

MODEL PREDICTIVE CONTROLLER FOR PIECEWISE AFFINE SYSTEM

Miguel Peña* Eduardo F. Camacho**
Sandra Piñón** Ricardo Carelli*

* *Instituto de Automática. F.I. Universidad Nacional de
San Juan. Argentina. mpena@inaut.unsj.edu.ar*

** *Dpto. Ing. de Sistemas y Automática. ESI, Universidad
de Sevilla. España.*

Abstract: This paper presents a hybrid procedure to solve Model Predictive Controller (MPC) for Piecewise Affine System (PWA) The approach presented here belong to the class of Branch and Bound (B&B) methods. The procedure uses the concepts of reachable set combined to the specific B&B methods, in order to reduce the number of Quadratic Problems (QP) needed to be solved by the optimization algorithm. *Copyright 2005 IFAC*

Keywords: Predictive control, piecewise linear, search trees

1. INTRODUCTION

MPC has become an accepted standard for complex constrained control problems in the process industry. The main limitation to which processes MPC could be used on is the computationally expensive on-line optimization requirement. Explicit solutions to the MPC problem from linear constrained systems have been derived (Bemporad *et al.* (2002)). Unfortunately these approaches are so complex and it is not applicable for great dimension systems.

PWA systems arise as an approximation of smooth nonlinear systems (Julián (1999)) and they are equivalent (under additional assumptions) to some classes of Hybrid Systems (HS) (Heemels *et al.* (2001)).

The application of MPC to HS requires to solve an optimization program with mixed, integer and real, decision variables (Bemporad and Morari (1999)). In spite of this combinatorial nature, several algorithmic approaches have been proposed to reduce the computed time. Bemporad and Mignone (2000) presents a B&B tree exploring

strategy for solving Mixed Integer Quadratic Program (MIQP) involving time evolutions of Mixed Logical Dynamical (MLD) model. The problem of this method is that the system has to be converted to a MLD system. This conversion, in some cases, increases the model complexity. In addition, simulation is much easier for PWA systems than for MLD systems (Bemporad (2002)).

This paper presents a hybrid procedure to solve MPC of PWA in order to reduce the computing time needed to solve the problem. The proposed algorithm use a PWA model instead of the MLD model. The approach presented here belong to the class of B&B methods. The procedure uses the concepts of reachable set (Kerrigan (2000)) combined to the specific B&B methods, in order to reduce the number of QP problems need to solves the optimization algorithm.

The paper is organized as follows: in section 2 the PWA systems are described and the used MPC strategy is developed. A simulation example is shown in section 3.1 and concluding remarks are given in section 4.

2. PROBLEM FORMULATION

A PWA systems is defined as

$$\mathbf{x}_{k+1} = A^i \mathbf{x}_k + B^i \mathbf{u}_k + \mathbf{f}^i \text{ for } : \mathbf{x}_k \in \mathcal{X}_i \quad (1)$$

where $\{\mathcal{X}_i\}_{i=1}^s$ is a polyhedral partition of the states space. Each \mathcal{X}_i is given by

$$\mathcal{X}_i \triangleq \{\mathbf{x}_k | Q^i \mathbf{x}_k \preceq \mathbf{q}^i\} \quad (2)$$

where \mathbf{x}_k and \mathbf{u}_k denote the state and the input vectors, respectively. The symbol \preceq indicate that all component of vector fulfil the inequality.

Each subsystem \mathbb{S}^i defined by the 5-uple $(A^i, B^i, \mathbf{f}^i, Q^i, \mathbf{q}^i)$, $i \in \{1, 2, \dots, s\}$ is termed a component of the PWA system (1). $A^i \in \mathbb{R}^{n \times n}$, $B^i \in \mathbb{R}^{n \times m}$, and (A^i, B^i) is a controllable pair. $Q^i \in \mathbb{R}^{p_i \times (n+m)}$ and $\mathbf{f}^i, \mathbf{q}^i$ are suitable constant vectors. Note that n is number of states, m is the number of inputs and p_i is the number of hyperplanes that define the i -polyhedral.

