Control and Cybernetics

vol. **27** (1998) No. 4

Learning of rule importance for fuzzy controllers to deal with inconsistent rules and for rule elimination

by

Kuhu Pal (nee Dutta) and Nikhil R. Pal*

Machine Intelligence Unit,
Indian Statistical Institute,
203, B. T. Road,
India
email: nikhil@isical.ac.in
* author for correspondence

Abstract: Extraction of correct and precise rules from experts is a difficult problem. Moreover, even when the extracted rules are correct, all of them may not have equal importance to achieve the goal of the fuzzy system. Rule tuning is usually achieved through modification of membership functions. Effect of changing a membership function is global in the sense, it influences all rules that involve the membership function. Here we propose an effective extension of the ordinary fuzzy controller model which incorporates an importance factor for each rule. The importance factor allows tuning of the system at the rule level. Of course, one can still tune the membership functions. The extended model enables us to cope with incorrect and/or incompatible rules and thereby enhances the robustness, flexibility and system modeling capability. It also helps us to eliminate redundant rules easily. For the Takagi-Sugeno framework, we derive the learning algorithm for the rule importance factor as well as that for the consequent. We demonstrate the superiority of the extended model through extensive simulation results using the inverted pendulum.

Keywords: rule importance, rule selection, fuzzy logic controllers, rule tuning

1. Introduction

The essential part of a fuzzy logic controller (FLC) is a set of linguistic control rules equipped with some fuzzy implication operator and a rule of inferencing Harris, Moore and Brown (1993), Lee (1990), Yamakawa (1992), Driankov, Hellendoorn and Reinfrank (1993). Literature suggests that FLCs sometimes perform better than the conventional control algorithms. In particular FLC appears more attractive when the processes is too complex to analyze by conventional quantitative technique or when the available sources of information are qualitative, inexact or imprecise. Since Mamdani and Assilian (1975) FLC has been successfully used in many applications Sugeno (1985), Yasunobu and Miyamoto (1985), Yagishita, Itho and Sugeno (1985), Procyk and Mamdani (1979), Shao (1988), Park, Moon and Lee (1995), Nomura, Hayashi and Wakami (1991), Jang (1992), Guely and Siarry (1993).

Two major factors which may restrict the application domain of FLCs are sound technique of knowledge acquisition and the availability of human experts. An operator can easily control a system, but may fail to express properly the rules (s)he uses for decision making. So there is a great need of learning, and tuning the control rules and associated parameters to achieve a desired level of controller performance.

Given a rule-set, tuning of any fuzzy set A will influence all rules that involve that particular fuzzy set A. Thus such tuning schemes have global impact on the rule-base. The designer has no tool to tune only a particular rule. Further, each rule may not have equal importance to control the system also. We extend the conventional fuzzy models to equip the designer with a steering to realize a more flexible system by adjusting each rule separately. The extended model associates an importance factor to each rule . The importance factor enhances the robustness, flexibility and modeling capability of the system. Initially we assume the importance factors to be unrestricted in sign. We derive the learning algorithm for the importance factor and establish its power for handling inconsistent rules using an inverted pendulum. Then we show how we can make importance factors non-negative and use them for rule selection.

The organization of this paper is as follows. Section 2 provides a brief review of some existing tuning schemes. Section 3 introduces the extended model, its motivation and merits. In Section 4 we present the tuning algorithms for the extended model. The proposed tuning scheme is used to control an inverted pendulum. Results are reported in Section 5. Section 6 discusses how non-negative importance factors can be realized and applied to rule elimination. Finally, the paper is concluded in Section 7.

2. Some existing tuning schemes

A fuzzy controller is defined in terms of if-then rules. In this investigation we use the Takagi-Sugeno (TS) model. Suppose the control system has n input variables $(x_1, \ldots, x_n)$ and one output variable u. The r-th rule, $R(r)$, for the TS type of controller takes the form

$R(r)$ If (x_1 is X_{r1}) and ... (x_n is X_{rn}) then $u = u_r = F_r(x_1, \ldots, x_n)$; $r = 1, 2, \ldots, k$;

where X_{rj}'s are fuzzy sets defined on x_r and u_r's are crisp values provided by the function F_r.

The TS model works as follows. For the given input values of the process state variables $x_1, x_2, \ldots, x_n$ the membership value $\mu(x_i)$ is calculated for each $i = 1, \ldots, n$. Each $\mu(x_i)$ gives the extent to which the corresponding fuzzy set is satisfied. The minimum of all $\mu(x_i)$'s or the product of all $\mu(x_i)$'s is usually taken as the firing strength. However, any T-norm can also be used to compute the firing strength. We denote the firing strength of the r-th rule by f_r. The firing strength modulates the output (consequent) function. A well known method of defuzzification (conflict resolution) is to find the weighted normalized sum of all pairs (f_r, u_r) which is given by

$$y'_i = \frac{\sum_{r=1}^{k} f_r * u_r}{\sum_{r=1}^{k} f_r}. \quad (1)$$

This crisp output will be the plant input in next phase.

Literature contains many methods for tuning of fuzzy controllers. We briefly discuss a few of them here.

Nomura, Hayashi and Wakami (1991) used the gradient descent method to tune rule-base parameters of TS rules with constant outputs and symmetric triangular membership functions. They used product as the conjunction (and) operator to find the firing strength of a rule. This method simultaneously modifies the crisp consequent value and, the center and width of the triangular input fuzzy sets. The tuning process continues until the change in the objective function between two successive iterations becomes suitably small.

Gradient descent method has also been used by Jang (1992) with TS rules having affine output function, assuming it to be potentially more efficient than the constant output function. Guely and Siarry (1993) empirically show that affine output functions were not more efficient than constant output functions. Guely and Siarry (1993) considered tuning of constant and affine output functions, and symmetric and asymmetric membership functions with minimum and multiplication as conjunction operators. Guely and Siarry used a modified form of TS rule, called the "centered Takagi Sugeno Rule" and showed on an example that it can achieve a much better learning accuracy in the same case. Berenji and Khedkar (1992), Berenji (1992) used softmin as their conjunction operator and used a reinforcement type tuning algorithm.

Lui, Gu, Goh and Wang (1994) proposed a self-tuning adaptive resolution (STAR) fuzzy control algorithm. STAR changes constantly the fuzzy linguistic concepts in response to states of the input signals. STAR is a heuristic algorithm that attempts to minimize both rise time and overshoot. Isomursu and Rauma (no date) also used meta rules to modify the scaling factor of one output variable and membership functions for a temperature controller. The meta rules use a performance measure based on oscillation amplitude and frequency.

Maeda and Murakami (1992) proposed fuzzy rule-based schemes for adjustment of input-output scaling factors as well as for tuning of control rules for Takagi - Sugeno (TS) model. The fuzzy rule-base for tuning has three sets of rules based on three different performance measures, overshoot, rise time and amplitude. After tuning of scaling factors the crisp consequent parts of the control rules are modified in each sampling time considering a fuzzy performance index and the deviation of the actual control response from a predefined target response.

