AN EFFICIENT ALGORITHM FOR PREDICTIVE CONTROL OF PIECEWISE AFFINE SYSTEMS WITH MIXED INPUTS

Sylvain Leirens * **Jean Buisson** *

* *Supélec–IETR, Cesson-Sévigné, France*

Abstract: This paper presents a mixed optimization algorithm devoted to predictive control of hybrid systems belonging to the PieceWise Affine (PWA) class with mixed (*i.e.* continuous and discrete) inputs. By using the particular structure of the optimization problem, the number and the dimension of the subproblems to solve in order to find the optimum are significantly reduced.
This approach is applied to a classical case study in the field of hybrid systems: the control of water levels of a three tank system. *Copyright © 2006 IFAC*

Keywords: predictive control, hybrid systems, PWA systems, mixed optimization

1. INTRODUCTION

A great part of the literature concerning *advanced control* involves the concept of optimality and needs suitable optimization methods. To address the control problems of complex and constrained industrial processes, predictive control has become a standard, although it requires an optimization procedure which has to be carried out on line. At each sampling time, an optimal control sequence (with respect to a given cost function) on a finished prediction horizon is found, whose first control only is applied to the system. The whole procedure is then repeated at the next time step (*receding* horizon).

Hybrid systems are dynamic systems involving the presence of continuous and discrete variables in the models. Consequently, the optimization methods used by the predictive control scheme are confronted to combinatorial aspects. Among the various paradigms allowing to build hybrid models, some classes of systems lend themselves particularly to the development of predictive control laws, especially in their discrete time formulation. A commonly used class of hybrid systems is piecewise affine systems (PWA) which are defined by a piecewise affine dynamic over non-overlapping regions of the state-input space.

The effort of research is mainly devoted to techniques leading to a reduction of the combinatorial complexity involved in the optimization problem which is associated with the predictive control law. The purpose of these techniques consists in avoiding the enumeration of all the possibilities for the discrete variables on the prediction horizon, which generates a (very) great number of subproblems to be solved to find the searched optimum at each sampling time.

For continuous PWA systems (*i.e.* no discrete state nor discrete control inputs), reach set computation based algorithms are proposed, coupled to a branch & bound strategy in (Bemporad *et al.*, 2000), and using a state transition graph in (Peña *et al.*, 2003). In (Stursberg, 2004), nonlinear programming and hybrid system simulation are embedded into a graph search algorithm. For switched affine systems (SAS) which define a subclass of PWA systems involving mixed control inputs, but no partitioning of the continuous state-input space, a partial enumeration algorithm is proposed in (Leirens *et al.*, 2005).

The Mixed Logical Dynamical (MLD) formalism, whose equivalence with PWA systems has been proved in (Heemels *et al.*, 2001), leads to a model made up of discrete time linear equations and a set of linear inequalities. They involve real as well as binary variables (Bemporad and Morari, 1999). The implementation of predictive control for hybrid systems modeled in a MLD form requires the solving of mixed linear or quadratic optimization problems, called MILP/MIQP, which need a significant amount of computation. These kinds of optimization problems are solved by using branch & bound techniques with relaxation of the binary variables and linear or quadratic programming, called LP/QP.

In this paper, the approach suggested in (Leirens *et al.*, 2005) is extended to PWA systems with mixed control inputs. It is organized as follows. PWA systems are presented in section 2. Hybrid predictive control principles are introduced in section 3 and section 4 is devoted to the solving of the associated mixed optimization problem. Section 5 presents the implementation of the proposed approach and the results obtained for the control of water levels of a classical hybrid control case study: the three tank system. The conclusions are given in section 6.

2. PWA SYSTEMS

The class of systems, called PWA systems with mixed inputs (but whose state is continuous), that are used in this paper is defined by the following discrete time equations:

$$\mathbf{x}(k+1) = \mathbf{A}_i\mathbf{x}(k) + \mathbf{B}_i\mathbf{u}_c(k) + \mathbf{a}_i \quad (1)$$

$$\mathbf{y}(k) = \mathbf{C}_i\mathbf{x}(k) + \mathbf{D}_i\mathbf{u}_c(k) + \mathbf{c}_i \quad (2)$$

