COMPUTATIONALLY EFFICIENT MIN-MAX MPC

D.R. Ramírez T. Alamo E.F. Camacho

Depto. de Ing. de Sistemas y Automática, Universidad de Sevilla,
Camino de los Descubrimientos s/n 41092 Sevilla, SPAIN
Phone: +34 954487346 Fax: +34 954487340
email: {danirr,alamo,eduardo}@cartuja.us.es

Abstract: Min-Max MPC (MMMPC) controllers (Campo and Morari, 1987) suffer from a great computational burden that is often circumvented by using upper bounds of the worst possible case of a performance index. These upper bounds are usually computed by means of LMI techniques. In this paper a more efficient approach is shown. This paper proposes a computationally efficient MMMPC control strategy in which the worst case cost is approximated by an upper bound which can be easily computed using simple matrix operations. This implies that the algorithm can be coded easily even in non mathematical oriented programming languages such as those found in industrial embedded control hardware. Simulation examples are given in the paper. *Copyright © 2005 IFAC.*

Keywords: Predictive control, Minimax techniques,Uncertain linear systems.

1. INTRODUCTION

In Min-Max MPC controllers, the value of the control signal to be applied is found by minimizing the worst case of a performance index (usually quadratic) which is in turn computed by maximizing over the possible expected values of disturbances and uncertainty. Solving these problems can be very time consuming as they are of the NP-hard kind. Thus, the implementation of this type of control is severely compromised leading to a lack of experimental results. Often this issue is solved by using a bound of the worst case cost instead of computing it explicitly (Alamo *et al.*, 2003). The upper bound can be efficiently computed by using LMI techniques such as in (Kothare *et al.*, 1996; Casavola *et al.*, 2000; Lu and Arkun, 2000; Wan and Kothare, 2003). The LMI problems have a computational burden that cannot be neglected in certain applications, like those in which the sampling rates are measured in seconds. Moreover, the interior point methods used to solve the LMI depend on the initial solution and the time needed to converge is not always the same.

Here an efficient upper bound of a quadratic maximization problem over a hypercube is presented. It is based on simple matrix operations that can be performed very efficiently by both personal computers and dedicated embedded control hardware. Within the MMMPC context it can be used as a substitute of the worst case cost for systems in which there is no computational power available to solve on line the LMI problems.

The paper is organized as follows: section 2 presents the MMMPC controller. Section 3 presents the efficient upper bound on the quadratic maximization problem and section 4 presents a performance analysis of the proposed bound compared to the LMI bound. On the other hand, a simulation example is given in section 5. Finally, section 6 presents the conclusions.

2. MIN-MAX MPC WITH BOUNDED ADDITIVE UNCERTAINTIES

Consider the following state space model with bounded additive uncertainties (Camacho and Bordóns, 1999):

$$\begin{aligned} x(t+1) &= Ax(t) + Bu(t) + D\theta(t) \\ y(t) &= Cx(t) \end{aligned} \qquad (1)$$

with $x(t) \in \Re^{dimx}$,$u(t) \in \Re^{dimu}$, $\theta(t) \in \Theta = \{\theta \in \Re^{dim\theta} : \|\theta\|_\infty \leq \theta_m\}$, $y(t) \in \Re^{dimy}$. Consider a sequence $\mathbf{u} = [u(t) \cdots u(t+N_u-1)]^T$ of values of the control signal over a control horizon N_u and $\boldsymbol{\theta} = [\theta(t+1) \cdots \theta(t+N)]^T$ a sequence of future values of $\theta(t)$ over a prediction horizon N. Furthermore, let $J(\boldsymbol{\theta}, \mathbf{u}, x)$ be a quadratic performance index of the form:

$$\begin{aligned} J(\boldsymbol{\theta}, \mathbf{u}, x) = & \sum_{j=1}^{N} x(t+j|t)^T Q_j x(t+j|t) \\ & + \sum_{j=0}^{N_u-1} u(t+j)^T L_j u(t+j) \end{aligned} \qquad (2)$$

where $x(t+j|t)$ is the prediction of the state for $t+j$ made at t when the future values of the uncertainty are supposed to be given by the sequence $\boldsymbol{\theta}$. On the other hand $Q_j \in \Re^{dimx \times dimx}$, $L_j \in \Re^{dimu \times dimu}$ are symmetric positive definite matrices used as weighting parameters.

