Proceedings of the
44th IEEE Conference on Decision and Control, and
the European Control Conference 2005
Seville, Spain, December 12-15, 2005

A Decomposition Algorithm for Feedback Min-Max Model Predictive Control

D. Muñoz de la Peña†, T. Alamo† and A. Bemporad*

Abstract— **An algorithm for solving feedback min-max model predictive control for discrete time uncertain linear systems with constraints is presented in the paper. The algorithm solves the corresponding multi-stage min-max linear optimization problem. It is based on applying recursively a decomposition technique to solve the min-max problem via a sequence of low complexity linear programs. It is proved that the algorithm converges to the optimal solution in finite time. Simulation results are provided to compare the proposed algorithm with other approaches.**

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

I. Introduction

Most control design methods need a model of the process to be controlled. These models are always subject to uncertainties and only describe the dynamics of the process in an approximate way. Model predictive control is one of the control strategies that is able to deal with the uncertainty in an explicit way. One approach used to design robust control laws is to minimize the cost function for worst case uncertainty realization. This is known as min-max as was first introduced by Witsenhausen in [1]. Early model predictive control approaches can be found in see [2], [3]. These works deal with open loop predictions and optimize a single sequence of control inputs for the worst possible uncertainty trajectory. Further results address the feedback min-max problem, where the optimization is done over a sequence of control laws in order to take into account that the control law is applied in a receding horizon manner. See [4], [5] and the references therein. In both formulations, the resulting min-max optimization problems can be computationally very demanding. Different strategies have been proposed in the literature to overcome this problem, see for example [6], [7], [8], [9], [10].

This paper deals with feedback min-max MPC based on a linear performance index, i.e. that can be cast on a linear program (LP). It is well known that this min-max problem can be solved using multi-parametric programming with a dynamic programming approach [11] or that it can be cast as a large scale linear program [12].

Work (partially) done in the framework of the HYCON Network of Excellence, contract number FP6-IST-511368.

†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 (Alberto Bemporad)

In this paper an algorithm to solve in a more efficient way the min-max problem is proposed. The algorithm is based on the structure and the convexity properties of the cost function. It applies a nested decomposition procedure to solve the min-max problems via a sequence of low order linear programs. The paper also proves that the algorithm converges to the optimal solution in finite time.

The paper is organized as follows: In Section II the feedback min-max problem is reviewed. In Section III is defined the multi-stage min-max linear programs, a general class of optimization problems. In Section IV the main contribution of the paper is presented. Some simulation results are given in Section V. The paper ends with a section of conclusions.

II. Problem Formulation

Consider the discrete time linear system

$$x(t+1) = A(w(t))x(t) + B(w(t))u(t) + D(w(t)), \quad (1)$$

subject to constraints

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

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 $w(t) \in \mathcal{W}$ here $\mathcal{W}$ is a closed polyhedron. The system matrices are defined by the uncertainty as

$$\begin{aligned} A(w(t)) &= A_0 + \sum_{k=1}^{n_w} e_k^T w(t) A_k, \\ B(w(t)) &= B_0 + \sum_{k=1}^{n_w} e_k^T w(t) B_k, \\ D(w(t)) &= D_0 + \sum_{k=1}^{n_w} e_k^T w(t) D_k, \end{aligned} \quad (3)$$

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 [11], [6]).

Feedback min-max MPC obtains a sequence of feedback control laws that minimizes the worst case cost, while assuring robust constraint handling. The feedback min-max optimal control problem is defined as

$$J_j^*(x(j)) = \min_{u(j)} J_j(x(j), u(j)) \quad (4)$$

subject to

$$\begin{aligned} & G_x x(j) + G_u u(j) \leq g, \\ & A(w(j))x(j) + B(w(j))u(j) + D(w(j)) \in \chi^{j+1}, \\ & \forall w(j) \in \mathcal{W}, \end{aligned} \quad (5)$$

0-7803-9568-9/05/$20.00 ©2005 IEEE

where

$$J_j(x(j),u(j)) = \|Qx(j)\|_p + \|Ru(j)\|_p + \max_{w(j)\in\mathcal{W}} J^*_{j+1}(A(w(j))x(j) + B(w(j))u(j) + D(w(j))). \quad (6)$$

The set χ^j is the set of states $x(j)$ such that (4) is feasible.

