GENETIC ALGORITHM – QUADRATIC PROGRAMMING BASED PREDICTIVE CONTROL FOR MLD SYSTEMS

Jean Thomas [1], Sorin Olaru [2], Jean Buisson [3], Didier Dumur [2]

[1] Industrial Education College Beni Swef, Egypt Phone : +20 (0)12 52 63 476
[2] Supélec F 91192 Gif sur Yvette cedex Phone : +33 (0)1 69 85 13 75
[3] Supélec F 35511 Cesson-Sévigné cedex Phone : +33 (0)2 99 84 45 43
jhh_thomas@yahoo.com, {sorin.olaru, jean.buisson, didier.dumur}@supelec.fr

Abstract: The Mixed Logical Dynamical (MLD) formalism is an efficient modeling framework for hybrid systems described by dynamics, logic and constraints. Furthermore, it allows solving practical problems such as control using for example predictive strategies. However, its main drawback remains the computation load due to the MIQPs to be solved. This paper proposes an alternative optimization technique based on genetic algorithms linked to QP optimization, providing suboptimal solutions for the binary variables and optimal solutions for the continuous variables respectively, in reasonable time even with a long prediction horizon. This strategy is applied in simulation to a three tanks benchmark.

Keywords: Hybrid systems, Mixed Logical Dynamical systems, Model Predictive Control, Genetic Algorithms, Quadratic Programming optimization.

1. 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 (Branicky, *et al.*, 1998), 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 (Bemporad and Morari, 1999). 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 (Fletcher and Leyffer, 1995).

In order to reduce the computational complexity alternative approaches have been developed, e.g. (Stursberg and Engell 2002, Thomas, *et al*. 2004a, b) where different techniques are used to reduce the number of binary optimization variables. An alternative optimization technique based on genetic algorithms is proposed in (Olaru, *et al.*, 2004). It is shown that the genetic algorithms provide suboptimal solutions in reasonable time even with a long prediction horizon. For this purpose, a modified MLD form with only discrete control actions is derived that results in a structure adequate for quadratic (0, 1)-problems formalism. The transformation of continuous inputs to discrete values is based on fuzzy intelligent techniques including an adaptation of the variables sets for a better precision. This approach considerably reduces the computation time and appears to be more effective with longer prediction horizons, as the computation time increases linearly and instead of exponentially with the number of binary optimization variables.

In this direction, this paper proposes a strategy solving the optimization problem related to model predictive control of hybrid systems under the MLD form. This technique combining genetic algorithms and quadratic programming optimization (GA-QP) provides suboptimal solution in reasonable time. As a first step, the genetic algorithm is used to find a suboptimal solution for the binary optimization variables. Then a quadratic programming optimization is applied to determine the optimal solution of the continuous optimization variables under the constraints imposed by these binary variables and the system itself. Even though this technique provides a suboptimal control sequence, it efficiently reduces the computation time that now linearly increases with the number of binary optimization variables. This technique appears to be more simple than the one in (Olaru, *et al.*, 2004), since it does not require modification of the MLD form, elaboration of the new matrices, discretization of the control actions through fuzzy adaptive filter, which takes about 30% of the global computation time at each iteration.

The paper is organized as follows: Section 2 gives a brief description of the MLD form for hybrid systems. General considerations about model predictive control (MPC) and its application to MLD systems are presented in section 3. Section 4 is dedicated to the genetic algorithms. Section 5 examines as the main contribution the GA-QP optimization solution. This strategy is applied in section 6 to a three tanks benchmark. Finally, conclusions are stated in Section 7.

2. 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 (Bemporad and Morari, 1999) 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:

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

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

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 (Torrisi, *et al.*, 2000).

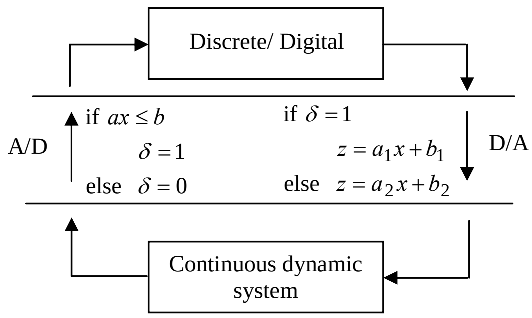

Fig. 1. MLD model structure.

