A Fuzzy Logic Method for Modulation Classification in Nonideal Environments

Wen Wei and Jerry M. Mendel, *Fellow, IEEE*

***Abstract*—In this paper, we present a fuzzy logic modulation classifier that works in nonideal environments in which it is difficult or impossible to use precise probabilistic methods. We first transform a general pattern classification problem into one of function approximation, so that fuzzy logic systems (FLS's) can be used to construct a classifier; then, we introduce the concepts of fuzzy modulation type and fuzzy decision and develop a nonsingleton fuzzy logic classifier (NSFLC) by using an additive FLS as a core building block. Our NSFLC uses two-dimensional (2-D) fuzzy sets, whose membership functions are isotropic so that they are well suited for a modulation classifier (MC). We establish that our NSFLC, although completely based on heuristics, reduces to the maximum-likelihood modulation classifier (ML MC) in ideal conditions. In our application of NSFLC to MC in a mixture of α-stable and Gaussian noises, we demonstrate that our NSFLC performs consistently better than the ML MC and it gives the same performance as the ML MC when no impulsive noise is present.**

***Index Terms*—Fuzzy systems, impulse noise, pattern classification, quadrature amplitude modulation.**

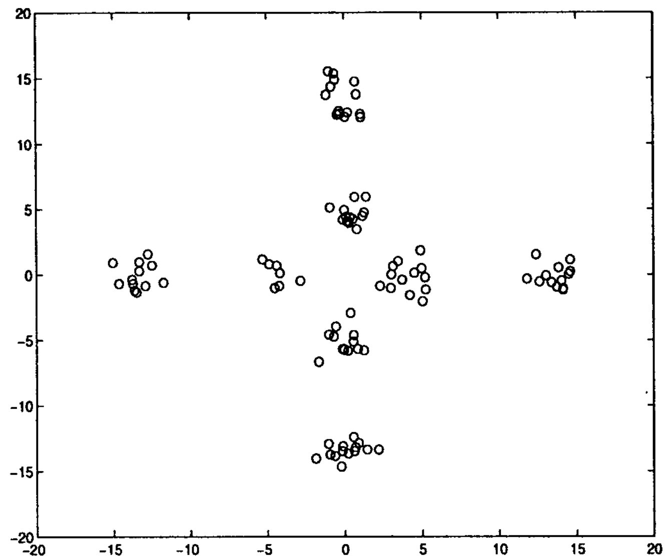


Fig. 1. One hundred points of Star 8-QAM data at SNR = 20 dB.

I. Introduction

MODULATION classification (MC) is a technique to identify the modulation type of a modulated signal corrupted by noise. It is an important problem in noncooperative communication applications such as electronic surveillance. A formal description of MC is as follows.

Definition 1: Given a measurement $r(t)$ $0 \le t \le \tau$, a modulation classifier is a system that recognizes the modulation type of $r(t)$ from c possible modulations $\{\mathcal{I}_1, \mathcal{I}_2, \cdots, \mathcal{I}_c\}$.

The received signal $r(t)$ is typically considered as a modulated signal received through a communication channel and corrupted by additive noise, i.e.,

$$r(t) = s(t) + n(t) \tag{1}$$

where $s(t)$ is the signal and $n(t)$ is the noise.

Various methods have been developed for this problem [1]–[7]. While most of the earlier works lean more toward the practical side rather than the theoretical aspects, a maximum-likelihood modulation classifier (ML MC) has been introduced in [1], and the theoretical limits of that classifier have been developed. The ML MC assumes ideal conditions that the noise is Gaussian and signal parameters (namely, carrier frequency and phase, symbol timing, signal power, noise power) are known. The method applies to any kind of digital modulation that can be described by a constellation, e.g., BPSK, QPSK, 8-PSK, 16-PSK, 32-PSK, 64-PSK, 16-QAM, V29, V32 (32-QAM), 64-QAM, V29c (Star 8-QAM), etc.

Unfortunately, the ideal conditions that are assumed by the ML MC are typically not the case in the real world. First, the signal parameters are typically not completely time-invariant, and, therefore, should be estimated from measurements and adjusted in real time. The use of estimated parameters introduces degradations in classifier performance that can be very difficult to model precisely. Second, non-Gaussian noise, especially impulsive noise, has been reported to exist in many communication environments. Because impulsive noise behaves quite differently from Gaussian noise, a ML MC based on Gaussian noise may perform poorly in such noise. Since the exact statistical nature of the impulsive noise is, in general, unknown, it is usually not possible to design a ML MC for it. A fuzzy logic (FL) MC is not based on a probability model and is the main subject of this paper. Our goal is to develop a classifier that gives comparable performance to that of a ML MC when ideal conditions hold (Gaussian) and has a more robust performance than the ML MC in nonideal environments.

In Section II, we review the signal modeling of the ML MC; the derivation of the ML MC is reviewed in Appendix A. In Section III, we present a fuzzy logic classifier that is

Manuscript received February 18, 1998; revised January 1, 1999. This material is based on work supported by the University of Southern California under Grant MIP-9419386 of the National Science Foundation.

The authors are with the Signal and Image Processing Institute and the Department of Electrical Engineering-Systems, University of Southern California, Los Angeles, CA 90089 USA.

Publisher Item Identifier S 1063-6706(99)04941-3.

1063–6706/99$10.00 © 1999 IEEE

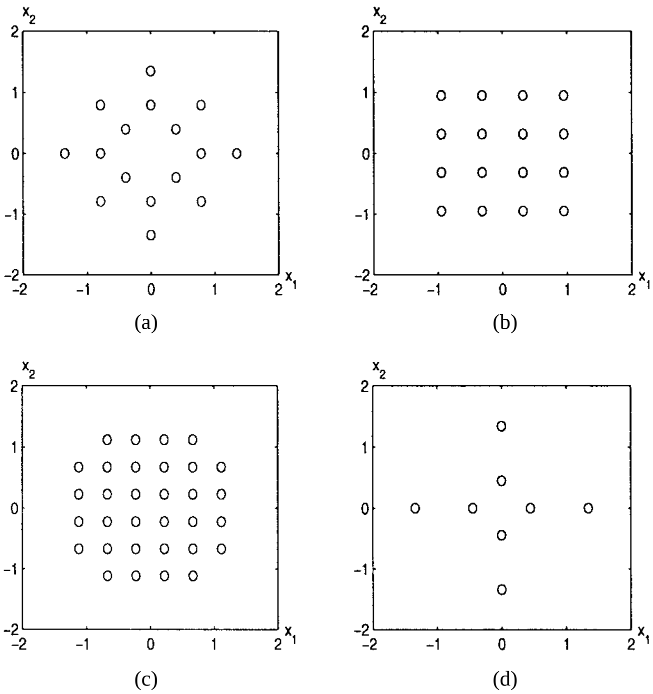


Fig. 2. Examples of normalized constellations. (a) V.29. (b) 16-QAM. (c) 32-QAM. (d) Star 8-QAM.

based on a nonsingleton fuzzy logic system. In Section IV, we apply our fuzzy logic classifier to MC in impulsive noise environments. Section V concludes the paper.

II. Review of Probabilistic Modulation Classification

A Maximum-Likelihood Classifier in Ideal Conditions: A digital amplitude-phase modulation uses the amplitude and phase information of a signal during time segments (symbols) to carry information. Each modulation is associated with a set of points on the complex plane $\{S_1, S_2, \cdots, S_M\}$ called a *constellation*. The information to be carried by a signal is first coded into a sequence of complex data, each element of which assumes a value from the constellation; then, the modulator maps each complex datum into one symbol length of continuous waveform.

A modulated signal, when received through a communication channel, can be expressed as

$$s(t) = \mathrm{Re}\left\{ \sum_{n=-\infty}^{k} A\hat{S}_n p(t-nT) e^{j(2\pi f_c t + \theta_c)} \right\}, \quad kT < t \leq (k+1)T, \; k = 0, 1, 2, \cdots \tag{2}$$

where f_c and θ_c are the carrier frequency and phase, respectively, T is the symbol period, $p(t)$ is a pulse-shape function (which represents the impulse response of the overall signal path, including transmitter, channel, and receiver), A is the signal amplitude, and $\hat{S}_n$ assumes a value from the M complex numbers in the constellation of the modulation. The constellation is usually normalized so that it has unity average power, i.e., $\sum_{i=1}^{M} |S_i|^2/M = 1$. Note that the key property of this family of signals is that the instantaneous frequency does not change within each symbol, which ensures an equivalence between time-domain and complex-domain representations of the signal. This equivalence makes it possible for us to map a received time-domain signal back into the complex domain, and match the received complex data to a library of given constellations. When a sequence of received data is plotted on a complex domain, cluster formations can be visually recognized that resemble the original constellation, if signal quality is high enough. Fig. 1 shows an example of a Star 8-QAM signal at a signal-to-noise ratio (SNR) of 20 dB. The constellations of Star 8-QAM and three other modulations are depicted in Fig. 2, where the real part x_1 and the imaginary part x_2 are also called the in-phase and quadrature components of the constellation, respectively.

