Modeling and Identification of complex systems using Neuro-Fuzzy Techniques

نمذجة المنظومات المعقدة و الاستدلال عليها باستخدام الأسلوب الفازى من خلال الشبكات العصبية

M. El-Hosiny[*], *M. Sherif*[*], *S. Saraya*[*], *M. R. El-Basyouni*[**], *F. G. Areed*[*]

ملخص البحث :

في هذا البحث حاولنا إيجاد طريقة تجمع ما بين البساطة والدقة في نمذجه الأنظمة المعقدة و الاستدلال عليها وذلك باستخدام تقنيات الأسلوب اللفظي (الفازى) باستخدام الشبكات العصبية حيث أن هذه التقنيات تعتمد على المنطق بشكل أكبر من التعقيدات الرياضية .

Abstract:

In this paper Adaptive-Network-Based Fuzzy Inference System (ANFIS) architecture is presented and has been used as a tool for System Identification. System Identification consists of three related steps: 1:) Structure specification, 2:) Parameter estimation, 3:) Model Validation. These steps are also discussed. We use a new method to determine the structure of the ANFIS model (Fuzzy Curves). Fuzzy Curves Help in determining the number of significant inputs from a number of candidates. We determine the number of membership functions in each input by using the subtractive clustering. Finally this method is tested on two cases.

1. Introduction:

Both fuzzy systems and ANN's are soft computing approaches for modeling Expert behavior. The goal is to mimic the actions of an expert who solves complex problems. In other words, instead of investigating the problem in detail, one observes how an expert successfully tackles the problem and obtains knowledge by instruction and/or learning [10]. If an operator cannot tell linguistically what kind of action he takes in a situation, then it is quite useful to model his control actions using numerical data [9].

Estimating an unknown function from a set of input-output (I/O) data pairs has been and is still a key issue in a variety of scientific and engineering fields [2]. The concept of system modeling is closely related to interpolative input-output mapping, pattern classification, case based reasoning, and learning from examples [1]. In fact, the problem of system modeling is only a part of a more general problem- the problem of system identification [4].

The purposes of system identification are multiple:

- To predict a system's behavior, as in time series prediction and weather forecasting.
- To explain the interactions and relationships between inputs and outputs of a system. For example, a mathematical model can be used to examine whether the demand indeed varies proportionally to the supply in an economic system.
- To design a controller based on the model of a system, as in aircraft and ship control. Also to do computer simulation of the system under control, you need a model of the system [5].

While the theory of traditional equation-based approaches is well developed and successful in practice (particularly in linear cases) there has been a great deal of

[*] Corresponding authors are at Computers & Systems Dept., Faculty of Engineering, Mansoura University.

[**] Faculty of Specific Education, Mansoura University.

Accepted March 31, 2002

interest in applying model-free method such as neural and fuzzy techniques for system identification. When comparing fuzzy systems and neural networks, the former provides an attractive alternative to the "black boxes" characteristic of neural models, because their behavior can be easily explained by a human being [2]. However, there are some basic aspects of this approach that are in need of better understanding [6]. More specifically:

- ☒ No standard methods exist for transforming human knowledge or experience into the rule base and database of a fuzzy inference system.
- ☒ There is a need for effective methods for tuning the membership function (MF's) so as to minimize the output error measure or maximize performance index.

As In [10], Neural networks do not provide a strong scheme for knowledge representation, while fuzzy logic controllers do not possess capabilities for automated learning. Neuro-Fuzzy computing [10], which is a judicious integration of the merits of neural and fuzzy approaches, enables one to build more intelligent decision-making system. This incorporates the generic advantages of artificial neural networks like massive parallelism, robustness, and learning in data-rich environments into the system. The modeling of imprecise and qualitative knowledge as well as the transmission of uncertainty is possible through the use of fuzzy logic. One of the most famous Neuro-Fuzzy architecture is Adaptive Neuro-Fuzzy Inference System (ANFIS). Loosely speaking ANFIS is a method for tuning an existing rule base with a learning algorithm based on a collection of training data. This allows the rule base to adapt [3].

System identification consists of three basic sub problems:1) Structure specification ;2) Parameter estimation; 3) Model validation.

We will discuss each one of these steps later. We will use the method of Fuzzy Curves to determine the significant inputs, using clustering to determine the number of membership function in each input, and then using ANFIS Architecture for modeling the system. Then, I will apply this method for a number of examples to check the validity of the algorithm.

