Proceedings of the 2006 American Control Conference
Minneapolis, Minnesota, USA, June 14-16, 2006

Feedback Min-Max Model Predictive Control Based on a Quadratic Cost Function

D. Muñoz de la Peña[†], T. Alamo[†], A. Bemporad[*] and E.F. Camacho[†]

Abstract— **Feedback min-max model predictive control based on a quadratic cost function is addressed in this paper. The main contribution is an algorithm for solving the min-max quadratic MPC problem with an arbitrary degree of approximation. The paper also introduces the "recourse horizon", which allows one to obtain a trade-off between computational complexity and performance of the control law. The results are illustrated by means of a simulation of a quadruple-tank process.**

Keywords: Predictive control for linear systems; Robust control; Optimization algorithms

I. INTRODUCTION

Most control strategies are based on a mathematical model of the process to be controlled. Using this model, the controller obtains the control input in such a way that a given cost criterion is minimized. This means that a controller's efficiency depends greatly on how precisely the mathematical model represents the real behavior of the system. This is even more critical in the case of model predictive control (MPC) where the control decision is taken on the base of the future predicted evolution of the system which is obtained using a model of the system assumed to be perfect in most cases [1].

One way to deal with uncertainties in MPC is to consider a worst case approach [2]. This approach is denoted min-max [3], [4], [5]. Recent works deal with feedback MPC [6], [7]. Feedback min-max MPC obtains a sequence of feedback control laws that minimizes the worst case cost while assuring robust constraint handling. It requires the solution of a very high dimensional problem that makes its practical implementation very hard.

For problems based on a cost function that can be evaluated with a linear programm (LP), the explicit solution has been obtained [8], [9], [10]. Lately, an efficient online algorithm was proposed by the authors in [11]. There are not any equivalent results in the literature for quadratic cost functions although several approximate solutions have been given (see [5], [6], [12], [13]).

In this paper, the algorithm presented in [11] is extended to deal with quadratic cost functions. The main contribution is that the modified version of the algorithm provides a feasible solution with an arbitrary degree of approximation. To the best knowledge of the authors there is no equivalent result in the literature. The algorithm is based on the structure and the convexity properties of the quadratic cost function. It applies a nested decomposition procedure to solve the min-max problems via a sequence of low order quadratic programs. This idea was first introduced by Benders in [14] for solving mixed integer problems.

However, the computational burden of the algorithm still grows exponentially with the length of the prediction horizon. In order to arbitrarily limit the possible combinatorial explosion, in this paper it is proposed to consider that the disturbance only acts on a "recourse horizon", which can be shorter than the prediction horizon. This idea is motivated by the engineering sense of not taking into account possible model uncertainties for long predictions because the MPC controller is implemented in a receding horizon way. This idea is related to the "control horizon" concept, which is generally used to keep manageable the number of free variables of the optimization problem.

II. PROBLEM FORMULATION

Consider the discrete time linear system

$$x(t+1) = \phi(x(t), u(t), w(t)), \quad (1)$$

with

$$\phi(x, u, w) = A(w)x + B(w)u + D(w),$$

subject to constraints

$$G_x x(t) + G_u u(t) \le g, \quad (2)$$

where $x(t) \in R^{n_x}$ is the state vector, $u(t) \in R^{n_u}$ is the input vector and $w(t) \in R^{n_w}$ is the uncertainty vector that is supposed to be bounded, namely $w(t) \in \mathcal{W}$ where $\mathcal{W}$ is a closed polyhedron. The system matrices are defined by the uncertainty as

$$\begin{aligned} A(w) &= A_0 + \sum_{k=1}^{n_w} e_k^T w A_k, \\ B(w) &= B_0 + \sum_{k=1}^{n_w} e_k^T w B_k, \\ D(w) &= \sum_{k=1}^{n_w} e_k^T w D_k, \end{aligned} \quad (3)$$

where e_k is the k-th column of the identity matrix of size n_w. This is a general description of uncertainty for linear systems and includes both parametric and additive uncertainties (see [5], [8]).

