n$ be represented by the linear inequalities $A\mathbf{x} \leqslant b$. Let the $i$th inequality be $a_i^{\mathrm{T}}\mathbf{x} \leqslant b_i$. The hyperplane defined as $a_i^{\mathrm{T}}\mathbf{x} = b_i$ is called the $i$th boundary hyperplane of the polyhedron.

**Definition 5.** Let a polyhedron $X \in \mathbb{R}^n$ be represented by the linear inequalities $A\mathbf{x} \leqslant b$. Let the $i$th hyperplane, $a_i^{\mathrm{T}}\mathbf{x} = b_i$ 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 6.** Two polyhedra are called neighboring polyhedra if they have a common facet. Two active sets are called neighboring sets if their critical regions are neighboring polyhedra.

**Note 2.** This means that in this paper, only those regions which have a common boundary of dimension $n_x - 1$ are considered neighboring regions.

**Proposition 7.** *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*:

(1) *$H$ is defined as* $\mathbf{u}_{I_1}(\mathbf{x}) = \mathbf{u}_{I_2}(\mathbf{x})$.
(2) *If* $\boldsymbol{\theta}_i \in I_1$, $\boldsymbol{\theta}_j \in I_2$ *and* $\boldsymbol{\theta}_j \notin I_1$, *then $H$ is defined as the hyperplane corresponding to the inequality defined as*
$$J_i(\mathbf{u}_{I_1}(\mathbf{x}),\mathbf{x}) \geqslant J_j(\mathbf{u}_{I_1}(\mathbf{x}),\mathbf{x}).$$
(3) *If* $\boldsymbol{\theta}_i \in I_1$, $\boldsymbol{\theta}_j \in I_2$ *and* $\boldsymbol{\theta}_i \notin I_2$, *then $H$ is defined as the hyperplane corresponding to the inequality defined as*
$$J_j(\mathbf{u}_{I_2}(\mathbf{x}),\mathbf{x}) \geqslant J_i(\mathbf{u}_{I_2}(\mathbf{x}),\mathbf{x}).$$

**Proof.** (1) Because of the uniqueness of the solution of problem (5):

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

(2) and (3) Due to Proposition 1,

$$\forall \mathbf{x} \in CR_{I_1},\ J_i(\mathbf{u}_{I_1}(\mathbf{x}),\mathbf{x}) \geqslant J_j(\mathbf{u}_{I_1}(\mathbf{x}),\mathbf{x}),$$

$$\forall \mathbf{x} \in CR_{I_2},\ J_j(\mathbf{u}_{I_2}(\mathbf{x}),\mathbf{x}) \geqslant J_i(\mathbf{u}_{I_2}(\mathbf{x}),\mathbf{x})$$

so $\forall \mathbf{x} \in F$,

$$\begin{aligned} J_i(\mathbf{u}_{I_1}(\mathbf{x}),\mathbf{x}) &\geqslant J_j(\mathbf{u}_{I_1}(\mathbf{x}),\mathbf{x}) = J_j(\mathbf{u}_{I_2}(\mathbf{x}),\mathbf{x}) \\ &\geqslant J_i(\mathbf{u}_{I_2}(\mathbf{x}),\mathbf{x}) = J_i(\mathbf{u}_{I_1}(\mathbf{x}),\mathbf{x}). \qquad \square \end{aligned}$$

The critical region for a given set of vertices $I$, if it exists, is defined as a polyhedron on the state space $\mathbb{R}^{n_x}$. If the neighboring sets are known, it is possible to characterize the inequalities that define this polyhedron with Proposition 7. For each neighbor, as $I_1 \neq I_2$, a linear inequality can be defined using the following proposition.

**Proposition 8.** *Given* $I$, $\boldsymbol{\theta}_i \in I$ *and* $\boldsymbol{\theta}_j \notin I$, *the inequality* $J_i(\mathbf{u}_I(\mathbf{x}),\mathbf{x}) \geqslant J_j(\mathbf{u}_I(\mathbf{x}),\mathbf{x})$ *is equivalent to* $a_j^{\mathrm{T}}(I)\mathbf{x} \leqslant b_j(I)$, *where* $a_j(I)$ *and* $b_j(I)$ *can be obtained as*

$$a_j(I) = -2(\boldsymbol{\theta}_i - \boldsymbol{\theta}_j)^{\mathrm{T}}(G_\theta^{\mathrm{T}}G_uK_I + G_\theta^{\mathrm{T}}G_x),$$

$$b_j(I) = 2(\boldsymbol{\theta}_i - \boldsymbol{\theta}_j)^{\mathrm{T}}G_\theta^{\mathrm{T}}G_uq_I + \boldsymbol{\theta}_i^{\mathrm{T}}G_\theta^{\mathrm{T}}G_\theta\boldsymbol{\theta}_i - \boldsymbol{\theta}_j^{\mathrm{T}}G_\theta^{\mathrm{T}}G_\theta\boldsymbol{\theta}_j.$$

This proposition stems directly from (4) and the definitions of $J_i(\mathbf{u},\mathbf{x})$ and $\mathbf{u}_I(\mathbf{x})$.

