Proceeding of the 2004 American Control Conference
Boston, Massachusetts June 30 - July 2, 2004

Predictive Control of Hybrid Systems under a Multi-MLD Formalism with State Space Polyhedral Partition

Jean Thomas, Didier Dumur, and Jean Buisson

Abstract— **The Mixed Logical Dynamical (MLD) formalism has proved to be an efficient modelling framework for hybrid systems described by dynamics, logic and constraints. Furthermore, it allows formulating and solving problems such as control, using for example predictive strategies. However, its main drawback remains the computation load due to the complexity of the Mixed Integer Quadratic Programming (MIQP) to be solved. To overcome this problem, this paper presents an innovative technique splitting the global state space to polyhedral subregions, leading to a multi-MLD model. Inside each subregion, a restricted smaller size MLD model can be developed taking into account only variables variations that may occur in this subregion. This approach enables to considerably reduce the computation time, for a more convenient real time implementation with small sampling time. This strategy is applied in simulation to the control of a three tanks benchmark**

I. INTRODUCTION

Many control problems involve hybrid systems, including both continuous and discrete variables, discrete variables coming from parts described by logic such as for example on/off switches or valves. Various approaches have been proposed to model hybrid systems [4], like Automata model, Petri nets model, and Linear Complementary (LC) model. It was shown that moving logical relations into linear constraints on integer variables provides a global modeling framework called Mixed Logical Dynamical (MLD) formalism [3]. It allows describing a large number of classes of hybrid systems. This formalism can also formulate and solve practical problems such as state estimation or control, and predictive strategies in that sense provide efficient tools, which enable MLD systems to track a desired reference trajectory.

The main drawback of this MLD formalism remains the computational burden related to the complexity of the derived Mixed Integer Quadratic Programming (MIQPs) problems. Indeed MIQP's problems are classified as NP-complete, so that in the worst case, the optimization time grows exponentially with the problem size, even if branch and bound methods may reduce this time [10]. In order to reduce the computational complexity, alternative approaches have been developed see [12].

In [13], a technique elaborating a multi-MLD model was developed, which divided the global state space domain into separate polyhedral regions inside which only feasible variables variations are considered. This leads, for each region, to the design of smaller size MLD models and MIQP's problems of restricted complexity. In this way, this paper proposes an off line systematic methodology for the elaboration of these polyhedral regions.

The paper is organized as follows. Section 2 presents a brief description of the MLD systems. General consideration about model predictive control (MPC) and its application to MLD systems are developed in section 3. Section 4 examines the multi-MLD model and the technique used to define the polyhedral partition of the global state space. Section 5 presents the application of this strategy to the water level control of a three tanks benchmark. Final conclusions are presented in section 6.

II. MLD MODEL

The MLD model permits the description of various classes of hybrid systems, like linear hybrid systems, constrained linear systems, sequential logical systems, some classes of discrete event systems, and non-linear dynamic systems, where nonlinearities can be expressed through logical combination. It describes the systems by linear dynamic equations subject to linear inequalities involving both real and integer variables, under the following form (see [3], for more details):

$$\begin{aligned}\mathbf{x}(t+1) &= \mathbf{A}\mathbf{x}(t) + \mathbf{B}_1\mathbf{u}(t) + \mathbf{B}_2\boldsymbol{\delta}(t) + \mathbf{B}_3\mathbf{z}(t) \\ \mathbf{y}(t) &= \mathbf{C}\mathbf{x}(t) + \mathbf{D}_1\mathbf{u}(t) + \mathbf{D}_2\boldsymbol{\delta}(t) + \mathbf{D}_3\mathbf{z}(t) \\ \mathbf{E}_2\boldsymbol{\delta}(t) + \mathbf{E}_3\mathbf{z}(t) &\le \mathbf{E}_1\mathbf{u}(t) + \mathbf{E}_4\mathbf{x}(t) + \mathbf{E}_5\end{aligned} \quad (1)$$

where:

$$\mathbf{x} = \begin{pmatrix}\mathbf{x}_c \\ \mathbf{x}_l\end{pmatrix} \in \Re^{n_c} \times \{0,1\}^{n_l}, \ \mathbf{u} = \begin{pmatrix}\mathbf{u}_c \\ \mathbf{u}_l\end{pmatrix} \in \Re^{m_c} \times \{0,1\}^{m_l},$$

Manuscript received September 22, 2003.

J. Thomas is with Supélec, 91192 Gif sur Yvette cedex, France (corresponding author phone: +33 (0)1 69 85 13 75, fax: +33 (0)1 69 85 13 89; e-mail: jean.thomas@supelec.fr).