When enough statistical information about the signal and communication channel is known, the matching of a received signal to the library of constellations can be done with likelihood tests. Fig. 3 shows a diagram of a maximum-likelihood modulation classifier (ML MC). Note that by working in the complex domain, we process only N complex-domain data for a time interval of N symbols, instead of having to process the continuous-time waveform. This greatly reduces the complexity of the classifier.

The ML MC assumes the following ideal conditions.

1) The communication channel can be perfectly equalized, i.e., $p(t) = 1$ for $0 < t < T$ and $p(t) = 0$ otherwise.
2) The additive noise is white and Gaussian and its power density N_0 is known.
3) All signal parameters, i.e., carrier frequency and reference phase, symbol epoch, and signal amplitude, are known.
4) Carrier frequency is a multiple of symbol rate, i.e., $f_c T$ is an integer.
5) All symbols of transmitted information are independent of each other, i.e., the sequence $\{\hat{S}_n\}$ is white.
6) The signal is independent of the noise.

It is shown in [1] that under these ideal conditions, a sufficient statistic for MC is the output sequence of a quadrature receiver, which is shown as part of Fig. 3. The in-phase ($r_{I,k}$) and quadrature ($r_{Q,k}$) outputs are

$$\begin{aligned} r_{I,k} &= \int_{(k-1)T}^{kT} r(t)\cos(2\pi f_c t + \theta_c)\,dt \\ &= \frac{AT}{2}\mathrm{Re}\{\hat{S}_k\} + n_{I,k} \end{aligned} \tag{3}$$

$$\begin{aligned} r_{Q,k} &= \int_{(k-1)T}^{kT} r(t)\sin(2\pi f_c t + \theta_c)\,dt \\ &= -\frac{AT}{2}\mathrm{Im}\{\hat{S}_k\} + n_{Q,k} \end{aligned} \tag{4}$$

where

$$n_{I,k} = \int_{(k-1)T}^{kT} n(t)\cos(2\pi f_c t + \theta_c)\,dt \tag{5}$$

and

$$n_{Q,k} = \int_{(k-1)T}^{kT} n(t)\sin(2\pi f_c t + \theta_c)\,dt \tag{6}$$

N symbols of complex data are generated for the time interval $[0, NT]$, i.e.,

$$\begin{aligned} \mathbf{x}_k &= r_{I,k} - jr_{Q,k} \\ &= \frac{AT}{2}\hat{S}_k + \mathbf{n}_k, \qquad k = 1, 2, \cdots, N \end{aligned} \tag{7}$$

where $\mathbf{n}_k = n_{I,k} - jn_{Q,k}$. In this paper, we refer to $r(t)$ and $n(t)$ as *time-domain* signal and noise, respectively, and $\mathbf{x}_k$ and $\mathbf{n}_k$ as *complex-domain* signal and noise, respectively. In a noiseless case, plotting all $\mathbf{x}_k$ on the complex plane will produce a pattern that is the same as a scaled version of the constellation in Fig. 2, assuming that each S_i has appeared in the data at least once.

Denote a group of c possible constellations by

$$\mathcal{I}_l = \{S_{l1}, S_{l2}, \cdots, S_{lM_l}\}, \qquad l = 1, 2, \cdots, c \tag{8}$$

where M_l is the number of points in constellation $\mathcal{I}_l$. Classification within the group of constellations can be considered as a test on the following c hypotheses:

H_l: the underlying constellation is $\mathcal{I}_l$, $l = 1, 2, \cdots, c$.

The maximum-likelihood classification method chooses the hypothesis whose likelihood or log-likelihood function is maximized, i.e.,

$$\begin{aligned} H_l^* &= \arg\max_{H_l} \ln(L(H_l|X_N)) \\ &= \arg\max_{H_l} \ln(p(X_N|H_l)). \end{aligned} \tag{9}$$

In Appendix A, it is shown that when the noise $n(t)$ is white and Gaussian, the log-likelihood function is

$$l(H_l|X_N) = \sum_{k=1}^{N} \ln\left\{\frac{1}{M_l}\sum_{i=1}^{M_l} \exp\left(-\frac{1}{2}|\mathbf{x}_k - AS_{li}|^2\right)\right\}. \tag{10}$$

When all constellations are equally likely, the ML criterion is equivalent to the maximum *a posteriori* criterion; therefore, the ML MC is optimal in the sense of minimum error-rate. However, as pointed out in the Introduction, the ideal conditions assumed by the ML MC typically do not hold in a real-world MC problem. Examples of nonideal conditions are: 1) signal parameters are unknown and 2) the noise is non-Gaussian, e.g., is impulsive. In case 1), the MC may use estimated parameters, assuming that the noise is still Gaussian (as studied in [8]), whereas in case 2), the ML MC will not be applicable theoretically, because the expression for the probability density in (58) relies on the fact that the noise $\mathbf{n}_k$ is Gaussian.

III. A Fuzzy Logic Classifier for Modulation Classification

A. Pattern Classification Using Fuzzy Logic

Before we proceed to use fuzzy logic for modulation classification, we explore how a typical FLS structure [9] fits into a general pattern classification problem. The input to a pattern classifier is usually represented by a vector in a feature space. Suppose there are c possible classes $\mathcal{I}_1, \mathcal{I}_2, \cdots, \mathcal{I}_c$. One way to represent a pattern classifier is in terms of a set of discriminant functions $\{g_i(\mathbf{x}), i = 1, 2, \cdots, c\}$, where $\mathbf{x}$ is a feature vector. The classifier assigns $\mathbf{x}$ to class i if $g_i(\mathbf{x}) > g_j(\mathbf{x})$, $\forall j \neq i$. The feature space is therefore partitioned into c disjoint regions, $\Gamma_1, \Gamma_2, \cdots, \Gamma_c$. These regions can be represented by c characteristic functions defined on the feature space as follows:

$$\mu_{\Gamma_i}(\mathbf{x}) = \begin{cases} 1, & \text{if } \mathbf{x} \in \Gamma_i \\ 0, & \text{otherwise} \end{cases} \qquad i = 1, 2, \cdots, c. \tag{11}$$

Using the expressions in (11), the classification result for $\mathbf{x}$ can be expressed as a fuzzy singleton $B_{\mathbf{x}}$ whose membership function is a function of $\mathbf{x}$, i.e.,

$$\mu_{B_{\mathbf{x}}}(\mathcal{I}_i) = \frac{\mu_{\Gamma_i}(\mathbf{x})}{\sum_{l=1}^{c} \mu_{\Gamma_l}(\mathbf{x})}, \qquad i = 1, \cdots, c. \tag{12}$$

Note that for each $\mathbf{x}$ there is only one value of i for which $\mu_{B_{\mathbf{x}}}(\mathcal{I}_i)$ is nonzero; therefore, this classification output is a *hard decision.*

In our scheme of fuzzy classification, we consider the set of classes $V = \{\mathcal{I}_1, \mathcal{I}_2, \cdots, \mathcal{I}_c\}$ as a universe of discourse on which fuzzy sets are defined to represent the concept of "vague classes."

Definition 2: A fuzzy class is a fuzzy set $G \subset V$ with fuzzy membership function $\mu_G(\mathcal{I})$, where $\mathcal{I} \in V$.

For example,

$$G = \{0.9/\mathcal{I}_1, 0.2/\mathcal{I}_2, 0.1/\mathcal{I}_3\} \tag{13}$$

is a fuzzy-set representation of "similar to class $\mathcal{I}_1$."

Now we generalize the $\mu_{\Gamma_i}(\mathbf{x})$'s in (11) into fuzzy membership functions, i.e., $\mu_{\Gamma_i}(\mathbf{x})$ assumes a value between zero and one and $\mu_{\Gamma_i}(\mathbf{x})$ can be nonzero for multiple values of i for the same $\mathbf{x}$. This makes the classification output in (12) a nonsingleton fuzzy set; therefore, $B_{\mathbf{x}}$ now becomes a *soft decision.*

Since the classifier is now defined by the functions in (12), the classification problem has been translated into the problem of approximating these functions. FLS's can be used as approximators for these functions.

B. Modulation Classification Using Fuzzy Logic

1) Architecture: Because MC can be considered as a pattern classification problem, we follow the definition of fuzzy class to introduce some basic concepts for fuzzy logic modulation classification.