In general the complexity of Feedback min-max MPC [6], [7] grows in an exponential manner with the prediction horizon. In order to improve the relation between computational complexity and performance, we propose to assume

[†]Dep. de Ingeniería de Sistemas y Automática, Universidad de Sevilla, Spain.
[*]Dip. di Ingegneria dell'Informazione, University of Siena, Italy.
E-mail: davidmps@cartuja.us.es (D. Muñoz de la Peña), alamo@cartuja.us.es (T. Alamo), bemporad@unisi.it (A. Bemporad), eduardo@cartuja.us.es (E.F. Camacho)

that the disturbance only acts for a finite time $N_r \leq N$ denoted recourse horizon, where N is the overall prediction horizon. This choice has two different motivations. First, from an engineering viewpoint it makes sense not to take into account possible model uncertainties after a certain prediction time, as the control law is implemented in a receding horizon scheme. Second, by fixing $N_r < N$ the complexity of the problem (in terms of optimization variables and constraints) will only grow linearly with N, which is clearly advantageous from a computational viewpoint. Note that this idea is equivalent to using as a terminal cost, the optimal cost function of a nominal MPC controller.

The feedback min-max optimal control problem with a recourse horizon is defined by the following problem which optimum value is denoted $J^*(x)$,

$$\begin{aligned}&\min_{u(0),x(0)}\{L(u(0),x(0))+\max_{w(0)}\{\min_{u(1),x(1)}\{L(u(1),x(1))+\\&\max_{w(1)}\{\dots\{\min_{u(N_r-1),x(N_r-1)}\{L(u(N_r-1),x(N_r-1))+\\&\max_{w(N_r-1)}\{\min_{u(t),x(t),t\geq N_r}\{\sum_{t=N_r}^{N-1}L(x(t),u(t))+F(x(N))\}\dots\}\end{aligned} \tag{4}$$

subject to

$$\begin{aligned}&x(0)=x,\\&x(t+1)=\phi(x(t),u(t),w(t)),\ t=0,\dots,N_r-1,\\&x(t+1)=\phi(x(t),u(t),\bar{0}),\ t=N_r,\dots,N-1,\\&G_x x(t)+G_u u(t)\leq g,\ t=0,\dots,N,\ \forall w(t)\in\mathcal{W},\end{aligned}$$

where $L(x,u)$ and $F(x)$ are the stage and terminal cost functions respectively and are given by

$$\begin{aligned}L(x,u)&=x^TQx+u^TRu,\\F(x)&=x^TPx,\end{aligned}$$

with Q, R and P positive definite matrices.

The control law is applied in a receding horizon scheme. At each sampling time the problem is solved for the current state x and $J^*(x)$ is obtained. The controller applies the optimal control input for the first time step which is denoted $u(0)^*$. Note that this optimization problem is of very high complexity.

A. Worst Case Scenario Tree

In order to solve the min-max problem, not all possible values of the uncertainty (which leads to an infinite dimensional problem) have to be taken into account, but only the extreme realizations (i.e. the vertices of $\mathcal{W}$). This is a well known result (see [7], [11]). In this way, in order to keep the sequence of *decision-uncertainty realization*, the extreme realizations of the uncertainty can be represented in a “scenario tree” as in [7]. This tree is used to solve the min-max problem as a finite dimensional deterministic problem. The root node of the tree represents the initial time step $j=0$ and each new level of the tree stands for a new time step so each node has q children, one for each vertex of $\mathcal{W}$. Each node is then defined by an uncertainty vector $w_i \in \text{vert}(\mathcal{W})$ which characterizes the uncertainty realization from the parent node. Note that the uncertainty is only taken into account in the recourse horizon. This means, that the scenario tree is made of N_r levels, and that each leaf node, has the information of the predictions along the rest of the prediction horizon.

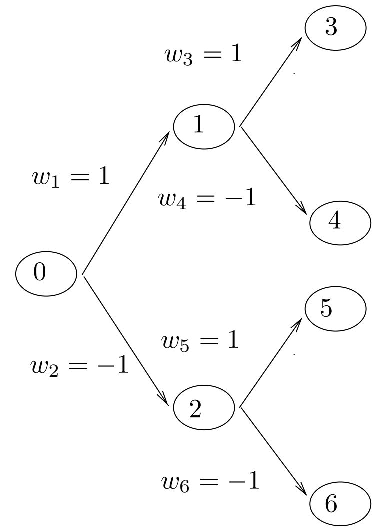


Fig. 1. Scenario tree with $N_r = 2$ and $q = 2$

All the nodes of the tree are numbered, starting from the root node (node 0) to the leaf nodes, stage by stage (so the enumeration of the nodes of a given stage is lower than their children nodes). M is the total number of nodes. Each node i has a set of children $I(i)$ and a parent node $p(i)$. The set of children is empty for the leafs nodes and the root node has no parent.