3. 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 (Camacho and Bordons, 1999).

3.1 General consideration

The main idea of predictive control is the use of a plant model 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 sequence is applied to the plant, the whole procedure is repeated again at the next sampling period according to the 'receding' horizon strategy (Boucher and Dumur, 1996). 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) (Clarke, *et al.*, 1987).

3.2 Model Predictive Control for MLD systems

For a 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 the control sequence $\mathbf{u}_t^{t+N-1} = (\mathbf{u}(t) \cdots \mathbf{u}(t+N-1))$ moving the state from $\mathbf{x}(t)$ to $\mathbf{x}_e$, minimizing the cost function:

$$\begin{aligned}\min_{\mathbf{u}_t^{t+N-1}} J(\mathbf{u}_t^{t+N-1}, \mathbf{x}(t)) = \sum_{k=0}^{N_u-1} \|\mathbf{u}(k)-\mathbf{u}_e\|_{Q_1}^2 + \\ \sum_{k=0}^{N-1} \|\boldsymbol{\delta}(k/t)-\boldsymbol{\delta}_e\|_{Q_2}^2 + \|\mathbf{z}(k/t)-\mathbf{z}_e\|_{Q_3}^2 \\ + \|\mathbf{x}(k+1/t)-\mathbf{x}_e\|_{Q_4}^2 + \|\mathbf{y}(k/t)-\mathbf{y}_e\|_{Q_5}^2\end{aligned} \quad (2)$$

subject to (1), and $\mathbf{u}(t)$ constant for $k \geq N_u$, 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, with the notation $\mathbf{x}(k/t) \hat{=} \mathbf{x}(t+k, \mathbf{x}(t), \mathbf{u}_t^{t+k-1})$ (in a similar way for the other input and output variables), $\mathbf{Q}_i = \mathbf{Q}_i^{'} > 0$, for $i = 1,4$, and $\mathbf{Q}_i = \mathbf{Q}_i^{'} \geq 0$, $i = 2,3,5$.

The equality constraints of the MLD form are used to replace the implicit variables in the optimization problem, leading to a MIQP problem under the form:

$$\begin{aligned}F(\chi, x(k)) = \min_{\chi} \frac{1}{2} \boldsymbol{\chi}^{\mathrm{T}} \mathbf{H} \boldsymbol{\chi} + \mathbf{f}^{\mathrm{T}} \boldsymbol{\chi} \\ \text{subject to } \mathbf{c}\boldsymbol{\chi} \prec \mathbf{b}\end{aligned} \quad (3)$$

where the optimization vector is:

$$\begin{aligned}\boldsymbol{\chi} = [\mathbf{u}(k)^{\mathrm{T}}, \cdots, \mathbf{u}(k+N-1)^{\mathrm{T}}, \boldsymbol{\delta}(k)^{\mathrm{T}}, \cdots \\ , \boldsymbol{\delta}(k+N-1)^{\mathrm{T}}, \mathbf{z}(k)^{\mathrm{T}}, \cdots, \mathbf{z}(k+N-1)^{\mathrm{T}}]^{\mathrm{T}}\end{aligned} \quad (4)$$

and the number of binary optimization variables is $L = N_u m_l + N r_l$. In the worst case, the optimization time increases exponentially with this number (Raman and Grossmann, 1991).

Branch and bound (B&B) technique can be used to solve MIQP problem, where the 0-1 combinations are explored through a binary tree, systematically partitioning the feasible region into sub-domains, and generating valid upper and lower bounds at different levels of this tree. Several authors agree on the fact that B&B methods are the most successful solving MIQP problems (Fletcher and Leyffer, 1995).

4. GENETIC ALGORITHMS

As mentioned above, the main drawback of the classical approach with the B&B technique is the exponential complexity with the number of binary optimization variables. It was shown in (Olaru, *et al.*, 2004) that the genetic algorithm as optimization technique can be an efficient solution providing a linear dependency over the binary optimization variables. This section summarizes the main concepts of genetic algorithms for further use for optimization.

