Available online at www.sciencedirect.com

SCIENCE DIRECT®

Control Engineering Practice 12 (2004) 1197–1205

CONTROL ENGINEERING PRACTICE

www.elsevier.com/locate/conengprac

Implementation of min–max MPC using hinging hyperplanes. Application to a heat exchanger☆

D.R. Ramírez*, E.F. Camacho, M.R. Arahal

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

Received 17 January 2003; accepted 23 December 2003

Abstract

Min–max model predictive control (MMMPC) is one of the few control techniques able to cope with modelling errors or uncertainties in an explicit manner. The implementation of MMMPC suffers a large computational burden due to the numerical min–max problem that has to be solved at every sampling time. This fact severely limits the range of processes to which this control structure can be applied. An implementation scheme based on hinging hyperplanes that overcome these problems is presented here. Experimental results obtained when applying the controller to the heat exchanger of a pilot plant are given.

Keywords: Predictive control; Minimax techniques; Hinging hyperplanes; Uncertain linear systems; Process control

1. Introduction

Model predictive control (MPC) is one of the few control techniques able to cope with modelling errors or uncertainties in an explicit manner. There are many ways of considering modelling errors: in some cases the process dynamics are considered to be described by equations of known structures and orders with uncertain parameters. In other cases, the frequency response of the process is considered to be uncertain (although the process is supposed to be linear). The approach used here is to consider that uncertainties affect the 1-step ahead prediction model in a global manner (Camacho & Bordóns, 1999). The only consideration made is that the 1-step ahead prediction error of the model is bounded. No supposition is made with regard to the real structure and order of equations describing process dynamics.

One of the ways of considering these types of uncertainties in the MPC context is by optimizing the objective function for the worst case situation of the uncertainties. That is, by solving a min–max model predictive control (MMMPC) problem. The solution of a min–max problem requires a substantial computing time and can only be applied in real time to processes with slow dynamics. In fact, because of this computational burden, there are very few applications of MMMPC to real processes. One of these applications can be found in (Camacho & Berenguel, 1997), where a MMMPC is used to control a solar power plant. In Kim and Kwon (1998) MMMPC is applied to a simulated complex model.

The application of MMMPC to processes with fast dynamics cannot be based on numerical methods. Some techniques have been proposed to overcome these problems. In Ramírez, Arahal, and Camacho (2001) a neural network based implementation is shown. Neural networks have proven themselves to be good for nonlinear function approximation, but there is always an approximation error. Furthermore, patterns from a large region of the process output space have to be computed.

The solution of a MMMPC with 1 or ∞-norm based cost function has been shown to be a piecewise affine function of the state. This fact was deduced because this problem can be posed as a multiparametric linear constrained LP problem (Bemporad, Borrelli, & Morari, 2003), and the solution of a multiparametric linear constrained LP problem is a piecewise linear function of the parameters (i.e. the process state). In Ramírez and

☆This work is an expanded and revised version of a paper presented at the 15th IFAC World Congress, B'02.

*Corresponding author. Tel.: +34-954-487-357; fax: +34-954-487-340.

E-mail addresses: danirr@cartuja.us.es (D.R. Ramírez), eduardo@esi.us.es (E.F. Camacho), arahal@cartuja.us.es (M.R. Arahal).

0967-0661/$ - see front matter
doi:10.1016/j.conengprac.2003.12.011

Camacho (2001) it is shown, by other means, that unconstrained MMMPC with a quadratic criterion, that cannot be posed as a linear constrained multiparametric LP problem, is a piecewise affine function of the process state. However, the number of regions needed to implement the controller is excessive for realistic control horizons. Efficient ways of solving the min–max problem have recently been developed (Ramírez, Alamo, & Camacho, 2002). These strategies involve the solution in real time of a reduced min–max problem whose solution is the same as that of the original problem. This is a good choice for processes with dynamics with time constants measured in tens of seconds or minutes, but it is not appropriate for faster dynamics.

The solution proposed in this paper is to approximate the controller by a set of hinging hyperplanes. An approximation made with hyperplanes would be a better option than the neural network approach since the controller has been shown to be piecewise affine (Ramírez & Camacho, 2001) and such an approximation could be made practically error free. Although neural networks exist based on piecewise linear sigmoidal nodes (Hush & Horne, 1998), the more general hinging hyperplanes (HH) technique is used here. The HH technique (Breiman, 1993) is a nonlinear function approximation method that uses hinge functions, i.e. hyperplanes joined together. With this technique, piecewise affine functions can be described using a basis function expansion. Hinging hyperplanes have mainly been used for modelling dynamical systems, as in (Chikkula, Lee, & Ogunnaike, 1998) and, in a more general context, piecewise linear models have been used as early as in Sontag (1981).

This paper presents a method to implement MMMPC with additive uncertainties and quadratic cost function for linear processes using closed formulae of the control law. These formulae are obtained using the HH technique over a set of patterns computed off-line by solving the min–max problem numerically.

Another difficulty arises when the process has a large deadtime. In this case, the effect of past values of the control signal have to be taken into account in the min–max optimization. This paper shows how to overcome this problem without losing the robust properties of MMMPC and with only a moderate gain of conservativeness in the control law.

A heat exchanger has been chosen to illustrate the implementation approach described in this paper. Experimental results given in the paper attest the suitability of this strategy.

The paper is organized as follows: Section 2 presents the basic MMMPC with bounded global uncertainties algorithm. Section 3 is devoted to the MMMPC with bounded global uncertainties algorithm for processes with long delays. The HH technique for nonlinear function approximation and the implementation strategy are described in Section 4. Finally, Section 5 shows the application of the MMMPC to a heat exchanger and Section 6 presents the conclusions.