Karr and Gentry (1993) have used genetic algorithms for tuning of membership functions of a FLC for pH control of a system where the process dynamics change in different ways. Homaifar and McCormick (1995) used GA for simultaneous determination of membership functions and the rule set.

The gain tuning method of Yoshida, Tsutsumi and Ishida (1990) assumes all processes as first order systems with dead time. The input and output scaling factors are calculated by some empirical relations involving process parameters. Good control performances for higher order systems cannot be ensured by this technique. Auto-tuning fuzzy controller of Hayashi (1991) considers two tuning functions. From the approximate parameters of the identified plant model (first-order lag with dead time) the input and output scaling factors are calculated using the concept of Chien-Hrones-Reswick (CHR) tuning rules for a conventional PI controller. Then the crisp consequent parts are modified using the overshoot value and rise time as performance measures. Linear first-order plant models with dead time have also been considered in the auto-tuning scheme of Iwasaki and Morita (1990). Here the parameters of the plant model are identified through fuzzy inference, using differences between the actual plant features (rise time and overshoot) and the plant model features. This procedure is repeated until the feature differences are smaller than some specified thresholds.

Palm (1995) proposed to achieve an optimal adjustment in the input scaling factor with the help of input-output cross-correlation function; though he assigned a higher priority to the tuning of output scaling factor over that of input scaling factors. Here the input data are assumed to follow a Gaussian distribution whose parameters are unknown. An optimal input scaling factor is obtained by maximizing the cross-correlation function which is a measure of the statistical dependence between input and output.

3. An extended model with rule importance

Deciding on the rules to be used for a fuzzy system is a difficult task. The problem becomes more difficult, when experts are not available. Even when experts are available, it is often difficult to extract the correct rules from them. Existence of just a single incorrect or inconsistent rule may degrade the performance of the system significantly. For the sake of arguments let us assume that experts provided rules are correct and consistent. Normally in a conventional fuzzy logic controller all rules are given equal importance. But, for all systems this may not desirable. Different rules may have different level of influence on the system behavior. Moreover, in a conventional FLC, tuning of a fuzzy set, say A, influences all rules that involve A. Thus, alteration of a fuzzy set has a global impact on the rule-base. The designer usually cannot adjust only a single rule in isolation. To overcome these problems and to incorporate relative importance of rules in the fuzzy model, we propose the following extension of the FLC.

Suppose the control system has n input variables $(x_1, x_2, \ldots, x_n)$ and one crisp output variable u. The extended model comprises rules of the form:
$R(1)$ If (x_1 is X_{11}) and ... (x_n is X_{1n}) then $u = U_1$ with importance α_1
...
...
...
$R(k)$ If (x_1 is X_{k1}) and ... (x_n is X_{kn}) then $u = U_k$ with importance α_k.

Here x_i and u are linguistic variables and X_{ri}'s and U_r's are fuzzy sets defined on the respective domains. For a given $x = (x_1, x_2, \ldots, x_n)$, let the firing strength of the i-th rule be f_r; $r = 1, 2, \ldots, k$. Note that some of the f_r may be equal to zero. Then the defuzzified output can be computed as

$$u^* = \frac{\sum_{r=1}^{k} f_r * \alpha_r * a_r^u}{\sum_{r=1}^{k} f_r * \alpha_r}. \tag{2}$$

where a_r^u is the peak of the fuzzy set U_r. Equation (2) is a modification of the height method Driankov, Hellendoorn and Reinfrank (1993) of defuzzification. Similarly, output can also be computed using extension of any other defuzzification scheme.

Now we consider the extension for the TS type of controllers. A typical rule under the extended TS model takes the form

If the temperature (t) is high and the pressure (p) is medium then the flow of gasoline is $u = f(p, t)$ with importance α

We call such model as "Extended Fuzzy Logic Controller under the T-S model"; in short the extended TS (ETS) Model. In general, under ETS Model the r-th $(r = 1, 2, \ldots, k)$ rule takes the form
$R(r)$ If (x_1 is X_{r1}) and ... (x_n is X_{rn}) then $u = u_r = F_r(x_1, \ldots, x_n)$ with importance α_r.

Now for a given input $\mathbf{x} \in \Re^p$, the defuzzified value is computed by

$$y_i' = \frac{\sum_{r=1}^{k} f_r * u_r * \alpha_r}{\sum_{r=1}^{k} f_r * \alpha_r} \tag{3}$$

where f_r is the firing strength of the r^{th} rule and k is the total number of rules that are fired.

We shall consider two versions of the model:
(i) α_r's are unrestricted, i.e. it can have positive, negative and zero values.
(ii) α_r's are all non-negative, i.e., $\alpha_r \geq 0 \ \forall r$.

In this sequel, we illustrate that both models have distinct advantages in different situations.

Due to the presence of α_r, equation (2) or (3) can model much more complex control surfaces than that by equation (1), resulting in more flexibility of the fuzzy system. One may get a false impression that the same flexibility can be obtained from equation (1) by tuning the parameters of consequent functions. This is not true. A close inspection of equation (1) and equation (3) reveals that when $\alpha_r = c \; \forall \; r$ ($c =$ a constant) equation (3) = equation (1), and under this situation, tuning of u_r changes only the numerator of (1) and (3), the denominator remains unaffected. In all other cases α_r alters both numerator and denominator of (3), and influences $y_i{}'$ in a nonlinear manner. Analogous arguments can also be given with respect to equation (2).

Given a training data set (X, Y), $X = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N\} \subseteq \Re^p$ and $Y = \{y_1, y_2, \ldots, y_N\} \subseteq \Re$, we can obtain suitable values for α_r minimizing $\sum_{i=1}^{N} (y_i - y_i{}')^2$. Note that as α_r is unrestricted in sign, theoretically the denominator $\sum_{r=1}^{k} f_r * \alpha_r$ of equation (3) could be very small, even zero. But the training process will never allow this. Because, if $\sum_{r=1}^{k} f_r * \alpha_r \to 0$ then $y_i{}' \to \infty$, but y_i is finite. This will result in infinite error. Thus when α_r's are learnt using training data with finite output values, $\sum_{r=1}^{k} f_r * \alpha_r$ will never be zero. In other words, if we start with $\alpha_r = 1 \; \forall r$, then the gradient based tuning algorithm (or any consistent tuning algorithm) will never change α in such a manner that $\sum_{r=1}^{k} f_r * \alpha_r$ goes to zero. And if the input-output relation is smooth and the training data that are used to learn the α_r, adequately represent the relation so that the actual input-output relation is captured by the identified fuzzy system, it is also not expected to occur for any test data.

In the present investigation, we consider the most simple form of the TS model, where the consequent of each rule is a crisp value. In other words, the r-th rule has the form

$R(r)$ If (x_1 is X_{r1}) and ... (x_n is X_{rn}) then $u = u_r = F_r(x_1, \ldots, x_n)$ with importance α_r

But how do we get α_r, $r = 1, 2, \ldots, k$. If an expert is available, α_r can be obtained from him/her. This may not always be possible. Moreover, it is better to learn α_r from a set of reliable input-output data.