The original concept of genetic algorithms was first introduced in the 1970s by John Holland of the University of Michigan (Dulay, 1996; Haupt, 1998). The basic principle is as follows: a random number generator is used to generate a finite set of random design variable vectors, which are referred to as individuals or chromosomes. A set of individuals is called a population. For each individual in the population, the objective function value (or fitness) is calculated. Individuals with a higher fitness are more likely to be chosen for reproduction. Single variables of the chosen individuals are then randomly mutated and crossovers between two parent individuals are performed on parts of the chromosomes. The resulting population generation is then examined as a basis for the next optimization cycle. The algorithm is stopped after a specified number of generations or a certain convergence development, and the individual that has produced the best value of the objective function is considered as output.

The tuning of the GA requires the choice of the variable type and related bounds, the crossover and mutation operators, the population size (affecting the quality of the optimality), the number of maximal genetic operations and the termination criteria.

In the original context, genetic algorithms worked on individuals consisting of binary design variables. Michalewicz introduced a type of genetic algorithm that works directly on continuous (or floating point) individuals (Michalewicz, 1996). In fact, genetic algorithms are able to process either binary or continuous individuals. However, variables of different types generally cannot be mixed due to limitations of current implementations. The classical approach is to transfer all variable types to binary design variables, either explicitly, or through a user interface, as it was realized for the Matlab Genetic Algorithm Optimization Toolbox (GAOT) (Lothrop, 2003). In this case, the value of the continuous variable is calculated as:

$$v_c = l_{\min} + \frac{l_{\max} - l_{\min}}{2^b} . v_b \quad (5)$$

where v_c is the continuous value, v_b is the binary value, $l_{\min}$ the lower variable limit, $l_{\max}$ the upper limit and b the number of bits.

But this discretization may lead to a loss of accuracy through quantification of the continuous optimization variables and may cause chattering or overshoots on the response. To avoid this, it was proposed in (Olaru, *et al.*, 2004) a fuzzy adaptation of the discrete sets of initial continuous variables. To cancel discretization of continuous variables, the next section proposes a technique combining GA and QP to solve the optimization problem of equation (2).

5. GA-QP OPTIMIZATION BASED PREDICTIVE CONTROL FOR MLD SYSTEMS

5.1 Suboptimal solution

The MLD form (1) can be rewritten as follows:

$$\begin{aligned} \mathbf{x}(t+1) &= \mathbf{A}\mathbf{x}(t) + \mathbf{B}_{1c}\mathbf{u}_c(t) + \mathbf{B}_{1l}\mathbf{u}_l(t) + \mathbf{B}_2\boldsymbol{\delta}(t) + \mathbf{B}_3\mathbf{z}(t) \\ \mathbf{y}(t) &= \mathbf{C}\mathbf{x}(t) + \mathbf{D}_{1c}\mathbf{u}_c(t) + \mathbf{D}_{1l}\mathbf{u}_l(t) + \mathbf{D}_2\boldsymbol{\delta}(t) + \mathbf{D}_3\mathbf{z}(t) \\ \mathbf{E}_2\boldsymbol{\delta}(t) + \mathbf{E}_3\mathbf{z}(t) &\le \mathbf{E}_{1c}\mathbf{u}_c(t) + \mathbf{E}_{1l}\mathbf{u}_l(t) + \mathbf{E}_4\mathbf{x}(t) + \mathbf{E}_5 \end{aligned} \quad (6)$$

The genetic algorithm is used to find the suboptimal solution of the binary variables of the MLD form i.e. the individuals including the following variables:

$$\begin{aligned} \chi_b = [\mathbf{u}_l(k)^{\mathrm{T}}, \cdots, \mathbf{u}_l(k+N_u-1)^{\mathrm{T}}, \\ \boldsymbol{\delta}(k)^{\mathrm{T}}, \cdots, \boldsymbol{\delta}(k+N-1)^{\mathrm{T}}]^{\mathrm{T}} \end{aligned} \quad (7)$$