2. MMMPC with bounded global uncertainties

The objective of MPC control is to compute the future control sequence $u_k, u_{k+1}, \ldots, u_{k+N-1}$ in such a way that the optimal j-step ahead predictions $y_{k+j|k}$ are driven close to the setpoint sequence $w_k, w_{k+1}, \ldots, w_{k+N-1}$ for the prediction horizon. The way in which the system will approach the desired trajectories will be indicated by a cost function J which depends on the present and future control signals and uncertainties.

When bounded uncertainties are explicitly considered, it would seem that a more robust control would be obtained if the controller tried to minimize the objective function J for the worst situation. That is, by solving the following min–max problem:

$$\min_{u} \max_{\theta \in \Theta} J(\theta, u), \tag{1}$$

where θ represents the sequence of future uncertainties, $\Theta = \{\theta / \underline{\theta} \leqslant \theta \leqslant \bar{\theta}\}$ the hypercube in which the future uncertainty sequence is contained and u is the future control sequence vector.

MMMPC is formulated either in state space (Lee & Yu, 1997) or by using input–output description. The latter is used here, since processes with large deadtime are easy to describe using transfer function representations.

The most usual form of $J(\theta, u)$ is a quadratic criterion:

$$J(\theta, u) = \sum_{j=N_1}^{N_2} (y_{k+j|k} - w_{k+j})^2 + \lambda \sum_{j=1}^{N_u} (\Delta u_{k+j-1})^2, \tag{2}$$

where $\Delta = 1 - q^{-1}$, N_1 and N_2 define the beginning and end of the cost horizon, N_u is the control horizon and $y_{k+j|k}$ is the worst-case output prediction made at time k for time $k+j$. If the process has a time delay of d sampling times, it is usual to set $N_1 = d + 1$ because the output will not be affected by the values of u until instant $d + 1$. Other types of objective functions have been used in the literature. In Campo and Morari (1987) a $\infty - \infty$ norm is used while in Allwright (1994) a $1 - \infty$ norm is proposed. Furthermore, a sort of adaptive worst-case design using min–max problems can be found in Veres and Norton (1993).

When a global uncertainties approach is used, the way of modelling the uncertainties is to assume that all modelling errors are globalized in a vector of variables that is added to the nominal prediction model. In the case of transfer function models it is common to use a linear prediction model like

$$\tilde{A}(q^{-1})y_k = q^{-d}B(q^{-1})\Delta u_{k-1} + \theta_k, \quad \theta_k \in \Theta \tag{3}$$

with $\tilde{A}(q^{-1}) = \Delta A(q^{-1})$, $\Delta = 1 - q^{-1}$, d is the process delay, θ_k the uncertainty at time k, y_k and u_k being the output and control sequence of the plant. Note that in this prediction model the error concept present in CARIMA models (commonly used in GPC) is extended to incorporate the effect of modelling uncertainties and disturbances (Camacho & Bordóns, 1999).

The prediction model (3) will be referred to as an integrated uncertainties prediction model. Using such a model, the prediction equation can be written as (Camacho & Bordóns, 1999)

$$y = G_u u + G_\theta \theta + F_x x, \tag{4}$$

where $y = [y_{k+N_1|k}, y_{k+N_1+1|k}, \ldots, y_{k+N_2|k}]^{\mathrm{T}}$, $u = [\Delta u_k, \Delta u_{k+1}, \ldots, \Delta u_{k+N_u-1}]^{\mathrm{T}}$, θ is the uncertainty vector, i.e. $\theta = [\theta_{k+N_1}, \theta_{k+2}, \ldots, \theta_{k+N_2}]^{\mathrm{T}}$ and x represents the process state in terms of the current and past outputs and past inputs, that is

$$x = [\, y_k, y_{k-1}, \ldots, y_{k-n_a}, \Delta u_{k-1}, \Delta u_{k-d-2}, \ldots, \Delta u_{k-d-n_b}]$$

with n_a and n_b the degree of $A(q^{-1})$ and $B(q^{-1})$, respectively. Matrices G_u and G_θ are lower triangular matrices that can be computed using the shifted terms of the step response of the process and the impulse response of the system $\tilde{A}(q^{-1})y_k = \theta_k$, respectively. Note that these matrices can also be computed using other methods such as the diophantine equation procedure used in the well known GPC controller (Clarke, Mohtadi, & Tuffs, 1987; Camacho & Bordóns, 1999). Furthermore, the term $F_x x$ represents the free response of the system that can also be computed as in the GPC controller (Camacho & Bordóns, 1999).

The prediction equation (4) is an affine function of θ. This implies that for $\lambda > 0$, the cost function $J(\theta, u)$ is positively definite and convex on Θ. Thus, the maximizer of $J(\theta, u)$ will be reached at one of the vertices of Θ (Camacho & Bordóns, 1999). However, all vertices of Θ must be explored to find the maximum for each u considered in the outer minimization problem. As the number of vertices of Θ exponentially depends on the prediction horizon, the computational burden grows exponentially with such a horizon.

