IEEE International Workshop on Intelligent Robots and Systems
IROS '90

AN ARCHITECTURE FOR THE SUPERVISION OF FUZZY CONTROLLERS

P. J. Oliveira P. U. Lima J. J. Sentieiro *
R. S. Bravo R. Galan A. Jimenez †

* LRPI – Av. Rovisco Pais, 1096 Codex, Lisboa, Portugal
† DISAM – C. J. Gutierrez Abascal, 2, 28006 Madrid, España

Abstract

A new approach to the supervision of fuzzy controllers is presented. The supervision loop intends to overcome some problems still unsolved, such as time-varying plants, high non-linear plants or fine tuning of the linguistic terms given by the expert in a fuzzy controller. The supervision concept is based on the continuous adjustment of the mathematical functions used for the definition of the linguistic terms which describe the actions of the rules. The amount of adjustment is the result of a weighted combination of the results using two features observed in the control system output – the rise time and the overshoot.

1 Introduction

The fuzzy set theory has been successfully applied to automatic control, in the last few years [6].

Complex industrial plants such as cement kilns or chemical processes are the best suited for fuzzy control, due to the lack of easy to use and robust mathematical models.

In order to overcome the difficulty of designing a process controller from an high order, non-linear, time varying model (if there is one) fuzzy rule based methodologies have been proposed by several authors [2,3,4].

A fuzzy controller consists of a set of linguistic rules expressing the control policy of the process operator. The validity of this method is supported on the fact that the control of some processes by an human operator achieves better results than the controllers based on mathematical models.

However, some problems remain unsolved. Even though a fuzzy controller can achieve good performance in the control of non-linear systems, high non-linearities demand for some kind of adaptiveness of the controller, namely if the working point of the process is time varying.

In fuzzy controllers the fine tuning of control rules is important. Incorrect control actions are usually the result of ill-designed mathematical functions for the description of the linguistic terms used by the controller, or due to the existence of slightly incorrect premises for the control rules.

This paper describes a new approach to the supervision of fuzzy controllers. The supervision concept is based on the continuous adjustment of the mathematical functions used for the definition of the linguistic terms which describe the actions of the rules.

A brief overview of the most important results of the fuzzy set theory and of its application to control is presented, in order to introduce a nomenclature and concepts. In section 3 and 4, the proposed architecture for the fuzzy adaptive controller and its detailed implementation are described. Simulation results of the application of this controller to several systems are displayed and compared with non-supervised fuzzy controllers and PI controllers.

Finally, conclusions and future developments are discussed.

2 Fuzzy control theory and concepts

A fuzzy subset A of an universe of discourse (support set) U is characterized by a membership function $\mu_A(x) : x \in U \rightarrow [0,1]$, representing the grade of membership of x in A [7].

Each word or linguist term in a natural language can be viewed as a label for a fuzzy subset A of a universe of discourse U. This language assigns atomic and composite labels describing words, phrases and sentences to subsets of U [7].

A fuzzy linguistic variable is a variable whose values are linguistic terms used as labels of fuzzy sets. For instance, the fuzzy subset labels *high*, *medium*, *low* and *ok* can be regarded as values of the fuzzy variable *temperature*.

The three basic operations among fuzzy sets (complementation, union and intersection) are described in terms of the membership functions for the intervening sets. They are related to negation *not* of labels and to connectives between labels *or* and *and*, respectively:

- **complementation**: $\mu_{\neg A}(x) = 1 - \mu_A(x)$
- **union**: $\mu_{A \cup B}(x) = max[\mu_A(x), \mu_B(x)]$
- **intersection**: $\mu_{A \cap B}(x) = min[\mu_A(x), \mu_B(x)]$

A controller can be interpreted as a mapping from an input to an output. A fuzzy conditional statement describes this mapping in a way related to the knowledge representation by production rules.

For a controller with two inputs and one output, a typical fuzzy conditional statement (or fuzzy rule) is:

IF V_1 is T_1 and V_2 is T_2 THEN V_o is T_o

where

V_i , $i = 1, 2$, is the linguistic variable for the input i;

T_i , $i = 1, 2$, is one of the linguistic terms assumed by V_i;

V_o,T_o are respectively the linguistic variable and one of its possible linguistic terms for the output.