For each individual, a QP optimization problem is solved to find the corresponding optimal solution for the continuous variables, of the form:

$$\begin{aligned} F(\chi_c, \mathbf{x}(k)) = \min \quad & \chi_c^{\mathrm{T}}\mathbf{M}_c^{\mathrm{T}}\mathbf{Q}\mathbf{M}_c\chi_c + \\ & + 2(\mathbf{M}_l\chi_l - \chi_e)^{\mathrm{T}}\mathbf{Q}\mathbf{M}_c\chi_c \end{aligned} \quad (8)$$

where the optimization vector χ_c includes the continuous variables over the prediction horizon:

$$\begin{aligned} \chi_c = [\mathbf{u}_c(k)^{\mathrm{T}}, \cdots, \mathbf{u}_c(k+N_u-1)^{\mathrm{T}}, \\ \mathbf{z}(k)^{\mathrm{T}}, \cdots, \mathbf{z}(k+N-1)^{\mathrm{T}}]^{\mathrm{T}} \end{aligned} \quad (9)$$

and

$$\chi_l = [\mathbf{x}(k)^{\mathrm{T}}, \chi_b^{\mathrm{T}}]^{\mathrm{T}}$$

$$\begin{aligned} \chi_e = [\mathbf{x}_e^{\mathrm{T}}, \cdots, \mathbf{x}_e^{\mathrm{T}}, \mathbf{u}_{le}^{\mathrm{T}}, \cdots, \mathbf{u}_{le}^{\mathrm{T}}, \boldsymbol{\delta}_e^{\mathrm{T}}, \cdots, \boldsymbol{\delta}_e^{\mathrm{T}}, \\ \mathbf{u}_{ce}^{\mathrm{T}}, \cdots, \mathbf{u}_{ce}^{\mathrm{T}}, \mathbf{z}_e^{\mathrm{T}}, \cdots, \mathbf{z}_e^{\mathrm{T}}, \mathbf{y}_e^{\mathrm{T}}, \cdots, \mathbf{y}_e^{\mathrm{T}}]^{\mathrm{T}} \end{aligned}$$

$$\mathbf{M}_c = \begin{bmatrix} \mathbf{B}_{1C} & 0 & \cdots & 0 & \mathbf{B}_3 & 0 & \cdots & 0 \\ \mathbf{A}\mathbf{B}_{1C} & \mathbf{B}_{1c} & & 0 & \mathbf{A}\mathbf{B}_3 & \mathbf{B}_3 & & \\ \vdots & & \ddots & & \vdots & & \ddots & \\ \mathbf{A}^{N-1}\mathbf{B}_{1c} & \mathbf{A}^{N-2}\mathbf{B}_{1c} & \cdots & \mathbf{B}_{1c} & \mathbf{A}^{N-1}\mathbf{B}_3 & \cdots & \mathbf{A}\mathbf{B}_3 & \mathbf{B}_3 \\ 0 & 0 & \cdots & 0 & 0 & 0 & \cdots & 0 \\ \vdots & 0 & \ddots & \vdots & \vdots & 0 & \ddots & \vdots \\ \vdots & \cdots & \ddots & \vdots & \vdots & \cdots & \ddots & \vdots \\ 0 & \cdots & \cdots & 0 & 0 & \cdots & \cdots & 0 \\ 0 & 0 & \cdots & 0 & 0 & 0 & \cdots & 0 \\ \vdots & 0 & \ddots & \vdots & \vdots & 0 & \ddots & \vdots \\ \vdots & \cdots & \ddots & \vdots & \vdots & \cdots & \ddots & \vdots \\ 0 & \cdots & \cdots & 0 & 0 & \cdots & \cdots & 0 \\ \mathbf{I}_{m_c \times m_c} & & & & 0 & 0 & \cdots & 0 \\ & \mathbf{I}_{m_c \times m_c} & & & \vdots & 0 & \ddots & \vdots \\ & & \ddots & & \vdots & \cdots & \ddots & \vdots \\ & & & \mathbf{I}_{m_c \times m_c} & 0 & \cdots & \cdots & 0 \\ 0 & 0 & \cdots & 0 & \mathbf{I}_{r_c \times r_c} & & & \\ \vdots & 0 & \ddots & \vdots & & \mathbf{I}_{r_c \times r_c} & & \\ \vdots & \cdots & \ddots & \vdots & & & \ddots & \\ 0 & \cdots & \cdots & 0 & & & & \mathbf{I}_{r_c \times r_c} \\ \mathbf{D}_{1C} & 0 & \cdots & 0 & \mathbf{D}_3 & 0 & \cdots & 0 \\ \mathbf{C}\mathbf{B}_{1c} & \mathbf{D}_{1C} & \ddots & & \mathbf{C}\mathbf{B}_3 & \mathbf{D}_3 & \ddots & \vdots \\ & & \ddots & 0 & \vdots & & \ddots & 0 \\ \mathbf{C}\mathbf{A}^{N-2}\mathbf{B}_{1c} & \cdots & \mathbf{C}\mathbf{B}_{1c} & \mathbf{D}_{1C} & \mathbf{C}\mathbf{A}^{N-2}\mathbf{B}_3 & \cdots & \mathbf{C}\mathbf{A}\mathbf{B}_3 & \mathbf{D}_3 \end{bmatrix}$$

