

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/251724531>

Computational burden reduction in min-max MPC

Article in *Journal of the Franklin Institute* · November 2011

DOI: 10.1016/j.jfranklin.2011.07.008

CITATIONS

8

READS

120

3 authors, including:



Daniel R. Ramirez
University of Seville

57 PUBLICATIONS 853 CITATIONS

[SEE PROFILE](#)



Teodoro Alamo
Universidad de Sevilla

240 PUBLICATIONS 4,675 CITATIONS

[SEE PROFILE](#)

Some of the authors of this publication are also working on these related projects:



Economic Model Predictive Control [View project](#)



ECOCIS Economic Operation of Critical Infrastructure Systems [View project](#)

Computational burden reduction in Min-Max MPC

D.R. Ramirez^{*}, T. Alamo, E.F. Camacho¹,

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

Abstract

Min-Max Model Predictive Control (MMMPC) is one of the strategies used to control plants subject to bounded uncertainties. The implementation of MMMPC suffers a large computational burden due to the complex numerical optimization problem that has to be solved at every sampling time. This paper shows how to overcome this by transforming the original problem into a reduced min-max problem whose solution is much simpler. In this way, the range of processes to which MMMPC can be applied is considerably broadened. Proofs based on the properties of the cost function and simulation examples are given in the paper.

1 Introduction

Model Predictive Control (MPC) is one of the few advanced control strategies with a significant success in the industrial community ([9]). MPC is able to take into account constraints in the computation of the control law. It can also handle multivariable processes without resorting to decoupler schemes. The tuning process is intuitive and the control effort or economic objectives can be taken into account. The effect of process model uncertainties can also be considered in the control law. The inherent robustness of MPC is enhanced with different approaches such as terminal costs, dual control laws, terminal constraints often designed using set invariance concepts and more recently by

^{*} Corresponding author.

Email addresses: danirr@cartuja.us.es (D.R. Ramirez),
alamo@cartuja.us.es (T. Alamo), eduardo@esi.us.es (E.F. Camacho).

¹ The authors acknowledge MCYT-Spain and the European Commission for funding this work (contracts DPI2007-66718 and FP7-223854).

means of tube-based formulations [28,37,35]. Another strategy used to improve performance under model uncertainties (thus enhancing the robustness of the control law) is worst case design. This is usually accomplished using the Min-Max MPC formulation [10].

In Min-Max MPC controllers ([9,10]), 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 ([42,24,40]). Thus, the implementation of this type of control is severely compromised leading to a lack of experimental results. Only a few applications to plants with slow dynamics ([8]) or complex simulated models ([21]) have been reported. For moderate fast dynamics the min-max problem can be solved numerically only when the number of extreme realizations of the uncertainty is relatively low. This is the case when the prediction horizon is small. When fast dynamics are to be controlled the min-max problem cannot be solved numerically, and approximate solutions have to be used ([33]). However, these techniques impose great rigidity in the controller parameters, as well as a certain degree of approximation error.

The MMMPC control law, traditionally regarded as highly nonlinear, has proven to be piecewise affine when a quadratic ([34]) or 1-norm based criterion ([5,20]) is used as the cost function. With these results, together with those obtained when multiparametric mathematical programming is applied ([5]), explicit forms of the control law can be built. Algorithms to obtain the explicit formulation of MMMPC controllers based on quadratic cost functions and linear models have been given in [31,30], in [14] where experimental results are also presented and in [6] in which LPV systems are considered. An approximate explicit description of MMMPC with nonlinear prediction model can be found in [16]. However, in explicit formulations of MPC or MMMPC controllers, the number of regions in which the state space has to be partitioned grows with the prediction horizon in a combinatorial explosion. Thus, storage requirements and searching time for the appropriate region can be very high for practical values of the prediction and control horizons. A search tree structure has been proposed to reduce the searching time in the MPC context ([18,41]). Hash tables have been also used for this in [3]. If the process model or the controller parameters change, however, the computation of the regions has to be done again. Also, different robust predictive controllers based on multiparametric programming have appeared in [38,39]. On the other hand, explicit formulations of predictive controllers not based on multiparametric programming can be found in [43]. A different approach that uses an offline robustification of an explicit predictive controller is presented in [36]. The reader is referred to [2] for a detailed survey in explicit model predictive control.

Often the computational burden issue is solved by using a bound of the worst case cost instead of computing it explicitly ([32]). The use of such upper bounds, turns the control law different as that of a MMMPC, but the on-line implementation can be easier. The upper bound can be computed using LMI techniques such as in [22] or [11,27,13]. In these strategies the control law takes the form of a fixed state feedback gain along the prediction horizon. This state feedback gain is computed from an LMI problem that minimizes the upper bound of the worst case cost. In [44], the strategy of [22] is improved by moving most of the computations offline using an algorithm based on asymptotically stable invariant ellipsoids (the online computations are reduced to a search in a lookup table). More recently, in [15] this strategy has been improved in two ways, first by using nominal performance cost in substitution of worst-case one, such that feasibility can be improved and, second, by using a heuristic varying-horizon formulation in which the feedback gains are obtained in a backward manner. Other approximation to lower the computational burden of LMI based strategies like [22] is that of [12], in which uncertain *norm-bounded* linear systems are considered instead of the usual polytopic systems, resulting in a number of LMIs that grows only linearly with the control horizon. On the other hand, the use of fixed state feedback along the prediction horizon is a restriction on the feasible control signals values. This limits the achievable performance and the size of the allowable set of initial conditions. These problems have been tackled in [23]. In that work, extra degrees of freedom are introduced through the use of perturbations on the fixed state-feedback law. Also, most of the computational burden is moved to an offline feasibility problem. A generalization of this approach with greater domain of attraction was presented in [17] where a in general non-convex (but often convex) optimization problem has to be solved offline together with a very efficient online optimization. Note however, that none of these control strategies are true MMMPC controllers because no min-max problem is really solved. On the other hand, the use of an upper bound of the worst case cost can result in a more conservative controller.

This paper shows a way of implementing MMMPC that requires only a fraction of the time required by the usual min-max solvers. The method is based on transforming the original min-max problem into an equivalent reduced problem whose solution is much simpler. Thus, for many processes in which time constants are measured in seconds or minutes, the reduced min-max problem can be solved on-line using standard numerical algorithms. This approach has been used in a previous work by the authors ([1]). However, solving a nonlinear optimization problem and a QP problem at each sampling time was required to obtain the reduced min-max problem. Thus, the computational efficiency improvement was greatly reduced. In this paper a more efficient way to obtain such a reduced problem using the solution of the min-max problem solved in the previous sampling time is presented. Although this work deals with unconstrained MMMPC, a constrained MMMPC controller which can be effi-

ciently implemented is also presented. It uses the solution of the unconstrained controller in a constrained min-max formulation with a related convex (and therefore tractable) optimization problem.

The paper is organized as follows: Section 2 presents the basic Min-Max MPC with bounded global uncertainties algorithm. The efficient implementation strategy is introduced in Section 3. The strategy to get a reduced min-max problem equivalent to the original problem to be solved is presented in Section 4. This strategy is illustrated in Section 5. Section 6 presents conclusions and questions to be addressed in future works. Finally, in appendix A the efficient constrained min-max formulation is presented.

2 Min-Max MPC with bounded additive uncertainties

The bounded additive uncertainties approach assume that all modelling errors are globalized in a vector of parameters, such that the plant can be described by the following family of models:

$$\begin{aligned} \hat{x}(t+1) &= \hat{f}(x(t), u(t)) + \theta(t+1, x(t), u(t), \varsigma(t)), \\ &\|\theta(t, x(t), u(t), \varsigma(t))\|_{\infty} \leq \epsilon \end{aligned} \tag{1}$$

where $\hat{f}(\cdot)$ is the nominal model (in this work only linear models are considered) and $\theta(\cdot)$ is the additive uncertainty which can depend on time, process state or inputs or any amplitude bounded signal that may be generated by any deterministic or stochastic process. It may be argued that these types of uncertainties are rather conservative and more disturbances than uncertainties, because they seem to work as external perturbations. However, the only assumption made is that they are bounded, thus they represent the total contribution of the uncertainty to the process state evolution. Also, this strategy can be compared favorably with the parametric uncertainty approach in the sense that it does not suppose that the process to be controlled can be described exactly by an unknown linear model. Furthermore, the type of uncertainty description used in this paper can be understood better by control practitioners, as they only have to consider which is the maximum error of their model for the next sampling time. This maximum error can be easily obtained from any identification algorithm (such as least squares identification), and thus, models for systems with bounded additive uncertainties are easy to obtain. Conservativeness can also be adjusted by considering a lower ϵ in the controller design.