Assume that a full measurement of the state is available at the current time k . Most formulation of MPC require that the problem

$$U = \arg \left(\min_U J \right) \quad (3)$$

$$\text{to : } J = \sum_{i=1}^N (\mathbf{x}_{k+i|k} - \mathbf{w}_k)^T Q_i (\mathbf{x}_{k+i|k} - \mathbf{w}_k)^2$$

$$+ \sum_{i=1}^N \mathbf{u}_{k+i-1}^T R_i \mathbf{u}_{k+i-1} \quad (4)$$

$$\text{s.t. : } \mathbf{u}_{k+i} \in \mathcal{U} \quad \mathcal{U} \triangleq \{\mathbf{u}_{k+i} | Q_u \mathbf{x}_k \preceq \mathbf{q}_u\} \quad (5)$$

is solved at each time k , where $\mathbf{x}_{k+i|k}$ denoted the predicted state vector at the $k+i$ time, obtained by applying the input sequence $U \triangleq \{\mathbf{u}_k, \dots, \mathbf{u}_{k+N-1}\}$ to model (1) starting from the state \mathbf{x}_k subjected to constraints. It be noted that $\mathbf{w}_k = \mathbf{w}$ is the constant output reference. $Q_i = Q$ and $R_i = R$ are the (time predicted) constant state and control weight matrix. The first control input is then applied to the process. At the next sample, measurements are used to update the optimization problem, and the optimization is repeated. In this way, this becomes a closed-loop approach.

Let us consider the prediction problems associated to the MPC in the case of PWA system. The subsystem describing the process is known if \mathbf{x}_k is known, but the following subsystems depends on the applied control sequence. It can be considered that a change (transition) of model is produced between a sampling instant and the next. In general a sequence of subsystems $I = \{I_k, I_{k+1}, \dots, I_{k+N-1}\}$ is activated, where $I_j = [1, \dots, s]$ to $j = [k, k+1, \dots, k+N-1]$. Only the initial value $I_k = I_k(\mathbf{x}_k)$ of this sequence is known. In order to solve the MPC problem (4)

the optimization sequence is added to the decision variables. The resulting optimization problem can be stated as

$$U = \arg(\min_{U, I} J) \quad (6)$$

where constraints relating the dependences of the possible sequences U and I have to be added, i.e.

$$Q^{I^{k+j}} \mathbf{x}_{k+j} \preceq \mathbf{q}^{I^{k+j}}, j = \{1, \dots, N-1\} \quad (7)$$

Due to the integer nature of sequence I , the problem of finding the optimum can be solved by finding the optimum of the solutions for all possible sequence of I , i.e.

$$U = \arg \left(\min_I \left(\min_U \left(\min_{Q^{IU} \mathbf{u} \preceq \mathbf{q}^{IU}} J \right) \right) \right) \quad (8)$$

where $Q^{IU} \mathbf{u} \preceq \mathbf{q}^{IU}$ indicate the constraints due to dependences between I and U .

If no constraints are considered, the number of possible sequences for a prediction horizon N is s^{N-1} . This implies that the number of real optimization problem to solve is prohibitive for large N and s .

Methods to reduce the number of problems with real variables should therefore be developed. This paper is devoted to this purpose. The algorithm uses the properties of MPC when applied to PWA in order to make the B&B method more efficient.

2.1 Branch and Bound Algorithms

The B&B optimization method is a discrete search technique, which has been successfully applied to different complex optimization problems. The B&B method is a structured search technique belonging to a general class of enumerative schemes. A graphical representation of the concepts and separation in B&B algorithms can be drawn with the help of trees. Figure 1 depicts a s-ary tree. The tree can be explored in several ways. The choice of the problem separation and the order in which the subproblems are considered, influences the average computational effort.

To reduce the number of alternatives, bounding is applied. A particular branch j at level i is followed only if the lower bound of the cumulative cost $J^{(i,j)}$ plus a lower bound on the cost from the level i to the terminal level L , denoted $J_L^{(i)}$ is lower than an best upper bound of the total cost (of the whole search tree), denoted J_U , that is $J^{(i)} + J_L^{(i)} < J_U$. The lower bound is difficult to derive, and can be simply set to zero $J_L^{(i)} = 0$.

