

A New Method of Combined Classifier Design Based on Fuzzy Neural Network

Kexin Jia, and Youxin Lu

Abstract—To make the modulation classification system more suitable for signals in a wide range of signal to noise rate (SNR), a novel method of designing combined classifier based on fuzzy neural network (FNN) is presented in this paper. The method employs fuzzy neural network classifiers and interclass distance (ICD) to improve recognition reliability. Experimental results show that the proposed combined classifier has high recognition rate with large variation range of SNR (success rates are over 99.9% when SNR is not lower than 5dB).

Keywords—Modulation classification, combined classifier, fuzzy neural network, interclass distance.

I. INTRODUCTION

WITH rapidly development of communication technology, the communication environment becomes more and more complicated these years. Many signal modulation types are used simultaneously in communication systems. Therefore, a need arises for modulation classification that can automatically detect the incoming modulation type. It plays an important role in many cooperative or non-cooperative communication applications such as software radio, threat analysis, electronic surveillance system, etc. In order to accomplish modulation identification, many methods of designing single classifier have been proposed, but most of them could not suitable for signals in a wide range of SNR. Suppose a classifier performs very well in a wide range of SNR, it must be trained with a great deal of samples produced in many different SNR and its structure is highly complex.

To solve this problem, a novel method of designing combined classifier is proposed in this paper. The designed combined classifier combines FNN classifiers and ICD to make the modulation classification more suitable for signals in a wide range SNR. It is superior to the other combined classifiers [1][2] in simple structure and generalization ability.

The rest of this paper is organized as follows: In section II we present a fuzzy neural network classifier. In section III, a new method of designing combined classifier is discussed. Experimental results on modulation identification of six kinds of communication signals using the proposed technique are given in section IV. Finally, discussion and suggestion for future research are addressed in section V.

II. FUZZY NEURAL NETWORK

A fuzzy neural network is a fuzzy logic system (FLS) which is designed with recursive algorithms such as backpropagation (BP) algorithm. Therefore, we firstly review a structure of fuzzy logic system [3]-[5], and then discuss the corresponding fuzzy neural network of FLS and its study algorithm in this section.

A. Fuzzy Logic System

Fuzzy logic system is both intuitive and numerical system that map crisp inputs, x , into a crisp output, y . Every FLS is associated with a set of rules with meaningful linguistic interpretations, such as

$$R^{(l)}: \text{If } \mu_1 \text{ is } F_1^l, \dots, \mu_n \text{ is } F_n^l \text{ then } v \text{ is } G^l,$$

This can be obtained either from numerical data, or experts familiar with the problem at hand. Based on this kind of statement, actions are combined with rules in an antecedent/consequent format, and then aggregated according to approximate reasoning theory, to produce a nonlinear mapping from the input space $U = U_1 \times U_2 \cdots \times U_n$ to the output space V , where $F_k^l \subset U_k, k = 1, 2, \dots, n$ are the antecedent membership functions, and $G^l \subset V$ is the consequent membership function. The input linguistic variables are denoted by $\mu_k, k = 1, 2, \dots, n$, and the output linguistic variable is denoted by v .

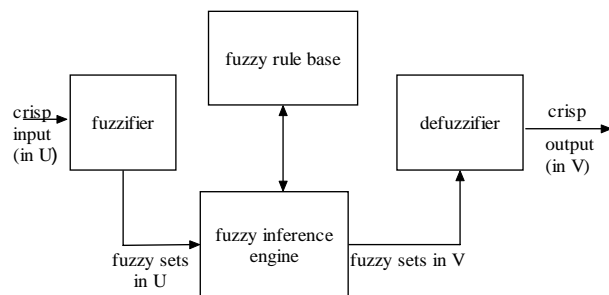


Fig. 1 Structure of fuzzy logic system

A fuzzy logic system consists of four basic elements (as shown in Fig. 1: the fuzzifier, the fuzzy rule base, the inference engine, and the defuzzifier. The fuzzy rulebase is a

Manuscript received March 21, 2007

K. X. Jia is with the Institute of Electronic of University of Electronic Science and Technology of China (phone: 028-83202451; e-mail: jiaxixin@sina.com).

collection of rules of the form of R^l , which are combined in the inference engine, to produce a fuzzy output (in essence, the inference engine produces mappings from fuzzy sets to fuzzy sets). The fuzzifier maps the crisp inputs into fuzzy sets, which are subsequently used as inputs to the inference engine, whereas the defuzzifier maps the fuzzy sets produced by the inference engine into crisp numbers.