First we consider the case with unrestricted α_r, and then in Section 6, we will concentrate on the case with non-negative α_r.

4. Learning of importance factors

Let (X, Y), $X = \{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$, and $Y = \{y_1, \ldots, y_N\}$, be a set of input-output data. Here y_i is the control action corresponding to state vector $\mathbf{x}_i$. In order to learn the importance factor of each rule, like Nomura, Hayashi and Wakami (1991), we minimize the squared error function E given by

$$E = \sum_{i=1}^{N} \frac{(y_i - y_i')^2}{2}.$$

This nonlinear optimization problem is solved by the steepest descent method. This is an iterative algorithm that reduces the value of the objective function with each iteration. Like online training in neural networks we minimize E by moving a small amount in the negative direction of the instantaneous error function e_i. The instantaneous error function for $(\mathbf{x}_i, y_i)$ is given by

$$e_i = \frac{(y_i - y_i')^2}{2}, \quad i = 1, 2, \ldots, N. \tag{4}$$

We can minimize e_i with respect to one or more parameters of the rule-base, like peak and base of the membership function, consequent parameters and rule importance factor. Here we derive update equations for only the relative importance α_r and the consequent value u_r for each rule.

Suppose for the input-output data point $(\mathbf{x}_i, y_i)$ the control output is (recalling equation 3)

$$y_i' = \frac{\sum_{r=1}^{k} f_r * u_r * \alpha_r}{\sum_{r=1}^{k} f_r * \alpha_r}. \tag{5}$$

Let μ_{rj} be the membership function of a fuzzy set defined on the j-th antecedent variable of the r-th rule. Different types of membership functions can be used. For simplicity, we use symmetric triangular membership function μ_{rj} defined as

$$\mu_{rj}(x) = 1 - \frac{2 \mid x - a_{rj} \mid}{b_{rj}} \tag{6}$$

where a_{rj} is the peak (i.e., the membership value is 1 at $x = a_{rj}$) and b_{rj} is the base or support of μ_{rj}. Given (6) and (5), the update equations for α_r and u_r to minimize (4) can be obtained by gradient descent as

$$\alpha_r(t+1) = \alpha_r(t) - \eta_\alpha(t) * \frac{\partial e_i}{\partial \alpha_r} \tag{7}$$

and

$$u_r(t+1) = u_r(t) - \eta_u(t) * \frac{\partial e_i}{\partial u_r}. \tag{8}$$

η_α and η_u are the learning co-efficients respectively for the importance factor and consequent value of the rules.

With some algebraic manipulation, we get

$$\frac{\partial e_i}{\partial \alpha_r} = \frac{(y_i' - y_i) * (u_r - y_i') * f_r}{\sum_{j=1}^{k} f_j * \alpha_j} \tag{9}$$

and

$$\frac{\partial e_i}{\partial u_r} = \frac{(y_i' - y_i) * f_r * \alpha_r}{\sum_{j=1}^{k} f_j * \alpha_j}. \tag{10}$$

If we start with $\alpha_r = 1\ \forall r$, then $\sum f_r * \alpha_r \neq 0$ for all data points to start with. The question is, can the learning process take $\sum f_r * \alpha_r$ to zero. Intuitively it will not, because then y_i' will be infinitely large (positive or negative) resulting in an infinite error, but gradient descent attempts to reduce the error. However, one can argue that due to bad choice of step-length (learning co-efficient) it could happen theoretically at some stage of training, then the remedy is to alter the value of α_r of one of the fired rules by some amount ϵ. Note that, it does not matter whether ϵ is positive or negative or large or small. The idea is to get out of the degenerate case. In practice, the probability of getting exact zero for the denominator is practically zero, but if it becomes very small, as explained earlier due to bad choice of step-length, the successive steps of gradient descent will modify the value of α in a direction so that $\sum f_r * \alpha_r$ moves away from zero in order to reduce the error.

The algorithm for tuning α_r proceeds as follows: For each pair of $(\mathbf{x}_i, y_i)$, update α_r using (7) and (9). Repeat the process until $\| \alpha(t) - \alpha(t+1) \| /k < \epsilon$, where ϵ is a small positive quantity and $\alpha(t)$ is the vector of weights after t epochs. A complete pass through the data is called an epoch. Tuning of u_r can be done in a similar manner using (8) and (10). We next provide a schematic description of the tuning algorithm.

Algorithm for tuning of rule importance

Algorithm Importance-Tune $(X, Y, \alpha, \eta_\alpha, Tmax_\alpha, \epsilon)$

Input : $X, Y, \alpha, \eta_\alpha, Tmax_\alpha, \epsilon$

Here $X = (\mathbf{x}_1, \ldots, \mathbf{x}_N)$ is the set of input vectors and $Y = (y_1, \ldots, y_N)$ is the set of corresponding output vectors. The constant ϵ and $Tmax_\alpha$ are used for termination of the algorithm, η_α is the learning co-efficient for all α_r, $\alpha = (\alpha_1, \alpha_2, \ldots, \alpha_k)^T$.

Begin Algorithm

1. $\eta_\alpha^1 = \eta_\alpha$, $\alpha(0) = \alpha$.
2. For $t = 1$ to $Tmax_\alpha$ do
   (a) Repeat for each $\mathbf{x}_i \in X$
      i. Compute f_r for all $r = 1, \ldots, k$
      ii. Compute $y_i\prime$ using equation (5)
      iii. Do for each rule r which is fired
         A. modify α_r using Eqns. (7) and (9).
         B. Recompute y_i' using equation (5)
      iv. End Do
   (b) End Repeat
   (c) Calculate $E_t = \| \alpha(t) - \alpha(t+1) \| /k$
   (d) If $E_t < \epsilon$ then stop
   (e) Adjust η_α^t

3. End For

End Algorithm

The algorithm for tuning of u_r can be written in the same manner. Note that such an algorithm imposes no restriction on the sign of α_r and hence α_r could be negative also. This may appear to be counter-intuitive as importance factor should preferably be non-negative. Later we shall see, that it is indeed a powerful feature and can help us to cope with inconsistent rules. However, we shall also show, how we can impose the non-negative constraint on the importance factor.

5. Implementation and results

To illustrate the effectiveness of the proposed extended model, we use the inverted pendulum problem as an example, because its physical model is well known and fairly simple. The inverted pendulum is a system, in which a rod of mass m is hinged on a cart (Fig. 1). Although the pendulum can fall in any direction, we restrict ourselves to the 2-dimensional version where the pendulum (rod) can move only in a vertical plane (i.e., in the plane of the page). We assume that the mass of the pendulum is concentrated at the end of the rod and the rod is massless. The control force u is applied to the cart to keep the pendulum in an equilibrium position. Let θ be the angle of the rod from the vertical line. The slanted pendulum can be brought back to the vertical position when a suitable control force is applied to the cart. If the cart is at rest, the stick is in the vertical position and the force u is zero then the system is in equilibrium. This equilibrium position is unstable in the sense that with any perturbation from this position, no matter how small, the stick will fall down. For this system we have two types of fuzzy control rules; one for controlling the rod and the other for the cart. In our investigation, for the sake of simplicity, we have considered only the pole balancing part.

