Proceedings of the 17th World Congress
The International Federation of Automatic Control
Seoul, Korea, July 6-11, 2008

Hybrid Fuzzy Predictive Control of a Batch Reactor Using a Branch and Bound and a Genetic Algorithm Approach

Javier Causa * Gorazd Karer ** Alfredo Núñez * Doris Sáez *
Igor Škrjanc ** Borut Zupančič **

* *Facultad de Ciencias Físicas y Matemáticas, Universidad de Chile, Chile (e-mail:dsaez@ing.uchile.cl).*
** *Faculty of Electrical Engineering, University of Ljubljana, Slovenia (e-mail: gorazd.karer@fe.uni-lj.si).*

Abstract: The paper deals with model predictive control (MPC) of nonlinear hybrid systems with discrete inputs. It is often required to take into account the hybrid and/or nonlinear nature of real systems, therefore, a hybrid fuzzy model is used for MPC in the paper. Two approaches that are suitable for MPC of nonlinear hybrid systems with discrete inputs are compared on a batch reactor example: a branch & bound and a genetic algorithm. We have established that both algorithms are suitable for controlling such systems. The main advantages of the genetic algorithm are boundedness of computational time in one step and whole computation-efficiency, whereas the main drawbacks are its inherent sub-optimality and the need for suitably tuned parameters. On the other hand, the branch & bound approach does not require parameter tuning and using a suitable cost function provides optimal results in considerably less time than an explicit enumeration method.

1. INTRODUCTION

Model predictive control (MPC) consists of optimizing of the process behavior to obtain optimal future control actions. In the MPC framework the use of non-linear models with continuous and/or discrete variables has been considered in order to obtain better representations of the process nonlinearities.

Firstly, simplified solutions of non-linear fuzzy predictive control methods were developed, such as the fuzzy predictive controller based on the Takagi-Sugeno (T-S) fuzzy model linearization proposed by Roubos et al. [1998]. Espinosa and Vandewalle [1998] propose a fuzzy predictive control algorithm based on the approximation of the free and forced responses of the fuzzy model. Hadjili and Wertz [1999] and Nounou and Passino [1999] describe similar predictive controllers, where the fuzzy predictor is linearly approximated by using constant satisfaction degrees for the future horizons and an analytical solution of a linear MPC is applied. More robust solutions of the fuzzy predictive control strategy have been proposed by Babuska [1998]. However, these solutions require a longer computation time. Mahfouf et al. [2002] consider a T-S fuzzy model with different fuzzy partitions of the input space.

Recently, in order to appropriately control processes that contain discrete and/or continuous variables (hybrid systems), hybrid predictive control techniques were developed. Slupphaug et al. [1997] and Slupphaug and Foss [1997] describe a predictive controller with continuous and integer input variables solved by nonlinear mixed integer programming. Bemporad and Morari [1999] present a predictive control scheme for hybrid systems solved by using Mixed Integer Quadratic Programming (MIQP). Borrelli et al. [2003] propose a finite-time optimal control solution for piecewise affine systems with a quadratic performance criterion. Baotic et al. [2003] present a linear criterion for the proposed algorithm that results in a reduced computation time. Thomas et al. [2004] propose a hybrid predictive controller partitioning in the state-space domain. Beccuti et al. [2003] present a hybrid predictive approach based on a temporal decomposition scheme. On the other hand, Potočnik et al. [2005] propose a hybrid predictive control algorithm with discrete inputs based on a reachability analysis, where the computation time is reduced by building and pruning an evolution tree. In a recent work, Núñez et al. [2006] present a hybrid predictive control strategy based on a fuzzy model. A self-adaptive supervisory predictive functional control for applications in a semi-batch reactor in which the optimal operation is to follow the reference trajectory without significant overshoot is presented in Škrjanc [2007].

Both fuzzy and hybrid predictive controllers correspond to non-linear predictive control strategies that are required to solve an NP-hard problem given by the non-linear optimization problem associated with the predictive objective function and the non-linear predictive model (fuzzy and/or hybrid model). To solve these kinds of NP-hard problems, evolutionary algorithms have been proposed (see Sarimveis and Bafas [2003], Shin and Park [1998], Woolley et al. [1998], Kennedy and Eberhart [2001]).

