RULE BASE REDUCTION ON A SELF-LEARNING FUZZY CONTROLLER

Ferreiro García, Ramón **Pardo Martinez, Xoán C.** **Vidal Paz, José**
Electronics & Systems Department.
University of A Coruña
E.S.Marina Civil, Paseo de Ronda, 5
15011 A Coruña
e-mail: {ferreiro, pardo, vidal}@des.fi.udc.es

Abstract

In this paper a methodology to develop a fuzzy rule-based controller is described. The rule-base is obtained by learning from a virtual feedback (PID) controller. The designed controller is supposed to support typical control requirements (i.e. fast-following, zero steady state error, overshoot suppression, etc.) using only a small number of rules. The virtual controller is designed by manual adjusting with one of two alternative methods: simulation of a feedback controlled system using a estimated process model or by means of an on-line training phase.
Keywords: Fuzzy control, rule-based fuzzy controllers, self-learning fuzzy controllers.

1 INTRODUCTION

In fuzzy control developing techniques, it is of great importance how to obtain the control rules. A traditional method, widely used, to obtain the control rules consists of gathering the knowledge from experienced operators. Knowledge is supplied by an expert operator and usually it is incomplete. When the knowledge is obtained from different experts, contradictions among results are usually found. Fuzzy modelling is another method based on process structure identification and parameter estimation. But it requires a process model, which is in contradiction with applying fuzzy controllers when the process model is unknown. Self-organising fuzzy control, proposed first by Mamdani, has been applied successfully in many applications. Also neural networks have been used in self-organising fuzzy control systems, giving rise to a class of neuro-fuzzy controllers.

All of them shares the same drawback: rule evaluation and network training are, in most practical applications, excessively time-consuming. In this paper a method to generate automatically fuzzy control rules -that have to be updated on-line- is proposed. The aim is to develop a self-organising rule-based fuzzy controller solving the following problems:

- To generate the control rules using a virtual controller model as a trainer, that would supply its inherent process knowledge to the rule base. This controller would be tuned by manual supervision.
- To develop an inference engine based on a simple strategy in order to reduce drastically the rule base.

A virtual controller is defined as a controller that is used as a trainer or teacher for generating a consistent rule base. Any technique used for rule base reduction requires that the control function or the output of the virtual controller was described using additive sub-functions of the form

$$f_1(e) = u_1 = f(e)$$

$$f_2(e) = u_2 = f(\Sigma e)$$

$$f_3(e) = u_3 = f(\Delta e)$$

hence,

$$u = \sum_{i=1}^{n} f_i(e) \tag{1}$$

where u is the controller output, $f_i(e)$ is any sub-function of the controller actions, such as derivative or integral actions. Consequently, the selected virtual controller (fuzzy controller trainer) must satisfy the requirement of being formed by partial additive sub-functions. Controllers with this characteristic can be achieved using some of the on-line auto-tuning techniques that are among the most popular self-tuning methods.

2 VIRTUAL CONTROLLER DESIGN

The aim of training a fuzzy rule base is that a pattern algorithm was learned by the rule base. This pattern algorithm, that is used as a trainer of the rule base, is a virtual control algorithm which must satisfy dynamically some performance criteria in accordance with operating requirements. There exists several virtual control algorithms that could be used as training patterns. The most important due to their reliability are auto-tuning methods, and among them, two frequency domain methods which are not used in this work: (a) *relay feedback named method of harmonic balance* [1][2][7]; (b) *a method based on determining the phase and magnitude at operating frequency under phase and gain margins specifications* [4]. The method used in this paper to solve the problem of finding a virtual training controller is a manual supervised method: *manual adjusting procedure using classical knowledge of a PI(D) controller.*

3 INFERENCE PROCEDURE

When there are many variables in premises, the direct method of fuzzy reasoning has the following difficulties:

- The number of rules increases exponentially with the number of premise-part variables
- As the number of rules increases, the task of constructing rules becomes excessively burdensome.

Takagi, Sugeno and Kang proposed a method [5][6] to solve these problems using linear functions in the consequence part of the rules. This method has the following features:

- The consequence part of the rules uses linear input-output functions
- It is possible the identification of rules on the basis of input-output data modelling.