Min-Max MPC (Campo and Morari, 1987) is based on finding the control sequence $\mathbf{u}$ that minimizes $J(\boldsymbol{\theta}, \mathbf{u}, x)$ for the worst possible case of the predicted future evolution of the process state or output signal. This is accomplished through the solution of a min-max problem like:

$$\begin{aligned} \mathbf{u}^*(x) = \arg & \min_{\mathbf{u} \in U} \max_{\boldsymbol{\theta} \in \Theta} J(\boldsymbol{\theta}, \mathbf{u}, x) \\ & \text{s.t.} \quad R\mathbf{u} \leq c_\theta + Fx \end{aligned} \qquad (3)$$

where $\boldsymbol{\Theta} = \{\boldsymbol{\theta} \in \Re^{N \cdot dim\theta} : \|\boldsymbol{\theta}\|_\infty \leq \boldsymbol{\theta}_m\}$, and $U \subseteq \Re^{N_u \cdot dimu}$ is compact. Linear time invariant constraints can be taken into account and defined through the matrices $R \in \Re^{nc \times (N_u \cdot dimu)}$, $F \in \Re^{nc \times dimx}$ and the vector $c_\theta \in \Re^{nc}$ where nc is the number of constraints. As usual in all predictive control schemes, the solution of problem (3) is applied in a feedback manner using a receding horizon strategy. The properties of the min-max problem depend on the structure and properties of the cost function $J(\boldsymbol{\theta}, \mathbf{u}, x)$ and the prediction model. A common choice for $J(\boldsymbol{\theta}, \mathbf{u}, x)$ is a quadratic function of $\boldsymbol{\theta}$, $\mathbf{u}$ and the predictions. The results presented in this paper can be applied to LTI SISO or MIMO input-output models, convolution models or state-space models with such a quadratic cost function. All these descriptions lead to a prediction equation which is an affine function of process uncertainties, inputs and state, i.e.:

$$\boldsymbol{\xi} = G_u \mathbf{u} + G_\theta \boldsymbol{\theta} + F_x x(t) \qquad (4)$$

where $\boldsymbol{\xi} \in \Re^{N \cdot dimx}$ can be either the predictions of process state or output over the prediction horizon, $G_u \in \Re^{(N \cdot dimx) \times (N_u \cdot dimu)}$, $G_\theta \in \Re^{(N \cdot dim\theta) \times (N \cdot dim\theta)}$ are lower triangular matrices and $F_x \in \Re^{(N \cdot dimx) \times dimx}$ is the free response matrix.

Although the results presented in this paper are not valid in general for closed-loop MMMPC with a quadratic cost (see (Mayne *et al.*, 2000) and references therein), they are valid when using a semi-feedback approach in which the control input is given by $u(t) = -Kx(t) + v(t)$ where the feedback matrix K is chosen to achieve some desired property such as nominal stability or LQR optimality without constraints. The MMMPC controller will compute the optimal sequence of correction control inputs $v(t)$. Rewritting the state equation of system (1) as

$$x(t+1) = A_{CL}x(t) + Bv(t) + D\theta(t+1) \qquad (5)$$

it is clear that such semi-feedback MMMPC can be casted into problem (3) with $A_{CL} = (A - BK)$. Moreover, in (Kerrigan and Alamo, 2004), an affine parameterization of the input signal that shorten even more the gap between open and closed-loop min-max controllers is presented. The idea is to parameterize the input signal as

$$u(t) = v(t) + \sum_{j=0}^{t-1} M_{t,j}\theta(j) \qquad (6)$$

This parameterization can also be embedded in (3).