In this study, the design of Hybrid Fuzzy Predictive Control is described and applied to a batch reactor described in Karer et al. [2007]. In Section 2 we present the control algorithm, which includes a prediction based on a hybrid fuzzy model of the process. Two approaches for solving such an NP-hard problem are described: Branch and Bound (BB) and the Genetic Algorithm (GA). In Section 3 the batch reactor process and its corresponding hybrid fuzzy modelling are described. In Section 4 the results of the experiments are presented. Finally, Section 5 concludes with an analysis, comments and further research directions.

2. MODEL PREDICTIVE CONTROL OF SYSTEMS WITH DISCRETE INPUTS

Model predictive control is an approach where a model of the system is used to predict the future evolution of the system (Camacho and Bordons [1998], Maciejowski [2002]). The most appropriate input vector is established and applied for every time-step. Its determination is an optimization problem that is solved within a finite horizon H, i.e., for a pre-specified number of time-steps ahead. For each time-step k a sequence of optimal input vectors (1) is acquired, which minimizes the selected cost function while considering the eventual constraints of the inputs, outputs and system states. However, only the first vector of the optimal sequence is actually applied during the current time-step. In the next time-step, a new optimal sequence is determined, etc.

$$U_k^{k+H-1} = \{u(k), u(k+1), ..., u(k+H-1)\} \qquad (1)$$

The Hybrid Predictive Control (HPC) strategy is a generalization of Model Predictive Control (MPC), where the prediction model includes both discrete-integer and continuous variables. In this study we propose a hybrid fuzzy prediction model.

In general, a hybrid predictive control design minimizes the following generic objective function. This particular case corresponds to the most common objective function used for predictive control purposes.

$$\min_{\{u(k),u(k+1),\ldots,u(k+N_u-1)\}} J = J_1 + \lambda J_2 \qquad (2)$$

$$J_1 = \sum_{j=N_1}^{N_y} \left(\hat{y}\left(k+j\right) - r\left(k+j\right)\right)^2, \; J_2 = \sum_{j=N_1}^{N_u} \Delta u\left(k+j-1\right)^2 \qquad (3)$$

Here, J is the objective function, $\hat{y}\left(k+j\right)$ corresponding to the j-step ahead prediction for the controlled variable, $r\left(k+j\right)$ is the reference, $\Delta u\left(k+j-1\right)$ is the increment of the control action and λ is the weighting factor. N_1, N_y and N_u are the prediction horizons and the control horizon, respectively. $U_k^{k+N_u-1} = \{u\left(k\right), ..., u\left(k+N_u-1\right)\}$ represents the control action sequence, which corresponds to the optimization variables.

For the hybrid fuzzy predictive control design proposed, the prediction model is given by a non-linear function as a T-S fuzzy hybrid model and the manipulated variable and/or state variable are integer/discrete. This non-linear optimization problem corresponds to NP-hard and therefore, we propose two approaches: a branch and bound (BB) method and a genetic algorithm (GA).

2.1 The Branch and Bound Approach

The control algorithm used in this paper is thoroughly described in Karer et al. [2007] and Potočnik et al. [2005]. Since it is limited to systems with discrete inputs only, the possible evolution of the system over time-steps h up to a maximum prediction horizon H can be illustrated by a tree of evolution. The nodes of the tree represent reachable states, and branches connect two nodes if a transition exists between the corresponding states.

For an insight into the computational complexity issues and the approaches and properties used for dealing with them, see Karer et al. [2007].

2.2 Optimization based on a Genetic Algorithm

The GA method is suitable for NP-hard optimization problems with discrete or integer variables, and therefore the binary codification is not necessary. In other words the genes of the individuals (feasible solutions) are given directly by the integer optimization variables. In addition, gradient computations are not necessary, as in conventional non-linear optimization solvers, which allows us to save a significant amount of computation time.