The range of each input linguistic variable (θ and $\dot{\theta}$) are divided into 7 overlapped intervals. We decided to choose the initial rule set in such a manner that for every possible input at least two rules are fired. If only one rule fires, then the fuzziness in the output will be lost and flexibility of the system will be reduced. The antecedent clauses of the rule set used are shown in the Table I with cross marks. The linguistic values for each linguistic variables (θ and $\dot{\theta}$) are: NB = Negative Big, NM = Negative Medium, NS = Negative Small, Z = Zero, PS = Positive Small, PM = Positive Medium, PB = Positive Big. The consequent of the rules are not shown in Table 1. We start with some arbitrary value of force for each rule. A good choice of the rule set and consequent values gives rise to faster convergence and better performance. The only reason for the selection of this particular rule set is the uniformity of rules over the entire rule space, except near the equilibrium position where we have more dense rules. There could be other choices too. In addition to this set of 29 rules, we also experimented with 49 rules.

We use the following computational protocols: half rod length 0.5 m; pole

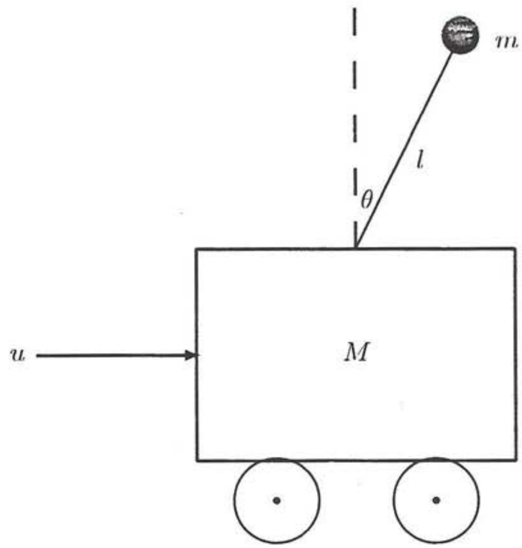

Figure 1. The inverted pendulum

θ \ $\dot{\theta}$	NB	NM	NS	Z	PS	PM	PB
NB	×		×		×		×
NM		×		×		×	
NS	×		×	×	×		×
Z		×	×	×	×	×	
PS	×		×	×	×		×
PM		×		×		×	
PB	×		×		×		×

Table 1. Rule set for the inverted pendulum

$\dot{\theta}$ / θ	NB	NM	NS	Z	PS	PM	PB
NB	-10.13		1.75		3.70		8.89
NM		-23.44		5.48		-1.38	
NS	25.28		13.57	5.22	7.21		1.27
Z		-1.26	15.96	5.73	7.67	3.35	
PS	4.03		12.26	8.55	4.64		1.32
PM		30.05		-0.38		6.10	
PB	75.48		6.9		2.65		.00

Table 2. Values of α for different rules after 3,000 epochs, corresponding to Figs. 2(b) & 3(b).

mass 0.1 kg; cart mass 2 kg. The initial value of each α_r is taken as 1. However we also experimented with initial α_r chosen randomly in $[0, 1]$. With random initialization, tuning is usually more cumbersome and $\alpha_r = 1$ is found to be a better choice for initialization. For all results reported, we started with $\alpha_r = 1$ $\forall r$, i.e., each rule is given equal importance at the beginning. The firing strength has been computed using product as the conjunction operator. The learning rate η_α for α_r was changed dynamically. We started with a high value of η_α. η_α is kept the same, as long as the total square error E reduces with iteration. Whenever E is found to increase after a pass through the training data, η_α is reduced by 10%.

We have done many simulation experiments and report here only a few typical of them. For our first simulation, we use a bad rule set with arbitrarily chosen values of force and parameters of the membership functions. The rule set was not able to bring the system to the equilibrium position. The ETS controller has been tuned using Importance-Tune algorithm for 3000 epochs. The values of the importance factors (α_r) after tuning are shown in Table 2. In some cases the values of weights are positive and for a few cases they are negative. Fig. 2 depicts the system response in terms of θ before and after weight-tuning. After tuning of importance factors, the performance of the controller, in terms of $\dot{\theta}$, is shown in Fig. 3. After tuning, we can see a significant improvement in the performance of the controller. It is clear from Figs. 2 and 3 that only an appropriate choice of importance factors of rules can make an uncontrollable system controllable.

In order to establish the effectiveness of the proposed model we report results from another bad rule set (We used different values of force and parameters of membership functions than those used in Table 2). Initially the system was not controllable, but after tuning of α only for 6000 epochs, it becomes controllable.

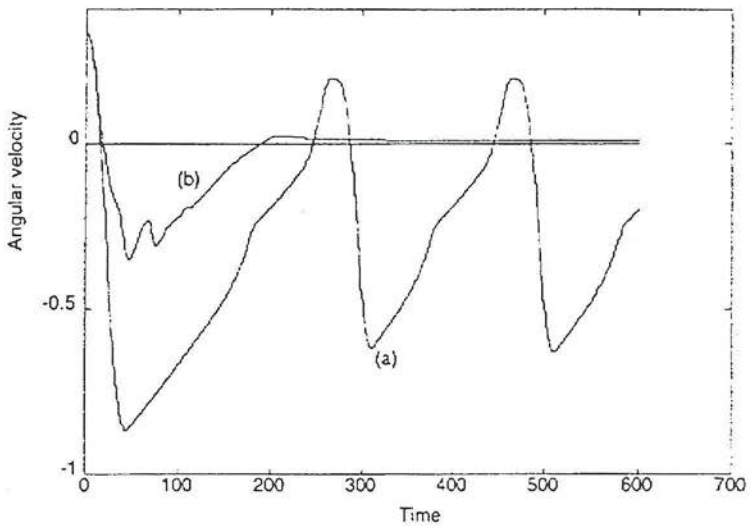

Figure 2. System response in terms of θ: (a) Before tuning of α, (b) After tuning of α for 3000 epochs

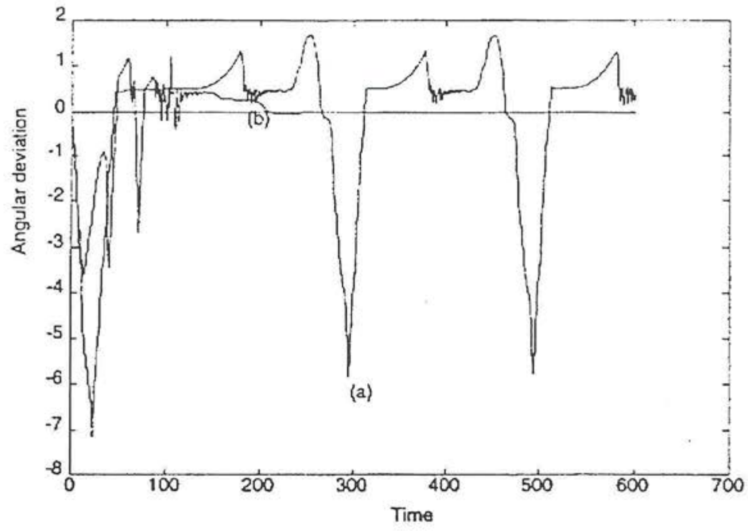

Figure 3. System response in terms of $\dot{\theta}$: (a) Before tuning of α, (b) After tuning of α for 3000 epochs