It can be seen that the cost function can be rewritten as:

$$\begin{aligned} J(\boldsymbol{\theta}, \mathbf{u}, x) = & \mathbf{u}^T M_{uu} \mathbf{u} + \boldsymbol{\theta}^T M_{\theta\theta} \boldsymbol{\theta} + 2\boldsymbol{\theta}^T M_{\theta u} \mathbf{u} \\ & + 2x^T M_{uf}^T \mathbf{u} + 2x^T M_{\theta f}^T \boldsymbol{\theta} + x^T M_{ff} x \end{aligned} \qquad (7)$$

where: $M_{uu} = G_u^T Q G_u + L$, $M_{\theta\theta} = G_\theta^T Q G_\theta$, $M_{ff} = F_x^T Q F_x$,$M_{\theta u} = G_\theta^T Q G_u$, $M_{uf} = G_u^T Q F_x$, $M_{\theta f} = G_\theta^T Q F_x$ and Q and L are symetric positive definite block diagonal matrices composed of the Q_j and L_j respectively. It can easily be seen that $M_{\theta\theta}$ is a Gram matrix and therefore at least positive semidefinite . On the other hand, M_{uu} is positive definite as $L > 0$. Note that as $M_{\theta\theta} \geq 0$ then $J(\boldsymbol{\theta}, \mathbf{u}, x)$ is convex on $\boldsymbol{\theta}$, and because $M_{uu} > 0$ strictly convex on $\mathbf{u}$. Therefore the solution of (3) will be unique (Camacho and Bordóns, 1999).

Finally, note that the stability properties of MMMPC controllers are not lost by the use of upper bounds based on the addition of a quadratic term on θ to the original functional (Alamo *et al.*, 2003).

3. UPPER BOUND FOR THE QUADRATIC MAXIMIZATION PROBLEM

In this section a procedure to compute efficiently an upper bound of the worst case cost is given. Consider the following maximization problem:

$$\gamma^* = \max_{\|\theta\|_\infty \leq 1} \theta^T Q\theta + 2\theta^T p$$

where $Q \geq 0$. Note that being the maximization of a convex function on a convex set the maximum is attained at least at one of the vertices of the hypercube described by $\Theta \triangleq \{\theta : \|\theta\|_\infty \leq 1\}$ (Bazaraa and Shetty, 1979). Thus, the problem can be rewritten as:

$$\gamma^* = \max_{\theta \in \text{vert}\{\Theta\}} \theta^T Q\theta + 2\theta^T p \qquad (8)$$

To find the value of γ^* it is necessary to evaluate the function for all the vertices of Θ. Taking into account that the number of vertices is $2^{\dim\{\theta\}}$ it is clear that the problem cannot be solved in real time for a certain dimension of θ (because this is a well known NP-hard problem).

Problem (8) is equivalent to the following augmented problem:

$$\max_{\begin{bmatrix}\theta_e \\ \theta\end{bmatrix} \in \text{vert}\{\Theta_A\}} \begin{bmatrix}\theta_e \\ \theta\end{bmatrix}^T \begin{bmatrix} 0 & p^T \\ p & Q\end{bmatrix} \begin{bmatrix}\theta_e \\ \theta\end{bmatrix} \qquad (9)$$

where Θ_A is the augmented unitary hypercube. In effect, suppose that the maximum of the augmented problem (9) is attained at $\theta_e = 1$ and $\theta = \theta^*$. Then the same maximum is attained also for $\theta_e = -1$ and $\theta = -\theta^*$, thus the maximum of the augmented problem is the same as that of (8) because we can assume that $\theta_e = 1$ maximizing then $\theta^T Q\theta + 2p^T\theta$ which is the original maximization problem. The augmented problem can be rewritten as:

$$\gamma^* = \max_{z \in \text{vert}\{\Theta_A\}} z^T H z \qquad (10)$$

where $H \in \mathbb{R}^{n \times n}$. Now suppose T a diagonal matrix such that $T \geq H$, then:

$$z^T Hz \leq z^T Tz = \sum T_{ii} z_i^2 \leq trace(T)\|z\|_\infty^2 \leq trace(T)$$

thus:

$$\gamma^* \leq \text{trace}(T)$$

Therefore a conservative upper bound of γ^* can be found solving the following LMI problem:

$$\begin{aligned} \sigma^* = \min & \ \text{trace}(T) \qquad (11) \\ s.t. & \\ & T \geq H \\ & T \quad \text{diagonal} \end{aligned}$$