The optimization problem is defined for $j = 0,\dots,N-1$ where N is the prediction horizon. The boundary conditions are:

$$J^*_N(x(N)) = \|Px(N)\|_p, \quad \chi^N = \chi^f. \quad (7)$$

In (4)-(7), $p = 1$ or $p = \infty$, where $\|x\|_1$ and $\|x\|_\infty$ are the one and infinity norm respectively. The cost function is defined by nonsingular matrices Q, R and P. Region χ^f is the terminal region and is supposed to be a polyhedron. This constraint is used to assure stability, see [5], [12].

The control law is applied in a receding horizon scheme. At each sampling time the problem is solved for the current state x and $J^*_0(x)$ 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. In the following sections an algorithm for solving this problem in an efficient way is presented.

A. Worst Case Scenario Tree

Due to the convex nature of the prediction and cost function, the min-max problem can be solved taking into account not all possible values of the uncertainty (which leads to an infinite dimension problem), but only the extreme realizations (i.e. the vertices of $\mathcal{W}$).

The enumeration of all the possible extreme realizations of the uncertainty along the prediction horizon gives rise to what is called a *scenario tree* as in [5]. 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 $j = 0$. Each new level of the tree stands for a new time step and contains all possible extreme uncertainty trajectories, i.e. all the possible combinations of vertices of the uncertainty set $\mathcal{W}$ along the prediction horizon. Each node has q children, one for each vertex of $\mathcal{W}$. Each node is then defined by an uncertainty vector w_i which characterizes the uncertainty realization from the parent node. By definition, w_i is one of the vertices of $\mathcal{W}$.

A possible uncertainty trajectory defining an scenario is then a path from the root node of the tree, down to a leaf. 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. The time step of the node is denoted by $n(i)$. 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.

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 function $\hat{V}_i(x_{p(i)}, u_{p(i)})$ defined by the following optimization problem

$$\hat{V}_i(x_{p(i)},u_{p(i)}) = \min_{x_i,u_i} \|Qx_i\|_p + \|Ru_i\|_p + \max_{j\in I(i)} \hat{V}_j(x_i,u_i) \quad (8)$$

subject to

$$\begin{aligned} &x_i = A(w_i)x_{p(i)} + B(w_i)u_{p(i)} + D(w_i),\\ &G_x x_i + G_u u_i \le g,\\ &A(w_j)x_i + B(w_j)u_i + D(w_j) \in \chi^{n(i)+1},\ \forall j \in I(i). \end{aligned} \quad (9)$$

The index of the variables denotes node enumeration. The set $\chi^{n(i)+1}$ corresponds to the feasible set of the next step problem and is a polyhedron. The cost function $\hat{V}_i$ depends on the previous decision variables, i.e. the variables of the father node $x_{p(i)}, u_{p(i)}$. Note that the optimization variable x_i of each node is fixed by (9).

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

$$\hat{V}_0(x) = \min_{x_0,u_0} \|Qx_0\|_p + \|Ru_0\|_p + \max_{j\in I(0)} \hat{V}_j(x_0,u_0)$$

subject to

$$\begin{aligned} &x_0 = x,\\ &G_x x_0 + G_u u_0 \le g,\\ &A(w_j)x_0 + B(w_j)u_0 + D(w_j) \in \chi^1,\ \forall j \in I(0). \end{aligned}$$

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

$$\hat{V}_i(x_{p(i)},u_{p(i)}) = \min_{x_i} \|Px_i\|_p$$

subject to

$$x_i = A(w_i)x_{p(i)} + B(w_i)u_{p(i)} + D(w_i),$$

for all leaf nodes (nodes such that $I(i)$ is empty). The value of the cost function at the leaf nodes do not depend on any other node. Also, $\chi^N = \chi^f$ as in the previous section.

Problems (4) and (8) are equivalent and that the following equality holds

$$J^*_0(x) = \hat{V}_0(x).$$

Note that in problem (4) the index of the cost functions denote time step (i.e. J^*_i is the cost function at time step i). On the other hand, in problem (8) the index of the cost function denote node (i.e. $\hat{V}_i$ is the cost function at node i).

