ADAPTIVE NEURO-FUZZY INFERENCE SYSTEM WITH SECOND ORDER SUGENO CONSEQUENTS

Mohanad Alata,[*] *Hisham Moaqet*[†]

Abstract: Adaptive Neuro-Fuzzy Inference System (ANFIS) with first order Sugeno consequent is used widely in modeling applications. Though it has the advantage of giving good modeling results in many cases, it is not capable of modeling highly non-linear systems with high accuracy. In this paper, an efficient way for using ANFIS with Sugeno second order consequents is presented. Better approximation capability of Sugeno second order consequents compared to lower order Sugeno consequents is shown. Subtractive clustering is used to determine the number and type of membership functions. A hybrid-learning algorithm that combines the gradient descent method and the least squares estimate is then used to update the parameters of the proposed Second Order Sugeno-ANFIS (SOS-ANFIS). Simulation of the proposed SOS-ANFIS for two examples shows better results than that of lower order Sugeno consequents. The proposed SOS-ANFIS shows better initial error, better convergence, quicker convergence and much better final error value.

Key words: *ANFIS, Subtractive clustering, Sugeno fuzzy inference systems, Fuzzy modeling*

Received: March 27, 2007
Revised and accepted: June 11, 2007

1. Introduction

Jang [1] presented the architecture and learning procedure underlying Adaptive Neuro-Fuzzy Inference System, which is a fuzzy inference system implemented in the framework of adaptive networks. By using a hybrid learning procedure, ANFIS can construct an input-output mapping based on both human knowledge (in the form of fuzzy if-then rules) and stipulated input-output data pairs which can be

[*]Mohanad Alata – Corresponding author
Assistant professor, Mechanical Engineering Department, Jordan University of Science and Technology, Irbid P.O. Box (3030), Jordan, E-mail: `alata@just.edu.jo`
[†]Hisham Moaqet – Graduate student
Mechanical Engineering Department, Jordan University of Science and Technology, Irbid P.O. Box (3030), Jordan

employed to model nonlinear functions and identify nonlinear components in a control system yielding remarkable results.

Since its presentation, ANFIS was implemented in many applications [2-7]. ANFIS implements a first order or a zero order Sugeno fuzzy system in an architecture composed of five layers. Layer 1 consists of membership functions described by generalized bell function. Layer 2 implements the fuzzy AND operator, while layer 3 acts to scale or normalize the firing strengths. The output of the fourth layer is a linear combination of the inputs multiplied by the normalized firing strength. Layer 5 is a simple summation of the outputs of layer 4. Layer 1 contains premise modifiable parameters, and layer 4 contains consequent parameters. A least squares estimator in the forward pass identifies the consequent parameters. In the backward pass, the consequent parameters are held fixed, and the premise parameters are modified using gradient descent. The user specified information is the number of membership functions for each input, the membership type, and the input-output training information.

Chiu [8] has presented an efficient method for estimating cluster centers of numerical data. It can be used to determine the number of clusters and their initial values for initializing iterative optimization-based clustering. It forms a basis of a fast and robust algorithm for identifying fuzzy models in which the number of membership functions and type can be generated automatically.

Alata et al. [9] and K. Demirli and P. Muthkumaran [10] showed that higher order Sugeno Consequent models compared with lower order Sugeno models could identify systems with less error for the same number of rules. It also might achieve the required performance with fewer rules due to better approximation capability of higher order Sugeno consequent models. This important property of Sugeno fuzzy systems is used in this paper to extend the original ANFIS to work with second order Sugeno consequents for one input and two input systems. Also, effective subtractive clustering technique is used to determine the number and type of membership functions for each input rather than being user specified which would result in better performance and less approximation error through starting from optimal initial membership functions generated by the clustering algorithm.

After building a fuzzy inference system using subtractive clustering and Sugeno second order consequents, a hybrid learning algorithm that combines gradient descent method and the least square estimate is then used to train the generated fuzzy inference system and update the proposed ANFIS parameters. Each epoch of this hybrid learning procedure is composed of a forward pass and a backward pass. In the forward pass, the input data is supplied and functional signals go forward to calculate each node output and the second order Sugeno consequent parameters are updated. The functional signals keep going forward till the error measure is calculated. In the backward pass, the error rates are calculated and the parameters of input membership functions are updated by gradient descent method.

2. Subtractive Clustering

Chiu [8] proposed subtractive clustering technique with improved computational effort, in which the data points themselves are considered as candidates for cluster centers instead of grid points. By using this method, the computation is simply proportional to the number of data points and independent of the dimension of the problem. In this method also, a data point with the highest potential which is a function of the distance measure, is considered as a cluster center and data points close to new cluster center are penalized in order to control the emergence of new cluster centers.