If $H \geq 0$ this upper bound of γ^* satisfies (Nesterov *et al.*, 2000):

$$\gamma^* \leq \sigma^* \leq \frac{\pi}{2}\gamma^*$$
$$\frac{2\sigma^*}{\pi} \leq \gamma^* \leq \sigma^*$$

This means that σ^* provides both an upper and lower bound. Moreover, the conservativeness of the bound does not depend on the dimension of H. Thus, σ^* is an appropiate bound to be used within a worst case MPC strategy. However, solving the LMI problem (11) is computationally demanding enough to pose a problem when the sampling time is small and the dimension of H is moderately high. We propose another method to find a conservative upper bound of γ^*.

The goal here is to find the smallest (i.e. minimum trace) diagonal matrix T such that $T > H$. The strategy is to obtain a diagonal matrix adding to H $n-1$ positive definite matrices of the form $v_i v_i^T$:

$$H + v_1 v_1^T + v_2 v_2^T + v_3 v_3^T + \cdots + v_{n-1} v_{n-1}^T = H_f$$

where H_f is a diagonal matrix. Thus the problem is to find v_i, $i = 1, \cdots, n-1$ such that H_f is diagonal and the conservativeness of the bound is kept as low as possible. Suppose that $H = \begin{bmatrix} a & b^T \\ b & H_r \end{bmatrix}$,$a \in \Re$ and that we want to add $v_1 v_1^T$ in such a way that:

$$\begin{bmatrix} a & b^T \\ b & H_r \end{bmatrix} + v_1 v_1^T = \begin{bmatrix} d & 0 \\ 0 & \hat{H}_r \end{bmatrix}, \qquad d \in \Re \qquad (12)$$

Once v_1 is found, the process continues by choosing v_2 such that $\hat{H}_r$ is also diagonalized and so on. If v_1 is chosen to be $\begin{bmatrix} \alpha & e^T \end{bmatrix}^T$ then $v_1 v_1^T$ becomes:

$$\begin{bmatrix} \alpha \\ e \end{bmatrix} \begin{bmatrix} \alpha & e^T \end{bmatrix} = \begin{bmatrix} \alpha^2 & \alpha e^T \\ \alpha e & ee^T \end{bmatrix}$$

with $\alpha > 0$. This implies that $\alpha e = -b$ thus $e = \frac{-b}{\alpha}$, $d = a + \alpha^2$ and $\hat{H}_r = H_r + \frac{bb^T}{\alpha^2}$.

The parameter α should be chosen to minimize the error introduced by the diagonalization in the original augmented maximization problem. This error is:

$$z^T v_1 v_1^T z = z^T \begin{bmatrix} \alpha \\ -\dfrac{b}{\alpha} \end{bmatrix} \begin{bmatrix} \alpha & -\dfrac{b^T}{\alpha} \end{bmatrix} z$$

The error is maximum when z turns out to be:

$$z^* = \text{sign} \begin{bmatrix} \alpha \\ -\dfrac{b}{\alpha} \end{bmatrix}$$

(and also when it is of opposite sign). Taking into account that

$$\begin{bmatrix} \alpha & -\dfrac{b^T}{\alpha} \end{bmatrix} z^* = \left\| \begin{matrix} \alpha \\ -\dfrac{b}{\alpha} \end{matrix} \right\|_1$$