The optimization based on a GA (Man et al. [1998]) can be described by the following steps:

(1) Initialize a random population of individuals corresponding to the feasible solutions.
(2) Evaluate the objective function for each individual of the current population.
(3) Select random parents from the current population.
(4) Apply genetic operators like *crossover* and/or *mutation* to the parents, for a new generation.
(5) Evaluate the objective function for all the individuals of the generation.
(6) Choose the best individuals according to the best values of the objective function.
(7) Replace the weakest individuals of the previous generation with the best ones of the new generation obtained in Step 6.
(8) If either the value of the objective function reaches a certain tolerance or the maximum number of generations is reached, then the algorithm stops. Otherwise, go to Step 2.

In general, genetic algorithms efficiently cope with nonlinear mixed/integer optimization problems. Another advantage is that the objective function gradient does not need to be calculated, which relaxes the computational effort.

A potential solution of the genetic algorithm is called individual. The individual can be represented by a set of parameters related to the genes of a chromosome and can be described in binary or integer form. The individual represents a possible control action sequence $U_k^{k+N_u-1} = \{u\left(k\right), ..., u\left(k+N_u-1\right)\}$, where each element is called a gene, and the individual length corresponds to the control horizon N_u.

Using genetic evolution, the fittest chromosome is selected to ensure the best offspring. The best parent genes are selected, mixed and recombined for the production of the offspring in the next generation. For the recombination of the genetic population, two fundamental operators are used: crossover and mutation. For the crossover mechanism, the portions of two chromosomes are exchanged with a certain probability in order to produce the offspring. The mutation operator alters each portion randomly with a certain probability (see Man et al. [1998]).

In summary, the proposed genetic algorithm solution provides a solution near to the optimum. The GA-method tuning parameters are the number of individuals, the number of generations, the crossover probability, the mutation probability and the stopping criteria.

3. THE BATCH REACTOR

The control approaches were tested on a simulation example of a real batch reactor that is situated in a pharmaceutical company and is used in the production of medicines. The goal is to control the temperature of the ingredients stirred in the reactor core so that they synthesize into the final product. In order to achieve this, the temperature has to follow the reference trajectory given in the recipe as accurately as possible.

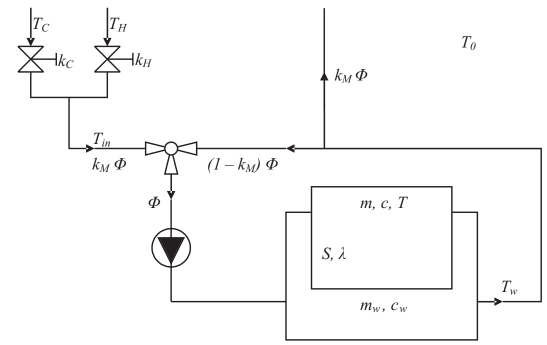

Fig. 1. Scheme of the batch reactor

A scheme of the batch reactor is shown in Fig. 1. The reactor's core (temperature T) is heated or cooled through the reactor's water jacket (temperature T_w). The heating medium in the water jacket is a mixture of fresh input water, which enters the reactor through on/off valves, and reflux water. The water is pumped into the water jacket with a constant flow ϕ. The dynamics of the system depend on the physical properties of the batch reactor, i.e., the mass m and the specific heat capacity c of the ingredients in the reactor's core and in the reactor's water jacket (here, the index w denotes the water jacket). λ is the thermal conductivity, S is the contact area and T_0 is the temperature of the surroundings.

The temperature of the fresh input water T_{in} depends on two inputs: the position of the on/off valves k_H and k_C. However, there are two possible operating modes of the on/off valves. In case $k_C = 1$ and $k_H = 0$, the input water is cool ($T_{in} = T_C = 12\,^0C$), whereas if $k_C = 0$ and $k_H = 1$, the input water is hot ($T_{in} = T_H = 75\,^0C$).

The ratio of fresh input water to reflux water is controlled by the third input, i.e., by the position of the mixing valve k_M. There are six possible ratios that can be set by the mixing valve. The share of fresh input water can be either 0, 0.01, 0.02, 0.05, 0.1 or 1.