where $\mathbf{x} \in \mathbb{X} \subset \mathbb{R}^{n_x}$ is the state vector, $\mathbf{u}_c \in \mathbb{U} \subset \mathbb{R}^{n_{uc}}$ is the continuous control input vector and $\mathbf{y} \in \mathbb{Y} \subset \mathbb{R}^{n_y}$ system output vector. A partition χ_j ($j \in \mathbb{J} \subset \mathbb{N}$) of the continuous state-input space, independent of the discrete input vector $\mathbf{u}_d$ ($\mathbf{u}_d \in \mathbb{U}_d$ with e.g., $\mathbb{U}_d \subset \mathbb{N}^{n_{ud}}$), is defined as follows :

$$\chi_j = \left\{(\mathbf{x}, \mathbf{u}_c) \,|\, \mathbf{F}_j\mathbf{x} + \mathbf{G}_j\mathbf{u}_c \leq \mathbf{f}_j\right\} \quad (3)$$

The subscript i in (1)–(2) denotes the mode $i(k) \in \mathbb{I} \subset \mathbb{N}$ with respect to the k^{th} time step and is defined by a value of the discrete inputs $\mathbf{u}_d(k)$ and the membership of a partition χ_j. A function $\varphi \,:\, \mathbb{X}\times\mathbb{U}_c\times\mathbb{U}_d \longrightarrow \mathbb{I}$ is then defined such that :

$$i(k) = \varphi(\mathbf{x}(k), \mathbf{u}_c(k), \mathbf{u}_d(k)) \quad (4)$$

The control inputs $\mathbf{u}$ of the system are composed of the continuous ones ($\mathbf{u}_c$) and discrete ones ($\mathbf{u}_d$) with $n_u = n_{uc} + n_{ud}$. The whole number of modes for the system is given by :

$$p = \text{card}(\mathbb{I}) = \text{card}(\mathbb{J}) \times \text{card}(\mathbb{U}_d) \quad (5)$$

3. PREDICTIVE CONTROL

The principle of predictive control consists in solving an open loop optimal control problem on a receding prediction horizon. The loop is closed by the use of a new measurement of the system state at every sampling time. The prediction model (PWA) is time invariant thus the time origin can be used as the current time step in the equations without loss of generality. Let $\mathbf{U}_N$ be the sequence of controls on the prediction horizon, chosen of length N :

$$\mathbf{U}_N = (\mathbf{u}^T(0)\ \mathbf{u}^T(1)\ \cdots\ \mathbf{u}^T(N-1))^T \quad (6)$$

with $\mathbf{u} = (\mathbf{u}_c^T\ \mathbf{u}_d^T)^T$ and given the following cost function :

$$J_N(\mathbf{x}(0), \mathbf{U}_N) = F(\mathbf{x}(N)) + \sum_{k=0}^{N-1} L(\mathbf{x}(k), \mathbf{u}(k)) \quad (7)$$

In practice, the cost function (or performance index to be optimized) includes a term based on the state $\mathbf{x}$ compared to a reference and a term based on the control inputs $\mathbf{u}$. Given :

$$L(\mathbf{x}(k), \mathbf{u}(k)) = \|\mathbf{x}(k+1) - \mathbf{x}_r\|_{\mathbf{Q}_x} + \|\mathbf{u}(k)\|_{\mathbf{Q}_u} \quad (8)$$

$$F(\mathbf{x}(N)) = \|\mathbf{x}(N) - \mathbf{x}_r\|_{\mathbf{Q}_f} \quad (9)$$

where $\mathbf{x}_r$ is the state reference. Using 2-norm, the cost function is quadratic, *i.e.* $\|\mathbf{w}\|_{\mathbf{Q}} \triangleq \mathbf{w}^T\mathbf{Q}\mathbf{w}$. The weighing matrices are such that $\mathbf{Q}_x \geq 0$, $\mathbf{Q}_u > 0$ (by taking account of constraints on $\mathbf{u}$, typically actuator constraints, it is sufficient for the matrix $\mathbf{Q}_u$ to be semi-positive definite) and $\mathbf{Q}_f \geq 0$.

At every time step, the following optimization problem $\mathcal{P}_N$ has to be solved, where superscript o means optimality :

$$\mathcal{P}_N(\mathbf{x}(0)) \ :\ J_N^o(\mathbf{x}(0)) = \min_{\mathbf{U}_N} J_N(\mathbf{x}(0), \mathbf{U}_N) \quad (10)$$

fulfilling the model constraints (1)–(3). A final state constraint $\mathbf{x}(N) = \mathbf{x}_f$ can be added, especially to guarantee the stability of the closed loop system (Mayne *et al.*, 2000).