The use of an integrated uncertainties prediction model allows step disturbances to be rejected but produces a continuous growth of the uncertainty band (Camacho & Bordóns, 1999), due to the integrated nature of the uncertainties. A possible solution to this problem is to consider bounds not only on the value of the uncertainty θ_k but also on its integrated value. This is accomplished by taking into account the following constraints in the inner maximization problem:

$$\underline{\Theta} \leqslant \sum_{j=1}^{R} \theta_{k+j} \leqslant \bar{\Theta}, \quad R = N_1, \ldots, N_2. \tag{5}$$

Thus, taking into account the prediction Eq. (4) the MMMPC controller can be expressed as:

$$\begin{aligned} &\kappa_{MMMPC}(x, w) \\ &\quad = \arg \min_{u} \max_{\theta \in \Theta} (G_u u + G_\theta \theta + F_x x - w)^{\mathrm{T}} \\ &\qquad \times (G_u u + G_\theta \theta + F_x x - w) + u^{\mathrm{T}} \lambda I u, \\ &\qquad \text{s.t. } \underline{\Theta} \leqslant \sum_{j=1}^{R} \theta_{k+j} \leqslant \bar{\Theta}, \quad R = N_1, \ldots, N_2. \end{aligned} \tag{6}$$

Solving this problem in real time can be very difficult as its computational burden grows exponentially with the prediction horizon. The idea proposed in this paper is to approximate the function $\kappa_{MMMPC}(x, w)$ by hinging hyperplanes. The strategy is to compute the control law for a set of points (x_i, w_i) in the process operating region and then obtain the approximator using the hinging hyperplanes technique. This approximation technique will be described in detail in Section 4.

3. Worst case prediction of delayed output MMMPC

The process state has to be fed to the controller in order to obtain the control signal. In an input–output description the state of a process is expressed in terms of past values of the control signal which affect the current output as well as the current and past values of the output signal. Things become more complicated if the process exhibits a pure delay d. In this case, the information needed to compute the future evolution of the output over the prediction horizon includes past values of the control signal that affect the output up to the deadtime, i.e. up to $k + d$. For processes with a large deadtime (i.e. above a half of the time constant), too many past values of the control signal have to be considered when the usual sampling ratios are used. As an example, consider a first-order process with a time delay equal to the time constant sampled with a sampling ratio of 10 sampling times per time constant. In this case, to predict the future evolution of the process output the 10 previous values of the control signal would be needed; that is, the state vector would have a much higher dimension (11 instead of 1). As was pointed out in Section 1, an approximator of the control law is used in this paper. In processes with a large deadtime the number of inputs to this approximator would be great, and therefore, the task of building the approximator would be very difficult, falling into what is often called the *curse of dimensionality*. Thus, a means of avoiding this has to be developed.

The strategy used here to apply the MMMPC to processes with large deadtime splits the problem into two stages. First, it is necessary to compute an estimation of process output after deadtime. In this work, an open-loop prediction of the process output after the deadtime is computed using the nominal prediction model, so that $\hat{y}_{k+d|k}$ is obtained.

The second step is the control law calculation for the process after the deadtime (i.e. solving a min–max problem in which the prediction horizon is $d+1,\ldots,N_2$) taking into account the uncertainty in the estimation of the process output up to the dead time. That is, y_{k+d} is affected by the uncertainties θ_{k+i} for $i=1,\ldots,d$. Thus, the uncertainty affects the predicted outputs for $k+d$ and those previous values needed to compute the predictions from $N_1=d+1$ to N_2. This is an additional source of uncertainty that does not appear when no deadtime is present, as the current and past values of the process output are known. The effect of this additional uncertainty has to be taken into account in the min–max problem. In this work bounds on the value of the process output up to deadtime are computed using all possible values of the uncertainty along the deadtime, i.e. $\theta_d=[\theta_{k+1},\ldots,\theta_{k+d}]^{\mathrm{T}}$:

$$\bar{\theta}' = \arg\max_{\theta_d\in\Theta_d}(\mathbf{y_{k+d}}-\hat{\mathbf{y}}_{\mathbf{k+d}}),$$
$$\underline{\theta}' = \arg\min_{\theta_d\in\Theta_d}(\mathbf{y_{k+d}}-\hat{\mathbf{y}}_{\mathbf{k+d}}), \tag{7}$$

where $\hat{\mathbf{y}}_{\mathbf{k+d}}=[\hat{y}_{k+d},\hat{y}_{k+d-1},\ldots,\hat{y}_{k+d-n_a}]^{\mathrm{T}}$ is the vector of nominal values of the process output up to the deadtime and $\mathbf{y_{k+d}}=[y_{k+d},y_{k+d-1},\ldots,y_{k+d-n_a}]^{\mathrm{T}}$ the vector of uncertain output values. These bounds are used to introduce the additional source of uncertainty in the min–max problem to be solved:

$$\min_{u}\max_{\theta,\theta'} J(u,\theta,\theta'), \tag{8}$$

where $\underline{\theta}'\leqslant\theta'\leqslant\bar{\theta}'$ is the additional uncertainty that affects the output prediction up to the deadtime. In this way, the prediction equation needs to be changed to incorporate this additional source of uncertainty.

$$y = G_u u + G_\theta\theta + F_x x + F_\theta\theta', \tag{9}$$

where F_θ is a matrix built with the columns of F_x associated to $y_k,\ldots,y_{k-n_a}$.

The prediction equation is an affine function of θ', as well as θ. This implies that the maximum of J for θ' will be reached at one of the vertices of the hypercube $\Theta'=\{\forall\theta' : \underline{\theta}'\leqslant\theta'\leqslant\bar{\theta}'\}$. Taking into account the prediction equation (9), the controller can be expressed as

$$\begin{aligned}\kappa_{MMMPC}(x,w) &= \arg\min_{u}\max_{\theta\in\Theta,\theta'\in\Theta'}(G_u u+G_\theta\theta+F_x x+F_\theta\theta'-w)^{\mathrm{T}}\\ &\quad\times(G_u u+G_\theta\theta+F_x x+F_\theta\theta'-w)+u^{\mathrm{T}}\lambda I u\\ &\quad \text{s.t. } \underline{\Theta}\leqslant\sum_{j=1}^{R}\theta_{k+j}\leqslant\bar{\Theta}\quad R=N_1,\ldots,N_2.\end{aligned} \tag{10}$$

The application of this controller to a heat exchanger will be shown in Section 5.

