FUZZY-NEURO PREDICTIVE CONTROL, TUNED BY GENETIC ALGORITHMS, APPLIED TO A FERMENTATION PROCESS

João A. Fabro, L. V. R. Arruda

Laboratório de Automação e Sistemas de Controle Avançado, Programa de Pós-Graduação em Engenharia Elétrica e Informática Industrial, Centro Federal de Educação Tecnológica do Paraná
Av. Sete de Setembro, 3165 CEP 80230-901, Curitiba, PR, Brasil
E-mails: {joao , arruda} @cpgei.cefetpr.br

Abstract— This paper proposes the development of a fuzzy predictive control. Genetic algorithms (GA's) are used to automatically tune the controller. A recurrent neural network is used to identify the process, and then provides predictions about the process behavior, based on control actions applied to the system. These predictions are used by the fuzzy controller, in order to accomplish a better control of an alcoholic fermentation process from the chemical industry. This problem has been chosen due to its non-linearity and inertial characteristics that make it hard to control by standard controllers. Comparison of performance is made with non-predictive approaches(PID and Fuzzy-PD), and also with another predictive approach, GPC(Generalized Predictive Control).

Key Words— Predictive control; Genetic algorithms; Fuzzy control; Neural Networks; Adaptive Control.

1 Introduction

The development of control systems to non-linear processes is an active research area. Classic control theory deals very well with linear processes, but have several limitations when faced with non-linear problems. The standard procedure is to find equilibrium points where the system can be considered linear, and use a linear controller for each point. One of the best known example is the gain schedule control (Åström and Wittenmark, 1995). Due to the difficulty in modelling of non-linear systems, the usual approach is to use control system that does not require an accurate mathematical model, but that incorporates some heuristical knowledge of how to control the system. This class of controllers are called "intelligent control", because it uses the techniques of artificial intelligence to represent, manipulate and implement the heuristic knowledge (Passino and Yurkovich, 1998). Among these control techniques, there are the Artificial Neural Networks (Rumelhart et al., 1986), the Fuzzy Control (Zadeh, 1973), and the Genetic Algorithms (Holland, 1975). These three techniques together are called "Computational Intelligence" (Zurada et al., 1994).

In this paper it is proposed an architecture for predictive control that combines these three techniques. A recurrent neural network is used to identify the process, and thus provides a prediction of the controlled system behavior. This information is then used by a fuzzy predictive controller. The controller is fine tuned by a genetic algorithm.

This paper is organized as follows. Section 2 presents a fermentation process that is used for simulations. This process presents several characteristics, such as non-linearity, non-minimul phase and considerable accommodation time, that present the necessity for an advanced control algorithm. Section 3 presents a fuzzy PD controller for the ship. In section 4 is presented the genetic algorithm used to tune the fuzzy controller. Section 5 presents the recurrent neural network used to identify the systems behavior (and to provide predictive information about it), and in section 6 is presented the proposed fuzzy-predictive controller, with comparisons and experiments. Section 7 presents a discussion of the results and the conclusions.

2 Alcoholic Fermentation Process

The alcoholic fermentation process used in this paper simulations was proposed by (Maher, 1995), and is depicted in figure 1. There are two controlled variables, F_{in} and F_{out}, that represent the input flow of substrate, and the output flow of product, respectively. The process has four state variables: concentrations of the substrate (S), of the biomass (C), and of the product (P), and volume inside the process (V). The concentrations are given in grams/liter (g/l), and the volume in liters (l).