A characteristic of the optimization problem which is formulated above is its mixed nature. The presence of continuous inputs ($\mathbf{u}_c$) and discrete ones ($\mathbf{u}_d$) leads to consider the searched control sequence $\mathbf{U}_N$ as a sequence of continuous inputs denoted $\mathbf{U}_{cN}$ and a sequence of discrete inputs denoted $\mathbf{U}_{dN}$ on the prediction horizon N. Then :

$$J_N^o(\mathbf{x}(0)) = \min_{\mathbf{U}_{dN}} \left(\min_{\mathbf{U}_{cN}} J_N(\mathbf{x}(0), (\mathbf{U}_{cN}, \mathbf{U}_{dN})) \right) \quad (11)$$

subject to the model constraints, for $k = 0, 1, \cdots, N-1$:

$$\mathbf{x}(k+1) = \mathbf{A}_i\mathbf{x}(k) + \mathbf{B}_i\mathbf{u}_c(k) + \mathbf{a}_i \quad (12)$$

$$\mathbf{y}(k) = \mathbf{C}_i\mathbf{x}(k) + \mathbf{D}_i\mathbf{u}_c(k) + \mathbf{c}_i \quad (13)$$

$$\mathbf{F}_i\mathbf{x}(k) + \mathbf{G}_i\mathbf{u}_c(k) \leq \mathbf{f}_i \quad (14)$$

with $i(k) = \varphi(\mathbf{x}(k), \mathbf{u}_c(k), \mathbf{u}_d(k))$.

Let $\mathbf{I}_N = \big(i(0)\ i(1)\ \cdots\ i(N-1)\big)^T \in \mathbb{I}^N$ be a sequence of modes on the horizon N. It defines a sequence $\mathbf{U}_{dN}$ and N sets of constraints (12)–(14) for $k = 0, 1, \cdots N-1$. The problem $\mathcal{P}_N$ can be rewritten this way :

$$J_N^o(\mathbf{x}(0)) = \min_{\mathbf{I}_N}\left(\min_{\mathbf{U}_{cN}} J_N(\mathbf{x}(0), (\mathbf{U}_{cN}, \mathbf{I}_N))\right) \quad (15)$$

For a given sequence of modes $\mathbf{I}_N$, the cost :

$$J_N^*(\mathbf{x}(0), \mathbf{I}_N) = \min_{\mathbf{U}_{cN}} J_N(\mathbf{x}(0), (\mathbf{U}_{cN}, \mathbf{I}_N)) \quad (16)$$

is the optimal cost that is found by solving a continuous optimization subproblem (the sequence $\mathbf{I}_N$ – then $\mathbf{U}_{dN}$ – is known) subject to constraints. This subproblem can easily be reformulated in a standard problem of quadratic programming (QP). The superscript $*$ means optimality regarding to a sequence of modes: the continuous control sequence called $\mathbf{U}_{cN}^*$ is said optimal with respect to $\mathbf{I}_N$. However, the constrained optimization problem (16) is not necessarily feasible, *i.e.* for the given sequence $\mathbf{I}_N$, no solution can satisfy the model constraints (12)–(14) on the horizon.

4. MIXED OPTIMIZATION

The optimization problem associated with the predictive control of PWA systems is now formulated, this section is then devoted to its solving.

4.1 Exhaustive enumeration

The simplest way to find the searched optimum consists in enumerating all the possible sequences of modes on the prediction horizon and then to solve the QP subproblems associated with the corresponding sequences. For a given sequence of modes, two situations may occur: either there is no solution satisfying the constraints (infeasibility) or the QP subproblem is feasible. For each feasible sequence $\mathbf{I}_N$ (defining a discrete control sequence $\mathbf{U}_{dN}$) an optimal continuous control sequence $\mathbf{U}_{cN}^*$ and a corresponding cost $J_N^*(\mathbf{x}(0), \mathbf{I}_N)$ can be obtained. The searched optimum is then given by the sequence of modes which minimizes (16):

$$J_N^o(\mathbf{x}(k)) = \min_{\mathbf{I}_N}\left(J_N^*(\mathbf{x}(k), \mathbf{I}_N)\right) \quad (17)$$

This method which is called exhaustive enumeration is quickly useless when the number of modes and/or the length of the prediction horizon increase because the problem is NP-hard.