We are therefore dealing with a multivariable system with three discrete inputs (k_M, k_H and k_C) and two measurable outputs (T and T_w). Due to the nature of the system, the time constant of the temperature in the water jacket is obviously much shorter than the time constant of the temperature in the reactor's core. Therefore, the batch reactor is considered as a stiff system.

The modelling procedure is explained in detail in Karer et al. [2007].

The sub-model for the temperature in the reactor's core T is shown in (4). The identified system parameters are given below.

$$\hat{T}(k+1) = \mathbf{\Theta}_c^T \, [T_w(k) \;\; T(k)]^T \tag{4}$$

$$\mathbf{\Theta}_c^T = [0.0033 \;\; 0.9967] \tag{5}$$

The sub-model for the temperature in the reactor water jacket T_w has two operating modes, which define the discrete part of the sub-model $q = 1$ is the case when the fresh input water is hot, i.e., $k_C(k) = 0$ and $k_H(k) = 1$; $q = 2$ is the case when the fresh input water is cool, i.e., $k_C(k) = 1$ and $k_H(k) = 0$.

$$q(k) = q(k_C(k), k_H(k)) = \left\{ \begin{array}{l} 1 \text{ if } k_C(k) = 0 \wedge k_H(k) = 1 \\ 2 \text{ if } k_C(k) = 1 \wedge k_H(k) = 0 \end{array} \right\} \tag{6}$$

The system is fuzzyfied with regard to the temperature in the reactor's water jacket $T_w(k)$. Simple triangular functions are used, which ensures that the normalized degrees of fulfillment $\beta_j(T_w)$ are equal to the membership values $\mu_j(T_w)$ across the whole operating range. In this case there are five membership functions, with maximums at 12, 20, 40, 60 and 70^0C, so that the whole operating range is covered.

The output of the model of the temperature in the reactor's water jacket is written in compact form in (7) and (8).

$$\hat{T}_w(k+1) = \boldsymbol{\beta}(k)\; \mathbf{\Theta}_w^T(k)\;\; [T_w(k) \;\; T(k) \;\; k_M(k) \;\; 1]^T \tag{7}$$

$$\mathbf{\Theta}_w(k) = \left\{ \begin{array}{lll} \mathbf{\Theta}_{w1} & \text{if} & q(k) = 1 \\ \mathbf{\Theta}_{w2} & \text{if} & q(k) = 2 \end{array} \right\} \tag{8}$$

$$\mathbf{\Theta}_{w1} = \begin{bmatrix} 0.9453 & 0.9431 & 0.9429 & 0.9396 & 0.7910 \\ 0.0376 & 0.0458 & 0.0395 & 0.0339 & 0.0225 \\ 19.6748 & 16.7605 & 10.5969 & 3.9536 & 1.6856 \\ 0.3021 & 0.2160 & 0.5273 & 1.2701 & 12.0404 \end{bmatrix} \tag{9}$$

$$\mathbf{\Theta}_{w2} = \begin{bmatrix} 0.9803 & 0.9740 & 0.9322 & 0.9076 & 0.8945 \\ 0.0025 & 0.0153 & 0.0466 & 0.0466 & 0.0111 \\ -0.0704 & -0.6956 & -7.8013 & -12.2555 & -18.7457 \\ 0.2707 & 0.2033 & 0.5650 & 1.9179 & 5.6129 \end{bmatrix} \tag{10}$$

4. RESULTS

For the hybrid predictive control optimization problem of the batch reactor we propose the cost function given by (11) (see also Karer et al. [2007]).

$$
\begin{aligned}
J &= w_1 \sum_{h=1}^{N} \left(T(k+h) - T_{ref}(k+h)\right)^2 \\
&+ w_2 \sum_{h=1}^{N} K_C(k+h) K_H(k+h-1) \\
&+ w_3 \sum_{h=1}^{N} \left|K_M(k+h) - K_M(k+h-1)\right| K_H(k+h-1) \\
& w_1 = 1/15,\ w_2 = 15,\ w_3 = 0.03
\end{aligned}
\tag{11}
$$