Figure 1 shows an example scenario tree with $N_r = 2$ and two possible uncertainties realizations ($w_i \in \{-1, 1\}$). The children sets I and the the parent nodes $p(i)$ are given by:

$$\left\{\begin{array}{l}I(0)=\{1,2\}\\I(1)=\{3,4\}\\I(2)=\{5,6\}\end{array}\right.,\left\{\begin{array}{l}p(1)=p(2)=0\\p(3)=p(4)=1\\p(5)=p(6)=2\end{array}\right..$$

The scenario tree is used to define an optimization problem that is equivalent to the min-max problem proposed in the previous section. To each node of the tree is assigned a set of variables and a cost to go function defined by the following optimization problem

$$\begin{aligned}\hat{V}_i(x_{p(i)},u_{p(i)}) \quad = \quad &\min_{x_i,u_i} L(x_i,u_i)+\max_{j\in I(i)}\hat{V}_j(x_i,u_i)\\\text{s.t.}\quad &x_i=\phi(x_{p(i)},u_{p(i)},w_i),\\&G_x x_i+G_u u_i\leq g.\end{aligned} \tag{5}$$

The index of the variables denotes node enumeration. The cost function $\hat{V}_i$ depends on the previous decision variables, i.e. the variables of the father node $x_{p(i)}, u_{p(i)}$ and on the cost function of its children.

To obtain the control input, the cost function of the root node is minimized for a given initial state x, i.e. the following optimization problem is solved:

$$\begin{array}{rcl} \hat{V}_0(x) & = & \min_{x_0,u_0} L(x_0,u_0) + \max_{j\in I(0)} \hat{V}_j(x_0,u_0) \\ & \text{s.t.} & x_0 = x, \\ & & G_x x_0 + G_u u_0 \le g. \end{array}$$

The definition of $\hat{V}_0(x)$ takes into account that the state of the root node is given by the measured state of the system.

The leafs node (nodes such that $I(i)$ is empty) are defined by the cost function of a nominal MPC control law. The cost to go $\hat{V}_i(x_{p(i)}, u_{p(i)})$ of each leaf node i is given by the following problem

$$\min_{u_i(t),x_i(t),t\ge N_r} \sum_{t=N_r}^{N-1} L(x_i(t), u_i(t)) + F(x_i(N))$$

subject to

$$\begin{array}{l} x_i(N_r) = \phi(x_{p(i)}, u_{p(i)}, w_i), \\ x_i(t+1) = \phi(x_i(t), u_i(t), \bar{0}),\ t = N_r, \ldots, N-1, \\ G_x x_i(t) + G_u u_i(t) \le g,\ t = N_r, \ldots, N. \end{array}$$

This problem optimizes the cost of the nominal trajectory from N_r to N as in standard MPC.

The most important difference between problems (4) and (5), is that in (5) all the parameters are deterministic. That is, each node has a corresponding known realization of the uncertainty and the maximization is done over the corresponding cost functions of the children nodes (which is a finite set). It is a multi-stage min-max quadratic program. In the following sections this kind of problems are introduced and an algorithm that exploits the problem structure is presented.

III. Multi Stage Min-Max Quadratic Programming

In this section the multi-stage min-max quadratic program in standard form is presented. Problem (5) can be formulated as a multi-stage min-max quadratic program. To pose it in standard form, auxiliary and slack variables have to be introduced as in linear programming (see [15]).

The multi-stage min-max quadratic problem in standard form is defined as

$$V_i^*(z_{p(i)}) = \min_{z_i} z_i^T H_i z_i + \max_{j\in I(i)} V_j^*(z_i) \quad (6a)$$

$$\text{s.t.} \quad W_i z_i = h_i - A_i z_{p(i)}, \quad (6b)$$

$$z_i \ge 0. \quad (6c)$$

with $H_i > 0$.

This kind of problems are defined by a scenario tree as the one introduced in the previous section. Each node i has a set of children $I(i)$ and a parent node $p(i)$. Note that this set is empty for the leafs nodes and the root node has no parent. Each node i is defined by matrices and vectors H_i, W_i, h_i and A_i. All these parameters are deterministic and can be different for each node. In the feedback min-max MPC case, the matrices and vectors H_i, W_i, h_i and A_i depend on the system, the cost function, the constraints and on the value of the uncertainty from the parent node to the node i (what in the previous section was defined as w_i). The initial state of the system defines the constraints in the root node which does not depends on any previous decision, namely h_0 depends on x. The objective is to minimize V_0^*, the cost function in the root node. The boundary conditions are given by the problem solved at each leaf node.