Using a tree shape representation, the depth of the tree of possibilities grows with the length of the horizon. For a given depth, the width of the tree is fixed by the number of possible modes for the system. Each leaf is a QP subproblem to solve and one of them is the searched optimum. The dimension of all the QP subproblems is identical, *i.e.* the dimension of the optimization vector is $\dim(\mathbf{U}_c) = n_{uc} \times N$.

4.2 Partial enumeration

Exhaustive enumeration consists of completely covering the tree of possibilities. The optimization problem associated with the predictive control has a particular structure: the cost is additive with positive terms. The key idea of the suggested partial enumeration algorithm is knowing a suboptimum of (11) – (14), evaluate partial costs in order to prune the tree, *i.e.* cut branches that cannot lead to the optimum. It is a kind of branch & bound algorithm.

For a partial horizon P ($P < N$), *i.e.* at a depth P in the tree, a partial cost is defined as follows:

$$J_P(\mathbf{x}(0), \mathbf{U}_P) = \sum_{k=0}^{P-1} L(\mathbf{x}(k), \mathbf{u}(k)) \quad (18)$$

The proposed approach is a recursive algorithm which is composed of a descent strategy to explore the tree of possibilities and a criterion of branch cutting.

Descent strategy: suppose to be P ($< N$) time steps in the future, *i.e.* at a depth P in the tree of possibilities. The proposed strategy a kind of *best first* one :

- compute the optimal costs J_{P+1} associated with the feasible subproblems for the possible choices of mode i;
- begin with the branch which gives the minimal cost on the horizon $P+1$ to continue the exploration.

Branch cutting: suppose to have a first suboptimum (all the leaves for which the associated QP subproblem is feasible are suboptima, the searched optimum is the best one). Prune the tree by cutting the branches for which:

- the optimal cost on a partial horizon is greater than the cost of the known suboptimum;
- the subproblem is infeasible.

To cut a branch means to eliminate all the branches following it. The *known* suboptimum is updated when a leaf is evaluated and whose cost is lower than the one obtained for the previous suboptimum.

Given a sequence of N modes $\mathbf{I}_N$ (the associated QP subproblem is assumed to be feasible) and a horizon P with $P < N$. The following notations are used :

- $\mathbf{I}_P^{(N)}$ is the sequence of the P first modes extracted from the sequence $\mathbf{I}_N$;
- $\mathbf{U}_{cP}^{(N)}$ is the continuous control sequence of length P extracted from the sequence $\mathbf{U}_{cN}$.

It is recalled that the superscript $*$ means optimality regarding to a given sequence of modes: *i.e.* the sequence $\mathbf{U}_{cN}^*$ is optimal regarding to a given sequence of modes $\mathbf{I}_N$. However the extracted sequence $\mathbf{U}_{cP}^{*(N)}$ is not necessarily optimal over the horizon P.

Proposition: Given a sequence of modes $\mathbf{I}_N$, for all $P < N$, the optimal cost that is obtained for the sequence $\mathbf{I}_N$ is greater than the optimal cost that is obtained for an extracted sequence $\mathbf{I}_P^{(N)}$:

$$\forall\, P < N, \quad J_N^*\big(\mathbf{x}(0), \mathbf{I}_N\big) \geq J_P^*\big(\mathbf{x}(0), \mathbf{I}_P^{(N)}\big) \quad (19)$$

Proof: The cost (16) can be split in two terms, for all $P < N$:

$$J_N^*\big(\mathbf{x}(0), \mathbf{I}_N\big) = J_P\big(\mathbf{x}(0), (\mathbf{U}_{cP}^{*(N)}, \mathbf{I}_P^{(N)})\big) + \sum_{k=P}^{N-1} L\big(\mathbf{x}(k), (\mathbf{u}_c^*(k), i(k))\big) + F(\mathbf{x}(N)) \quad (20)$$

The cost is additive with respectively positive and semi-positive definite functions L and F:

$$J_N^*\big(\mathbf{x}(0), \mathbf{I}_N\big) \geq J_P\big(\mathbf{x}(0), (\mathbf{U}_{cP}^{*(N)}, \mathbf{I}_P^{(N)})\big) \quad (21)$$