The results presented in this paper are valid for linear systems, either for state space models with bounded additive uncertainties:

$$\begin{aligned}x(t+1) &= Ax(t) + Bu(t) + D\theta(t+1, x(t), u(t), \varsigma(t)) \\y(t) &= Cx(t)\end{aligned}\quad (2)$$

with $\|\theta(\cdot)\|_\infty \leq \epsilon$, or by input-output models like CARIMA models with bounded additive uncertainties:

$$\tilde{A}(z^{-1})y(t) = z^{-d}B(z^{-1})\Delta u(t-1) + \theta(t, x(t-1), u(t-1), \varsigma(t-1)) \quad (3)$$

where $\|\theta(\cdot)\|_\infty \leq \epsilon$, $\tilde{A}(z^{-1}) = \Delta A(z^{-1})$, $\Delta = 1 - z^{-1}$ being $y(t)$ and $u(t)$ the output and control signals 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. Also, in this case, the state can be considered to be formed by the present values of process outputs and finite series of past input increments and past output signals.

A cost function J is used in Min-Max MPC to indicate how well the process follows the desired trajectory over the time interval t to $t + N$, where N is called the prediction horizon. That cost function depends on the present process state, present and future control signals and uncertainties and also the present and future values of the setpoint. An usual form of this cost function is a quadratic criterion given by:

$$J(\boldsymbol{\theta}, \mathbf{u}, x) = \sum_{j=1}^N x(t+j|t)^T Q_j x(t+j|t) + \sum_{j=1}^{N_u} u(t+j-1)^T R_j u(t+j-1) \quad (4)$$

where N is the prediction horizon, N_u is called the control horizon, $\boldsymbol{\theta} = [\theta(t+1, \dots), \dots, \theta(t+N, \dots)]$ is the sequence of the future uncertainty values², $\mathbf{u} = [u(t), \dots, u(t+N_u-1)]$ is the sequence of future control signals and $x(t+j|t)$ is the prediction of the state at $t+j$ made at time t taking into account $\boldsymbol{\theta}$, \mathbf{u} and x . On the other hand, $Q_j = Q_j^T \geq 0$ and $R_j = R_j^T > 0$ are used as weighting parameters. Note that this cost function allows the inclusion of a quadratic terminal cost $x(t+N)^T P x(t+N)$ by making $Q_N = P$. In input-output models like (3), it is usual to consider a setpoint and also, the sequence \mathbf{u} is composed by the future control increments, i.e., $\mathbf{u} = [\Delta u(t), \dots, \Delta u(t+N_u-1)]$. Then, the cost function is given by:

$$J(\boldsymbol{\theta}, \mathbf{u}, x) = \sum_{j=1}^N (y(t+j|t) - r(t+j))^2 + \lambda \sum_{j=1}^{N_u} (\Delta u(t+j-1))^2 \quad (5)$$

² Note that $\boldsymbol{\theta}$ is unknown, but this is not a problem as the worst case cost is attained at the extreme realizations of $\boldsymbol{\theta}$ (that is, the vertices of the set of possible uncertainty sequences), which are known.

where $\lambda > 0$ is called the control effort penalty factor, $r(t+j)$ is the set point value for time $t+j$ and $y(t+j|t)$ is the prediction of the output at $t+j$ made at time t taking into account $\boldsymbol{\theta}$, \mathbf{u} and x . In both cases, the control horizon is always $N_u \leq N$ and the control signal is assumed to be constant in the time interval $t+N_u$ to $t+N$.

The objective of MPC control is to compute the future control sequence $u(t), u(t+1), \dots, u(t+N_u-1)$ in such a way that the optimal j -step ahead predictions of the process output $y(t+j|t)$ or process state $x(t+j|t)$ are driven close to the desired set point sequence over the prediction horizon. When bounded uncertainties are considered explicitly, there are experimental evidences [9,14], that a better performance can be obtained if the controller minimizes the objective function for the worst situation. Also, the local margin of stability is increased [26]. Thus, the optimal control sequence will be computed solving the following min-max problem:

$$\mathbf{u}(x) = \arg \min_{\mathbf{u}} \max_{\boldsymbol{\theta} \in \Theta} J(\boldsymbol{\theta}, \mathbf{u}, x) \quad (6)$$

where $\Theta = \{\boldsymbol{\theta} \in \mathbb{R}^L : \|\boldsymbol{\theta}\|_\infty \leq \epsilon\}$ is the set of possible uncertainty sequences, and L is $N \cdot \dim(y)$ in an input-output description or $N \cdot \dim(x)$ in a state-space approach. The optimal cost function (that is the minimum maximum cost) of the resulting min-max problem can be written as:

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

where

$$J^*(\mathbf{u}, x) = \max_{\boldsymbol{\theta} \in \Theta} J(\boldsymbol{\theta}, \mathbf{u}, x) \quad (7)$$

is the worst case cost function.

The properties of the min-max problem depend on the structure and properties of the cost function $J(\boldsymbol{\theta}, \mathbf{u}, x)$ which in turn is closely related to the model structure used to predict the future evolution of the process state. It can be seen ([9]) that the predicted values of $x(t)$ and $y(t)$ using models (2) and (3) depend on the process uncertainties, inputs and state in an affine way³, i.e.:

$$\xi = G_u \mathbf{u} + G_\theta \boldsymbol{\theta} + F_x x(t) \quad (8)$$

³ This is also true for other types of convolution models like FIR and finite step response models. In this case, the state can be considered to be formed by the present value of process output and finite series (normally much longer than in transfer function models) of past inputs. Thus, the results in this paper can also be used with MMMPC based on these models.

where ξ can be either the vector of predictions of process state or output over the prediction horizon ([9]). Assume without loss of generality a zero reference. Taking into account (8), the cost functions (4) and (5) can be rewritten as ([9]):

$$J(\boldsymbol{\theta}, \mathbf{u}, x) = \mathbf{u}^T M \mathbf{u} + 2(Nx + n(\boldsymbol{\theta}))^T \mathbf{u} + x^T C x + 2d(\boldsymbol{\theta})^T x + h(\boldsymbol{\theta}) \quad (9)$$

In the case of (5) the matrices of (9) are $M = G_u^T G_u + \lambda I$, $C = F_x^T F_x$, $N = G_u^T F_x$, $n(\boldsymbol{\theta}) = G_u^T G_\theta \boldsymbol{\theta}$, $d(\boldsymbol{\theta}) = F_x^T G_\theta \boldsymbol{\theta}$, $h(\boldsymbol{\theta}) = \boldsymbol{\theta}^T G_\theta^T G_\theta \boldsymbol{\theta}$. On the other hand, for the state space model: $M = G_u^T \mathbf{Q} G_u + \mathbf{R}$, $C = F_x^T \mathbf{Q} F_x$, $N = G_u^T \mathbf{Q} F_x$, $n(\boldsymbol{\theta}) = G_u^T \mathbf{Q} G_\theta \boldsymbol{\theta}$, $d(\boldsymbol{\theta}) = F_x^T \mathbf{Q} G_\theta \boldsymbol{\theta}$, $h(\boldsymbol{\theta}) = \boldsymbol{\theta}^T G_\theta^T \mathbf{Q} G_\theta \boldsymbol{\theta}$, where \mathbf{Q} and \mathbf{R} are diagonal matrices

$$\mathbf{Q} = \begin{bmatrix} Q_1 & & \\ & \ddots & \\ & & Q_N \end{bmatrix}, \quad \mathbf{R} = \begin{bmatrix} R_1 & & \\ & \ddots & \\ & & R_{N_u} \end{bmatrix}$$

Recall that $\lambda > 0$ and that $R_j = R_j^T > 0$, thus $M > 0$ for both cost functions. On the other hand, $h(\boldsymbol{\theta}) \geq 0$. These conditions imply that J is convex on $\boldsymbol{\theta}$ and strictly convex on \mathbf{u} . Thus the maximum of $J(\boldsymbol{\theta}, \mathbf{u}, x)$ will be attained at least at one of the 2^L vertices of the polytope Θ ([4]). Each of these vertices will be related to an extreme realization of the uncertainty sequence $\boldsymbol{\theta}$, being an extreme realization a sequence in which all its components are either $-\epsilon$ or ϵ . Also, any local minimizer of $J^*(\mathbf{u}, x)$ will be the global minimizer ([4]) and the solution of the min-max problem (6) is unique.