This strategy is more conservative than a traditional MMMPC in the sense that only the maximum and minimum values for y_{k+d} are considered as a starting point to the mm optimization. To illustrate this point, consider a process described by a first order model (i.e. the state is fully characterized by $[y_k,y_{k-1}]$ when a prediction model like (3) is used) with a deadtime of $d=6$. In a traditional MMMPC there are 32 different possible values for y_{k+d-1} and 64 for y_{k+d}, which will be used as the starting point for the rest of the optimization. With the strategy presented here only two possible values for each of them are considered, which yields 4 combinations of the most extreme values of y_{k+d-1} and y_{k+d}. In addition, simulation and experimental tests show that two of the combinations ($\{\underline{\theta}'_{k+d-1},\bar{\theta}'_{k+d}\}$ and $\{\bar{\theta}'_{k+d-1},\underline{\theta}'_{k+d}\}$) are unrealistic. Finally, two sources of uncertainty are considered in the prediction of the y_{k+d+1} and y_{k+d+2}, i.e. the global uncertainty θ_k and the bounds for y_{k+d-1} and y_{k+d}.

To avoid this excess of conservatism only the combinations $\{\underline{\theta}'_{k+d-1},\underline{\theta}'_{k+d}\}$ and $\{\bar{\theta}'_{k+d-1},\bar{\theta}'_{k+d}\}$ are considered in the optimization which has the added benefit of reducing the amount of vertices to be considered to compute the max part of the min–max problem. For higher order models only the combinations $\{\underline{\theta}'_{k+d-n_a},\ldots,\underline{\theta}'_{k+d}\}$ and $\{\bar{\theta}'_{k+d-n_a},\ldots,\bar{\theta}'_{k+d}\}$ should be used. This has proved to be a realistic assumption in the tests carried out.

The amount of time required to compute the control sequence can be further reduced by considering that for linear prediction models the superposition principle holds, hence the part of the output prediction due to the uncertainty θ_k can be separated from the part due to control actions. The bound values $\underline{\theta}'$ and $\bar{\theta}'$ are independent of the values of the output and also the past and future control actions. These considerations imply that the bounds for y_{k+d} have to be computed only once.

4. Hinging hyperplanes implementation

As mentioned in Section 2 the implementation strategy proposed in this paper is to approximate the control law, that is $\kappa_{MMMPC}(x,w)$, by hinging hyperplanes. Such an approximation is computed in an off-line manner. Once the HH approximation is computed, the control law can be obtained in real time without many computational requirements. To fit the approximator, the control law has to be computed off-line for a significant number of points in the process state and setpoint space. These points in (x, w) and their corresponding control action form patterns that are fed to the approximation algorithm. Then the HH algorithm, which will be described later in this section, produces an approximator of the control law that is used in real time to obtain the control signal to be applied to the process. Different strategies can be followed in order to provide setpoint tracking with the computed approximator. The setpoint can be considered as an extra input to the approximator, therefore a single approximator can be computed for a range of setpoint values. This is very close to considering the setpoint as part of an augmented state vector as in (Bemporad, Morari, Dua, & Pistikopoulos, 2002). Adding such an extra input to the approximator involves more complexity in the task of generating the necessary patterns to fit the approximator. Although computed off-line, such a task can be time consuming. Moreover, the task of computing the approximator itself can be more difficult. Taking into account that many processes only work with fixed values of the setpoint it makes sense to compute a separate approximator for every setpoint value. Thus, less patterns are necessary to obtain the approximator and the approximation error is reduced. A different procedure can be used when no constraints over the predicted output are considered in the outer minimization problem. In this case, a single approximator can be computed for a single setpoint value. When operating the plant, the deviation of the actual setpoint from the training setpoint is added to the inputs of the approximator related to process output values. This approach has the drawback, however, of having to fit the approximator over a large process state space, thus needing more patterns and, possibly, more hinges. In the following section, the HH approximation technique is discussed.

The *hinging hyperplane* method is a nonlinear function approximation technique introduced by Breiman (1993), which is described as a basis of function expansion

$$f(x) = \sum_{i=1}^{K} h_i(\varphi_i, x), \tag{11}$$

where h_i are called the hinge functions. In this work, $f(x)$ is the control law (10) and x is the process state. A hinge function consists of two hyperplanes joined together. These two hyperplanes are given by

$$h^+ = \varphi^+ x, \quad h^- = \varphi^- x, \tag{12}$$

where $x = [1, x_1, \ldots, x_m]$. The hyperplanes are joined at $(\varphi^+ - \varphi^-)x = 0$. The hinge is defined by $\varphi^+ - \varphi^-$ or its multiples. The hinge function is given by

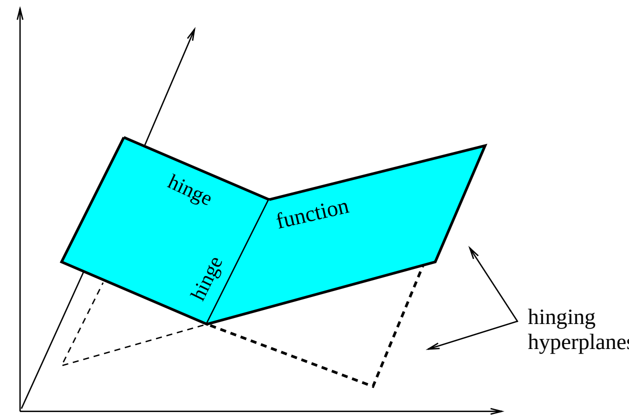

Fig. 1. Two hinging hyperplanes. The hinge function (of the "max" kind) and the hinge are labeled.