The cost $J_P\big(\mathbf{x}(0), (\mathbf{U}_{cP}^{*(N)}, \mathbf{I}_P^{(N)})\big)$ is the cost obtained with an extracted sequence $\mathbf{U}_{cP}^{*(N)}$ and is then suboptimal over the horizon P:

$$\underbrace{J_P\big(\mathbf{x}(0), (\mathbf{U}_{cP}^{*(N)}, \mathbf{I}_P^{(N)})\big)}_{\text{suboptimal}} \geq \underbrace{J_P^*\big(\mathbf{x}(0), \mathbf{I}_P^{(N)}\big)}_{\text{optimal}} \quad (22)$$

□

Suppose being at a depth $P < N$ and that the associated cost J_P is greater than a known suboptimum, with respect to the preceding proposition, the corresponding branch and all the following ones can be cut.

The partial enumeration algorithm is a branch & bound algorithm which leads to the optimal solution by taking advantage of the particular structure of the optimization problem associated with predictive control. Mixed integer programming, which was developed to solve standard problems, purely combinatorial or mixed, cannot exploit this feature. In the framework of predictive control of MLD systems, the associated optimization problem is reformulated in such a standard MIQP problem (Bemporad and Morari, 1999; Mignone, 2002).

The suggested descent strategy is a *heuristic* which allows to obtain a first suboptimum that is hoped of good quality, *i.e.* not far from the optimum. The suboptimal character comes from the choice of the sequence of modes (best first strategy at one prediction step) but the suboptimum is obtained by solving a QP subproblem on the full horizon N.

In this approach by partial enumeration, the dimension of QP subproblems to be solved starts with n_{uc} at the top of the tree (one step time prediction) to grow with the horizon until $n_{uc} \times N$ at the bottom of the tree (horizon N).

This algorithm is illustrated on figure 1. The numbers indicate the way followed out in the tree, *i.e.* how the tree has been explored. Bold lines gives the path to the first suboptimum (best first strategy). The presence of a cross means the result of a branch cutting: either the cost is greater than the one of the known suboptimum or the subproblem is infeasible. The searched optimum is marked out by a triangle.

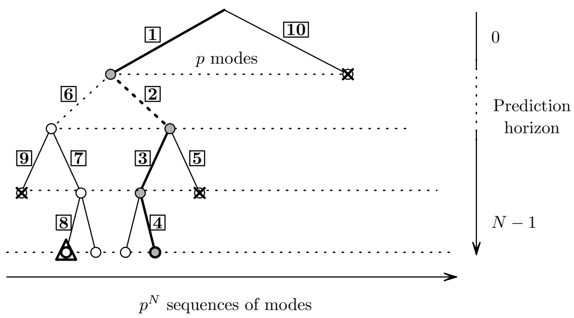

Fig. 1. Partial enumeration

Remarks:

(1) The presence of an equality constraint on the final state $\mathbf{x}(N) = \mathbf{x}_r$ has to be considered only for the evaluation of a cost on the complete horizon (N). This constraint does not exist for the evaluation of a partial cost ($P < N$).
(2) In the case of a purely combinatorial optimization problem, the cost at a node in the tree of possibilities is obtained by the sum of the costs associated with the branches leading to this node (the cost is evaluated downward). The problem considered here has a mixed nature. The cost of a branch is not known *a priori* since it depends on the continuous control $\mathbf{u}_c^*$ found by solving the QP subproblem associated with the path in the tree.

5. APPLICATION EXAMPLE

5.1 Description of the case study

The proposed algorithm is applied to the three tank system (Lunze, 1998) whose diagram is represented figure 2. The system is composed of three

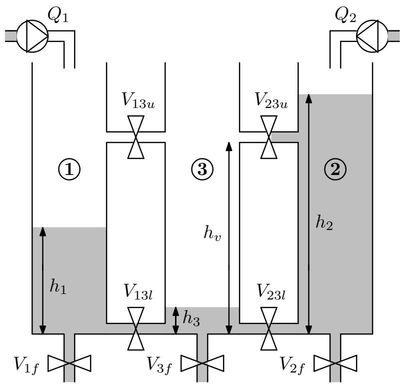

Fig. 2. Three tank system

tanks numbered from 1 to 3 with a maximal height h_{max}. Tanks 1 and 2 are supplied with water by two pumps whose respective flows Q_1 and Q_2 can vary between 0 and Q_{max}. Four valves V_{13l}, V_{23l}, V_{13u} and V_{23u} allow to control flows between the tanks. It is assumed that they can take only two states : opened (1) or closed (0). The upper valves are located at a height h_v. The valves V_{1f}, V_{2f} and V_{3f} define outgoing flows. The levels of water in the three tanks, which are the variables to be controlled, are noted h_1, h_2 and h_3 respectively.