(where $\|x\|_1$ is the 1-norm equal to the sum of the absolute values of the components of x), the maximum error is

$$\left\| \begin{matrix} \alpha \\ -\dfrac{b}{\alpha} \end{matrix} \right\|_1^2$$

The value of α that minimizes the maximum error can easily be computed by finding the value that makes the derivative of

$$\left\| \begin{matrix} \alpha \\ -\dfrac{b}{\alpha} \end{matrix} \right\|_1 = \alpha + \frac{1}{\alpha}\|b\|_1$$

equal to zero. Such value is:

$$\alpha = \sqrt{\|b\|_1} \tag{13}$$

The procedure to compute the upper bound σ_u is summarized in the following steps:

```
(1) Let $T = H \in \mathbb{R}^{n\times n}$.
(2) for $k = 1$ to $n-1$
(3)     Let $H_{sub} = [T_{ij}]$ for $i, j = k \cdots n$.
(4)     Compute $\alpha$ for $H_{sub} = \begin{bmatrix} a & b \\ b^T & H_r \end{bmatrix}$ from (13).
(5)     Make $v_k^T = \begin{bmatrix} \alpha & \frac{-b^T}{\alpha} \end{bmatrix}$.
(6)     Make $v_e^T = \begin{bmatrix} 0 & \cdots & 0 & v_k^T \end{bmatrix} \in \mathbb{R}^n$
(7)     Update $T$ by making $T = T + v_e v_e'$.
(8) endfor
(9) Compute the upper bound from $\sigma_u = \sum_{i=1}^{n} T_{ii}$.
```

Remark 1. Note that if H_{sub} of step 2 has all its elements nonnegative at a certain iteration, then a least conservative bound is computed by adding all the elements of the partly diagonalized T matrix.

4. PERFORMANCE ANALYSIS OF THE PROPOSED BOUND

Here the accuracy of the proposed upper bound is discussed . The upper bound will be tested against the LMI bound [1] and the 1-norm of the matrix H, which is itself a very rough upper bound:

$$\sigma_n = \|H\|_1 = \sum_i^N \sum_j^N |H_{ij}|$$

The 1-norm bound will be equal to the maximum when all the elements of z^* are nonnegative. In the following it will be shown that $\sigma_u \leq \sigma_n$.

Taking into account the block structure of H given in (12) the 1-norm of H can be computed as

$$\sigma_n = \|H_r\|_1 + 2\|b\|_1 + |a|$$

On the other hand, the 1-norm after a diagonalization step can be computed as

$$\sigma_{n-1} = \left\| \begin{matrix} a + \|b\|_1 & 0 \\ 0 & H_r + \dfrac{bb^T}{\|b\|_1} \end{matrix} \right\|_1 \leq |a| + \|b\|_1 + \|H_r\|_1 + \left\| \frac{bb^T}{\|b\|_1} \right\|_1$$

Taking into account that $\left\| \frac{bb^T}{\|b\|_1} \right\|_1 = \|b\|_1$ it follows that

$$\sigma_{n-1} \leq \sigma_n$$

Thus, the diagonalization scheme proposed in section 3 provides a succession of improved upper bounds.

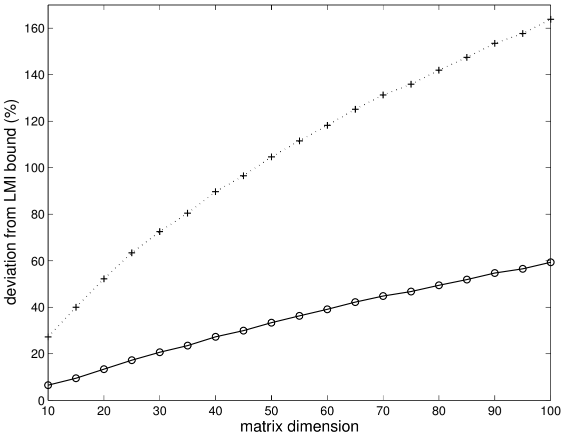

Fig. 1. Mean deviation from the LMI bound plotted as a function of matrix dimension for a group of randomly generated matrices (see text for details). 1-norm bound dotted, proposed bound solid.