This paper is organized into six sections. In the next section, the basics of ANFIS are introduced. Section 3 explains the Identification algorithm. Application of the algorithm to Examples is demonstrated in section 4. Section 5 presents the analysis of results. Concluding remarks are discussed in section 6.

2. ANFIS ARCHITECTURE:

For simplicity, we assume that the fuzzy inference systems under consideration has two inputs x and y and one output z. suppose that the rule base contains two fuzzy if-then rules of Takagi and Sugeno's type.

Rule 1: if x is A1 and y is B1, then $f1= p1x + q1y + r1$
Rule 2: if x is A2 and y is B2, then $f2= p2x + q2y + r2$

The corresponding equivalent ANFIS is shown in Fig. (1) The node functions in the same layer are of the same function family as described below:

Layer 1: every node i in this layer is a square node with a node function

$$o_i^1 = \mu_{A_i}(x)$$

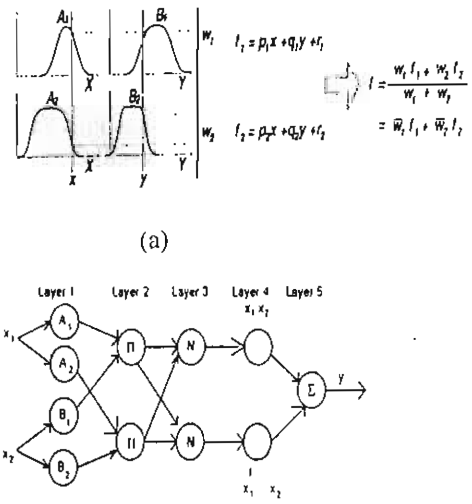
Where x is the input to node i, and A_i is the linguistic label (small, large, etc.) associated with this node function. In other words, o_i^1 is the membership function of A_i and it specifies the degree to which the given x satisfies the quantifier A_i. Usually we choose $\mu_{A_i}(x)$ to be bell-shaped with maximum equal to 1 and minimum equal to 0, such as

$$\mu_{A_i}(x) = \frac{1}{1 + [(\frac{x - c_i}{a_i})^2]^{b_i}}$$

Or

$$\mu_{A_i}(x) = \exp\{ -(\frac{x - c_i}{a_i})^2 \}$$

Where {ai, b_i, c_i} is the parameter set. As the values of these parameters change, the bell-shaped functions vary accordingly, thus exhibiting various forms of membership functions on linguistic label A_i. In fact, any continuous and piecewise differential functions, such as commonly used trapezoidal or triangular-shaped membership functions, are also qualified candidates for node functions in this layer. Parameters in this layer are referred to as premise parameters.



(a)

(b)

Fig.(1). (a) First-order Sugeno fuzzy model; (b) corresponding ANFIS

Layer 2: Every node in this layer is labeled $\prod$ which multiplies the incoming signals and sends the product out. For instance,

$$w_i = \mu_{A_i}(x) \times \mu_{B_i}(y)$$

Each node output represents the firing strength of a rule. (In fact other T-norm operators that perform generalized AND can be used as the node function in this layer.)

Layer 3: Every node in this layer is labeled N. the *ith* node calculates the ratio of the *ith* rule's firing strength to the sum of all rules' firing strengths:

$$\bar{w} = \frac{w_i}{w_1 + w_2}$$

For convenience, outputs of this layer will be called normalized firing strengths.

Layer 4: Every node I in this layer has a node function

$$o_i^4 = \bar{w}_i f_i = \bar{w}_i (p_i x + q_i y + r_i)$$

Where $\overline{w}_i$ is the output of layer 3, and $\{p_i, q_i, r_i\}$ is the parameter set. Parameter in this layer will be referred to as consequent parameters.

Layer 5: The single node in this layer labeled Σ that computes the overall outputs the summation of all incoming signals, i.e.,

$$o_1^5 = \text{Overall output} = \sum_i \overline{w}_i f_i = \frac{\sum_i w_i f_i}{\sum_i w_i}$$

Thus we have constructed an adaptive network that is functionally equivalent to type-3 fuzzy inference system [3.4,6,8].