**Note 3.** If $a^{\mathrm{T}}\mathbf{x}=b$ defines the boundary hyperplane between two regions, $a^{\mathrm{T}}\mathbf{x}\leqslant b$ characterizes one of them and $a^{\mathrm{T}}\mathbf{x}\geqslant b$ characterizes the other.

With an exhaustive search of all the possible sets it is assured that the real neighboring sets are explored, and so the critical region is determined. This exploration is not efficient as the amount of these sets grows in a combinatorial explosion and, in practice, it is not possible to search them all. In this paper, a heuristic algorithm that obtains a characterization of the critical region of a given set without having to resort to an exhaustive search is proposed.

**Hypothesis 9.** Consider two neighboring regions $CR_1$, $CR_2$ with corresponding optimal active sets $I_1$, $I_2$. Then

$$I_1 = I_2 \cup \boldsymbol{\theta}_j \quad \text{or} \quad I_2 = I_1 \cup \boldsymbol{\theta}_j,$$

where $\boldsymbol{\theta}_j$ is a vertex that belongs either to one set but not to the other.

The proposed algorithm builds the possible collection of neighboring sets supposing Hypothesis 9 holds. Hypothesis 9 states that two neighboring regions have active vertices sets that only differ in one component. So, given an active set $I$, if the hypothesis is satisfied, it has at most $2^N$ neighboring sets, one for each vertex in $I$, and one for each vertex not in $I$.

This hypothesis is based on the fact that multi-vertex transitions happen when the boundary between critical regions is defined by coincident hyperplanes. These transitions occur only for special structures of the min-max problem (5) and are very unlikely, moreover, any small change of the controller parameters will define a min-max problem which does satisfy the hypothesis. A similar situation is also encountered in multi-parametric quadratic programming (see Bemporad et al., 2002).

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

(1) *Build the collection of sets $\Phi(I)$ of all possible neighboring sets under the assumption that Hypothesis* 9 *holds.*
(2) *For each $I_i \in \Phi(I)$*
  - *characterize the boundary by Proposition* 7.

The algorithm defines $2^N$ inequalities for each region. The nonredundant inequalities define the polyhedron. Moreover, each inequality has been obtained as the frontier between set $I$ and another known set, thus the neighboring sets of $I$ are also defined. The neighboring sets are those for which the linear constraints define a facet of $CR_I$.

## 5. Characterization of the partition

The explicit solution of the min-max problem could be obtained exploring all the possible active sets. 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 active in a full-dimensional region of the state space.

**Algorithm 2.** *Algorithm for constructing the explicit solution of a min-max problem.*

*Let $S_c$ be the collection of active set candidates and $S_e$ the collection of explored sets. Then the explicit solution of MMMPC can be found using the following procedure*:

(1) *Find a valid solution set $I_0$ solving problem* (5) *for a given $\mathbf{x}_0$ using numerical methods.*
(2) $S_c = I_0$.
(3) $S_e = \emptyset$.
(4) *Extract a set of vertices $I$ from $S_c$.*
(5) $S_e = S_e \cup I$.
(6) *Build $u_I$ as in Proposition* 1.
(7) *Build $CR_I$ as in Algorithm* 1.
(8) *For each facet of $CR_I$*
  - *calculate neighboring set $I_a$,*
  - *if $I_a$ is not in $S_c \cup S_e$ then $S_c = S_c \cup I_a$.*
(9) *If $S_c$ is not empty go to step* 4 *else stop.*

This algorithm is based on the ideas for partitioning the state space presented in Tøndel, Johansen, and Bemporad (2001),where the state space of a multi-parametric quadratic programming problem was explored. The algorithm explores a given set from a list of candidates. The critical region for that given set is computed and all of its neighboring sets determined. The algorithm then adds to the list of candidates those neighboring sets that have not been previously explored or are already on the list. This ensures that each set is only explored once and that all possible sets are explored given that any full-dimensional region must have at least a neighboring region. The algorithm finishes when the list of candidates is empty.

