A modified Algorithm to Model Highly Nonlinear System

Tharwat O. S. Hanafy

Al_Azhar University,Faculty of Engineering, Systems and Computers Department
s_ewiss@yahoo.com

Abstract: In this paper, the Fusion of neural and fuzzy Systems will be investigated. Membership Function Generation and its mapping to Neural Network are introduced. An adaptive network fuzzy inference system (ANFIS) is introduced, and Multiple Inputs /Outputs Systems (Extended ANFIS Algorithm) is implemented. A Modification algorithm of ANFIS, Coupling of ANFIS called coactive neuro fuzzy system (CANFIS), is introduced and implemented using Matlab. The software of the modified algorithm of MIMO model identification is built. To test the validity of the modified algorithm ANFIS (CANFIS algorithm), an example is simulated from the numerical equation. The result of modified algorithm (CANFIS) showed a conformance with the simulated example and the root mean square (RMSE) is very small.

[Tharwat O. S. Hanafy. A modified Algorithm to Model Highly Nonlinear System. Journal of American Science 2010;6(12):747-759]. (ISSN: 1545-1003). http://www.americanscience.org.

Keywords: A modified Algorithm to Model Highly Nonlinear System

1. Introduction:

Fuzzy logic was first developed by Zadeh [1] in the mid-1960s for representing uncertain and imprecise knowledge. It provides an approximate but effective means of describing the behavior of systems that are too complex, ill-defined, or not easily analyzed mathematically. Fuzzy variables are processed using a system called a fuzzy logic controller. It involves fuzzification, fuzzy inference, and defuzzification[2]. An adaptive network fuzzy inference system is investigated. In this paper we will introduce two different learning algorithms. Also, Multiple Inputs /Outputs Systems (Extended ANFIS Algorithm) is implemented using two methods. Modification algorithm, Coupling of ANFIS is called coactive neuro fuzzy system (CANFIS), is introduced and implemented using Matlab. Also, software for modification algorithm for MIMO is built. To test the validity of the modified algorithm ANFIS (CANFIS) algorithm, an example is simulated from the numerical equation[3]. This paper organized seven sections. The first section is introduction. The structure of neuro fuzzy systems is introduced in section 2. The generating fuzzy rules are investigated in section three. Two different ANFIS learning algorithm presented in section four. The modified algorithm (CANFIS) is introduced in section five. Section six introduced conclusion.

2. Neural Fuzzy Systems

In order to process fuzzy rules by neural networks, it is necessary to modify the standard neural network structure appropriately [2]. Since fuzzy systems are universal approximators, it is expected that their equivalent neural network representations will posses the same property. The reason to represent a fuzzy system in terms of neural network is to utilize the learning capability of neural networks to improve performance, such as adaptation of fuzzy systems. Thus, the training algorithm in the modified neural networks should be examined.

2.1 Membership Function Generation

A generalized bell membership function, commonly referred to as bell MF, is characterized by three parameters namely a, b, c.

$$\mu_{Ai}(x_1) = \exp\{-[(\frac{x-c_i}{a_i})^2]^{b_i}\}, \qquad (1)$$

A desired, generalized bell membership function can be obtained with the proper selection of the parameters a, b, c. The parameters a and c represent the width and the center of the bell function, and b represents the slopes at the crossover points. Various other membership functions such as triangular, trapezoidal, Gaussian, and sigmoidal can be used in the formulation of membership functions [3]. The triangular and trapezoidal membership functions, due to their simplicity and computational efficiency, are used extensively in the formulation of membership functions (MF) consist. The Gaussian, the generalized bell function, and the sigmoidal membership functions are smooth and nonlinear functions and are increasingly popular for specifying fuzzy sets. The generalized bell function has one parameter more than the Gaussian membership functions, resulting in an extra degree of freedom to adjust the steepness at the crossover points [4].

The reason to represent a fuzzy system in terms of a neural network is to utilize the learning capability of neural networks to improve performance, such as adaptation of fuzzy systems. To convert Membership Function from FIS structure to construct the same membership using NN layer. The parameter of membership equals to weights and bias in neural network. Fig. (2) Shows the representation of parameters for membership function using NN. Let the bell MF from eq. (1)

Where $a(n) = e^{-n^2}$ Transfer Function (TF) (2)

Let

$$b = 1$$

$$n = \frac{x-c}{a} = \frac{1}{a}x - \frac{c}{a}, \qquad (3)$$

$$\because\ n = wx + bias$$

$$w = \frac{1}{a}, \quad b\text{ias} = -\frac{c}{a}$$

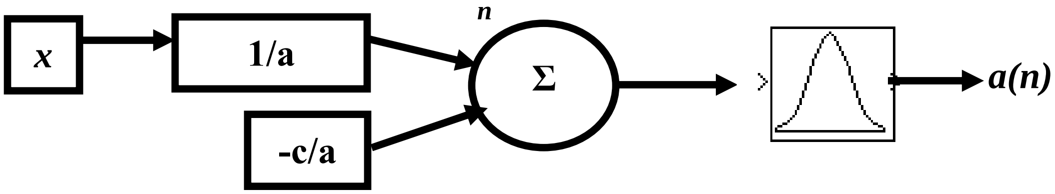


Fig. (2) Representations of membership function using NN

2.2 Adaptive network fuzzy inference systems

To illustrate the use of neural networks for fuzzy inference, we present some successful adaptive neural network fuzzy inference systems, along with training algorithm known as ANFIS. These structures, also known as adaptive neuro fuzzy inference systems, were proposed by Jang [6]. It should be noted that similar structures were also proposed independently by Jang [6]. These structures are useful for control and for many other applications. To fix the ideas, consider the problem of graphically representing the way fuzzy control is achieved in the Sugeno-Takagi model. For a simple example, consider a fuzzy rule base consisting of only two rules [7]:

R_1: if x_1 is $A_1$and x_2 is B_1 then y=$f_1(x)$

R_2: if x_1 is A_2 and x_2 is B_2 then y=$f_2(x)$

(4)

where A_i and B_i are fuzzy sets and

$$f_1(x) = a_0^1 + a_1^1 x_1 + a_2^1 x_2$$

$$f_2(x) = a_0^2 + a_1^2 x_1 + a_2^2 x_2$$

Recall that when numerical input x= (x_1, x_2) is presented, the inference mechanism will produce the numerical output [8]

$$y^* = \frac{A_1(x_1)B_1(x_2)f_1(x) + A_2(x_1)B_2(x_2)f_2(x)}{A_1(x_1)B_1(x_2) + A_2(x_1)B_2(x_2)} \qquad \mathbf{(5)}$$

The ANFIS model suggested by Takagi and Sugeno represents a mathematical tool, which is used to build an ANFIS model of a system. ANFIS model of a non-linear system implication rule contains fuzzy variables with unimodal membership functions. Since such membership functions are linguistically understandable, the fuzzy variables are also called linguistic variables. Takagi and Sugeno's fuzzy model approximates a nonlinear system with a combination of several linear systems by decomposing the input space into several subspaces and representing the input/output relationship, in each subspace, with a linear equation.