D. Dumur is with Supélec, 91192 Gif sur Yvette cedex, France (e-mail: didier.dumur@supelec.fr).

J. Buisson is with Supélec, 35 511 Cesson-Sévigné cedex, France (e-mail: jean.buisson@supelec.fr)

$$\mathbf{y} = \begin{pmatrix} \mathbf{y}_c \\ \mathbf{y}_l \end{pmatrix} \in \Re^{p_c} \times \{0,1\}^{p_l}\ ,\ \boldsymbol{\delta} \in \{0,1\}^{r_l}\ ,\ \mathbf{z} \in \Re^{r_c}$$

are respectively the vectors of continuous and binary states of the system, of continuous and binary (on/off) control inputs, of output signals, of auxiliary binary and continuous variables. The auxiliary variables are introduced when translating prepositional logic into linear inequalities (Fig. 1.) $\mathbf{A}, \{\mathbf{B}_j\}_{j=1\cdots3}, \mathbf{C}, \{\mathbf{D}_j\}_{j=1\cdots3}, \{\mathbf{E}_j\}_{j=1\cdots5}$ matrices in (1) are obtained through the specification language HYSDEL as explained in [14].

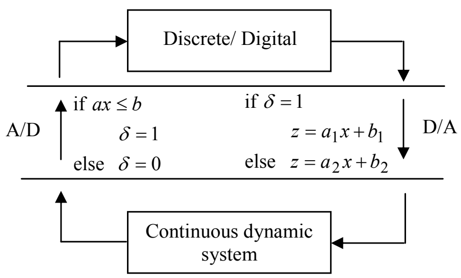


Fig. 1. MLD model structure.

III. MODEL PREDICTIVE CONTROL

Model predictive control (MPC) has proved to efficiently control a wide range of applications in industry. It is capable to control a great variety of processes, including systems with long delay times, non-minimum phase systems, unstable systems, multivariable systems, and constrained systems [6].

A. General consideration

The main idea of predictive control is to use a model of the plant to predict future outputs of the system. Based on this prediction, at each sampling period, a sequence of future control values is elaborated through an on-line optimization process, which maximizes the tracking performance while satisfying constraints. Only the first value of this optimal sequence is applied to the plant, the whole procedure is repeated again at the next sampling period according to the 'receding' horizon strategy [9].

The cost function to be minimized is generally a weighted sum of square predicted errors and square future control values, e.g. in Generalized Predictive Control (GPC) [7]. GPC for a class of hybrid systems is proposed in [5].

B. Model Predictive Control for MLD systems

For an MLD system of the form (1), the following model predictive control problem is considered. Let t be the current time, $\mathbf{x}(t)$ the current state, $(\mathbf{x}_e, \mathbf{u}_e)$ an equilibrium pair or a reference trajectory value, and N the prediction horizon, find $\mathbf{u}_t^{t+N-1} = (\mathbf{u}(t) \quad \cdots \quad \mathbf{u}(t+N-1))$ the control sequence which moves the state from $\mathbf{x}(t)$ to $\mathbf{x}_e$ and minimizes the cost function:

$$\begin{aligned} \min_{\mathbf{u}_t^{t+N-1}} J(\mathbf{u}_t^{t+N-1}, \mathbf{x}(t)) = \sum_{k=0}^{N_u-1} \left\| \mathbf{u}(k) - u_e \right\|_{\mathbf{Q}_1}^2 + \sum_{k=0}^{N-1} \left\| \boldsymbol{\delta}(k/t) - \boldsymbol{\delta}_e \right\|_{\mathbf{Q}_2}^2 + \\ + \left\| \mathbf{z}(k/t) - \mathbf{z}_e \right\|_{\mathbf{Q}_3}^2 + \left\| \mathbf{x}(k+1/t) - \mathbf{x}_e \right\|_{\mathbf{Q}_4}^2 + \left\| \mathbf{y}(k/t) - \mathbf{y}_e \right\|_{\mathbf{Q}_5}^2 \end{aligned} \quad (2)$$

subject to (1), and $\mathbf{u}(t)$ constant for $k \ge N_u$, where N_u is the control horizon, $\boldsymbol{\delta}_e$, $\mathbf{z}_e$ are the auxiliary variables of the equilibrium point or the reference trajectory value, calculated by solving a MILP problem for the inequality. $\mathbf{x}(k/t) \hat{=} \mathbf{x}(t+k, \mathbf{x}(t), \mathbf{u}_t^{t+k-1})$ (in a similar way for the other input and output variables), $\mathbf{Q}_i = \mathbf{Q}_i^{'} > 0$, for $i = 1,4$, and $\mathbf{Q}_i = \mathbf{Q}_i^{'} \ge 0$, $i = 2,3,5$.

The optimization procedure of (2) leads to MIQP problems where the optimization vector is:

$$\begin{aligned} \boldsymbol{\chi} = [\mathbf{u}(k), \cdots, \mathbf{u}(k+N-1), \boldsymbol{\delta}(k), \cdots \\ , \boldsymbol{\delta}(k+N-1), \mathbf{z}(k), \cdots, \mathbf{z}(k+N-1)]^{\mathrm{T}} \end{aligned} \quad (3)$$

and the number of binary optimization variables is $L = N\,(m_l + r_l)$. In the worst case the optimization time increases exponentially with the number of binary optimization variables [11]. From this point of view, a single MLD model describing the complete behavior of the system overall the state space and including all the variables may lead to a large size model with a huge number of binary variables, which causes a computational problem.