Considering a collection of n data points $(x_1, x_2, \ldots, x_n)$ in an M-dimensional space, subtractive clustering for the these data points can be performed as follows:

1. We consider each data point as a potential cluster center and define a measure of potential of data point x_i as

$$P_i = \sum_{j=1}^{n} e^{-\alpha \|x_i - x_j\|^2}, \tag{1}$$

where:

$$\alpha = \frac{4}{r_a^2} \tag{2}$$

and r_a is a positive constant defining the neighborhood range of the cluster or simply the radius of hyper sphere cluster in the data space (this radius also represents the support of fuzzy sets representing the cluster), n is the total number of data points, and x_i, x_j are vectors in the combined data space of input and output dimensions. The potential of a data point is a function of its distances to all data points including it. The higher number of neighborhood data points results in higher potential values. After calculating the potential of each data point, the data point with the highest potential is selected as the first cluster center.

2. The identification of other cluster centers is carried out through the subtraction process. In this process the potential of all data points is revised each time a new cluster center is obtained, by using the following equation:

$$P_i \quad \Leftarrow \quad P_i \quad - P_k^* \zeta \tag{3}$$

where:

$$\zeta = e^{-\beta \|x_i - x_k^*\|^2} \tag{4}$$

$$\beta = \frac{4}{r_b^2} \tag{5}$$

$$r_b = \eta * r_a \tag{6}$$

and where x_k^* and p_k^* are position and potential of k^{th} cluster center, respectively, r_b is a positive constant representing the radius of penalizing zone for a given cluster center, η is called a squash factor and is also a positive constant. Chiu [8] proposed r_b to be somewhat greater than r_a in order to avoid obtaining closely spaced cluster centers.

After revising the potential of all the clusters after subtraction, a cluster center is selected based on its new potential value in relation to an upper acceptance threshold $\bar{\varepsilon}$ called accept ratio, lower rejection threshold $\underline{\varepsilon}$ called reject ratio, and a relative distance criterion. The acceptance of a data point with the potential between the upper and the lower thresholds depends on the smallest of the distances between that point and all other previously found cluster centers as given by the following equation:

$$\frac{d_{\min}}{r_a} + \frac{P_k^*}{P_1^*} \tag{7}$$

where d_{min} is the smallest of the distances between x_k^* and all previously found cluster centers. This will avoid the emerging of new clusters close to the existing ones even though they have the potential higher than the lower threshold enhancing the uniform representation of a system in entire data space. Chiu [8] proposed the optimum default values of 1.5, 0.5 and 0.15 for η, $\bar{\varepsilon}$, and $\underline{\varepsilon}$, respectively.

Subtractive clustering is used to generate membership functions in the proposed SOS-ANFIS. Using subtractive clustering is a better way to determine the number and type of membership functions than being user specified because the use of subtractive clustering gives the benefit of automatic generation of membership functions and rules and hence it gives an optimal solution if its parameters have been chosen properly using the enumerative search technique [10]. On the other hand, letting the number and type of input membership functions to be user specified will result in an under fitted fuzzy inference system (FIS) or over-fitted FIS which is computationally expensive.

3. Second Order Sugeno Fuzzy Inference System

When the cluster estimation method is applied to a collection of input/output data points, each cluster center is in essence a prototypical data point that exemplifies a characteristic behavior of the system. Hence, each cluster center can be used as the basis of a rule that describes the system behavior. The initial fuzzy inference system proposed in this paper is built using subtractive clustering and second order Sugeno consequents. Model identification for fuzzy inference systems with input membership functions generated by subtractive clustering and second order Sugeno consequents is derived below.

Consider a set of c cluster centers (x_1^*, x_2^*, …, x_c^*) in an M dimensional space. Let the first N dimensions correspond to input variables and the last $M - N$ dimensions correspond to output variables. We decompose each vector x_i^* into two components vector y_i^* and z_i^*, where y_i^* contains the first N elements of x_i^* (i.e. the coordinates of the cluster center in the input space).

Each cluster center x_i^* is considered as a fuzzy rule that describes the system behavior. Given an input vector y, the degree to which rule i fulfilled is defined as:

$$\mu_i = e^{-\alpha \|y - y_i^*\|^2} \tag{8}$$

where α is a constant defined by Equation 2. The output vector z is computed via

$$z = \frac{\sum_{i=1}^{c} \mu_i z_i^*}{\sum_{i=1}^{c} \mu_i} \tag{9}$$

This computational model can be viewed in terms of a fuzzy inference system employing traditional fuzzy if-then rules. Each rule has the following form:

If $\mathbf{Y}_1$ is $\mathbf{A}_1$ & $\mathbf{Y}_2$ is $\mathbf{A}_2$ &....then $\mathbf{Z}_1$ is $\mathbf{B}_1$ & $\mathbf{Z}_2$ is $\mathbf{B}_2$...

where Y_j is the j^{th} input variable and Z_j is j^{th} output variable; A_j is an exponential membership function and B_j is a singleton. For the i^{th} rule that is represented by cluster center x_i^*, A_j and B_j are given by:

$$A_j(q) = e^{-\alpha(q-y_{ij}^*)^2} \tag{10}$$

$$B_j = z_{ij}^* \tag{11}$$

where y_{ij}^* is j^{th} element of y_i^* and z_{ij}^* is the j^{th} element of z_i^*. This computational scheme is equivalent to an inference method that uses multiplication as the AND operator, weights the output of each rule by the rule's firing strength, and computes the output value as a weighted average of the output of each rule.