3 Generating Fuzzy Rules

Linguistic labels in our natural language convey useful information in human control strategies as well as in other cognitive decision processes. The fuzzy set theory approach to modelling this type of information is based on this thesis that each linguistic label can be represented as a fuzzy subset of an appropriate set U, expressing the semantics of the label. While this seems quit reasonable from a modelling point of view, the concern in applications is determining the membership function of a label. This is related to the more general and more difficult problem of determining rules. There are several approaches to answer this concern. Rules and membership functions can be given by experts, either in a subjective manner or by using some statistical sampling methods. When experts are not available, but instead, numerical experimental data are at hand, it is possible to use neural networks as a solution to the problem of rule and membership function determination. With ANFIS, the structure of the rules and the types of the membership functions are specified in advance, and the parameters of the membership functions are learned from the data [11]. However, rules and membership functions can also be determined by using methods that do not presuppose a rule structure. Both the extraction of rules and the determination of membership functions can be implemented by some kid of clustering. Clustering using neural networks belongs to the domain of unsupervised learning that relies on input data and no corresponding outputs. As in conventional clustering, the goal is to group data points that are similar to each other in some way- that is forming clusters. Given a set of crisp input-output tuples, or training data (x_i, y_i), $i=1,\dots.n$, fuzzy clustering techniques are applied to the input data to determine a collection of fuzzy clusters. Each fuzzy cluster represents one fuzzy "IF….then…"rule, where the fuzzy membership functions in the rule are obtained by projecting the cluster to input and output spaces [2, 6, 9].

In the forward pass of the hybrid learning algorithm, node outputs go forward until layer 4 and the consequent parameters are identified by the least squares method outlined above. In backward pass, the signals that propagate backwards are the error signals and the premise parameters are updated by the gradient descent method.

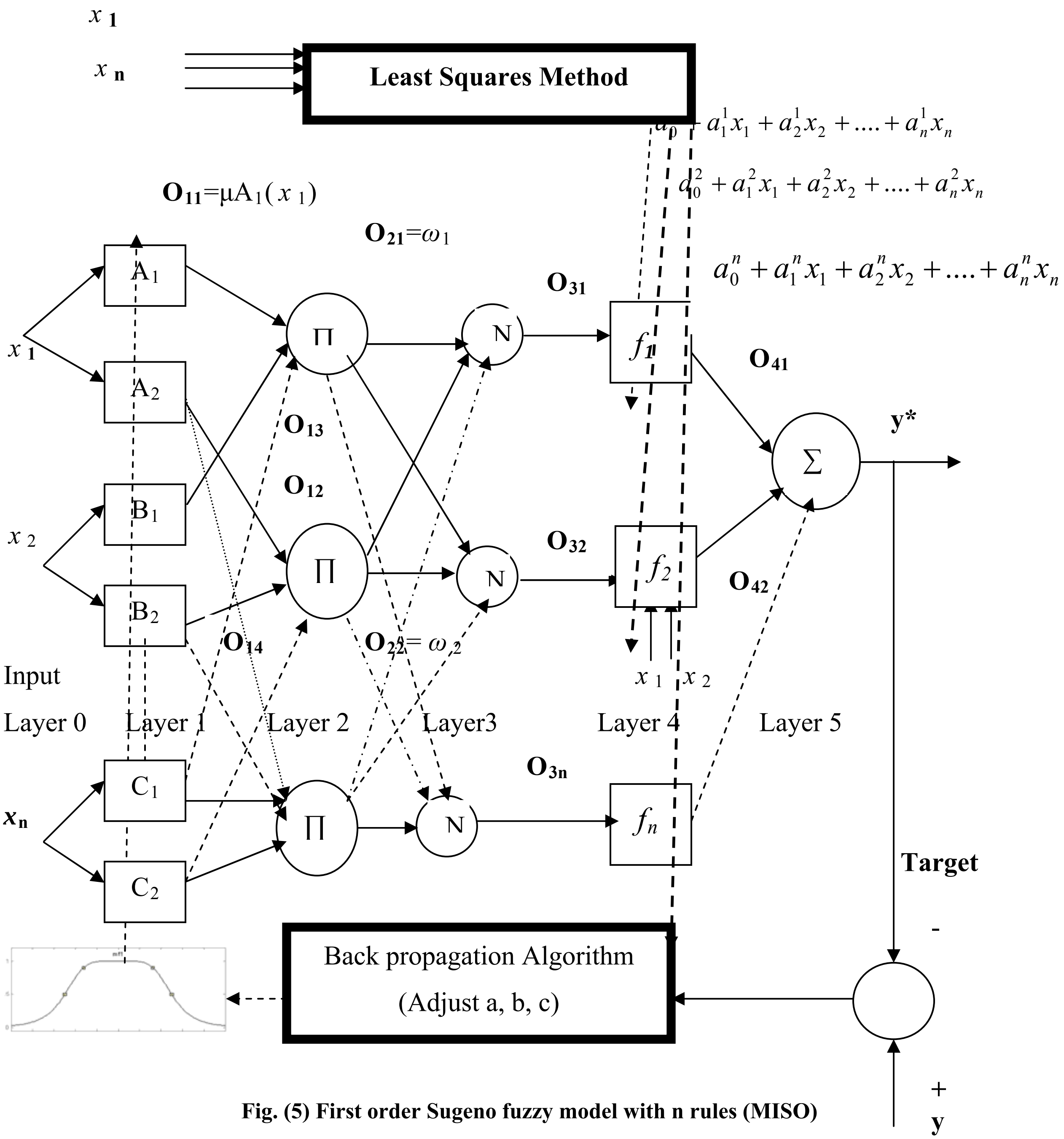


Fig. (5) First order Sugeno fuzzy model with n rules (MISO)

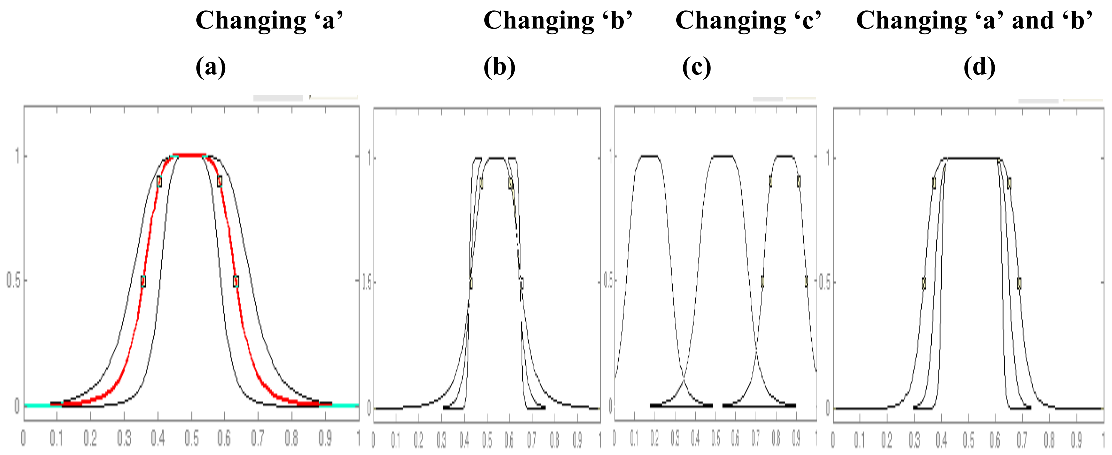


Fig.(6) The effects of changing parameters in Bell MFs: (a)Changing 'a' (b)Changing 'b' (c) Changing 'c' (d) Changing 'a' and 'b'