Given the fuzzy subsets $A \subset U$ and $B \subset V$, a fuzzy conditional statement R of the form: IF A THEN B, is defined by the bivariate membership function

$$\mu_R(x,y) = min[\mu_A(x), \mu_B(y)], x \in U, y \in V \qquad (1)$$

Now, if R is a fuzzy relation from U to V, and A' a fuzzy subset of U, the fuzzy subset B' of V inferred from A' given R has the membership function

$$\mu_{B'}(y) = \max_{x} \; min[\mu_{A'}(x), \mu_R(x,y)] \qquad (2)$$

as a result of the application of the **compositional rule of inference**(CRI) [7].

In this work, controller input values are crisp rather than fuzzy sets. Then, the CRI can be simplified by the interpretation of an input x_0 as a fuzzy input set A' with

$$\mu_{A'}(x) = \begin{cases} 0 & x \neq x_0 \\ 1 & x = x_0 \end{cases}$$

yielding

$$\mu_{B_r'}(y) = min[\mu_A(x_0), \mu_B(y)] = \mu_R(x_0, y) \qquad (3)$$

for the fuzzy rule r [2].

The fuzzy controller is composed by a set of rules with the form described before. Then, the final output fuzzy set, resulting from an input x_0 and an inference cycle over all the rules, is given by

$$\mu_{B'}(y) = \max_{r} \; min[\mu_A(x_0), \mu_B(y)] \qquad (4)$$

As with the input values, the output must be a crisp value. The **centroid method** has been chosen for this *defuzzification* operation:

$$y_0 = \frac{\int \mu_{B'}(y) y dy}{\int \mu_{B'}(y) dy} \qquad (5)$$

3 Architecture of a fuzzy controller and supervisor

The overall architecture of the fuzzy controller and supervisor is presented in figure 1. It is essentially a two loop hierarchical system where the basic loop is the control loop and the higher level loop is the supervision loop.

The control loop with the fuzzy controller is similar to those found in the literature [2,4]. Usual controller inputs are the *error* ($error(t) = reference(t) - output(t)$), the *change in error* ($\Delta error(t) = error(t) - error(t-1)$), the

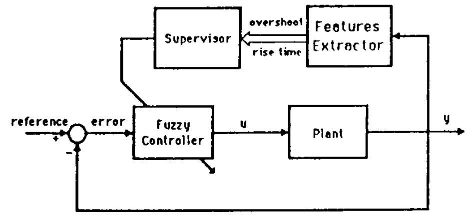

Figure 1: Architecture of the fuzzy controller and supervisor

integral of error or other process features.

The fuzzy controller is composed by four different elements.

- The *input fuzzy encoder* consisting of a set of analog membership functions, describing the *input linguistic terms*, and is designed by the human operator.
- The *linguistic control rules*, initially provided by an operator, in the form IF *premises* THEN *actuation*. Here, the *premises* are described by the input linguistic terms (one for each input variable) and the *actuation* by the *output linguistic terms*.
- A *defuzzifier*, which converts the output from the entire set of rules (determined by *max-min* fuzzy inference method) to a crisp control action.
- An *integral action* — the actuation is incremental, i. e., each sampling time the controller outputs the desired increment to the plant input.

No restrictions are made about the SISO plant to be controlled, except that the operator has some knowledge about its behaviour.

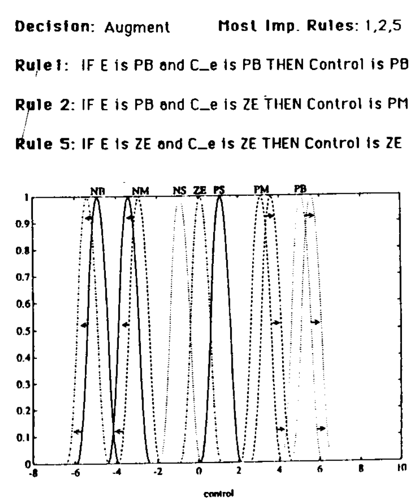

Figure 2: Example of actuation of the supervisor.

The supervision loop intends to overcome some problems still unsolved, associated with time-varying plants, high non-linear plants or fine tuning of the linguistic terms given by the expert in a fuzzy controller.