IV. MULTI MLD MODELS

The contribution of this paper, in order to simplify the original problem, consists in partitioning the continuous state space in domains where a subset of the boundaries defining the "continuous/discrete" interface are not crossed, and consequently where the corresponding boolean auxiliary variables are known and remain constant [13]. This partition induces a reduction of the size of the unknown $\boldsymbol{\delta}$, and can also imply in some cases a reduction in the size of $\mathbf{u}$ and $\mathbf{z}$. The subsequent simplification of the model may overcome the computation problem, which may allow if required for predictive control purposes an increased prediction horizon without facing a real time implementation problem.

A. State space partition

The continuous input $\mathbf{u}_c$ is supposed to be bounded:

$$U := \left\{ \mathbf{u}_c \in \Re^{m_c} \mid \mathbf{L}\mathbf{u}_c \le \mathbf{p}, \mathbf{L} \in \Re^{q \times m_c}, \mathbf{p} \in \Re^{q \times 1} \right\} \quad (4)$$

And the continuous state space vector $\mathbf{x}(t)$ is defined over the state space X:

$$X := \left\{ \mathbf{x} \in X \mid \mathbf{F}\,\mathbf{x} \geq \mathbf{d},\, \mathbf{F} \in \Re^{q\times n},\, \mathbf{d} \in \Re^{q\times 1} \right\} \tag{5}$$

The partition can be formally determined as follows:

Note $\Delta^{l}_{\alpha,j}$ one of the 2^{α} possible combinations ($l = 1 \cdots 2^{\alpha}$) of values of a subset of $\boldsymbol{\delta}$ containing α distinct elements, $j = 1 \cdots C^{\alpha}_{r_l}$. $\Delta^{l}_{\alpha,j}(i)$ ($i = 1 \cdots \alpha$) will denote the value of the corresponding δ element with $\delta = 1$ if and only if $\mathbf{F}_{\alpha,j,i}\mathbf{x}_c \geq d_{\alpha,j,i}$ (e.g. determined by physical consideration).

From this, $\mathbf{R}\left(\Delta^{l}_{\alpha,j}, N\right)$ is the domain of the continuous state space defined by:

$$\mathbf{R}\left(\Delta^{l}_{\alpha,j}, N\right) = \left\{ \begin{array}{l} \mathbf{x}_c(k) \in \Re^{n_c} \mid \forall \mathbf{u}(k+i), \forall i = 1 \cdots N, \\ \mathbf{x}_c(k+i) \in \mathbf{R}\left(\Delta^{l}_{\alpha,j}, 0\right) \end{array} \right\} \tag{6}$$

$$\text{with: } \mathbf{R}\left(\Delta^{l}_{\alpha,j}, 0\right) = \left\{ \begin{array}{l} \mathbf{x}_c \in \Re^{n_c} \;\Big|\; \text{if } \;\Delta^{l}_{\alpha,j}(i) = 1 \\ \text{then } \;\; \mathbf{F}_{\alpha,j,i}\,\mathbf{x}_c \geq d_{\alpha,j,i} \\ \text{else } \;\; \mathbf{F}_{\alpha,j,i}\,\mathbf{x}_c < d_{\alpha,j,i},\, i = 1 \cdots \alpha \end{array} \right\}$$

which implies that a trajectory starting in this domain will induce constant $\Delta^{l}_{\alpha,j}$ values for at least N steps. This domain can be recursively defined for $K \leq N$:

$$\mathbf{R}\left(\Delta^{l}_{\alpha,j}, K\right) = \left\{ \begin{array}{l} \mathbf{x}_c \in \mathbf{R}\left(\Delta^{l}_{\alpha,j}, K-1\right) \mid \forall \mathbf{u}(k), \\ \mathbf{x}_c(k+1) \in \mathbf{R}\left(\Delta^{l}_{\alpha,j}, K-1\right) \end{array} \right\} \tag{7}$$

Next section presents a new technique for computing those domains off line.

B. State space domains computation with polyhedral technique

For this technique, only the δ variables corresponding to the interface (continuous/discrete) that depend on the continuous states may be included in $\Delta^{l}_{\alpha,j}$.

First, assuming that the system has only continuous control inputs, and that all the $\boldsymbol{\delta}$ variables are included in $\Delta^{l}_{\alpha,j}$: For constant $\Delta^{l}_{\alpha,j}$ values and defining the $\mathbf{F}_v, \mathbf{d}_v$ matrices as in (5) according to these values, let us search the domain for any inputs values over the N future steps:

$$\mathbf{F}_v \mathbf{x}(k) \geq \mathbf{d}_v \tag{8}$$

The system dynamics in this domain are defined by:

$$\mathbf{x}(k) = \mathbf{A}\mathbf{x}(k-1) + \mathbf{B}\mathbf{u}_c(k-1) \tag{9}$$