3.1 Input Space Partitioning

Now it should be clear that the spirit of fuzzy inference systems resembles that of "divide and conquer"- the antecedent of fuzzy rule defines local fuzzy region, while the consequent describes the behavior within the region via various constituents [24]. The consequent constituent can be a consequent MF, a constant value (Zero-order Sugeno Model), or a linear equation (first order Sugeno Model). Different consequent constituents result in different fuzzy inference systems, but their antecedents are always the same. Therefore, the following discussion of methods of partitioning input spaces to form the antecedent of fuzzy rules is applicable to all three types of fuzzy inferences [24].

1- Grid partitioning 2- Tree partitioning3- Scatter partitioning

3.2 Data Clustering Algorithms

Clustering algorithms are used extensively not only to organize and categorize data, but are also useful for data compression and model construction. Two different clustering algorithms are introduced [14]. The first algorithm introduces the most representative of line clustering techniques frequently used in conjunction with radial basis function networks and fuzzy modeling: subtractive clustering. Second, introduces the most representative on line clustering techniques frequently called evolving clustering methods (ECM). The advantages of this method are

1-Creating new cluster through training.

2-New fuzzy rules are created and updated during the operation of the system (through learning). Clustering partitions a data set into several groups such that the similarity within a group is larger than the among groups [15].

Clustering techniques are used in conjunction with radial basis function networks or fuzzy modeling primarily to determine initial locations for radial basis functions or fuzzy if-then rules. For this purpose, clustering techniques are validated on the basis of the following assumptions [16]:

Assumption 1:- States that the target system to be modeled is a smooth input outputs; mapping this is generally true for real world systems.

Assumption 2:- requires the data set to conform to some specific type of distribution; however, this is not always true. Therefore, clustering techniques used for structure identification in neural or fuzzy modeling are highly heuristic, and finding a data set to which clustering techniques cannot be applied is not uncommon.

4 ANFIS learning algorithm

In this section we introduce two different learning algorithms [26].

4.1 The First Learning Algorithm (Hybrid – Learning Algorithm)

From the ANFIS architecture in Fig. (5) We observe that when the values of the premise parameters are fixed, the overall output can be expressed as a linear combination of the consequent parameters [1].. Therefore, the formulas are usually referred to as kalman filter algorithm. In the forward pass of the hybrid learning algorithm, node outputs go forward until layer 4 and the consequent parameters are identified by the least squares method outlined above. In backward pass, the signals that propagate backwards are the error signals and the premise parameters are updated by the gradient descent method [16]**.** Table 3.2 parameters update during the forward and backward passes in the hybrid learning procedure for ANFIS [12].

Table 2. Two passes in the hybrid learning procedure for ANFIS

Signal flow direction	forward pass	backward pass
Consequent parameters	Least-Squares estimator	Fixed
Premise Parameters	Fixed	Gradient descent method
Signals	Node outputs	Error signals

4.2 The Second Learning Algorithm (ANFIS)

The representation in the preceding section of a neural network is simply a graphical display of the computation steps in the sugeno-Tackagi procedure. In order for this representation to be more useful in implementing the model, one needs to equip it with an efficient learning algorithm [16]. In conventional neural networks, the back propagation algorithm is used to learn, or adjust, weights, on connecting arrows between neurons from input-output training samples. In the ANFIS structure, the parameters of the premises and consequents play the role of weights. Specifically, that is, the shape is specified and the function is determined by a finite number of parameters, these parameters are called **premise parameters**, whereas the parameters a$_i$, b$_i$, c$_i$, i=1, 2 in the "then" part of the rules are referred to as **consequent parameters**. The ANFIS learning algorithm consists of adjusting the above set of parameters from sample data $((x_1^k, x_2^k), y^k)$, k=1,…,N.

It is important to keep in mind that when we develop a set of prototype fuzzy rules, we are in fact representing possibly nonlinear input-output relationships [16]. The effectiveness of fuzzy models representing nonlinear input-output relationships depends on the membership functions, is an important issue in fuzzy modelling [16]. This tuning task can be viewed as an optimization problem; neural network offer a possibility to solve this problem [3]. In order to train a fuzzy neural network, we need a set of training data in the form of input – output tuples, and a specification of the rules, including a preliminary definition of the corresponding membership functions [4]. A standard approach is to assume a certain shape for the membership functions depend on parameters that can be learned by the neural network. We describe one method for learning membership functions of the antecedent and consequent parts of fuzzy "If….then…" rules. Suppose an unknown function, to be realized by a fuzzy inference system is known only through the training set [16]

$$\{(x^1, y^1),\ldots,(x^k, y^k)\} \qquad (22)$$

where $x^k =((x_1^k,\ldots,x_n^k) \in R.$ and, $y^k \in R$. To model the unknown function, we use fuzzy "If….then…" rules R_i, *i=1,…., m,* of the following type

R_i : If x_1^k *is* A_i^1 and … and x_n^k is A_i^n then

$$y= \sum_{j=1}^{n} z_i^j x_j^k + z_i \qquad (23)$$

where A_i^j are fuzzy membership functions and z_i^j are real numbers.

5 Multiple Input /Output Systems (Extended ANFIS Algorithm)

One way to get multiple outputs is to place, as many ANFIS models side by side as there are required outputs. In this MANFIS (multiple ANFIS) no modifiable parameters set of fuzzy a rule, which makes it difficult to realize possible certain correlation between outputs [24]. An additional concern resides in the number of adjustable parameters, which drastically increases as outputs increase. This approach is represented in Fig. (7) and Fig. (8). another way of generating multiple outputs is to maintain the same antecedents of fuzzy rules among multiple CANFIS models. In short, fuzzy rules are constructed with shared membership values to express correlations between outputs. In this thesis we investigate the Second approach CANFIS to produce multiple outputs. This model is illustrated in Fig.(9).