The implementation of the method requires a modelling approach based on input-output data. This data could be gathered from the virtual controller. The general reasoning model is given by rules of the form

IF x_1 is A_1 **and** x_n is A_n (2)
THEN $Y_1 = f(x_1,..,x_n) = C_0x_1 +..+ C_nx_n$

This method is extremely useful in cases where the number of variables is high. For instance, processes such as a position servo, a temperature controlled process, a power engine, a level controlled process and, in general, any process that could operate with several level set points under different loads. So that, the method is suitable when designing controllers that have to take into account the actual process operating point and load for a wide range of set points and loads. These requirements add two degrees of freedom to the designed virtual controller, which means that two more input variables are needed besides the classic error based variables (i.e. the error, the error derivative and/or the error integral). Theoretically, the number of rules needed to cover all possible input variations with a five term fuzzy controller would be

$$n_1 \times n_2 \times n_3 \times n_4 \times n_5 \qquad (3)$$

where n_1, n_2, n_3, n_4, and n_5 are the number of membership functions or linguistic labels of the five input variables. For instance, if $n_1 = n_2 = n_3 = n_4 = n_5 = 5$ then the number of rules would be 3125. In practice, the implementation of such a large rule base would require too much reasoning time besides a large amount of process memory.

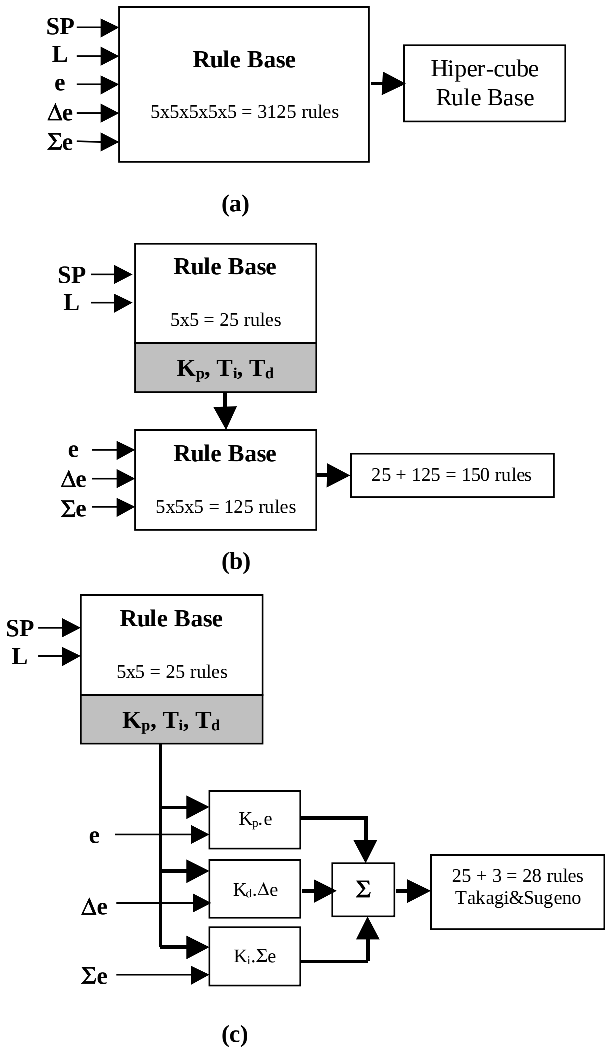

Figure 1: Some strategies in reducing Rule Bases: *(a)* Original hyper-cube rule base *(b)* Two rule bases in series *(c)* Reduction based on Takagi&Sugeno method.

If a system with five input variables is considered, where *SP* and *L* are the set point and load as shown in Figure 1(a), and *e*, *Σe* and *Δe* are a set of error dependent variables, the number of rules needed to implement a rule base is too high. This number could be drastically reduced using the additive procedure of association of rules. The necessary condition is that the system to be modelled must be linear. If we consider a PID algorithm, where the nominal load and the set point define a constant operating point (see Figure 1(b)), then a reduced rule base with only three rules is enough as shown in Figure 1(c). Table 1 shows the structure of the rule base used to get the set of PID parameters in adaptive mode by a common gain scheduling strategy.

Table 1: Structure of a two input/three output rule base

		Load		
		Large	***Medium***	***Small***
Set-Point	*Large*	Kp_1,Ti_1,Td_1	Kp_2,Ti_2,Td_2	Kp_3,Ti_3,Td_3
	Medium	Kp_4,Ti_4,Td_4	Kp_5,Ti_5,Td_5	Kp_6,Ti_6,Td_6
	Small	Kp_7,Ti_7,Td_7	Kp_8,Ti_8,Td_8	Kp_9,Ti_9,Td_9

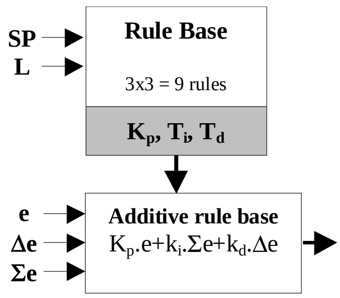

Figure 2: The structure of the reduced rule base.