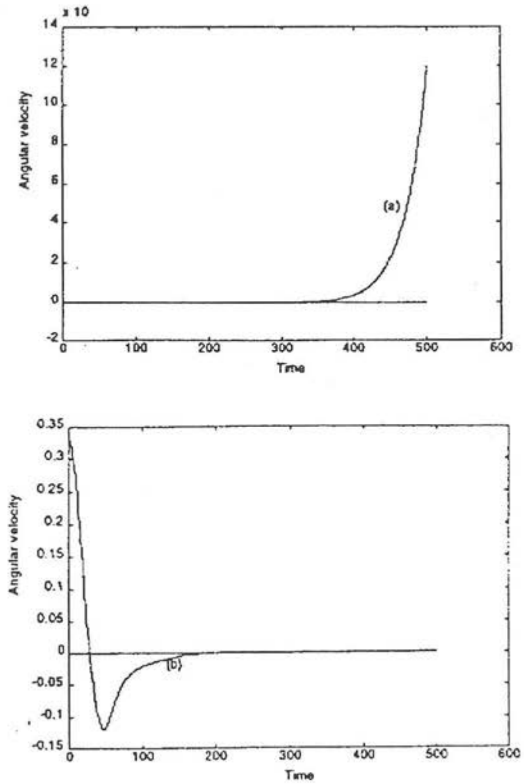

Figure 4. Controller's performance in terms of θ: (a) Before tuning of α, (b) After tuning of α for 6000 epochs

Figures 4(a) and 4(b) show the performance of the controller in terms of θ before and after tuning of α_r respectively. The system performance in terms of $\dot{\theta}$ is shown in Fig. 5. Table 3 shows the values of α_r after tuning.

To illustrate further that an *improper* or *inconsistent* choice of consequent value can be accounted for by modifying α_r, we consider the rule set corresponding to Table 3. The consequent value corresponding to the rule (PM,Z) is 122.07. We now change the consequent value of the rule to -1.0. In other words, the rule

If θ is PM and $\dot{\theta}$ is Z then $u = 122.07$ with $\alpha = 1.79$

is changed to

If θ is PM and $\dot{\theta}$ is Z then $u = -1.0$ with $\alpha = 1.79$

Note that, not only we reduced drastically the magnitude of the force, but also changed its sign. Fig. 6(a) (refers to the curve labeled (a) in Fig. 6) shows the system behavior (in terms of θ) after the rule damage. The variation of $\dot{\theta}$

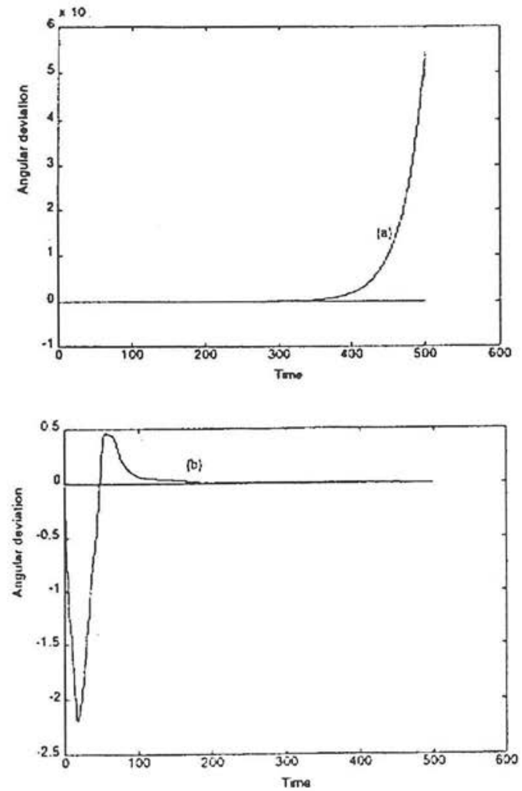

Figure 5. Controller's performance in terms of $\dot{\theta}$: (a) Before tuning of α, (b) After tuning of α for 6000 epochs

$\dot{\theta}$ / θ	NB	NM	NS	Z	PS	PM	PB
NB	.78		1.0		1.96		.65
NM		-.22		-.03		.79	
NS	.97		1.11	1.44	1.48		1.51
Z		.47	1.11	1.41	.34	1.11	
PS	.93		.65	.78	.22		.00
PM		1.54		1.79		.91	
PB	.04		.51		-.07		.72

Table 3. Values of α for different rules after 6,000 epochs, corresponding to Figs. 4(b) & 5(b).

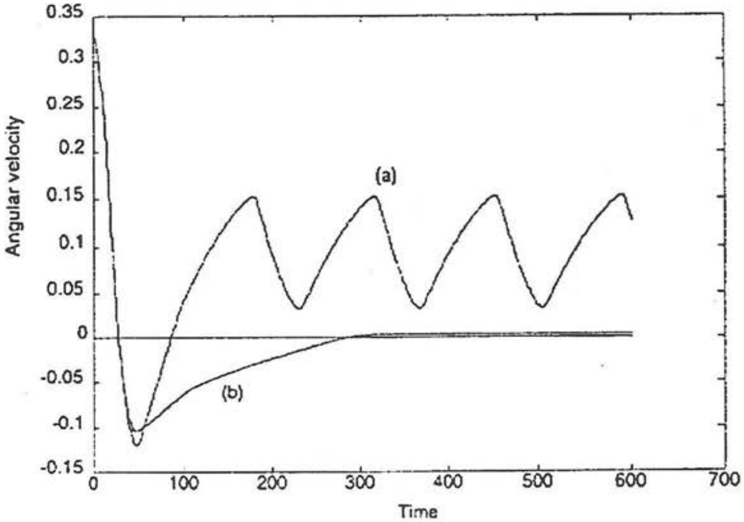

Figure 6. Performance of the controller in terms of θ: (a) After damaging the consequent of the rule *If* θ *is PM and* $\dot{\theta}$ *is Z then* $u = 122.07$ *with importance 1.79 to -1.0* (b) After tuning of only α for 2000 epochs (corresponds to Table 4)

under the same situation is given by Fig. 7(a). These two curves clearly show that with the damaged rule, the system is not controllable. The controller is now tuned for only 2000 epochs and Figs. 6(b) and 7(b) show the controller performance after tuning; while the corresponding weights are included in Table 4. Here also the uncontrollable system becomes controllable with only adjustment of relative importance factor. It is interesting to note that to account for the damaged rule the relative importance of one of the neighboring rules (with $\theta = PS$ and $\dot{\theta} = PS$) has changed from a positive value (0.22) to a negative one (-.13). Observe that not only the weight of the neighboring rules, but also the importance of several other rules have changed.