Taking into account that the maximum of $J(\boldsymbol{\theta}, \mathbf{u}, x)$ always will be attained at least at one of the vertices $\boldsymbol{\theta}_i$ it is clear that the original min-max problem (6) is equivalent to:

$$J^s(x) = \min_{\mathbf{u}} \max_{\boldsymbol{\theta}_i \in \text{vertices}(\Theta)} J(\boldsymbol{\theta}_i, \mathbf{u}, x) = \min_{\mathbf{u}} J^*(\mathbf{u}, x)$$

with:

$$J^*(\mathbf{u}, x) = \max_{\boldsymbol{\theta}_i \in \text{vertices}(\Theta)} J(\boldsymbol{\theta}_i, \mathbf{u}, x)$$

where $\text{vertices}(\Theta)$ is the set of the 2^L vertices of Θ .

In the following, $J_i(\mathbf{u}, x)$ denotes $J(\boldsymbol{\theta}_i, \mathbf{u}, x)$, where $\boldsymbol{\theta}_i$ is the uncertainty extreme realization related to vertex i . Thus,

$$J_i(\mathbf{u}, x) = \mathbf{u}^T M \mathbf{u} + 2(Nx + n_i)^T \mathbf{u} + x^T C x + 2d_i^T x + h_i \quad (10)$$

where M , N and C are the same matrices as in (9) and n_i , d_i , h_i are equal to $n(\boldsymbol{\theta}_i)$, $d(\boldsymbol{\theta}_i)$, $h(\boldsymbol{\theta}_i)$. Thus, the optimal control sequence can be computed by solving:

$$\mathbf{u}(x) = \arg \min_{\mathbf{u}} \max_{i \in [1, \dots, 2^L]} J_i(\mathbf{u}, x) \quad (11)$$

Note that, in order to solve problem (11), for each candidate solution \mathbf{u} of the outer minimization problem, the cost function J must be evaluated at all the 2^L vertices of Θ ⁴. This is the cause of the great computational burden required to obtain the optimal control signal $\mathbf{u}(x)$. In fact, problem (11) is NP-hard, and usually cannot be solved in real time for reasonable values of the prediction horizon or sampling time. A possible solution to this problem would be to reduce the number of vertices considered in the inner maximization problem. The following sections present a strategy to reduce the computational burden of (11) based on this idea.

3 Efficient implementation strategy

In this section the main ideas of the efficient implementation strategy are introduced. This strategy is based on the reduction of the number of vertices to be explored in the maximization part of the min-max problem. Solving such a reduced min-max problem would involve a much lower computational burden than that of the original problem. Thus, the range of processes to which MMMPC can be applied would be considerably broadened.

A main concept in this paper is that of active vertices. As stated in Section 2, due to the convexity of the cost function on $\boldsymbol{\theta}$, the maximum cost for any given \mathbf{u} , will be attained at least at one or more vertices of Θ . This also holds for the optimal solution of the min-max problem, $\mathbf{u}(x)$. Thus for $\mathbf{u} = \mathbf{u}(x)$ there will be one or more vertices of Θ for which it holds that:

$$J^s(x) = J_i(\mathbf{u}(x), x) \quad (12)$$

where i is any of the vertices in which the maximum cost is attained for $\mathbf{u} = \mathbf{u}(x)$. Those vertices that satisfy (12), i.e., those that give the maximum cost at the solution of the min-max problem, are called active vertices. To illustrate the concept of active vertices let us consider the simplest of all examples, that is a system in which $\dim(y) = 1$, $N_u = N = 1$. In this case Θ has only two vertices, $\boldsymbol{\theta}_1 = \{-\epsilon\}$ and $\boldsymbol{\theta}_2 = \{\epsilon\}$. Those vertices are related

⁴ Note that in input-output models, $L = N \cdot \dim(y)$, thus the order of the system has no effect in the complexity of the min-max problem.

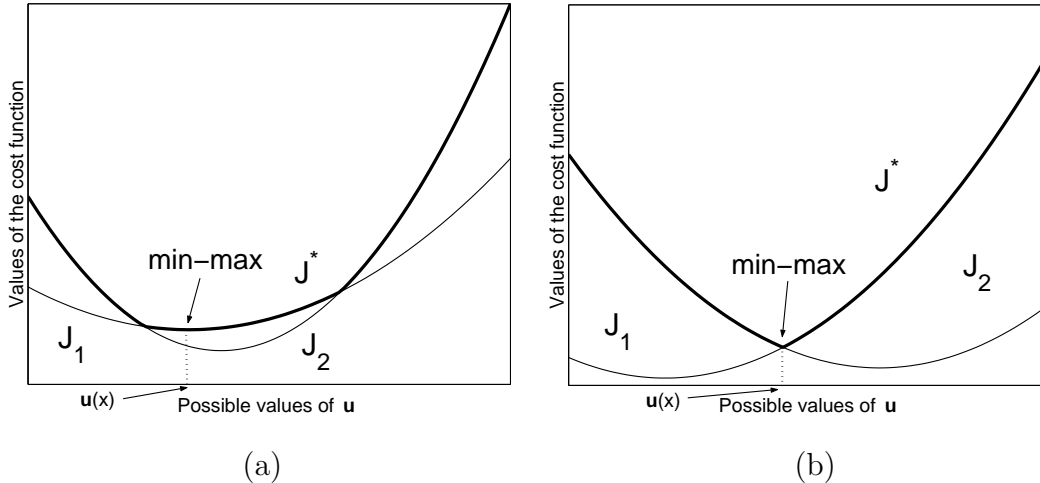


Fig. 1. Types of solutions of min-max problem with two quadratic functions ($N_u = N = 1$): (a) One active vertex. (b) Two active vertices.

to two quadratic cost functions $J_1(\mathbf{u}, x)$, $J_2(\mathbf{u}, x)$. Also, in this case the active vertices can be either θ_1 or θ_2 or both. Figure 1a shows the value of J_1 and J_2 for a given x . Note that both J_1 and J_2 give the maximum cost for some values of \mathbf{u} , but the minimum maximum cost, that is the min-max, is attained only at J_1 . Thus in this case the active vertex is θ_1 . Now consider Figure 1b. In this case, the min-max is attained at an intersection of J_1 and J_2 , that is, both θ_1 and θ_2 are active vertices.

For a given system and state x , the solution of (6) can be characterized by the active vertices. At any of the active vertices the cost function will be equal to the optimal cost $J^s(x)$. Thus, the following definition can be given:

Definition 1 $I(x)$, the set of active vertices is given by:

$$I(x) = \{i : J_i(\mathbf{u}(x), x) = J^s(x)\} \quad (13)$$

It is easy to see that the solution of (6) is also the solution of a reduced min-max problem given by:

$$\mathbf{u}(x) = \arg \min_{\mathbf{u}} \max_{i \in I(x)} J_i(\mathbf{u}, x) \quad (14)$$

The reason is that those quadratic functions that are related to not active vertices do not contribute to the solution of the min-max problem. Thus, they can be discarded. This situation is illustrated in Figure 2. Figure 2a shows the quadratic functions for a min-max problem with $\dim(y) = 1$, $N_u = 1$, $N = 2$. That is, there are four vertices $\theta_1 = \{-\epsilon, -\epsilon\}$, $\theta_2 = \{-\epsilon, \epsilon\}$, $\theta_3 = \{\epsilon, -\epsilon\}$, $\theta_4 = \{\epsilon, \epsilon\}$. Note that the maximum cost takes place only three of them (those related to J_1 , J_2 and J_4) for any value of \mathbf{u} . It is clear that J_3 can be dropped