Equations 8 and 9 provide a simple way to translate a set of cluster centers into a fuzzy model. The way used in building fuzzy inference system in this paper is by allowing z_i^* in Equation 9 to be a second order Sugeno form. That is:

$$z_i^* = G_i y^2 + K_i y + h_i \tag{12}$$

where G_i is an $(M-N)XN$ constant matrix, K_i is an $(M-N)XN$ constant matrix, h_i is a constant column vector with M-N elements, and y^2 is a vector obtained by squaring each element of y. The equivalent if-then rules become a second order Sugeno consequent fuzzy inference system. It should be emphasized here that models that employ Sugeno type rules have been shown to be able to accurately represent complex behavior with only few rules. It was also shown that higher order Sugeno consequent models compared with lower order Sugeno models could identify systems with less error for the same number of rules or could achieve the required performance with fewer rules due to better approximation capability of higher order Sugeno consequent models [9, 10].

Expressing z_i^* as a second order Sugeno consequent type allows a significant degree of rule optimization to be performed, as pointed out by Sugeno [11]. Given a set of rules with fixed premises and optimizing the parameters in the consequent equations with respect to training data reduces the problem to a linear least square estimation problem. Such problems can be solved easily and the solution is optimal.

To convert the parameter optimization problem into the linear least square estimation problem, define

$$\rho_i = \frac{\mu_i}{\sum_{i=1}^{c} \mu_i} \tag{13}$$

Equation 9 can be rewritten as

$$z = \sum_{i=1}^{c} \rho_i z_i^* = \sum_{i=1}^{c} \rho_i (G_i y^2 + K_i y + h_i) \tag{14}$$

or

$$z^T = [\rho_1 (y^2)^T, \rho_1 y^T, \rho_1, \cdots, \rho_c (y^2), \rho_c y^T, \rho_c] \begin{bmatrix} G_1^T \\ K_1^T \\ h_1^T \\ \vdots \\ G_c^T \\ K_c^T \\ h_c^T \end{bmatrix} \tag{15}$$

where z^T, y^T are row vectors and $(y^2)^T$ is a row vector contains the squares of y. Given a collection of data points (y_1, y_2,..., y_n), the resultant collection of the output is given by:

$$\begin{bmatrix} z_1^T \\ \vdots \\ z_n^T \end{bmatrix} = \begin{bmatrix} \rho_1 (y_1^2)^T, \rho_1 y_1^T, \rho_1, \cdots, \rho_c (y_1^2)^T, \rho_c y_1^T, \rho_c \\ \vdots \\ \rho_1 (y_n^2)^T, \rho_1 y_n^T, \rho_1, \cdots, \rho_c (y_n^2)^T, \rho_c y_n^T, \rho_c \end{bmatrix} \begin{bmatrix} G_1^T \\ K_1^T \\ h_1^T \\ \vdots \\ G_c^T \\ K_c^T \\ h_c^T \end{bmatrix} \tag{16}$$

Note that given (y_1, y_2, ..., y_n), the first matrix on the right hand side of Equation 16 is constant, while the second matrix contains all the parameters to be optimized. To minimize the sum of error squares between the model output and that of the training data, we solve the linear least square estimation problem given by Equation 16, replacing the matrix on the left hand side by the actual output of the training data. Of course, implicit in the least squares estimation problem is the assumption that the number of training data is greater than the parameters to be optimized.

Using the standard notation adopted in most of literature, the least squares estimation problem of Equation 16 has the form

$$AX = B \tag{17}$$

where B is a matrix of output values, A is a constant matrix, and X is a matrix of parameters to be estimated. The well-known pseudo-inverse solution that minimizes $||\text{AX-B}||^2$ is given by

$$X = (A^T A)^{-1} A^T B \tag{18}$$

As a summary, model identification is performed in two steps:

1. Use subtractive clustering to find cluster centers and establish the number of fuzzy rules and the rule premises.

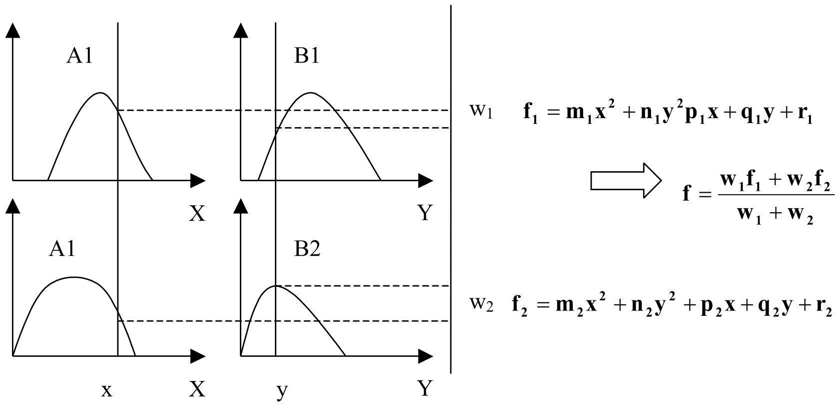
2. Optimize the rule consequents with second order Sugeno consequents.

