USING FUZZY LOGIC TO TUNE OPTIMIZATION PARAMETERS IN NEURAL MODEL BASED PREDICTIVE CONTROL TECHNIQUE*

SANCHEZ, GUSTAVO
CCS: 91537
PO BOX 025323
Miami, FL. 33102-5323
E-mail: gsanchez@usb.ve

STREFEZZA, MIGUEL
Universidad Simón Bolívar, Departamento de Procesos y Sistemas. Universidad Simón Bolivar.

REYES, ORLANDO
Universidad Central de Venezuela, Facultad de Ingeniería, Escuela de Ingeniería Mecánica.

CASTILLO, EDUARDO
Universidad Nacional Experimental Politécnica de la Fuerza Armada Nacional (UNEFA), Venezuela.

ABSTRACT
The underlying idea of Model Based Predictive control can be summarized as follows: A plant model is used in order to predict its future behavior. Thus, Neural Networks (NN) are used in this work to build the target plant dynamic model. The control signal is calculated by means of an optimization problem, solved at each sampling period. Selecting "good" optimization parameters ("weights") is a very important issue. However, it is not easy to select these parameters so that some optimization requirements are achieved. In this work, we propose an algorithm for weights fuzzy self-tuning. Optimization requirements are translated in terms of fuzzy rules. As an application, this design technique was applied to solve the perturbation rejection problem for a multivariable benchmark "Continuous Stirred Tank Reactor" (CSTR).

KEYWORDS. Non-Linear Systems and Modeling. Process Control.

INTRODUCTION

Model Based Predictive Control (MBPC) is a well-known technique [1] that employs a model of the plant in order to predict its future behavior. In this approach, the control signal is calculated by means of an optimization problem, solved at each sampling period. The high computational complexity has restricted the use of this technology to relatively slow systems encountered in the chemical industry [2]. In other hand, Neural Networks (NN) have been used succesfully in order to model and control complex nonlinear plant [3-4]. In this work, NN are used to "learn" the nonlinear dynamics of the target plant: a multivariable benchmark "Continuous Stirred Tank Reactor" (CSTR). This

* This research was supported by CONICIT grant N° S1-99000111

model is based on a recurrent Elman type NN, which acts as a nonlinear "universal approximator".

Whoever has worked with MBPC knows that selecting "good" weights is a crucial issue to solve the optimization problem. Yet, it is not easy to select these weights so that some optimization requirements (for instance, balanced optimization terms) are achieved. In general, there exist some "rules of thumb" that allow selecting it heuristically, by "trial and error" procedures based on simulations results. In this way, finding a good parameters set can be long and annoying. In this work, we propose a weight self-tuning procedure using a Fuzzy Inference System (FIS). This kind of intelligent systems has been used to help making decisions in many engineering fields. The self-tuning procedure is run “on-line” during a training period before starting the MBPC optimization at each sampling period. Optimization requirements are translated in terms of fuzzy rules, whose input variables are the same as the optimization problem.

The following exposition is organized as follows. First, a process description is given. Then, the controller design method is described in detail. Simulations results are given to show the controller ability to reject perturbations. Finally some conclusions are presented .

PROCESS DESCRIPTION

The controlled process, depicted in **Fig. 1**, consists on a benchmark CSTR, which is a multivariable highly coupled and nonlinear chemical plant. It has been described in [5] and is adopted unchanged in this paper. Within the tank, it is assumed that the liquid can be considered homogeneous. The reaction takes place at the temperature value T and concentration values Ca and Cb. The molar input flow of product A, labeled F, has a concentration Ca0 and its temperature is T_0. The heat Q_k is removed by means of a cooling jacket. Energy and matter balance may be set for the whole reactor volume.

CONTROLLER DESIGN METHOD.

The control scheme is shown in **Fig.2**. The controlled process has two inputs (F and Qk) and two outputs (Cb and T). The proposed controller consists mainly of three different sub-systems:

1. A “Neuronal Model”.
2. An Optimizer.
3. A Fuzzy “Weight Tuner”.

Each controller block is described in following sections.