For N steps backwards in time, the domain that respects the constraints of (8) in N steps is recursively defined as follows for $K \leq N$:

$$\mathbf{R}_0 = \left\{ \mathbf{F}_v \mathbf{x}(k) \geq \mathbf{d}_v \right\},\; \mathbf{R}_1 = \left\{ \begin{array}{l} \mathbf{F}_v\left(\mathbf{A}\mathbf{x}(k-1) + \mathbf{B}\mathbf{u}_c(k-1)\right) \geq \mathbf{d}_v \\ \mathbf{x} \in \mathbf{R}_0 \\ \forall \mathbf{u}_c : \mathbf{L}\mathbf{u}_c \leq \mathbf{p} \end{array} \right\}$$

$$\mathbf{R}_K = \left\{ \begin{array}{l} \overbrace{\mathbf{F}_v \mathbf{A}^{K}}^{\mathbf{P}^K} \mathbf{x}(k-K) + \\ + \underbrace{\mathbf{F}_v\left[\mathbf{A}^{K-1}\mathbf{B} + \mathbf{A}^{K-2}\mathbf{B} \cdots + \mathbf{B}\right]}_{\mathbf{S}^K} \mathbf{U}_K \geq \mathbf{d}_v \\ \mathbf{x} \in \mathbf{R}_{K-1} \quad , \qquad \forall \mathbf{u}_c : \mathbf{L}\mathbf{u}_c \leq \mathbf{p} \end{array} \right\} \tag{10}$$

Equation (10) can be rewritten at step N as follows:

$$\mathbf{P}^N \mathbf{x}(k-N) \geq \mathbf{d}_\mathbf{v} - \mathbf{S}^N \mathbf{U} \tag{11}$$

where $\mathbf{U} \subset U$ is the control inputs vector for the instants $(k-1),\ (k-2), \cdots, (k-N)$.

Looking for the state space domain whatever the inputs values could be, the worst case for these inputs has to be calculated as follows, using linprog Matlab code [1]:

$$\begin{array}{l} J_i = \min_{\mathbf{u}_c} \quad \underset{i}{\mathbf{S}^N}\ \mathbf{u}_c \\ \qquad \text{subject to} \quad \mathbf{L}\,\mathbf{u}_c \leq \mathbf{p} \end{array} \tag{12}$$

where $\underset{i}{\mathbf{S}^N}$ are the rows of $\mathbf{S}^N$. Let $\mathbf{J}$ be a vector of J_i for all the rows of $\mathbf{S}^N$; then it comes from (11) and (12):

$$\mathbf{P}^N \mathbf{x}(k-N) \geq \mathbf{dd}\,,\ \text{where: } \mathbf{dd} = \mathbf{d}_\mathbf{v} - \mathbf{J} \tag{13}$$

Equation (13) ensures that the dynamics of the system will not violate the constraints of (8) during the N future steps. Adding (5) to respect the physical constraints for the global state space leads to the polyhedral domain where $\Delta^{l}_{\alpha,j}$ have constant values for at least N future steps:

$$\begin{pmatrix} \mathbf{P}^N \\ \mathbf{F} \end{pmatrix} \mathbf{x} \geq \begin{pmatrix} \mathbf{dd} \\ \mathbf{d} \end{pmatrix} \tag{14}$$

Second, if there is now m_l binary (on/off) control inputs:

$$\begin{array}{c} \mathbf{x}(k) = \mathbf{A}\mathbf{x}(k-1) + \mathbf{B}\mathbf{u}_c(k-1) + \mathbf{B}_3\mathbf{z} \\ \text{where} : \mathbf{z} = \lambda \mathbf{x}(k-1) + b \end{array} \tag{15}$$

The value of $\mathbf{z}$ depends on the m_l binary control inputs, leading to 2^{m_l} possible combinations with 2^{m_l} different system dynamics.

$$\mathbf{P}^h \mathbf{x}(k-N) \geq \mathbf{d}_h \;\;,\;\; h = \left\{1, 2, \cdots, 2^{m_l}\right\} \tag{16}$$

so that the polyhedral is now computed by:

$$\mathbf{P}^T\, \mathbf{x}(k-N) \geq \mathbf{d}_T \Leftarrow \bigcap_{h=1}^{2^{m_l}} \mathbf{P}^h\, \mathbf{x}(k-N) \geq \mathbf{d}_h \tag{17}$$

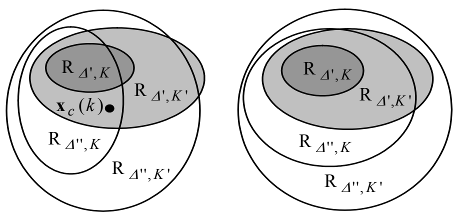
In the same way other $\mathbf{z}$ variables may be added, which depend on the free elements of $\boldsymbol{\delta}$ (elements that are not included in $\Delta^{l}_{\alpha,j}$).

The program Polylib [15] has been used to find the intersection of (10) and (17) and also for deleting the redundant constraints in (14) and (17), which results in a compact form for the polyhedral domain. This final domain ensures that the dynamics of the system will not violate the constraints of (8) during the N future steps whatever the value of the control inputs and the free $\boldsymbol{\delta}$ elements.