## 6. Application to a scaled laboratory process

The laboratory process to be controlled is the well known Feedback PT-326 (Ljung, 1987; Li et al., 1998; Bandyopadhyay & Patranabis, 2001; Horácek, 2000). In this equipment, a centrifugal blower draws air from the atmosphere and forces it through a heater grid inside a tube. The heater can be controlled to keep the temperature of the air at the output of the tube at a desired value. The air temperature is detected downstream of the grid by a bead thermistor. The air stream velocity can be adjusted by means of an inlet throttle attached to the blower. In this work the system input is the voltage applied to the heater and the output the voltage at the sensor. A schematic diagram of the physical setup is shown in Fig. 1(a).

A model of the process is needed to apply the proposed control strategy. First, a physical model of the temperature response for the system will be introduced. A diagram of the PT-326 is shown in Fig. 1(b).

An energy balance yields

$$\text{Heat stored} = \text{Heat in} - \text{Heat out}. \tag{8}$$

Replacing the terms of this equation by their thermodynamic equivalents yields:

$$\frac{\mathrm{d}}{\mathrm{d}t}(\rho V c_p T) = P - \rho c_p Q T, \tag{9}$$

where $c_p$ is the specific heat of air, $Q$ is the air flow rate, $\rho$ is the density, $P$ is the power, $V$ is the volume between the heater and the thermistor and $T$ is the temperature above room temperature. Combining terms yields the first order differential equation for the exhaust temperature:

$$\frac{\mathrm{d}T}{\mathrm{d}t} = \dot{T} = -\frac{Q}{V}T + \left(\frac{1}{\rho V c_p Q}\right)P. \tag{10}$$

Solving this equation yields

$$T(t) = (T_0 - T_f)\mathrm{e}^{-((t-t_0)/\tau)} + T_f. \tag{11}$$

Note that in order to obtain this model some simplifying assumptions were made, the effect of air turbulence, time delay, and other factors have been neglected. Obtaining a first principle model (that is, using physical principles) cannot usually be done reliably except for the simplest processes. Thus, the usual strategy is to resort to system identification.

For a nominal inlet throttle of 60° the system has a time constant of 0.6 s and a time delay of 0.18 s. These parameters are highly dependent on the environment conditions as well as the setting of the inlet throttle. The sampling time has been chosen to be 0.04 s. Note that the sampling time is so small that it is very difficult to use a numerical solver to obtain the control sequence even for small values of the horizon.

A discrete linear model of the PT-326 has been identified using the least squares identification method (note that this method has been applied to this process in Ljung (1987, p. 440)). Several models of different orders have been fitted to the experimental data. The best model found was

$$A(z^{-1})y_k = B(z^{-1})u_{k-1-d}, \tag{12}$$

with $A(z^{-1}) = 1 - 0.5510z^{-1} - 0.4072z^{-2}$, $B(z^{-1}) = 0.0090 + 0.0127z^{-1} + 0.0105z^{-2}$ and $d = 4$. Thus the prediction model will be

$$\Delta A(z^{-1})y(k) = B(z^{-1})\Delta u(k-1-d) + \boldsymbol{\theta}_k. \tag{13}$$

The upper graphic in Fig. 2 shows the one-step prediction error in the test set for model (12). It can be seen that the error is always between $-0.3$ and 0.3. The uncertainty bounds have been set according to the error computed using the predicted output which came from the prediction model (13) with $\boldsymbol{\theta}_k = 0$. The errors computed using this model are shown in the bottom graphic of Fig. 2. The uncertainty bounds will be set to $-0.05$ and 0.05, as 97.25% of the errors in the test set are within these bounds.

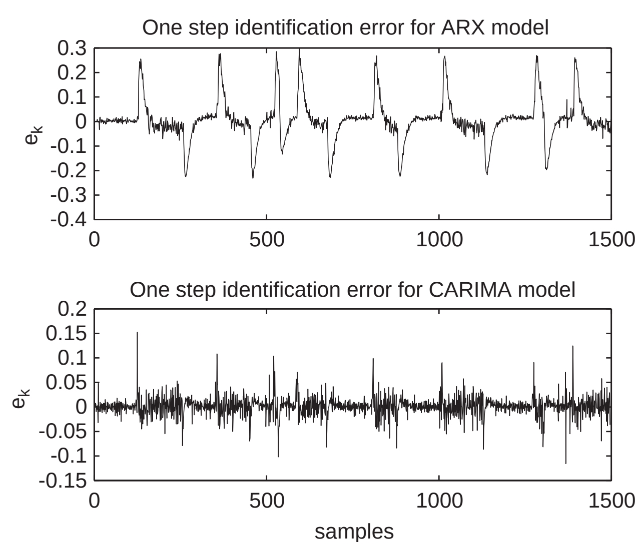


Fig. 2. One-step prediction error on the test set for models (12) and (13).

