

A neuro-fuzzy inference system with improved performance

S.G. Tzafestas & G.B. Stamou

Intelligent Robotics and Control Unit, Department of Electrical and Computer Engineering, National Technical University of Athens, Zografou 15773, Athens, Greece

Abstract

Fuzzy systems and neural systems belong to the class of model free estimators and possess a high degree of parallelism. Due to these features several investigators have tried to produce several models of neuro-fuzzy inference systems with combined properties. The purpose of the present paper is to extend and improve one of these neuro-fuzzy structures such that to obtain better inferences. The system is based on the Keller-Yager-Tahani (K-Y-T) neuro-fuzzy model and uses Hamacher's intersection function $t_H(a,b)$ or Sugeno's complement function $c_\lambda(a)$. The operation of the system is briefly described and its features are established in the form of four theorems. The capabilities of the system are shown by a set of simulation results derived for the case of trapezoidal fuzzy sets. These results are shown to be better than the ones obtained with the original neuro-fuzzy system of Keller, Yager and Tahani.

1. Introduction

Neural systems and fuzzy systems have essential differences in the way they function. Neural systems are based on pattern recognition supervised or unsupervised learning techniques, whereas fuzzy systems are based on fuzzy set theory and fuzzy inference mechanisms. However neural and fuzzy systems have in common two important characteristics, namely:

- i) they both possess a high degree of parallelism, and
- ii) they both belong to the class of the model free estimators.

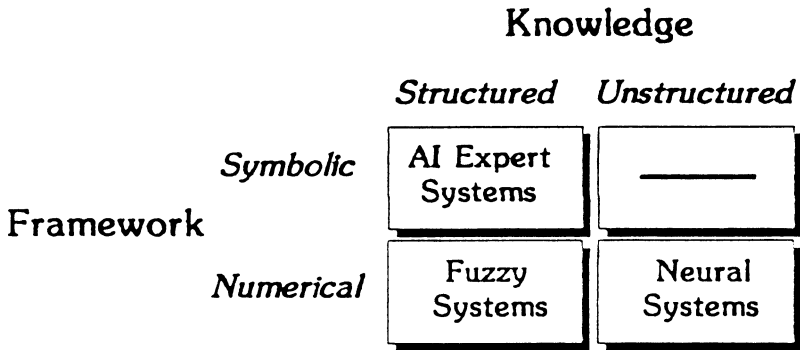


Fig.1: Taxonomies of AI Expert, Fuzzy and Neural systems.

Fig.1 shows the position of fuzzy and neural systems according to the knowledge framework (symbolic, numeric) and knowledge type (structured, unstructured) taxonomies. It is clear that the combination of fuzzy and neural systems is possible and indeed there are currently available several forms of fuzzy-neural systems [1-5]. Our purpose here is to present and study a convenient neural network which implements a kind of fuzzy inference.

Each neural net is known to consist of a set of elementary units with n synapses and weights w_{ij} which are connected to a node f described by a strongly nonlinear operation. These units are referred to as 'neurons' (Fig.2).

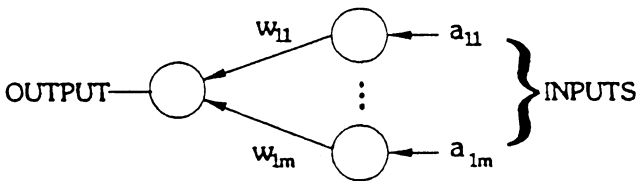


Fig.2: Basic structure of a neuron.

The purpose of any learning procedure is to update weights w_{ij} such that for each input for the neural network, the output is to be the desired one. Actually, a plethora of neural network models and learning algorithms have been proposed in the technical literature (see e.g. [6-11]). Of course there exist many situations

where the weights are selected right from the beginning by the designer without the application of any learning algorithms. The system presented here belongs to this category.