In this study the prediction horizon considered is $N = 4$. The sampling time of the prediction model equals $T_s = 10$ s. Note that the inputs are allowed to change only every 15 time steps (see Karer et al. [2007]). This means that the time of the allowed input changes, which is denoted here as the control sampling time T_{cs}, equals 150 s. The set of possible input variables $u(k+j-1)$, $j = 1...N$, is defined in (12).

$$
M = \left\{ \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}, \begin{bmatrix} 0.01 \\ 0 \\ 1 \end{bmatrix}, \begin{bmatrix} 0.02 \\ 0 \\ 1 \end{bmatrix}, \begin{bmatrix} 0.05 \\ 0 \\ 1 \end{bmatrix}, \begin{bmatrix} 0.1 \\ 0 \\ 1 \end{bmatrix}, \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix}, \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix} \right\}
\tag{12}
$$

- The first row denotes the mixed valve input $k_M \in \{0, 0.01, 0.02, 0.05, 1\}$.
- The second row is the cool-water on/off valve input $k_C \in \{0, 1\}$.
- The third row denotes the hot-water on/off valve input $k_H \in \{0, 1\}$.

4.1 The Branch and Bound Approach – Results

The results of the experiment using the HFPC-BB approach are shown in Fig. 2 and Fig. 3.

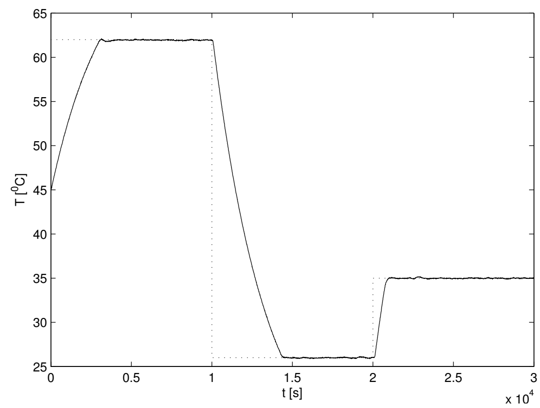

Fig. 2. Results of HFPC based on BB: Core temperature T (solid line) and reference temperature T_{ref} (dotted line)

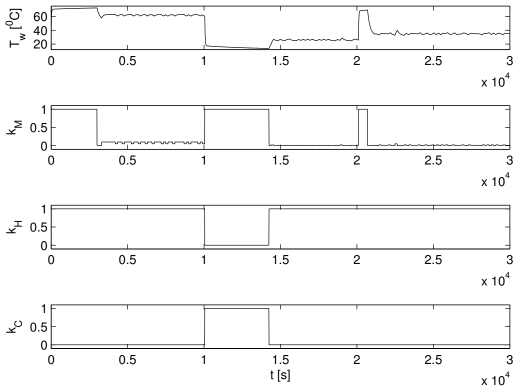

Fig. 3. Results of HFPC based on BB: Other system states

4.2 Optimization based on a Genetic Algorithm – Results

In this case the individuals for the HPC based on a GA are defined as feasible future control action sequences:

$$individual_j = \{u(k), ..., u(k+N_u-1)\}$$

An individual consists on N_u genes and each gene represents one control action.

For simplicity, we consider the following notation for representing the seven possible control actions for the batch reactor:

$$
\begin{gathered}
0 = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}, 1 = \begin{bmatrix} 0.01 \\ 0 \\ 1 \end{bmatrix}, 2 = \begin{bmatrix} 0.02 \\ 0 \\ 1 \end{bmatrix}, 3 = \begin{bmatrix} 0.05 \\ 0 \\ 1 \end{bmatrix}, \\
4 = \begin{bmatrix} 0.1 \\ 0 \\ 1 \end{bmatrix}, 5 = \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix}, 6 = \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix}.
\end{gathered}
\tag{13}
$$

Now, the possible control action $u(k+j-1) \in \{0, 1, 2, 3, 4, 5, 6\}$, which represents the possible values or the states of the input variables.