The most important difference between problems (4) and (8) is that in (8) 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 linear program. In the following section this kind of problems are introduced and an efficient algorithm for solving the problem is presented.

III. Multi Stage Min-Max Linear Programming

In this section the multi-stage min-max linear program in standard form is presented. The main result of the paper is an algorithm that solves this kind of problems in an efficient way. The standard form is introduced to simplify the notation. Problem (8) belongs to this kind of optimization problems so the algorithm presented in the following sections can be applied to obtain the optimal control input of the feedback min-max MPC controller.

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

$$\begin{aligned} V_i^*(z_{p(i)}) \quad &= \quad \min_{z_i} c_i^T z_i + \max_{j \in I(i)} V_j^*(z_i) && \text{(10a)} \\ &\text{s.t.} \quad W_i z_i = h_i - A_i z_{p(i)}, && \text{(10b)} \\ &\qquad z_i \geq 0. && \text{(10c)} \end{aligned}$$

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 $p(i)$ and is defined by matrices and vectors c_i, W_i, h_i and A_i. All these parameters are deterministic and can be different for each node. The objective is to minimize V_0^*, the cost function in the root node.

$$\begin{aligned} V_0^* \quad &= \quad \min_{z_0} c_0^T z_0 + \max_{j \in I(0)} V_j^*(z_0) \\ &\text{s.t.} \quad W_i z_0 = h_0, \\ &\qquad z_0 \geq 0. \end{aligned}$$

The boundary conditions are given by the problem solved at each leaf node.

$$\begin{aligned} V_i^*(z_{p(i)}) \quad &= \quad \min_{z_i} c_i^T z_i \\ &\text{s.t.} \quad W_i z_i = h_i - A_i z_{p(i)}, \\ &\qquad z_i \geq 0. \end{aligned}$$

The multi-stage min-max linear program is related to the multi-stage stochastic linear program [13].

Problem (8) can be formulated as a multi-stage min-max linear program because the one and the infinity norm can be evaluated using a linear program as done for the first time in [14]. To pose it in standard form, some auxiliary and slack variables have to be introduced. The matrices and vectors c_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 set of variables z_i corresponding to each node includes the state, the input, the auxiliary variables introduced to evaluate the cost function and the slack variables needed to represent the feedback min-max problem in standard form.

The constraints represent both the system constraints and the cost definition constraints. The value $V_i^*(z_{p(i)})$ represents the cost function at node i depending on the previous decision variable $z_{p(i)}$ (Recall $p(i)$ is the index of the parent node of i).

IV. Nested Decomposition Algorithm

In this section the algorithm for solving multi-stage min-max linear programs based on Benders decomposition is presented. This algorithm exploits the specific structure of the problem and the convexity properties of the objective function. The general idea of Benders decomposition is to project a problem in two vector variables, into a problem in one of these variables and then use a cutting plane method to handle the projected objective function. This idea was first introduced by Benders in [15] for solving mixed integer problems and has been successfully applied to stochastic programming [13], [16]. To the best knowledge of the authors this approach has not been applied to min-max problems and in particular, to evaluate feedback MPC control laws.

The method consists on solving a sequence of LP problems that approximate the value of the original problem. In (10) the value of the function $V_j^*(z_i)$ has to be evaluated. These functions are complex nonlinear functions. In the proposed algorithm, these functions are substituted by an outer linearization, i.e. a lower bound that can be evaluated using a linear problem. This lower bound is improved at each iteration and converges to the real value of $V_j^*(z_i)$.

In this paper is addressed a particular case of multi-stage min-max linear programs to simplify the algorithm. It is assumed that $V_i^*(z)$ is finite and greater than zero for all nodes. This assumption holds in the case of feedback min-max problems, where the objective function satisfies this condition by definition. It is not difficult no generalize the algorithm. In what follows the algorithm is presented. Also, optimality and finite termination of the same is proved.