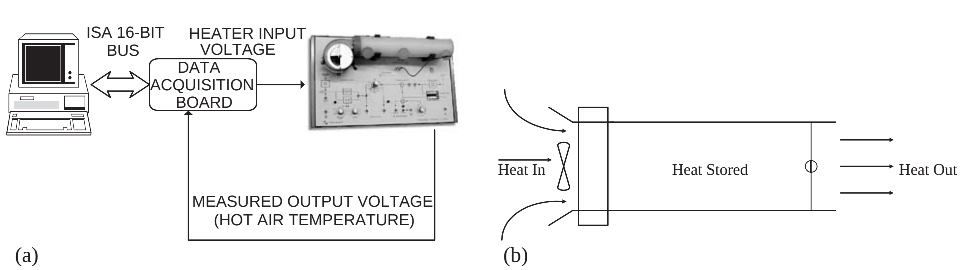


Fig. 1. A schematic diagram of the feedback PT-326 and the physical setup.

### 6.1. Results of the MMMPC

The system model has a pure delay of $d = 4$ sampling times. A standard Smith predictor dead time compensator has been used to compute the predicted delayed output which will be denoted as $\hat{y}_{k+d}$. This prediction of $y_{k+d}$ is subsequently used to compute the control sequence for the nominal system without the time delay. In this way, $\hat{y}_{k+d}$ is computed as

$$\hat{y}_{k+d} = y_{k+d|k} + (y_{k|k-d} - y_k), \tag{14}$$

where $y_{k+d|k}$ is the predicted value of $y_{k+d}$ using the information (i.e., the process state) available at time $k$ in an open-loop manner, $y_k$ is the process output at time $k$ and $y_{k|k-d}$ is the predicted process output for time $k$ using the information available at time $k-d$. Thus, the state of the process at time $k$ will be

$$\mathbf{x}_k = [\hat{y}_{k+d-1}\ \ \hat{y}_{k+d-2}\ \ \Delta u_{k-1}\ \ \Delta u_{k-2}\ \ \Delta u_{k-3}]^{\mathrm{T}}.$$

Reference tracking is needed in the experiments presented here. The algorithm presented in Section 4 assumes a constant reference equal to zero. However, the extension to nonzero references is straightforward. The only modification needed is to augment the state of the system with reference $r_k$ which we considered constant for the whole prediction horizon (see Bemporad et al., 2002). The augmented state vector is

$$\mathbf{z}_k = [\mathbf{x}_k^{\mathrm{T}}\ \ r_k]^{\mathrm{T}}.$$

The weight of the output error has been modified in this experiment with a parameter $\alpha \leqslant 1$. The cost function is then defined as

$$J = \sum_{j=1}^{N} \alpha^j (y_{k+j} - r_{k+j})^2 + \lambda \sum_{j=1}^{N} (\Delta u_{k+j-1})^2.$$

Using the algorithm of Section 5, the explicit form of the controller was computed for each tested configuration of the controller parameters. The explicit form of the controller (omitted due to lack of space) defined by (7) is a set of rules as the following:

If

$$\begin{bmatrix} 1.501 & 0.726 & 78.847 & -34.087 & -28.186 & -16.573 \\ 0.023 & 0.011 & 1.155 & -0.530 & -0.426 & -0.198 \\ 0.020 & 0.010 & 1.247 & -0.443 & -0.400 & -0.404 \\ -0.057 & -0.029 & -3.520 & 1.232 & 1.144 & 1.144 \\ 0.057 & 0.029 & 3.520 & -1.232 & -1.144 & -1.144 \\ -0.020 & -0.010 & -1.247 & 0.443 & 0.400 & 0.404 \\ -0.023 & -0.011 & -1.155 & 0.530 & 0.426 & 0.198 \\ -1.501 & -0.726 & -78.847 & 34.087 & 28.186 & 16.573 \end{bmatrix} \mathbf{z}_k \leqslant \begin{bmatrix} 4.264 \\ 0.232 \\ 3.029 \\ 3.280 \\ 3.280 \\ 3.029 \\ 0.232 \\ 4.264 \end{bmatrix} \tag{15}$$

then

$$\Delta u_k = [-0.005\ \ -0.002\ \ -0.273\ \ 0.134\ \ 0.104\ \ 0.034]\mathbf{z}_k + 0.$$

Note that the number of constraints in each rule is not limited to that of (15), usually being higher.

In the experiments presented in this work the number of regions that laid in the feasible set of the state space was always between 3 and 6.