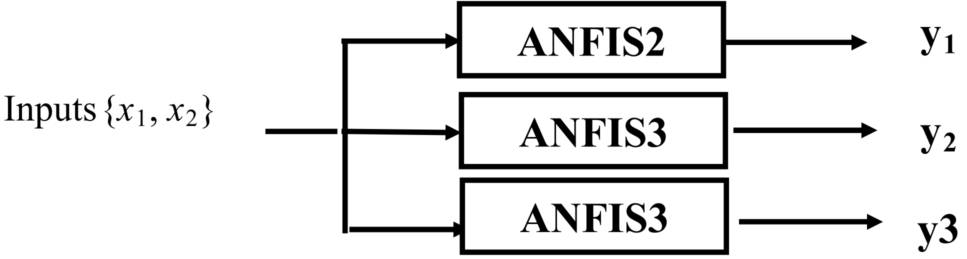


Fig (7). MANFIS Structure Decoupling MANFIS Structure

In this work we convert Fuzzy inference system structure to neural network structure, such as, membership functions for input variables, membership functions for outputs variables, Rules base.

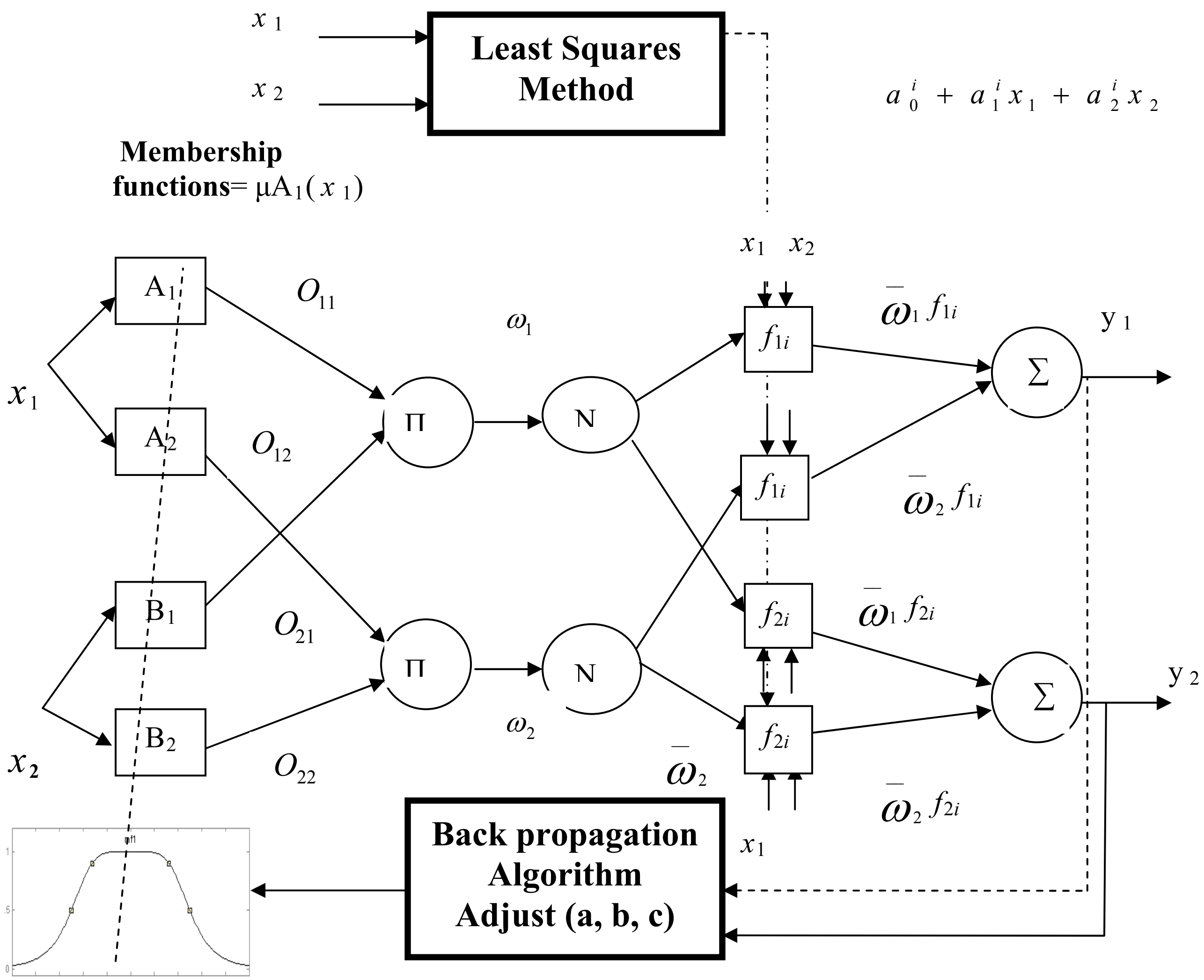


Fig. (8) Two outputs CANFIS Architecture with two rules per outputs

After conversion of FIS structure to NN, we train fuzzy neural network (FNN) and compute weights of each layer, weights of inputs, and weights of outputs to generate model. Then the weights are transformed to membership function parameters to construct FIS structure. The layer structure of CANFIS is defined as follows [2, 24, 26].

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

$$O_{1,i} = \mu_{Ai}(x_1), \qquad \text{For } i\text{=1, 2,} \qquad \text{or}$$

$$O_{1,i} = \mu_{Bi-2}(x_2) \qquad \text{For } i\text{=3, 4} \qquad (33)$$

where x_1 (or x_2) is the input to node i, and A_i or(B_{i-2})is a linguistic label, such as (small, large, etc.) associated with this node. In other words, O_i^1 is the membership grade of a fuzzy set A=(=A_1, A_2, B_1 or B_2) and it specifies the degree to which the given x_1 or x_2 satisfies the quantifier A. Here the membership function for A can be any appropriate parameterized membership function such as the generalized bell function:

$$\mu A_i(x_1) = \frac{1}{1+\left|\frac{x-c_i}{a_i}\right|^{2b}} \qquad (34)$$

Or

$$\mu A_i(x_1) = \exp\{-[(\frac{x-c_i}{a_i})^2]^{b_i}\}, \quad (35)$$

where {a$_i$, 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 for fuzzy set A. In fact, any continuous and piecewise differentiable functions, such as commonly used trapezoidal or triangular-shaped membership functions, are also qualified candidates for node functions in this layer. In this thesis we convert equation (35) to NN structure. Parameters in this layer are referred as premise parameters.

Layer 2: Every node in this layer is a circle node labeled п, which multiplies the incoming, signals and sends the product out. For instance,

$$\omega_i = \mu A_i(x_1) \times \mu B_i(x_2) \qquad i = 1,2. \quad (36)$$

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

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

$$\varpi_i = \frac{\omega_i}{\omega_1 + \omega_2}, \quad \text{i=1, 2} \qquad (37)$$

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

Layer 4: Every node *i* in this layer is a square node with a node function

$$o_i^4 = \varpi_i f_i = \varpi_i(a_0^1 + a_1^1 x_1 + a_2^1), \qquad (38)$$

Where ϖ_i is the output of layer 3, and (a_{0i}, a_{1i}, a_{2i}) is the parameter set. Parameters in this layer will be referred to as consequent parameters. $f_{ji} = a_0^i + a_1^i x_1 + a_2^i x_2$, *i=1,2,..n* where *n* is number of rule, *j* number of *output*

Layer 5: The single node in this layer is a circle node labeled $\sum$ that computes the overall output as the summation of all incoming signals, i.e.,

$$o_j^5 = overall\text{outputj} = \sum_i \varpi_i f_{j_i} = \frac{\sum \omega_i f_{ji}}{\sum_i \omega_i}$$

where *j=1, 2, 3* (39)

Thus we have constructed an adaptive network, which is functionally equivalent to a type-3 fuzzy inference system as shown in Fig. (5) from chapter 2. Fig. (10) Shows the NN layers corresponding to FIS of multiple inputs three outputs system.