Fuzzy sets can be interpreted as membership functions u_X that associate with each element x of the universe of discourse, U , a number $u_X(x)$ in the interval $[0,1]$:

$$u_X : U \rightarrow [0,1]. \quad (1)$$

The fuzzifier maps a crisp point $x \in U$ into a fuzzy set $X \in U$. In the case of a singleton fuzzifier, the crisp point $x \in U$ is mapped into a fuzzy set X with support x_i , where $u_X(x_i) = 1$ for $x_i = x$ and $u_X(x_i) = 0$ for $x_i \neq x$, i.e., the single point in the support of X with nonzero membership function value is $x_i = x$. In the case of a nonsingleton fuzzifier, the point $x \in U$ is mapped into a fuzzy set X with support x_i , where u_X achieves maximum value at $x_i = x$ and decreases while moving away from $x_i = x$. We assume that fuzzy set X is normalized so that $u_X(x_i) = 1$.

Nonsingleton fuzzification is especially useful in cases where the available training data, or the input data to the fuzzy logic system, contain any kind of uncertainty. Conceptually, the nonsingleton fuzzifier implies that the given input value x is the most likely value to be the correct one from all the values in its immediate neighborhood; however, because of the presence of uncertainty, neighboring points are also likely to be the correct values, but to a lesser degree.

The shape of the membership function u_X can be determined by the system designer, based on an estimate of the kind and quantity of uncertainty present. It would be the logical choice, though, for the membership to be symmetric about x , since the effect of noise is most likely to be equivalent on all points. Examples of such membership

function is the Gaussian $u_X(x_i) = \exp\left(-\frac{(x-x_i)^2}{2\sigma^2}\right)$,

where the variance σ^2 reflects the width (spread) of $u_X(x_i)$; Note that larger values of the spread of the above membership function implies that a higher degree of uncertainty is anticipated to exist in the given data.

Each input to the fuzzy logic system, after having been processed through the fuzzifier, will activate each rule in the rulebase to a different degree. The fuzzy rules $R^l, l = 1, 2, \dots, M$ will produce output fuzzy sets Y^l which will be aggregated, typically using t-conorm operations, to produce the total output fuzzy set:

$$\gamma = \bigcup_{l=1}^M Y^l \quad (2)$$

where $\bigcup_{l=1}^M$ denotes a sequence of t-conorm operations.

Most engineering applications of FLS's require a crisp output, therefore the fuzzy set γ is mapped to a single point, f , using a defuzzification operation:

$$D: f = D(\gamma). \quad (3)$$

A general way to express an n-input single-output FLS with M rules in its rulebase is as follow:

$$f = D\left(\bigcup_{l=1}^M \int_V u_{G^l}(y) * T_{k=1}^n u_{Q_k^l}(x_{k,\text{sup}}^l) / y\right) \quad (4)$$

Where $*$ denotes t-norm operation; $T_{k=1}^n$ denotes a sequence of t-norm operations; \int denotes union of points in the continuum; D is a general defuzzifier that maps fuzzy sets in the input space V to crisp points in V ; $u_{Q_k^l}(x_{k,\text{sup}}^l) \equiv u_{F_k^l}(x_{k,\text{sup}}^l) * u_{X_k}(x_{k,\text{sup}}^l)$; $u_{F_k^l}(x_{k,\text{sup}}^l)$ is the membership function for the k th antecedent of the l th rule; $u_{X_k}(x_{k,\text{sup}}^l)$ is the membership function for the k th input fuzzy set; and $x_{k,\text{sup}}^l$ is the point that maximizes $u_{Q_k^l}$.

A. Fuzzy Neural Network

The corresponding FNN[3]-[6] of FLS is briefly reviewed. It consists of four layers as shown in Fig. 2. Note that only three-input single-output system is considered in Fig. 1 for simplicity. Consider an FNN with n input x_i ($i=1, \dots, n$) and output y .