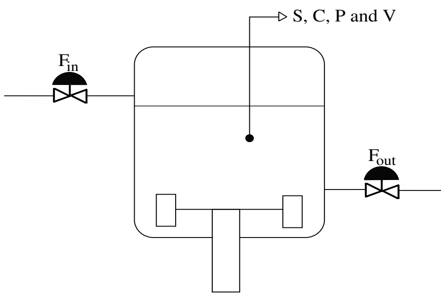

Figure 1. Alcoholic Fermentation Process

The value of each of the state variables is given by the following set of differential equations.

$$\frac{dS}{dt} = -\frac{1}{Y_{C/S}}\mu C + \frac{F_{in}}{V}S_a - \frac{F_{out}}{V}S \qquad (1)$$

$$\frac{dC}{dt} = \mu C - \frac{F_{out}}{V}S \qquad (2)$$

$$\frac{dP}{dt} = \frac{Y_{P/S}}{Y_{C/S}}\mu C - \frac{F_{out}}{V}P \qquad (3)$$

$$\frac{dV}{dt} = F_{in} - F_{out} \qquad (4)$$

Where S_a is the concentration of substrate at the feed, and $Y_{P/S}$ and $Y_{C/s}$ are the constants of conversion of biomass and product, respectively. The value of μ is equivalent to the function of biomass grow, given by:

$$\mu = \mu_0 \frac{S}{K_s + S}\left(1 - \frac{P}{P_m}\right) \qquad (5)$$

These equations are solved for each time-step of the simulation, using the 4^{th} Order *Runge-Kutta* Algorithm.

3 Fuzzy PD Control

The first step was the development of a Fuzzy Logic Controller(FLC) for the process. The first controller developed was a PD one (Proportional Derivative), that uses the information about the error between the product concentrations and the reference signal, and the derivative of this error. A reference signal used for the concentrations of product is presented in figure 2.

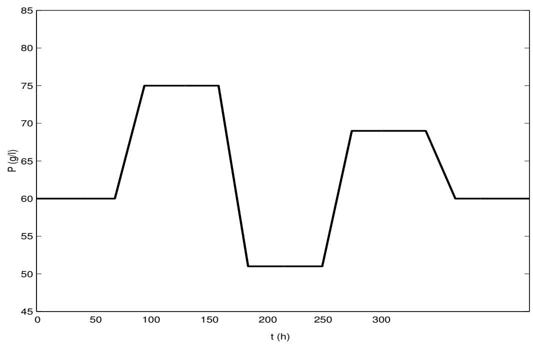

Figure 2. Reference Signal

To automatically tune the fuzzy controllers developed in this work, Genetic Algorithms were used. The tuning of FLC's is done by adjusting the position and shape of the membership functions that compose the linguistic variables used by the controller. In the case of a PD control, the variables are the **error**, the derivative of the error(**deltaerror**) and the control the opening and closing of the two valves, one controlling the input of substrate, and the other controlling the output of product.(**control**). The next section presents the genetic algorithm used to achieve the automatic tuning of the controller.

4 Use of Genetic Algorithms to Fine Tune Fuzzy Logic Controllers

Genetic Algorithms (GA's) are a technique of optimization and search proposed by Holland (Holland, 1975). This technique is based on the theory of biological evolution, and uses operators such as mutation and crossover to find good solutions for a great range of problems (Goldberg, 1989). The only requisites for the use of GA for a given problem are:

- that a solution for the problem can be coded, and this codification is called an chromosome or individual;
- that exists an evaluation function, called fitness function, that gives a score to each individual.

The genetic search starts with a "population" of randomly generated solutions, and uses the evaluation function and the genetic operators of crossover and mutation to improve the solutions, searching for better solutions. This technique has proved efficient in dealing with diverse kinds of optimization and search problems, such as the travelling salesman, and also to find good controllers for mobile vehicles (Michalewicz, 1996).

There are three ways to use genetic algorithms and fuzzy controllers together (Alander, 1997). The first approach, presented by Karr (Karr, 1991), uses GA's to find the position and shape of the membership functions. These parameters are coded in a chromosome, and the genetic search finds the functions that best control the system, given some evaluation function. This approach has the advantage that the controller can adapt to changes in the process during the interactions, turning the FLC into an adaptive controller. The second approach uses the GA to evolve a fuzzy rule base. In this approach, fuzzy rules to control the system are generated and evaluated by the GA, leading to a set of rules that best represents the knowledge about how to control the system (Bonarini, 1996) (Hoffmann, 2001).