The structure of NN layer for MIMO (2 i/p-3 o/p) Simulink Model of Fuzzy Neural Network is introduced. We add layer 1 for distribution. We use layer 1 to distribute inputs; one input per one membership function. Also, we add sub layer in the consequent parameter for layer 5 to compute multiple outputs. Layer 6 is the overall outputs. Table 3.2 Present the comparison between ANFIS and CANFIS

Table 3. Comparison between ANFIS and CANFIS

	Layer 1	Layer 2	Layer 3	Layer 4	Layer 5	Layer 6
ANFIS	------	Convert Crisp To member ship value	Product	Normalization	Adaptive node	Over all outputs
CANFIS	Distribution	Convert Crisp To membership value	Product	Normalization	Add sub Layer for rule sharing	Over all outputs

6 Procedure of Neuro fuzzy System Identification

In this section we introduce the procedure to construct Neuro fuzzy model. Fig. (13) Presents the preprocessing data Using Subtractive clustering Data. The Flow chart of learning ANFIS model is introduced in Fig. (12).

7- Testing of the Modified Algorithm (CANFIS)

To test the validity of the modified algorithm ANFIS (CANFIS) algorithm, an example is simulated from the numerical equation. The result of modified algorithm (CANFIS) showed a conformance with the simulated example and the root mean square (RMSE) is in the range of 0.00248735. This section presents the simulation results of the proposed CANFIS with off line learning. In this example, CANFIS is used to model highly nonlinear functions and the result is discussed as follows. Modeling (2) two inputs (3) three outputs nonlinear functions, in this example we consider using CANFIS to model a nonlinear sin-cos equations.

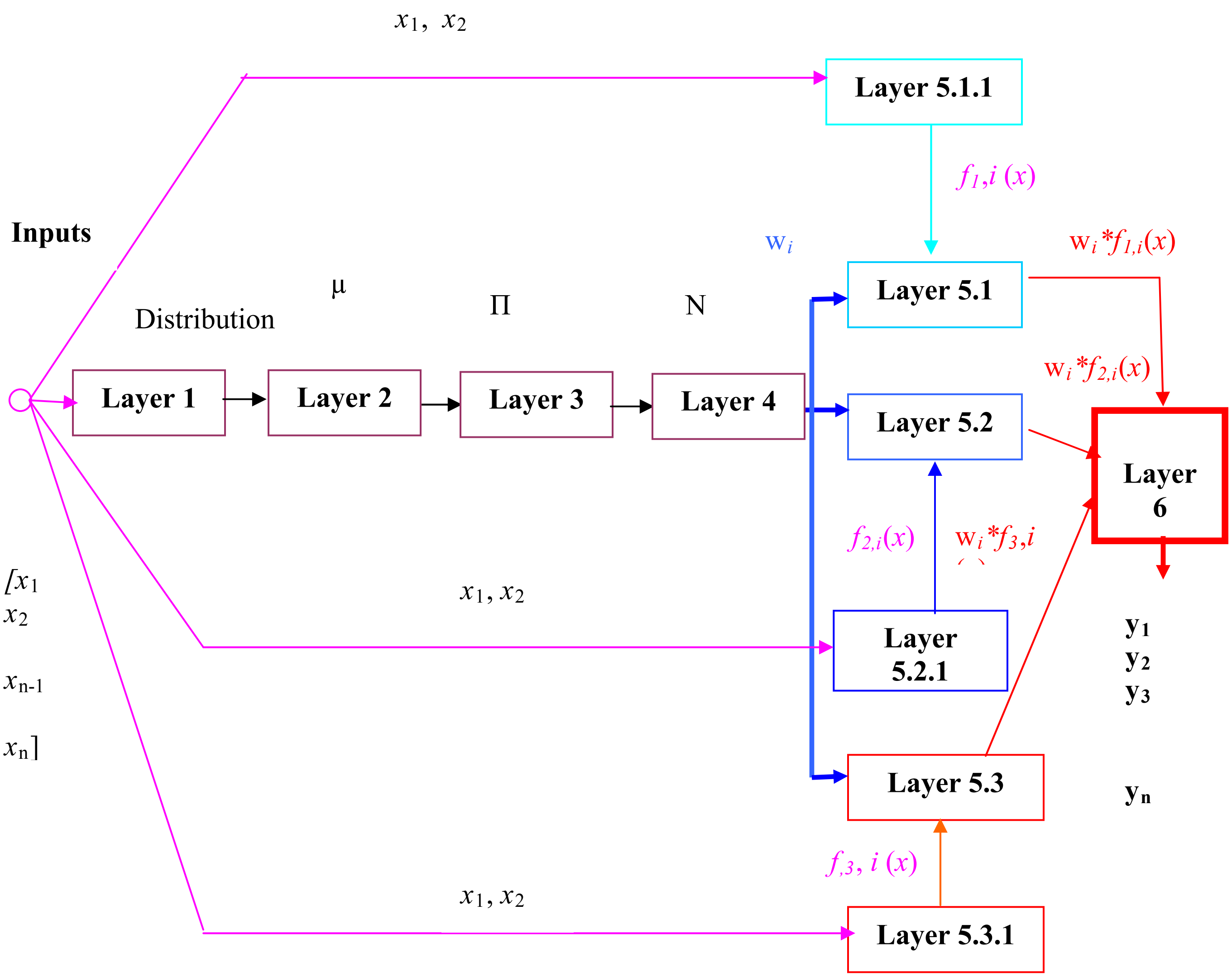


Fig. (10) NN Implementation to FIS

The following three equations are presented to model 2 two inputs (3) three outputs using CANFIS. Fig. (14) Shows the relationship between inputs and outputs before training (above figure) after training (below figure) for first output. Fig. (15) Shows the relationship between inputs and outputs before training (above figure) after training (below figure) for second output. Fig. (16) Shows the relationship between inputs and outputs before training (above figure) after training (below figure) for third output where x_1, x_2=sample number, period time=0.01 second.

$$y_1 = (\sin(\frac{x_1 * \pi}{180}))'*\sin(\frac{x_2 * \pi}{180}); \qquad (40)$$