3. Identification algorithm

System identification consists of three basic sub problems: 1:) Structure specification; 2) Parameter estimation; 3:) Model validation. This is shown in Fig.(2)

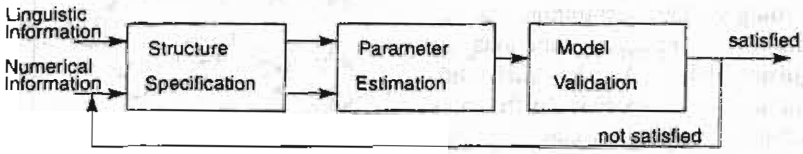


Fig.(2) Fuzzy System Identification Diagram

Structure specification involves finding the important input variables from all possible input variables, specifying membership functions, partitioning the input space, and determines the number of fuzzy rules comprising the underlying model.

Parameter estimation involves the determination of unknown parameters in the model using some optimization method based on both linguistic information obtained from human experts and numerical data obtained from the actual system to be modeled. Structure specification and parameter estimation are interwoven and neither of them can be independently identified without resort to the other.

Model validation involves testing the model based on some performance criteria (e.g. accuracy). If the model cannot pass the test, we must modify the model structure and re-estimate the model parameter. It may be necessary to repeat this process many times before a satisfactory model is found [4,5].

We will discuss each one of these through our algorithm.

3.1 Structure Specification

To determine the structure of the model, we need to determine: 1) number of input to the system 2) the structure of membership function (MFs) 3:) number of rules.

3.1.1 Input selection

The selection of appropriate input variables is a major issue in constructing any model (including ANFIS models). This is because there are usually many factors that can potentially affect the output of the model. However, we only want to include the most relevant factors. Including irrelevant factors not only complicates the task of the model identification but also may lead to misleading interpretations of the model [4]. Therefore, there are varieties of practical and economical reasons for reducing the number of input variables in the final model [7]:

- Remove noise/irrelevant inputs.
- Remove inputs that depend on other inputs.
- Make the underlying model more concise and transparent.
- Reduce the time for model construction.

On the other hand, we do not wish to exclude any important relevant factor. Doing so will obviously limit the accuracy of the model being constructed. The decision as to which of these input variables should be included in the final model is not an easy one. There are several methods for selecting input variables (Forward Selection Procedure, Backward Elimination Procedure, Best Subset Procedure, etc...) discussed in [4]. We will use here the algorithm of Best Subset Procedure and instead of computing Mean Square Error (MSE) we draw Fuzzy Curves. For instance, if we have a modeling problem with 10 candidate input variables and we want to find the most important 3 variables as the inputs of the model, we need to construct $c_3^{10} = 120$ models (each with different combination of 3 input variables) and draw all fuzzy curves of 120 models.

Consider a multiple-input, single-output system for which we have input-output data with possible extraneous inputs. We wish to determine the significant inputs; we call the input candidates $x_i (i = 1,2,...,n)$, and the output variable y. Assume that we have m training data points available and that $x_{ik} (k = 1,2,...,m)$ are the ith coordinate of each of the m training points. The I/O data is shown in Table (1)

No.	x_1	x_2	x_3	y	*No.*	x_1	x_2	x_3	y
1	14.9	2.0	09.3	3.8	11	06.9	2.3	10.3	3.3
2	16.6	1.7	05.8	3.7	12	07.4	1.9	09.4	2.8
3	21.3	2.1	09.1	3.0	13	11.3	2.0	08.4	3.0
4	24.3	2.9	07.0	4.7	14	17.6	2.0	08.2	2.7
5	26.6	1.7	04.8	4.1	15	19.5	2.4	06.5	3.3
6	23.2	1.3	04.7	3.1	16	21.6	2.5	05.1	4.9
7	22.2	2.5	04.5	3.1	17	26.5	1.9	06.3	4.8
8	18.1	2.5	05.9	2.5	18	26.4	2.2	08.1	4.4
9	13.7	2.8	07.9	3.3	19	23.1	3.8	04.9	2.2
10	07.7	2.6	09.3	2.3	20	21.1	2.5	05.1	1.8

Table (1)

Table (1) shows an example with $n = 3$ and $m = 20$, for each input variable x_i, we plot the m data points in $x_i - y$ space. Fig. (3) illustrates the data points from Table (1) in the $x_1 - y, x_2 - y, x_3 - y$ spaces. For every point (x_{ik}, y_k) in $x_i - y$ space, we draw a fuzzy membership function for the input variable x_i defined by:

$$\phi_{ik}(x_i)=\exp(-(\frac{x_{ik}-x_i}{b})^2), k=1,2,3,...,m.$$

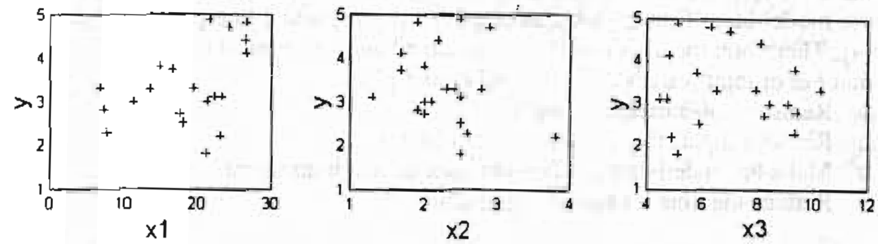


Fig.(3) Data points plotted in x_1-y, x_2-y, x_3-y spaces

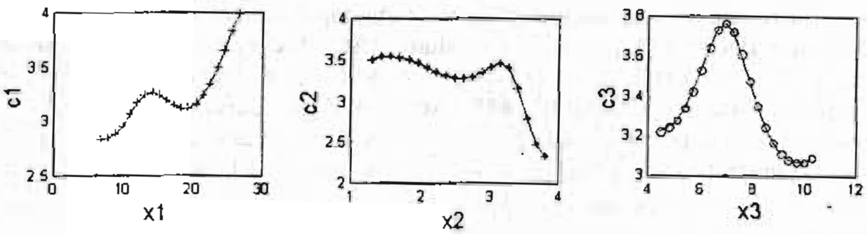


Fig.(4) Fuzzy Curves c_1, c_2, and c_3

Each pair of ϕ_{ik} and the corresponding y_k provide a fuzzy rule for y with respect to x_i. The rule is represented as " if x_i is $\phi_{ik}(x_i)$, then y is y_k." ϕ_{ik} is the input variable fuzzy membership function for x_i corresponding to the data point k. ϕ_{ik} can be any fuzzy membership function, including triangle, trapezoidal, Gaussian, and others. Here we use Gaussian. We typically take b as about 20% of the length of the input interval of x_i. For m training data points, we have m fuzzy rules for each input variable.

Fig.(4) shows the fuzzy membership functions for the points in Fig.(3). We use centroid defuzzification to produce a fuzzy curve c_i, for each input variable x_i by

$$c_i(x_i) = \frac{\sum_{k=1}^{m}\phi_{ik}(x_i).y_k}{\sum_{k=1}^{m}\phi_{ik}(x_i)}$$

If the fuzzy curve [12] for given input is flat *"range between Maximum and minimum is small"*, then this input has little influence on the output data and it is not a significant input. The fuzzy curve tells us that the output is changing when x_i is changing. We rank the importance of the input variables x_i according to the range covered by their fuzzy curves c_i. The ranges of fuzzy curves in Fig.(4) are 1.1535 for c_1, 1.2291 for c_2, and 0.7023 for c_3. Hence, we deduce that x_2 is most significant input, x_1, is the second, and x_3 is third. We will demonstrate these ideas with an example in section 3.

3.1.2 Determining the number of Membership functions

Clustering of numerical data forms the basis of many classification and system modeling algorithms. The purpose of clustering is to identify natural groupings of data from a large data set to produce a concise representation of a system's behavior [11].

Clustering partitions a data set into several groups such that the similarity within a group is larger than that among groups [5]. Achieving such a partitioning requires a similarity metrics that takes two input vectors and returns a value reflecting their similarity. Since most similarity metrics are sensitive to the ranges of elements in the input vectors, each of the input variables must be normalized to within, say, the unit interval 0,1.

There are many techniques [4,5,11] that are used for clustering (Ex. Fuzzy C-Means), we will use here the *Subtractive Clustering*

Consider a collection of n data points $\{x_1,...,x_n\}$ in an M-dimensional space. Since each data point is a candidate for cluster centers, a density measure at data point x_i is defined as

$$D_i = \sum_{j=1}^{n} \exp\left(-\frac{\|x_i - x_j\|^2}{(r_a/2)^2}\right),$$

Where r_a is a positive constant. Hence, a data point will have a high-density value if it has many neighboring data points. The radius r_a defines a neighborhood; data points outside this radius contribute only slightly to the density measure.