$$h = \max(h^+, h^-) \text{ or } h = \min(h^+, h^-). \tag{13}$$

The correct form is that which gives the smallest approximation error when the parameters, φ^+ and φ^-, are estimated. Fig. 1 illustrates the elements of the HH. The HH technique has advantages over other nonlinear approximation methods such as neural networks, e.g. the existence of an upper bound of the approximation error (Breiman, 1993). Also, the function parameters are estimated using a fast and efficient least squares algorithm. Furthermore, the approximation by piecewise affine functions allows for the application of existing analysis techniques.

In the HH approach there are two key elements: the *hinge finding algorithm* (HFA) and the *HH algorithm*. In Breiman (1993) and Pucar, Predrag, and Sjöberg (1998), and Docampo and Baldomir (1997) different types of algorithms are presented. In this work, the original algorithms presented in Breiman (1993) are used. Short descriptions of these algorithms are given in the following section.

4.1. Hinge finding algorithm

The initial data are N regression pairs $\{y_i, x_i\}$. The algorithm takes the following steps:

(1) Choose an initial partition of $\{y_i, x_i\}$ into two sets named S^+ and S^- with approximately half the data of $\{y_i, x_i\}$ in each set.
(2) Compute φ^+ as the parameter vector of the hyperplane with the best fit in the least-squares sense. Also, compute φ^- from S^- in the same way.
(3) Update S^+ and S^- according to $S^+ = \{x_i : \varphi x_i > 0\}$ and $S^- = \{x_i : \varphi x_i \leqslant 0\}$.
(4) Repeat from step 2, until the convergence criterion is satisfied. In this work the decrease in the squared sum of errors is used.

4.2. The HH algorithm

More than one hinge is usually necessary to approximate a function. The HH algorithm allows as many hinges to be used as necessary to approximate a function. Here, the refitting method (Breiman, 1993) is used:

(1) For $K = 1$, find $h_1(x)$ with the HFA.
(2) For $K > 1$:
  (a) Run the HFA over the residuals $f(x) - \sum_{k=1}^{K-1} h_k(x)$ to find $h_K(x)$. Add $h_K(x)$ to the set of computed hinges.
  (b) Refit all the computed hinges according to the following scheme:
    (i) refit $h_1(x)$ using $f(x) - \sum_{k=2}^{K} h_k(x)$
    (ii) for $j \leqslant K$ refit $h_j(x)$ using $f(x) - \sum_{k=1,k\neq j}^{K} h_k(x)$
  (c) Repeat from step (b), until the convergence criterion is satisfied. In this case until $f(x) - \sum_{k=1}^{K} h_k(x)$ ceases to significantly decrease.

5. Application to a heat exchanger

The controller presented in Sections 2 and 3 has been applied to the heat exchanger of a pilot plant. A photograph and a diagram of the pilot plant showing its main elements as well as the localization of the various instruments are given in Fig. 2.

The main elements are:

- *Feed circuit*. The pilot plant has two input pipes, a cold water one (whose temperature is regulated by a cooling plant) and a hot water one (at about 70°C) with nominal flow and pressure conditions of 10 l/min and 2 bar for the cold water and 5 l/min and 1 bar for the hot. The temperatures and the flows of the inputs are measured by thermocouples and orifice plates respectively, with controlled pneumatic valves for regulating the input flows.
- *Tank*. This has a height of 1 m and an interior diameter of 20 cm, it is thermally insulated and has an approximate volume of 31 l. It can work both when pressurized (up to a limit of 4 bar) or at atmospheric pressure, depending on the position of the vent valve. In its interior there is a 15 kW electric resistor for heating, as well as an overflow pipe, an output pipe and a pipe for recirculating the water through the exchanger.
- *Recirculation circuit*. The hot water in the tank can be cooled by the cold water entering through the cooling circuit. This circuit is composed of a centrifugal pump that circulates the hot water from the bottom of the tank through a tube bundle heat exchanger returning at a lower temperature at the top.