The third approach is the conjunction of the previous ones, and uses GA's to simultaneously evolve both rule base and membership functions (Homaifar and McCormick, 1995). The advantage of this approach is that no "a priori" knowledge about the controlled system is needed, so this is

Control		DeltaError						
		NB	**NS**	**NVS**	**QZ**	**PVS**	**PS**	**PB**
E	**NB**	PB	PB	PB	PB	PS	PVS	QZ
R	**NS**	PB	PB	PB	PS	PVS	QZ	NVS
R	**NVS**	PB	PB	PS	PVS	QZ	NVS	NS
O	**QZ**	PB	PS	PVS	QZ	NVS	NS	NB
R	**PVS**	PS	PVS	QZ	NVS	NS	NB	NB
	PS	PVS	QZ	NVS	NS	NB	NB	NB
	PB	QZ	NVS	NS	NB	NB	NB	NB

Table 1. Fuzzy rulebase for the controllers

the better choice when there are little information about the process.

In this work, the first approach has been used. The rule base for the fuzzy PD control is well known (Passino and Yurkovich, 1998), and is presented in table 1. Each linguistic variable has 5 membership functions, that are: NB (Negative Big), NS (Negative Small), QZ (Quasi Zero), PS (Positive Small) and PB (Positive Big). To use GA's to fine tune the controller, it is necessary to define a codification for the controller parameters, and an fitness function.

4.1 Codification of the FLC for tuning by GA

Each membership function is coded as a trapezoid, thus having 4 parameters. The figure 3 present the parameters p_1, p_2, p_3 and p_4 that are used to code each function. These numbers were coded with real numbers, as proposed by Michalewicz (1996). With 5 of these units, it is possible to compose one linguistic variable, so the complete controller has 15 trapezoids (5 x 3), for the three variables. The defuzzyfication method used was the center of area.

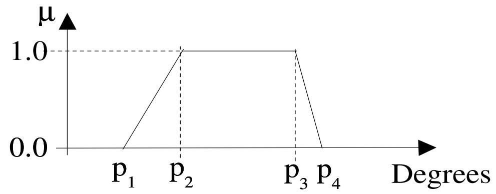

Figure 3. Example of codification of a membership function

4.2 The fitness function

To evaluate the effectiveness of each controller, the process simulation has been used. The controller acted over the simulated system, trying to make it follow the reference (figure 2). The total sum of the errors over this simulation was the evaluation of the controller, thus the GA minimized this value. The error of the controller at each step is the difference between the actual concentration of the product and the reference signal applied at that time step. It is not possible to the system to follow exactly the reference signal, due to its inertia, but the objective of the genetic search is to minimize the sum of these errors.

4.3 Simulations with the Fuzzy-PD FLC

Initially, the genetic algorithm was used to evolve a Fuzzy-PD controller for the ship. Each simulation has 3000 steps. The population was of 250 controllers, and 50 generations were executed. The rate of crossover was of 90% and the mutation rate of 10%.

The best controller found had the behavior shown in figure 4.

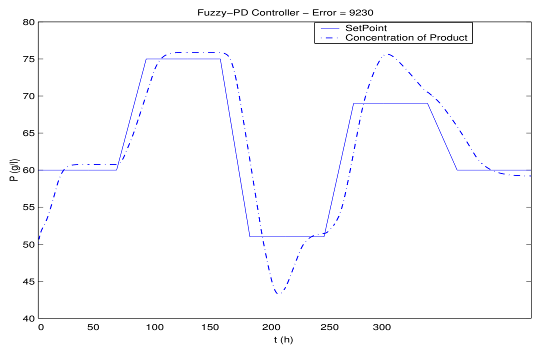

Figure 4. Best Fuzzy PD Controller found by GA

The idea of using predictive information to control this process arise from the observation that the controller couldn't avoid the large overshot when controlling the ship, due to the large inertia of this process. Thus, the only way to avoid (or at least minimize) the error was to use predictive information about the expected behavior of the system. With this information, the controller would be able to take the right action to avoid the overshot. To obtain the information about the future behavior of the controlled process, a recurrent neural network was used to identify the process, and then provide information about its behavior beforehand.