After the density measure of each data point has been calculated, the data point with the highest density measure is selected as the first cluster center. Let x_{c1} be the point selected and D_{c1} its density measure. Next, the density measure for each data point x_i is revised by the formula

$$D_i = D_i - D_{c1} \exp\left(-\frac{\|x_i - x_{c1}\|^2}{(r_b/2)^2}\right)$$

Where r_b is a positive constant. Therefore, the data points near the first cluster center x_{c1} will have significantly reduced density measures, thereby making the points unlikely to be selected as the next cluster center. The constant r_b defines a neighborhood that has measurable reductions in density measure. The constant r_b is normally larger than r_a to prevent closely spaced cluster centers; generally r_b is equal to $1.5r_a$ [5]. After the density measure for each data point is revised, the next cluster center x_{c2} is selected and all of the density measures for data points are revised again. This process is repeated until a sufficient number of cluster centers are generated.

When applying subtractive clustering to a set of input-output data, each of the cluster centers represents a prototype that exhibits certain characteristics of the system to be modeled. These cluster centers would be reasonably used as the centers for the fuzzy rules' premise in a zero-order Sugeno fuzzy model [5].

3.1.3 Determining the number of rules

When deciding the number of fuzzy rules for a fuzzy model, there are two contradictory concerns. On one hand, it is desirable that the model includes as many rules as possible so that it minimizes errors from training data with sufficient "patches". On the other hand, it is also desired that the model include as few rules as possible because the generalization capability (Overfitting) of the model decreases as the number of fuzzy rules increases [4]. By generalization capability, we are referring to the model's ability in approximating testing data, which are not used in training the model. With ANFIS the number of rules is dependent on the number of MFs in each variable. For example if we have two inputs of 3 and 4 MFs then the number of the rules is 12 rules.

3.2 Parameter Estimation

By parameter estimation we mean the determination of the shapes, the parameters of MFs that best fit the system to be modeled. This will be satisfied by using the ANFIS learning algorithm.

3.3 Model Validation

Testing whether a given model is appropriate is known as model validation. A suggestive way of comparing two different models is to evaluate their performance when applied to a data set (testing) to which neither of them was trained. We would then favor that model that shows the better performance

4. Application Example

We will use a nonlinear system with two inputs, x_1 and x_2, and a single output, y defined by

$$y = (2 + x_1^{1.5} - 1.5\sin(3x_2))^2 \ , \qquad 0 \le x_1, x_2 \le 3$$

We randomly take 100 points from $0 \le x_1, x_2 \le 3$ and obtain 100 input-output data. To illustrate input variable identification, we add random variables, x_3 and x_4, in the range 0, 3 as dummy inputs [12].

Fig.(5) shows the $x_1 - y, x_2 - y, x_3 - y, x_4 - y$ spaces.

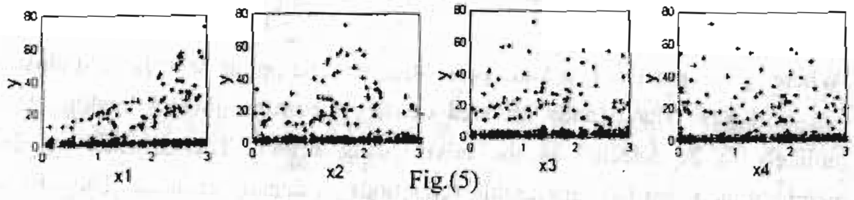


Fig.(5)

To identify the significant input variables, we draw the four fuzzy curves c_1, c_2, c_3 and c_4 for the four input candidates x_1, x_2, x_3, and x_4. The four fuzzy curves are shown in Fig.(6)

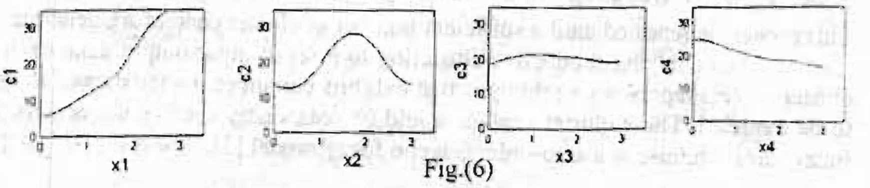


Fig.(6)

From Fig.(6) we find that the ranges of c_i, for c_1 to c_4, are 31.9924, 15.7023, 2.9509, and 8.2536 respectively. From this, we easily and correctly identify x_1 and x_2 as the significant input variables for this system.

Making subtractive clustering of the inputs; Fig.(7) shows that the first input need 3 MFs, and the second input need 4 MFs and so 12 rules are required.