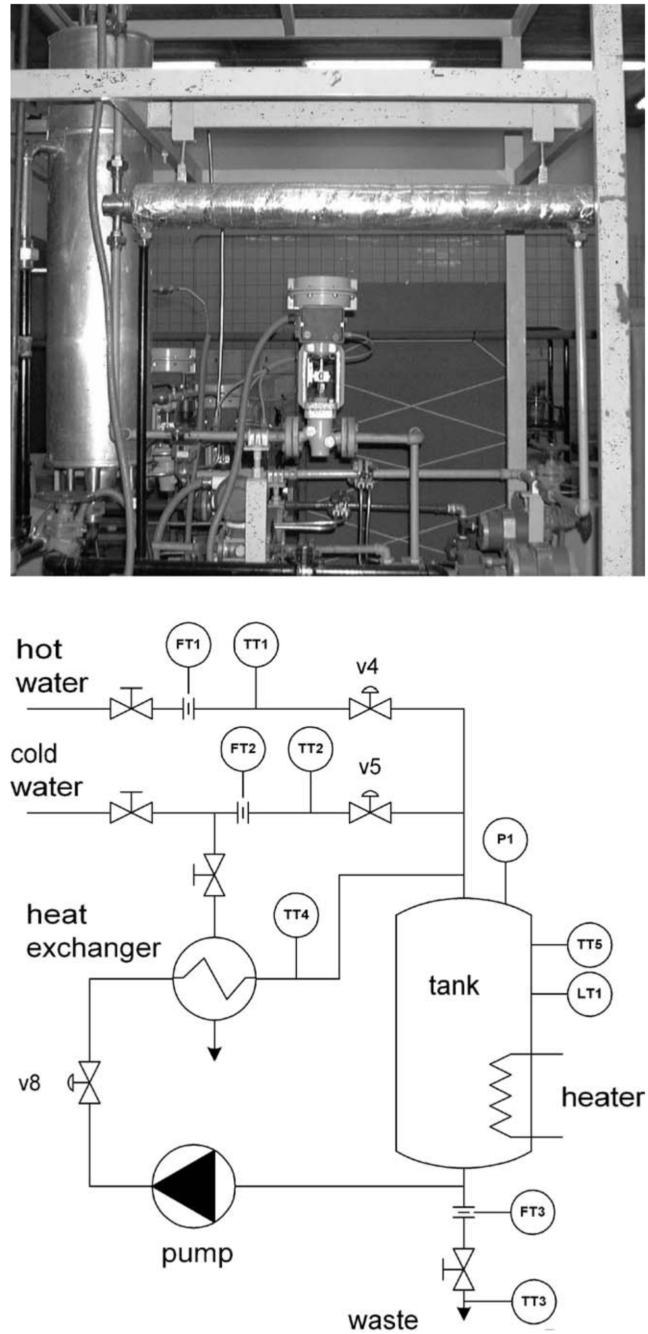

Fig. 2. Photograph (showing the tank, heat exchanger and valve V8) and diagram of the Pilot Plant.

The pilot plant control elements are connected to a PMC10 unit operated under the ORSI CUBE control software. The PMC10 architecture allows the implementation of control algorithms programmed in a PC using the ITER II language. However, the execution time is restricted to 100 ms and taking into account the fact that PMC10 CPU is an old Intel 8086 it is clear that the min–max problem cannot be solved numerically in the PMC10. Therefore, it is a suitable scenario for the HH implementation described in this paper.

A first-order linear model for the transfer function from V8 to TT4 has been obtained by step response. The initial conditions were V8 = 50%, TT4 = 31.73°C, TT5 controlled by a PID around 50°C, a constant tank level of 76.8% and the setpoint for the cooling plant was set in order to keep TT2 around 23.3°C. A step in the aperture of valve V8 from 50% to 70% yields the following model:

$$G(s) = \frac{0.135}{6s+1} e^{-6s}. \tag{14}$$

The MMMPC controllers with bounded additive uncertainty require the determination of uncertainty bounds. These bounds were obtained computing the one step ahead prediction error of the model from extensive experimental data. In the application presented in this paper the model obtained from (14) with a sampling time of 1 s was used (thus, $d = 6$). Many experiments have been carried out to find the uncertainty bounds. The error was always bounded by -0.2°C and 0.2°C. Thus the uncertainty bounds can be set to these values (that is, $\bar{\theta} = 0.2, \underline{\theta} = -0.2$). Furthermore, the bounds for the integrated uncertainty can be selected according to the shape of the uncertainty bands. When the prediction horizon corresponds to a time constant of the process dynamics it may be better to leave constraints (5) out of the optimization procedure. However, if greater horizons are used, constraints (5) have to be taken into account, as the continuous growth of the uncertainty band may affect the control adversely. The reason for this is that the final part of the prediction horizon will be affected by a very unrealistic predicted uncertainty that, in addition, will have a great weight in the cost function value. In the experiments presented here, the integrated uncertainty bounds used were $\underline{\Theta} = -0.6$, and $\bar{\Theta} = 0.6$, as this value proved itself to produce smooth control. The remaining parameter controllers have been chosen to be $N_u = 3$, $N_1 = 7$, $N_2 = 12$, $\lambda = 3.0$. The control law is defined as the solution of problem (10). The control hardware available does not allow the real time solution of the min–max problem, thus the HH implementation will be used.

The experiments carried out do not consider output constraints, thus a single controller can easily be computed for setpoint tracking using the algorithms from Section 4. A training set has been generated, having 1000 patterns obtained from a temperature range of 31.5°C to 37.5°C. Each pattern represent the first control move (i.e. Δu_k) of the optimal sequence of future control moves for a given process state and a set point value of 35°C. The HH algorithm is applied to the set until the squared sum of errors ceases to have any significant decrease. In the training set two hinges lead to a squared sum of errors less than 10^{-4}. Such a small amount of error means that only three hyperplanes exist in this region of the process state space, each one for a