The set of variables z_i corresponding to each node includes the state, the input, and the slack variables needed to represent the feedback min-max problem in standard form. Note that because of the recourse horizon, the variables z_i of the leafs nodes are of a higher dimension than the rest of the nodes. Note also, that as the set I is empty for these nodes, the cost is defined as a deterministic quadratic programming (QP) problem.

IV. Nested Decomposition Algorithm

The general idea of decomposition algorithms was first introduced by Benders in [14] for solving mixed integer problems and has been successfully applied to stochastic programming [16], [17]. In this section the algorithm for solving multi-stage min-max linear programs based on Benders decomposition presented by the authors in [11] is extended to deal with the quadratic case.

At each step, the algorithm obtains a feasible set of variables for the original problem. These variables might not be optimal, but a bound on the error is given. This bound decreases at each iteration.

In order to obtain the feasible set of variables, at each step m, a subproblem is solved for each node. These subproblems have the same constraints on z_i, but additional variables and linear constraints are added to evaluate the maximization and approximate the functions $V_j^*(z_i)$ by an outer linearization, i.e. a lower bound that can be evaluated using a linear problem. These lower bounds are improved at each iteration and converge to the real values of $V_j^*(z_i)$.

The lower bounds are denoted by $V_i^m(z_{p(i)})$ and are defined as follows (problems P_i^m)

$$\min_{z_i,\theta_i,\theta_{i,j}} z_i^T H_i z_i + \theta_i \quad (7a)$$

$$\text{s.t.} \quad W_i z_i = h_i - A_i z_{p(i)}, \quad (7b)$$

$$z_i \ge 0, \quad (7c)$$

$$D_{i,j}^k z_i + \theta_{i,j} \ge d_{i,j}^k,\ \forall j \in I(i),\ k \le r_{i,j}^m, \quad (7d)$$

$$\theta_i \ge \theta_{i,j},\ \forall j \in I(i), \quad (7e)$$

$$\theta_i \ge 0, \theta_{i,j} \ge 0,\ \forall j \in I(i). \quad (7f)$$

Constraints (7b) and (7c) are the constraints (6b) and (6c) of the original problem. The rest of the constraints evaluate the maximization and the lower bounds on $V_j^m(z_i)$. Note that these constraints do not limit the feasible set of z_i.

Each $\theta_{i,j}$ is a lower bound on the value of $V_j^m(z_i)$, that is, the value of the cost function of the children. This value is obtained evaluating an outer linearization, namely constraints (7d). In this way, the following inequalities hold

$$V_j^*(z_i) \ge V_j^m(z_i) \ge \theta_{i,j},\ \forall j \in I(i).$$

At the first iteration $r_{ij}^1 = 0$ for all nodes $i = 0, \ldots M$. This means that for the first iteration, the value of $\theta_{i,j}$ of the children of each node i is considered to be zero (recall that $\theta_{i,j} \geq 0$). Each time a new optimality cut is added ($r_{i,j}^m$ increases), the approximation of $V_j^m(z_i)$ becomes tighter.

The following theorem shows how to define constraints (7d) in order to evaluate a lower bound of $V_j^m(z_i)$.

Theorem 1 (c.f. [15]): Define $D_i = \lambda_i^T A_i$ and $d_i = \lambda_i^T A_i z_{p(i)} + V_i^m(z_{p(i)})$, where λ_i are the dual variables of the equality constraints (7b) for a given $z_{p(i)}$ and $V_i^m(z_{p(i)})$ is the optimal value of the cost function. Then it holds that for all z,

$$V_i^m(z) \geq d_i - D_i z. \tag{8}$$

Proof: The dual problem D_i^m is defined as (note that strong duality holds):

$$\begin{aligned} V_i^m(z_{p(i)}) \quad = \quad & \max_{\lambda_i, s_i, \mu_{i,j}^k, \mu_j} g(z_{p(i)}, \lambda_i, s_i, \mu_{i,j}^k, \mu_j) \\ \text{s.t.} \quad & 1 - \sum_{j \in I(i)} \mu_j \geq 0, \\ & \mu_j - \sum_{k=1}^{r_{ij}^m} \mu_{ij}^k \geq 0, \ \forall j \in I(i), \\ & s_i, \mu_j, \mu_{i,j}^k \geq 0. \end{aligned} \tag{9}$$

where $\lambda_i, s_i, \mu_{i,j}^k, \mu_j$ are the dual variables corresponding to constraints (7b), (7c), (7d) and (7e) respectively and $g(z, \lambda_i, s_i, \mu_{i,j}^k, \mu_j)$ is the dual function for a given value of $z_{p(i)} = z$ defined as

$$-\tfrac{1}{2} f^T H_i^{-1} f + (h_i - A_i z)^T \lambda_i + \sum_{j \in I(i)} \sum_{k=1}^{r_{ij}^m} d_{ij}^k \mu_{ij}^k$$

with

$$f = W_i^T \lambda_i + \sum_{j \in I(i)} \sum_{k=1}^{r_{ij}^m} D_{ij}^{kT} \mu_{ij}^k + s_i.$$