At each step m of the algorithm, subproblems P_i^m are solved, one for each node i of the scenario tree, to obtain the lower bound of the cost function $V_i^*(z)$. As each of the functions of the children nodes are also approximated by an outer linearization, the resulting subproblem is a linear program. The lower bound $V_i^m(z_{p(i)})$ is obtained solving the following optimization problem (P_i^m)

$$\begin{aligned} &\min_{z_i, \theta_i, \theta_{i,j}} c_i^T z_i + \theta_i && \text{(11a)} \\ &\text{s.t. } W_i z_i = h_i - A_i z_{p(i)}, && \text{(11b)} \\ &D_{i,j}^k z_i + \theta_{i,j} \geq d_{i,j}^k, \quad \forall j \in I(i),\ k \leq r_{i,j}^m, && \text{(11c)} \\ &\theta_i \geq \theta_{i,j},\ \forall j \in I(i), && \text{(11d)} \\ &z_i, \theta_i \geq 0,\ \theta_{i,j} \geq 0,\ \forall j \in I(i). && \text{(11e)} \end{aligned}$$

These problems are modified at each iteration. The number of constraints in (11c) at step m is $r_{i,j}^m$. These constraints are optimality cuts added in order to give a lower bound on the value of $V_j^m(z_i)$(See Theorem 1, namely

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

At the first iteration $r_{ij}^1 = 0$ for all nodes and $j \in I(i)$. 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 $V_j^m(z_i)$ is tighter.

Each $\theta_{i.j}$ is a lower bound on the value of $V_j^m(z_i)$. To evaluate the maximization over the set of childrens, the auxiliary variable θ_i is included. Constraints (11d) evaluate the maximization over all the children of node i using an epigraph approach.

In this section is proved, that $V_i^m(z)$ is a lower bound on $V_i^*(z)$ and that when the algorithm stops, both values are equal.

As in the previous section, when solving the root node, constraints (11b) 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_0.

When solving a leaf node, variables $\theta_{i,j}$ and constraints (11c) 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 [13], 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 [13], [16]. Note that feedback min-max is formulated to have relatively complete recourse. By definition, if x_0 lies in χ^0, there exists a sequence of control laws that drive x_0 to the terminal region satisfying the constraints for all possible uncertainties because constraint (5) holds for each node.

The algorithm is based on adding optimality cuts at each time step. These optimality cuts are obtained from feasible solutions to the dual problem of P_i^m. The dual problem D_i^m is defined as (note that strong duality holds):

$$\begin{aligned}
&\max_{\lambda_i, \mu_{i,j}^k, \mu_j} (h_i - A_i z_{p(i)})^T \lambda_i + \sum_{j \in I(i)} \sum_{k=1}^{r_{ij}^m} d_{ij}^k \mu_{ij}^k \\
&\text{s.t. } c_i - W_i^T \lambda_i - \sum_{j \in I(i)} \sum_{k=1}^{r_{ij}^m} D_{ij}^{kT} \mu_{ij}^k \geq 0, \\
&1 - \sum_{j \in I(i)} \mu_j \geq 0, \\
&\mu_j - \sum_{k=1}^{r_{ij}^m} \mu_{ij}^k \geq 0, \ \forall j \in I(i), \\
&\mu_j, \mu_{i,j}^k \geq 0.
\end{aligned} \tag{12}$$

where $\lambda_i, \mu_{i,j}^k, \mu_j$ are the dual variables corresponding to constraints (11b), (11c) and (11d) respectively.

Theorem 1: 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 (11b) 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,

$$V_i^m(z) \geq d_i - D_i z. \tag{13}$$

Proof: From duality, given the set of optimal dual variables $\lambda_i, \mu_{i,j}^k, \mu_j$ for $z_{p(i)}$ with optimal value $V_i^m(z_{p(i)})$, it is easy to see that the following inequality holds for any given z:

$$V_i^m(z) \geq (h_i - A_i z)^T \lambda_i + \sum_{j \in I(i)} \sum_{k=1}^{r_{ij}} d_{ij}^k \mu_{ij}^k. \tag{14}$$

Note also that

$$V_i^m(z_{p(i)}) = (h_i - A_i z_{p(i)})^T \lambda_i + \sum_{j \in I(i)} \sum_{k=1}^{r_{ij}} d_{ij}^k \mu_{ij}^k. \tag{15}$$

Subtracting (14) and (15):

$$V_i^m(z) \geq \lambda_i^T A_i z_{p(i)} + V_i^m(z_{p(i)}) - \lambda_i^T A_i z.$$

In this way, recalling the definition of d_i and D_i, it is proved that (13) holds. ■

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 differ on each iteration because optimality cuts are added, the lower bounds on the optimal value remain valid.