C. Domains characteristics

Fig. 2(a) presents a general situation where $K > K'$, for two different subsets Δ' and Δ'', then $\mathbf{R}_{\Delta',K} \subset \mathbf{R}_{\Delta',K'}$ and $\mathbf{R}_{\Delta'',K} \subset \mathbf{R}_{\Delta'',K'}$.

Fig. 2(b) presents a case where $\Delta'' \subset \Delta'$ and where the common boolean auxiliary variables have the same value, which implies $\mathbf{R}_{\Delta',J} \subset \mathbf{R}_{\Delta'',J}$ for any J. Note that notations in this figure have been simplified for clarity reasons.



(a) (b)

Fig. 2. State space partition.

D. Multi MLD development

For each previous region (6), as some binary auxiliary variables are known, a simplified MLD model can be developed. It should be noticed that this model which is only valid for K iterations, does not depend on K. Hence, the dynamics are expressed as:

$$\begin{bmatrix} \mathbf{x}(t+1) = \mathbf{A}_i\,\mathbf{x}(t) + \mathbf{B}_{i1}\,\mathbf{u}_i(t) + \mathbf{B}_{i2}\,\boldsymbol{\delta}_i(t) + \mathbf{B}_{i3}\,\mathbf{z}_i(t) \\ \mathbf{y}(t) = \mathbf{C}_i\,\mathbf{x}(t) + \mathbf{D}_{i1}\,\mathbf{u}_i(t) + \mathbf{D}_{i2}\,\boldsymbol{\delta}_i(t) + \mathbf{D}_{i3}\,\mathbf{z}_i(t) \\ \mathbf{E}_{i2}\,\boldsymbol{\delta}_i(t) + \mathbf{E}_{i3}\,\mathbf{z}_i(t) \le \mathbf{E}_{i1}\,\mathbf{u}_i(t) + \mathbf{E}_{i4}\,\mathbf{x}(t) + \mathbf{E}_{i5} \end{bmatrix} \quad (18)$$

Where i is the index of the model for the considered region, $\boldsymbol{\delta}_i$ is composed with the binary auxiliary variables which do not belong to $\Delta^{\,l}_{\alpha,j}$ and: $\mathbf{z}_i \in \Re^{r_{ic}}, r_{ic} \le r_c$, $\mathbf{u}_i \in \Re^{m_{ic}} \times \{0,1\}^{m_{il}}, m_{ic} \le m_c, m_{il} \le m_l$

Two main strategies can be used to build the MIQP that must be solved at time k.

- Look for all the $\mathbf{R}\left(\Delta^{\,l}_{\alpha,j}, I\right)$ containing $\mathbf{x}_c(k)$, where $I \ge N$, select the simplest model (the model with the smallest number of binary variables) corresponding to one of these domains and use it to build the MIQP problem, as in *III.B* with this single model.
- Look for all the $\mathbf{R}\left(\Delta^{\,l}_{\alpha,j}, K\right)$ containing $\mathbf{x}_c(k)$, select the simplest model corresponding to one of these domains for the prediction at time $k+K$, and use it to build the MIQP problem, as in *III.B* with those multiple models.

For the example of Fig. 2(a), suppose that $\mathbf{x}_c(k) \in \mathbf{R}_{\Delta'',K} \cap \mathbf{R}_{\Delta',K'}$ and that the model corresponding to Δ' is simpler than that of Δ'' and that $N = K$. The first strategy will lead to use only the model corresponding to Δ'' while the second strategy allows using the model corresponding to Δ' for prediction over $k, \cdots, k+K'$ and the model corresponding to Δ'' over $k+K'+1, \cdots, k+N$.

V. APPLICATION

A. Description of the benchmark

The proposed control strategy is applied on the three tanks benchmark used by [2]. The simplified physical description of the three tanks system proposed by COSY as a standard benchmark for control and fault detection problems is presented in Fig. 3 (see [8] for more details).

The system consists of three tanks, filled with water by two independent pumps Q_1 and Q_2 acting on tanks 1 and 2 respectively. These two pumps are continuously manipulated from 0 up to a maximum flow. Four switching valves V_1, V_2, V_{13} and V_{23} control the flow between the tanks, those valves are assumed to be either completely opened or closed ($V_i = 1$ or 0 respectively). The V_{L3} manual valve controls the nominal outflow of the middle tank. It will be assumed in further simulations that the V_{L1} and V_{L2} valves are always closed and V_{L3} is open. The liquid levels to be controlled are denoted h_1, h_2 and h_3 for each tank respectively.