Thus, the efficiency of the B&B tree search strongly depends on good lower and upper bounds, especially on a good initial upper bound, i.e., a good initial estimate of J_U .

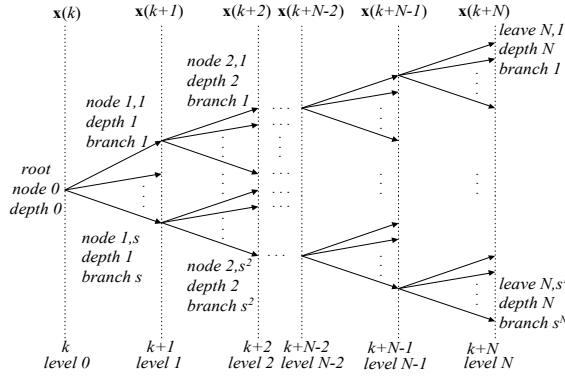


Fig. 1. Graphical representation of the concepts in B&B algorithms.

2.2 Resolution of the Computational Problem when I is fixed

If the sequence $I = \{I_k, I_{k+1}, \dots, I_{k+N-1}\}$ is given, equation (4) can be written as

$$J = (\bar{\mathbf{x}} - \bar{\mathbf{w}})^T \bar{Q} (\bar{\mathbf{x}} - \bar{\mathbf{w}}) + \bar{\mathbf{u}}^T \bar{R} \bar{\mathbf{u}} \quad (9)$$

where $\bar{R} = \text{diag}[R_i]$, ($\bar{R} = \bar{R}^T \succ 0$), $\bar{Q} = \text{diag}[Q_i]$, ($\bar{Q} = \bar{Q}^T \succ 0$) for $i = 1, \dots, N$, and $\bar{\mathbf{x}} = [\mathbf{x}_{k+1}^T \dots \mathbf{x}_{k+N}^T]^T$, $\bar{\mathbf{w}} = [\mathbf{w}_{k+1}^T \dots \mathbf{w}_{k+N}^T]^T$, $\bar{\mathbf{u}} = [\mathbf{u}_k \dots \mathbf{u}_{k+N-1}^T]^T$ the predicted state vector can be written as

$$\bar{\mathbf{x}} = F_x \mathbf{x}_k + H_x \bar{\mathbf{u}} + \mathbf{f}_{o_x} \quad (10)$$

See Peña *et al.* (2003) for detail.

Replacing (10) in (9) the index is

$$J(I, U) = \bar{\mathbf{u}}^T H_{QP} \bar{\mathbf{u}} + \mathbf{f}_{QP}^T \bar{\mathbf{u}} + g_{QP} \quad (11)$$

where $H_{QP} = [H_x^T \bar{Q} H_x + \bar{R}]$, $\mathbf{f}_{QP}^T = [2\mathbf{x}_k^T F_x^T \bar{Q} H_x + 2\mathbf{f}_o^T \bar{Q} H_x - 2\bar{\mathbf{w}}^T \bar{Q} H_x]$, $g_{QP} = \mathbf{x}_k^T F_x^T \bar{Q} F_x \mathbf{x}_k + 2\mathbf{f}_o^T \bar{Q} F_x \mathbf{x}_k - 2\mathbf{f}_o^T \bar{Q} \bar{\mathbf{w}} + \mathbf{f}_o^T \bar{Q} \mathbf{f}_o - 2\bar{\mathbf{w}}^T \bar{Q} F_x \mathbf{x}_k + \bar{\mathbf{w}}^T \bar{Q} \bar{\mathbf{w}}$. The constrain over the control (5) can be written as $Q_u \bar{\mathbf{u}} \leq \mathbf{q}_u$, and the constraints due to I and U dependence (7) can be written as

$$Q^I \bar{\mathbf{x}}_I \leq \mathbf{q}^I \quad (12)$$

where $\bar{\mathbf{x}}_I = [\mathbf{x}_{k+1}, \dots, \mathbf{x}_{k+N-1}]^T$, $Q^I = \text{diag}(Q^{I_{k+1}}, \dots, Q^{I_{k+N-1}})$, $\mathbf{q}^I = [(\mathbf{q}^{I_{k+1}})^T, \dots, (\mathbf{q}^{I_{k+N-1}})^T]^T$, $\bar{\mathbf{x}}_I$ can be written by