5 Recurrent Neural Network for Prediction

To obtain the prediction for the future behavior of the controlled system, it is necessary to find a model to the process. To achieve this goal, neural networks have been chosen. The data for training was obtained by simulating the process with the software Matlab[a].

The concentrations of product and substrate depends of the inputs, but also of the present rate of change (variation) of the concentrations. So it was necessary to include recurrent connections to the net (Lang and Hinton, 1988). In this case, three previous outputs served as input. Three delayed inputs where used. These 6 inputs, plus the actual control input and the bias, give a net of 8 inputs. The output was the predicted variation of the concentration, and the net has 41 neurons at the hidden layer. All these parameters were obtained by experiment. In future works, it is planned to use constructive algorithms(Kwok and Yeung, 1996) to find the best parameters during the training process, so that this same approach can be used to obtain the model of other systems.

[a]MATLAB is a trademark of MathWorks, Inc.

6 Fuzzy Predictive Control

The fuzzy-PD controller was altered to use the information provided by the recurrent neural network. This was done by the substitution of the **error** variable by the **predicted error**, obtained from the neural network.

Thus, the controller takes actions to minimize the error in the future, acting in a predictive way. During the experiments with this approach, the best solution was found using the 8 steps ahead predicted error. This means that the recurrent neural network returns the prediction for the concentration of the product for eight tenths of hour latter. Due to the large inertia of the process, it is necessary to wait some time until the control actions effect the process. The prediction ahead of time is obtained by applying a given control action to the predictive network for 8 simulation steps. The complete fuzzy-predictive architecture is presented schematically on figure 5.

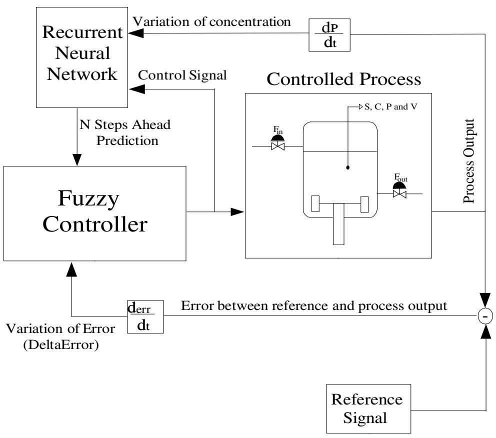

Figure 5. Proposed Control System Architecture

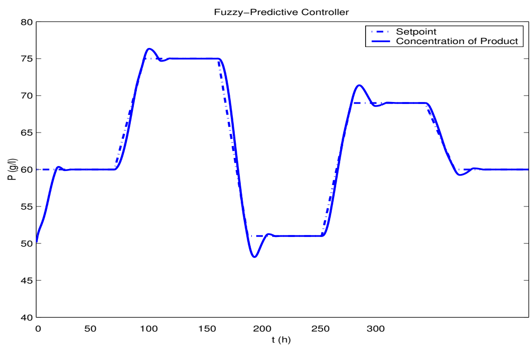

Figure 6. Predictive Controller Output

The predictive controller has a better solution than the PD, as can be seen in figure 6, where the size of the peaks when getting to the setpoint are much smaller than that of the fuzzy-PD controller. This is due to the predictive actions taken by this new approach. Another advantage is that the system gets to the setpoint faster, because the control actions are strong at first, and when the system gets closer to the setpoint, the reverse action is taken to cancel the inertia of the ship, and then it stabilizes around the setpoint. The control actions are shown in figure **??**.

7 Conclusions

This paper proposes a fuzzy-predictive controller, tuned by genetic algorithms. The proposed architecture uses recurrent neural networks to provide predictive information about the behavior of the system, after training. This information is used by a fuzzy controller to take predictive actions, thus having time to overcome the large inertia of the system. The proposed controller has achieved better control over the system during simulations, in comparison to the fuzzy-PD controller.