5.1. The utility of negative α_r

Inspection of the Tables 2, 3 and 4 reveals that most of the weights are positive, while a few are negative. As mentioned earlier at first sight, it might appear counter-intuitive because α's are interpreted as "importance" of rules. No; it is not counter-intuitive; on the contrary, it is a very powerful flexibility which can account for inconsistent or wrong rules that may be provided by the designer or user of the system. Let us explain it with an example. Let θ is PB and $\dot{\theta}$ is also PB. Under this situation, the force should also be positive (preferably, PB). Suppose, the designer provides a rule with a negative high consequent value, when θ is PB and $\dot{\theta}$ is PB. This is a bad rule, and under the original TS model,

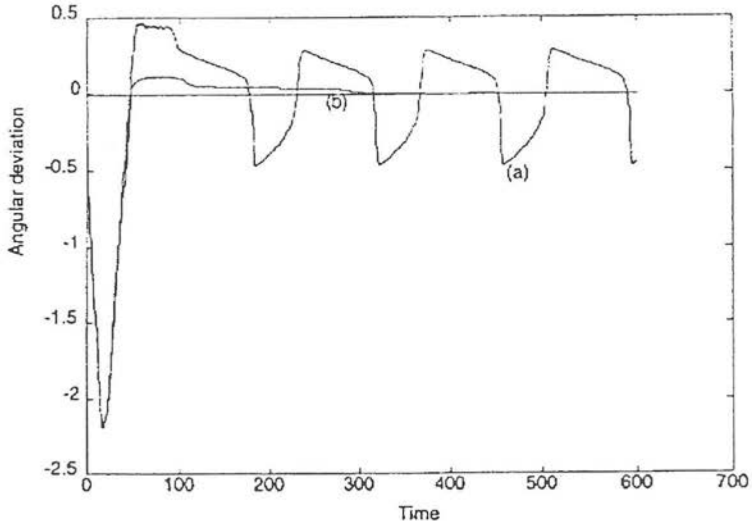

Figure 7. System response in terms of $\dot{\theta}$ for 49 rules (corresponds to Strategy 1) (a) After damaging the consequent of the rule *If* θ *is PM and* $\dot{\theta}$ *is Z then* $u = 122.07$ *with importance 1.79 to -1.0* (b) After tuning of only α for 2000 epochs (corresponds to Table 4)

θ \ $\dot{\theta}$	NB	NM	NS	Z	PS	PM	PB
NB	.19		1.00		2.49		.15
NM		-.05		-.12		.01	
NS	1.31		1.14	.47	1.72		1.67
Z		.26	1.25	.69	.04	1.17	
PS	1.2		.38	1.76	-.13		-.003
PM		1.69		.32		1.10	
PB	.06		.37		-.83		.52

Table 4. Values of α for different rules after 2,000 epochs, corresponding to Figs. 6(b) & 7(b)

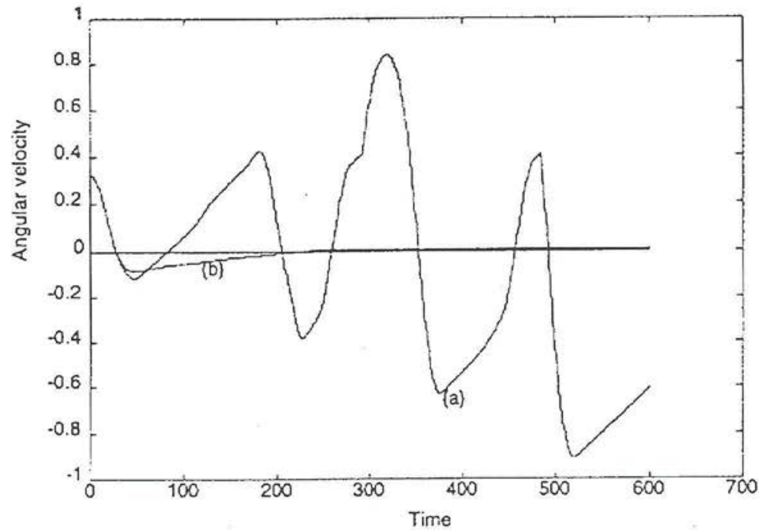

Figure 8. Figure explaining the role played by negative weights (in terms of θ) (a) After changing a positive consequent value (corresponding to Table 3) to a negative one, (b) After tuning of α for 15000 epochs

the system may not be controllable. But ETS may convert this inconsistent rule to a consistent one by adjusting the sign and magnitude of importance factor of the concerned rule and some of its neighboring rules, so that the contribution of this rule to the defuzzification scheme becomes negative. Changing the sign and / or magnitude of only the corresponding rule may not always be enough because the changed α influences both numerator and denominator of Eqn. (5). As an example, consider the following rule from Table 3,

if θ *is PM and* $\dot{\theta}$ *is Z, the force is 122.07 with importance 1.79.*

Changing the value of the force for this rule to a negative big value, say -200.07 (by keeping the importance factor same as before) gives rise to a system which is not controllable. After further tuning of the entire rule set for α with the help of Importance-Tune algorithm, the system again becomes controllable. Figs. 8 and 9 depict the system characteristics before and after modification of the consequent value of the above particular rule, and after tuning of α. After tuning, the values of weights are presented in Table 5. Comparison of Table 3 and Table 5 reveals that the change of the consequent value from 122.07 to -200.07 leads to a change of weight from 1.79 to a negative value of -0.29 for that particular rule we modified the consequent value of. We also observe that the value of the importance factor of a neighboring rule *(*θ *is PS and* $\dot{\theta}$ *is Z)* is also changed from 0.78 to 1.4. As expected, α_r for other rules have not changed much.

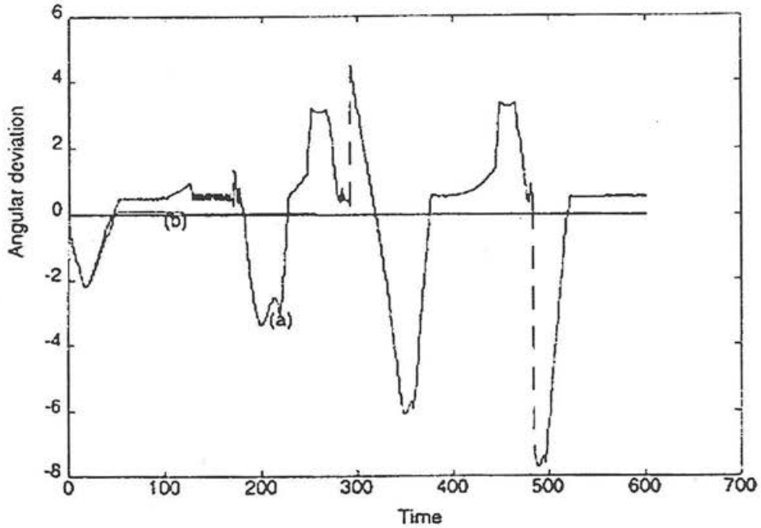

Figure 9. Figure illustrating the role played by negative weights (in terms of $\dot{\theta}$) (a) After changing a positive consequent value (corresponding to Table 3) to a negative one, (b) After tuning of α for 15000 epochs

θ \ $\dot{\theta}$	NB	NM	NS	Z	PS	PM	PB
NB	.73		1.00		2.01		.62
NM		-.21		-.04		.84	
NS	1.17		1.36	1.21	1.59		1.51
Z		.22	1.34	1.29	.61	1.12	
PS	1.01		.79	1.4	.49		-.004
PM		1.56		-.29		.997	
PB	.05		.50		-.12		.71

Table 5. Values of α after tuning of 15000 epochs, when Table 3 is used as the initial α and the consequent of one rule is changed from 122.07 to -200.07.