$$\bar{\mathbf{x}}_I = C_x \bar{\mathbf{x}} \quad (13)$$

where $C_x = [I_{n \times n} \times n \times N \ 0]$. Replacing (13) and (10) in (12), the constraints due to the dependency between U and I results in $Q^{IU} \bar{\mathbf{u}} \leq \mathbf{q}^{IU}$, $Q^{IU} = Q^I C_x H_x$, $\mathbf{q}^{IU} = \mathbf{q}^I - Q^I C_x F_x \mathbf{x}_k - Q^I C_x \mathbf{f}_{o_x}$. If constraints on the control actions are also considered then

$$Q_{QP} \bar{\mathbf{u}} \leq \mathbf{q}_{QP} \quad (14)$$

$$Q_{QP} = [(Q_u)^T (Q^{IU})^T]^T, \mathbf{q}_{QP} = [(\mathbf{q}_u)^T (\mathbf{q}^{IU})^T]^T.$$

Therefore, once the sequence I is fixed, the problem can be solved by minimizing (11) subject to the constraints (14).

Once the sequence of subsystems that are going to be reached is defined, it results in a QP problem that can be solved with standard QP optimization methods.

2.3 Bound on the objective function

This section shows how a lower bound for the objective function can be found when the sequence of regions is fixed. This bound will be used by the B&B algorithm to reduce the number of QP problems.

The performance index (9) can be write as

$$J = \sum_{i=1}^N J_i \quad (15)$$

where

$$J_i = (\mathbf{x}_{k+i|k} - \mathbf{w})^T Q (\mathbf{x}_{k+i|k} - \mathbf{w}) + \mathbf{u}_{k+i-1}^T R \mathbf{u}_{k+i-1} \quad (16)$$

for $i = 1, \dots, N$. Due to $\mathbf{x}_{k+i|k} = A^{k+i-1} \mathbf{x}_{k+i-1|k} + B^{k+i-1} \mathbf{u}_{k+i-1} + \mathbf{f}^{k+i-1}$ this equation can be write as

$$J_i = [L\check{\mathbf{x}} + \mathbf{p}]^T Q [L\check{\mathbf{x}} + \mathbf{p}] + \check{\mathbf{x}}^T R \check{\mathbf{x}}$$

where $\check{\mathbf{x}} = \begin{bmatrix} \mathbf{x}_{k+i-1|k} \\ \mathbf{u}_{k+i-1} \end{bmatrix}$, $L = [A^{k+i-1} \ B^{k+i-1}]$, $\mathbf{p} = \mathbf{f}^{k+i-1} - \mathbf{w}$, $M = \begin{bmatrix} 0 & 0 \\ 0 & R \end{bmatrix}$. It is rearranged as

$$J_i = \check{\mathbf{x}}^T H \check{\mathbf{x}} + \mathbf{f}^T \check{\mathbf{x}} + q$$

where $H = L^T Q L + M$, $\mathbf{f}^T = 2\mathbf{p}^T Q L$, $q = \mathbf{p}^T Q \mathbf{p}$. If the sequence of subsystems is defined that is $I = \{I_k, I_{k+1}, \dots, I_{k+N-1}\}$, $I_j \in \{1, \dots, s\}$ then, the state and input fulfil that

$$\mathbf{x}_j \in \mathcal{X}_{I_j} \text{ and } \mathbf{u}_j \in \mathcal{U} \text{ for } j = \{k, \dots, k+N-1\}$$

as well as their predicted values. Therefore, it is possible to obtain a minimum bound of J_i . It is

$$\begin{aligned} \min(J_i) &\geq \min_{\check{\mathbf{x}}} (\check{\mathbf{x}}^T H \check{\mathbf{x}} + \mathbf{f}^T \check{\mathbf{x}} + q) \\ \text{s.t.} &: Q^{I_j} \mathbf{x} \leq \mathbf{q}^{I_j}, \quad Q_u \mathbf{u} \leq \mathbf{q}_u \end{aligned}$$

This is a QP Problem and can be resolve by standard software. Due to $J_{I_k} \geq 0$ then

$$\min(J) \geq \sum_{i=k}^{k+N-1} \min(J_{I_i}) \quad (17)$$

The equation 17 is and lower bound of the index 9.