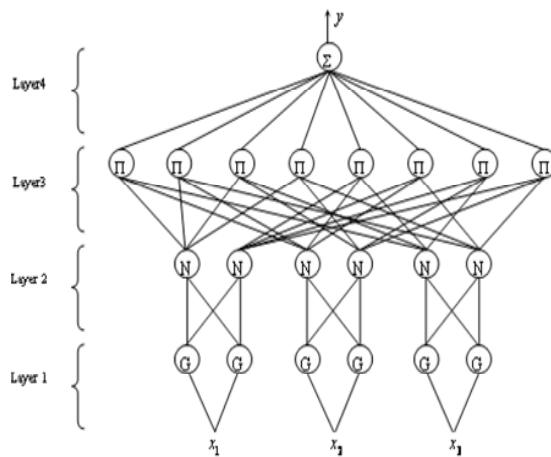


Fig. 2 Fuzzy neural network structure

Layer 1: each node in this layer corresponds to a membership function $\mu(x_i)$. Suppose that each variable x_i is

divided into n_i fuzzy partitions and a gauss function is adopted, the output is defined as

$$\mu_{F_i^j}(x_i) = \exp\left(-\left(\frac{x_i - m_{ij}}{\sigma_{ij}}\right)^2\right) \quad (5)$$

Where, F_i^j represents input variable x_i to the j th node and $\{m_{ij}, \sigma_{ij}\}$ is the mean and variation of gauss function.

Layer 2: each node in this layer normalizes the output of layer 1

$$\mu'_{F_i^j}(x_i) = \frac{\mu_{F_i^j}(x_i)}{\sum_{k=1}^{n_i} \mu_{F_i^k}(x_i)} \quad (6)$$

where, $\mu'_{F_i^j}(x_i)$ is the normalized output of layer 1.

Layer 3: the output of each node in this layer represents the firing strength of a rule. So the number of nodes in this layer is equal to the number of rules N . If input variable x_i is divided into n_i fuzzy partitions and a product t-norm is adopted, then

$$out_j = \prod_{i=1}^n \mu'_{F_i^k}(x_i) \quad j=1, 2, \dots, N \quad (7)$$

$$N = \prod_i n_i$$

Layer 4: the final output is computed by summing the weighted outputs of all rules.

$$y = \sum_{j=1}^N \beta_j out_j \quad (8)$$

C. back-propagation (BP) method

For the development of the BP training algorithm for updating the design parameters of a FNN, we focus on a neural network with the following format:

$$y(x) = \sum_{j=1}^m \beta_j^j \left(\prod_{i=1}^n \mu'_{F_i^j}(x_i) \right) \quad (9)$$

Given an input-output training pair (x, d) , $x \in R^n$ and $d \in R$, we wish to design a FNN in the form of (1) such that the following error function is minimized

$$E = \frac{1}{2} (y(x) - d)^2 \quad (10)$$

It is evident from (9), that y is completely characterized by the parameters of gauss function $\{m_{ij}, \sigma_{ij}\}$ and the weight values $\{\beta_j\}$. Using a steepest descent algorithm to minimize E , it is straight forward to obtain the following recursions to update all design parameters of a FNN:

$$\delta^4 = \alpha - y \quad (11)$$

$$\delta_j^3 = \delta^4 \beta_j \quad (12)$$

$$\sigma_{ij}^2(t) = \sum_p \delta_p^3 out_p^3 - \mu_{F_i^j}(x_i) \sum_{k=1}^{n_i} \sum_q \delta_q^3 out_q^3 \quad (13)$$

$$\beta^j(t+1) = \beta^j(t) + \eta \delta^4 out_j^3 + \alpha \Delta \beta^j(t) \quad (14)$$

$$\Delta \beta^j(t) = \beta^j(t) - \beta^j(t-1)$$

$$m_{ij}(t+1) = m_{ij}(t) + \eta \sigma_{ij}^2 \frac{2(x_i - m_{ij})}{\sigma_{ij}^2} + \alpha \Delta m_{ij}(t) \quad (15)$$

$$\Delta m_{ij}(t) = m_{ij}(t) - m_{ij}(t-1)$$

$$\sigma_{ij}(t+1) = \sigma_{ij}(t) + \eta \sigma_{ij}^2 \frac{2(x_i - m_{ij})^2}{\sigma_{ij}^3} + \alpha \Delta \sigma_{ij}(t) \quad (16)$$

$$\Delta \sigma_{ij}(t) = \sigma_{ij}(t) - \sigma_{ij}(t-1)$$

where η is a study step, α is a parameter with $0 < \alpha < 1$.

Equations (11)-(16) are referred to as backpropagation algorithm because of their dependence on backpropagation error δ^4 .

III. NEW METHOD OF DESIGNING COMBINED CLASIFFIER

The structure of the combined classifier is depicted in Fig.3. T is a feature vector extracted from an input signal. $f(r_i)$ denotes the i th classifier output. These classifiers are realized by the FNN discussed in section II.