Let U denote the signal space and V denote the set of all modulation types of interest. Generally, U is the set of all possible time-domain waveforms. When we focus on complex-domain data, U is the set of all vectors of complex data. A *fuzzy modulation* is then introduced as a fuzzy class in the universe of all modulations.

Definition 3: A fuzzy modulation is a fuzzy set $B \subset U$ with fuzzy membership function $\mu_B(\mathcal{I})$, where $\mathcal{I} \in V$.

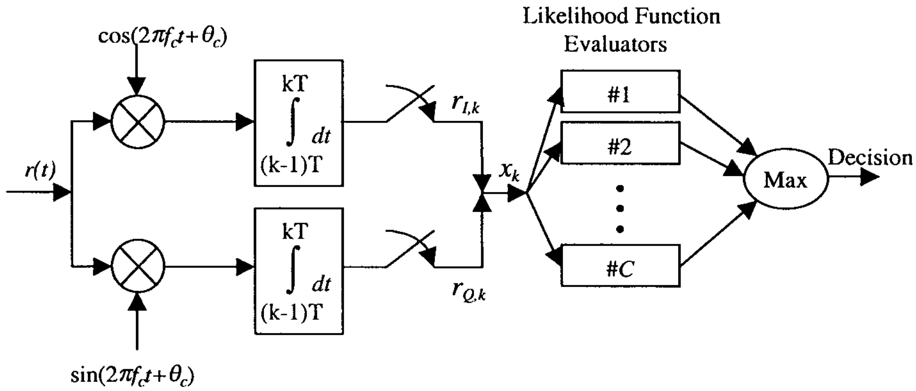


Fig. 3. Block diagram of a maximum-likelihood modulation classifier.

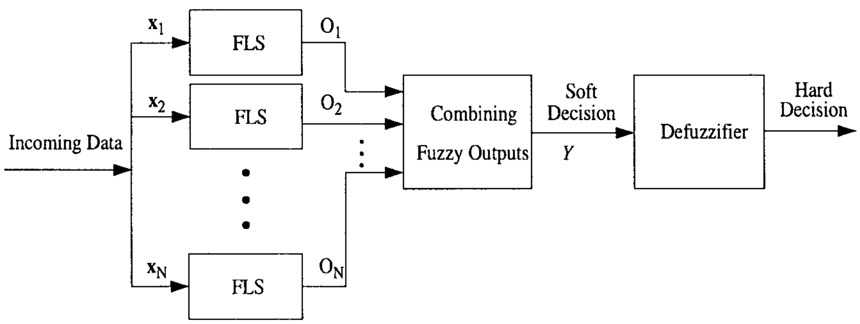


Fig. 4. Architecture of an FL MC.

Definition 4: A fuzzy decision of a modulation classifier is a fuzzy modulation on V.

Although MC and pattern classification are fundamentally similar, they differ in that a typical pattern classifier makes a decision for each vector input, whereas an MC makes one decision for input data from multiple symbols. Concatenating the data from all symbols into a single vector will not make the two problems the same because the resulting vector is then of a variable dimension that changes as more symbols are available, whereas the pattern classifier input has a fixed dimension. Consequently, we use a two-dimensional (2-D) FLS as a core building block that processes one input complex datum at a time and produces a fuzzy output for each input, and the N fuzzy outputs are then combined to obtain an overall result. Fig. 4 illustrates a batch-processing architecture of our FL MC. In this architecture, the FLS has the structure of a typical FLS [9] less a defuzzifier; its output $O_k(k = 1, 2, \cdots, N)$ is a fuzzy modulation, i.e., a fuzzy decision based on a single datum. The fuzzy decisions from all FLS's are then combined by a fuzzy intersection operation to form an overall fuzzy decision, Y. The defuzzifier produces a hard decision from the fuzzy decision. Details about the blocks shown in Fig. 4 are discussed below.

2) Generating Fuzzy Rules: There are two different information sources for generating rules: 1) training data and 2) heuristic interpretations of the ML MC. We consider the latter as more significant because we still assume that there is an underlying probability distribution for the received signal. Heuristic interpretations of the ML MC can help capture the structural information that is difficult to extract with a model-free system. Using this information, we can set up a basic framework for the classifier, and use available training data to adjust the classifier so that it fits into individual working environments. Consequently, the resulting fuzzy logic classifier will be able to mimic the ML MC in basic structure, which ensures that the FL MC can achieve a comparable performance when ideal conditions hold. In the meantime, the FL MC will also be applicable to nonideal conditions because it is much easier to heuristically describe a nonideal condition than model it precisely.

Consider the geometric formation of the complex-domain data. Suppose $\mathcal{I}^*$ is the true constellation; then the data points should form M^* clusters centered at the original constellation points scaled by an amplitude factor. For example, Fig. 1 shows a 100-point Star 8-QAM data set generated at SNR $=$ 20 dB. Observe that the clusters form a geometric pattern that resembles the constellation in Fig. 2(d). Consequently, if each point in $\mathcal{I}_l$ is associated with a cluster, then, whether every input data point falls into at least one of the M_l clusters can be used as an indicator of whether the true constellation is $\mathcal{I}_l$. This observation can be described by the following linguistic rule:

$$\begin{aligned}&\text{IF every received data point } (\mathbf{x}_k) \text{ belongs}\\&\quad\text{to one or more of the } M_l \text{ clusters}\\&\text{THEN the constellation is probably } \mathcal{I}_l.\end{aligned} \tag{14}$$

Note that the clusters in the above rule do not have clear boundaries; therefore, whether a data point "belongs to" a

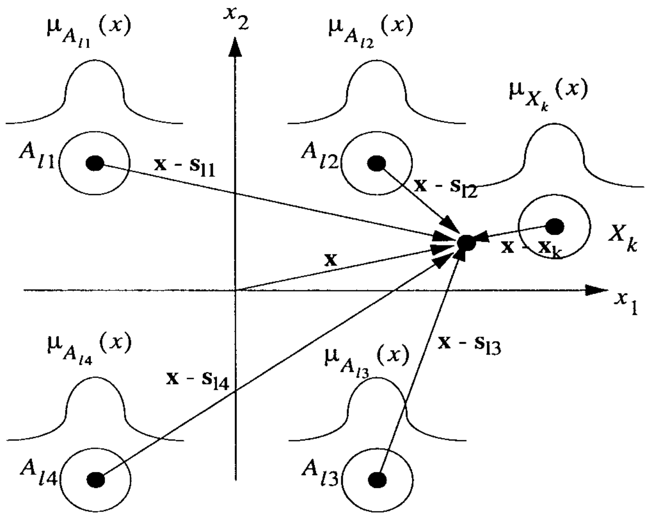


Fig. 5. Two-dimensional fuzzy sets. The membership functions are defined on the complex domain. The value of each membership function at a certain point depends on the distance between this point and the center of the fuzzy set.

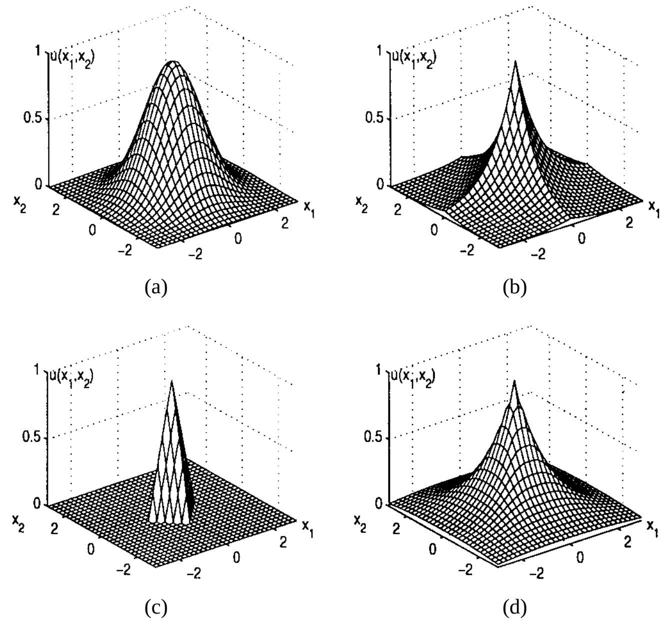


Fig. 6. Two-dimensional membership function. (a) Gaussian kernel and Euclidean distance. (b) Exponential kernel with Hamming distance. (c) Triangular kernel with Hamming distance. (d) Exponential kernel with L_3 distance.

cluster is a vague concept. Fuzzy sets can be used to model such clusters.