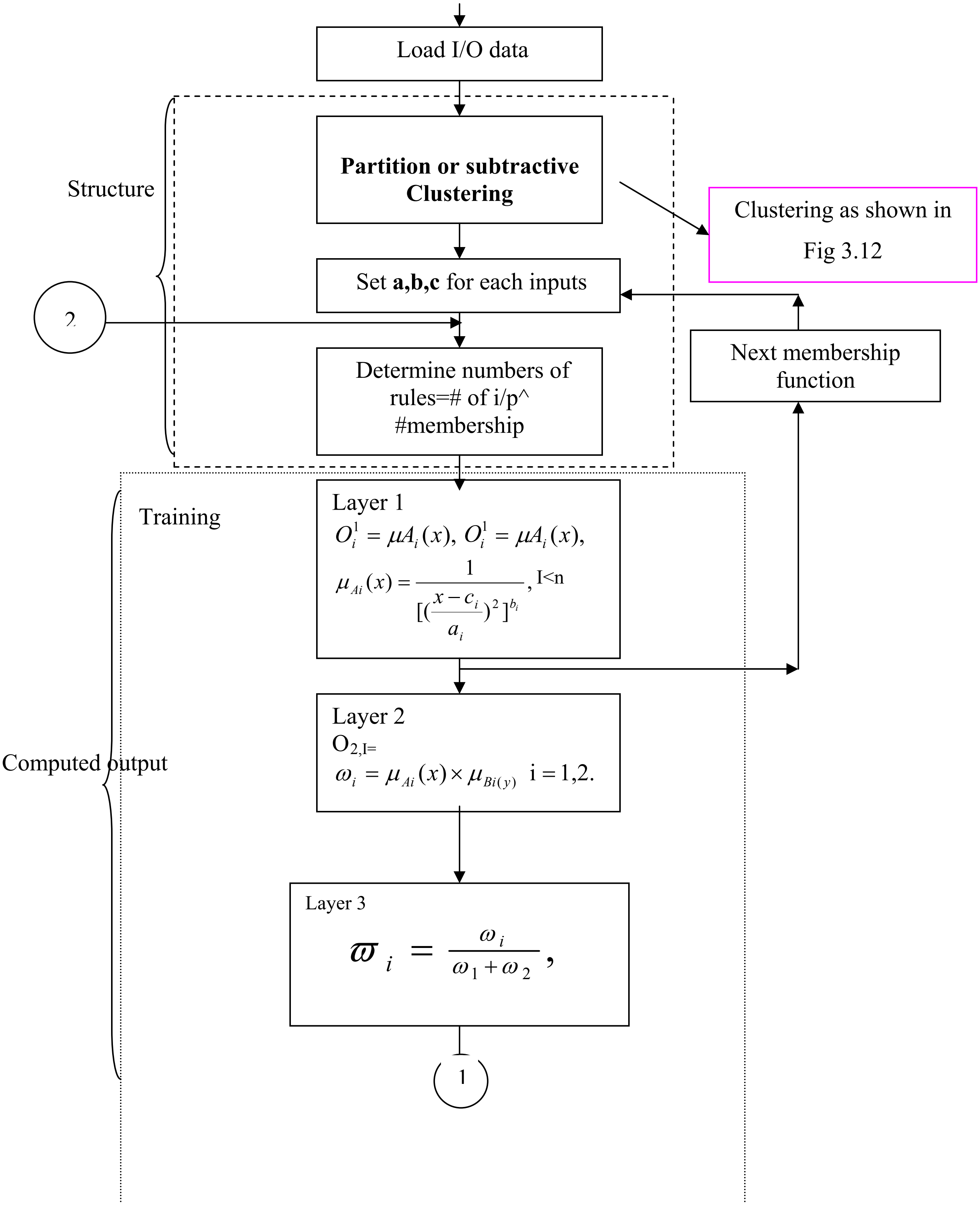
$$y_2 = (\cos(\frac{x_1 * \pi}{180}))'*\cos(\frac{x_2 * \pi}{180}); \qquad (41)$$

$$y_3 = (\sin(\frac{x_1 * \pi}{180}))'*\cos(\frac{x_2 * \pi}{180}); \qquad (42)$$

e$_i$= $y_j - \hat{y}_j$, The root mean square error for model is given by

$$\text{RMSE} = (\frac{1}{N}\sum_{1}^{n} e_i^{\,2})^{\frac{1}{2}} \qquad i=1,2,.........N,\ j=1,\ 2,\ 3\ (43)$$

Where N is number of samples, y$_j$ is the output of numerical equation; $\hat{y}_j$ is the actual output of CANFIS Model. Mean square error of modified algorithm is equal to (MSE) =0.00248735, after execution of program (TestFIS) which is found in the CD of programs. The RMSE is computed from equation (43). This result shows very good performance of the modified algorithm.


Load I/O data
Structure
Partition or subtractive Clustering
Clustering as shown in Fig 3.12
Set a,b,c for each inputs
2
Next membership function
Determine numbers of rules=# of i/p^ #membership
Training
Layer 1
$O_i^1 = \mu A_i(x),\ O_i^1 = \mu A_i(x),$
$\mu_{Ai}(x) = \frac{1}{[(\frac{x-c_i}{a_i})^2]^{b_i}},$ I<n
Computed output
Layer 2
O2,I=
$\omega_i = \mu_{Ai}(x) \times \mu_{Bi(y)}$ $i = 1,2.$
Layer 3
$\varpi_i = \frac{\omega_i}{\omega_1 + \omega_2},$
1