Now both the proposed bound and the 1-norm will be compared to the LMI bound. Consider figure 1 in which it is plotted the mean deviation of the proposed bound (solid plot) from the LMI bound as a function of the dimension of Q. For this comparison, a group of randomly positive definite matrices where the mean value of its elements is zero have been generated [2] (200 matrices for each dimension). It can be seen that the deviation from the LMI bound grows with matrix dimension, as it can be expected from the error introduced at each diagonalization step. Even though, it is noteworthy that for much of the range needed in control applications (up to $\dim\{Q\} = 30$, which accounts for $1,073,741,824$ vertices in Θ) the deviation from the LMI bound remains under 20%. Moreover, it can be seen that the 1-norm bound is always worse (more conservative) than the proposed bound when the mean of the elements of Q are around zero.

Another interesting property is that, taking into account remark 1, the deviation of the proposed upper bound from the LMI bound depends on the mean value

[1] Additionally the proposed bound has been tested against the eigenvalue bound $n \cdot \max\{eig(H)\}$. The proposed bound has better accuracy for practical values of the prediction horizon (up to 30) and lower computational burden.

[2] These matrices are generated subtracting two uniformly distributed random matrices created using the *Matlab* **rand** function. Then, every matrix is multiplied by itself transposed to make it positive semidefinite.

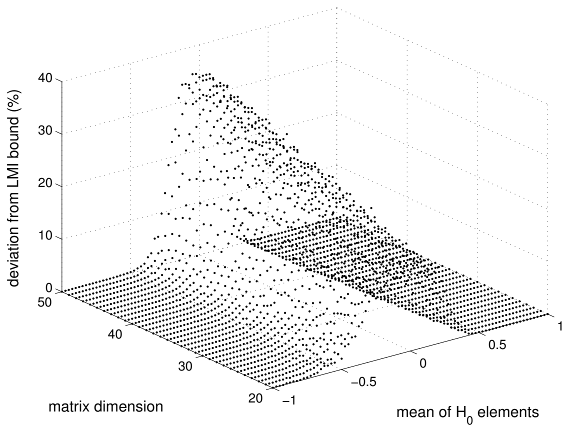

Fig. 2. Mean deviation from the LMI bound plotted as a function of the mean of the elements of H_0 for a group of randomly generated matrices.

of the elements of H_0, with $Q = H_0'H_0$. As illustrated by figure 2, the biggest error is when the mean is around zero, quickly dropping to very small values when the mean goes to positive or negative values. The reason for that behavior is that, for a positive or negative biased matrix, the maximum is attained at vertices with progressively more elements of the same sign. Thus, the maximum tend to be at the sum of the elements of the partly diagonalizated T matrix. In MMMPC control problems, as found by the authors through many simulations, the mean of the sum of the elements of matrix H, which ultimately depends on that of H_0, is generally neatly different from zero, although not all its entries share the same sign. This means that the bound accuracy will be close to that of the LMI bound.

4.1 Computational burden of the upper bound

The proposed bound has lower computational burden than the LMI bound. Figure 3 shows the speed-up factor for group of random matrices with different mean and dimensions. To make this comparison as fair as possible, the LMI bound has been obtained using the solver by F. Rendl that can be downloaded from `http://www.math.uni-klu.ac.at/or/Software`. This solver, specific for problem (11), proved to be very much efficient than the standard solver provided with the LMI Toolbox of *Matlab*. Remark 1 has been taking into account when computing the proposed bound. It can be seen that the proposed bound can be computed using many times less floating point operations [3] than the LMI bound. This can be exploited to apply worst case control for systems with fast dynamics or use hardware with low computational power. Moreover, the proposed bound can be computed using simple matrix operations, whereas in the case of LMI solvers more complex operations such as the Cholesky decomposition are needed. This means that the algorithm can be coded easily even in non mathematical oriented programming languages such as those found in industrial embedded control hardware.