For a multi-class modulation identification problem, assume there is a training data set $S: \{x_m, y_m\}_{m=1}^M$, where x_m represents the m th input feature vector and $y_m \in \{0, 1, \dots, C-1\}$ is the desired output of x_m , C is the number of communication signals. FNN classifiers r_n ($n=1, 2, \dots, N$) in the combined classifier are trained by training data set and their output are denoted by $f(r_n)$ ($n=1, 2, \dots, N$).

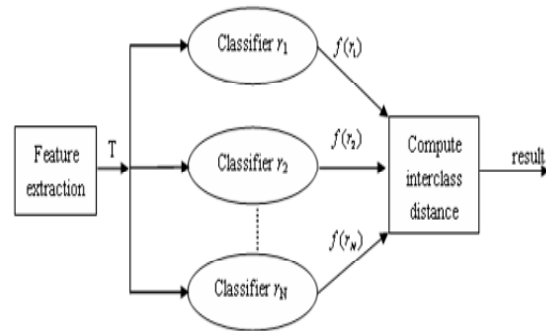


Fig. 3 The structure of combined classifier for modulation classification

For the sake of computing interclass distance, the following matrix can be created:

$$D = \begin{bmatrix} 0 & 0 & \cdots & 0 \\ 1 & 1 & \cdots & 1 \\ \vdots & \vdots & \vdots & \vdots \\ C-1 & C-1 & \cdots & C-1 \end{bmatrix}_{C \times N} \quad (17)$$

The output of classifiers $f(r_n) (n=1,2,\dots,N)$ and each row of matrix D are used to calculate interclass distance and column vector MR can be gotten.

$$MR = [m_0, m_1, \dots, m_{C-1}]' \quad (18)$$

Where m_i is the distance between $f(r_n) (n=1,2,\dots,N)$ and the i th row of matrix D. It can be computed by the following equation:

$$m_i = \frac{1}{N} \sum_{n=1}^N |f(r_n) - D(i, n)| \quad (19)$$

Where, $D(i, n) (n=1,2,\dots,N)$ is the i th row of matrix D.

Therefore, input feature vector is classified into class j , which the row index is corresponding to the minimum of column vector MR.

IV. SIMULATION RESULT

The performance of the proposed combined classifier is investigated in the MATLAB environment. Six digital modulation types commonly used in communication are tested. They are band-limited digital signals with band-limited AWGN and their carrier frequencies f_c are prior information. The range of SNR is from 5 dB to 20 dB. Five feature parameters $\{\gamma_{\max}, \sigma_{ap}, \sigma_{dp}, \sigma_{aa}, \sigma_{af}\}$, proposed in [7], are used to identify six communication signals in this paper.

For each modulation types, every 1000 samples from a simulated signal construct a frame to get one feature vector. At every 5 dB (i.e. 5dB, 10dB, 15dB, 20dB), 1200 different feature vectors are achieved for each signal, 200 of them for training and the rest for testing. In this way, four groups of training set are gotten for each modulation type. Take out the training set in every group and combine two of them to make five new groups: 5dB and 10 dB training set, 5 dB and 15 dB training set, 10 dB and 15 dB training set, 10 dB and 20 dB training set, 5 dB and 20 dB training set. The corresponding classifiers are labeled in I, II, III, IV, V.

TABLE I
SUCCESS RATES (%) OF SINGLE CLASSIFIER WITH SNR=5dB

Modulation type	Classifier				
	I	II	III	IV	V
2ASK	100	100	0.02	0.01	99.5
4ASK	100	100	100	100	99
2FSK	100	99	7.2	1.4	100
4FSK	100	100	100	100	100
2PSK	100	100	59	40.3	100
4PSK	100	98.5	0.09	1	100

The test results of single classifiers and combined classifier are shown in Fig. 4, which is generated by linear regression at SNR=5 dB. Fig. 4 (a)-(f) stand for the test results of five single classifiers and combined classifier respectively. Take

TABLE II
SUCCESS RATES (%) OF SINGLE CLASSIFIER WITH SNR=10dB

Modulation type	Classifier				
	I	II	III	IV	V
2ASK	99.7	99.2	100	97.7	98.3
4ASK	100	100	100	100	100
2FSK	100	100	100	100	100
4FSK	100	100	100	100	100
2PSK	100	100	100	100	100
4PSK	100	100	100	100	100