The conservation of mass in the tanks provides the following differential equations:

$$\begin{aligned} \dot{h}_1 &= \frac{1}{A}(Q_1 - Q_{13V1} - Q_{13V13}) \\ \dot{h}_2 &= \frac{1}{A}(Q_2 - Q_{23V2} - Q_{23V23}) \\ \dot{h}_3 &= \frac{1}{A}(Q_{13V1} + Q_{13V13} + Q_{23V2} + Q_{23V23} - Q_N) \end{aligned} \quad (19)$$

where the $Q's$ denote the flows and A is the cross-sectional area of each of the tanks. From these expressions, a MLD model is derived as in [2], introducing the following variables:

$$\begin{aligned} \mathbf{x} &= [h_1 \;\; h_2 \;\; h_3]' \\ \mathbf{u} &= [Q_1 \;\; Q_2 \;\; V_1 \;\; V_2 \;\; V_{13} \;\; V_{23}]' \\ \boldsymbol{\delta} &= [\delta_{01} \;\; \delta_{02} \;\; \delta_{03}]' \\ \mathbf{z} &= [z_{01} \;\; z_{02} \;\; z_{03} \;\; z_1 \;\; z_2 \;\; z_{13} \;\; z_{23}]' \end{aligned} \quad (20)$$

$$\text{where:} \quad \begin{aligned} &[\delta_{0i}(t) = 1] \leftrightarrow [h_i(t) \ge h_v] & i = 1,2,3 \\ &z_{0i}(t) = \delta_{0i}(t)\left(h_i(t) - h_v\right) & i = 1,2,3 \\ &z_i(t) = V_i\left(z_{0i}(t) - z_{03}(t)\right) & i = 1,2 \\ &z_{i3}(t) = V_{i3}\left(h_i(t) - h_3\right) & i = 1,2 \end{aligned} \quad (21)$$

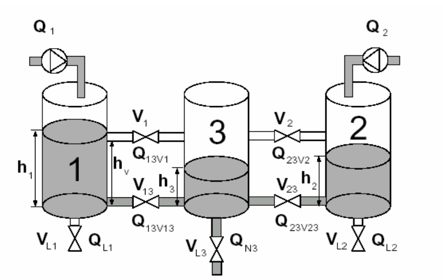


Fig. 3. COSY three tank benchmark system.

B. *Application of the multi MLD model*

Let consider now the specification: starting from zero levels (the three tanks being empty), the objective of the control strategy is to reach the liquid levels $h_1 = 0.5\,\text{m}$, $h_2 = 0.5\,\text{m}$ and $h_3 = 0.1\,\text{m}$. The state space is characterized by the domain bounded by the level constraints:

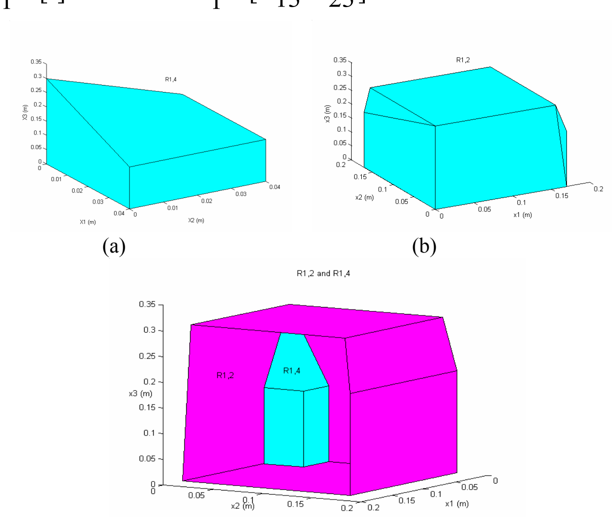
$$X = [0,0.62] \times [0,0.62] \times [0,0.3] \tag{22}$$

A comprehensive study of the dynamic behavior of the three tanks, starting from zero levels to the desired ones, enables to divide the state space into three main regions, each one with its adequate simple MLD model.

In the first region of the state space R_1, the auxiliary binary variables $\boldsymbol{\delta}$ are completely determined and have a constant value ($\boldsymbol{\delta} = [000]'$). From the control viewpoint, this region can be split into two sub-regions as well, $R_{1,4}$ and $R_{1,2}$, where $\boldsymbol{\delta}$ remains constant for $N = 4$ and $N = 2$ respectively. The two sub-regions are calculated using the technique of the previous sections. Fig. 4 presents the two polyhedral $R_{1,4}$ and $R_{1,2}$ where $R_{1,4}$ is included in $R_{1,2}$.

In this first region it clearly appears that the two valves V_1 and V_2 are not in progress, as the liquid level in this region is always less than the valves level. Consequently, the continuous auxiliary variables $\{z_{0i}\}_{i=1,2,3}$ and $\{z_i\}_{i=1,2}$ corresponding to the flows that pass through the upper pipes are useless. It results from this a simple model M_1:

$$\begin{aligned} &\mathbf{x} = [h_1 \;\; h_2 \;\; h_3]' && \mathbf{u}_1 = [Q_1 \;\; Q_2 \;\; V_{13} \;\; V_{23}]' \\ &\boldsymbol{\delta}_1 = [\;]' && \mathbf{z}_1 = [\, z_{13} \;\; z_{23}]' \end{aligned} \tag{23}$$



(a) (b) (c)

Fig. 4: (a) polyhedral $R_{1,4}$, (b) polyhedral $R_{1,2}$, (c) polyhedrals $R_{1,4}$ & $R_{1,2}$