Now we need to develop the linguistic rule into a standard form of fuzzy IF–THEN rules. First, the M_l clusters are modeled by M_l fuzzy sets A_{li} each with the following membership function:

$$\mu_{A_{li}}(\mathbf{x}) = \phi_1\left(\frac{d_1(\mathbf{x}, \mathbf{s}_{li})}{\sigma}\right), \qquad i = 1, 2, \cdots, M_l \quad l = 1, 2, \cdots, c \tag{15}$$

where $\phi_1(\bullet)$ is an arbitrary membership function, $d_1(\bullet, \bullet)$ is a distance metric, $\mathbf{x}$ is a complex variable for the membership function, $\mathbf{s}_{li}$ is the cluster center that takes into account the amplitude factor (i.e., if the signal amplitude A is known, then $\mathbf{s}_{li} = AS_{li}$), and σ is a parameter used to control the dispersion of the fuzzy set. Moreover, each input datum $\mathbf{x}_k$ is fuzzified to form an input fuzzy set, X_k, with the following membership function:

$$\mu_{X_k}(\mathbf{x}) = \phi_2\left(\frac{d_2(\mathbf{x}, \mathbf{x}_k)}{\alpha}\right), \qquad k = 1, 2, \cdots, N \tag{16}$$

where $\phi_2(\bullet)$ is an arbitrary membership function, $d_2(\bullet, \bullet)$ is a distance metric, and α is a scale factor used to control the dispersion of the fuzzy set.

Note that unlike a typical FLS, where fuzzy sets are defined and tuned on one-dimensional spaces, we use 2-D fuzzy sets here. To examine their difference, consider the following rules:

$$R^{(1)}: \quad \text{IF } u_1 \text{ is } F_1 \text{ and } u_2 \text{ is } F_2 \text{ THEN } v \text{ is } G \tag{17}$$

$$R^{(2)}: \quad \text{IF } w \text{ is } E_1 \text{ THEN } v \text{ is } G \tag{18}$$

where u_1, u_2 are scalars, and w is 2-D. If we let $E_1 = F_1 \times F_2$, i.e., $\mu_{E_1}(\mathbf{x}) = \mu_{F_1}(x_1) \star \mu_{F_2}(x_2)$, $[\mathbf{x} = (x_1, x_2)]$, then $R^{(1)}$ and $R^{(2)}$ are the same fuzzy implication. On the other hand, it is not always possible to decompose a bivariate membership function into a t-norm combination of two univariate functions; therefore, a fuzzy IF–THEN rule using 2-D fuzzy sets is a more general form. Furthermore, the bivariate membership functions in (15) and (16) are automatically isotropic, i.e., the membership functions are uniform in all directions with respect to their centers. This property is suited for MC because in most cases the distribution of the complex-domain noise is isotropic. A 2-D membership function consists of two factors: the kernel (ϕ) and the distance metric. Fig. 5 illustrates these fuzzy sets.

Example 1: Choices for kernel functions

Triangular:

$$\phi(x) = \begin{cases} 1 - |x|, & \text{if } |x| \le 1 \\ 0, & \text{otherwise} \end{cases} \tag{19}$$

Gaussian:

$$\phi(x) = \exp(-x^2/2) \tag{20}$$

Exponential:

$$\phi(x) = \exp(-|x|). \tag{21}$$

Choices for distance metrics

Euclidean:

$$d(\mathbf{x}, \mathbf{y}) = [(x_1 - y_1)^2 + (x_2 - y_2)^2]^{1/2} \tag{22}$$

Hamming:

$$d(\mathbf{x}, \mathbf{y}) = |x_1 - y_1| + |x_2 - y_2| \tag{23}$$

L_p:

$$d(\mathbf{x}, \mathbf{y}) = [(x_1 - y_1)^p + (x_2 - y_2)^p]^{1/p} \quad p > 0. \tag{24}$$

The combination of the membership functions and distant metrics lead to a large variety of 2-D membership functions. Some examples of these are depicted in Fig. 6. □

In our FL MC, fuzzy sets A_{li} model the additive noise, which is the major cause for why the received data forms clusters around the constellation points; therefore, the selection of $\phi_1(\bullet)$ and σ should be based on the type of the noise and the signal-to-noise ratio, respectively. Fuzzy sets X_k play an auxiliary role in modeling the uncertainties that are normally not accounted for by the A_{li}'s. Examples of these uncertainties are: 1) non-Gaussian noise distribution that cannot be well modeled with a typical membership function and 2) inaccurate constellation points caused by imperfect modulators/demodulators, time-variant signal power, phase drift, etc. Consequently, selection of $\phi_2(\bullet)$ is highly dependent on individual applications.

We can now translate the linguistic rule in (14) into the following fuzzy rules:

$$R^{(li)}: \quad \text{IF } X \text{ is } A_{li}, \text{ THEN } Y \text{ is } B_l \qquad i=1,\cdots,M_l,\, l=1,\cdots,c \tag{25}$$

where X is a fuzzy variable on the complex plane, Y is a fuzzy decision, and B_l is a fuzzy modulation that is the fuzzy-set representation of "probably $\mathcal{I}_l$."

3) Fuzzy Inference: When the rules in (25) are activated by an input $X = X_k$, the output of the rule represents the contribution of datum $\mathbf{x}_k$ to the whole system and the classification decision will be based on the overall contributions from all data points (i.e., $k = 1, 2, \cdots, N$).

Denote the response of $R^{(li)}$ to X_k as $O_{li,k}$, $i = 1, 2, \cdots, M_l$. The membership functions of $O_{li,k}$ are given by compositional fuzzy inference as:

$$\mu_{O_{li,k}}(\mathcal{I}) = \sup_{\mathbf{x}} \mu_{X_k}(\mathbf{x}) \star \mu_{A_{li}\to B_l}(\mathbf{x},\mathcal{I}), \quad l=1,2,\cdots,c \quad k=1,2,\cdots,N,\ i=1,2,\cdots,M_l \tag{26}$$

where $\star$ is a t norm. The $\prod_{l=1}^{c} M_l$ outputs are then combined to form an overall output O_k

$$\mu_{O_k}(\mathcal{I}) = \bigoplus_{l=1}^{c}\bigoplus_{i=1}^{M_l} \sup_{\mathbf{x}}(\mu_{X_k}(\mathbf{x}) \star \mu_{A_{li}\to B_l}(\mathbf{x},\mathcal{I})) \quad k=1,2,\cdots,N \tag{27}$$

where $\oplus$ is usually a t conorm (e.g., the maximum operator) in an FLS or it can be a weighted average in an additive FLS [10]. In our FL MC, we use a weighted average, i.e.,

$$\mu_{O_k}(\mathcal{I}) = \sum_{l=1}^{c}\sum_{i=1}^{M_l} w_{li} \sup_{\mathbf{x}}(\mu_{X_k}(\mathbf{x}) \star \mu_{A_{li}\to B_l}(\mathbf{x},\mathcal{I})) \tag{28}$$

where $w_{li} \in [0, 1]$ is a weight factor associated with rule $R^{(li)}$.

Because the fuzzy rules come from heuristic interpretations of the data formation for each modulation type, we use the following principles in determining w_{li}.

1) The total weight of each constellation is based on the preference of the modulation type.
2) Within each constellation, the weight reflects the preference of each constellation point.

Under the assumption that there is no preference for any modulation type and for any point within a constellation, the following is a choice for the weights that satisfies the above principles:

$$w_{li} = 1/M_l, \qquad i=1,\cdots,M_l,\, l=1,\cdots,c. \tag{29}$$

Hence, we have (see Fig. 4)

$$\mu_{O_k}(\mathcal{I}) = \sum_{l=1}^{c}\sum_{i=1}^{M_l} \frac{1}{M_l} \sup_{\mathbf{x}}(\mu_{X_k}(\mathbf{x}) \star \mu_{A_{li}\to B_l}(\mathbf{x},\mathcal{I})), \quad k=1,2,\cdots,N. \tag{30}$$

4) Combining FLS Outputs: Each O_k represents a highly uncertain decision because it is based on only one datum. Now we use fuzzy intersection to combine all NO_k to obtain an overall output Y, i.e.,

$$\mu_Y(\mathcal{I}) \equiv \mu_{O_1}(\mathcal{I}) \star \mu_{O_2}(\mathcal{I}) \star \cdots \star \mu_{O_N}(\mathcal{I}), \quad l=1,2,\cdots,c. \tag{31}$$

This is a soft decision for MC. A hard decision can be found by a maximum defuzzifier, i.e.,

$$\mathcal{I}^* = \arg\max_{\mathcal{I}} \mu_Y(\mathcal{I}) \tag{32}$$

or in the form of constellation index

$$l^* = \arg\max_{l} \mu_Y(\mathcal{I}_l). \tag{33}$$

Fig. 7 depicts a flow chart of our fuzzy logic classifier.

Because the input is not a singleton, we are actually using a nonsingleton fuzzy logic system [11]; therefore, we call our FL classifier a nonsingleton FL classifier (NSFLC). Nonsingleton FLS has not been widely used because the supremum in (26) does not generally have a closed-form expression; however, it is shown [11] that this supremum is solvable in some special cases.