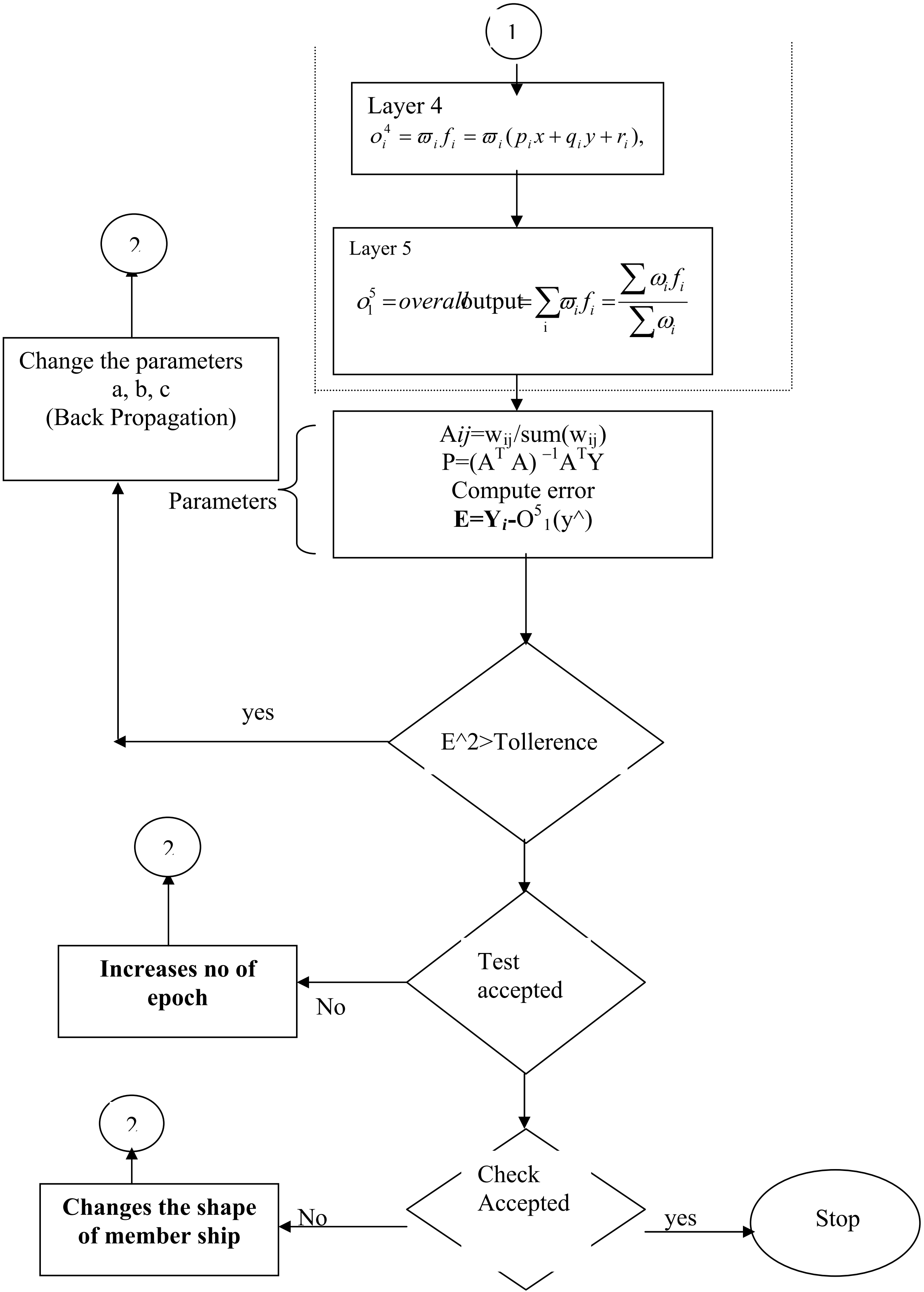


Fig. (12) The Flow chart of learning ANFIS Model

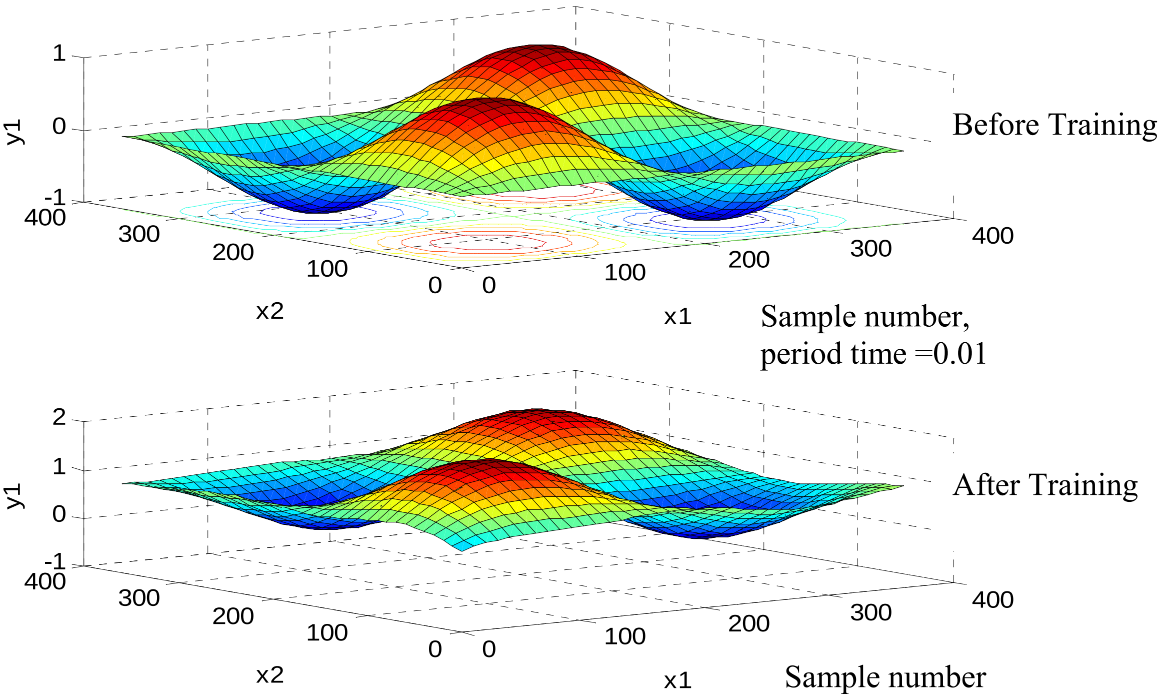


Fig. (14) System Response of CANFIS after Training for 2 inputs three outputs for first output

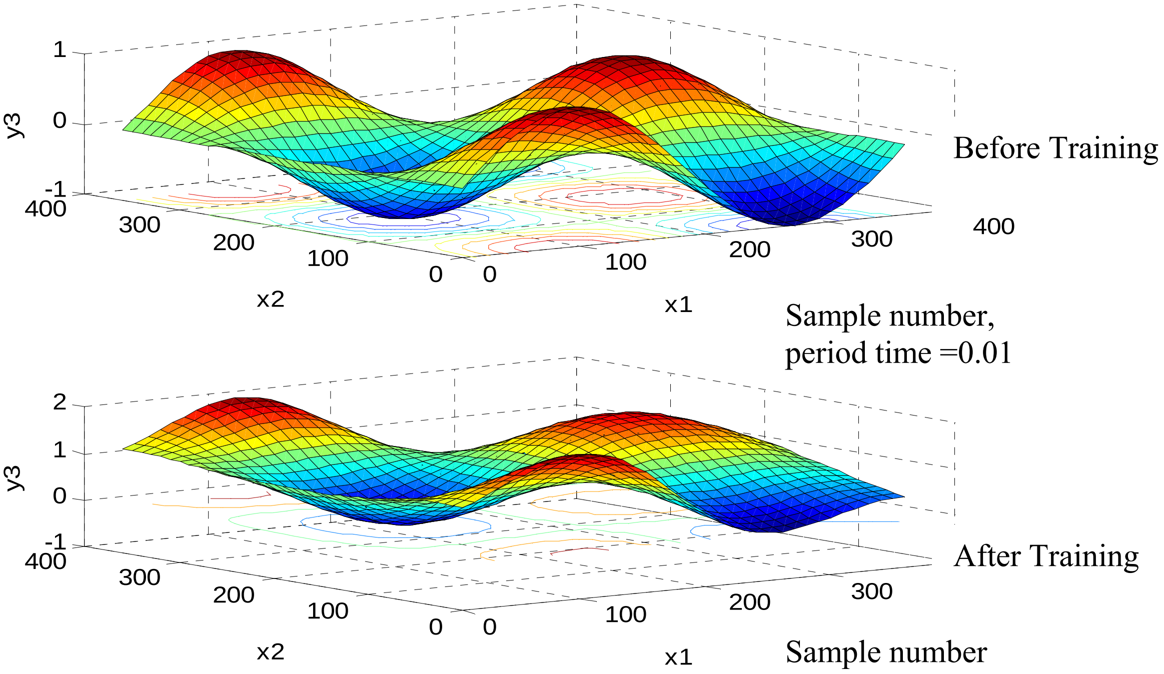


Fig. (16) System Response of CANFIS after Training for 2 inputs three outputs for third output

8 Conclusions

In this paper, the principles of neuro fuzzy Systems have been investigated. Membership Function Generation and its mapping to Neural Network are introduced. An adaptive network fuzzy inference system (ANFIS) is introduced, and Multiple Inputs /Outputs Systems (Extended ANFIS Algorithm) is implemented. A Modification algorithm of ANFIS, Coupling of ANFIS called coactive neuro fuzzy system (CANFIS), is introduced and implemented using Matlab. The software of the modified algorithm of MIMO model identification is built.