The parameter $z_{p(i)}$ does not affect the constraints of the dual problem so given the set of optimal dual variables $\lambda_i, s_i, \mu_{i,j}^k, \mu_j$ for $z_{p(i)}$, the following inequality holds for all z

$$V_i(z) \geq g_(z, \lambda_i, s_i, \mu_{i,j}^k, \mu_j),$$

because the set of dual variables is feasible for (9) in z. Then taking into account this inequality, that

$$g(z, \lambda_i, s_i, \mu_{i,j}^k, \mu_j) - g(z_{p(i)}, \lambda_i, s_i, \mu_{i,j}^k, \mu_j),$$

is equal to $\lambda_i^T A_i z_{p(i)} - \lambda_i^T A_i z$, that $g(z_{p(i)}, \lambda_i, s_i, \mu_{i,j}^k, \mu_j) = V_i(z_{p(i)})$ and the definition of d_i and D_i, it is proved that (8) holds. ■

These optimality cuts are obtained from feasible solutions to the dual problem of P_i^m. Note that as the number of optimality cuts r_{ij}^m is increased at each step m for each children node j, the set of dual constraints of a previous optimum solution may not be optimal, but still remains feasible if new zero variables μ_{ij}^k of the new optimality cuts are added. This way, although problems P_i^m may differ on each iteration, the lower bounds on the optimal value remain valid.

Constraints (7e) evaluate the maximization over all the children of node i using an epigraph approach.

As in the previous section, when solving the root node, constraints (7b) are replaced by $W_0 z_0 = h_0$, because the root node has no parent. For the feedback min-max problem, h_0 depends on the initial state of the system x.

When solving a leaf node, variables $\theta_{i,j}$ and constraints (7d) are omitted because these nodes do not have children. Note that this means that by definition, $V_i^m(z) = V_i^*(z)$ for each leaf node and every algorithm iteration m.

The algorithm solves problems with relative complete recourse [16], i.e. feasibility of the root problem P_0^1 assures feasibility of all the problems of the nodes of the scenario tree for all steps m. For general problems, feasibility cuts can be added to the algorithm as in stochastic linear programming [16], [17]. Note that feedback min-max can be formulated to have relatively complete recourse (see [8], [11]).

Summing up, the general idea is to approximate the cost function of each node with an outer linearization from the leaf nodes, up to the root node. At each iteration, if the error on a node is greater than a given value, the algorithm adds a new optimality cut. The bound of the error is evaluated from the leaf nodes to the root node in a recursive way, taking into account that no approximation is done at the leaf nodes. As both problems are subject to the same constraints on z_i, a feasible solution of V_i^m, is a feasible solution for the original multi-stage min max problem.

The proposed algorithm is the following:

Algorithm 1: Proposed algorithm.

- $m = 0,\ r_{ij}^0 = 0,\ i = 1, \ldots, M-1,\ j \in I(i).$
- if P_0 is unfeasible
  - Multi-stage min-max problem is unfeasible.
  - End of the algorithm.
- do
  - $(V_0^m, z_0, \lambda_0, e_0) =$ `solvep` (0,-).
  - $m = m + 1.$
- while $e_0 > \epsilon$.

with `solvep` a recursive function that obtains for a given node i and a given father variable $z_{p(i)}$, the value of the cost function V_i^m, the variable z_i, the dual variables of the equality constraints λ_i and a bound on the error e_i for iteration m. Note that for the root node (node 0), there is no parent. This function is defined as follows:

$(V_i^m, z_i, \lambda_i, e_i) =$ `solvep` (i,$z_{p(i)}$)