Many experiments were carried out with the PT-326 controlled by the explicit MMMPC controller. Fig. 3(a) shows a selection of these experiments for different controller parameters. Fig. 3(b) shows what we consider a reasonably good tuning ($N = 16, \lambda = 10, \alpha = 0.9$).

It can be seen from the control signal plot, that the control is somewhat aggressive, making more changes to the voltage applied to the heater than would seem to be necessary. The explanation for this behavior is that the control law is obtained using open-loop predictions, that is, they do not take into account that the control law is applied in a closed-loop manner. The solution is to use closed-loop predictions, that is, a closed-loop or feedback MMMPC. Examples of this kind of MMMPC can be found in Lee and Yu (1997); Scokaert and Mayne (1998); Bemporad, Borrelli, and Morari (2001).

### 6.2. Results of the MMMPC controller with linear feedback

The computational burden of feedback MMMPC strategies is so high that they are too difficult to apply in real time (when the cost function is based on 1 or $\infty$ norm a piecewise linear description of the controller can be found in Bemporad et al., 2001; Kerrigan & Maciejowski, 2004, but this cannot be applied to quadratic cost functions). Another way of introducing some feedback into the predictions is to use a linear feedback law as a pre-control, i.e. the control sequence will be computed as

$$\Delta u_k = K\mathbf{z}_k + v_k, \tag{16}$$

where $v_k$ is the control correction effort, which is computed with an MMMPC controller. This strategy allows us to reduce the effect of the uncertainty without increasing the complexity of the problem (see Lee & Yu, 1997; Bemporad, 1998). Fig. 4 shows an example of the predicted uncertainty bands for a simple first order system, with and without a linear feedback.

As in Section 6.1, different settings of the controller parameters have been tested in many experiments. The following stabilizing pole placement feedback gain was used as the inner control:

$$K = [-0.0052\ \ -0.0026\ \ -0.3036\ \ 0.1138\ \ 0.1016\ \ 0.0882]. \tag{17}$$

Fig. 5(a) shows a series of these experiments. It should be noted that tuning proved to be a major issue here, with the additional problem of the choice of the inner feedback gain. Fig. 5(b) shows what we consider to be a reasonably good tuning ($N = 8, \lambda = 20, \alpha = 1$). The control is smoother than that produced when no feedback is used in the predictions.

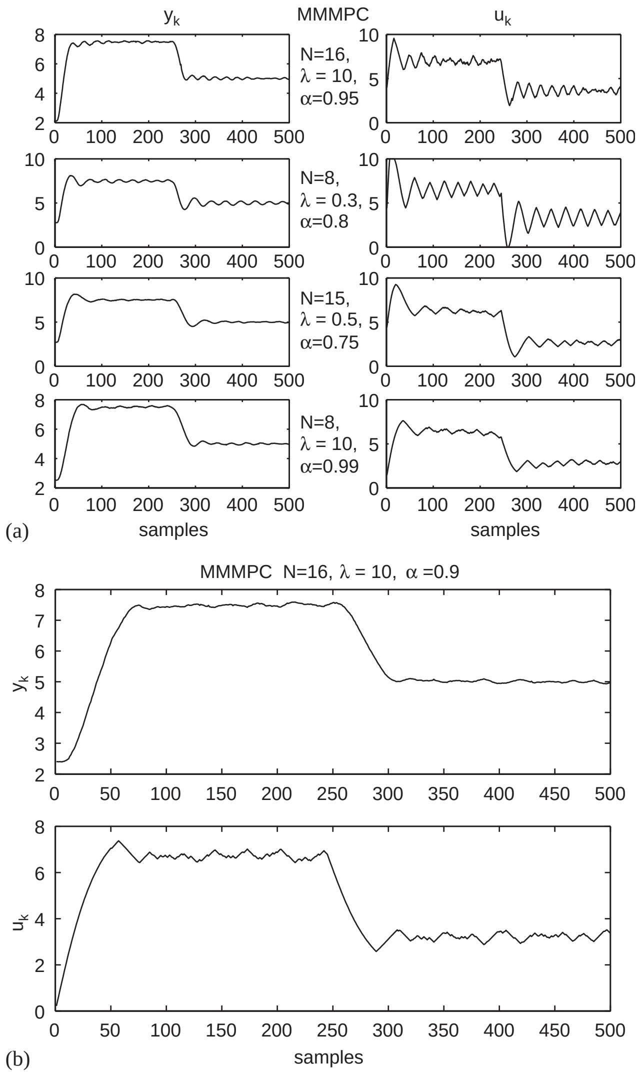


Fig. 3. Output and control signal values for different experiments with an MMMPC controller.