From the point of view of the fuzzy controller, given the system performance and a desired performance index, expressed by output features of the system, the supervisor performs a fine-tuning procedure. The chosen features are *overshoot* and *rise time*. Others could be used, related with *control activity* or *output offset*.

Due to the characteristics of the fuzzy inference procedure, a choice of only the most important rules fired is done. Those are the rules that the supervisor, based on the comparison between determined and desired features, will change.

Based on the result of the comparison, a decision to decrement or increment the central value of the output membership function is made (fig. 2). However, some restrictions must be considered.

- If a linguistic term is changed, its symmetrical linguistic term must also change.
- The linguistic term ZERO (ZE) must always be centered at 0.
- For consistency, the central values of POSITIVE_BIG (PB), POSITIVE_MEDIUM (PM) and POSITIVE_SMALL (PS) must satisfy:

$$PB_c \geq PM_c \geq PS_c \geq ZE_c$$

These restrictions intend to provide a "natural" change of the rules actuation preserving their linguistic meaning and close relation to operator actions.

4 Implementation

The implemented fuzzy controller has two inputs – the error and the change in error.

Three linguistic terms are defined for each of the input variables: POSITIVE_BIG (PB), ZERO (ZE) and NEGATIVE_BIG (NB) (figs. 3 and 4). Each linguistic term is described by the membership function $2^{-(x-a)^2}$, where a is its central value and x ranges in the universe of interest.

The initial control protocol, a set of nine rules, can be seen in table 1, where for instance the rule 1 can be read as:

IF
error is POSITIVE_BIG and
change in error is POSITIVE_BIG
THEN
control is POSITIVE_BIG

This protocol agrees with some constraints related with symmetry and stability [1]. The outputs of the table are the seven linguistic terms associated with the control (fig. 5).

The defuzzification is done by the simplified centroid formula:

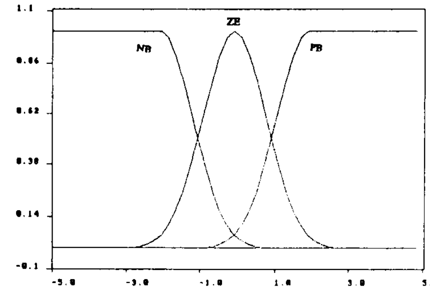

Figure 3: Input membership functions of variable *Error*.

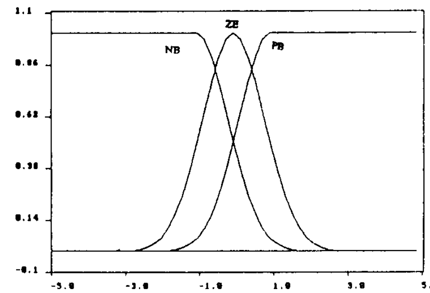

Figure 4: Input membership functions of variable *Change in error*.

	Change in error		
	PB	ZE	NB
Error PB	PB	PM	NS
ZE	PM	ZE	NM
NB	PS	NM	NB

Table 1: Rules protocol

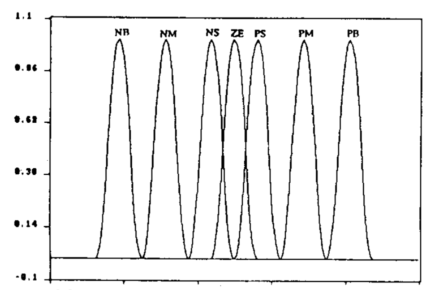

Figure 5: Output membership functions.

$$y_0 = \frac{\sum_{xx} control_{xx} \mu_{xx}(control_{xx})}{\sum_{xx} \mu_{xx}(control_{xx})} \qquad (6)$$

where $control_{xx}$ is the central value of the membership function xx.

Some results obtained with this fuzzy controller are presented in the next section.

The features chosen as supervisor inputs are:

- The overshoot.
- The rise-time percentage error (related to an estimated

delay of the system output).

The set of most important rules during the last considered period is also required.

Different options were considered in order to choose the supervisor actuation instants:

- Constant supervision sampling time (greater than system sampling time).
- Variable supervision sampling time, related to the reference input.
- Variable supervision sampling time, given by a delay after output stabilization.