Example 2: Let all membership functions be Gaussian, i.e.,

$$\mu_{X_k}(\mathbf{x}) = \exp\left(-\frac{|\mathbf{x}-\mathbf{x}_k|^2}{2\alpha^2}\right), \qquad k=1,2,\cdots,N \tag{34}$$

$$\mu_{A_{li}}(\mathbf{x}) = \exp\left(-\frac{|\mathbf{x}-\mathbf{s}_{li}|^2}{2\sigma^2}\right), \qquad i=1,2,\cdots,M_l \tag{35}$$

and let $\mathbf{x} = x_1 + jx_2$, $\mathbf{x}_k = x_{k1} + jx_{k2}$, and $\mathbf{s}_{li} = s_{l1} + js_{l2}$. For each data point, if $\star$ is the arithmetic product and product inference is used, then

$$\begin{aligned}
\mu_{O_{li,k}}(\mathcal{I}) &= \sup_{\mathbf{x}} \mu_{X_k}(\mathbf{x})\mu_{A_{li}}(\mathbf{x})\mu_{B_l}(\mathcal{I})\\
&= \mu_{B_l}(\mathcal{I}) \sup_{x_1,x_2}\left[\exp\left(-\frac{(x_1-x_{k1})^2+(x_2-x_{k2})^2}{2\alpha^2}\right)\right.\\
&\quad \left.\times \exp\left(-\frac{(x_1-s_{i1})^2+(x_2-s_{i2})^2}{2\sigma^2}\right)\right]\\
&= \mu_{B_l}(\mathcal{I}) \sup_{x_1} \exp\left[-\left(\frac{(x_1-x_{k1})^2}{2\alpha^2}+\frac{(x_1-s_{i1})^2}{2\sigma^2}\right)\right]\\
&\quad \times \sup_{x_2} \exp\left[-\left(\frac{(x_2-x_{k2})^2}{2\alpha^2}+\frac{(x_2-s_{i2})^2}{2\sigma^2}\right)\right].
\end{aligned} \tag{36}$$

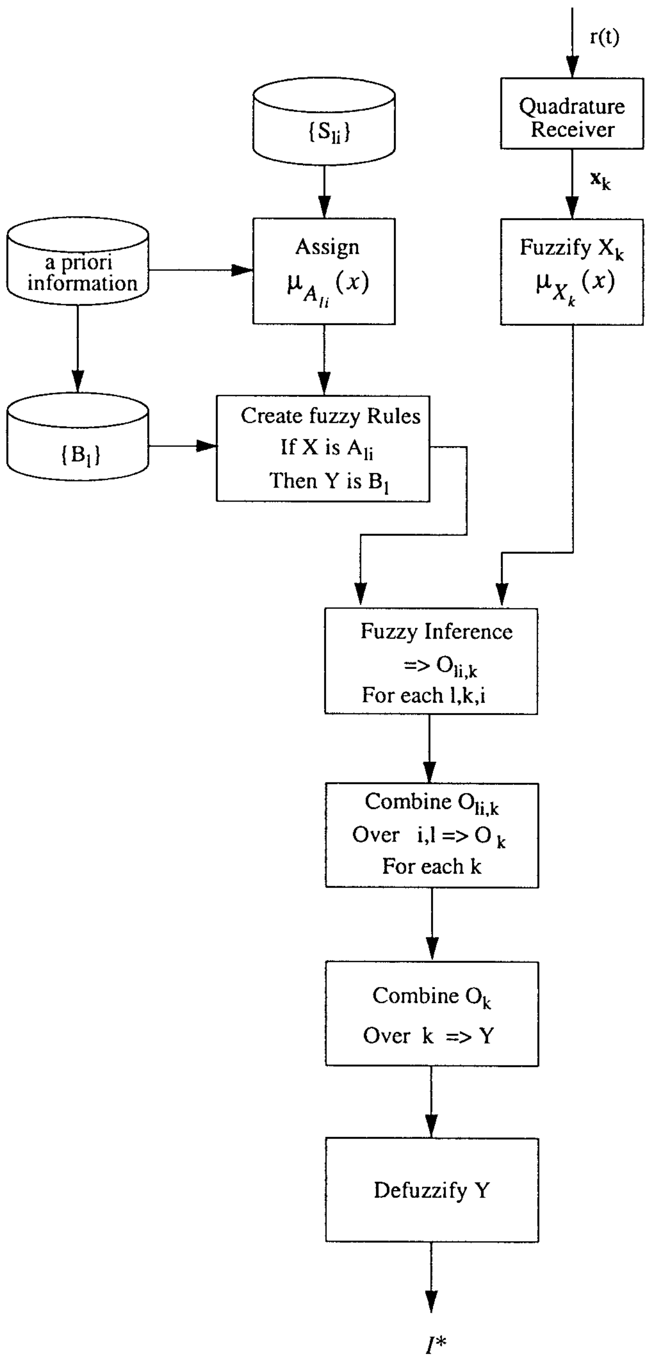


Fig. 7. Flow chart of NSFLC.

According to [11], the supremum is reached at

$$x_1^* = \frac{\sigma^2 x_{k1} + \alpha^2 s_{i1}}{\alpha^2 + \sigma^2} \tag{37}$$

and

$$x_2^* = \frac{\sigma^2 x_{k2} + \alpha^2 s_{i2}}{\alpha^2 + \sigma^2}. \tag{38}$$

By substituting x_1^* and x_2^* into (36), we obtain

$$\mu_{O_{li,k}}(\mathcal{I}) = \mu_{B_l}(\mathcal{I}) \exp\left(-\frac{|\mathbf{x}_k - \mathbf{s}_{li}|^2}{2(\alpha^2+\sigma^2)}\right). \tag{39}$$

Hence, from (30) and (31) the overall inferred result is

$$\begin{aligned}\mu_Y(\mathcal{I}) &= \prod_{k=1}^{N}\left\{\sum_{l=1}^{c}\sum_{i=1}^{M_l}\frac{1}{M_l}\mu_{O_{li,k}}(\mathcal{I})\right\}\\ &= \prod_{k=1}^{N}\left\{\sum_{l=1}^{c}\sum_{i=1}^{M_l}\frac{1}{M_l}\mu_{B_l}(\mathcal{I})\exp\left(-\frac{|\mathbf{x}_k - \mathbf{s}_{li}|^2}{2(\alpha^2+\sigma^2)}\right)\right\}.\end{aligned} \tag{40}$$

Unfortunately, Gaussian membership functions will not be adequate for all situations. Consider a new FLS in which we use a singleton fuzzifier and use the same Gaussian membership functions for the antecedent fuzzy sets as in our NSFLC except that σ is replaced with $\sqrt{\sigma^2+\alpha^2}$. It is easy to see that this new FLS is equivalent to the FLS in our NSFLC; in other words, our NSFLC reduces to a singleton FL MC. This reduces the capability of the NSFLC to model complex types of noise such as a mixture of Gaussian and impulsive noises. On the other hand, (40) reveals an important relation between the NSFLC and the ML MC as we explain in Section III-C.

Example 3: Let $\mu_{A_{li}}(\mathbf{x})$ be Gaussian, as in (35) and μ_{X_k} be exponential with Hamming distance, i.e.,

$$\mu_{X_k}(\mathbf{x}) = \exp\left(-\frac{|x_1 - x_{k1}|}{\sigma}\right)\exp\left(-\frac{|x_2 - x_{k2}|}{\sigma}\right) \quad k = 1, 2, \cdots, N \tag{41}$$

where x_1 and x_2 are the real and imaginary part of $\mathbf{x}$. Lemma 1 in Appendix B shows that the supremum in (27) has a closed-form expression if $\star$ is product. This result will be used when we apply our NSFLC in impulsive noise environments.

C. Relation Between FL and ML Classifiers

Suppose $\alpha = 0$, $\sigma = 1$, $\mathbf{s}_{li} = AS_{li}$, and $\mu_{B_l}(\mathcal{I}) = \delta_k(\mathcal{I}-\mathcal{I}_l)$, where $\delta_k(\mathcal{I}-\mathcal{I}_l) = 1$ if $\mathcal{I} = \mathcal{I}_l$ and $\delta_k(\mathcal{I}-\mathcal{I}_l) = 0$, otherwise. Then, from (32) and (40) we see that

$$\mathcal{I}^* = \arg\max_{\mathcal{I}} \prod_{k=1}^{N}\left\{\frac{1}{M_l}\sum_{i=1}^{M_l}\exp\left(-\frac{|\mathbf{x}_k - AS_{li}|^2}{2}\right)\right\}. \tag{42}$$

Because logarithm is a monotonic function, we can take the logarithm of the right-hand side of (42); therefore

$$\mathcal{I}^* = \arg\max_{\mathcal{I}} \sum_{k=1}^{N}\ln\left\{\frac{1}{M_l}\sum_{i=1}^{M_l}\exp\left(-\frac{|\mathbf{x}_k - AS_{li}|^2}{2}\right)\right\}. \tag{43}$$

By comparing (43) with the combination of (9) and (10), we see that the FL and MC classifiers give the same hard decision. Consequently, we have the following.