The fuzzy systems were introduced by Zadeh in 1965 [12] and have shown a significant development due to their ability for solving complex reasoning problems. They are based on fuzzy logic and fuzzy sets. The fundamental operations of fuzzy sets are: fuzzy complement, fuzzy union and fuzzy intersection [13]. With the aid of these operations, the extension of logical tautologies becomes possible.

A particular tautology which is directly useful in fuzzy controllers is the modus ponens (MP) rule:

$$((A \rightarrow B) \wedge A) \rightarrow B$$

which indicates that in a cause-effect relationship, when the cause is satisfied the consequence is obtained. The question is "what happens when A and B are fuzzy sets?". In this case the above MP rule is extended to the following generalized modus ponens (GMP) rule:

$$((A \rightarrow B) \wedge A') \rightarrow B'$$

which shows that when the fuzzy cause is almost satisfied we nearly get the consequence. One way to implement the GMP rule is through the use of Zadeh's max-min composition operator [12,18].

A fuzzy controller consists of the following parts (Fig.3) [6,14,18]:

- fuzzification
- fuzzy inference mechanism
- defuzzification

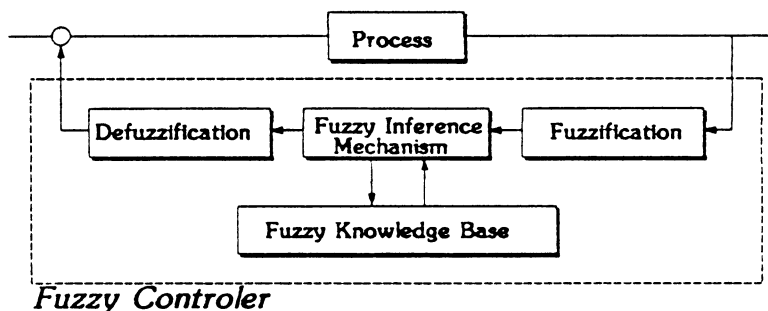


Fig.3: General structure of a fuzzy controller.

The fuzzy inference mechanism is implemented with the aid of a fuzzy knowledge base, i.e. a knowledge base that contains fuzzy rules (MIMO: Multi-Input/Multi-Output rules). It is shown in [16] that these rules can be transformed to MISO (Multi-Input/Single-Output) rules. If the knowledge base contains many such rules, then one can assume that the inference is done in the union of them. It can be shown [16] that it is sufficient to apply the input to every rule and get as a result the union of the outputs. Thus the design of rules of the type:

IF X_1 IS A_1 AND X_2 IS A_2 AND ... X_n IS A_n THEN Y IS B (1)

plays an important role in the design of a fuzzy controller. The inference is performed by using the GMP rule.

2. Fuzzy inference via neural nets

From the discussion of the previous section it is clear that there are some basic similarities between neural and fuzzy systems. Our purpose in the present section is to try to use a neural net for drawing fuzzy conclusions. To this end we shall extend and improve the solution proposed by Keller, Yager and Taham [15]. This solution has the drawback that, although it possesses a correct behaviour, it is away from the desired solution, especially when the fact (event) occurs. Thus an improvement will be proposed here together with a discussion on the application domain of the network.

The proposed neuro-fuzzy network must satisfy the following requirements:

a. The fuzzy sets must be given in sampled form, since every neural network has a finite number of inputs, i.e. A_i and B in the rule (1) must be of the form:

$$A_i = \{a_{i1}, \dots, a_{in_i}\}, \quad n = m_1, \dots, m_n \text{ for } i = 1, \dots, n$$

$$B = \{b_1, \dots, b_n\}$$

Therefore, for input

$$A'_i = \{a'_{i1}, \dots, a'_{in_i}\}, \quad n = m_1, \dots, m_n \text{ for } i = 1, \dots, n$$

one must get the output

$$B' = \{b'_1, \dots, b'_n\}$$

b. In case the sets A'_i are crisp sets the network must produce results that agree with the rules of classical logic.