4. Second Order Sugeno-ANFIS

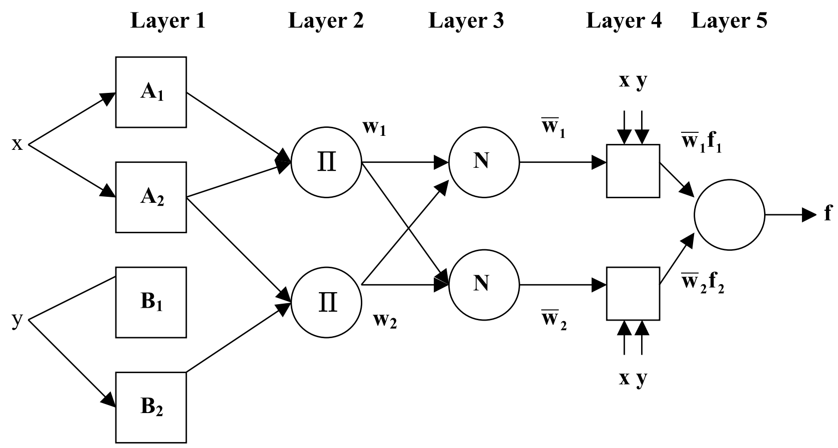
For simplicity, the fuzzy inference system under consideration is assumed to have two inputs x and y and one output z (the same example that was discussed in original ANFIS). Suppose also that the rule base contains two fuzzy if-then rules of second order Sugeno consequent type:

Rule 1: If x is A1 and y is B1, then $\boldsymbol{f}_1 = \boldsymbol{m}_1\boldsymbol{x}^2+\boldsymbol{n}_1\boldsymbol{y}^2+\boldsymbol{p}_1\boldsymbol{x}+\boldsymbol{q}_1\boldsymbol{y}+\boldsymbol{r}_l$,
Rule 2: If x is A2 and y is B2, then $\boldsymbol{f}_2 = \boldsymbol{m}_2\boldsymbol{x}^2+\boldsymbol{n}_2\boldsymbol{y}^2+\boldsymbol{p}_2\boldsymbol{x}+\boldsymbol{q}_2\boldsymbol{y}+\boldsymbol{r}_2$.

Then the second order Sugeno consequent fuzzy reasoning is illustrated in Fig. 1a and the corresponding equivalent ANFIS architecture is shown in Fig. 1b.



(a)



(b)

Fig. 1 *(a) Second order Sugeno fuzzy model, (b) Equivalent ANFIS.*

As in the original ANFIS the node functions in the proposed ANFIS who are in the same layer are of the same function family. Layers of the proposed ANFIS can be summarized as follows:

- **Layer 1**:

Every node i in this layer is a square node with a node function

$$O_i^1 = \mu_{A_i}(x) \tag{19}$$

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 linguistic label A_i. In the proposed approach μ_{A_i}(x) is selected to be Gaussian membership function using

$$\mu_{A_i}(x) = e^{\frac{-(x-c_i)^2}{2\sigma_i^2}} \tag{20}$$

where $\{c_i, \sigma_i\}$ is the parameter set of the Gaussian membership function. It should be noted here that we selected this type of membership functions to match the membership functions types generated from subtractive clustering method. Comparing this equation to Equation 8, the initial values for the parameter set of the Gaussian membership function can be found as follows:

$$c_i = y_i^* \tag{21}$$

$$\sigma = \sqrt{\frac{r_a}{2}} \tag{22}$$

which means that the initial values of c_i equal the cluster centers generated by applying subtractive clustering technique for input space. Also initial values of σ are related to r_a (radius of hyper sphere cluster in data space) as shown in Equation 22. As the values of these parameters change, the membership functions characteristics vary accordingly. It should be noted here that the parameters in this layer are referred to as premise parameters.

- **Layer 2**:

Every node in this layer is a circle node labeled $\mathbf{\Pi}$, which multiplies the incoming signals and sends the product out as follows:

$$w_i = \mu_{A_i}(x).\mu_{B_i}(x), i = 1, 2 \tag{23}$$

Each node output represents the firing strength of a rule.

- **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:

$$\overline{w}_i = \frac{w_i}{w_1 + w_2} \tag{24}$$

For convenience, the 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:

$$O_i^4 = \overline{w}_i f_i(m_i x^2 + n_i y^2 + p_i x + q_i y + r_i) \tag{25}$$

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

- **Layer 5**:

The single node in this layer is a circle node labeled by $\sum$ and it computes the overall output as the summation of all incoming signals as follows:

$$O_1^5 = overalloutput = \sum_i \overline{w}_i f_i = \frac{\sum_i w_i f_i}{\sum_i w_i}. \tag{26}$$