Using the lower bound of the objective function 17 the B&B algorithm described in Section 2.1 can be used to solve the problem. However, this bound is too conservative and it is not too useful for the algorithm.

2.4 B&B in a MPC of PWA Model

Given a depth i and branch j there is an associated unique subsequence characterized by $I^{(i,j)}$ ($0 < i < (Np - 1)$, $1 < j < s^i$). Solving the associated QP problem, a lower bound is obtained for the objective function $J^{(i,j)}$. This allows the decision tree to be pruned because the minimal cost function obtained by the QP problem associated to the sequence $\{I_k, \dots, I_{k+p}\}$ is an accumulative lower bound of the cost function.

Given the state in the k time \mathbf{x}_k and the integer sequence of subsystems $I^{(i,j)} = \{I_k, \dots, I_{k+p}\}$ fixed by the node (i, j) . Then the minimal value of the cost function 9 that can be obtained for this sequence is smaller than any other that can be obtained for any other sequence $I = \{I_k, \dots, I_{k+p}, I_{k+p+1}\}$ where $I_{k+p+1} = [1, \dots, s]$

The sequence of control moves $U = \{\mathbf{u}_k, \dots, \mathbf{u}_{k+p}\}$ is obtained minimizing J by solving the convex problem (11), (14). The cost function is the addition of positive elements associated to each of the subsystems (16) encountered along the trajectory. When a new system is added, a positive component is added to the function and the cost increases.

That is, the optimal value obtained for the sequence $\{I_k, \dots, I_{k+p}\}$ is a accumulative lower bound of the cost function $J^{(i,j)}$ and this allows to prune the decision tree.

Given the state in the k time \mathbf{x}_k and the integer sequence of subsystems $I = \{I_k, \dots, I_{k+p}\}$ if the QP problem (11), (14) is infeasible then the QP problem related with the sequence $I = \{I_k, \dots, I_{k+p}, I_{k+p+1}\}$ is infeasible too.

The QP problem associated to sequence $I = \{I_k, \dots, I_{k+p}\}$ is unfeasible when there are no admissible control sequence $U = \{\mathbf{u}_k, \dots, \mathbf{u}_{k+p}\}$ can take the system from state $\mathbf{x}_k \in \mathcal{X}_{I_k}$ to $\mathbf{x}_{k+p} \in \mathcal{X}_{I_{k+p}}$ and passing through states $\mathbf{x}_{k+1} \in \mathcal{X}_{I_{k+1}}$, \dots , $\mathbf{x}_{k+p-1} \in \mathcal{X}_{I_{k+p-1}}$. Therefore, if the system cannot be taken to $\mathbf{x}_{k+p} \in \mathcal{X}_{I_{k+p}}$ then it would be impossible to go from this last state to $\mathbf{x}_{k+p+1} \in \mathcal{X}_{I_{k+p+1}}$ and the sequence $I = \{I_k, \dots, I_{k+p}, I_{k+p+1}\}$ would therefore be unfeasible.

This allows the decision tree to be pruned further. Because if the QP problem associated to sequence $\{I_k, \dots, I_{k+p}\}$ is unfeasible, all descending branches will also be unfeasible.

The proposed algorithm is based on a depth search along the prediction horizon. The reason is that as soon as one obtains a feasible sequence the associated cost can be used as a upper bound of the global index J_U and branches can be

pruned using the bound established by algorithm described in Section 2.3 and using Proposition .

The best upper bound of the total cost J_U is fixed equal to infinite to start the algorithm.

For a given depth and branch (this implies a determined sequence of subsystems), the algorithm can be described as follows.

1- Compute the minimum bound of the corresponding node using (17)

Each of the subsystems have an associated minimum bound that can be computed in an off line manner. A minimum bound for a sequence can be computed as the addition of the these minimum bounds for all subsystems of the sequence.