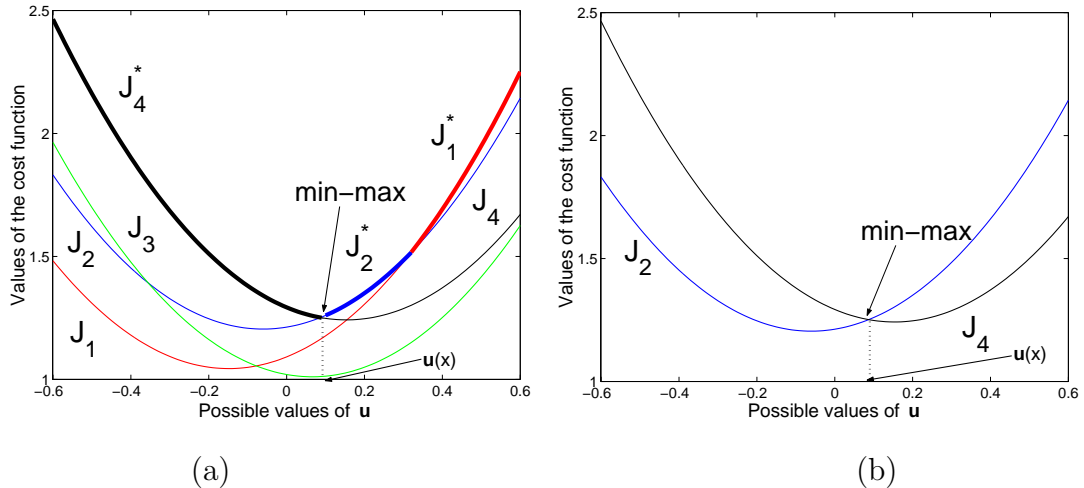


Fig. 2. Two min-max problems with the same solution: (a) Full min-max with all cost functions. (b) Reduced min-max with cost functions related to active vertices.

from the min-max problem as the solution cannot be at θ_3 . But J_1 can also be dropped, because the minimum maximum cost does not occur on it. That is, the active vertices in this case are θ_2 and θ_4 . Now consider Figure 2b, in which it is shown how solving a min-max problem with just J_2 and J_4 (the active vertices of the original min-max problem) yields the same solution of the original problem of Figure 2a. Thus, the solution of a min-max problem can be obtained taking into account only the active vertices. As the number of active vertices is generally much lower than the total number of vertices, problem (14) can be solved with little computational burden.

If the process state changes from x to $x + \Delta x$ the active vertices set also changes so that $I(x + \Delta x) \neq I(x)$. Although $I(x + \Delta x)$ cannot be obtained without solving the min-max problem for $x + \Delta x$, it will be shown in this paper that a conservative estimation of $I(x + \Delta x)$ can be efficiently computed using $I(x)$ and Δx . This estimation, which will be denoted as $I_e(x + \Delta x)$, satisfies that $I(x + \Delta x) \subseteq I_e(x + \Delta x)$. Thus, the reduced min-max problem (14) for $x + \Delta x$ can be substituted by the following reduced problem:

$$\mathbf{u}(x + \Delta x) = \arg \min_{\mathbf{u}} \max_{i \in I_e(x + \Delta x)} J_i(\mathbf{u}, x + \Delta x) \quad (15)$$

which is also equivalent to the original min-max problem. As the number of vertices included in I_e will be a small fraction of the whole 2^L vertices, the computational burden will be accordingly much lower.

The procedure to obtain I_e will be given in the next section. A preliminary proposition that characterizes the change in a quadratic function $J_i(\mathbf{u}, x)$ when its parameters are perturbed is stated in the following:

Proposition 1 *Given $J_i(\mathbf{u}, x)$ as in (10) the value of $J_i(\mathbf{u} + \Delta \mathbf{u}, x + \Delta x)$ can*

be expressed as:

$$J_i(\mathbf{u} + \Delta\mathbf{u}, x + \Delta x) = J_i(\mathbf{u}, x) + Q_i(\Delta\mathbf{u}, \Delta x) \quad (16)$$

where $Q_i(\Delta\mathbf{u}, \Delta x)$ is a quadratic function given by:

$$Q_i(\Delta\mathbf{u}, \Delta x) = \Delta\mathbf{u}^T M \Delta\mathbf{u} + 2g_{u,i}^T \Delta\mathbf{u} + 2g_{x,i}^T \Delta x + 2\Delta x^T N^T \Delta\mathbf{u} + \Delta x^T C \Delta x \quad (17)$$

where $g_{u,i} = (M\mathbf{u} + Nx + n_i)$ and $g_{x,i} = (N^T\mathbf{u} + Cx + d_i)$.

4 On-line estimation of the set of active vertices

This section shows how to build an estimation of the set of active vertices for a given process state by means of the known solution of another value of the process state. This strategy can be used to estimate $I(x(t))$ using $\mathbf{u}(x(t-1))$ and $I(x(t-1))$ both available at sampling time t . With this estimation of $I(x(t))$, which will be denoted as $I_e(x + \Delta x)$, the min-max problem to be solved at sampling time t can be replaced by an equivalent reduced min-max problem. The following theorem presents a strategy to build such an estimation.

Theorem 1 *Given a known x and its corresponding optimal solution $\mathbf{u} = \mathbf{u}(x)$, Δx and γ defined as follows:*

$$\gamma = J^*(\mathbf{u}, x + \Delta x) - J^*(\mathbf{u}, x) \quad (18)$$

For each pair of vertices k, i , where i is an active vertex for x and k is a vertex candidate to be active for $x + \Delta x$, define $R_{k,i}$ as follows:

$$R_{k,i} = \max_z Q_k(z, \Delta x) - Q_i(z, \Delta x) \quad (19)$$

s.t.

$$Q_i(z, \Delta x) \leq \gamma$$

$$g_{u,i}^T z \geq 0$$

The extended set of active vertices, I_e , is defined as follows:

$$I_e(x + \Delta x) = \left\{ k : J^s(x) - J_k(\mathbf{u}, x) \leq \max_{i \in I(x)} R_{k,i} \right\} \quad (20)$$

Then it holds that $I(x + \Delta x) \subseteq I_e(x + \Delta x)$

Proof: Let $\Delta \mathbf{u} = \mathbf{u}(x + \Delta x) - \mathbf{u}(x)$ be the change of the solution of problem 6 when the process state changes from x to $x + \Delta x$. Let j be any of the active vertices for x , that is $j \in I(x)$. Being j an active vertex for x it is true that:

$$J_j(\mathbf{u}, x) = J^*(\mathbf{u}, x) = J^s(x) \quad (21)$$

Moreover it holds that:

$$J^s(x + \Delta x) = \max_i \{J_i(\mathbf{u} + \Delta \mathbf{u}, x + \Delta x)\} \geq J_j(\mathbf{u} + \Delta \mathbf{u}, x + \Delta x) \quad (22)$$

In fact, if $j \in I(x + \Delta x)$ then, inequality (22) turns into an equality. Otherwise it is a strict inequality due to the fact that the maximum is not attained at J_j .

Let us suppose that $k \in I(x + \Delta x)$. Then:

$$J_k(\mathbf{u} + \Delta \mathbf{u}, x + \Delta x) = J^s(x + \Delta x) \geq J_j(\mathbf{u} + \Delta \mathbf{u}, x + \Delta x) \quad (23)$$

Using Proposition 1:

$$\begin{aligned} J_k(\mathbf{u} + \Delta \mathbf{u}, x + \Delta x) &= J_k(\mathbf{u}, x) + Q_k(\Delta \mathbf{u}, \Delta x) \\ J_j(\mathbf{u} + \Delta \mathbf{u}, x + \Delta x) &= J_j(\mathbf{u}, x) + Q_j(\Delta \mathbf{u}, \Delta x) \end{aligned} \quad (24)$$

therefore, from (23):

$$J_k(\mathbf{u}, x) + Q_k(\Delta \mathbf{u}, \Delta x) \geq J_j(\mathbf{u}, x) + Q_j(\Delta \mathbf{u}, \Delta x) = J^s(x) + Q_j(\Delta \mathbf{u}, \Delta x) \quad (25)$$

which yields:

$$J^s(x) - J_k(\mathbf{u}, x) \leq Q_k(\Delta \mathbf{u}, \Delta x) - Q_j(\Delta \mathbf{u}, \Delta x) \quad (26)$$

Then, every $k \in I(x + \Delta x)$ must verify (26), thus this inequality can be used to build the active vertex set for $x + \Delta x$. However, the exact value of $\Delta \mathbf{u}$ is unknown, so an estimation has to be found. Using such an estimation of $\Delta \mathbf{u}$ a conservative estimation of $I(x + \Delta x)$ can be built, as shown in the following.