## 6.3. Comparisons

To test robustness, the inlet throttle of the PT-326 has been set at different positions to the nominal one. In this way the dynamics of the system changes and the robustness properties of the controller are stressed. The same series of experiments have been repeated for an MPC controller and the MMMPC controllers with the parameters of Figs. 3(b) and 5(b). The output obtained each time has been plotted in a graphic to see the overall performance of the controllers.

The MPC controller was tuned to obtain a good control under nominal conditions, better indeed than both MMMPC controllers. The parameters used in the MPC were: $N = 16$, $\lambda = 0.5$ and $\alpha = 0.75$. The output of the MPC controller is shown in Fig. 6(a). Fig. 6(b) shows the output with the MMMPC controller. It can

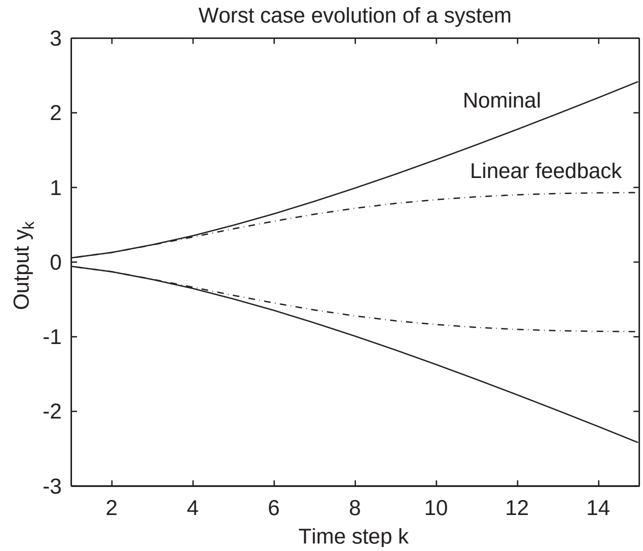


Fig. 4. Uncertainty bands with and without linear feedback.

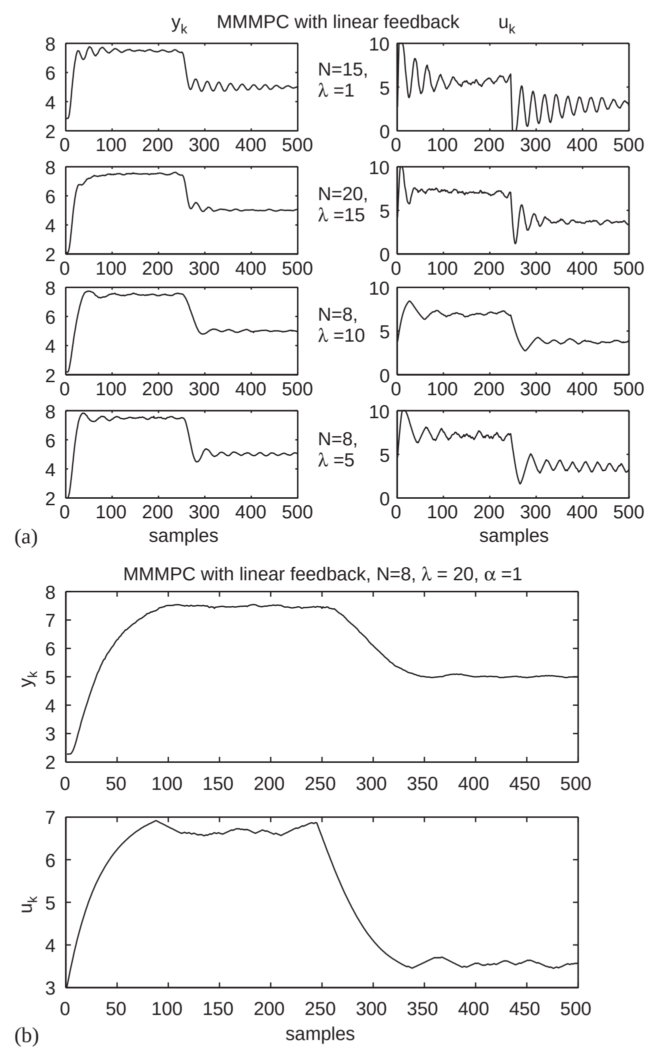


Fig. 5. Output and control signal values for different experiments with an MMMPC with a linear feedback controller.