The proposed algorithm is the following:

Algorithm 1: Proposed algorithm.

INITIALIZATION $m = 0, r_{ij}^0 = 0, \forall i, j$.

IF P_0 is unfeasible

  Multi-stage min-max problem is unfeasible.

  End of the algorithm.

END IF

DO

  FOR $i = 0, \ldots, M-1$

    Solve P_i^m using $z_{p(i)}$ and obtain $V_i^m(z_{p(i)})$, z_i, $\theta_{i,j}$ and λ_i.

  END FOR

  FOR $i = M-1, \ldots, 0$

    FOR $j \in I(i)$

      IF $\theta_{i,j} < V_j^m(z_i)$.

        $r_{i,j}^{m+1} = r_{i,j}^m + 1$.

        $D_{i,j}^{r_{i,j}^{m+1}} = \lambda_j^T A_j$.

        $d_{i,j}^{r_{i,j}^{m+1}} = \lambda_j^T A_j z_i + V_j^m(z_i)$.

      ELSE

        $r_{i,j}^{m+1} = r_{i,j}^m$.

      END IF

    END FOR

    $e_i = \max_{j \in I(i)} V_j^m(z_i) - \theta_{i,j} + e_j$.

  END FOR

  $m = m + 1$.

WHILE $e_0 > 0$

End of the algorithm.

Theorem 2: The solution obtained applying Algorithm 1, denoted z_0^*, is an optimal solution of (10). The algorithm converges in finite time.

Proof: The optimality cuts are obtained from a set of optimal dual variables of P_j^m as sub-gradients of the optimal solution $V_j^m(z)$. These sub-gradients are defined using the

dual variables of the equality constraints (11b) λ_j and the optimal value of the cost function $V_j^m(z)$ which includes the remaining dual variables for different values of z. Taking into account Theorem 1 and the definition of $D_{i,j}^k$ and $d_{i,j}^k$ given in Table 1, the following inequality holds for all iteration m and node j

$$V_j^m(z) \geq \max_{k=1,\ldots,r_{p(j),j}^m} d_{p(j),j}^k - D_{p(j),j}^k z. \qquad (16)$$

By construction, if $I(i)$ is empty it holds $V_i^m(z) = V_i^*(z)$ because in this case (10) and (11) are equal. Taking this into account and applying (16) backwards from the leaf nodes, it holds for all iteration m and node j

$$V_j^*(z) \geq V_j^m(z). \qquad (17)$$

The algorithm finishes when $e_0 = 0$. By construction, if $e_0 = 0$, then $e_i = 0$ for $i = 0, \ldots, M-1$, i.e. for the values z_i of the algorithm solution it holds:

$$V_j^m(z_{p(j)}) = \max_{k=1,\ldots,r_{p(j),j}^m} d_{p(j),j}^k - D_{p(j),j}^k z_{p(j)}, \qquad (18)$$

for $j = 0, \ldots, M-1$.

By definition, all of these functions of z are convex so the minimization problems have an unique minimum value. Then if z_i is a minimizer of P_i^m (and so minimizes θ_i), $e_i = 0$ and $V_j^*(z_i) = V_j^m(z_i)$, then the following holds:

$$V_i^*(z_{p(i)}) = V_i^m(z_{p(i)}). \qquad (19)$$

Again taking into account that for the leaf nodes the hypothesis $V_j^*(z_i) = V_j^m(z_i)$ is true, applying backwards (19) it is proved:

$$V_0^* = V_0^m. \qquad (20)$$

Finite termination of the algorithm is assured because the optimality cuts are generated from the dual variables of the optimal solutions of P_i^m. As P_i^m are LP problems, the optimal dual variables are attained at the vertices of the feasibility set of the dual problem. As the number of vertices is finite and no cut can be repeated, the maximum number of iterations of the algorithm is finite. ■

V. Examples

The computation time and the memory requirements of the proposed algorithm is compared for a simple problem with the corresponding solution to solving the min-max problem with a single large scale LP problem. The comparisons have been realized in Matlab in a AMD Athlon Xp 2800+ using GLPK [17] as the linear solver for the decomposition algorithm and the large scale LP.