Taking into account Proposition 1 and (21) $J_j(\mathbf{u} + \Delta \mathbf{u}, x + \Delta x)$ can be expressed as:

$$J_j(\mathbf{u} + \Delta\mathbf{u}, x + \Delta x) = J_j(\mathbf{u}, x) + Q_j(\Delta\mathbf{u}, \Delta x) \quad (27)$$

$$\begin{aligned} &= J^*(\mathbf{u}, x) + Q_j(\Delta\mathbf{u}, \Delta x) \\ &= J^*(\mathbf{u}, x + \Delta x) + Q_j(\Delta\mathbf{u}, \Delta x) - (J^*(\mathbf{u}, x + \Delta x) \\ &\quad - J^*(\mathbf{u}, x)) \\ &= J^*(\mathbf{u}, x + \Delta x) + Q_j(\Delta\mathbf{u}, \Delta x) - \gamma \end{aligned} \quad (28)$$

Moreover, taking into account (22) it holds that:

$$J^s(x + \Delta x) \geq J^*(\mathbf{u}, x + \Delta x) + Q_j(\Delta\mathbf{u}, \Delta x) - \gamma \quad (29)$$

But, from the definition of J^s and J^* and because \mathbf{u} can be suboptimal for $x + \Delta x$, it can be seen that:

$$J^s(x + \Delta x) \leq J^*(\mathbf{u}, x + \Delta x) \quad (30)$$

This implies that $Q_j(\Delta\mathbf{u}, \Delta x) - \gamma$ in (29) must be less or equal to zero, thus:

$$Q_j(\Delta\mathbf{u}, \Delta x) \leq \gamma \quad (31)$$

Constraint (31) states a condition to be fulfilled by all active vertices for x , and it is a conservative bound for $\Delta\mathbf{u}$.

From Proposition 1 it is inferred that $2g_{u,m}$ is the gradient of $J_m(\mathbf{u}, x)$ with respect to u . At the minimizer of $J^*(\mathbf{u}, x)$, for any $\Delta\mathbf{u}$ (and in particular for $\Delta\mathbf{u} = bu(x + \Delta x) - \mathbf{u}(x)$), there is at least one vertex $m \in I(x)$ such that ([34]):

$$g_{u,m}^T \Delta\mathbf{u} \geq 0 \quad (32)$$

Let m be such an active vertex. Then, $\Delta\mathbf{u}$ verifies (31) and (32). Thus, from the definition of $R_{k,m}$ (19):

$$Q_k(\Delta\mathbf{u}, \Delta x) - Q_m(\Delta\mathbf{u}, \Delta x) \leq R_{k,m} \leq \max_{i \in I(x)} R_{k,i} \quad (33)$$

On the other hand, taking into account (26) (which hold for any active vertex k) and (33):

$$J^s(x) - J_k(\mathbf{u}, x) \leq \max_{i \in I(x)} R_{k,i} \quad (34)$$

will hold for all $k \in I(x + \Delta x)$. Then, the set of all k for which (34) holds will contain all the active vertices for $I(x + \Delta x)$, thus $I(x + \Delta x) \subseteq I_e(x)$ and this completes the proof. \square

Theorem 1 can be used to provide a conservative estimation of $I(x + \Delta x)$. However, solving (19) can be time consuming, so a more efficient way of computing I_e has to be developed.

It can easily be seen that if z satisfies $Q_i(z, \Delta x) \leq \gamma$ and $g_{u,i}^T z \geq 0$ then it also verifies:

$$Q_i(z, \Delta x) - 2g_{u,i}^T z \leq \gamma \quad (35)$$

which can be expressed as (recall Proposition 1):

$$z^T M z + 2\Delta x^T N^T z + \Delta x^T C \Delta x + 2g_{x,i}^T \Delta x \leq \gamma \quad (36)$$

On the other hand:

$$Q_k(z, \Delta x) - Q_i(z, \Delta x) = c_{k,i}^T z + d_{k,i} \quad (37)$$

where $c_{k,i} = 2(g_{u,k} - g_{u,i}) = 2G_u^T G_\theta(\theta_k - \theta_i)$ and $d_{k,i} = 2(g_{x,k} - g_{x,i})^T \Delta x = 2(\theta_k - \theta_i)^T G_\theta^T F_x \Delta x$.

Constraint (36) can be used to get a conservative bound of the $R_{k,i}$ in (19) as shown below:

$$\begin{aligned} R_{k,i} &\leq \max_z c_{k,i}^T z + d_{k,i} \\ &\quad s.t. \\ &\quad z^T M z + 2\Delta x^T N^T z + \Delta x^T C \Delta x + 2g_{x,i}^T \Delta x - \gamma \leq 0 \end{aligned} \quad (38)$$

The estimation of $I(x + \Delta x)$ obtained using (38) is more conservative than the one obtained using (19) as it can include more vertices in $I_e(x + \Delta x)$. However, problem (38) can be solved very efficiently, and a closed solution can be provided as it is shown in the following.

Applying a known result ([7]), the quadratic constraint of (38) can be expressed as an ellipsoid:

$$(z - a_i)^T A^{-1} (z - a_i) \leq 1 \quad (39)$$

where:

$$A = (\Delta x^T N^T M^{-1} N \Delta x - \Delta x^T C \Delta x - 2g_{x,i}^T \Delta x + \gamma) M^{-1} \quad (40)$$

$$\begin{aligned} &= \sigma_i M^{-1} \\ a_i &= -M^{-1} N \Delta x \end{aligned} \quad (41)$$

It is noteworthy that for small Δx , γ is small (due to the continuity of J^*). Also, all terms multiplied by Δx in σ_i are small for small Δx . This implies that the ellipsoid containing $\Delta \mathbf{u}$ is small, so an estimation of $I(x + \Delta x)$ with a low degree of conservativeness will be obtained. Ellipsoid (39) can be described as:

$$(z - a_i)^T \frac{M}{\sigma_i} (z - a_i) \leq 1 \quad (42)$$

A new variable, μ , is defined as follows:

$$\mu \doteq \frac{M^{\frac{1}{2}}}{\sqrt{\sigma_i}} (z - a_i) \quad (43)$$

thus:

$$z = \sqrt{\sigma_i} M^{-\frac{1}{2}} \mu + a_i \quad (44)$$

using (43) and (44), problem (38) can be rewritten as:

$$\begin{aligned} R_{k,i} &\leq \max_{\mu} c_{k,i}^T (\sqrt{\sigma_i} M^{-\frac{1}{2}}) \mu + c_{k,i}^T a_i + d_{k,i} \\ &\quad s.t. \\ &\quad \mu^T \mu \leq 1 \end{aligned} \quad (45)$$

Problem (45) can be solved analytically and its solution becomes:

$$\mu^* = \frac{M^{-\frac{1}{2}} c_{k,i}}{\sqrt{c_{k,i}^T M^{-1} c_{k,i}}} \quad (46)$$

Substituting into (45):

$$R_{k,i} \leq \sqrt{\sigma_i c_{k,i}^T M^{-1} c_{k,i}} + c_{k,i}^T a_i + d_{k,i} \quad (47)$$

With this bound of $R_{k,i}$ a more conservative, but easily computable, estimation of $I(x + \Delta x)$ can be obtained as stated in the following corollary:

Corollary 1 *A computationally efficient estimation of $I(x + \Delta x)$ (that is $I(x + \Delta x) \subseteq I_e^*(x + \Delta x)$) can be obtained using the following rejection criterion:*

$$I_e^*(x + \Delta x) = \left\{ k : J^s(x) - J_k(\mathbf{u}, x) \leq \max_{i \in I(x)} \sqrt{\sigma_i c_{k,i}^T M^{-1} c_{k,i}} + c_{k,i}^T a_i + d_{k,i} \right\} \quad (48)$$

where \mathbf{u} is the optimal solution for x (usually the optimal solution for the previous sampling time), $c_{k,i}$, $d_{k,i}$ are computed as in (37), and σ_i and a_i are computed from (40) and (41) respectively.