Theorem 1: The FL classifier reduces to the ML classifier if α, σ, $\mathbf{s}_{li}$, and B_l are properly selected. □

This is important, because it guarantees that the NSFLC can match the performance of the ML MC when ideal conditions hold.

Compared to the ML MC, the FL classifier has much more flexibility, because it is not dependent on an *a priori* probability model, i.e., it is heuristic. We are free to choose membership functions, t norms, and t conorms to make the classifier adapt to different nonideal environments. Of course, the disadvantage of a heuristic method is also obvious—there is no analytical guarantee of good performance except for the above special case. The only way we can evaluate the performance is through simulations. In the following section, we apply our FL classifier in an impulsive noise environment and compare its performance against the ML classifier.

IV. Using NSFLC in Impulsive Noise Environments

A. Impulsive Noise

It is known that in communication channels there often exists non-Gaussian noise, which is typically some kind of impulsive noise. In the literature, α-stable distributions [12] have been widely used to model such impulsive noise.

First, we introduce the basic probability model for impulsive noises. Time-domain impulsive noise is usually modeled by one-dimensional symmetric α-stable (denoted by $S\alpha S$) distributions [12] with characteristic function in the form

$$\varphi(\omega) = \exp(ja\omega - \gamma|\omega|^\alpha) \tag{44}$$

where $0 < \alpha \leq 2$ is the characteristic exponent, $\gamma > 0$ is the dispersion of the distribution, and $-\infty < a < \infty$ is the location parameter. The location parameter a corresponds to the median of the distribution. The dispersion parameter γ determines the spread of the distribution around its location parameter, similar to the variance of the Gaussian distribution. The distribution in (44) is called *standard* if $a = 0$ and $\gamma = 1$. Unfortunately, no closed-form expression exists for general $S\alpha S$ probability distribution functions other than Cauchy ($\alpha = 1$) and Gaussian ($\alpha = 2$). In general, the density of a standard $S\alpha S$ is given by

$$f_\alpha = \begin{cases} \dfrac{1}{\pi x}\displaystyle\sum_{k=1}^{\infty}\dfrac{(-1)^{k-1}}{k!}\Gamma(\alpha k+1)x^{-\alpha k}\sin\left(\dfrac{k\alpha\pi}{2},\right) \\ \quad \text{If } 0 < \alpha < 1 \\ \dfrac{1}{\pi(1+x^2)}, \qquad \text{If } \alpha = 1 \\ \dfrac{1}{\pi\alpha}\displaystyle\sum_{k=0}^{\infty}\dfrac{(-1)^{k-1}}{2k!}\Gamma\left(\dfrac{2k+1}{\alpha}\right)x^{2k}, \\ \quad \text{If } 1 < \alpha < 2 \\ \dfrac{1}{2\sqrt{\pi}}\exp\left(-\dfrac{x^2}{4}\right), \qquad \text{If } \alpha = 2. \end{cases} \tag{45}$$

When the noise is $S\alpha S$, the equivalent noise output of the quadrature receiver, i.e., the complex-domain noise, has a bivariate isotropic α-stable distribution, which has a characteristic function of the form

$$\varphi(\omega_1, \omega_2) = \exp(j(a_1\omega_1 + a_2\omega_2) - \gamma(\omega_1^2 + \omega_2^2)^{\alpha/2}). \tag{46}$$

Again, α and γ are the characteristic exponent and the dispersion, respectively, and a_1 and a_2 are the location parameters. Note that the marginal distributions of the isotropic stable distribution are $S\alpha S$ with parameters (a_1, γ, α) and (a_2, γ, α). As in the case of univariate $S\alpha S$, no closed-form expression exists for the density function of the bivariate $S\alpha S$ density function. The density function in (46) can be expressed as a power series by using polar coordinates [12].

Although a $S\alpha S$ density behaves approximately the same as a Gaussian density near the origin, its tails decay at a lower rate than the Gaussian tails. The smaller the characteristic exponent α is the heavier the tails of the $S\alpha S$ density.

An important property of the $S\alpha S$ distributions is that only moments of an order less than α exist for the non-Gaussian members. This means that all non-Gaussian $S\alpha S$ distributions have infinite variances. In other words, a true α-stable random sequence carries infinite energy. In this paper, we consider a milder impulsive environment, one in which the noise is modeled as a Gaussian noise plus a small percentage of impulsive noise.

Because the two components of a bivariate isotropic random vector are not independent, we need a special model to generate isotropic noises in our simulations. We will use an isotropic $S\alpha S$ random number generator described in [13].

B. Performance Degradation of ML MC in Impulsive Noise

In practical applications, the actual nature of the impulsive noise is usually unknown, so it is impossible to use a rigorous probabilistic method. A commonly used way for dealing with an unknown probability density function (pdf) is to pretend that it is Gaussian and, therefore, only first- and second-order moments need to be estimated. Unfortunately, because impulsive noise has infinite second-order moments, it is imaginable that this naive method will lead to poor results. A possible remedy is to suppress the impulses in the noise so that the noise is reasonably close to being Gaussian. This can be done by preprocessing the noise samples using a nonlinear function. In the following, we use an example to show the results for both methods.

Because the noise is a mixture of impulsive and Gaussian noise, we will no longer be able to use the noise power for the definition of SNR. In our discussion, we use the signal-to-Gaussian-noise ratio (this will be referred to as SNR later) in conjunction with the percentage of impulsive noise to describe the real SNR. The percentage of impulsive noise is defined as the ratio of the dispersion of the impulsive noise to the standard deviation of the Gaussian noise (multiplied by 100). Note that this percentage of impulsive noise does not represent a typical ratio of impulsive noise amplitude to Gaussian noise amplitude.

We conducted simulations to study the effect of impulsive noise on ML MC in which the ML MC was assumed to have no knowledge of the actual pdf of the noise; it treated the noise as Gaussian and used the maximum-likelihood method to estimate the variance of the noise. The latter was done using a set of noise training samples that contain pure noise. In reality, such data may be obtained from measurements of noise when the channel is silent. The following three modulations were used: 16-QAM, V.29, and 32-QAM. The SNR is 10 dB. Five hundred symbols of noise were used for training and 100 symbols were used for classification.

The impulsive noise suppressor we used is the following zero-memory-nonlinearity (ZMNL), which was suggested by

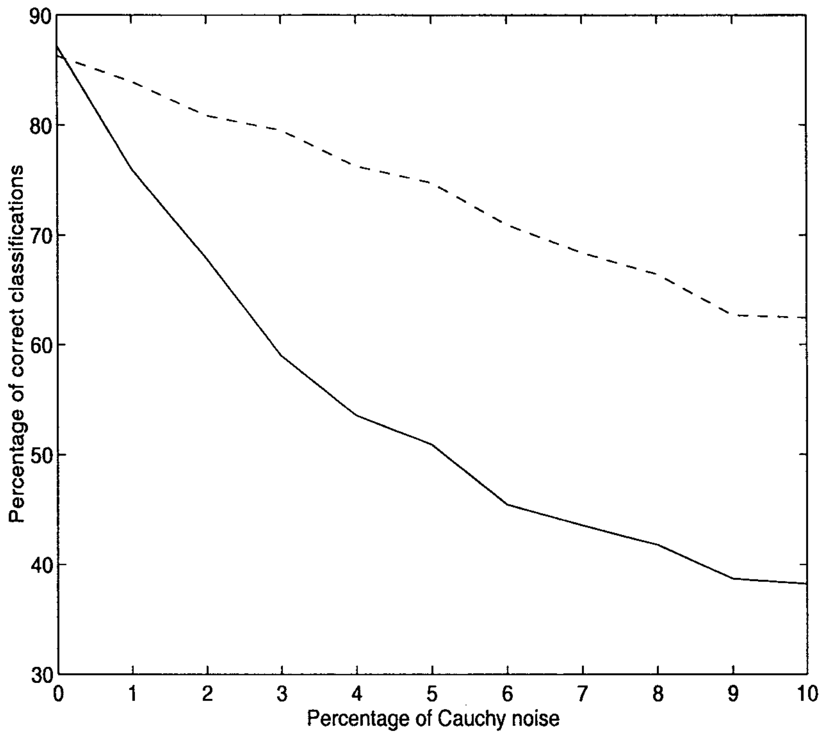


Fig. 8. Performance of ML MC for 16-QAM/V.29/32-QAM classification in a mixture of Gaussian and impulsive noises. Solid: ML MC using a naive ML estimate of noise standard deviation; dash dot: ML MC with ZMNL.