The second region is characterized by $\boldsymbol{\delta} = [{*}{*}0]'$, since switches can occur as the levels in the first and second tanks may pass the h_v level. The second MLD model M_2 is thus:

$$\begin{aligned} &\mathbf{x} = [h_1 \;\; h_2 \;\; h_3]' && \mathbf{u}_2 = \mathbf{u} \\ &\boldsymbol{\delta}_2 = [\,\delta_{01} \;\; \delta_{02}\,]' && \mathbf{z}_2 = [\, z_1 \;\; z_2 \;\; z_{13} \;\; z_{23}]' \end{aligned} \tag{24}$$

The polyhedral domain for this region for $N = 2$ is presented in Fig. 5.

The third region is characterized by $\boldsymbol{\delta} = [110]'$, thus the model M_3 in this third region is related to the variables:

$$\begin{aligned} &\mathbf{x} = [h_1 \;\; h_2 \;\; h_3]' && \mathbf{u}_3 = \mathbf{u} \\ &\boldsymbol{\delta}_3 = [\;]' && \mathbf{z}_3 = [\, z_1 \;\; z_2 \; z_{13} \;\; z_{23}]' \end{aligned} \tag{25}$$

This region can also be split into two sub-regions: $R_{3,2}$ with $N = 2$ and $R_{3,4}$ with $N = 4$, as shown in Fig. 6. The global polyhedral partition is presented in Fig. 7.

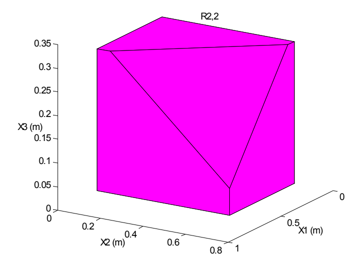
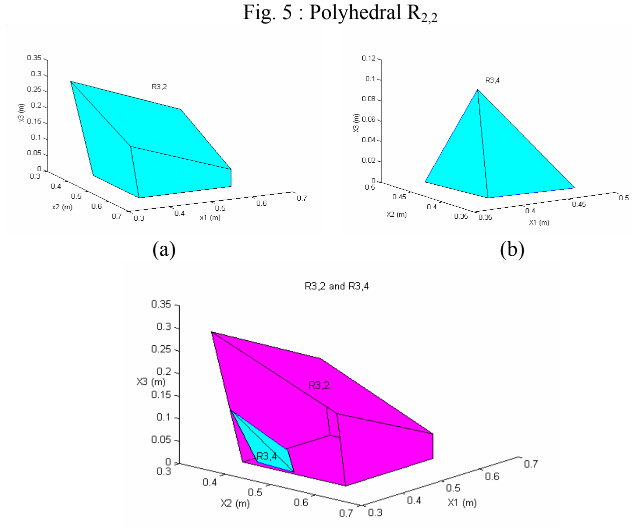


Fig. 5 : Polyhedral $R_{2,2}$



(a) (b) (c)

Fig. 6: (a) polyhedral $R_{3,2}$, (b) polyhedral $R_{3,4}$, (c) polyhedrals $R_{3,2}$ & $R_{3,4}$

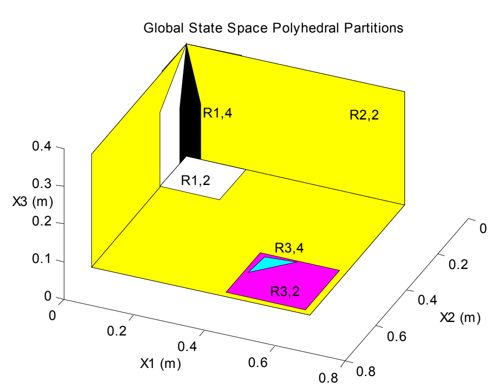


Fig. 7 Global state space polyhedral partitions.

C. *The results*

All this has been applied in simulation to reach the level specification with the control horizon $N_u = 2$ for the three regions, and a prediction horizon $N = 2$ for $R_{1,2}, R_{2,2}, R_{3,2}$, and $N = 4$ for $R_{1,4}, R_{3,4}$. The results are presented on Fig. 8 for the three tanks levels and on Fig. 9 for the control signals.

The level of the third tank oscillates around 0.1 as $h_3 = 0.1$ does not correspond to an equilibrium point. Consequently, the system opens and closes the two valves V_1 and V_2 to maintain the level in the third tank around the desired level of 0.1m.

As a comparison purpose between the multi-MLD models technique and the classical global MLD model strategy, the same previous level specifications have been considered with a global MLD model of the benchmark, and the two prediction horizons equal to 2, i.e. $N = N_u = 2$. Table 1 illustrates the total time required to reach the specification, the total number of QP's solved and the maximum time to find the optimized solution. It can be seen that the difference between the two techniques is quite large, the multi-MLD models technique allowing real time implementation and avoiding exponential explosion of the algorithm. All data given above were obtained using the MIQP Matlab code [1] for solving a mixed integer quadratic programming, on a 1.8 MHz Pc with 256 Mo of ram.