5. Hybrid Learning Algorithm in SOS-ANFIS

In the original ANFIS, the gradient method can be applied to identify the parameters in an adaptive network; this simple method usually takes a long time. It may be observed, however, that an adaptive network's output is linear in some of the network's parameters; thus these linear parameters can be identified by the linear least squares method. This approach leads to a hybrid learning rule, which combines the gradient descent method and the least square estimator for the fast identification of parameters. For simplicity, assume that the adaptive network under consideration has only one output

$$Output = F(\vec{I}, S) \tag{27}$$

where I is the set of input variables and S is the set of parameters. If there exists function H such that the composite function $H \circ F$ is linear in some of the elements of S, then these elements can be identified by the least squares method. More formally, if the parameter set S can be decomposed into two sets

$$S = S_1 \oplus S_2 \tag{28}$$

such that HoF is linear in the elements of S_2, and then upon applying H to Equation 27, we have

$$H(Output) = H \circ F(\vec{I}, S) \tag{29}$$

This is linear in the elements of S_2. Now the given values of elements of S_1, P training data can be plugged into Equation 29 and obtain a matrix equation:

$$AX = B \tag{30}$$

where :

- X is an unknown vector whose elements are parameters in S_2.

- A is the matrix of coefficients of parameters in S_2 obtained from supplying the inputs of the training data to the network.
- B is the matrix of outputs of the training data.

Let $|\mathrm{S}_2| = M$, then the dimensions of **A**, **X** and **B** are$P * M, M * 1$ and $P * 1$, respectively. Since P (the number of training data pairs) is usually greater than M (the number of linear parameters), this is an over determined problem and, generally, there is no exact solution to Equation 30. Instead, a least squares estimate (LSE) of **X**, **X*** is sought to minimize the squared error $||AX - B||^2$. This is a standard problem which forms the grounds for linear regression, adaptive filtering and signal processing. The most well-known formula for $X*$ uses the pseudo-inverse of X:

$$X = \left(A^T A\right)^{-1} A^T B \tag{31}$$

where $\mathbf{A}^T$ is the transpose of **A**, and $(\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T$ is the pseudo-inverse of **A** if $\mathbf{A}^T\mathbf{A}$ is non-singular.

Now the gradient method and the least squares estimate can be combined to update the parameters in an adaptive network. Each epoch of this hybrid learning procedure is composed of a forward pass and a backward pass. In the forward pass, the input data are supplied and the functional signals go forward to calculate each node output until the matrices **A** and **B** in Equation 30 are obtained, and the parameters in S_2 are identified by the sequential least squares formulas in Equation 31. After identifying the parameters in S_2, the functional signals keep going forward till the error measure is calculated. In the backward pass, the error rates (the derivative of the error measure with respect to each node output, propagate from the output end toward the input end, and the parameters in S_1 are updated by the gradient method.

For the given fixed values of parameters in S_1, the parameters in S_2 thus found are guaranteed to be the global optimum point in the S_2 parameter space due to the choice of the squared error measure. Not only can this hybrid learning rule decrease the dimension of the search space in the gradient method, but, in general, it will also cut down the convergence time substantially.

Let us consider a one-hidden-layer backpropagation neural network with sigmoid activation functions. If this neural network has p output units, then the output in Equation 27 is a column vector. Let us suppose $H(.)$ to be the inverse sigmoid function as follows:

$$H(x) = \ln\left(\frac{x}{x-1}\right). \tag{32}$$

As result Equation 29 becomes a linear (vector) function such that each element of *H(output)* is a linear combination of the parameters (weights and thresholds) pertaining to layer 2. In other words,

S_1 = weights and thresholds of a hidden layer
S_2 = weights and thresholds of an output layer.

Therefore the back-propagation learning rule can be applied to tune the parameters in the hidden layer, and the parameters in the output layer can be identified by the least squares method. It should be noted here that using the least squares method on the data transformed by $H(.)$, the obtained parameters are optimal in terms of the transformed squared error measure instead of the original one.

From the proposed ANFIS architecture (as in Fig. 1), it is observed that given the values of premise parameters, the overall output can be expressed as a linear combinations of the consequent parameters. More precisely, the output f in Fig. 1 can be rewritten as

$$\begin{aligned} f &= \frac{w_1}{w_1+w_2} + \frac{w_2}{w_1+w_2} \\ &= \overline{w}_1 f_1 + \overline{w}_2 f_2 \\ &= (\overline{w}_1 x^2)m_1 + (\overline{w}_2 y^2)n_1 + (\overline{w}_1 x)p_1 + (\overline{w}_1 y)q_1 + (\overline{w}_1)r_1 \\ &\quad (\overline{w}_2 x^2)m_2 + (\overline{w}_2 y^2)n_2 + (\overline{w}_2 x)p_2 + (\overline{w}_2 y)q_2 + (\overline{w}_2)r_2 \end{aligned} \tag{33}$$

which is linear with respect to the consequent parameters (m_1, $n_1, p_1, q_1, r_1, m_2, n_2,$ p_2, q_2, r_2). As a result, we have

S $=$ set of total parameters
S_1 $=$ set of premise parameters
S_2 $=$ set of consequent parameters

It can be noted that the hybrid-learning algorithm can be extended to work with second order Sugeno Consequents. As in the original ANFIS, in the forward pass of the hybrid learning algorithm, the functional signals go forward till layer 4 and the consequent parameters are identified by the least squares estimate. In the backward pass, the error rates propagate backward and the premise parameters are updated by the gradient descent.