Consider the problem of robustly regulating to the origin the system:

$$x(t+1) = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} x(t) + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u(t) + Dw(t). \qquad (21)$$

subject to constraints in the state and the input, namely,

$$-10 \leq x(t) \leq 10,$$
$$-3 \leq u(t) \leq 3.$$

TABLE I

Computational aspects for different prediction horizons for **S1**.

	node	iter	Ben(s)	mem(Ben)	LP(s)	mem(LP)
$N=1$	3	1	0.01	3Kb	0	2.23Kb
$N=2$	7	2.4	0.01	5Kb	0.01	9.22Kb
$N=3$	15	3.5	0.04	9Kb	0.01	30.5Kb
$N=4$	31	4.2	0.09	18Kb	0.05	89.1Kb
$N=5$	63	5.4	0.23	36Kb	0.29	244Kb
$N=6$	127	6.1	0.53	72Kb	2.04	635Kb
$N=7$	255	6.8	1.19	144Kb	17.77	1.5Mb
$N=8$	511	7.3	2.54	288Kb	143.7	3.1Mb

TABLE II

Computational aspects for different prediction horizons for **S2**.

	node	iter	Ben(s)	mem(Ben)	LP(s)	mem(LP)
$N=1$	5	1	0.04	4Kb	0.02	3.84Kb
$N=2$	21	2.2	0.03	11Kb	0.02	31.6Kb
$N=3$	85	2.6	0.13	41Kb	0.37	203.Kb
$N=4$	341	3.1	0.69	163Kb	25.2	1.41Mb
$N=5$	1365	4.3	3.34	648Kb	203.2	8.4Mb

No terminal region is taken into account.

The disturbance in supposed to be bounded in the hypercube $\|w(t)\|_\infty \leq 1.5$. We consider two different cases: a one-dimensional disturbance and a two-dimensional one, i.e.

- **S1:** $D = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$,
- **S2:** $D = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$.

We consider the performance measure based on infinity norm with

$$P = Q = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix},\ R = 1.8$$

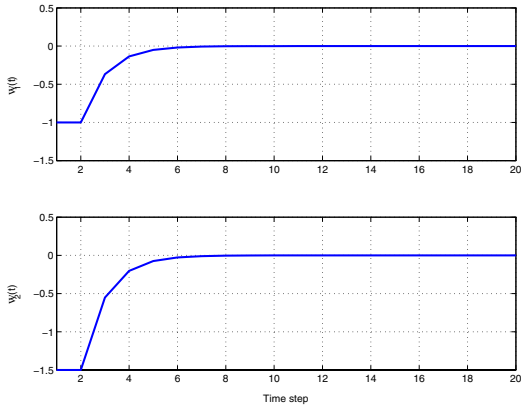
and different predictions horizons.

Table I shows different results for prediction horizons from $N = 1$ to 15 for system **S1**. The results are obtained from over a hundred different initial states. Entry *node* denotes the number of nodes of the scenario tree. *iter* is the mean number of iterations of the decomposition algorithm and *Benders* is the time in seconds. Entry *mem(Ben)* is the size of the file that the Benders decomposition algorithm needs to solve the optimization problem. The SMPS format [18] has been used to store the problem. Entry *mem(LP)* is the size of the MPS [19] file of the large scale LP and *LP* is the time needed to solve the problem. Table II shows the same simulations for **S2** and different predictions horizons.