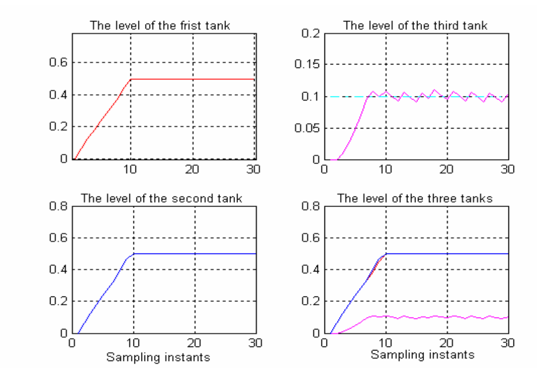


Fig. 8. Water levels in the three tanks

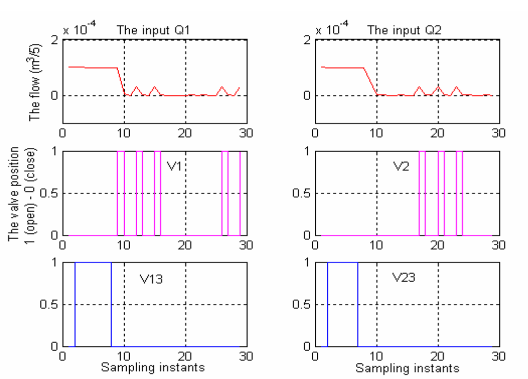


Fig. 9. Controlled variables

TABLE I
COMPUTATIONAL LOAD.

Approach	Total time	No of QP's solved	Maximum time
Classical MLD	1068 s	11 732	160.7 s
Multi-MLD	27.18 s	524	6.4 s

VI. CONCLUSIONS

This paper presents a new technique to partition the global state space into polyhedral regions where no switching could happen, so that a restricted number of variables are required, and other switching regions. Each region is then coupled to a specific simpler MLD model suitable for control. This leads to the multi-MLD models structure which successfully improves the computational problem of the MLD formalism. Moreover, in each region, particular weighting factors could be defined, according to the priority of each region. All the calculation of state space partition and the development of multi-MLD models are made off line.

REFERENCES

[1] Bemporad A. and D. Mignone. Miqp.m: A Matlab function for solving mixed integer quadratic programs. Technical Report. http://www.aut.ee.ethz.c-h/~hybrid/, 2000.

[2] Bemporad A., D. Mignone and M. Morari. Moving horizon estimation for hybrid systems and fault detection. In Proceedings of the American Control Conference, San Diego, 1999.

[3] Bemporad A. and M. Morari. Control of systems integrating logical, dynamics and constraints. Automatica, 35(3):407-427, 1999.

[4] Branicky M.S., V.S. Borkar, and S.K. Mitter. A unified framework for hybrid control: model and optimal control theory. IEEE Transactions on Automatic Control, 43(1):31-45, 1998.

[5] Buisson J. and P. Bergomi. Application of GPC to he control of switching linear systems. Journal A, 35(3):59-64, 1994.

[6] Camacho E.F. and C. Bordons. Model Predictive Control. Springer, 1999.

[7] Clarke, D.W., C. Mohtadi and P.S. Tuffs. Generalized Predictive Control, Part I "The Basic Algorithm", Part II "Extensions and Interpretation". Automatica, 23(2):137-160, 1987.

[8] Dolanc G., D. Juricic, A. Rakar, J. Petrovcic and D. Vrancic. Three-tank Benchmark Test. Technical Report Copernicus Project Report CT94-02337. ftp://ftp.utia.cas.cz/pub/staff/tesar/fault/latest/, 1997.

[9] Dumur D. and P. Boucher. A Review Introduction to Linear GPC and Applications. Journal A, 39(4), pp. 21-35, 1998.

[10] Fletcher R. and S. Leyffer. Numerical experience with lower bounds for MIQP branch and bound. Technical report, Dept. of Mathematics, University of Dundee, Scotland. http://www.mcs.dundee.ac.uk:8080/sleyffer/miqp_art.ps.Z, 1995.

[11] Raman R. and I.E. Grossmann. Relation between MILP modelling and logical inference for chemical process synthesis. Computer and Chemical Engineering 15(2), 73-84, 1991.

[12] Stursberg O. and S. Engell (2002). Optimal control of switched continuous systems using mixed-integer programming. 15th IFAC World Congress on Automatic Control.

[13] Thomas J., Buisson J., Dumur D., Guéguen H. Predictive Control of Hybrid Systems under a Multi-MLD Formalism. pp. 64-69, IFAC Conf. on Analysis and Design of Hybrid Systems, Saint-Malo, 2003.

[14] Torrisi F., Bemporad A. and D. Mignone. Hysdel – a tool for generating hybrid models. Technical report, AUT00-03, Automatic control laboratory, ETH Zuerich, 2000.

[15] Polylib: A library of polyhedral functions. http://www.irisa.fr-/polylib.