We do not give details about the model in this paper: a complete review of the equations used to build the prediction model can easily be found, e.g. in (Mignone, 2002). We focus on the results obtained for a step change in the water height references. The state space generated by h_1, h_2 and h_3 is divided in eight partitions. Considering both positions, open or closed, of the four valves V_{13l}, V_{23l}, V_{13u} and V_{23u}, the total number of modes for this case study is $8 \times 2^4 = 128$.

Using a sample time $T = 10\,\text{s}$, simulations of $20 \times T = 200\,\text{s}$ have been carried out for various prediction horizons ($2 \leq N \leq 6$). The first column of table 1 gives the numbers of QP subproblems which would have been necessary to solve at each sample step while proceeding by exhaustive enumerations. The three following columns show the minimum (QPmin), average (QPmoy) and maximum (QPmax) numbers of solved subproblems at each sample step obtained using partial enumerations (the number of solved QP may vary at each sample step during the simulation). The minimum number of QP can not be less than the one given by a straight descent in the tree, just enumerating the p possible modes at each time step while choosing the path with the smallest partial cost: QPmin $\geq p \times N$.

N	p^N	QP min	QP moy	QP max
2	16384	256	420	1024
3	$2.1\,10^6$	384	1390	6528
4	$2.7\,10^8$	512	4815	44928
5	$3.4\,10^{10}$	1664	9045	86912
6	$4.4\,10^{12}$	2688	15104	139264

Table 1. Performances of the partial enumeration algorithm

	part. enum. (PWA)		miqp (MLD)	
N	QP max	time (s)	QP max	time (s)
2	1024	4.34	589	35.1
3	6528	27.8	6821	591
4	44928	233.3	27609	3583

Table 2. Comparison with miqp–MLD

Comparisons have been made between the results obtained by using partial enumeration and miqp algorithms. Although miqp is not recognized as an efficient MIQP solver (such as commercial ones, e.g. CPLEX), it is used as a basis to make the comparison. Both algorithms are Matlab scripts using quadprog from the optimization toolbox. The miqp algorithm is based on branch & bound and relaxation strategies and is associated with the MLD formalism, quoted in introduction. The MLD model of the three tank system has been obtained using the software Hysdel 2.05 (Torrisi *et al.*, 2002). The results are presented in table 2 for $2 \leq N \leq 4$ with the maximum numbers of solved subproblems at each time step and the associated computation times (not taken into account in the control loop). These results have been obtained with Matlab 7.0 running on a 2 GHz clocked PC computer.

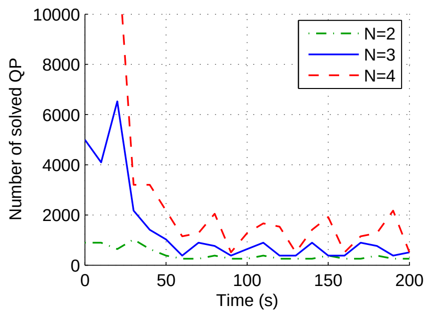

Fig. 3. Number of solved QP versus simulation time ($N = 2, 3, 4$) with partial enumeration

In addition to the number of subproblems solved to obtain the optimum, the computation time is also related to the dimension of the subproblems. The MLD formalism requires the addition of $n_\delta = 3$ auxiliary binary variables and $n_z = 7$ auxiliary real variables. The maximum dimension of a subproblem is then $(n_{uc} + n_{ud} + n_\delta + n_z) \times N$. For the partial enumeration, the maximal dimen-