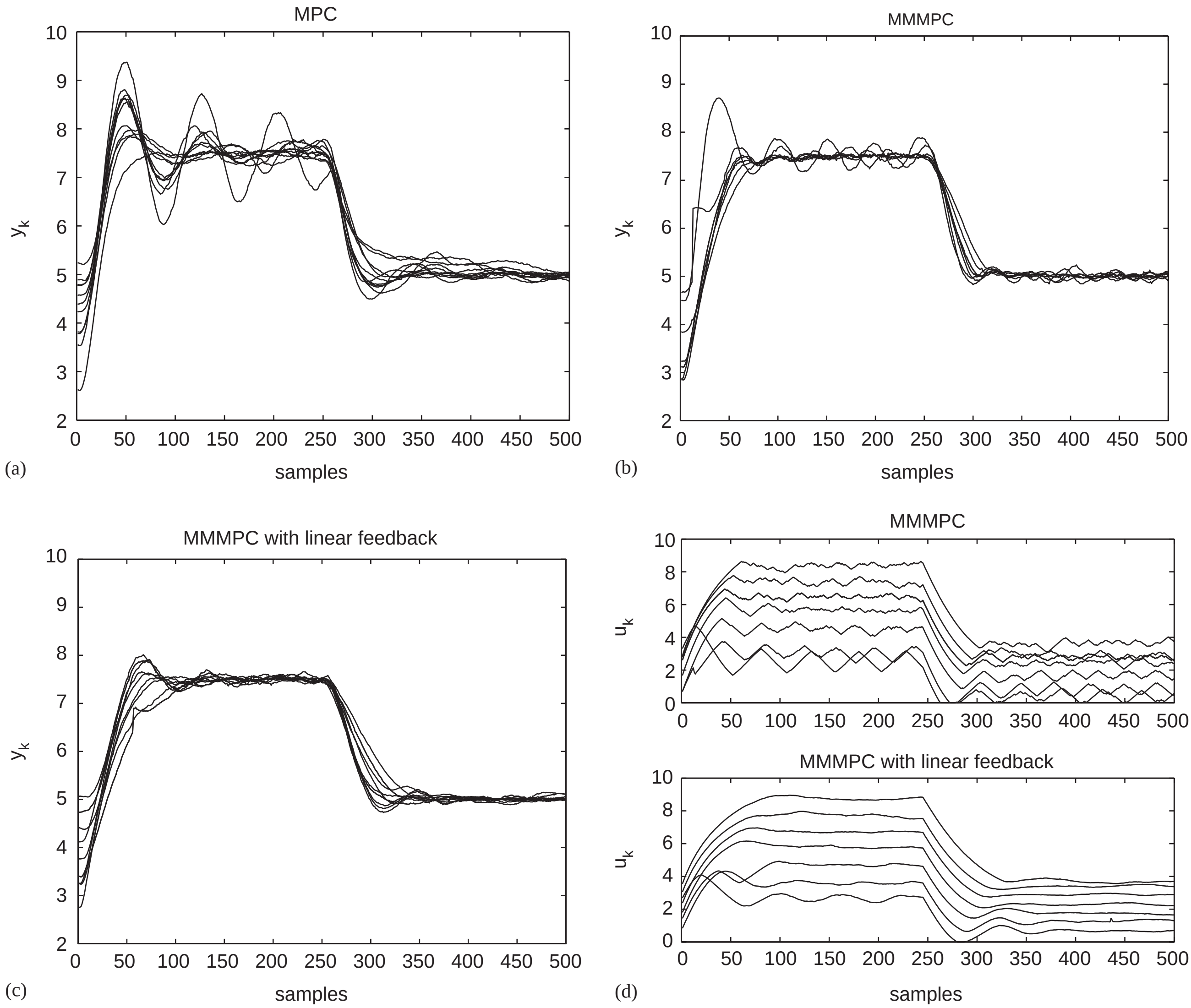


Fig. 6. Output and control signal values for experiments with different positions of the inlet throttle from 20° to 100°. (a) MPC, (b) MMMPC, (c) MMMPC with linear feedback, (d) Control signal values for experiments of Figs. 6(b) and (c).

be seen that the MMMPC controller produces a better average control when the process dynamics change in a wide range of model parameters.

Fig. 6(c) shows the output when the MMMPC with linear feedback controller is used. The output with this controller shows that its performance is better than the MMMPC controller. Furthermore, as can be seen in Fig. 6(d), the control is smoother than that produced in the MMMPC controller.

It can be seen in all these experiments that the initial state varies. This is due to the great sensitivity of the systems dynamics to the room temperature and other noncontrolled factors.

Although constraints are not explicitly considered in the controller, the region of the state space where the controller satisfies a set of linear constraints at the input and at the output can always be evaluated. Input constraints can be taken into account with the weight $\lambda$ on the cost function.

## 7. Conclusions

An algorithm to obtain the explicit form of the MMMPC control law for linear unconstrained systems has been presented. An application to a process with fast dynamics has also been shown.