In the following, it is proved that every neural network of the type shown in Fig.4 with describing equations

$$w_{ij} = c(a_{ij})$$

$$d_k = u_j \{i(a'_{ij}, w_{ij})\}$$

$$s = c(u_k(d_k))$$

$$u_i = c(b_i)$$

$$b'_i = c\{i(u_i, s)\}$$

where

c : involutive fuzzy complement with the property $c(c(a)) = a, \forall a$,

u : fuzzy union with the property $u(a, 0) = a$ and $u(a, 1) = 1$,

i : fuzzy intersection with the property $i(a, 0) = 0$ and $i(a, 1) = a$,

satisfies the above conditions a. and b.

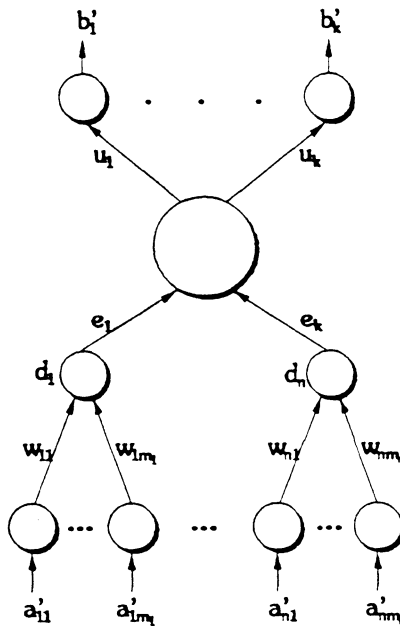


Fig.4: Structure of the proposed neural network (assuming that $e_i = 1, \forall i = 1, 2, \dots, k$).

Theorem 1

If $A'_i = A_i$ for all $i = 1, 2, \dots, n$, then the result is B .

Proof

From $A_i = A'_i$ it follows that $a_{ij} = a'_{ij}$ for all $i = 1, 2, \dots, n$ and so for all v_{ij} in the domain of discourse it is true that $\mu(v_{ij}) = a_{ij} = 1$ or 0 and $\mu(c(v_{ij})) = 0$ or 1 , respectively, where $\mu(x)$ is the membership function of concern. That is $\forall i = 1, 2, \dots, n$ the following relation holds:

$$d_k = u \{ \{ i(a_{ij}, c(a_{ij})) \} \} = 0$$

since $\{ i(a_{ij}, c(a_{ij})) \} = i(1, 0) = 0$ for all i and j of interest. Thus

$$s = c \{ u(d_k) \} = c(0) = 1$$

and so $b'_i = c \{ i(u_i, s) \} = c(u_i) = c(c(b_i)) = b_i$ for $i = 1, 2, \dots, n$. This means that the result is B as required. q.e.d.

Theorem 2

If for some i from 1 to n the condition $A_i^c \cap A'_i \neq 0$ holds, then the neural network produces the result "y is UNKNOWN" (which means that $b'_i = 1 \forall i = 1, 2, \dots, n$).

Proof

Since $\mu_{A_i^c}(v_{ij}) = \mu_{A'_i}(v_{ij}) = 1$ for some v_{ij} in the domain of discourse and $i(a'_{ij}, w_{ij}) = 1$ for this v_{ij} ,

$$d_k = u \{ \{ i(a'_{ij}, w_{ij}) \} \} = 1$$

Thus $s = c \{ u(d_k) \} = 0$ and so $b'_i = c \{ i(0, u_i) \} = c(0) = 1, \forall i = 1, 2, \dots, n$. q.e.d.

Theorem 3

If $A'_i \subseteq A_i$ (i.e. A'_i is more specific than A_i - $\mu_{A'_i} \leq \mu_{A_i}$) for some i from 1 to n , then the result is B .

Proof

From $A'_i \subseteq A_i$ for some i , it follows that $A'_i \cap A_i^c = 0$. Hence $d_k = 0$ which gives the result $\neg Y$ is B' . q.e.b.

This network is used to solve problems similar to the following example [15]. Consider the rule:

IF a person is OLD THEN the person is UNHEALTHY.