6. Non-negative importance factors

The proposed extended model discussed in Section 3 does not restrict the importance factor to be non-negative. This equips us with the capability of dealing with inconsistent rules. But as explained earlier, if the training data do not adequately represent the system to be identified, it may lead to the degenerate case with zero denominator. Moreover we cannot use α_r for rule elimination. However, if we could restrict the importance factor to be non-negative, then it would enable us to solve the difficult problem of rule selection easily, and we would be able to avoid the possibility of the degenerate case. Moreover, interpretation of α_r as rule importance would then be very natural. To achieve this, we modify the extended model with rules of the form:

$R(r)$ If (x_1 is X_{r1}) and ... (x_n is X_{rn}) then $u = u_r = F_r(x_1, \ldots, x_n)$ with importance α_r

where $\alpha_r \geq 0$ (non-negative). Although we can start with a set of non-negative α_r, we need to ensure that the learning process maintains the same. As such gradient descent on $\sum (y_i - y_i')^2$ does not guarantee that α_r will remain non-negative after the training. To impose this constraint we model α_r by ${\beta_r}^2$, i.e., we assume $\alpha_r = {\beta_r}^2$, where β_r is unrestricted in sign and we learn it using gradient descent.

The defuzzification scheme is then accordingly modified as

$$y_i' = \frac{\sum_{r=1}^{k} f_r * u_r * (\beta_r)^2}{\sum_{r=1}^{k} f_r * (\beta_r)^2}. \tag{11}$$

The learning rules for updating of β_r can be easily computed as

$$\beta_r(t+1) = \beta_r(t) - \eta_\beta(t) * \frac{(y_i' - y_i) * (u_r - y_i')}{\sum_{j=1}^{k} f_j * {\beta_j}^2} \tag{12}$$

where η_β is the learning co-efficient for the importance factors of the rules.

Note that the learnt value of β_r could be negative, but the actual rule importance factor α_r ($= {\beta_r}^2$) would always be non-negative. To select a necessary subset of rules, we propose to start with a large number of rules, each having the same importance factor of unity, 1. Then after tuning (learning) of the importance factor (β_r^2) we can delete all rules with $\beta_r^2 < \epsilon$ - a preassumed positive constant.

We now illustrate the use of non-negative rule importance factors for rule selection.

7. Results

We start with a set of 49 rules having initial importance factor of $\alpha_r = 1$ for all rules. Table 6 shows the values of α_r's after tuning the rule base with equations (7) and (9); i.e., assuming unrestricted α_r as the rule importance. With Table 6 the RMS error for the given data set is 7.84.

$\dot{\theta}$ / θ	NB	NM	NS	Z	PS	PM	PB
NB	1.02	1.03	1.05	0.99	0.9	1.03	1.93
NM	1.03	0.92	1.08	1.00	1.22	1.30	0.01
NS	0.94	0.68	0.75	1.23	0.62	1.07	0.00
Z	0.14	0.82	1.62	1.55	0.52	-.78	0.00
PS	-.06	-.65	0.63	1.19	0.90	0.56	0.98
PM	0.07	0.86	0.99	0.97	0.98	0.94	1.03
PB	2.0	1.18	1.04	1.08	0.99	1.03	1.02

Table 6. Final values of α_r for different rules after tuning for 5000 epochs with (7) and (9)

The same initial rule set is again separately tuned using equation (12), i.e., assuming $\alpha_r = {\beta_r}^2$, as the importance factor. Table 7 shows the values of β_r's (i.e., the positive square roots of the importance factors) after 5000 epochs of training. Since ${\beta_r}^2$ (≥ 0) is used as importance, we dropped the sign of β_r, and all are shown as positive values. The RMS error with Table 7 is found to be 4.94, which is much smaller than that with Table 6. Observe that in Table 7 many entries are practically zero. The rules, whose importance factors are nearly zero, do not contribute much to the computation of the defuzzified values and hence they can be deleted. In this particular case after deleting all rules with $\alpha_r < 0.01$, we get a rule base with only 26 rules. The deleted rules are marked with asterisk (*) in Table 6. The RMS error with this reduced rule set is 4.94 which is exactly the same as that before deletion. Thus we see that the extended model can be easily used for rule selection / elimination.

8. Conclusion

We have extended the conventional fuzzy controller model with the introduction of importance factor for each rule. In a rule based system, each rule may not have equal influence to accomplish the objective of the system. Moreover, rules obtained from experts may not always be reliable or consistent. The rule importance factor can account for such uncertainty associated with rules. The extended system enables the FLC to model more complex control surfaces and thereby makes the system more flexible. We demonstrated the effectiveness of this extended model under the Takagi-Sugeno framework. We empirically established (through extensive experiments) that an improper choice of rule-base can be very efficiently handled by our extended model. We also discussed how one can impose the non-negativity constraint on importance factors and use them to select a small subset of rules to achieve the goal of the system.

$\dot{\theta}$ / θ	NB	NM	NS	Z	PS	PM	PB
NB	1.11	1.18	2.08	0.03	*0.00	*0.00	*0.00
NM	1.18	*0.00	3.21	*0.00	0.90	*0.00	*0.00
NS	*0.00	*0.00	0.34	0.46	0.40	*0.00	*0.00
Z	*0.00	0.03	2.54	2.17	0.02	*0.00	*0.00
PS	*0.00	*0.00	0.07	0.75	0.16	*0.00	*0.00
PM	*0.00	*0.00	*0.00	0.01	0.27	*0.00	1.41
PB	*0.00	1.14	0.17	2.98	0.62	1.41	1.30

Table 7. Final positive values of β_r for different rules after tuning for 5000 epochs with (12)

Experiments are underway to establish the usefulness of the importance factor for fuzzy controllers having fuzzy consequent.

Acknowledgements

Authors gratefully acknowledge Prof. S.K. Pal for his encouragement and help to complete this work. The work has been partially supported by the Department of Science and Technology, Govt. of India (Grant No. III.5(21)/96-ET) and the Council for Scientific and Industrial Research, India.

References

BERENJI, H.R. (1992) A reinforcement learning-based architecture for fuzzy logic control. *International Journal of Approximate Reasoning*, **6**, 267-292.

BERENJI, H.R. and KHEDKAR, P. (1992) *Learning and tuning fuzzy logic neural networks*, **3**, 5.