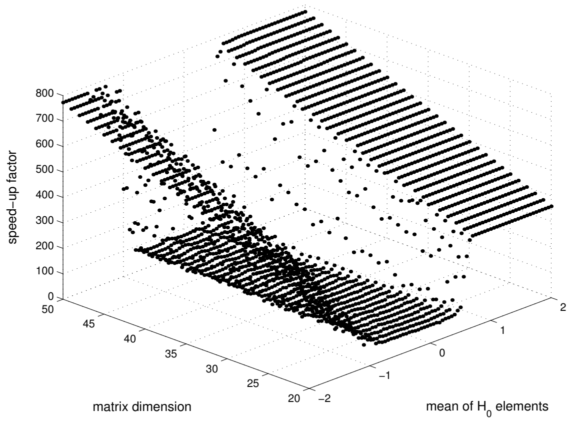

Fig. 3. Speed-up factor between the proposed upper bound and the LMI bound (computed as $\frac{\text{flops}_{\text{LMI}}}{\text{flops}_{\text{proposed}}}$) for a group of randomly generated matrices with different mean and dimension.

It is noteworthy that as matrix dimension grows no clear trend is seen when the mean of H_0 is around zero. The computational burden for the proposed bound grows as $O(n^3)$ where n is the matrix dimension. On the other hand, the interior point methods used for this class of LMI problems are also $O(n^3)$ algorithms, hence the relatively constant speed-up factor seen in figure 3 for zero mean matrices. Note however that the underlying constant in the number of operations needed by the LMI solver is bigger, leading to the speed-up factor shown. On the other hand, as illustrated in figure 3, the mean of the entries and the dimension of H_0, where $Q = H_o'H_o$, affects the speed-up factor severely when this mean is different from zero. In this case, in the light of remark 1, the proposed upper bound is computed much faster than the LMI bound. In fact, when most of the entries of matrix H_0 (and therefore H) shares the same sign, the speed-up factor between the proposed bound and the LMI bound grows progressively showing a linear trend. This can be interpreted as an evidence that for those matrices, the proposed bound tends to be $O(n^2)$ instead of the worst case (i.e., zero mean H_0) in which the algorithms behaves as $O(n^3)$. Table 1 shows how in a real MMMPC, the proposed bound can be computed with significantly less operations than the LMI bound. Furthermore, the speed-up factor grows with the prediction horizon, meaning that longer prediction horizons can be used with a given hardware. These figures are justified because, as pointed out before, the mean of the sum of the elements of matrix H is different from zero.

5. SIMULATION EXAMPLE

As pointed out in section 2, the results of the paper can be applied to different model descriptions. As

[3] The number of operations needed for computing each bound have been obtained using the *Matlab* **flops** function. The precision of the LMI solver was left to default.

Table 1. Mean speed-up factor for different values of the prediction horizon (N) in the simulation example of section 5.

N	Speed-up factor
11	62.11
20	92.74
30	118.11
40	159.28
50	178.05

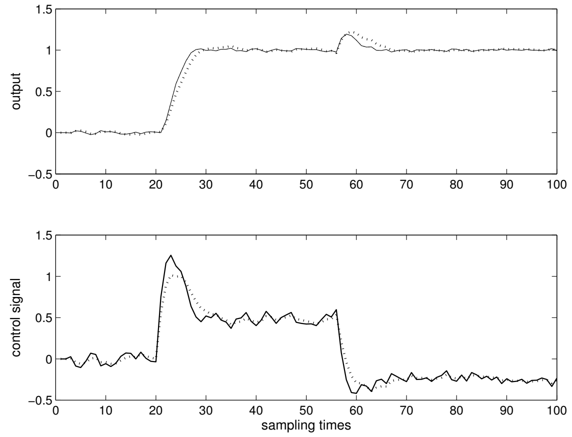

Fig. 4. Simulation example. Output and control signal shown for the proposed worst case controller (solid) and the exact Min-Max MPC (dotted).