If some time we learn that the person is VERY OLD, then this information is sufficient to say that the person is UNHEALTHY. To confirm this fact the following theorem will be shown.

Theorem 4

Consider two inputs A'_i and A''_i , $i = 1, 2, \dots, n$ of the neural network and suppose that $A''_i \subseteq A'_i \subseteq A_i$ for all $i = 1, 2, \dots, n$. Then the corresponding results B' and B'' will satisfy the condition

$$B \subseteq B'' \subseteq B'$$

Proof

From the relation $A''_i \subseteq A'_i \subseteq A_i$, $\forall i = 1, 2, \dots, n$, it follows that

$$0 \leq \mu_{A''_i}(\cdot) \leq \mu_{A'_i}(\cdot) \leq \mu_{A_i}(\cdot), \quad \forall i = 1, 2, \dots, n$$

for any point in the domain of discourse.

Thus

$$d''_k = \bigcup_j \{I(a''_{ij}, w_{ij})\} \leq \bigcup_j \{I(a'_{ij}, w_{ij})\} = d'_k, \quad \forall i = 1, 2, \dots, n$$

and so if $d'_k \leq d''_k$ then $\bigcup_k (d'_k) \geq \bigcup_k (d''_k)$ which implies that

$$c\{\bigcup_k (d'_k)\} \leq c\{\bigcup_k (d''_k)\}$$

or

$$s' \leq s''$$

Thus we have $b'_i \geq b''_i$ i.e. $B \subseteq B'' \subseteq B'$ because $b'_i, b''_i \geq b_i$, $\forall i = 1, 2, \dots, n$ where

$$b'_i = c\{I(u_i, s)\} = c\{I(c(b_i), s)\} \geq c\{c(b_i)\} = b_i$$

Remark

For any input A'_i it will be true that $B \subseteq B'$. It is therefore clear that for any triple of functions (c, u, I) the network will be working for examples of the

“unhealthy old person” type mentioned above. For this purpose we propose the following neuron model:

$$\begin{aligned}w_{ij} &= 1 - a_{ij} \\d'_k &= \max_j (i_H(a'_{ij}, w_{ij})) \\s &= 1 - \max_k \{d'_k\} \\u_i &= 1 - b'_i \\b'_i &= 1 - s \cdot u_i\end{aligned}$$

where $i_H(\cdot)$ is Hamacher's function [13]:

$$i_H(a, b) = \frac{ab}{\gamma + (1 - \gamma)(a + b - ab)}, \quad \gamma \in (0, \infty)$$

which satisfies the conditions $i_H(a, 1) = a$ and $i_H(a, 0) = 0$. As it will be seen in the next section, this model works correctly for relatively large values of γ ($\gamma=100, 200, \dots$). However, there exist rules of other types that are also compatible with the GMP rule. To see this we give the following example.

Suppose that two taps feed water to a vessel. The temperature of the water flowing from the first tap is known but not controlled, whereas the temperature of the water in the second tap is controlled with a switch that has the following indications: VERY VERY WARM, VERY WARM, WARM, NEARLY WARM, A LITTLE WARM, MODERATELY WARM, A LITTLE COOL, NEARLY COOL, COOL, VERY COOL, VERY VERY COOL. We want to keep the water in the vessel MODERATELY WARM.

The logic of the previous model is as follows. If the water of tap 1 is A LITTLE WARM or NEARLY WARM or VERY WARM or VERY VERY WARM then open the COOL water in tap 2. In this way, we will get the correct result with a delay. It would be better to work in the following way. If the water in tap 1 is WARM then open in tap 2 the COOL water, if it is VERY WARM then open the VERY COOL water and if it is VERY VERY WARM then open in tap 2 the VERY VERY COOL water, and so on. In this way we will get the desired MODERATELY WARM water very fast. However, the proposed inference model cannot produce this logic due to Theorem 4. Of course if in the computation of the u_i 's one uses a different function for the complement operation than that used for the b'_i 's, then it could be possible to obtain $b'_i < b_i$.