2- If the bound found in Step 1 is bigger than the best upper bound J_U , then the node and all descending branches are eliminated from the tree. Next node is evaluated.

3- Otherwise the QP problem associated to the sequence is solved.

4- If the QP problem is unfeasible, then the node and all descending branches are eliminated from the tree. Next node with the same depth is evaluated.

5- If the QP problem is feasible, the cost 9 is a lower bound for the cost of all descending branches, If the bound is bigger than the best upper bound J_U , then the node and all descending branches are eliminated from the tree. Next node is evaluated.

6- If the bound obtained in Step 5 is smaller than the best upper bound J_U then depth is increased by adding a new subsystem to the sequence.

These steps are carried out until the final depth (the prediction horizon N). If the resulting index is smaller than the best upper bound J_U , the new bound becomes the best upper bound J_U and the evaluation continues to the higher non evaluated level.

2.5 One step Reachable subsystem

To further reduce the computational problem the concept of one step reachable subsystem can be used.

The key idea of the proposed algorithm is to determine the set of possible regions that can be reach from the actual region in the next sampling time. The *Reach Set* (RS, $\mathcal{R}(\Omega)$) concept is used for this purpose. See Kerrigan (2000) for more details.

One-Step Reachable Subsystems (1-SRS) can be defined based in the reach sets idea. The \mathcal{S}^j

subsystem is 1-SRS of the \mathbb{S}^i subsystem, denoted by $\mathbb{S}^i \xrightarrow{1} \mathbb{S}^j$, if the set \mathcal{X}_j can be reached from the set \mathcal{X}_i in one step. It is possible to determine if $\mathbb{S}^i \xrightarrow{1} \mathbb{S}^j$ due to $\mathbb{S}^i \xrightarrow{1} \mathbb{S}^j \iff \mathcal{R}(\mathcal{X}_i) \cap \mathcal{X}_j \neq \emptyset$. In the same way it can be determine the n -SRS.

The index of all subsystems that are n -SRS of the subsystem i form the index set of n -SRS.

Definition: Index set of n -SRS \mathbb{N}_n^i

$$\mathbb{N}_n^i \triangleq \left\{ k \in [1, \dots, s] / \mathbb{S}^i \xrightarrow{n} \mathbb{S}^k \right\}$$

Transition from state I_j to I_{j+1} remains bounded by the 1-SRSs of the corresponding subsystem to I_j . This allow to prune the transition: a transition from I_j to I_{j+1} are not considered if $I_{j+1} \notin \mathbb{N}_1^{I_j}$. If this concept is extended, the search tree can be pruned further considering that I_{j+k} should belong to $\mathbb{N}_k^{I_j}$. Therefore, if only the 1-SRS for a prediction horizon N are considered, the number of possible sequences to test by the QP algorithm is $s^{(I_k)} s^{(I_{k+1})} \dots s^{(I_{k+N-1})}$ (instead of s^N , $s^{(I_k)} \leq s$), where s^{I_k} is the number of 1-SRSs corresponding to subsystem I_k .

It should be noted that the determination of the 1 to n -SRSs for each subsystem can be done off-line. The neighbor list for each of the subsystems is included in the model description. Each subsystem is, therefore, defined by the $(7 + N)$ -uple $(A^i, B^i, C^i, f^i, g^i, Q^i, q^i, \mathbb{N}_1^i, \dots, \mathbb{N}_N^i)$, $i \in \{1, 2, \dots, s\}$ where, \mathbb{N}^i is list containing the reachable neighbors of subsystem i .

In this article only 1-SRSs are used and they are computed off line using the algorithm proposed by Kerrigan (2000).