The procedure for the HFPC-GA consists of:

(1) Initialize a random population of individuals, i.e., create random integer feasible solutions of manipulated variables for the hybrid fuzzy predictive control problem. As an example, the size of the population could be seven individuals per generation. Then, as the control horizon is 4, there are 7^4 possible individuals. However, for the GA per generation, the following population is considered:

$$
Population_i = \begin{bmatrix} individual_1 \\ individual_2 \\ individual_3 \\ individual_4 \\ individual_5 \\ individual_6 \\ individual_7 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 6 & 3 \\ 2 & 1 & 0 & 0 \\ 5 & 4 & 2 & 3 \\ 3 & 6 & 3 & 4 \\ 4 & 1 & 3 & 1 \\ 2 & 5 & 4 & 3 \\ 0 & 3 & 0 & 1 \end{bmatrix}
$$

(2) Evaluate the fitness function for all the initial individuals of the population using equation (11). Note that the prediction $\hat{y}(k+j)$ is calculated recursively by using the future control action. In general

$$\hat{y}(k+j) = f\left(\hat{y}(k+j-1), ..., u(k+j-1), ...\right)$$

where f is a non-linear function defined by a hybrid fuzzy model.

(3) Select random parents from the population (different vectors of the future control actions). For example, Individual 1 and Individual 5 are chosen as the parents:

$$\overbrace{\underbrace{[0\ 1]}_{1A}\underbrace{[6\ 3]}_{1B}}^{individual\ _1} \qquad \overbrace{\underbrace{[4\ 1]}_{5A}\underbrace{[3\ 1]}_{5B}}^{individual\ _5}$$

(4) Apply crossover and mutation to the parents in order to generate an offspring.
After the crossover step

$$\overbrace{\underbrace{[0\ 1]}_{1A}\underbrace{[3\ 1]}_{5B}} \qquad \overbrace{\underbrace{[4\ 1]}_{5A}\underbrace{[6\ 3]}_{1B}}$$

After the mutation step

$$\overbrace{[0\ 1\ 2\ 1]}^{New\ Individual\ _1} \qquad \overbrace{[0\ 1\ 6\ 3]}^{New\ Individual\ _2}$$

(5) Evaluate the fitness given by the objective function (x) of all the individuals of the offspring population.
(6) Select the best individuals according to the objective function.
(7) Replace the weakest individuals from the previous generation with the strongest individuals of the new generation selected in step 6.
(8) If the objective function value reaches the defined tolerance or the maximum generation number is reached (stopping criteria), then stop. Otherwise, go to step 2.

The genetic algorithm approach in HFPC-GA provides a sub-optimal discrete control law close to the optimal one. The tuning parameters of the GA method are the number of individuals, the number of generations, the crossover probability, the mutation probability and the stopping criteria.

Considering a reasonable trade off between accuracy and computational effort, 10 generations with 30 individuals are selected.

The computation time of HFPC-GA is linearly dependent on the generation number, and its slope slightly increases with the number of individuals. Thus, the computation time is smaller than the simulation time (30000s). This means that all the proposed HFPC-GA control strategies are suitable for real-time control in the sense of the time consumed. With 10 generations and 30 individuals, the computation time was approximately 144 s (a 0.48% of the total simulation time) as well as the time at each iteration being smaller than the sampling time.

The results of the experiment are shown in Fig. 4 and Fig. 5. These results are very similar to the ones obtained with HFPC-BB (see Figs. 2 and 3).

Fig. 6 shows the time per iteration for both HFPC-BB and HFPC-GA.With the GA the iteration time remains constant and with BB the time per iteration varies with set-point changes. This feature of HFPC-GA makes it possible to ensure a real-time implementation as HFPC-GA provides solutions within a bounded time per iteration, which is designed to be less than the sampling time.