The last option has been chosen, because it is the most closely related with the instants when new information about feature values is obtained.

In each of the actuation instants, the samples of the feature values and the most important rules during the last time interval are considered. Then, a decision of displacement for the central value of the output function is taken for every rule in the supervisor input set. The decision is based on the overshoot and rise-time percentage error values. The amount of displacement is given by those values and by an adaptation parameter – ϵ.

$$control_{xx} = control_{xx}(1 - \epsilon.Overshoot)$$

$$control_{xx} = control_{xx}(1 + \epsilon^2.Rise\_time)$$

The displacement decisions fulfil the restrictions presented in the last section. The value of ϵ is important, because there is a compromise between speed of convergence and oscillation in convergence. High ϵ's give rapid convergence with oscillations. Low ϵ's do not present oscillations but the convergence time increases.

5 Results

5.1 Methodology

Simulation of minimum and non-minimum phase 2nd. order discrete systems was done. The response to a sequence of positive and negative steps was tested for the two systems.

A PI controller was designed for the minimum-phase system with the goal of achieving the best rise-time without overshoot ($\xi = 1$). In one of the experiments an arbitrary change in the dynamics of the system has been made at the middle of the simulation, without changing the PI gains. The PI performance and the fuzzy controller with and without supervision were compared for the system, with changed and unchanged dynamics.

The equations describing the two systems are:

- Minimum phase system

$$\begin{aligned} y(t) &= 1.06y(t-1) - 0.22y(t-2) + \\ &\quad 1.99E-2u(t-1) + 1.99E-2u(t-2) \end{aligned}$$

- New minimum phase system, after change in dynamics

$$\begin{aligned} y(t) &= 0.9y(t-1) - 0.22y(t-2) + \\ &\quad 1.99E-2u(t-1) + 1.99E-2u(t-2) \end{aligned}$$

- Non-minimum phase system

$$\begin{aligned} y(t) &= 1.2y(t-1) - 0.35y(t-2) - \\ &\quad u(t-1) + 2u(t-2) \end{aligned}$$

and the PI was implemented by

$$u(t) = Kp(error(t) + Ki\sum_{k=t_0}^{t} error(k)) \; Ki = 0.5, Kp = 1$$

The same ϵ has been used for all the simulations.

5.2 Comments

Figure 6 a), b) and c) show the results obtained for the minimum phase system, respectively applying PI, fuzzy and adaptive fuzzy controllers.

In the latter, the membership function central values were initially set to incorrect values. This enhances the adaptation process of the output to the desired performance. It can be seen that approximately after 800 simulation steps, and 14 interventions of the supervisor, the adaptive fuzzy controller reaches the results for the PI controller and for the fuzzy controller without supervision. This one has carefully tuned central values for the membership functions.

It must be pointed out that, at the end of the simulation, a small overshoot can sometimes be observed. This is due to the oscillatory nature of the sequence (see 4). A smaller ϵ would solve this, at the cost of a slower convergence.

The results for the same system but with a change in the dynamics are presented in figure 7.

The PI (with fixed gains) can not cope with the change in the dynamics, and its action upon the modified system results in a slow response. In the other hand, the supervised fuzzy controller can adapt the response of the new system, with better results than its unsupervised version. The latter achieves similar results to those of the PI, in what concerns the sensibility to changes in the dynamics.

For the non-minimum phase system, results for the PI and fuzzy controller with and without supervision are presented in figure 8 a), b) and c).

The initial central values for the membership functions in these controllers were displaced from its correct values, providing generally higher actuations in response to high error and change in error. This explains the bad performance of the unsupervised version. Due to the non-minimum phase characteristics of the controlled system, the initial step response error and change in error are big, resulting in an initial large actuation with overshoot.

The supervised controller, however, achieves good results after only a few simulation steps, which corresponds to fewer supervision cycles. Its action tends to reduce the central values of the membership functions, as it was expected.