Finally it follows a simple analysis of the expected performance of this strategy. Suppose that the number of vertices contained in $I_e(x + \Delta x)$ is $\frac{1}{\alpha}2^L$. Extensive simulation tests shows that α is usually between 20 and 90. Moreover, it is assumed that the estimated number operations for a min-max problem depends mainly on the number of vertices to be considered⁵. For the full min-max problem (2^L vertices) let that number be CP_1 . Thus, for the reduced min-max problem, the number of operations can be approximated by $\frac{CP_1}{\alpha}$. In the case of the efficient approach described in this paper, it has to be taken into account CI_e operations necessary to compute $I_e^*(x + \Delta x)$. So the speed up factor between the original formulation and the efficient approach will be:

$$\text{speed up} = \frac{CP_1}{\frac{CP_1}{\alpha} + CI_e} \quad (49)$$

In many different simulations it has been found that the time required to obtain I_e is always lower than that required to solve the reduced min-max problem. Thus a conservative estimation for CI_e is $\frac{CP_1}{\alpha}$ leading to:

$$\text{speed up} \geq \frac{\alpha}{2} \quad (50)$$

Moreover, it can happen that almost all vertices are rejected, e.g. if Δx is small. In this case, the computational burden of the reduced problem is very little and the number of operations required for the efficient approach is dominated by CI_e . It can be seen from (48) that the number of operations needed to compute CI_e is of the same order of a single evaluation of J^* , thus the speed up factor will depend on the number of evaluations needed by the minimization algorithm to solve the min-max problem (it depends on the method used). These results will be illustrated in the example presented in the next section.

Remark 1 *Although simulation tests show that α is usually greater than 35, there is no guarantee that the number of rejected vertices will be enough. If α becomes too low to allow the realtime solution of the min-max problem a simple workaround can be used. Instead of solving one min-max problem for*

⁵ The complexity of a min-max problem like that of Min-Max MPC lies in the inner maximization problem, which is in itself NP-hard (and must be solved for every candidate solution of the outer minimization problem). In this maximization problem, operational and stability constraints do not take part, so its computational burden depends only on the number of vertices to be explored. This imply that the speed-up should be computed only in terms of the vertices that are not rejected.

$I_e^*(x + \Delta x)$, a serie of successive approximations from x to $x + \Delta x$ can be used, computing at each one I_e^* and solving a simpler min-max subproblem. This can be improved even more if a historian with previous values of x and its associated $\{\mathbf{u}(x), I(x)\}$ is kept. In this case some of the subproblems (closer to $x + \Delta x$) could be picked from those that are already solved.

5 Example

The results obtained in Section 4 will be illustrated with a simulation example. The example chosen is that of a very typical case in the process industry, an integrating process, which can be found for example in temperature control of the liquid in a tank. This kind of processes are usually harder to control than those that are open-loop stable. The prediction model for this example is:

$$G(s) = \frac{1}{s(2s + 1)} \quad (51)$$

An integrated uncertainties prediction model is obtained sampling (51) with a sampling time of 0.2. In this case the process state is defined by $[y_k \ y_{k-1} \ \Delta u_{k-1}]^T$. The controller parameters were $N_u = 7$, $N = 15$, $\lambda = 5$ and $\epsilon = 0.2$. The controller has been applied using both the standard formulation and the proposed strategy. Figure 3 shows the plant output, the percentage of rejected vertices at each sampling time (i.e., those vertices not in I_e^*) and the ratio (t_r) between the computation time of the original formulation and the proposed strategy. In the simulation a random disturbance has been added to the plant output to simulate more realistic conditions. At sampling time $t = 75$ a stationary step disturbance suddenly hits the process output, causing the great deviation from the set point seen in the output plot. In this case the percentage of rejected vertices is always higher than 90%. The average value of α is 43.4. The average computation time using the proposed strategy is 50 times lower than that using the standard formulation (well over the lower bound for the speed up factor). Thus, the proposed strategy achieves a remarkable improvement over the standard formulation of MMMPC. Note that the ratio of rejected vertices is lower when Δx is greater.

Also, the proposed Min-Max MPC has been compared with a regular MPC. An error of -15% in the estimation of the process gain has been considered together with a random plus step disturbance. The results are shown in Figure 4. Note that the controller parameters and the disturbance are the same for both controllers. It can be seen that the results of the MMMPC are better, especially when the step disturbance hits the process output at sampling time $t = 200$. Note how the MMMPC is able to react faster to the uncertainty or disturbance effects obtaining a better performance with just a moderately

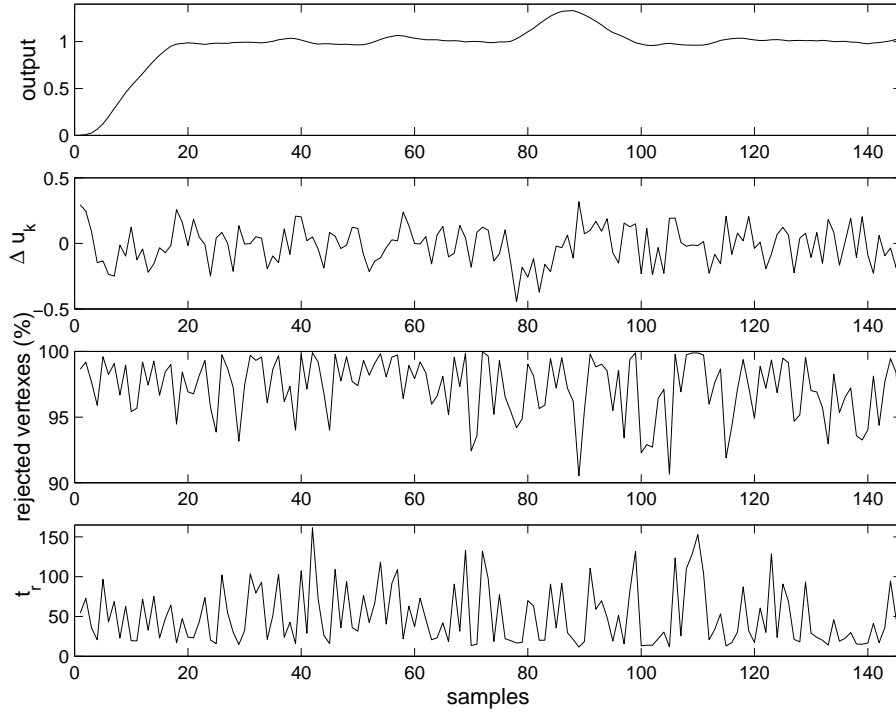


Fig. 3. Results of the simulation example.

higher level of the control signal. Other comparisons between MMMPC and MPC with simulated or experimental results can be found in [33] and [14].

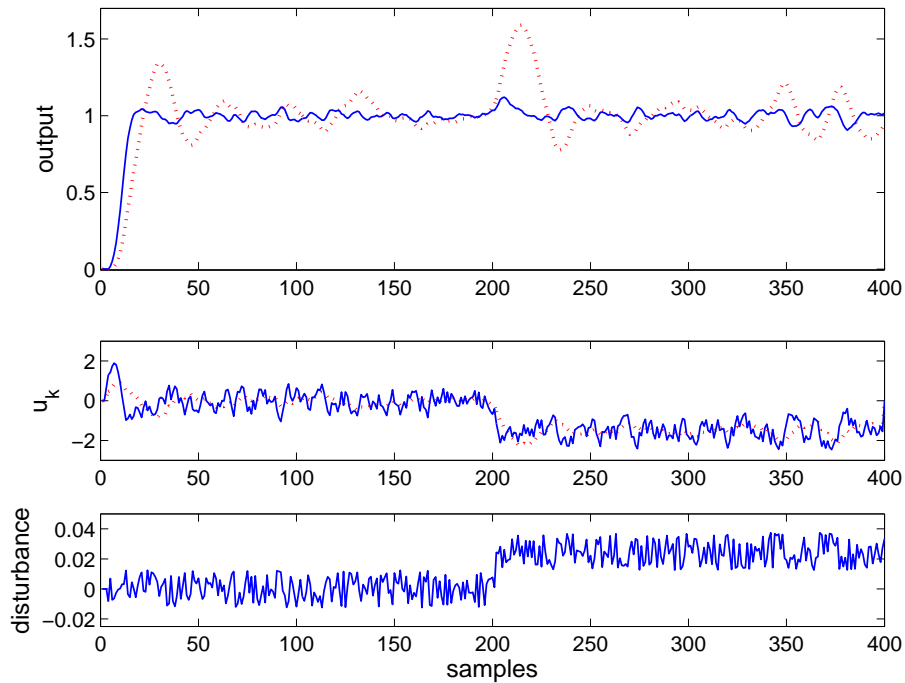


Fig. 4. Results of the comparison between the Min-Max MPC (solid plots) and regular MPC (dotted plots). Disturbance is the same for both controllers.