TABLE III
SUCCESS RATES (%) OF SINGLE CLASSIFIER WITH SNR=15dB

Modulation type	Classifier				
	I	II	III	IV	V
2ASK	100	100	100	99.5	100
4ASK	100	100	100	100	100
2FSK	99.9	100	100	100	100
4FSK	100	100	100	100	100
2PSK	100	100	100	100	100
4PSK	100	100	100	100	100

TABLE IV
SUCCESS RATES (%) OF SINGLE CLASSIFIER WITH SNR=20dB

Modulation type	Classifier				
	I	II	III	IV	V
2ASK	100	100	100	100	100
4ASK	100	100	100	100	100
2FSK	100	100	100	100	100
4FSK	100	100	100	100	100
2PSK	100	100	100	100	100
4PSK	100	100	100	100	100

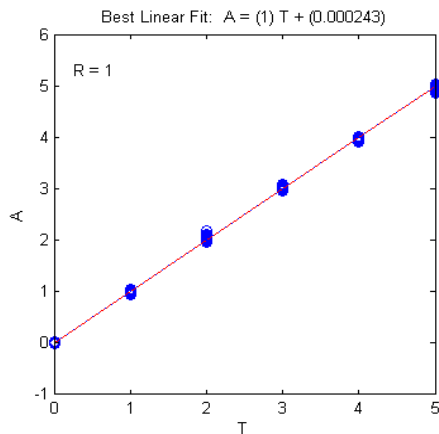
TABLE V
SUCCESS RATES (%) OF COMBINED CLASSIFIER

Modulation type	Classifier			
	5dB	10dB	15dB	20dB
2ASK	99.9	100	100	100
4ASK	100	100	100	100
2FSK	100	100	100	100
4FSK	100	100	100	100
2PSK	100	100	100	100
4PSK	100	100	100	100

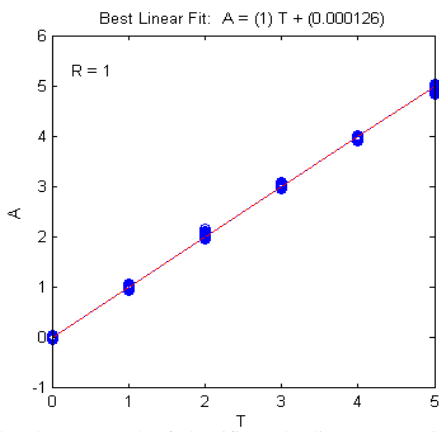
Fig.4 (a) for example, on the top of the figure there is an expression:

$$A = (1)T + (0.000243)$$

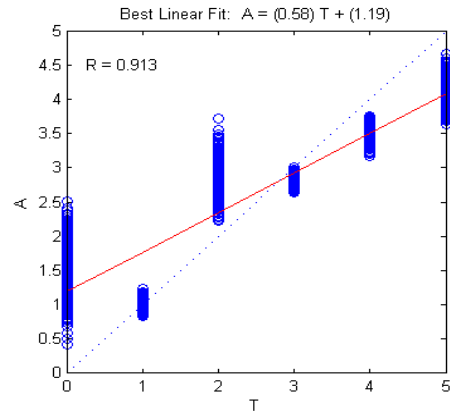
where the value $m=1$ stands for the best regress line's slope and $b=0.000243$ stands for the line's intercept. When $m=1$ and $b=0$, the output of the classifier is entirely uniform with the goals. R stands for the correlation coefficient of the output of the classifier and the goals. The smaller the difference between R and 1 is the better the performance of the classifier. In Fig. 4 (a), the horizontal axe is the goal; the vertical axe is the output of the classifier. The ideal regress line (the output of the classifier equals to the goals) is expressed by the real line; the best regress line is denoted by dashed line. From the figure we can see that only classifier I, II, V has very good performances, however, the combined classifier works well.