The experimental results show that the min-max strategy produces more robust controllers than the nominal MPC one, that is, less sensitive to system dynamics changes. Although the MPC had a better response with the nominal system, the close loop performance became much worse when the inlet throttle was changed.

A pre-control has been used to improve the MMMPC control law. This inexpensive strategy, in terms of the computational burden, proves to be a good compromise between the more computationally expensive feedback MMMPC and the more conservative open-loop MMMPC.

The way of implementing MMMPC presented in this paper broadens the range of processes to which, in practice, this controller can be applied and shows that the min max strategy provides good robustness results for the aerothermal system.

## References


Alamo, T., Muñoz de la Peña, D., Limon, D., & Camacho, E. F. (2003). Constrained min-max predictive control: A polynomial time approach. In *Proceedings of the 42nd conference on decision and control*, Maui, Hawaii, USA.

Bandyopadhyay, R., & Patranabis, D. (2001). A new autotuning algorithm for pid controllers using dead-beat format. *ISA Transactions*, *40*, 255–266.

Bazaraa, M., & Shetty, C. (1979). *Nonlinear programming. Theory and algorithms*. New York: Wiley.

Bemporad, A. (1998). Reducing conservativeness in predictive control of constrained systems with disturbances. In *Proceedings of the 37th IEEE conference on decision and control* (pp. 1384–1389).

Bemporad, A., Borrelli, F., & Morari, M. (2001). Robust model predictive control: Piecewise linear explicit solution. In *Proceedings of the European control conference*, *ECC'01*.

Bemporad, A., Borrelli, F., & Morari, M. (2003). min-max control of constrained uncertain discrete-time linear systems. *IEEE Transactions on Automatic Control, 48*(9), 1600–1606.

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

Borrelli, F. (2002). *Discrete time constrained optimal control*. Ph.D. thesis, Swiss Federal Institute of Technology (ETH).

Camacho, E. F., & Bordóns, C. (1999). *Model predictive control*. Berlin: Springer.

Campo, P., & Morari, M. (1987). Robust model predictive control. In: *Proceedings of American control conference* (pp. 1021–1026).

Horácek, P. (2000). Laboratory experiments for control theory courses: A survey. *Annual Reviews in Control*, *24*, 151–162.

Kerrigan, E., & Maciejowski, J. (2004). Feedback min-max model predictive control using a single linear program: robust stability and the explicit solution. *International Journal of Robust and Nonlinear Control*, *14*, 395–413.

Kim, Y., & Kwon, W. (1998). An application of min-max generalized predictive control to sintering processes. *Control Engineering Practice*, *6*, 999–1007.

Lee, J., & Yu, Z. (1997). Worst-case formulations of model predictive control for systems with bounded parameters. *Automatica*, *33*(5), 763–781.

Li, G., Tsang, K., & Ho, S. (1998). Fuzzy based variable step approaching digital control for plants with time delay. *ISA Transactions*, *37*, 167–176.

Ljung, L. (1987). *System identification, theory for the user*. Englewood Cliffs, NJ: Prentice-Hall.

Porfirio, C., Almeida, E. F., & Odloak, D.F. (2002). Multi-model predictive control of an industrial c3/c4 splitter. *Control Engineering Practice 15*, 765–779.

Ramírez, D., & Camacho, E.F. (2001). On the piecewise linear nature of min-max model predictive control with bounded uncertainties. In: *Proceedings of 40th conference on decision and control*, *CDC'2001*.

Ramírez, D., & Camacho, E. F. (2002). Characterization of min-max MPC with GLOBALn uncertainties. In: *Proceedings of American control conference*, *ACC*.

Scokaert, P., & Mayne, D. (1998). min-max feedback model predictive control for constrained linear systems. *IEEE Transactions on Automatic Control*, *43*(8), 1136–1142.

Seron, M., Goodwin, G., & De Doná, J. (2002). Finitely parameterised implementation of receding horizon control for constrained linear systems. In *American control conference*, Anchorage, AK (pp. 4481–4485).

Tøndel, P., Johansen, T., & Bemporad, A. (2001). An algorithm for multi-parametric quadratic programming and explicit MPC solutions. In *Proceedings of 40th conference on decision and control*, *CDC'2001*.

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

Witsenhausen, H. S. (1968). A minimax control problem for sampled linear systems. *IEEE Transactions on Automatic Control*, *13*(1), 5–21.


## Further Reading


Mayne, D., & Rakovic, S. (2002). Optimal control of constrained piecewise affine discrete time systems using reverse transformation. In *Conference on decision and control*, Las Vegas, Nevada USA (pp. 1546–1551).