But in this case, the proofs of Theorems 1, 2 and 3 should be modified. One must therefore be very careful in using this model in each particular case.

In the light of the above, we propose here the following modification of the inference network:

$$w_{ij} = c_1(a_{ij})$$

$$d_k = \max_j(a'_{ij}, w_{ij})$$

$$s = c_1(\max_k(d_k))$$

$$u_i = c_2(b_i)$$

$$b_i = c_1(su_i)$$

where c_1 and c_2 are complementation functions of the Sugeno type [13]:

$$c_\lambda(a) = \frac{1 - a}{1 + \lambda a}, \quad \lambda \in (-1, \infty)$$

In practice one must select suitable values for λ_1 and λ_2 . This choice can be made through simulation as described below.

Table 1: The meaning of linguistic terms defined on the domain [1,11] and sampled at integer points

Label	Values of the membership function										
LOW	1.00	0.67	0.33	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
VERY LOW	1.00	0.45	0.11	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
MORL LOW	1.00	0.82	0.57	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
NOT LOW	0.00	0.33	0.67	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
MEDIUM	0.00	0.00	0.25	0.50	0.75	1.00	0.75	0.50	0.25	0.00	0.00
MORL MEDIUM	0.00	0.00	0.50	0.71	0.87	1.00	0.87	0.71	0.50	0.00	0.00
NOT MEDIUM	1.00	1.00	0.75	0.50	0.25	0.00	0.25	0.50	0.75	1.00	1.00
HIGH	0.00	0.00	0.00	0.00	0.00	0.00	0.20	0.40	0.60	0.80	1.00
VERY HIGH	0.00	0.00	0.00	0.00	0.00	0.00	0.04	0.16	0.36	0.64	1.00
MORL HIGH	0.00	0.00	0.00	0.00	0.00	0.00	0.45	0.63	0.77	0.89	1.00
NOT HIGH	1.00	1.00	1.00	1.00	1.00	1.00	0.80	0.60	0.40	0.20	0.00
UNKNOWN	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00

3. Simulation example

The proposed neural networks were simulated for trapezoidal fuzzy sets [17] according to the rule:

IF X IS LOW THEN Y IS HIGH

The results obtained are shown in Tables 2 through 5. We observe that they approach the desired result, the first without deviation and the second with some tolerable deviation.

Table 2: The results of the simulation of the first model using Hamacher's function ($v=5.7$)

[illegible]

Table 3: The results of the simulation of the first model using Hamacher's function ($v=100$)

[illegible]

Table 4: The results of the simulation of the first model using Hamacher's function ($v=200$)

[illegible]

Table 5: The results of the simulation of the second model using Sugeno's function

[illegible]

In the first case (Tables 2-4) we present the results of the network for the Hamacher function with $\gamma=50, 100$ and 200 , respectively (the plots are omitted since they almost coincide with the desired results). In the second case (Table 5) we use the complement function of Sugeno with $\lambda_1 = 10$ and $\lambda_2 = 6$ and give the corresponding graphical representations.

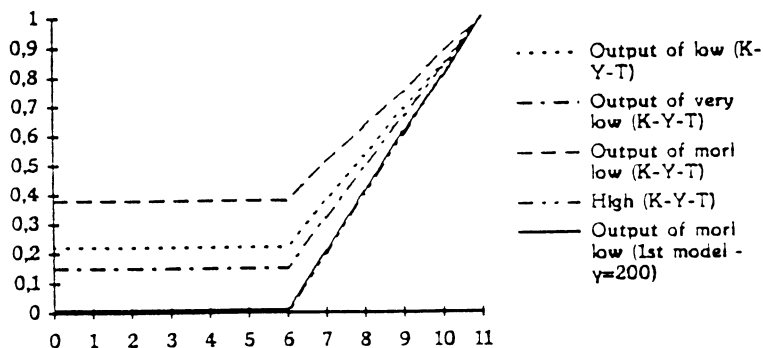


Fig.5: Comparison between the first model ($\gamma=200$) and the K-Y-T model (the outputs of low and very low of the first model are almost identical to High).