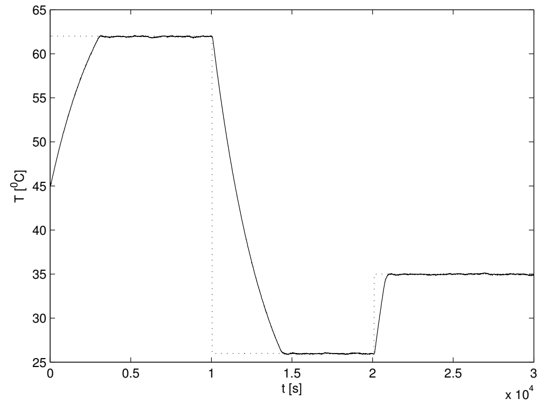

Fig. 4. Results HFPC based on GA: Core temperature T (solid line) and reference temperature T_{ref} (dotted line)

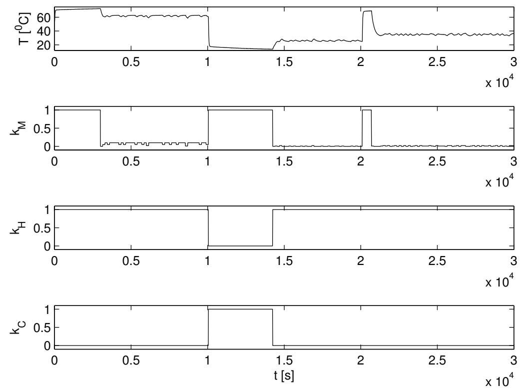

Fig. 5. Results HFPC based on GA: Other system states

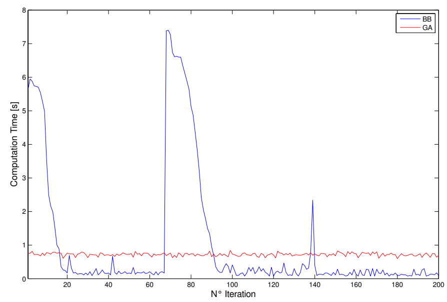

Fig. 6. Time per iteration

Table 1 shows the computation time per algorithm iteration and the objective function values using branch and bound (HFPC-BB), the genetic algorithm (HFPC-GA) and, in addition, explicit enumeration (HFPC-EE).

5. CONCLUSION

We obtained equal mean values of the objective function for BB and EE, and also a similar value was provided by the GA. Thus, the three proposed optimization algorithms

Table 1. Comparison between BB, GA, and EE

	Mean time s	Std time s	Mean J	Std J
BB	220.4	5.3	3500003	0
GA (30,10)	144.45	0.84	3569002	30867
EE	3495	10	3500003	0

allow us to solve the HFPC strategy and to control the temperature of a batch reactor minimizing both the trajectory error and the control energy. However, in terms of computation time, there are significant differences between them. The EE and BB provide the global optimum at each iteration; however, the long computation time required does not allow us to ensure a real-time implementation in the case od EE.

Regarding the HFPC-BB and HFPC-GA strategies, the mean computation time was 220 s and 144 s, respectively. Therefore, a computation time saving of 35% approximately is obtained when using the GA in comparison with the BB. Although HFPC-GA returns a sub-optimal solution at each iteration, the overall behavior of the controlled plant is practically identical to the HFPC-BB (and HFPC-EE), which provides optimal results. On the other hand, the BB approach does not require any parameter tuning. However, the GA ensures a steady and bounded computation time at each iteration, which is critical in real-time applications.

In this study, the HFPC-GA is presented as a heuristic, systematic and efficient algorithm that allows us to solve NP-hard problems as the HFPC strategy. Future work will focus on extending the proposed HFPC-GA to solve predictive control with both discrete and continuous manipulated variables.

ACKNOWLEDGEMENTS

This work was supported in part by the Ministry of Science, Higher Education and Technology of the Republic of Slovenia and by Fondecyt grants 1061156 (Chile) and 7070293 (Chile-Slovenia).

REFERENCES

R Babuska. *Fuzzy Modelling for Control.* KAP, 1998.