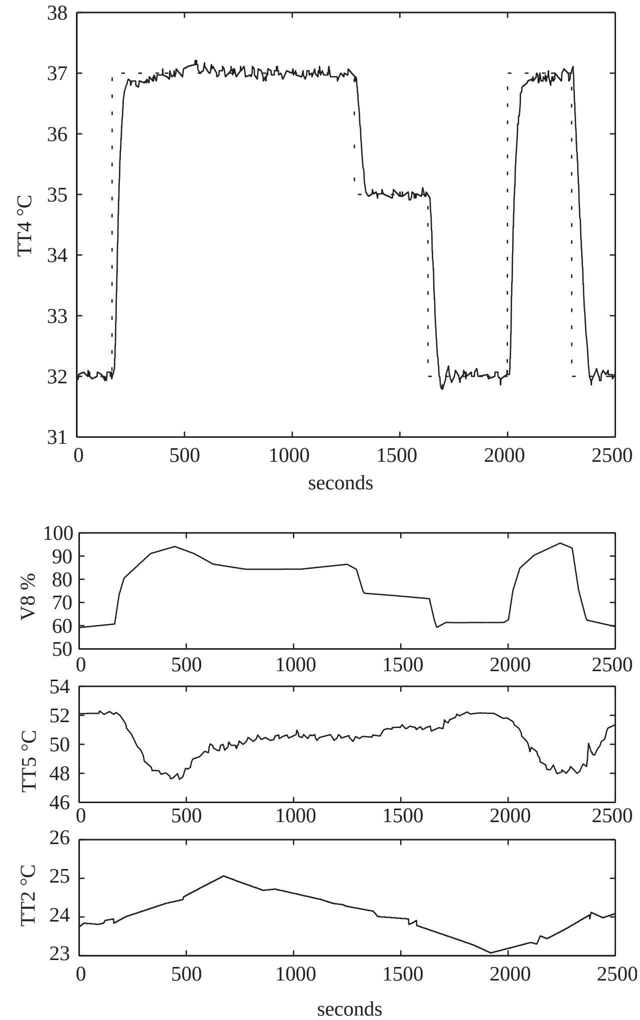

Fig. 3. Setpoint tracking experiment. Top to bottom: setpoint and heat exchanger output temperature, valve V8 aperture, tank temperature TT5 and cold water temperature TT2.

different type of solution to the min–max problem (Ramírez & Camacho, 2001). Theoretically the error should be zero, but the numerical method used in the MATLAB function *fmincon* sometimes fails to find the solution accurately.

$$\begin{aligned}\Delta u_k &= f(x) = h_1(\varphi_1, x) + h_2(\varphi_2, x),\\ &= \min\left\{ x\begin{bmatrix}14.0144\\-4.5948\\4.1988\end{bmatrix}, \quad x\begin{bmatrix}309.6598\\16.3106\\-25.1589\end{bmatrix}\right\}\\ &\quad + \max\left\{ x\begin{bmatrix}-305.9432\\-16.0135\\24.7503\end{bmatrix}, \quad x\begin{bmatrix}-9.9174\\4.8917\\-4.6074\end{bmatrix}\right\},\end{aligned} \tag{15}$$

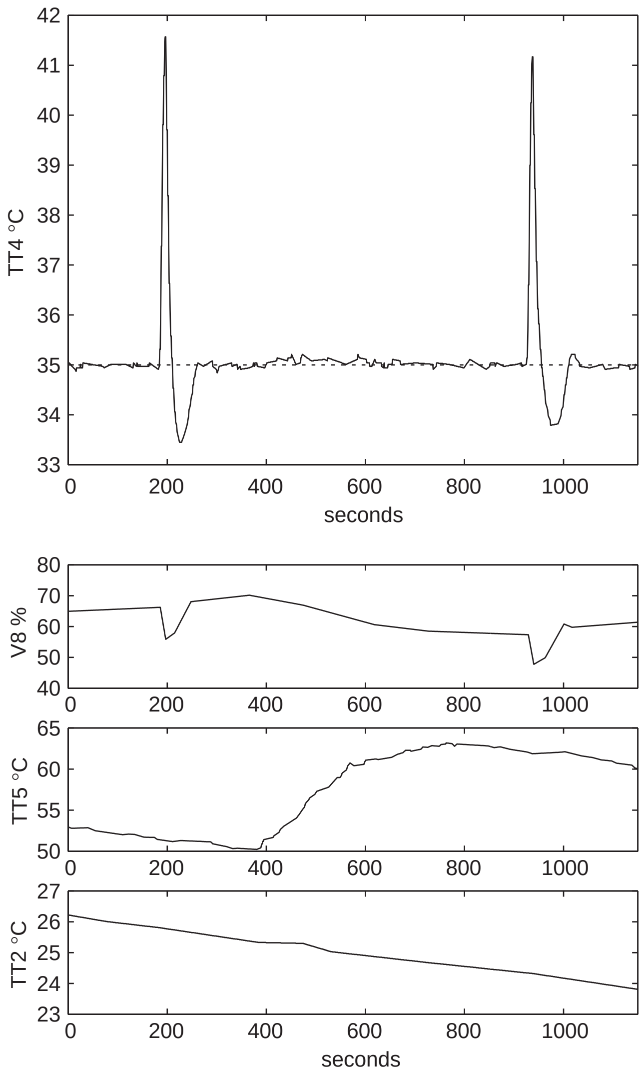

Fig. 4. Disturbance rejection experiment. Top to bottom: setpoint and heat exchanger output temperature, valve V8 aperture, tank temperature TT5 and cold water temperature TT2.

where x is the row vector

$$x = [1 \;\; y_{k-1} + (35 - w_{ref}) \;\; y_{k-1} + (35 - w_{ref})]$$

and w_{ref} is the setpoint. This expression was tested with a test set of 100 patterns obtained from a temperature range of 29−41°C and the squared sum of errors was again less than 10^{-4}. This illustrates that, as expected, the HH description obtained is a closed formula of the control law in this region of the process state space and not an approximation.

The saving of computational time using the HH implementation is very large. In the controller used in this paper, solving the min–max problem numerically takes about 174958 floating point operations (as counted by *Matlab*) and 1.7 s to obtain the solution in a 450 Mhz Pentium III computer. The HH implementation takes only 21 floating point operations and the computation time is negligible. Thus, more than 99.98% of the needed computations are moved offline. This saving is more evident when higher horizons are used:[1] for an MMMPC with $N_2 - N_1 = 10$; a numerically solved solution takes about 7.1×10^6 floating point operations (as counted by *Matlab*) and 52 s to obtain the solution in a 450 Mhz Pentium III computer. These figures can be even worse in some cases as the conditioning of the numerical problem depends on the process state. On the other hand, the HH implementation always takes a number of floating point operations measured in tens not in millions, yielding a negligible computation time.