4. Conclusions

The neural network proposed here has the same structure as the K-Y-T model but it is generalized such that one can use any meaningful fuzzy intersection, union or complement function. This is reflected in the more generality of Theorems 1 through 4. Here we selected and used the Hamacher intersection function which leads to more accurate inferences than those obtained by the K-Y-T network. Also we used the Sugeno complement function which gave us another improvement. A further extension would be obtained if one uses rules of the type: "IF water is WARM THEN open COOL tap", "IF water is VERY WARM THEN open VERY COOL tap", and so on (as mentioned at the end of section 2).

The authors are currently working to design a system that can accomodate such rules and also in the area of tuning PID and other controllers with the aid of the present inference system in place of the so called "fuzzy



matrices" [19]. Hopefully the use of the present system will give equivalent tuning results as in [19] but more rapidly and with increased robustness.

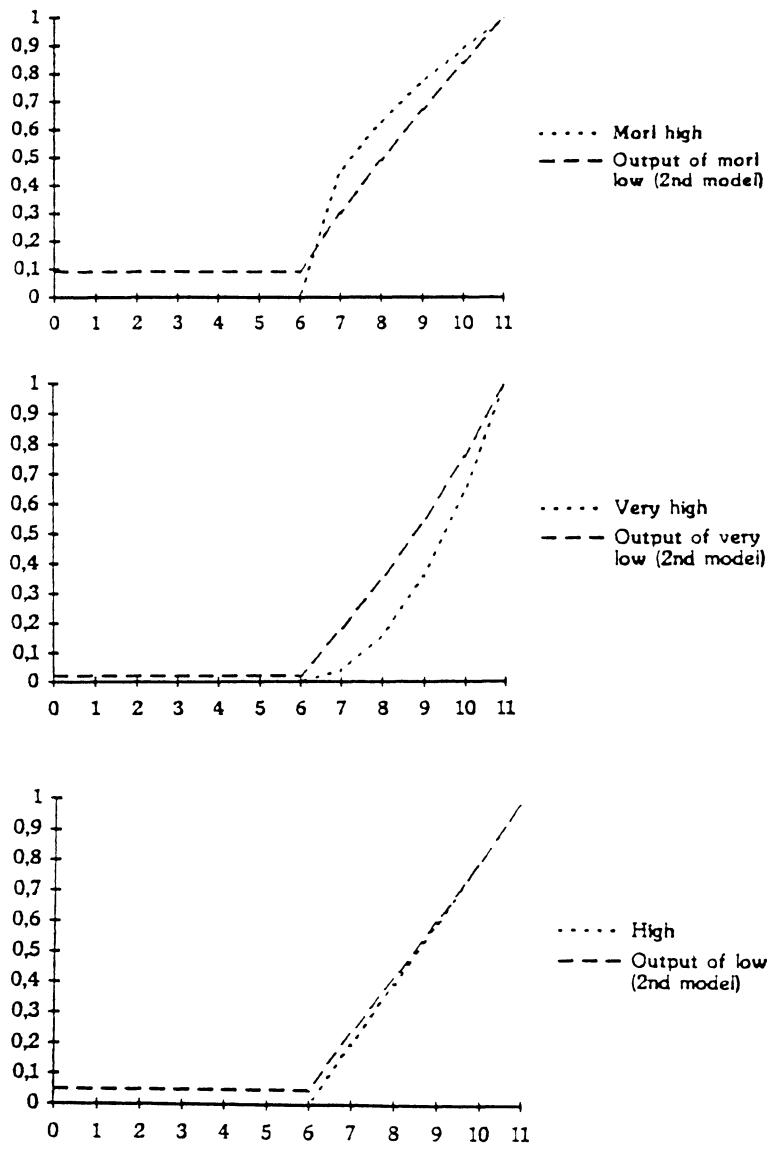


Fig.6: Comparison between the output of the second model and the desired one.