$$\mathbf{M}_l = \begin{bmatrix} \mathbf{A} & \mathbf{B}_{1l} & 0 & \cdots & 0 & \mathbf{B}_2 & 0 & \cdots & 0 \\ \mathbf{A}^2 & \mathbf{A}\mathbf{B}_{1l} & \mathbf{B}_{1l} & \ddots & \vdots & \mathbf{A}\mathbf{B}_2 & \mathbf{B}_2 & \ddots & \vdots \\ \vdots & \vdots & \cdots & \ddots & 0 & \vdots & \cdots & \ddots & 0 \\ \mathbf{A}^N & \mathbf{A}^{N-1}\mathbf{B}_{1l} & \mathbf{A}^{N-2}\mathbf{B}_{1l} & \cdots & \mathbf{B}_{1l} & \mathbf{A}^{N-1}\mathbf{B}_2 & \cdots & \mathbf{A}\mathbf{B}_2 & \mathbf{B}_2 \\ 0 & \mathbf{I}_{m_l \times m_l} & & & & 0 & 0 & \cdots & 0 \\ \vdots & & \mathbf{I}_{m_l \times m_l} & & & \vdots & 0 & \ddots & \vdots \\ \vdots & & & \ddots & & \vdots & \cdots & \ddots & \vdots \\ 0 & & & & \mathbf{I}_{m_l \times m_l} & 0 & \cdots & \cdots & 0 \\ 0 & 0 & 0 & \cdots & 0 & \mathbf{I}_{r_l \times r_l} & & & 0 \\ \vdots & \vdots & 0 & \ddots & \vdots & & \mathbf{I}_{r_l \times r_l} & & \\ \vdots & \vdots & \cdots & \ddots & \vdots & & & \ddots & \\ 0 & 0 & \cdots & \cdots & 0 & & & & \mathbf{I}_{r_l \times r_l} \\ 0 & 0 & 0 & \cdots & 0 & 0 & 0 & \cdots & 0 \\ \vdots & \vdots & 0 & \ddots & \vdots & \vdots & 0 & \ddots & \vdots \\ \vdots & \vdots & \cdots & \ddots & \vdots & \vdots & \cdots & \ddots & \vdots \\ 0 & 0 & \cdots & \cdots & 0 & 0 & \cdots & \cdots & 0 \\ 0 & 0 & 0 & \cdots & 0 & 0 & 0 & \cdots & 0 \\ \vdots & \vdots & 0 & \ddots & \vdots & \vdots & 0 & \ddots & \vdots \\ \vdots & \vdots & \cdots & \ddots & \vdots & \vdots & \cdots & \ddots & \vdots \\ 0 & 0 & \cdots & \cdots & 0 & 0 & \cdots & \cdots & 0 \\ \mathbf{C} & \mathbf{D}_{1l} & 0 & \cdots & 0 & \mathbf{D}_2 & 0 & \cdots & 0 \\ \mathbf{C}\mathbf{A} & \mathbf{C}\mathbf{B}_{1l} & \mathbf{D}_{1l} & \ddots & \vdots & \mathbf{C}\mathbf{B}_2 & \mathbf{D}_2 & \ddots & \vdots \\ \vdots & \vdots & \cdots & \ddots & 0 & \vdots & \cdots & \ddots & 0 \\ \mathbf{C}\mathbf{A}^{N-1} & \mathbf{C}\mathbf{A}^{N-2}\mathbf{B}_{1l} & \cdots & \mathbf{C}\mathbf{B}_{1l} & \mathbf{D}_{1l} & \mathbf{C}\mathbf{A}^{N-2}\mathbf{B}_2 & \cdots & \mathbf{C}\mathbf{B}_2 & \mathbf{D}_2 \end{bmatrix}$$

$$\begin{aligned} \mathbf{Q} = diag[&diag(\mathbf{Q}_4)_N, diag(\mathbf{Q}_{1b})_{N_u}, diag(\mathbf{Q}_2)_N, \\ &diag(\mathbf{Q}_{1c})_{N_u}, diag(\mathbf{Q}_3)_N, diag(\mathbf{Q}_5)_N] \end{aligned}$$

where $diag(\mathbf{Q}_i)_N$ is a diagonal matrix of $\mathbf{Q}_i$ elements of dimension N.