6. Simulation Results

The best way to introduce the results of the proposed approach in this paper is through presenting two examples for modeling highly nonlinear functions. Each example is discussed; results are plotted, tabulated, and compared to formerly published results.

Example 1 – Modeling a Two Input Nonlinear Function

To test the proposed approach, the same example presented by Jang [1] is considered. In this example:

$$z = \frac{\sin(x)}{x} * \frac{\sin(y)}{y}. \tag{34}$$

From the grid points of the range [-10.5, 10.5]*[-10.5. 10.5] within the input space of the above equation, 484 data pairs are obtained. SOS-ANFIS is built to model these points. The proposed ANFIS is built through the following steps:

- **Step (1): Model identification:**

A fuzzy model is built using subtractive clustering technique and the rules are optimized using Sugeno second order consequents. The results are compared with the first order Sugeno fuzzy inference system.

Tab. I shows the root mean square modeling error of the second order Sugeno fuzzy inference system compared to the first order Sugeno system. It apparently shows the better modeling capability of the second order system.

Order of Sugeno model	RMS modeling error
First order	0.12122
Second order	0.0405977

Tab. I *The second order Sugeno RMSE compared to the first order model for example 1.*

- **Step (2): Second order Sugeno consequent ANFIS:**

The second order Sugeno Consequent ANFIS is built to model and train the above fuzzy inference system. Based on the results of subtractive clustering for input space, the proposed ANFIS contains 6 rules with 6 membership functions assigned to each variable. It's composed of 28 nodes as shown in Fig. 2:

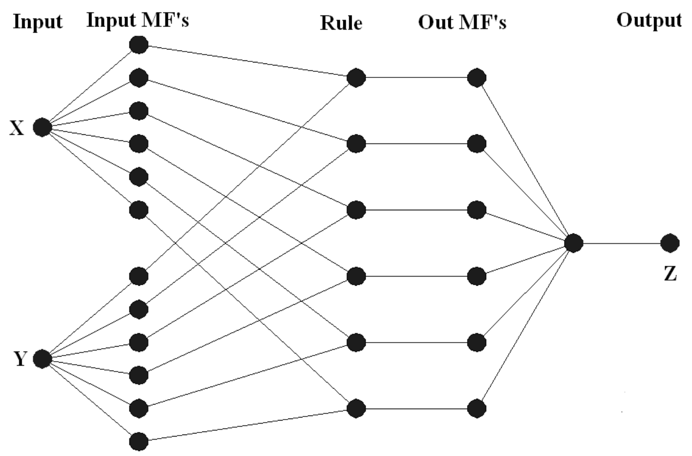


Fig. 2 *SOS-ANFIS structure for example 1.*

The proposed ANFIS model is used to train the fuzzy inference system generated in step 1. Results are superior and the proposed ANFIS shows better and quicker convergence and much better approximation capabilities. This can be summarized Fig. 3 and Tab. II:

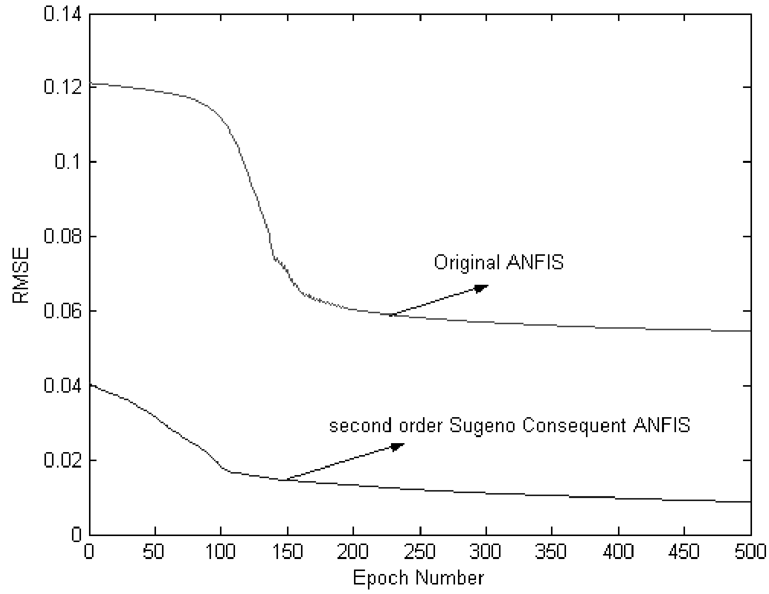


Fig. 3 *RMSE of original ANFIS compared to SOS-ANFIS for example 1.*

Epoch Number	RMSE of Original ANFIS	RMSE of Second order Sugeno Consequent ANFIS
50	0.119107	0.0315826
100	0.11178	0.0182965
200	0.0604216	0.0131787
300	0.0570506	0.011042
400	0.0554785	0.009782
500	0.0546667	0.00875871

Tab. II *Comparison of RMSE of Original ANFIS and SOS-ANFIS for example 1.*

Fig. 3 and Tab. II show clearly better approximation capability of the proposed ANFIS. It also shows better and quicker convergence of the proposed ANFIS. RMSE for SOS-ANFIS reaches 0.00875871 at epoch 500 while its only 0.0546667 using Original ANFIS architecture.