3. ILLUSTRATIVE EXAMPLES

3.1 Continuous PWA system

The proposed tree exploring strategy has been applied to the MPC of a simple pendulum system. A linearized equation of the discrete dynamic of the simple pendulum system is used as model. Consider the following linear system

$$ml\ddot{\theta} + kl\dot{\theta} + mg \sin(\theta) = \frac{\tau}{l}$$

where θ is pendulum angle, l is length of pendulum, m is mass of pendulum, g is the gravitational force, k is a friction coefficient and τ is a torque applied. Then, the state space model discretized to a sample time T_0 is

$$x_{k+1} = x_k + T_0 x_{k2}$$

$$x_{k1} = \left[1 - \frac{T_0 k}{m} \right] x_{k2} - \frac{T_0 g}{l} \sin(x_{k1}) + \frac{T_0}{l^2 m} \mathbf{u}_k$$

where $\mathbf{u}_k = \tau(k)$, $x_{k1} = \theta(k)$ and $x_{k2} = \dot{\theta}(k)$. Using $m = 1$, $l = 1$, $k = 0.5$, $g = 9.8$, $T_0 =$

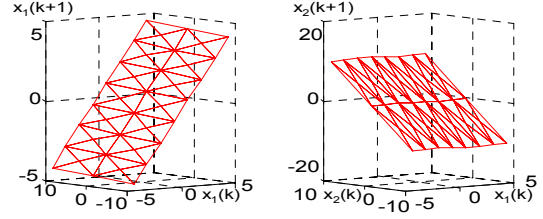


Fig. 2. PWA model representation of the pendulum

0.02 as modelling parameters. Starting from a discretized model, a continuous PWA system as (1) has been obtained using a sectors linearization over the state space uniform grid. A partition $\mathcal{X}_i \triangleq \{ \mathbf{x}_k | Q^i \mathbf{x}_k \leq q^i \}$ is defined after obtaining 56 sectors over the state space. Matrixes B^i are considered invariants, i.e.: $B^i = B = [0 \frac{\tau_0}{l^2 m}]^T$. Figure 2 shows the state linearization functions. Once the linearized model has been obtained, the neighbors corresponding to each sector are found using a reach set algorithm. In this example, the bounds of the torque are $10.78 \leq \tau \leq 10.78$ ($10.78 = 1.1 \text{ mgl}$), where the maximum torque in stable state to reach any position is $\tau = \text{mgl}$. All simulations test have been carried out in Matlab.

Figure 3 shows the results with $w = [0 \ 0]$ and $x_0 = [3, 3]$. A prediction horizon $N = 1$ to $N = 10$ is considered. The weights of the error and control action are $\bar{Q}_{x_1} = 1000$, $\bar{Q}_{x_2} = 10$, $R = 0.1$, respectively. Figure 4 show the number of (minimum (dot), maximum (plus) and means (star) values) QP evaluations Vs. the prediction horizon for this experiment. In circles, this figure show the number of QP evaluation when all possible sequences of regions ($N_{QP} = S^{N-1}$) are enumerated. For $s = 54$ and a $N = 10$, the number of QP evaluations with the enumerative method is $N_{QP} = 3904305912313344$. The maximum number of QP evaluations for the proposed algorithm is 4899, requiring a computation time (with a PC computer) of 93.45 seconds.

Note that with the proposed algorithm, the number of decision variables of the QP problems increases with the depth in the tree to a maximum (equal to the prediction horizon). Figure 5 shows the number of QP problems solved versus the decision variable size for a prediction horizon of 10. It can be observed that the maximum number of problems of length 10 is 2460 over a total of 4899.

4. CONCLUSIONS

In this paper we have proposed a new approach for solving MPC control of PWA processes. The algorithm is based on a B&B algorithm specific for a PWA systems. The exploitation of decision

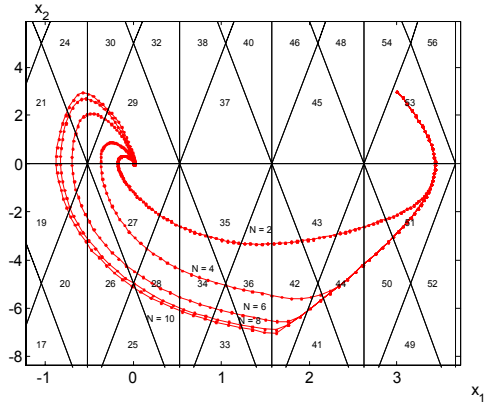


Fig. 3. Phase postrait (x_1 vs x_2) for diferent pridiction horizon ($N = 2, 4, 6, 8$), $x_0 = [3, 3]$, $w = [0, 0]$.