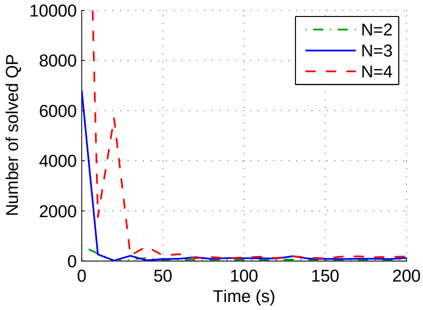

Fig. 4. Number of solved QP versus simulation time ($N = 2, 3, 4$) with miqp

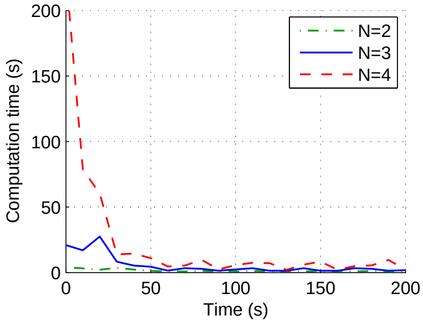

Fig. 5. Computation time versus simulation time ($N = 2, 3, 4$) with partial enumeration

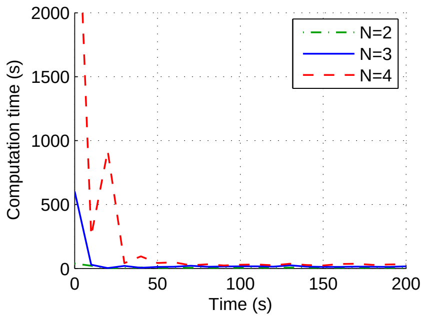

Fig. 6. Computation time versus simulation time ($N = 2, 3, 4$) with miqp – vertical time scale factor = 10 compared to figure 5

sion of a subproblem is $n_{uc} \times N$. The interest of the suggested approach also lies in the small dimension of the subproblems to solve, which induces small computation times. At the beginning of the optimization procedure, it is necessary to solve a significant number of subproblems but of very small dimensions. To cut branches in the tree allows to restrict the number of subproblems with a more significant size to be solved. In this example, it should be noticed that the farther from its reference the state is (actually at the beginning of the simulation), the bigger the computational effort required to solve QP subproblems is.

6. CONCLUSIONS

This article presents a kind of branch & bound algorithm to solve the mixed optimization problem associated with the predictive control of PWA systems with mixed inputs. Knowing a suboptimum, the key idea of this algorithm is to evaluate partial costs (*i.e* at intermediate or partial horizons) in order to prune the tree of the possibilities by cutting the branches which cannot lead to the searched optimum.

REFERENCES

Bemporad, A. and M. Morari (1999). Control of systems integrating logic, dynamics, and constraints. *Automatica* **35**(3), 407–427.

Bemporad, A., L. Giovanardi and F.D. Torrisi (2000). Performance driven reachability analysis for optimal scheduling and control of hybrid systems. In: *Proceedings of the 39th Conference on Decision and Control*. Sydney, Australia.

Heemels, W. P. M. H., B. De Schutter and A. Bemporad (2001). Equivalence of hybrid dynamical models. *Automatica* **37**(7), 1085–1091.

Leirens, S., J. Buisson, P. Bastard and J.-L. Coullon (2005). An efficient algorithm for solving model predictive control of switched affine systems. In: *Proceedings of the Scientific Computation, Applied Mathematics and Simulation World Congress*. Paris, France.

Lunze, J. (1998). Laboratory three tanks system benchmark for the reconfiguration problem. Technical report. Technical University of Hamburg-Harburg, Institute of Control Engineering. Hamburg, Germany.

Mayne, D. Q., J. B. Rawlings, C. V. Rao and P. O. M. Scocaert (2000). Constrained model predictive control: Stability and optimality. *Automatica* **36**(6), 789–814.

Mignone, D. (2002). Control and Estimation of Hybrid Systems with Mathematical Optimization. PhD thesis. Swiss Federal Institute of Technology (ETH). Zürich.

Peña, M., E. F. Camacho and S. Piñón (2003). Hybrid systems for solving model predictive control of piecewise affine system. In: *Proceedings of the IFAC Conference on Analysis and Design of Hybrid Systems*. Saint-Malo, France.

Stursberg, O. (2004). A graph search algorithm for optimal control of hybrid systems. In: *Proceedings of the 43rd Conference on Decision and Control*. Nassau, Bahamas.

Torrisi, F., A. Bemporad and D. Mignone (2002). Hysdel - a tool for generating hybrid models. Technical Report AUT00-03. Automatic Control Laboratory, ETH. Zürich, Switzerland.