References

- [1] T. Yamakawa, 'Pattern Recognition Hardware System Employing a Fuzzy Neuron', *Proceedings of the International Conference on Fuzzy Logic and Neural Networks*, Izuka, Japan, July 20-24, 1990, pp. 943-948.
- [2] H. R. Berenji, 'Fuzzy and Neural Control', Chapter 9 in *An Introduction to Intelligent and Autonomous Control*, Antsaklis and Passino Eds Kluner, May 1992.
- [3] M. Brown and C. Harris, 'A Nonlinear Adaptive Controller: A Comparison between Fuzzy Logic and Neurocontrol', *IMA Journal of Mathematical Control and Information*, 1991.
- [4] K. Uehara and M. Fujise, 'Learning of Fuzzy-Inference Criteria with Artificial Neural Network', *Proceedings of the International Conference on Fuzzy Logic and Neural Networks*, Izuka, Japan, July 20-24, 1990, pp. 193-198.
- [5] S. Nakanishi, T. Takagi, K. Uehara and Y. Gotoh, 'Self-organizing Fuzzy Controllers by Neural Networks', *Proceedings of the International Conference on Fuzzy Logic and Neural Networks*, Izuka, Japan, July 20-24, 1990, pp. 187-193.
- [6] B. Kosko, *Neural Networks and Fuzzy Systems. A Dynamical Approach to Machine Intelligence*, Prentice Hall, Englewood Cliffs, NJ07632.
- [7] J. S. Judd, *Neural Network Design and the Complexity of Learning*, A Bradford Book, The MIT Press, Cambridge, Massachusetts, London, England.
- [8] P. J. Antsaklis, *Neural Networks in Control Systems*, IEEE Systems Magazine, April 1990.
- [9] R. P. Lippman, 'An Introduction to Computing with Neural Nets', *IEEE ASSP Magazine*, April 1987.
- [10] R. Battiti, 'First and Second Methods for Learning: Between Steepest Descent and Newton's Method', *Neural Computation* 4,141-166, Massachusetts Institute of Technology, 1992.
- [11] K. Watanabe and S. G. Tzafestas, 'Learning Algorithms for Neural Networks with the Kalman Filters', *Journal of Intelligent and Robotic Systems* 3: 305-319, 1990.
- [12] L. A. Zadeh, 'Fuzzy Sets', *Information and Control*, pp. 338-353, 1965.
- [13] G. J. Klir, T. A. Folger, *Fuzzy Sets, Uncertainty and Information*, Prentice Hall, Englewood Cliffs, N. J., 1988.



- [14] Flor Van Der Rhee, Hans R. Van Nauta Lemke, Jaap G. Dijkman, 'Knowledge Based Fuzzy Control of Systems', *IEEE Trans. on Automatic Control*, Vol. 35, NO. 2, February 1990.
- [15] J. M. Keller, R. R. Yager, H. Tahani, 'Neural Network Implementation of Fuzzy Logic', *Fuzzy Sets and Systems* 45, North-Holland, 1992, pp. 1-12.
- [16] C. C. Lee, 'Fuzzy Logic in Control Systems: Fuzzy Logic Controllers, Part I, II', *IEEE Trans. on Systems, Man and Cybernetics*, Vol. SMC-20, No. 2, March/April 1990.
- [17] M. Mizumoto, S. Fukami and K. Tanaka, 'Some methods of Fuzzy Reasoning', in: M. M. Gupta, R. K. Ragade and R. R. Yager, Eds., *Advances in Fuzzy Set Theory and Applications*, North Holland, Amsterdam, 1979, pp. 117-126.
- [18] S. G. Tzafestas, 'Fuzzy Expert Control: Recent Results with Applications to Robotic Systems', *Advanced Study Institute on knowledge Based Robot Control*, France, 1988.
- [19] S. Tzafestas and N. P. Papanikolopoulos, 'Incremental Fuzzy Expert PID Control', *IEEE Trans. on Industrial Electronics*, Vol. 37, NO. 5, October 1990.