In contrast with last results, the adaptive controller algorithm obtained using the classical approach of gain scheduling strategy will be quite similar:

$$\left.\begin{aligned} Kp &= fp(SP,L) \\ Ki &= fi(SP,L) \\ Kd &= f(SP,L) \end{aligned}\right\} \quad U=f(Kp,\, Ki,\, Kd,\, e\,,\, \Delta e,\, \Sigma e)$$

4 EXPERIMENTAL RESULTS

An automatic tool to design a controller of reduced fuzzy rule base using the Takagi&Sugeno method has been developed. This tool can handle on-line input/output data gathered from the process or off-line data gathered from an estimated model of the controlled process. This tool was developed using a visual engineering environment that is available for several commercial hardware platforms (i.e. PCI, VXI and VME bus architectures). The specific architecture used to test the method was a Data Translation I/O card on a PCI-based computer.

The proposed method was validated using the design tool on-line (gathering data directly from the process) and off-line (simulating the process). The selected process to be simulated was a second order open loop system (i.e. a double heater in cascade). A virtual PID controller is installed in the forward path. The PID parameters that supply the desired response are tuned manually by an operator. The fuzzy rule base is automatically generated using a number of membership functions that is also supplied by the operator. Figure 3 shows the excitation and response signals obtained during the manual tuning of the virtual PID controller. In this phase, PID parameters are adjusted to get the desired time response to an excitation of the controlled system. As it is shown in Figure 4, the fuzzy controller uses three rule bases in additive mode to supply a crisp output. The resultant rules are linear because the virtual controller used in the training phase was linear. In Figure 5 it is shown a typical response obtained by applying the designed fuzzy controller to control the process.

As conclusion, it has been showed that the fuzzy reasoning procedure adopted first by Takagi&Sugeno is a proper method to get a fuzzy rule base with reduced number of rules. The advantage of having this reduced rule base is that it could be easily modified and adjusted by a plant operator acting only in one or more premises (i.e. the proportional action, the derivative action, etc.) with total independence.

References

[1] Astrom, K. J. and Hägglund, T.; Automatic tuning of PID Regulators. Research Triangle Park, N.C.: Instrument Society of America, 1988.

[2] Astrom, K.J. and Wittenmark; On Self-Tuning Regulators. *Automatica*, 9, pp. 185-199, 1973.

[3] Brehm, T. Hybrid fuzzy logic PID controller, *Proceedings of the 3rd IEEE Conf. On Fuzzy Systems*, 3, pp 1682-1687, 194.

[4] Phillips, C.L. and Nagle, H.T.; *Digital control systems: Analysis and design.* Prentice Hall Inc., Englewood Cliffs, N.J. pp230-267, 1982.

[5] Sugeno, M. and Kang, G.T.; Structure identification of a fuzzy model. *Fuzzy Sets and Systems*, 28, pp 15-33, 1988.

[6] Takagi and Sugeno, Fuzzy Identification of Systems and its Applications to Modelling and Control. *IEEE Transactions on Systems, Man and Cybernetics*, 15, 1, pp 116-132, 1985.

[7] Tsypkin Y.Z., *Relay Control Systems*. Cambridge University Press, U.K., 1984.

[8] Tzafestas, S. and Papanikolopoulos, N.P., Incremental fuzzy expert PID control, *IEEE Trans. Ind. Electron*, 37, 365-371, 1990.

[9] Wang, F.Y. and Hung-man, K.; Implementation of adaptive fuzzy logic controllers with neural networks: a design paradigm. *Journal of Intelligent Fuzzy Systems*, 3, pp 165-180, 1995.

[10] Zhang, Z. and Mizumoto, M.; On rule self-generating for fuzzy control, *International Journal of Intelligent Systems*, 9, pp 1047-1057, 1994.