The resulting controller was applied to the heat exchanger and some of the experimental results are depicted in Figs. 3 and 4. Fig. 3 shows a setpoint tracking experiment. It is noteworthy that the setpoint is much higher than the operating point considered for modelling the heat exchanger dynamics. The noisy output is due to the variations in the tank temperature (TT5) which is regulated by a local PI. The first setpoint change raises the V8 aperture which in turn lowers the tank temperature. This causes the V8 to be opened even more because a greater flow through the heat exchanger is required to reach the setpoint. As TT5 returns to its nominal value, V8 closes to keep the output near the setpoint.

Disturbance rejection is illustrated in Fig. 4. In this case, the manual valve of cold water was closed for 12 and 11 s, causing the temperature to greatly deviate from the setpoint. The controller reacts by closing the recirculation valve V8 to lower the temperature by having less hot water to be cooled. Re-opening the valve is another disturbance and the controller has to open the V8 valve again to bring the temperature TT4 to the desired value. Moreover, before the second disturbance a change in the setpoint of TT5 is brought about, forcing it to reach a much higher value than that existing when the model was identified. This is a slow variation of process dynamics. The controller is able to keep the output close to the setpoint as soon as the tank temperature reaches the new operating point. Meanwhile, as is expected, a small offset is observed because the uncertainty grows like a ramp and the integrated uncertainties are formulated to reject step disturbances. Finally, it is noteworthy that the disturbance rejection capability is similar to the previous case when the cold water valve was closed, even when the process dynamics have changed from the nominal values.

[1] The number of operations needed to solve the inner maximization problem increases exponentially with the prediction horizons, as the cost function must be evaluated at all the vertexes of the polytope that contain the feasible uncertainty sequences.

6. Conclusions

The paper shows how hinging hyperplanes can be used to implement Min-Max MPC. Most of the computation required is done in an off-line manner and the technique allows the implementation of MMMPC in real time.

The proposed MMMPC controller has been applied to control the outlet temperature of a heat exchanger in a pilot plant. Problems related to process delay have been tackled taking into account real time implementation requirements.

Some open questions remain to be addressed. Further improvements can be accomplished if the future values of reference are included as inputs to the HH function. This would allow a time-varying (over the prediction horizon) reference at the cost of a higher dimension function.

The inclusion of constraints on the process outputs and inputs is also a task that has to be tackled. Moreover, the HH implementation may also be suitable for other control schemes in which the control law is piecewise affine or linear, such as constrained MPC. Finally, although the process controlled here is SISO, this strategy can be used on multivariable processes provided that enough patterns are generated to cover the higher dimensional domain.

Acknowledgements

The authors acknowledge MCYT-Spain for funding this work (contracts DPI2001-2380-C02-01 and DPI2002-04375-C03-01). The authors would like to thank the reviewers for their useful comments and suggestions.

References

Allwright, J. C. (1994). *On min–max model-based predictive control. Advances in model-based predictive control*. Oxford: Oxford University Press.

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. N. (2002). The explicit linear quadratic regulator for constrained systems. *Automatica*, *38*, 3–20.

Breiman, L. (1993). Hinging hyperplanes for regression, classification and function approximation. *IEEE Transactions on Information Theory*, *39*(3), 999–1013.

Camacho, E. F., & Berenguel, M. (1997). Robust adaptive model predictive control of a solar plant with bounded uncertainties. *International Journal of Adaptive and Signal Processing*, *11*, 311–325.

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

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

Chikkula, Y., Lee, J. H., & Ogunnaike, B. (1998). Dynamic scheduled model predictive control using hinging hyperplane models. *AIChE Journal*, *44*, 1691–1724.

Clarke, D. W., Mohtadi, C., & Tuffs, P. S. (1987). Generalized predictive control-part I. The basic algorithm. *Automatica*, *23*(2), 137–148.

Docampo, D., & Baldomir, S. R. (1997). An efficient algorithm for function approximation with hinging hyperplanes. In: *Proceedings of the NSIP'97 conference*, Michigan, USA.

Hush, D. R., & Horne, B. (1998). Efficient algorithms for function approximation with piecewise linear sigmoidal networks. *IEEE Transactions on Neural Networks*, *9*(6), 1129–1141.

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

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

Pucar, Predrag, & Sjöberg, J. (1998). On the hinge finding algorithm for hinging hyperplanes. *IEEE Transactions on Information Theory*, *44*(3), 1310–1319.

Ramírez, D. R., Álamo, T., & Camacho, E. F. (2002). Efficient implementation of constrained min-max model predictive control with bounded uncertainties. In: *Proceedings of the 41st conference on decision and control*, *CDC'2002*, Las Vegas, USA, Vol. 3 (pp. 3168–3173).

Ramírez, D. R., Arahal, M. R., & Camacho, E. F. (2001). Neural network based min–max MPC. Application to a heat exchanger. In: *IFAC workshop on adaptation and learning in control and signal processing*, *ALCOSP'2001*, Cernobbio-Como, Italy (pp. 115–120).

Ramírez, D. R., & 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*, Orlando, USA, Vol. 5 (pp. 4845–4850).

Sontag, E. D. (1981). Nonlinear regulation: The piecewise linear approach. *IEEE Transactions on Automatic Control*, *26*(2), 346–358.

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