Description	Transfer function	Ts	N=10	N=12	N=14	N=16
First order system	$\frac{10}{100s+1}$	10	37.6	71.1	121.4	262.7
Integrator system	$\frac{1}{s}$	1	33.7	55	115.5	226.5
Inverse response system	$\frac{-200s+7}{1000s^2+110s+1}$	13	37.8	54.9	173.8	257.9
Undamped system with 53% overshoot	$\frac{26}{s^2+2s+26}$	0.05	34.7	46.3	90.1	206.4
Double integrator	$\frac{1}{s^2}$	1	27.9	41.7	71.5	147.1
Induction motor position control [25]	$\frac{168.0436}{s(s^2+25.921s+168.0436)}$	0.05	35.2	74.1	144.3	203.2

Table 1

Average value of α for different test systems and different values of the prediction and control horizons (it has been assumed $N = N_u$). The strategy described in Remark 1 has not been used. Sampling time Ts expressed in seconds.

Finally, the proposed strategy has been applied to different test systems and the average value of α for different prediction and control horizons can be seen in Table 1. For that simulation tests, the parameters are the same as in the previous example, except the sampling time (which cannot be the same for all systems) and the prediction and control horizons. It is noteworthy that α increases with higher values of N meaning that the cardinal of I_e^* does not grow at the same rate of that of $\text{vertices}(\Theta)$. On the other hand, the values of α are lower for systems that are more difficult to control (such a double integrator), but even in these cases, α is high enough to ensure that more than 95% of the vertices of Θ are rejected.

6 Conclusions

An efficient implementation of the MMMPC control law has been presented. The results presented in this paper broaden the range of processes to which, in practice, MMMPC can be applied. The strategy proposed in this work ensures much lower computational burden in most cases. This is achieved by reducing the number of vertices to be considered in the min-max problem. Simulation results show that the number of vertices can be reduced between 40 and 90 times on typical values of the prediction and control horizons.

However, some open questions remain to be addressed. The strategy presented here is based on a conservative estimation of the set of active vertices for the current process state. The less conservative the estimation, the more efficient the implementation. So, it is worth investigating how a tighter estimation of $I(x + \Delta x)$ can be obtained.

A Constraint handling

This paper deals with the efficient implementation of unconstrained Min-Max MPC. The optimization problem structure is different for the constrained formulation and the procedures presented here cannot be applied to it. However, both strategies are strongly related.

Here a min max controller which takes into account constraints is proposed. The corresponding optimization problem is convex and can be solved efficiently with existing optimization methods.

This controller is based on evaluating the control correction effort that makes the optimal solution for the unconstrained problem feasible and minimizes an upper bound of the min max cost function. The future control inputs are defined as $\mathbf{u} = \mathbf{u}_E(x) + \mathbf{v}$ where $\mathbf{u}_E(x)$ is the optimum solution for the unconstrained problem and $\mathbf{v} = [v_k \cdots v_{k+N_u-1}]^T$ are the control correction efforts. Furthermore, suppose that $\Theta = \{\forall \boldsymbol{\theta} \in \mathbb{R}^L : \|\boldsymbol{\theta}\|_\infty \leq \epsilon\}$.

By definition:

$$\begin{aligned}\mathbf{u}_E(x) &= \arg \min_{\mathbf{u}} \max_{\boldsymbol{\theta} \in \Theta} J(\boldsymbol{\theta}, \mathbf{u}, x) \\ J^s(x) &= \min_{\mathbf{u}} \max_{\boldsymbol{\theta} \in \Theta} J(\boldsymbol{\theta}, \mathbf{u}, x) = \max_{\boldsymbol{\theta} \in \Theta} J(\boldsymbol{\theta}, \mathbf{u}_E(x), x)\end{aligned}$$

The original constrained optimization problem can be posed as:

$$\begin{aligned}\min_{\mathbf{u}} \max_{\boldsymbol{\theta} \in \Theta} J(\boldsymbol{\theta}, \mathbf{u}, x) \\ s.t. \Gamma_u \mathbf{u} + \Gamma_x x + \Gamma_\theta \boldsymbol{\theta} \leq q, \quad \forall \boldsymbol{\theta} \in \Theta\end{aligned}$$

Notice that this formulation allows the inclusion of a terminal region constraint, and so may include all the ingredients for stability design and analysis of MPC controllers (see [29]).

The optimum solution for the unconstrained problem can be used to propose a modified problem with guaranteed performance. Taking into account Proposition 1:

$$J(\boldsymbol{\theta}, \mathbf{u}_E(x) + \mathbf{v}, x) = J(\boldsymbol{\theta}, \mathbf{u}_E(x), x) + \mathbf{v}^T M \mathbf{v} + 2\mathbf{v}^T (Nx + M\mathbf{u}_E(x) + G_u^T G_\theta \boldsymbol{\theta})$$

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

$$\begin{aligned} \max_{\boldsymbol{\theta} \in \Theta} J(\boldsymbol{\theta}, \mathbf{u}, x) &\leq \max_{\boldsymbol{\theta} \in \Theta} J(\boldsymbol{\theta}, \mathbf{u}_E(x), x) + \max_{\boldsymbol{\theta} \in \Theta} \mathbf{v}^T M \mathbf{v} \\ &\quad + 2\mathbf{v}^T (Nx + M\mathbf{u}_E(x) + G_u^T G_\theta \boldsymbol{\theta}) \end{aligned}$$

and both functions can be evaluated on-line using the solution of the unconstrained problem.

$$\max_{\boldsymbol{\theta} \in \Theta} J(\boldsymbol{\theta}, \mathbf{u}, x) \leq J^s(x) + \mathbf{v}^T M \mathbf{v} + 2\mathbf{v}^T (Nx + M\mathbf{u}_E(x)) + 2\epsilon \|G_\theta^T G_u \mathbf{v}\|_1$$

Then, the max function is bounded by a convex function that can be easily evaluated. It is also important to note that when \mathbf{v} is equal to zero, $\mathbf{u} = \mathbf{u}_E(x)$ satisfies the constraints and the exact value of the max function is obtained.

Note that $V_E(x)$ does not depend on \mathbf{v} so the proposed problem is:

$$\begin{aligned} \min_{\mathbf{v}} \mathbf{v}^T M \mathbf{v} + 2\mathbf{v}^T Nx + 2\mathbf{v}^T M\mathbf{u}_E(x) + 2\epsilon \|G_\theta^T G_u \mathbf{v}\|_1 \\ \text{s.t. } \Gamma_u \mathbf{v} + \Gamma_x x + \Gamma_\theta \boldsymbol{\theta} \leq q - \Gamma_u \mathbf{u}_E(x) \quad \forall \boldsymbol{\theta} \in \Theta \end{aligned}$$

Using standard techniques for robust constraint satisfaction (see [19]) the proposed constrained problem can be posed as:

$$\begin{aligned} \min_{\mathbf{v}} \mathbf{v}^T M \mathbf{v} + 2\mathbf{v}^T G_u^T F_x x + 2\mathbf{v}^T M\mathbf{u}_E(x) + 2\epsilon \|G_\theta^T G_u \mathbf{v}\|_1 \\ \text{s.t. } \Gamma_u \mathbf{v} + \Gamma_x x \leq q - \Gamma_u \mathbf{u}_E(x) - d\epsilon \end{aligned} \tag{A.1}$$

where only the constraints have been changed and the components of d are the 1-norm of the corresponding rows of Γ_θ . Therefore, the proposed constrained min-max problem can be solved efficiently by computing the solution of the unconstrained problem (using the approach presented in this paper) and then solving the QP problem (A.1). Note that using standard QP solvers, the computational burden of a QP problem like (A.1) is little compared to that of the min-max problem.

References

- [1] T. Alamo, D.R. Ramirez, and E.F. Camacho. Efficient implementation of constrained min-max model predictive control with bounded uncertainties: A vertex rejection approach. *Journal of Process Control*, 15:149–158, 2005.