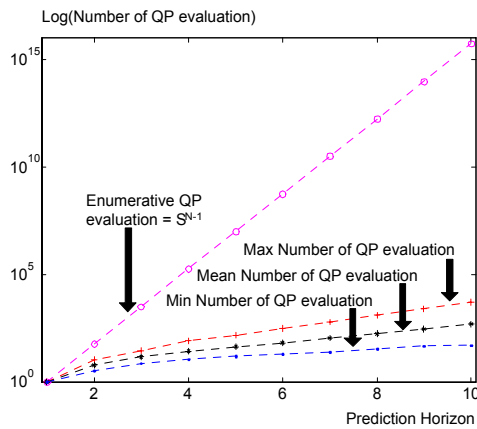


Fig. 4. Logarithm of the number of QP evaluation versus the prediction horizon.

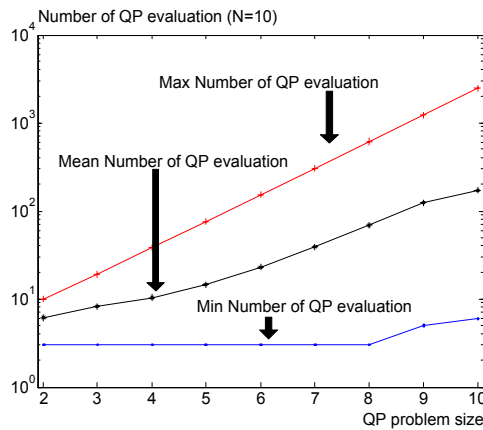


Fig. 5. Number of QP evaluation vs. the size of QP optimization problem for $N=10$.

tree and the concept of reach sets . The proposed tree exploring strategy chooses the reach set to diminish the number of possible realizations of the integer variables thus reducing considerably the number of QP problems needed to be solved.

The efficiency of the bounding mechanism depends on the quality of the bound estimates. In

this approach the bound are estimated in a coarse form, but the use of proposition 2.4 increase the prune ability. The availability of these propositions depends on the particular problem. If the PWA systems is a obtain through a linearization of a smooth non linear process (Julián (1999)), the transition between subsystems are continuous and prune ability is high. One should not forget that the computational complexity of the algorithm remains exponential, which makes it prohibitively expensive for large control horizons.

If it is used this B&B optimization technique to solve a MPC over a linearization of a smooth non linear process have the advantage over other nonlinear optimization methods that the global discrete minimum is always found, guaranteeing the optimality of control.

REFERENCES

- Bemporad, A. and M. Morari (1999), Control of systems integrating logic, dynamics, and constraints. *Automatica*, **35**, pp. 407-427
- Bemporad A. and D. Mignone (2000) MIQP.M: A Matlab function for solving mixed integer quadratic programs. Technical Reports AUT00-22. ETH Zurich.
- Bemporad, A. (2002) An Efficient Technique for Translating MLD Systems into PAS. Proc. of 41st IEEE CDC Las Vegas, Nevada USA, December.
- Bemporad, A., M. Morari , V. Dua , E. N. Pistikopoulos (2002) The explicit linear quadratic regulator for constrained systems *Automatica* Vol. **38** ,pp 3-20.
- Heemels W.P.M.H., B. De Schutter and A. Bemporad; (2001);Equivalence of hybrid dynamical models, *Automatica* Vol. **37** pp. 1085-1091.
- Julián P. (1999) Phd Tesis: A High Level Canonical Piecewise Linear Representation. Departamento de Ingeniería Eléctrica, Universidad Nacional del Sur, Bahia Blanca, Argentina.
- Kerrigan E., (2000), Phd Thesis: Robust Constraint Satisfaction: Invariant sets and predictive control, University of Cambridge.
- Peña, M; E. F. Camacho and S. Piñón (2003) Hybrid Systems for Solving MPC Of Piecewise Affine System. ADHS03, IFAC Conference on Analysis and Design of Hybrid Systems; June 2003; Saint-Malo, Brittany, France, EU.