The QP optimization (8) is solved subject to:

$$\mathbf{a}\chi_c \le \mathbf{b}\chi_l + \mathbf{e} \quad (10)$$

where: $\mathbf{e} = \left[\mathbf{E}_5^{\mathrm{T}} \;\; \mathbf{E}_5^{\mathrm{T}} \;\; \cdots \;\; \mathbf{E}_5^{\mathrm{T}}\right]^{\mathrm{T}}$

$$\mathbf{a} = \begin{bmatrix} -\mathbf{E}_{1C} & 0 & \cdots & 0 & \mathbf{E}_3 & & \cdots & 0 \\ -\mathbf{E}_4\mathbf{B}_{1c} & -\mathbf{E}_{1C} & & & -\mathbf{E}_4\mathbf{B}_3 & \mathbf{E}_3 & & \\ \vdots & & \ddots & & \vdots & & \ddots & \\ -\mathbf{E}_4\mathbf{A}^{N-2}\mathbf{B}_{1c} & -\mathbf{E}_4\mathbf{A}^{N-3}\mathbf{B}_{1c} & \cdots & -\mathbf{E}_{1C} & -\mathbf{E}_4\mathbf{A}^{N-2}\mathbf{B}_3 & \cdots & -\mathbf{E}_4\mathbf{B}_3 & \mathbf{E}_3 \end{bmatrix}$$

$$\mathbf{b} = \begin{bmatrix} \mathbf{E}_4 & \mathbf{E}_{1b} & & \cdots & 0 & -\mathbf{E}_2 & & \cdots & 0 \\ \mathbf{E}_3\mathbf{A} & \mathbf{E}_4\mathbf{B}_{1b} & \mathbf{E}_{1b} & & & \mathbf{E}_4\mathbf{B}_2 & -\mathbf{E}_2 & & \\ \vdots & & & \ddots & & \vdots & & \ddots & \\ \mathbf{E}_3\mathbf{A}^{N-1} & \mathbf{E}_4\mathbf{A}^{N-2}\mathbf{B}_{1b} & \cdots & \mathbf{E}_4\mathbf{B}_{1b} & \mathbf{E}_{1b} & \mathbf{E}_4\mathbf{A}^{N-2}\mathbf{B}_2 & \cdots & \mathbf{E}_4\mathbf{B}_2 & -\mathbf{E}_2 \end{bmatrix}$$

The solution of the optimization problem (8) under the constraints (10) is considered as the objective function value (or fitness) which has to be calculated for each individual in the GA.

5.2 Implementation issue

In many cases, $\boldsymbol{\delta}(k)$ and $\mathbf{z}(k)$ can be calculated from the current state $\mathbf{x}(k)$ which is assumed to be known and from $\mathbf{u}_l(k)$ which is derived from the GA. But, if some δ and/or z elements depend on the continuous control variables $\mathbf{u}_c$, then these variables δ and/or z have to be optimized.

The initialization of the GA has to include feasible individuals, i.e. individuals leading to feasible solution of the QP problem. This can be achieved by choosing among an important number of random individuals only the feasible ones as initial population.

6. APPLICATION

6.1 Description of the benchmark

The proposed control strategy is applied on the three tanks benchmark used by (Bemporad, *et al.*, 1999). 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. 2 (see Dolanc, *et al.*, 1997, 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 ($V_i = 1 \text{ or } 0 \text{ respectively}$). 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:

$$\begin{aligned} \dot{h}_1 &= \frac{1}{A}(Q_1 - Q_{13V1} - Q_{13V13}) \\ \dot{h}_2 &= \frac{1}{A}(Q_2 - Q_{23V2} - Q_{23V23}) \\ \dot{h}_3 &= \frac{1}{A}(Q_{13V1} + Q_{13V13} + Q_{23V2} + Q_{23V23} - Q_N) \end{aligned} \quad (11)$$