DRIANKOV, D., HELLENDOORN, H. and REINFRANK, M. (1993) *An Introduction to Fuzzy Control*. Springer-Verlag, New York.

GUELY, F. and SIARRY, P. (1993) Gradient descent method for optimizing various rulebase. *Proceedings of the second IEEE conference on Fuzzy Systems FUZZ IEEE '93*, 1241-1246.

HARRIS, C.J., MOORE, C.G. and BROWN, M. (1993) *Intelligent Control - aspects of Fuzzy Logic and Neural Nets*. Singapore, World Scientific.

HAYASHI, S. (1991) Auto-Tuning Fuzzy PI Controller. *Proc. IFSA '91*, 41-44.

HOMAIFAR, A. and MCCORMICK, E. (1995) Simultaneous design of membership functions and rule sets for fuzzy controllers using genetic algorithms. *IEEE Trans. Fuzzy Systems*, **3**, 2, 129-139.

ISOMURSU, P. and RAUMA, T. (NO DATE) A self-tuning fuzzy logic controller for temperature control. *Proceedings of the third IEEE International Conference on Fuzzy Systems*, **III**, 1560-1563.

IWASAKI, T. and MORITA, A. (1990) Auto-Tuning Controller with Fuzzy Identification. *Proc. Int. Conf. Fuzzy Logic and Neural Networks*, 401-404.

JANG, J.S.R. (1992) Fuzzy controller design without domain expert. *Proceedings of the first international conference on Fuzzy Systems FUZZ IEEE '92*, 289-296.

KARR, C.L. and GENTRY, E.J. (1993) Fuzzy control of pH using Genetic Algorithms. *IEEE Trans. Fuzzy Systems*, **1**, 1, 46-53.

LEE, C.C. (1990) Fuzzy logic in control system: fuzzy logic controller - Part I & II. *IEEE Trans. Syst. Man, Cybern.*, **SMC-20**, 2, 404-435.

LUI, H.C., GU, M.K., GOH, T.H. and WANG, P.Z. (1994) A self-tuning adaptive resolution (STAR) fuzzy control algorithm. *Proceedings of the third IEEE International Conference on Fuzzy Systems*. **III**, 1508-1513.

MAEDA, M. and MURAKAMI, S. (1992) A self-tuning fuzzy controller. *Fuzzy Sets and Systems*, **51**, 29-40.

MAMDANI, E.H. and ASSILIAN, S. (1975) An experiment in linguistic synthesis with a fuzzy logic controller. *Int. J. Man Mach. Studies*, **7**, 1, 1-13.

NOMURA, H., HAYASHI, I. and WAKAMI, N. (1991) A self tuning method of fuzzy control by gradient descent method. *Proceedings of international Fuzzy Systems Association IFSA '91 Bruxelles, Engineering*, 155-158.

PALM, R. (1995) Scaling of Fuzzy Controller Using the Cross-Correlation. *IEEE Trans. on Fuzzy Syst.*, **3**, 1, 116-123.

PARK, Y.M., MOON, U. and LEE, K.Y. (1995) A self-organizing fuzzy logic controller for dynamic systems using a fuzzy autoregressive moving average (FARMA) model. *IEEE Trans. Fuzzy Systs.*, **3**, 1, 75-82.

PROCYK, T.J. and MAMDANI, E.H. (1979) A linguistic self-organizing process Controller. *Automatica*, **15**, 15-30.

SHAO, S. (1988) Fuzzy self-organizing controller and its application for dynamic Processes. *Fuzzy Set Syst.*, **26**, 151-164.

SUGENO, M. (1985) *Industrial Applications of Fuzzy Control.* Amsterdam, Elsevier Science.

SUGENO, M. and KANG, G.T. (1988) Structure identification of fuzzy model. *Fuzzy sets and Systems*, **28**, 15-33.

TAKAGI, T. and SUGENO, M. (1985) Fuzzy identification of systems and its applications to modeling and control. *IEEE Trans. Syst. man and Cybern.*, **15**, 116-132.

YAGISHITA, O., ITHO, O. and SUGENO, M. (1985) An application of fuzzy reasoning to water purification process. *Industrial Applications of Fuzzy Control.* New York, North-Holland, 19-40.

YAMAKAWA, T. (1992) A fuzzy logic controller. *Journal of Biotechnologies*, **24**, 1-32.

YASUNOBU, S. and MIYAMOTO, S. (1985) Automatic train operation by predictive fuzzy control. *Industrial Applications of Fuzzy Control.* New York, North-Holland, 1-18.

YOSHIDA, M., TSUTSUMI, Y. and ISHIDA, T. (1990) Gain Tuning Method for Design of Fuzzy Control Systems. *Proc. Int. Conf. Fuzzy Logic and Neural Networks*, 405-408.

EURO Prize for the best applied paper (EURO 2000 Conference, Budapest, July 2000)

Conference venue and dates:

The EURO 2000 Conference (in Budapest, July 2000) will include the competition for the "EURO Prize for the best applied paper". All interested authors are invited to submit papers describing an application of Operational Research which has original features, whether in methodology, application or implementation.

The purposes of the competition are:

- to recognise outstanding accomplishments in OR practice,
- to attract more application-oriented papers to EURO Conferences,
- to promote OR practice in general.

In submitting a paper, the authors should pay attention to the following:

- it will be presented at the EURO-2000 Conference in case of selection,
- it has not been submitted concurrently to another competition,
- the jury may take any contact it feels appropriate to evaluate the paper,
- in case it is selected in the short-list, the paper will be submitted to a feature issue of EJOR guest-edited by the chairman of the jury; the jury will review submissions and select a short-list of finalist papers,
- the fees of the authors of the finalist papers will be waived (in case of multiple authors the fee is waived for only one of the authors); these papers will be presented in "special session(s)".

The winner of the Prize will be determined by the jury after the presentation of the finalist papers. The winner of the Prize will be announced by the chairman of the jury during the closing session. The winner of the Prize will give a brief presentation (15 minutes) of the paper. The EURO Prize for the best applied paper should be considered as a considerable honour. The Prize will consist of:

- an ad-hoc certificate and medal,
- the exemption of fees for the next three EURO Conferences (in case of multiple authors, there will be only three exemptions).

The schedule for the EURO 2000 Conference is as follows

- 30 November 1999: Deadline for the submission of papers.
- December 1999 - March 2000: Jury examines submissions.
- April 2000: Jury announces the finalists papers. Referees are selected for the finalist papers. Composition of the special sessions transmitted to the Programme and Organising Committees.
- July 2000: Special sessions organised by the chairman with the finalist papers. Winner of the Prize announced during the closing session. Winning paper is briefly presented during the closing session.

All papers must be submitted in paper form as well as in electronic form (Word 6.0/95) attachment to an email to the Chairman of the Jury, Professor Costas P.Pappis at the following address:

University of Piraeus, Dept.of Industrial Management
80 Karaoli & Dimitriou Str., 18534 Piraeus, Greece
Tel. (+301) 4222060, 4120751/ext.316, Fax (+301) 4179064
email: `pappis@unipi.gr`