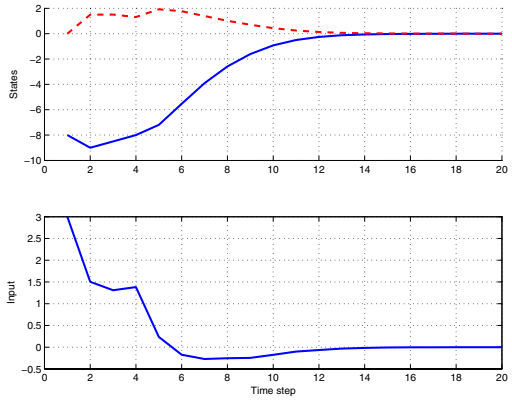
For each node and iteration of the Benders decomposition algorithm, a small LP is solved ($nx+nu+q+2$ variables and a number of constraints of the same order of magnitude). The number of nodes of the scenario tree grows in an exponential manner. The number of constraints and variables of the large scale LP is proportional to the number of nodes.

In the simulation results it is seen that the proposed algorithm gives promising results. It outperforms the LP solver used and memory requirements are lower. Moreover, the optimization problem has been solved with prediction horizons $N = 15$ and $N = 8$ for **S1** and **S2** respectively. The

Fig. 1. Simulation results for **S2**.



(a) Uncertainty trajectory.



(b) Simulation for the uncertainty trajectory of Figure 2(a).

mean computations time for over a hundred different initial states were 625.55 and 350.11 seconds. Note that despite the high computation time, this problems are hardly solved by a single large scale output as the number of variables and constraint is near half a million.

Figure 2(b) shows the optimal state and input trajectory for the uncertainty trajectory shown in Figure 2(a) when a min-max controller with a prediction horizon of $N = 4$ is applied to system **S2** from initial state $x(0) = \left[\begin{smallmatrix} 0 \\ -8 \end{smallmatrix}\right]$.

VI. Conclusions

The paper has shown how to compute the solution of a multi-stage min-max linear program in an efficient way. The proposed algorithm can be applied to implement any feedback min-max control problem with a stage cost that can be evaluated via a LP problem. Although the complexity of the algorithm still grows exponentially with the prediction horizon, it broadens the family of processes to which this control technique can be applied.

The decomposition algorithm also gives an insight into the underlying structure of the problem formulation and highlights that worst case minimization is closely related with stochastic programming. Future works include efficient implementations of the algorithm and extension to feedback min-max based on quadratic performance indexes.

VII. Acknowledgements

We acknowledge the PhD advisor of D. Muñoz de la Peña, E.F. Camacho, who as the general chair of this conference did not think should also authored any of the papers, for his participation.

References

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

[2] P. Campo and M. Morari, "Robust model predictive control," in *Proc. American Contr. Conf.*, vol. 2, 1987, pp. 1021–1026.

[3] 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.

[4] 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.

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

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

[7] A. Casavola, M. Giannelli, and E. Mosca, "Minmax predictive control strategies for input-saturated polytopic uncertain systems," *Automatica*, vol. 36, no. 1, pp. 125–133, 2000.

[8] J. Schuurmans and J. Rossiter, "Robust predictive control using tight sets of predicted states," in *Control Theory and Applications, IEE Proceedings*, vol. 1, 2000, pp. 1021–1026.

[9] J. Löfberg, "Minimax approaches to robust model predictive control," Ph.D. dissertation, Department of Electrical Engineering, Linköping University, Sweeden, April 2003, downloadable from http://www.control.isy.liu.se/publications/doc?id=1466.

[10] T. Alamo, D. Muñoz de la Peña, and E. Camacho, "Constrained min-max predictive control: Modifications of the objective function leading to polynomial complexity," *IEEE Trans. Automatic Control*, vol. In press, 2005.

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

[12] 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.

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

[14] A. Propoi, "Use of linear programming methods for synthesizing sampled-data automatic systems," *Automation and Remote Control*, vol. 24, no. 7, pp. 837–844, 1963.

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

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

[17] A. Makhorin, *GNU Linear Programming Kit*. http://www.gnu.org/software/glpk, 2005.

[18] H. Gassmann, "The smps format for stochastic linear programs," in *http://www.mgmt.dal.ca/sba/profs/hgassmann/SMPS2.htm*.

[19] J. Edwards, J. Birge, A. King, and L. Nazareth, "A standard input format for computer codes which solve stochastic programs with recourse and a library of utilities to simplify its use," in *Working Paper WP-85-03*, International Institute for Applied Systems Analysis, Laxenburg, Austria, 1985.