- Solve P_i^m using $z_{p(i)}$ and obtain V_i^m, z_i, $\theta_{i,j}$ and λ_i.
- for $j \in I(i)$
  - $(V_j^m, z_j, \lambda_j, e_j) =$ `solvep` (j,z_i)
  - if $V_j^m(z_i) - \theta_{i,j} > 0$.
    $r_{i,j}^{m+1} = r_{i,j}^m + 1.$
    $D_{i,j}^{r_{i,j}^{m+1}} = \lambda_j^T A_j.$
    $d_{i,j}^{r_{i,j}^{m+1}} = \lambda_j^T A_j z_i + V_j^m(z_i).$
  - else
    $r_{i,j}^{m+1} = r_{i,j}^m.$
- end for.
- $e_i = -\theta_i + \max_{j \in I(i)} V_j^m(z_i) + e_j.$

Theorem 2: The solution obtained applying Algorithm 1, denoted z_0^*, is a feasible suboptimal solution of (6) and the following inequalities hold

$$V_0^m \leq V_0^* \leq V_0^m + e_0.$$

Proof: Feasibility of the solution is assured because both sets of problems are subject to the same constraints on z_i (recall constraints (6b)-(6c) and (7b)-(7c)).

In order to proof that the error is bounded by e_0, a lower and an upper bound of the optimal cost function are obtained. First we derive the lower bound.

Taking into account Theorem 1, the following inequality holds for all iteration m and node j

$$V_j^m(z) \geq \max_{k=1,\dots,r^m_{p(j),j}} d^k_{p(j),j} - D^k_{p(j),j} z. \tag{10}$$

For the leafs nodes, $I(i)$ is empty so it holds that $V_i^m(z)$ = $V_i^*(z)$ because in this case (6) and (7) are equal. Taking this into account and applying (10) backwards from the leaf nodes, it holds for all iteration m and node i

$$V_i^m(z) \leq V_i^*(z). \tag{11}$$

The upper bound is obtained in a recursive way. Suppose that for a given node i and a given variable z_i

$$V_j^*(z_i) \leq V_j^m(z_i) + e_j, \tag{12}$$

for all $j \in I(i)$. Then, the following expressions hold

$$\begin{aligned} &V_i^m(z_{p(i)}) = z_i^T H_i z_i + \theta_i, \\ &V_i^*(z_{p(i)}) \leq z_i^T H_i z_i + \max_{j\in I(i)} V_j^*(z_i), \\ &z_i^T H z_i + \max_{j\in I(i)} V_j^*(z_i) \leq z_i^T H_i z_i + \max_{j\in I(i)} V_j^m(z_i) + e_j, \end{aligned}$$

so taking into account the definition of e_i, the following inequality holds

$$V_i^*(z_{p(i)}) \leq V_i^m(z_{p(i)}) + e_i. \tag{13}$$

For the leafs nodes, (12) holds for $e_j = 0$. Applying (12) backwards to the root node, it is easy to see that

$$V_0^* \leq V_0^m + e_0.$$

■

Convergence of the algorithm is not proved in this paper. Future works will tackle this issue.

V. Quadruple-Tank Process

In this example, feedback min-max MPC is applied to a quadruple-tank process like the one presented in [18]. This plant has four connected tanks. See Figure 2 for a layout of the plant. The plant is made of $0.6m^2$ section tanks with normalized damping factors of $8.7932e-4m^2$. The 3-way valves have factors $\gamma_1 = 0.3$ and $\gamma_2 = 0.4$. (see [18] for a complete description of the dynamics)[1] .

The nonlinear model is linearized in the equilibrium point given by the following heights ($[m]$) and flows ($[m^3/h]$):

$$h_0 = \begin{bmatrix}0.22 & 0.43 & 0.20 & 0.45\end{bmatrix}^T, \quad q_0 = \begin{bmatrix}1.5 & 1.7\end{bmatrix}^T.$$

TABLE I

COMPUTATIONAL TIMES [S] FOR DIFFERENT RECOURSE HORIZONS FOR THE QUADRUPLE-TANK PROCESS.

$N = N_r$	2	3	4	5	6	7
	0.094	0.496	1.75	6.93	27.4	+2min

TABLE II

COMPUTATIONAL TIMES [S] FOR DIFFERENT RECOURSE AND PREDICTION HORIZONS FOR THE QUADRUPLE-TANK PROCESS.