M Baotic, F Christophersen, and Manfred Morari. A new algorithm for constrained finite time optimal control of hybrid systems with a linear performance index. University of Cambridge, UK, September 2003. European Control Conference.

A G Beccuti, T Geyer, and M. Morari. Temporal lagrangian decomposition of model predictive control for hybrid systems. pages 2509–2514. European Control Conference, IEEE, December 2003.

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

F Borrelli, M Baotic, Alberto Bemporad, and Manfred Morari. An efficient algorithm for computing the state feedback solution to optimal control of discrete time hybrid systems. pages 4717–4722, Denver, Colorado, USA, June 2003. American Control Conference.

Eduardo F. Camacho and Carlos Bordons. *Model predictive control.* Advanced Textbooks in Control and Signal Processing. Springer-Verlag, London, 1998.

J Espinosa and J Vandewalle. Predictive control using fuzzy models applied to a steam generating unit. pages 151–160. 3rd International FLINS Workshop on Fuzzy Logic and Intelligent Technologies for Nuclear Science Industry, September 1998.

M Hadjili and V Wertz. Generalized predictive control using takagi-sugeno fuzzy models. pages 405–410. IEEE International Symposium on Intelligent Control, Intelligent Systems & Semiotics, IEEE, September 1999.

Gorazd Karer, Gašper Mušič, Igor Škrjanc, and Borut Zupančič. Hybrid fuzzy model-based predictive control of temperature in a batch reactor. *Computers and Chemical Engineering*, 31:1552–1564, 2007.

J Kennedy and R Eberhart. *Swarm Intelligence.* Morgan Kaufmann Publishers, 2001.

Jan Marian Maciejowski. *Predictive control: with constraints.* Prentice Hall, Harlow, 2002.

M Mahfouf, S Kandiah, and D A Linkens. Fuzzy model-based predictive control using an arx structure with feedforward. *Fuzzy Sets and Systems*, 125:39–59, 2002.

K Man, K Tang, and S Kwong. *Genetic Algorithms, Concepts and Designs.* Springer-Verlag, 1998.

H Nounou and K Passino. Fuzzy model predictive control: techniques, stability issues, and examples. pages 423–428. IEEE International Symposium on Intelligent Control, Intelligent Systems & Semiotics, IEEE, September 1999.

A Núñez, D Sáez, S Oblak, and I Škrjanc. Hybrid predictive control based on fuzzy model. pages 9079–9085. IEEE World Congress on Computational Intelligence, IEEE, July 2006.

Boštjan Potočnik, Gašper Mušič, and Borut Zupančič. Model predictive control of discrete time hybrid systems with discrete inputs. *ISA Transactions*, 44(2):199–211, 2005.

J Roubos, R Babuska, P Bruijn, and H Verbruggen. Predictive control by local linearization of a takagi-sugeno fuzzy model. In *Proceedings of the IEEE International Conference on Fuzzy Systems*, pages 37–42, 1998.

H Sarimveis and G Bafas. Fuzzy model predictive control of non-linear processes using genetic algorithms. *Fuzzy Sets and Systems*, 139:59–80, 2003.

S C Shin and S B Park. Ga - based predictive control for nonlinear processes. *IEEE Electronics Letters*, 34(20): 1980–1981, 1998.

O Slupphaug and B Foss. Model predictive control for a class of hybrid systems. European Control Conference, EUCA, July 1997.

O Slupphaug, J Vada, and B Foss. Mpc in systems with continuous and discrete control inputs. American Control Conference, June 1997.

J Thomas, D Dumur, and J Buisson. Predictive control of hybrid systems under a multi-mld formalism with state space polyhedral partition. Boston, Massachusetss, USA, July 2004. American Control Conference.

Igor Škrjanc. Self-adaptive supervisory predictive functional control of a hybrid semi-batch reactor with constraints. *Chemical Engineering Journal*, doi:10.1016/j.cej.2007.04.012, 2007.

I Woolley, C Kambhampati, D Sandoz, and K Warwick. Intelligent control toolkit for an advanced control system. pages 445–450. UKACC International Conference on Control, IEE, September 1998.