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 (Bemporad, *et al.*, 1999), introducing the following variables:

$$\begin{aligned} \mathbf{x} &= [h_1\ \ h_2\ \ h_3]^{\mathrm{T}} \\ \mathbf{u} &= [Q_1\ \ Q_2\ \ V_1\ \ V_2\ \ V_{13}\ \ V_{23}]^{\mathrm{T}} \\ \boldsymbol{\delta} &= [\delta_{01}\ \ \delta_{02}\ \ \delta_{03}]^{\mathrm{T}} \\ \mathbf{z} &= [z_{01}\ \ z_{02}\ \ z_{03}\ \ z_1\ \ z_2\ \ z_{13}\ \ z_{23}]^{\mathrm{T}} \end{aligned} \quad (12)$$

$$\text{where:} \quad \begin{aligned} &[\delta_{0i}(t) = 1] \leftrightarrow [h_i(t) \geq h_v] \quad i = 1,2,3 \\ &z_{0i}(t) = \delta_{0i}(t)(h_i(t) - h_v) \quad i = 1,2,3 \\ &z_i(t) = V_i\,(z_{0i}(t) - z_{03}(t)) \quad i = 1,2 \\ &z_{i3}(t) = V_{i3}\,(h_i(t) - h_3) \quad i = 1,2 \end{aligned} \quad (13)$$

6.2 Application of the GA-QP technique

The proposed control strategy is applied to the three tanks benchmark, where:

$$\mathbf{u}_c = [Q_1\ \ Q_2]^{\mathrm{T}}, \quad \mathbf{u}_l = [V_1\ \ V_2\ \ V_{13}\ \ V_{23}]^{\mathrm{T}} \quad (14)$$

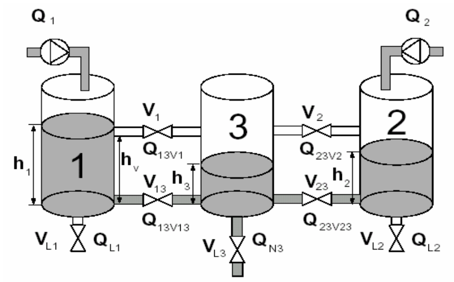

Fig. 2. COSY three tank benchmark system.

In this system, $\boldsymbol{\delta}(k)$ can be calculated from $\mathbf{x}(k)$, and $\mathbf{z}(k)$ deduced from $\mathbf{u}_l(k)$ and $\mathbf{x}(k)$. Thus, for $N = N_u = 3$, the optimization vectors are:

$$\boldsymbol{\chi}_b = \begin{bmatrix} \mathbf{u}_l(k) \\ \mathbf{u}_l(k+1) \\ \mathbf{u}_l(k+2) \\ \boldsymbol{\delta}(k+1) \\ \boldsymbol{\delta}(k+2) \end{bmatrix}, \quad \boldsymbol{\chi}_c = \begin{bmatrix} \mathbf{u}_c(k) \\ \mathbf{u}_c(k+1) \\ \mathbf{u}_c(k+2) \\ \mathbf{z}(k+1) \\ \mathbf{z}(k+2) \end{bmatrix} \quad (15)$$

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 $h_1 = 0.5\,\mathrm{m}$, $h_2 = 0.5\,\mathrm{m}$ and $h_3 = 0.1\,\mathrm{m}$. The GA-QP optimization technique has been applied in simulation to reach the level specification previously given with the two prediction horizons $N = N_u = 3$, a population size equal to 20 and a number of maximal genetic operations equal to 50. The result is presented in Fig. 3 for the three tanks levels and in Fig. 4 for the control signals.

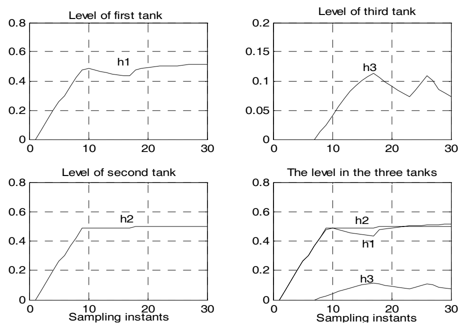

Fig. 3. Water levels in the three tanks.