Ljung [14], and was used in [15] and [16] for handling impulsive noise. This ZMNL is

$$\phi(u) = \begin{cases} -\delta \exp(-(u+\delta)^2/2\sigma^2) & u < -\delta \\ u & |u| \le \delta \\ \delta \exp(-(u-\delta)^2/2\sigma^2) & u > \delta \end{cases} \tag{47}$$

in which σ is a tuning parameter that we arbitrarily set equal to $\hat{\sigma}_N$ where

$$\hat{\sigma}_N = \overline{\sigma}/\sqrt{0.7} \tag{48}$$

$$\overline{\sigma} = \text{median}\{|u - \text{median}\{u\}|\} \tag{49}$$

and

$$\delta = 3\hat{\sigma}_N. \tag{50}$$

The outputs of this ZMNL were used to estimate the variance of the noise (the maximum-likelihood estimate of the variance for a zero-mean Gaussian distribution is the sample average of signal power), which, in turn, was used by the Gaussian distribution-based ML MC.

Fig. 8 depicts the classification results for various percentages of Cauchy noise. The results were obtained from 1000 Monte Carlo simulations for each percentage of Cauchy noise. Observe that even a small amount of Cauchy noise can lead to disastrous results for the naive ML MC (solid curve) and that a boost in performance is obtained by using the ZMNL impulse noise suppressor without sacrificing performance when no impulsive noise is present. However, in spite of this improvement, the performance still degrades rapidly as the percentage of Cauchy noise increases. In the next section, we develop a new MC that is based on FL and demonstrate significantly improved performance for it.

C. Performance of NSFLC in Impulsive Noise

Recall that the NSFLC can handle two kinds of uncertainties. Here we utilize this property to model the structure of the additive noise; i.e., we use the fuzzifier to model the uncertainty caused by impulsive noise, and the antecedent fuzzy sets to model the uncertainty of the Gaussian noise. Specifically, we use a Gaussian kernel with Euclidean distance for the antecedent fuzzy sets and use an exponential kernel with Hamming distance for the fuzzifier, i.e.,

$$\mu_{A_{li}}(\mathbf{x}) = \exp\left(-\frac{|\mathbf{x} - \mathbf{s}_{li}|^2}{2\alpha^2}\right), \qquad i = 1, 2, \cdots, M_l \tag{51}$$

and

$$\mu_{X_k}(\mathbf{x}) = \exp\left(-\frac{|x_1 - x_{k1}|}{\sigma}\right) \exp\left(-\frac{|x_2 - x_{k2}|}{\sigma}\right), \quad k = 1, 2, \cdots, N. \tag{52}$$

The reason why we chose the exponential membership function for $\mu_{X_k}(\mathbf{x})$ is that it has a heavy tail that mimics the heavy tail in the pdf of impulsive noise. Fuzzy set A_{li} is used to account for the Gaussian noise; therefore, we use Gaussian functions for its membership functions. As a result, from (30), the FLS output for $\mathbf{x}_k$ is

$$\mu_{O_k}(\mathcal{I}) = \sum_{l=1}^{c} \sum_{i=1}^{M_l} \frac{1}{M_l} \sup_{x_1} \exp\left[-\left(\frac{|x_1 - x_{k1}|}{\alpha} + \frac{(x_1 - s_{li1})^2}{2\sigma^2}\right)\right] \times \sup_{x_2} \exp\left[-\left(\frac{|x_2 - x_{k2}|}{\alpha}\right.\right.$$

$$+ \frac{(x_2 - s_{li2})^2}{2\sigma^2}\bigg)\bigg]\mu_{B_l}(\mathcal{I}),$$
$$l = 1, 2, \cdots, c,\ k = 1, 2, \cdots, N$$
$$i = 1, 2, \cdots, M_l. \tag{53}$$

From Lemma 1 in Appendix B, the supremum is at

$$x_1^* = \begin{cases} s_{li1} - \dfrac{\sigma^2}{\alpha}, & \text{if } x_{k1} < s_{li1} - \dfrac{\sigma^2}{\alpha} \\ s_{li1} + \dfrac{\sigma^2}{\alpha}, & \text{if } x_{k1} > s_{li1} + \dfrac{\sigma^2}{\alpha} \\ x_{k1}, & \text{otherwise} \end{cases} \tag{54}$$

$$x_2^* = \begin{cases} s_{li2} - \dfrac{\sigma^2}{\alpha}, & \text{if } x_{k2} < s_{li2} - \dfrac{\sigma^2}{\alpha} \\ s_{li2} + \dfrac{\sigma^2}{\alpha}, & \text{if } x_{k2} > s_{li2} + \dfrac{\sigma^2}{\alpha} \\ x_{k2}, & \text{otherwise} \end{cases} \tag{55}$$

where the ranges of i, k, and l are as in (53).

The values of σ and α are determined by analyzing the training set of noise $\mathbf{t}^K = \{\mathbf{t}_n,\ n = 1, 2, \cdots, K\}$. As in Section IV-B, we use a ZMNL to process the training data of pure noise in order to have more stable values for σ and α. The following training procedure is proposed for the calculation of σ and α.

1) Form a $2K$-dimensional vector by concatenating the real and imaginary parts of the K complex training data. This vector will be used as if it were a sample population of a one-dimensional distribution. Calculate its standard deviation denoted by D_t.
2) Use the ZMNL, as described in Section IV-B, to produce a new training set $\{t'_n,\ n = 1, 2, \cdots, 2K\}$.
3) Calculate the standard deviation of the new training set and use this value as σ.
4) Compute α, as

$$\alpha = \min\left(\sqrt{\frac{D_t^2 - \sigma^2}{2}},\ \frac{\sum_{n=1}^{2K} |t'_n|}{2K}\right). \tag{56}$$

Equation (56) comes from a heuristic decomposition of the noise mixture. The first term in the $\min$ function is an attempt to estimate the parameter of an exponential distribution by using second-order moments. Note that, if the noise is a mixture of independent Gaussian and exponential noises, where σ^2 is the variance of the Gaussian noise, then $D_t^2 - \sigma^2$ is the variance of the exponential noise portion. Of course, the noise we are considering is impulsive rather than exponential, so there is no guarantee that this estimate is the best. The second term in the $\min$ function limits the value of α to a reasonable level, because the first term can become infinitely large for an impulsive noise.

Note that the NSFLC has a built-in mechanism to guarantee that it works virtually the same as the ML MC (with ZMNL) when no impulsive noise is present. Specifically, it can be seen from (56), that α will be very small if no impulsive noise is present because the standard deviations of the preprocessed and postprocessed training data will be very close. In this case, (54) and (55) give $x_1^* \cong x_{k1}$ and $x_2^* \cong x_{k2}$; therefore, the effects of the exponential membership function in (53) are nullified. Consequently, the results for the NSFLC and ML MC are about the same.

Experiments: Three modulations were used in our simulations: V.29, 16-QAM, and 32-QAM. We used 500 symbols of noise for training, SNR (Gaussian noise) ranging from 6 to 14 dB in 2-dB step size and Cauchy noise ranging from 0% to 5%. For each SNR and Cauchy-noise percentage, 1000 Monte Carlo simulations were run for each of the three signal types. Fig. 9 compares the percentage of correct classification, averaged over the three signal types, of the NSFLC and the ML MC. Note that the plots for SNR = 10 dB compare the result of NSFLC with those shown in Fig. 8. Observe that the NSFLC performs consistently better than the ML MC. The more impulsive noise that is present, the larger is the performance difference.

The experiments demonstrate that the NSFLC is more robust in impulsive noise environments than the ML MC with ZMNL. Moreover, this is accomplished without using *a priori* information on the statistics of the impulsive noise, or knowing in advance whether or not impulsive noise is present. The experiments demonstrate that our NSFLC is capable of dealing with complicated noise environments by using vague information (i.e., the noise is impulsive), something that is difficult to do using probabilistic methods.

V. CONCLUSIONS

We have developed a fuzzy logic modulation classifier that works in nonideal environments in which it is difficult or impossible to use precise probabilistic methods. We began by transforming a general pattern classification problem into one of function approximation, so that FLS's can be used to construct a classifier. After introducing the concepts of fuzzy modulation type and fuzzy decision, we set up an architecture for an NSFLC by using an additive nonsingleton FLS as a core building block. Our NSFLC uses 2-D fuzzy sets that are defined on the complex domain whose membership functions are isotropic so that they are well suited for MC.