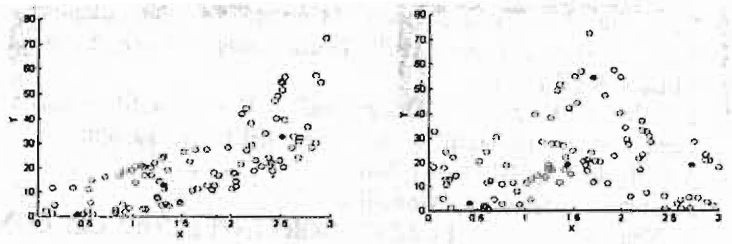


Fig.(7)

Till this point we succeed to determine the structure of the ANFIS; 3 MFs for x_1 and 4MFs for x_2, 12 rules are required. Then we move to the point of parameter estimation. We will train this ANFIS with **hybrid learning**. Fig.(8) shows the ANFIS structure of the system.

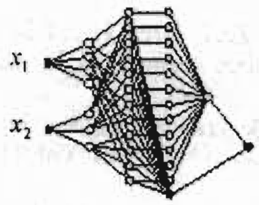


Fig.(8)

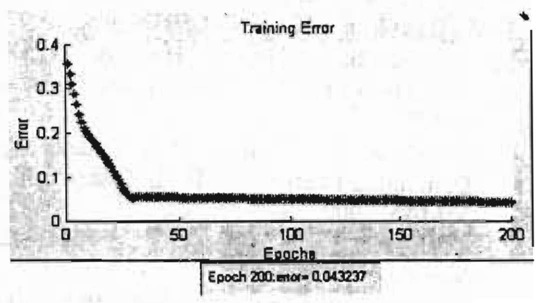


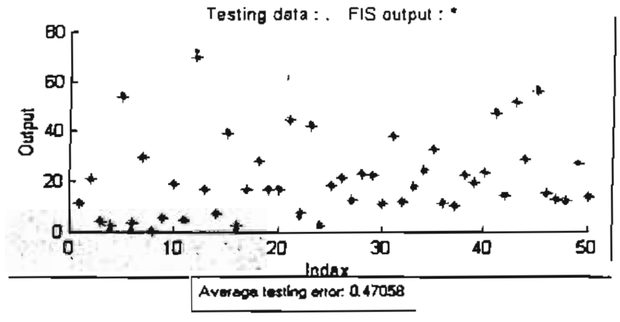
Fig.(9)

And this Fig.(9) shows the learning error after 200 epochs.

5. Analysis of the results

Identification of complex system by this method is almost simpler than previously proposed model, they usually yield better performance, and they train very rapidly.

Fig.(10)



We also note our models are easy to produce. And this can be assured by computing the testing error of the model; this is shown in Fig.(10) . The testing error is 0.047058

References:

1) C.-T. Sun: Rule-Base Structure Identification in an Adaptive-Network-Based Fuzzy Inference System. IEEE Transactions on Fuzzy Systems, Vol. 2, No.1, FEBRUARY 1994
2) I. Rojas, H. Pomares, J. Ortega, and A. Prieto: Self-Organized Fuzzy System Generation from Training Examples. IEEE Transactions on Fuzzy Systems, Vol. 8, No.1, FEBRUARY 2000
3) J. Jantzen: Neurofuzzy Modeling
4) J. Yen, R. Langari: FUZZY LOGIC INTELLIGENCE, CONTROL, AND INFORMATION. NJ: Prentice Hall 1999
5) J.-S. R. Jang, C.-T. Sun, and E. Mizutani: Neuro-Fuzzy and Soft Computing. NJ: Prentice Hall 1997
6) J.-S. R. Jang. ANFIS: Adaptive-Network-based Fuzzy Inference Systems. IEEE Transactions on Systems, Man, and Cybernetics, Vol. 23, No. 3, MAY/JUNE 1993
7) J.-S. R. Jang: Input Selection for ANFIS Learning
8) L. Reznik: Fuzzy Controllers. Newnes 1997
9) M. Lee, S.-Y. Lee, C. H. Park: NEURO-FUZZY IDENTIFIERS AND CONTROLLERS. Journal of Intelligent and Fuzzy Systems, Vol. 2, 1-14 (1994)
10) S. Mitra, Y. Hayashi: Neuro-Fuzzy Rule Generation: Survey in Soft Computing Framework. IEEE Transactions on Neural Networks, Vol. 11, No. 3, MAY 2000
11) T. J. Ross: FUZZY LOGIC WITH ENGINEERING APPLICATIONS. McGraw-Hill 1995
12) Y. Lin, G. Cunningham: A New Approach to Fuzzy-Neural System Modeling. IEEE Transactions on Fuzzy Systems, Vol. 3, No.2, MAY 1995