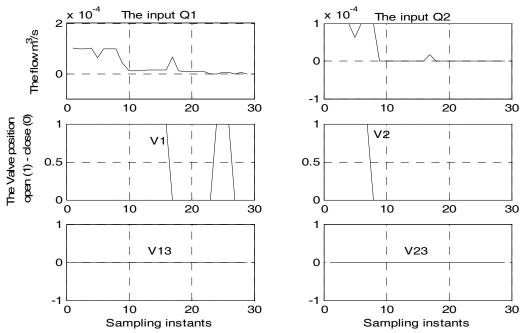

Fig. 4. Controlled variables.

It must be noticed that $h_3 = 0.1$ does not correspond to an equilibrium point. Consequently, the system opens and closes the valves to maintain the level in the third tank around the desired level of 0.1m.

For a comparison purpose between the classical approach of MPC for MLD systems (Bemporad and Morari, 1999) using the MIQP Matlab code (Bemporad and Mignone, 2000), GA with modified MLD form (Olaru *et al.*, 2004) and GA-QP optimization technique, the same previous level specification has been considered with $N = N_u = 3$. Table 1 illustrates the average computation time on a 2.5 GHz PC with 256 Mo of ram.

Table 1 Comparison between the classical approach of MPC for MLD systems, GA with modified MLD form and GA-QP optimization technique

Approach	MPC for MLD	GA with modified MLD	GA-QP optimization technique
Computation time (sec)	498.8	140.9	64.2

7. CONCLUSION

Previous work proposed a strategy based on genetic algorithm with fuzzy adaptive filter and modified MLD form as an alternative technique to apply predictive control for hybrid systems under the MLD formalism. In this paper, another technique combining genetic algorithm and QP optimization is developed, which does not require a modified MLD form or fuzzy adaptive filter, thus reducing the computation time at each iteration. The GA-QP technique offers suboptimal solution in reasonable time even for long prediction horizons. It is therefore more convenient for systems with a large number of binary variables and long prediction horizon as the computation time increases linearly and not exponentially with the number of binary optimization variables.

REFERENCES

Bemporad, A., and D. Mignone (2000). Miqp.m: A Matlab function for solving mixed integer quadratic programs. *Technical Report*.

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

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

Boucher, P. and D. Dumur (1996). *La Commande Prédictive*. Editions Technip, Paris.

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

Camacho E.F. and C. Bordons (1999). *Model Predictive Control*. Springer.

Clarke, D.W., C. Mohtadi and P.S. Tuffs (1987) Generalized Predictive Control, Part I “The basic algorithm”, Part II “Extensions and interpretation”. In: *Automatica*, **23(2)**: 137-160, March.

Dolanc, G., D. Juricic, A. Rakar, J. Petrovcic and D. Vrancic (1998). Three-tank benchmark test. *Technical report Copernicus project report* CT94-02337.

Dulay, N. (1996). Introduction to genetic algorithms. http://www.doc.ic.ac.uk/~nd/surprise_96/journal -/vol1/hmw article1.html.

Fletcher, R. and S. Leyffer 1995. Numerical experience with lower bounds for MIQP branch and bound. *Technical report, Dept. of Mathematics, University of Dundee*, Scotland.

Haupt, R.L. and S.E. Haupt (1998). *Practical genetic algorithms*. Wiley, New York.

Lothrop, K.W. (2003). Conceptual design optimization of a Cis-Lunar transportation architecture using genetic algorithms. *Thesis*.

Michalewicz, Z. (1996). *Genetic Algorithms + Data Structures = Evolution Programs*. Springer, NY.

Olaru, S., J. Thomas, D. Dumur and J. Buisson (2004). Genetic algorithm based model predictive Control for hybrid systems under a modified MLD form. In: *International journal of Hybrid Systems,* **4(1-2)**:113-132.

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

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

Thomas, J., D. Dumur and J. Buisson (2004a). Predictive control of hybrid systems under a multi-MLD formalism with state space polyhedral partition. In: *American Control Conference ACC'2004*, Boston, USA, June.

Thomas, J., D. Dumur and J. Buisson (2004b). Model predictive control for hybrid systems under a state partition based MLD approach (SPMLD). In: *International Conference on Informatics in Control, Automation and Robotics ICINCO*, Setúbal, Portugal, August.

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