N/N_r	2	3	4	5
5	0.11	0.51	1.90	6.93
10	0.14	0.61	2.56	9.19
15	0.16	0.71	3.20	13.86
20	0.19	0.85	3.80	16.84

and the following linear discrete time model with a sampling time $10s$ is obtained

$$\begin{aligned} x_{k+1} = &\begin{bmatrix} 0.8541 & 0 & 0.1032 & 0 \\ 0 & 0.9100 & 0 & 0.0503 \\ 0 & 0 & 0.8883 & 0 \\ 0 & 0 & 0 & 0.9473 \end{bmatrix} x_k \\ &+ \begin{bmatrix} 0.0129 & 0.0015 \\ 0.0008 & 0.0177 \\ 0 & 0.0262 \\ 0.0316 & 0 \end{bmatrix} u_k + \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ -1 & 0 \\ 0 & -1 \end{bmatrix} w_k. \end{aligned} \tag{14}$$

The state and the input are the error from the equilibrium point. The system has flow constrains $1.2 \leq (q_0 + u) \leq 2$ and state constraints

$$0 \leq \|h_0 + x\|_\infty \leq 1.$$

The uncertainty is restricted to the set $W = \{w : \|w\|_\infty \leq 0.01\}$. The weighting matrices are $Q = P = 10I,\ R = I$. Note that the complexity of the feedback formulation grows with 4^{N_r} because $n_w = 2$, rendering the computation of this control law a very hard problem.

The proposed algorithm is applied to implement a feedback min-max controller for this system[2]. Tables I and II show the mean computation time over a hundred different initial states of the proposed algorithm for different prediction and recursive horizon. The bound of the error is set to $\epsilon = 10^{-3}$. It can be seen that the computation time grows exponentially with the recourse horizon but not with the prediction horizon. As the sampling time is $10s$, a min-max controller with a recursion horizon of up to 5 and a prediction horizon of 10 can be applied. Note that a feasible solution with a bound on the error is available at any iteration.

A simulation is shown in figure V. The simulation lasts 40 time steps.The dashed lines represent the reference, while the full lines represent the simulated trajectories. Note that around time step 16, h_1 goes beyond the reference and the control law changes.

[1]This model belongs to a real quadruple tank process of the University of Seville.

[2]The simulations have been done with MATLAB 6.3 in an Athlon 2800 using CPLEX 9.1 as QP solver.

Fig. 2. Quadruple-tanks process.

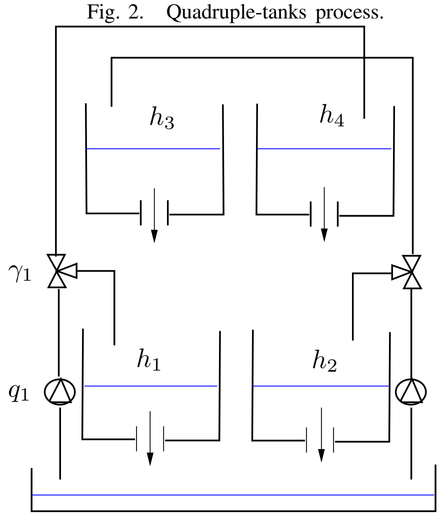
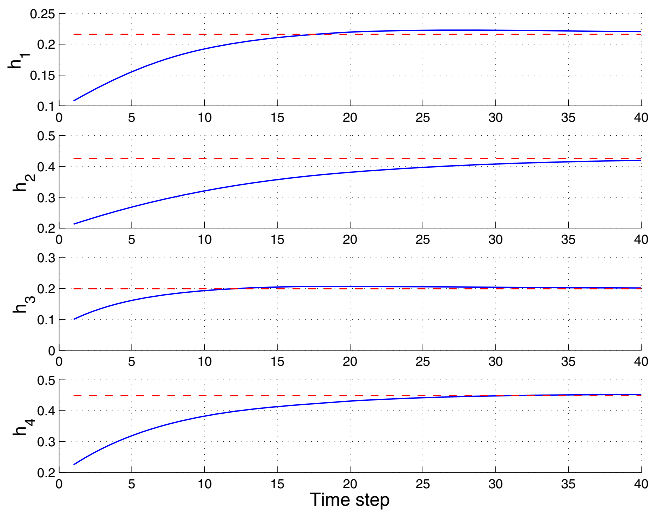
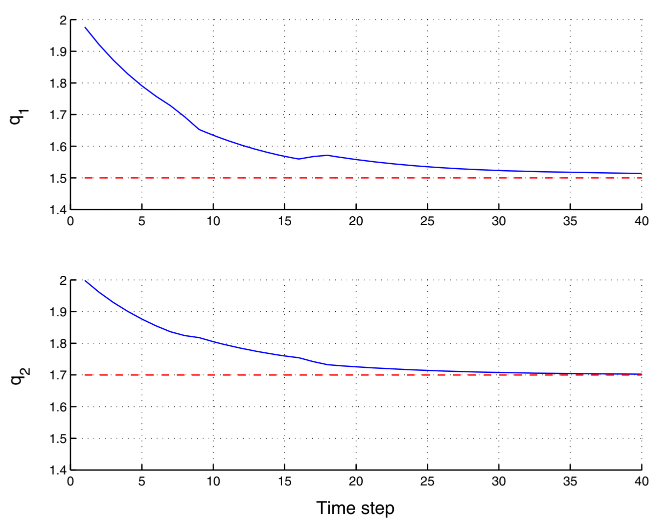