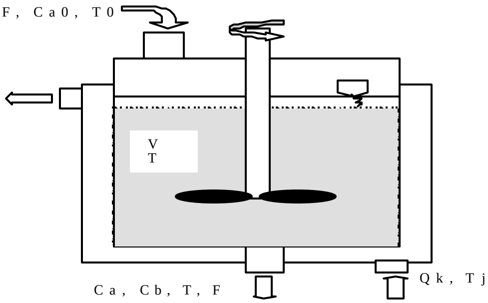

Fig.1. Process Description

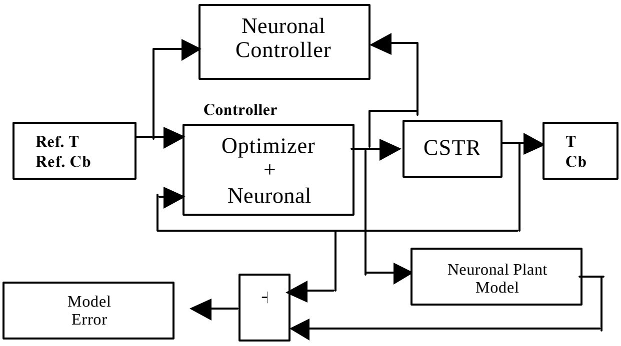

Fig.2. The control scheme.

The neuronal model.

The recurrent Elman type NN structure is shown in **Fig.3**. It was adapted from [6]. It is a simple not coupled structure because there is not interaction between units of hidden layer belonging to different outputs. Each hidden layer has 10 units. When this number was increased the quality of the model did not improve. A tangential function was used in the hidden layer and a linear function in the output layer. In order to train the net, several simulations were carried out, letting some perturbations act on the process.

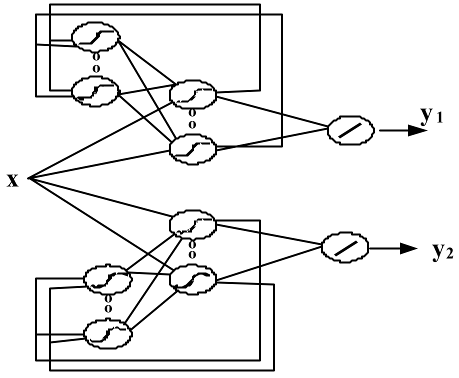

Fig.3 Recurrent NN.

The control goal was rejecting perturbations in variable T_0. An integral action was included in the process (which does not have one itself) in order to reduce steady state offsets. The NN was trained taking into account this integral action. Simulations were carried and data was collected using a *Simulink* block model. Training was done off-line using *MATLAB NN Toolbox*.

The **NN** model equations are:

$$y_1(t) = T(t) \quad y_{r1}(t) = \mathrm{y1} \quad (\text{Set - Point}) \quad u_1(t) = F(t) \quad \hat{y}_1(t+1) = N_1(X(t))$$

$$y_2(t) = Cb(t) \quad \mathrm{y}_{\mathrm{r2}}(\mathrm{t}) = \mathrm{y2} \quad (\text{Set - Point}) \quad u_2(t) = Q_k(t) \quad \hat{y}_2(t+1) = N_2(X(t))$$

Optimizer

The optimization program was written using the *MATLAB Optimization Toolbox* function "fmincon", that minimizes a multivariable scalar function with restrictions. The optimization problem was stated as follows.

At each sampling period, solve:

$$\min_{\substack{u_1(t+i) \\ u_2(t+i) \\ i \in 1,2,\dots nu}} \quad J(t+1) = \left(\begin{array}{l} \displaystyle\sum_{i=1}^{nu} a_{1i} e_1(t+i)^2 + a_{2i} e_1(t+i)^2 \\ + \displaystyle\sum_{i=1}^{nu} l_{1i} \Delta u_1(t+i)^2 + l_{2i} \Delta u_2(t+i)^2 \end{array} \right)$$

where,

$$e_1(t) = y_{r1}(t) - \hat{y}_{1i}(t) \quad \Delta u_1(t) = u_1(t) - u_1(t-1)$$
$$e_2(t) = y_{r2}(t) - \hat{y}_{2i}(t) \quad \Delta u_2(t) = u_2(t) - u_2(t-1)$$

Parameters á$_{1i}$, á$_{2i}$, ë$_{1i}$, ë$_{2i}$ must be tuned carefully to weigh the importance of each term. In fact this is a key subject in the problem. Selecting it heuristically, by "trial and error" procedures based on simulations results, can be long, annoying or even impossible. In this work we propose a fuzzy "self-tuning" of these parameters (called "weights").

Fuzzy Weight Tuner.

The Fuzzy Inference System (FIS) are Takagi-Sugeno Zero Order, PI-like type. In fact four tables were used to determine values for á$_{1i}$, á$_{2i}$, ë$_{1i}$, ë$_{2i}$. Fuzzy input variables are e1, e2, Äu1 and Äu2. An IF-THEN type fuzzy rule example is shown next:

"**IF** E1(i) is BB and E2(i) is ZZ **THEN** Äá$_{1i}$ = 0 , Äá$_{2i}$ = 4"

The main optimization requirement that has been considered was trying to balance the influence of each term on the cost functional *J*. In **Tab. 1**, conclusion values Ää$_{1i}$ and Ää$_{2i}$ are shown for fuzzy rules whose input variables e1(i) and e2(i).