- [2] A. Alessio and A. Bemporad. *A survey on explicit model predictive control*, volume 384 of *Lecture Notes in Control and Information Sciences*. 2009.
- [3] F. Bayat, T. A. Johansen, and A. A. Jalali. Using hash tables to manage the time-storage complexity in a point location problem: Application to explicit model predictive control. *Automatica*, 47(3):571–577, 2011.
- [4] M.S. Bazaraa and C.M. Shetty. *Nonlinear Programming. Theory and Algorithms*. John Wiley & Sons, 1979.
- [5] A. Bemporad, F. Borrelli, and M. Morari. Min-max Control of Constrained Uncertain Discrete-Time Linear Systems. *IEEE Transactions on Automatic Control*, 48(9):1600–1606, 2003.
- [6] T. Besselmann, J. Löfberg, and M. Morari. Explicit model predictive control for systems with linear parameter-varying state transition matrix. In *IFAC Proceedings Volumes (IFAC-PapersOnline)*, volume 17, 2008.
- [7] S. Boyd, L.E. Ghaoui, E. Feron, and V. Balakrishnan. *Linear matrix inequalities in system and control theory*. SIAM, Philadelphia, 1994.
- [8] E. F. Camacho and Manuel Berenguel. Robust Adaptive Model Predictive Control of a Solar Plant with Bounded Uncertainties. *International Journal of Adaptive and Signal Processing*, 11:311–325, 1997.
- [9] E.F. Camacho and C. Bordóns. *Model Predictive Control*. Springer-Verlag, second edition, 2004.
- [10] P.J. Campo and M. Morari. Robust Model Predictive Control. In *Proc. American Control Conference*, pages 1021–1026, June 10-12 1987.
- [11] A. Casavola, M. Giannelli, and Edoardo Mosca. Min-Max Predictive Control strategies for input-saturated polytopic uncertain systems. *Automatica*, 36:125–133, 2000.
- [12] Alessandro Casavola, Domenico Famularo, and Giuseppe Franzé. Robust constrained predictive control of uncertain norm-bounded linear systems. *Automatica*, 40(11):1865 – 1876, 2004.
- [13] F. A. Cuzzola, J. C. Geromel, and M. Morari. An improved approach for constrained robust Model Predictive Control. *Automatica*, 38(7):1183–1190, 2002.
- [14] D. Muñoz de la Peña, D.R. Ramírez, E.F. Camacho, and T. Álamo. Application of an explicit min-max MPC to a scaled laboratory process. *Control Engineering Practice*, 13:1463–1471, 2005.
- [15] BaoCang Ding, YuGeng Xi, Marcin T. Cychowski, and Thomas O’Mahony. Improving off-line approach to robust MPC based-on nominal performance cost. *Automatica*, 43(1):158 – 163, 2007.
- [16] A. Grancharova and T. A. Johansen. Computation, approximation and stability of explicit feedback min-max nonlinear model predictive control. *Automatica*, 45(5):1134–1143, 2009.

- [17] Lars Imsland, Nadav Bar, and Bjarne A. Foss. More efficient predictive control. *Automatica*, 41(8):1395 – 1403, 2005.
- [18] T. Johansen and A. Grancharova. Approximate explicit constrained linear model predictive control via orthogonal search tree. *IEEE Transactions on Automatic Control*, 48(5):810–815, 2003.
- [19] E.C. Kerrigan. *Robust Constraint Satisfaction: Invariant Sets and Predictive Control*. PhD thesis, University of Cambridge, 2000.
- [20] E.C. Kerrigan and J.M. Maciejowski. Feedback min-max model predictive control using a single linear program: Robust stability and the explicit solution. *International Journal of Robust Nonlinear Control*, 14:295–413, 2004.
- [21] Y.H. Kim and W.H. Kwon. An application of min-max generalized predictive control to sintering processes. *Control Engineering Practice*, 6:999–1007, 1998.
- [22] M.V. Kothare, V. Balakrishnan, and M. Morari. Robust constrained model predictive control using linear model inequalities. *Automatica*, 32(10):1361–1379, 1996.
- [23] B. Kouvaritakis, J.A. Rossiter, and J. Schuurmans. Efficient robust predictive control. *Automatic Control, IEEE Transactions on*, 45(8):1545–1549, 2000.
- [24] J.H. Lee and Zhenghong Yu. Worst-case formulations of model predictive control for systems with bounded parameters. *Automatica*, 33(5):763–781, 1997.
- [25] G.M. Liaw and F.J. Lin. A robust speed controller for induction motor drives. *Industrial Electronics, IEEE Transactions on*, 41(3):308 –315, jun 1994.
- [26] D. Limón, T. Alamo, J.M. Bravo D.M. Raimondo, D. Muñoz de la Peña, A. Ferramosca, and E.F. Camacho. Input-to-state stability: a unifying framework for robust model predictive control. In *Nonlinear Model Predictive Control: Towards New Challenging Applications (LNCIS 384)*, pages 1–26. Springer-Verlag, 2009.
- [27] Y. Lu and Y. Arkun. Quasi-Min-Max MPC algorithms for LPV systems. *Automatica*, 36(4):527–540, 2000.
- [28] J. M. Maciejowsky. *Predictive Control with Constraints*. Addison Wesley, 2002.
- [29] D.Q. Mayne, J.B. Rawlings, C.V. Rao, and P.O.M. Scokaert. Constrained model predictive control: Stability and optimality. *Automatica*, 36:789–814, 2000.
- [30] D. Muñoz de la Peña, T. Alamo, D. R. Ramírez, and E. F. Camacho. Min-max model predictive control as a quadratic program. *IET Control Theory and Applications*, 1(1):328–333, 2007.
- [31] D. Muñoz de la Peña, D. R. Ramírez, E. F. Camacho, and T. Alamo. Explicit solution of min-max mpc with additive uncertainties and quadratic criterion. *Systems and Control Letters*, 55(4):266–274, 2006.

- [32] D.R. Ramirez, T. Alamo, E.F. Camacho, and D. Muñoz de la Peña. Min-Max MPC based on a computationally efficient upper-bound of the worst case cost. *Journal of Process Control*, 16:511–519, 2006.
- [33] D.R. Ramírez, M.R. Arahal, and E.F. Camacho. Min-Max Predictive Control of a Heat Exchanger using a Neural Network Solver. *IEEE Trans. on Control Systems Technology*, 12(5):776–786, 2004.
- [34] D.R. Ramirez and E.F. Camacho. Piecewise affinity of min-Max MPC with bounded additive uncertainties and a quadratic criterion. *Automatica*, 42:295–302, 2006.
- [35] J. B. Rawlings and D.Q. Mayne. *Model Predictive Control: Theory and Design*. Nob Hill Publishing, 2009.
- [36] P. Rodriguez-Ayerbe and S. Olaru. Off-line robustification of explicit control laws. In *IFAC Proceedings Volumes (IFAC-PapersOnline)*, volume 17, 2008.
- [37] J.A. Rossiter. *Model-Based Predictive Control: A Practical Approach*. CRC Press, 2003.
- [38] V. Sakizlis, N. M. P. Kakalis, J. D. Perkins V. Dua, and E. N. Pistikopoulos. Design of robust model-based controllers via parametric programming. *Automatica*, 40:189–201, February 2004.
- [39] V. Sakizlis, J. D. Perkins V. Dua, and E. N. Pistikopoulos. Robust model-based tracking control using parametric programming. *Computers & Chemical Engineering*, 28:195–207, January 2004.
- [40] P.O.M. Scokaert and D.Q. Mayne. Min-max feedback model predictive control for constrained linear systems. *IEEE Transactions on Automatic Control*, 43(8):1136–1142, 1998.
- [41] P. Tøndel, T. A. Johansen, and A. Bemporad. Evaluation of piecewise affine control via binary search tree. *Automatica*, 39(5):945–950, 2003.
- [42] S.M. Veres and J.P. Norton. Predictive Self-Tuning Control by Parameter Bounding and Worst Case Design. *Automatica*, 29(4):911–928, 1993.
- [43] V. Veselý, D. Rosinová, and M. Foltin. Robust model predictive control design with input constraints. *ISA transactions*, 49(1):114–120, 2010.
- [44] Z. Wan and M.V. Kothare. An efficient off-line formulation of robust model predictive control using linear matrix inequalities. *Automatica*, 39:837–846, 2003.