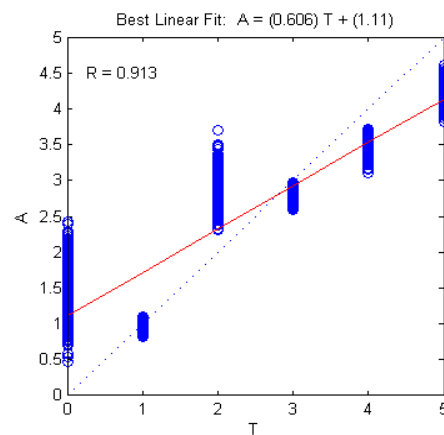
(a) The test result of classifier I by linear regression



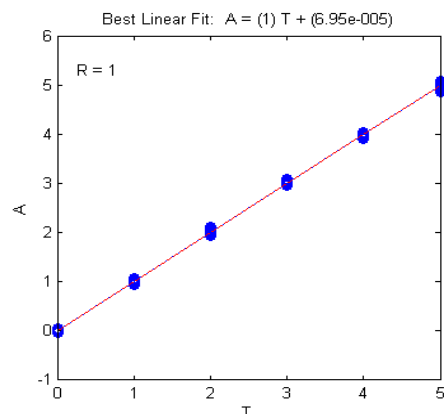
(b) The test result of classifier II by linear regression



(c) The test result of classifier III by linear regression

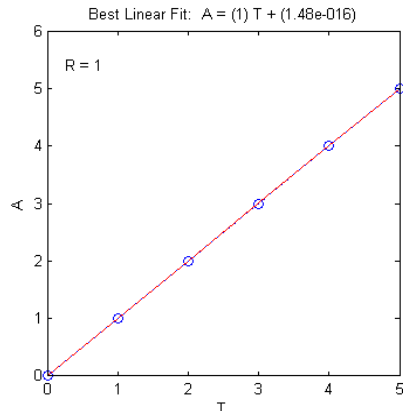


(d) The test result of classifier IV by linear regression



(e) The test result of classifier V by linear regression

Signals at SNR=5 dB, 10 dB, 15 dB, 20dB are tested to show the performance of single classifiers and results are presented in Table I, II, III, IV. In Table I, classifier I, II and V have good performances, because these three classifiers have been trained at SNR 5dB. They accord with what we have seen in Fig. 4. In Table II, while classifier I, IV and V has not been trained at SNR 15dB, they also have good performance.



(f) The test result of combined classifier by linear regression

Fig. 4 The test results of single classifiers and combined classifier

The same result can be obtained from Table III and IV, because FNN classifiers have very good generalization ability. Table V gives the results of the proposed combined classifier. Success rates are high and stable when SNR changes, which proves that the combined classifier are suitable for signals in a wide range of SNR. Its overall success rates of the six modulation types are over 99.9% when SNR is not lower than 5 dB.

V. CONCLUSION

A novel combined classifier combined FNN and interclass distance for modulation classification is presented in this paper. Its structure is simple and it has good generalization ability. Computer simulation shows that the proposed method

can classify six digital modulation types in a wide range of SNR with high success rates. One issue worth considering in future investigation is the selection of feature parameters. The dependence of the feature parameters used in this paper on the SNR has been discussed in [7]. It has shown that the five feature parameters could classify six signals when SNR is not lower than 5db. Therefore, the proposed method can identify six signals at 5 dB. When SNR is lower than 5 dB, other better feature parameters must be selected.

REFERENCES

- [1] D. Wu, X.M Gu, and Q. Guo, "Classification using a novel support vector machine fuzzy network for digital modulations in satellite communication," in *Proc. Asia Conf. on Wireless Communications Networking and Mobile Computing, Wuhan, China*, pp. 508-512, Sept. 2005.
- [2] T. J. Lu, S.G. Guo, and X.C. Xiao, "A method of modulation identification for communication signals based on fuzzy integral," *ACTA Electronica Sinica*, Vol. 29, No. 6, June, 2001.
- [3] M.M. Jerry, C.M. George. "Designing fuzzy logic system," *IEEE Transactions on Circuits and Systems-II: Analog and Digital Signal Processing*, vol. 44, no.11, pp 885-893, Feb.1997.
- [4] H.H Tsai, P.T Yu, "On the optimal design of fuzzy neural networks with robust learning for function approximation," *IEEE Trans. on Systems*. Vol.30, no. 1, pp. 217-223, Feb. 2000.
- [5] C.H Kim, J.J Lee, "Adaptive network-based fuzzy inference system with pruning," in *Proc. Asia Conf. on SICE, Fukui, Japan*, pp. 140-143, Aug.4-6, 2003.
- [6] F.L Chung, J.C Duan, and D.S Yeung, "Deriving multistage FNN models from takagi and sugeno's fuzzy systems," in *Proc. European Conf. on Computational Intelligence*, pp. 1259-1264, May 1998.
- [7] E. E Azzouz, A.K Nandi, "Automatic identification of digital modulation types," *IEEE Signal Processing*, Vol.47, pp. 55-69, Nov. 1995.