Fig. 3. Simulation results for the quadruple-tank process with a feedback min-max MPC with $Nr = 5$ and $N = 10$.



(a) Levels.



(b) Flows.

VI. Conclusions

Min-max MPC approaches are used to increase the robustness properties of a controller. However, most worst case approaches, have a great computational burden in common, specially in the case of feedback MPC. The algorithm presented in this paper along with the notion of the "recourse horizon" makes possible to implement for real plants an approximated feedback min-max control law based on a quadratic cost function. The approximation error bound is computed at each step of the algorithm and can be made arbitrarily small increasing the number of iterations.

It is important to remark that to the best knowledge of the authors, this is one of the few results available to implement the feedback min-max MPC controller presented. The computational times, although grow in an exponential manner, are manageable at least for the system under consideration.

Future works include the proof of the algorithm convergence, and an application to the real quadruple tank process.

References

[1] E. Camacho and C. Bordóns, *Model Predictive Control, 2nd Edition*. Springer-Verlag, 2004.

[2] H. S. Witsenhausen, "A min-max control problem for sampled linear systems," vol. 13, no. 1, pp. 5–21, 1968.

[3] P. Campo and M. Morari, "Robust model predictive control," vol. 2, 1987, pp. 1021–1026.

[4] J. Allwright and G. Papavasiliou, "On linear programming and robust model-predictive control using impulse-responses," *Systems & Control Letters*, vol. 18, pp. 159–164, 1992.

[5] M. Kothare, V. Balakrishnan, and M. Morari, "Robust constrained model predictive control using linear matrix inequalities," *Automatica*, vol. 32, pp. 1361–1379, 1996.

[6] J. Lee and Z. Yu, "Worst-case formulations of model predictive control for systems with bounded parameters," *Automatica*, vol. 33, no. 5, pp. 763–781, 1997.

[7] P. O. M. Scokaert and D. Q. Mayne, "Min-max feedback model predictive control for constrained linear systems," *IEEE Transactions on Automatic Control*, vol. 43, no. 8, pp. 1136–1142, 1998.

[8] A. Bemporad, F. Borrelli, and M. Morari, "Min-max control of constrained uncertain discrete-time linear systems," *IEEE Transactions on Automatic Control*, vol. 48, no. 9, pp. 1600–1606, September 2003.

[9] E.C.Kerrigan and J. Maciejowski, "Feedback min-max model predictive control using a single linear program: Robust stability and the explicit solution," *International Journal of Robust and Nonlinear Control*, vol. 14, pp. 395–413, 2004.

[10] M. Diehl and J. Björnberg, "Robust dynamic programming for min-max model predictive control of constrained uncertain systems," vol. 49, no. 12, pp. 2253–2257, 2004.

[11] D. Muñoz de la Peña, T. Alamo, and A. Bemporad, "A decomposition algorithm for feedback min-max model predictive control," in *Proceedings of the 44rd IEEE Conference on Decision and Control*, Seville, Spain, December 2005.

[12] Y. Wang and J. Rawlings, "A new robust model predictive control method i: theory and computation," *Journal of Process Control*, vol. 14, p. 231247, 2004.

[13] V. Sakizlis, N. Kakalis, V. Dua, J. Perkins, and E. Pistikopoulos, "Design of robust model-based controllers via parametric programming," *Automatica*, vol. 40, pp. 189–201, 2004.

[14] J. Benders, "Partitioning procedures for solving mixed-variables programming problems," *Numer. Math*, vol. 4, pp. 238–252, 1962.

[15] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge University Press, 2004.

[16] J. Birge and F. Louveaux, *Introduction to Stochastic Programming*. Springer, New York, 1997.

[17] ——, "A multicut algorithm for two-stage stochastic linear programs," *European J. Operational Research*, vol. 34, no. 3, pp. 384–392, 1988.

[18] K. Johansson, "The quadruple-tank process," *Transactions on Control Systems Technology*, 2000.