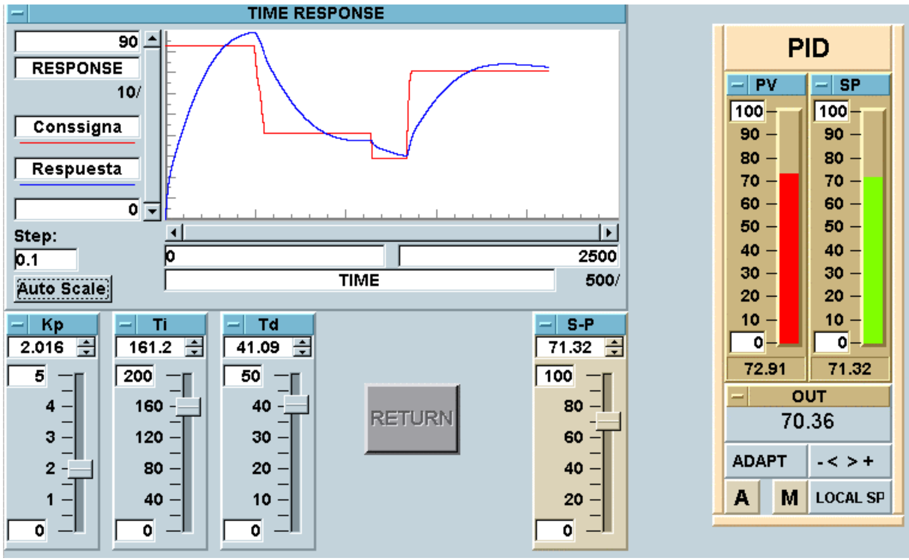

Figure 3: The virtual controller tuning phase.

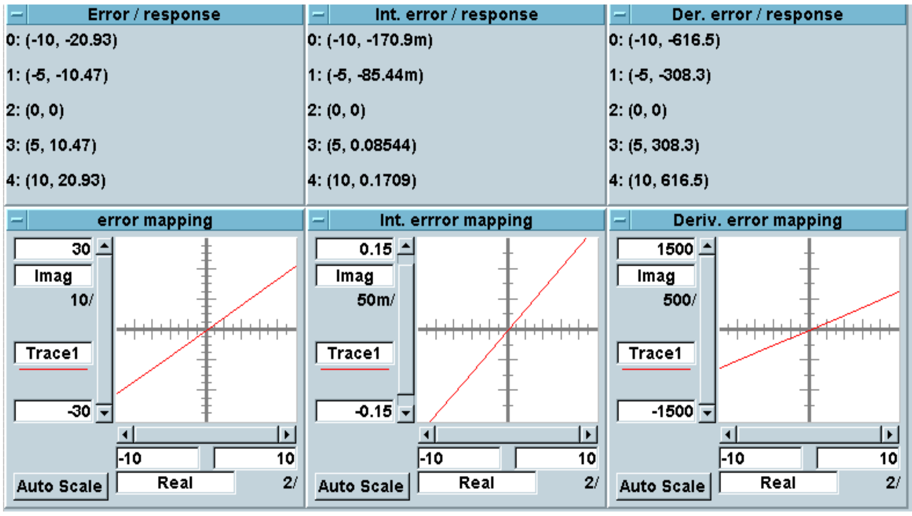

Figure 4: The reduced rule base.

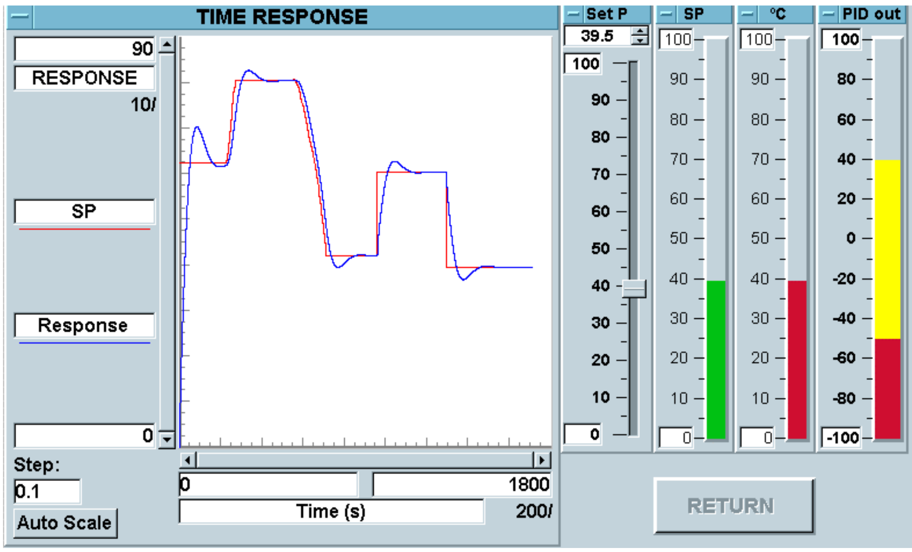

Figure 5: Fuzzy controller response.