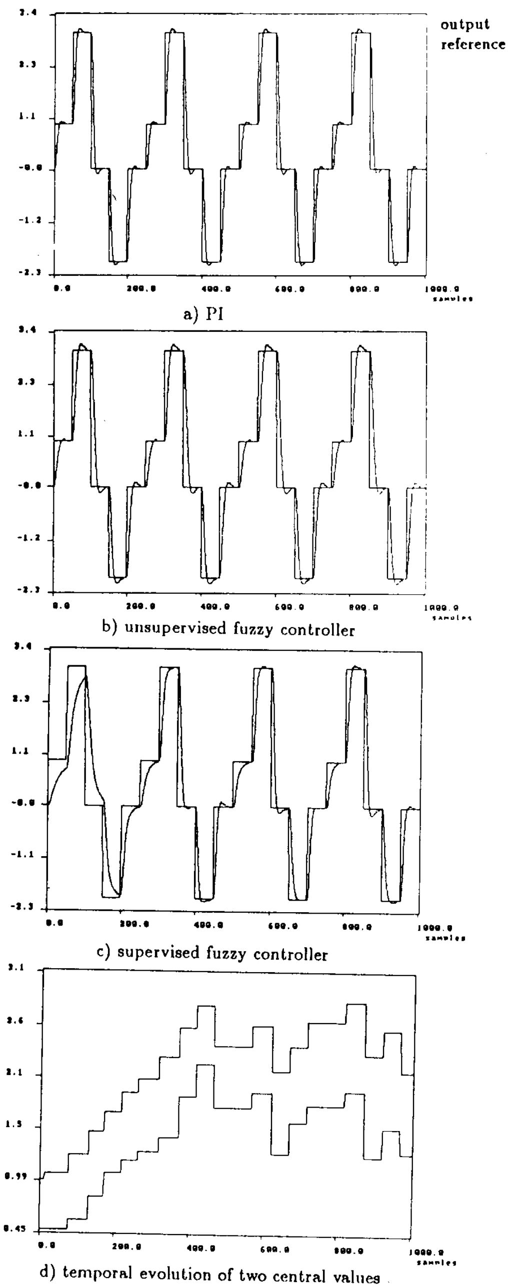

Figure 6: Minimum phase system results

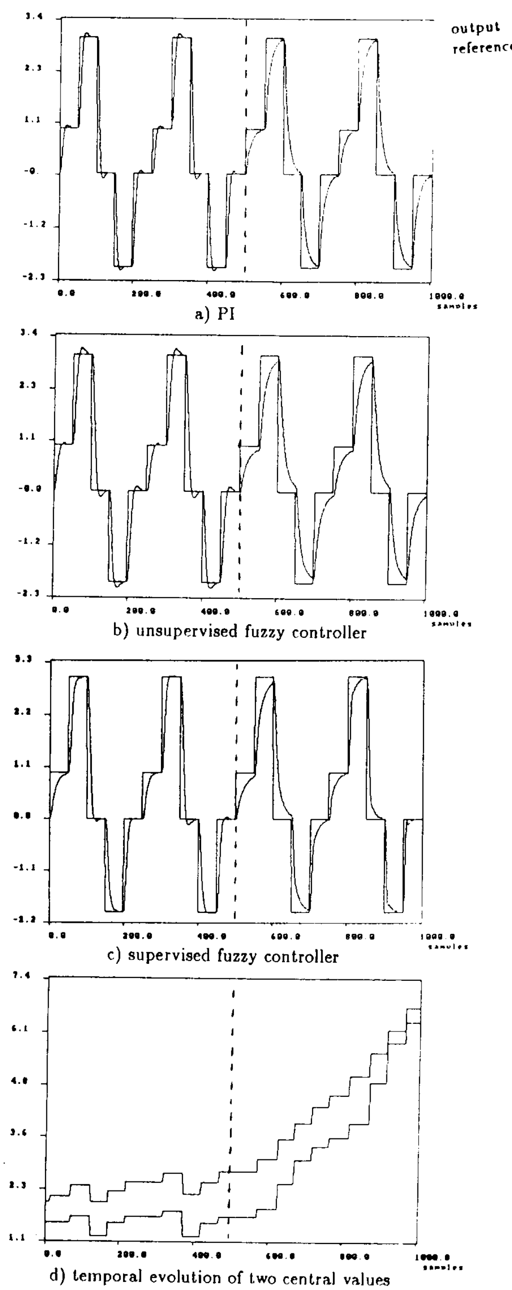

Figure 7: Results of minimum phase system, with change in dynamics

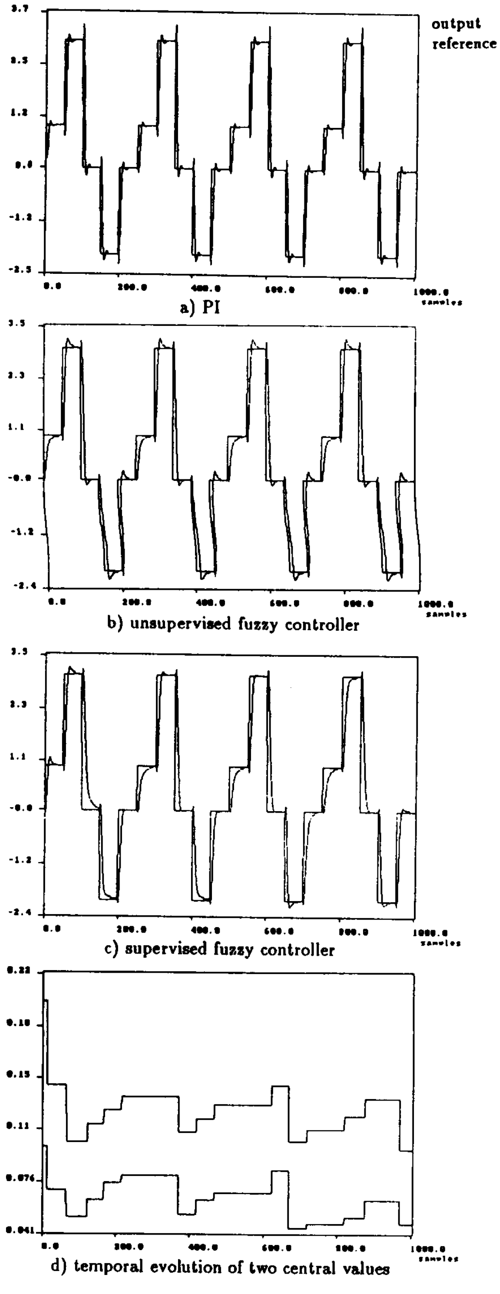

Figure 8: Non-minimum phase system results

6 Conclusions and future trends

A new strategy for supervision of fuzzy controllers was presented. It consists of a continuous adjustment of central values for the membership functions describing the actuation linguistic terms for the most intervening rules in each supervision cycle. This differs from other approaches to fuzzy supervision [1,5] in the sense that the linguistic meaning of the rules and its relation to initial operator actions are preserved. The amount of adjustment is the result of a weighted combination of the results using two features observed in the control system output – the rise time and the overshoot.

The results of this strategy were compared with those obtained from the application of a PI controller and an unsupervised fuzzy controller. Equal or better results were achieved in each case, with the adaptation reached in only a few supervision steps.