Tab. 1. Look-up table for fuzzy rules.

Ää$_{1i}$ / Ää$_{2i}$	E1SB(i)	E1SS(i)	E1ZZ(i)	E1BS(i)	E1BS(i)
E2SB (i)	0/0	0/1	0/2	0/3	0/4
E2SS(i)	1/0	0/0	0/1	0/2	0/3
E2ZZ(i)	2/0	1/0	0/0	0/1	0/2
E2BS(i)	3/0	2/0	1/0	0/0	0/1
E2BB(i)	4/0	3/0	2/0	1/0	0/0

SIMULATION RESULTS

In Fig.5 and Fig.6 results of an open loop simulation for Cb (mol/l) and T(°C), with perturbation steps in T_0, are shown.

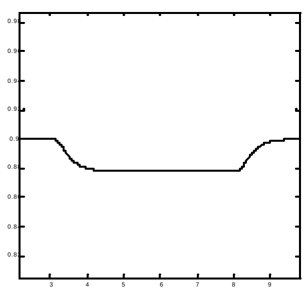

Fig.5. Cb(mol/lts) open-loop.

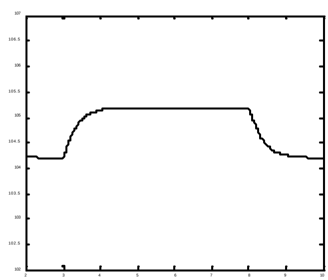

Fig.6. T(°C) open-loop.

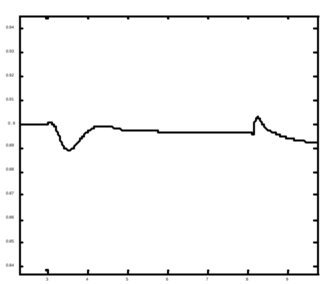

Fig.7. Cb(mol/lts) closed-loop.

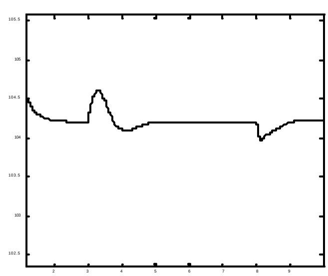

Fig.8. Cb(mol/lts) closed-loop.

In order to show the controller performance, step perturbations were simulated in temperature T_0 in closed-loop mode. Fig.7 and Fig.8 show results of this simulation.

CONCLUSIONS

Designers have to cope with many practical problems when trying to use advanced control techniques. In fact, the main idea behind this work was to develop a method for weights self-tuning in order to ease the controller design. The proposed method shows good results and has a major advantage because of its simplicity. Simulations were carried in a standard PC Intel Celerom 766Mhz using *MATLAB* version 6.0. It was found that with 3000 iterations ($\approx$2hours), dependent on fuzzy look-up tables values, parameter vectors were well tuned, better than all values calculated by "trial and error". The method could be easily implemented with standard industrial controllers. However, improvements in fuzzy rules could be achieved by using more powerful techniques for optimizing conclusions and fuzzy variable membership.

REFERENCES

[1] S. J. Qin and T. A. Badgwell. « **An overview of industrial model predictive control technology** ». *In Je rey C. Kantor, Carlos E. Garcia, and Brice Carnahan, editors, Fifth International Conference on Chemical Process Control, AIChE Symposium* Series 316, 93, pages 232-256. AIChE, 1997.

[2] Donkelaar, Edwin Van. « **Improving of efficiency in Identification and Model Predictive Control of Industrial Process** ». *PhD Thesis. Technische Universiteit Delft, The Neederlands.2000.*

[3] Hunt,K., Sbarbaro, D.,Zbikowski, R.and Gawthrop, G. « **Neural networks for control systems: a survey** ». *Automatica, 28, nº 6, 1083-1112, 1992.*

[4] Zhang J. and Morris A.J. « **Long Range Predictive Control of Nonlinear Processes Based on Recurrent Neuro-Fuzzy Network Models** ». *Neural Computing and Applications.50-59.* 2000.

[5] Chen, H. and Allgower, F. « **Nonlinear predictive control of a benchmark CSTR**». *Proceedings of the 3rd European Control Conference, Rome-Italy.1995.*

[6] Derrick H. Nguyen and Bernard Widrow. « **Neural Networks for Self-Learning Control Systems** ». *IEEE Control Systems Magazine. 1990.*