Example 2: Modeling a One Input Nonlinear Function

An example to show the proposed SOS-ANFIS capability in modeling one input nonlinear functions is presented here. A function that was used by Chiu [3] is used here. An SOS-ANFIS is built to model the following nonlinear function:

$$y = \frac{\sin(x)}{x} \tag{35}$$

From the grid points of the range [-20.5, 20.5] within the input space of the above equation, 42 data pairs are obtained. An SOS-ANFIS is built for modeling these points. The proposed ANFIS is built through the following steps:

- **Step (1): Model identification:**

A fuzzy model is built using subtractive clustering technique for generating the number and the type of membership functions and the rules are optimized using Sugeno second order consequents. The results are compared with the first order Sugeno fuzzy inference systems. Tab. III shows the root mean square modeling an error of the second order Sugeno fuzzy inference system compared to first order Sugeno system showing better approximation capabilities of the second order Sugeno system.

Order of Sugeno model	RMS modeling error
First order	0.214914
Second order	0.10653

Tab. III *SOS-ANFIS RMSE compared to the first order model for example 2.*

- **Step (2): second order Sugeno consequent ANFIS:**

SOS-ANFIS is built to model and train the above fuzzy inference system. The proposed ANFIS contains 3 rules with 3 membership functions assigned for the input. The proposed ANFIS architecture is composed of 12 nodes as in the following figure:

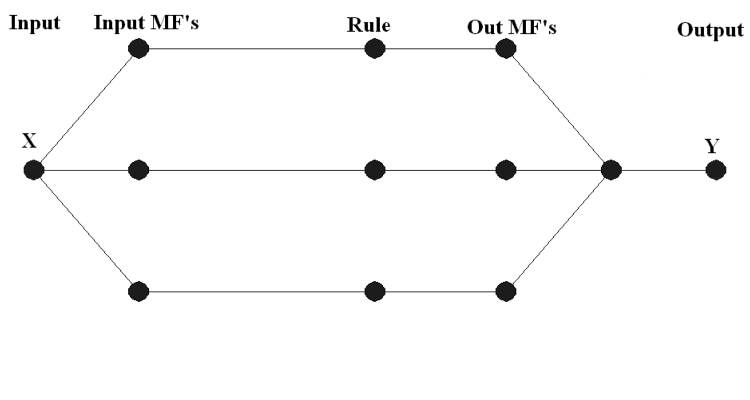


Fig. 4 *Second order Sugeno ANFIS structure for example 3.*

The total number of fitting parameters for the proposed ANFIS in this example is 15 including 6 premise (non linear) parameters and 9 SOS parameters.

The proposed ANFIS model is used to train the fuzzy inference system generated in step 1 up to 500 epochs. The results in this example are also better and the proposed ANFIS shows better convergence and less approximation error

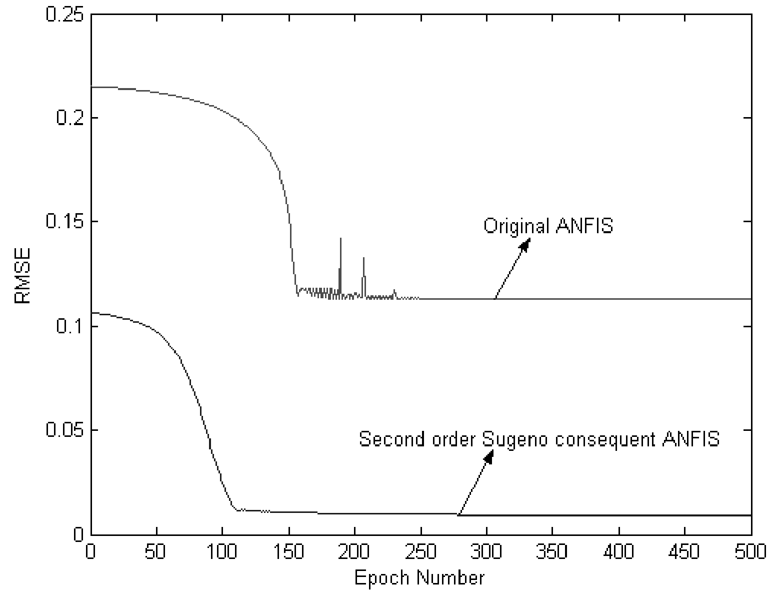


Fig. 5 *RMSE of original ANFIS compared to SOS-ANFIS for example 2.*

Epoch Number	RMSE of Original ANFIS	RMSE of Second order Sugeno Consequent ANFIS
50	0.212188	0.0971482
100	0.20351	0.0250073
200	0.115582	0.00997567
300	0.11294	0.00938936
400	0.11294	0.00903446
500	0.11294	0.00878385

Tab. IV *Comparison between the RMSE of Original ANFIS and SOS-ANFIS for example 2.*

As indicated by Fig. 5 and Tab. IV, this example shows again better results of the second order Sugeno consequent ANFIS. The proposed ANFIS in this paper gives the chance to start training from a better initial point and to converge better, quicker and with better approximation error during the training epochs. The RMSE of the second order Sugeno ANFIS reaches 0.00878385 at epoch 500 while it is only 0.11294 using Original ANFIS Training data, Original ANFIS and second order Sugeno Consequent ANFIS with 500 epochs are indicated in Fig. 6.