an example of simulated application of the MMMPC controller proposed in section 2 consider the following first order process described by a CARIMA model:

$$y_{k+1} = 1.9y_k - 0.9y_{k-1} + 0.2\Delta u_k + \theta_{k+1}$$

with $\Delta = (1 - z^{-1})$. In figure 4 it is shown a simulation of the MMMPC applied with parameters $\theta_m = 0.1$, $N_u = N = 11$ and $L = 2$ (solid plot). Note that the horizon is set to a relatively low value so that the exact Min-Max MPC controller can be simulated in an average computer. Also in the same figure it is shown a Min-Max MPC in which the cost is computed exactly (dotted plot). For both controllers a small random noise of 0.05 of amplitude has been added to the output. Furthermore, at sampling time $k = 55$ a step disturbance of amplitude 0.15 is added to the process output. Computing the MMMPC simulation was 44.3 times faster than the Min-Max MPC (i.e., a speed-up factor of 44.3). This speed up would be more dramatic as the horizon grows, because the number of vertices of Θ grows exponentially whereas the computational burden of the proposed bound grows polynomially. Also in the figure it is shown the value of the control signal applied for both controllers. As it is shown in figure 4 the performance achieved by both controllers is very close. The proposed worst case controller is slightly more aggressive than the exact Min-Max MPC. The explanation for this is that the upper bound based controller is more conservative than the exact Min-Max MPC. Thus, expecting a worse situation, it reacts to the disturbance in a more aggressive manner. The behavior of both controllers, however, is pretty much the same, a sign of the tightness of the upper bound.

6. CONCLUSIONS

An MMMPC based on an efficient upper bound has been presented in this paper. It has much lower computational burden than other approaches based on LMI techniques and can be implemented in dedicated industrial control hardware. The price to be paid is a moderately increment in the conservativeness of the bounds obtained. However, its little computational burden opens new fields of applications of MMMPC controllers.

REFERENCES

Alamo, T., D. Muñoz de la Peña, D. Limón Marruedo and E.F. Camacho (2003). Constrained min max predictive control: a polynomial time approach. In: *Proccedings of the 42st IEEE Conference on Decision and Control*. Maui, Hawai, USA.

Bazaraa, M.S. and C.M. Shetty (1979). *Nonlinear Programming. Theory and Algorithms*. John Wiley & Sons.

Camacho, E.F. and C. Bordóns (1999). *Model Predictive Control*. Springer-Verlag.

Campo, P.J. and M. Morari (1987). Robust Model Predictive Control. In: *Proc. American Control Conference*. pp. 1021–1026.

Casavola, A., M. Giannelli and Edoardo Mosca (2000). Min-Max Predictive Control strategies for input-saturated polytopic uncertain systems. *Automatica* **36**, 125–133.

Kerrigan, E.C. and T. Alamo (2004). A convex parameterization for solving constrained min-max problems with a quadratic cost. In: *Proceedings of 2004 American Control Conference*. Boston, MA, USA.

Kothare, M.V., V. Balakrishnan and M. Morari (1996). Robust constrained model predictive control using linear model inequalities. *Automatica* **32**(10), 1361–1379.

Lu, Y. and Y. Arkun (2000). Quasi-Min-Max MPC algorithms for LPV systems. *Automatica* **36**(4), 527–540.

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

Nesterov, Y.E., H. Wolkowicz and Y. Ye (2000). Semidefinite programming relaxations of nonconvex quadratic optimization. In: *Handbook of Semidefinite Programming: Theory, Algorithms and Applications* (H. Wolkowicz, R. Saigal and L. Vanderberghe, Eds.). Kluwer Academics Publishers. Boston,MA. p. 34.

Wan, Z. and M.V. Kothare (2003). An efficient offline formulation of robust model predictive control using linear matrix inequalities. *Automatica* **39**, 837–846.