To test the validity of the modified algorithm ANFIS (CANFIS algorithm), an example is simulated from the numerical equation. The result of modified algorithm (CANFIS) showed a conformance with the simulated example and the root mean square (RMSE) is in the range of 0.00248735.

Corresponding author

Dr. Eng Tharwat O. S. Hanafy
Al_Azhar University, Faculty of Engineering, Systems and Computers Eng. Department, Cairo, Egypt
s_ewiss@yahoo.com

References

1. Hossein Salehfar, Nagy B., Jun Huang, “Systematic Approach to Linguistic Fuzzy Modeling Based on Input-Output Data ", proceeding of the 2000 winter Simulation Conference.
2. Jang J.-S. R. “ANFIS: Adaptive Network-based Fuzzy Inference System”, IEEE Transactions on Systems, Man and Cybernetics, (1993), 23 (3), 665-685.
3. E. Kim, M. Park, S. Ji, and M. Park, “A new approach to fuzzy modeling”, IEEE Trans. on Fuzzy Syst., vol. 5, no. 3, pp.328-337, 1997.
4. Amine Tabelsi, Frederic, “Identification of Nonlinear Multivariable Systems by Adaptive Fuzzy Takagi-Sugeno Model”, International Journal of Computational Cognition, September 2004, page 137-153.
5. Azeem M. F., Hanmandlu M. and Ahmad N. “Generalization of adaptive neuro-fuzzy inference system”, IEEE Transactions on Neural Networks, 11(6), 1332-1346. 2000
6. Kim J. and Kabasov N. “Adaptive Neuro-Fuzzy inference systems and their application to nonlinear dynamical systems”, IEEE Transactions on Neural Networks, (1999), 12 (9), 1301-1319.
7. Lin C.-T., “A neural fuzzy control scheme with structure and parameter learning”, Fuzzy Sets and Systems, (1995), 70, 183-212.
8. M. Sugeno and T. Yasukawa, “A fuzzy-logic-based approach to qualitative modeling”, IEEE Trans. on Fuzzy Systems, vol. 1, no. 1, pp. 7-31, 1993
9. Mackey M. C. and Glass L. “Oscillation and chaos in physiological control systems”, Science, 1997, 287-289.
10. Mamdani, E.H. "applications of fuzzy algorithms for control simple dynamic plants, Proc. of the IEE, (1984), 121, 1585-1588.
11. Nauck D. and Kruse R., “Neuro-fuzzy systems for function approximation”, Fuzzy Sets and Systems, 1999, 101, 261-271.
12. Pal K. and N. R. Pal. “A neuro-fuzzy system for infringing”, International Journal of Intelligent and Fuzzy Systems, 1999, 14(11), 1155-1182.
13. Ann L Casebeer and Marja J Verhoef, " Combining Qualitative and Quantitative Research Methods: Considering the Possibilities for Enhancing the Study of Chronic Diseases", Volume 18, No.3 -1997
14. Paiva R. P., “Neuro-Fuzzy Identification: Interpretability Issues”, MSc Thesis, Department of Informatics Engineering, Faculty of Sciences and Technology, University of Coimbra, Portugal, 1999.
15. Paiva R. P., Dourado A. and Duarte B. “Applying subtractive clustering for neuro-fuzzy modeling of a bleaching plant”, Proceedings of the European Control Conference -1999, ECC’99, CD-ROM.
16. Jyh-Shing, Roger Jang “Neuro Fuzzy Modeling and Control”, proceeding of the IEEE, March 1995.
17. R. Fuller, “Introduction to Neuro-Fuzzy Systems, Advances in Soft Computing Series”, Springer-Verlag, Berlin, 1999.
18. Chiu S. L. “Fuzzy model identification based on cluster estimation”, Journal of Intelligent and Fuzzy Systems, 1994, 2 (3), 267-278
19. Dave R. N. and Krish napuram R. “Robust clustering methods: a unified view”, IEEE Transactions on Fuzzy Systems, 1997, Vol. 5 No. 2, 270-293.
20. Robert Shorten “On The Interpretation And Identification of Dynamic Takgi-Sugeno Fuzzy Models”. IEEE Trans. on Fuzzy Systems, Vol. 8, No 3, Jun 2000.
21. R. Babuska J.A. Roubos H.B. Verbruggen, “Identification of MIMO systems by input -output TS fuzzy models” Delft University of Technology, Department of Electrical Engineering, Control Laboratory-Mekelweg, 2004, P.O. Box 5031, 2600 GA Delft, The Netherlands.
22. Carmona, P. Castro, J.L. Zurita, J.M, “FRIWE: Fuzzy Rule Identification with Exceptions”, IEEE

Transactions on Fuzzy Systems, Publication Date: Feb. 2004 on page(s):140-151Volume: 12.

23. Yu, W. Li, X., "Fuzzy Identification Using Fuzzy Neural Networks With Stable Learning Algorithms", IEEE Transactions on Fuzzy Systems, June 2004 On page(s): 411-420 Volume: 12, Issue:3, ISSN: 1063-6706.
24. Jang R, Sun C T and Mizutani E., "Neuro-fuzzy and soft computing: A computational Approach to Learning and Machine Intelligence", Prentice Hall NJ, 1997.
25. Robert Babuska, "Neuro fuzzy methods for modeling and identification", Delft university of technology, faculty of information technology and systems, Recent advances in intelligent Paradigms and applications, Springer- 2002.
26. Hung T. R. Prasad, Carol L. "First Course in Fuzzy and Neural Control", @2003 by Chapman & Hall/CRC.
27. Ho Jae Lee, Hagbae Kim, Young Hoon Joo, Wook Chang, Jin Bae Park, "A new intelligent digital redesign for T-S fuzzy systems: global approach" IEEE Transactions on Fuzzy Systems" April 2004 On page(s): 274 284,Volume:12,Issue:2ISSN: 1063-6706.
28. Jyh-Shung, Roger Jang, 1995, "Neuro Fuzzy Modeling and Control", proceeding of the IEEE, March 1995.

10/10/2010