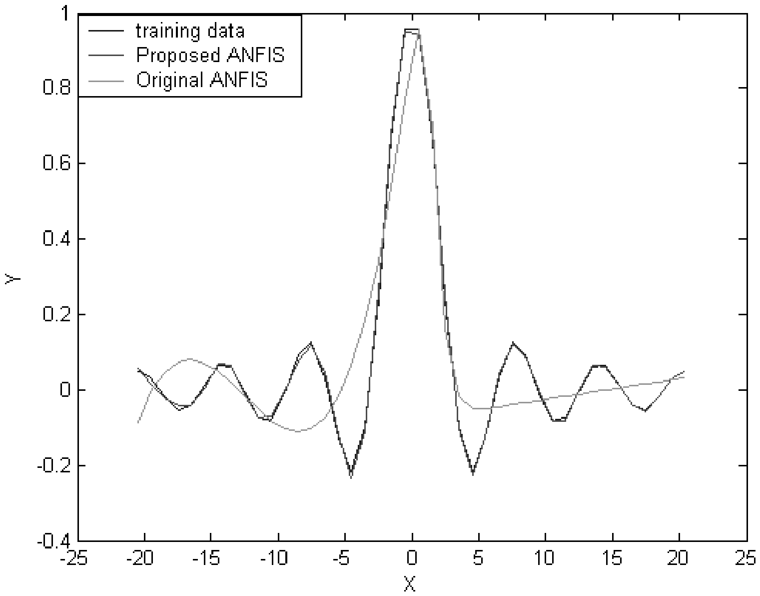


Fig. 6 *Comparison between training data, original ANFIS, and SOS-ANFIS for example 2.*

7. Conclusion

In this paper a modeling approach is presented using ANFIS with Sugeno second order consequents. Better approximation capability of the Sugeno second order consequents compared to lower order Sugeno consequents is shown. The original ANFIS proposed by Jang is extended in this paper to work with the second order Sugeno consequent models.

An effective subtractive clustering technique is used to determine the number and the type of membership functions for each input rather than being user specified which results in better performance and less approximation error. Fuzzy inference system is constructed using subtractive clustering and Sugeno second order consequents and then a hybrid learning algorithm that combines a gradient descent method and the least square estimate is used successfully to train and update the proposed ANFIS parameters.

Simulation of the proposed SOS-ANFIS for two examples shows better results than that of the original ANFIS. It is strongly recommended to use the SOS-ANFIS for systems with highly nonlinear behavior, where the use of higher order Sugeno models is expected to give much better modeling capabilities than lower orders.

References

[1] Jang J.: ANFIS: Adaptive-Network-Based Fuzzy inference System. IEEE Transactions on Systems, Man, and Cybernetics, **23**, 1993, pp. 665–685.

[2] Engin S., Kuvulmaz J., Omurlo V.: Fuzzy control of an ANFIS model representing a nonlinear liquid-level system. Neural Computing and Application, **13**, 2004, pp. 202-210.

[3] Babuska R., Verbruggen H.: Neuro-fuzzy methods for nonlinear system identification. Annual Reviews in Control, **27**, 2003, pp. 73-99.

[4] Chiu S. L.: Fuzzy model identification based on cluster estimation. Intelligent and Fuzzy Systems, **2**, 1994, pp. 267-278.

[5] Jang J. R., Sun C. T., Mizutan E.: Neuro-fuzzy and soft computing: a computational approach. Prentice Hall, 1996.

[6] Andrews R., Tickle A. B.: A survey and critique of techniques for extracting rules from trained artificial neural networks. Knowledge based Systems, **8**, 6, 1995, pp. 373–389.

[7] Werbos P.: An overview of neural networks for control. IEEE Control System Magazine, **11**, 1991, pp. 40-41.

[8] Chiu S.: Fuzzy model identification based on cluster estimation. Intelligent and Fuzzy Systems, **2**, 1994, pp. 267–278.

[9] Alata M., Demirli K., Su C. Y.: Adaptive/fuzzy control of a class of nonlinear systems with a first order parameterized Sugeno fuzzy approximator. IEEE Transactions on Systems, Man, and Cybernetics. **31**, 2001, pp. 410-420.

[10] Demirli K., Muthkumaran P.: Fuzzy system identification with higher order subtractive clustering. Intelligent and Fuzzy Systems, **9**, 2000, pp. 129-145.

[11] Takagi T., Sugeno M.: Fuzzy identification of systems and its applications to modeling and control. IEEE Transactions on System, Man, and Cybernetics. **15**, 1985, pp. 116-132.