In future work, constructive algorithms will be used to automatically adjust the recurrent neural network architecture. The number of delayed inputs and outputs, as well as the number of neurons, will be automatically found by these techniques, making it easier for the designer to apply this control approach to other problems.

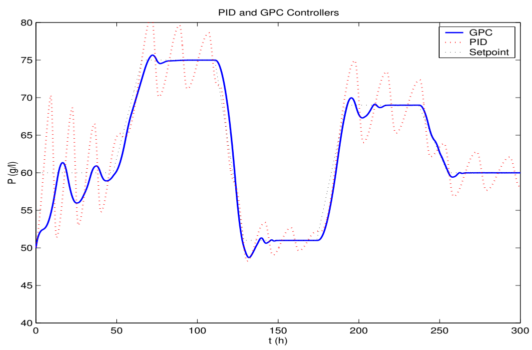

Figure 7. Output of PID and GPC Controllers

Acknowledgements

The first author is at CEFET/PR pursuing his PhD, when on leave from UNIOESTE - State University of West-Paraná, and thanks CAPES for his scholarship.

References

Alander, J. T. (1997). An indexed bibliography of genetic algorithms with fuzzy logic, *in* W. Pedrycz (ed.), *Fuzzy Evolutionary Computation*, Kluwer Academic, Boston, pp. 299–318.

Åström, K. J. and Wittenmark, B. (1995). *Adaptive Control*, 2nd edition edn, Addison - Wesley, USA.

Bonarini, A. (1996). Evolutionary Learning of Fuzzy rules: competition and cooperation, *in* W. Pedrycz (ed.), *Fuzzy Modelling: Paradigms and Practice*, Norwell, MA: Kluwer Academic Press, pp. 265–284.

Goldberg, S. E. (1989). *Genetic Algorithm*, Addison-Wesley.

Hoffmann, F. (2001). Evolutionary algorithms for fuzzy control system design, *Proceedings of the IEEE - Special issue on Industrial Innovation using Soft Computing* **89**(9): 1318 – 1333.

Holland, J. H. (1975). *Adaption in Natural and Artificial Systems*, University of Michigan, Ann Arbor.

Homaifar, A. and McCormick, V. E. (1995). Simultaneous design of membership functions and rule sets for fuzzy controllers using genetic algorithms, *IEEE Transactions on Fuzzy Systems* **3**(2): 129–139.

Karr, C. L. (1991). Genetic algorithms for fuzzy controllers, *AI Expert* **6**(2): 26–33.

Kwok, T. Y. and Yeung, D. Y. (1996). Constructive algorithms for structure learning in feedforward neural networks for regression problems, *IEEE Transactions on Neural Networks* **7**: 1168–1183.

Lang, K. and Hinton, G. (1988). The development of the time-delay neural network architecture for speech recognition, *Technical Report CMU-CS-88-152*, Carnegie-Mellon University, Pittsburgh, PA.

Maher, M. (1995). *Modélisation et Elaboration d'Algorithms d'Estimation et de Commande? Application à un Bioprocédé*, PhD thesis, Université Paul Sabatier, LAAS/CNRS, Toulouse.

Michalewicz, Z. (1996). *Genetic Algorithms + data structures = evolution programs*, 3rd edn, Springer-Verlag.

Passino, K. and Yurkovich, S. (1998). *Fuzzy Control*, Addison - Wesley Pub. Co., USA.

Rumelhart, D. E., McClelland, J. L. and Group, P. (1986). *Parallel Distributed Processing*, Vol. 1, The MIT Press, Cambridge, Massachusets.

Zadeh, L. A. (1973). Outline of a new approach to the analisys of complex systems and decision processes, *IEEE Transaction on System, Man and Cybernetics* **3**: 28–44.

Zurada, J., Marks II, R. J. and Robinson, R. (1994). *Computational Intelligence: Imitating Life*, IEEE Press, Piscataway (NJ).