At the present, every intervening rules classified among the most important ones is adjusted by the same amount. In the future, displacements weighted by the relative importance of the rules will be implemented. Also, an heuristic rather than algorithmic approach to the superviser will be considered.

References

[1] Cerezo A. J. (1987). "Aplicaciones del Razonamiento Aproximado en el Control y Supervision de Procesos", *Phd Thesis*, ETSII de Vigo.

[2] Kickert W. J.,.Van Nauta Lemke H. R. (1976). "Application of a Fuzzy Controller in a Warm Water Plant", *Automatica*, vol. 12, pp. 301-308.

[3] King R. E., Karonis F. C. (1988). "Multi-Level Expert Control of a Large-Scale Industrial Process", *Fuzzy Computing*, pp. 323–339.

[4] King P. J., Mandani E. H. (1977) . "The Application of Fuzzy Control Systems to Industrial Processes", *Automatica*, vol. 13, pp. 235-242.

[5] Procyk J. J., Mandani E. H. (1979). "A Linguistic Self-Organizing Process Controller", *Automatica*, vol. 15, pp. 15–30.

[6] Sugeno M. (1985). "An Introductory Survey of Fuzzy Control", *Information Sciences*, vol. 36, pp. 59–83.

[7] Zadeh L. A. (1973). "Outline of a New Approach to the Analysis of Complex Systems and Decision Processes", *IEEE Trans. on Sys. Man and Cyber.*, vol. SMC-3, No. 1, January.