We have discovered an important property of our NSFLC, i.e., it reduces to the ML MC when relevant parameters are properly selected. Specifically, when the ideal conditions hold, the known parameters can be used in our NSFLC and doing this makes our NSFLC the same as the ML MC. This guarantees that the NSFLC can match the performance of ML MC when ideal conditions hold. It is interesting to note that our NSFLC is not constructed using a probability model; instead, it is constructed using heuristic interpretations of the clustering formation of complex-domain data.

Compared to the ML MC, the FL classifier has much more flexibility because it is not dependent on an *a priori* probability model, i.e., it is heuristic. We are free to choose membership functions t norms and t conorms to make the classifier adapt to different nonideal environments.

One situation that we have identified as not appropriate for using the ML MC is when impulsive noise is present. We

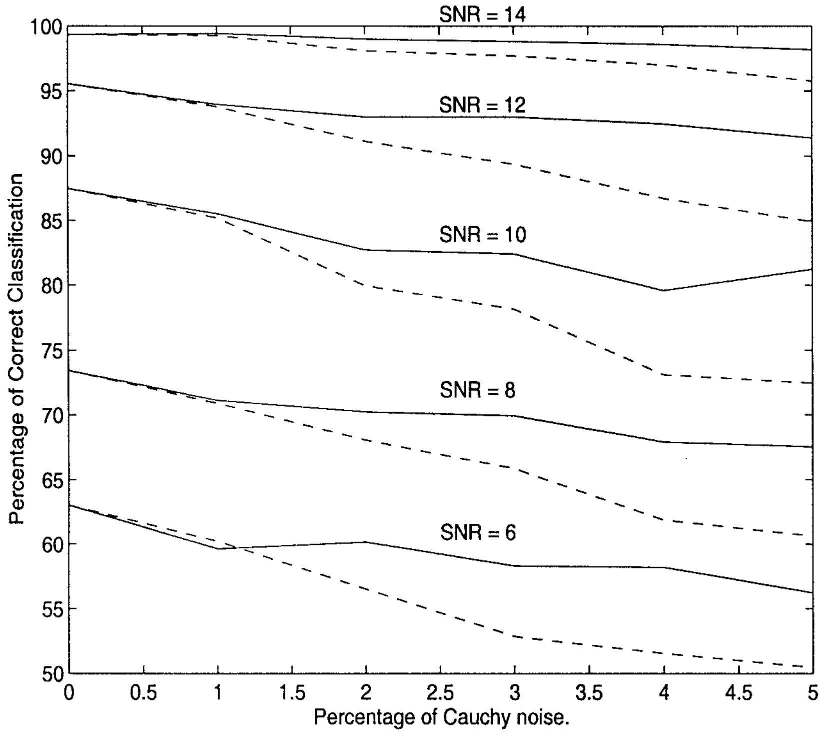


Fig. 9. Performance of ML MC and NSFLC for 16-QAM/V.29/32-QAM classification in a mixture of Gaussian and impulsive noises. Number of symbols $= 100$. Solid: NSFLC; dash: ML MC.

examined the behavior of the maximum-likelihood modulation classifier in a mixture of Gaussian and impulsive noises, and found that the naive way of treating the noise as Gaussian led to severe degradation in performance. This problem could be alleviated by using a zero-memory nonlinear system to preprocess the training set of noisy data; but there was still significant degradation in performance. We applied our fuzzy logic classifier to this situation and found that our NSFLC performed consistently better than the ML MC, and it gave the same performance as the ML MC when no impulsive noise was present. It did this without using any *a priori* information about whether impulsive noise was or was not present. In addition, the performance differences between the NSFLC and the ML MC widened as the percentage of impulsive noise increased.

The major drawback to the NSFLC is that no performance analysis exists for it; however, the same is true for an ML MC in a non-Gaussian noise environment.

Finally, we wish to note that this is only one example of an application of our NSFLC, and that the general form of the NSFLC can lead to many variations, if heuristic information for other applications guides us to select different membership functions, t norms, and t conorms for the NSFLC.

Appendix A
Derivation of Log-Likelihood Function

Using the assumption that the noise is Gaussian and white, it can easily be shown that $n_{I,k}$ and $n_{Q,k}$ are zero-mean white Gaussian sequences each with variance equal to $N_0T/2$ and that they are mutually independent.

Using the total probability formula, we have

$$p(\mathbf{x}_k|H_l) = \sum_{i=1}^{M_l} P(S_{li}|\mathcal{I}_l)p(\mathbf{x}_k|S_{li}) \tag{57}$$

where $P(S_{li}|\mathcal{I}_l)$ is the *a priori* probability of S_{li} in $\mathcal{I}_l$. From (7), we have

$$p(\mathbf{x}_k|S_{li}) = \frac{1}{\pi N_0 T}\exp\left(-\frac{1}{N_0T}\,|\mathbf{x}_k - ATS_{li}/2|^2\right). \tag{58}$$

Assuming that the data from different symbols are independent, then the likelihood function is

$$\begin{aligned} L(H_l|X_N) &= \prod_{k=1}^{N} p(\mathbf{x}_k|H_l) \\ &= \frac{1}{(\pi N_0 T)^N}\prod_{k=1}^{N}\Bigg\{P(S_{li}|\mathcal{I}_l)\sum_{i=1}^{M_l} \\ &\qquad \exp\left(-\frac{1}{N_0T}\,|\mathbf{x}_k - ATS_{li}/2|^2\right)\Bigg\}. \end{aligned} \tag{59}$$

Define SNR as

$$\mathrm{SNR} = \frac{A^2T}{2N_0}. \tag{60}$$

It can be shown (e.g., [1]) that the likelihood function depends only on SNR instead of A, N_0, and T individually. To simplify the notation, we let $T = 2$ and $N_0 = 1$ and use only A to represent SNR. As a result, the log-likelihood function becomes

$l(H_l|X_N)$

$$= \sum_{k=1}^{N} \ln \left\{ P(S_{li}|\mathcal{I}_l) \sum_{i=1}^{M_l} \exp\left(-\tfrac{1}{2} \left|\mathbf{x}_k - AS_{li}\right|^2\right) \right\}. \quad (61)$$

It is common sense that all points in a constellation should be used equally, i.e., $P(S_{li}|\mathcal{I}_l) = 1/M_l$; hence, we then arrive at the expression for $l(H_l|X_N)$ in (10).

APPENDIX B

Lemma 1: The function

$$f(x) = \exp\left[-\left(\frac{|x-t_1|}{\alpha} + \frac{(x-t_2)^2}{2\sigma^2}\right)\right] \quad (62)$$

reaches its maximum at

$$x^* = \begin{cases} t_2 - \dfrac{\sigma^2}{\alpha}, & \text{if } t_1 < t_2 - \dfrac{\sigma^2}{\alpha} \\ t_2 + \dfrac{\sigma^2}{\alpha}, & \text{if } t_1 > t_2 + \dfrac{\sigma^2}{\alpha} \\ t_1, & \text{otherwise.} \end{cases} \quad (63)$$

Proof: Suppose $t_1 < t_2$. It is easy to see [from a sketch of the two components of $f(x)$] that the supremum has to occur at a point $x^* \in [t_1, t_2]$. When $x > t_1$, the derivative of $f(x)$ is

$$f'(x) = -\frac{1}{\sigma^2}\left(x - t_2 + \frac{\sigma^2}{\alpha}\right) f(x). \quad (64)$$

Since $f(x)$ is always positive, $f'(x)$ is negative when $x > t_2 - (\alpha^2/\sigma)$, and positive when $x < t_2 - (\alpha^2/\sigma)$; therefore, if $t_2 - (\alpha^2/\sigma)$ falls in $[t_1, t_2]$, i.e., if $t_1 \le t_2 - (\alpha^2/\sigma)$, then $f(x)$ reaches its maximum at $x = t_2 - (\alpha^2/\sigma)$. On the other hand, if $t_1 > t_2 - (\alpha^2/\sigma)$, then $f'(x)$ is always negative on $[t_1, t_2]$, which means that $f(x)$ is decreasing on $[t_1, t_2]$; therefore, $f(x)$ is maximum at $x = t_1$.

The result for the case when $t_1 > t_2$ can be obtained similarly.

REFERENCES

[1] W. Wei and J. M. Mendel, "Maximum-likelihood classification for digital amplitude-phase modulations," *IEEE Trans. Commun.*, to be published.

[2] Y.-C. Lin and C.-C. Kuo, "Classification of quadrature amplitude modulation (qam) signals via sequential probability ratio test (sprt)," *Signal Processing*, vol. 60, no. 3, pp. 263–280, 1997.

[3] A. Polydoros and K. Kim, "On the detection and classification of quadrature digital modulations in broad-band noise," *IEEE Trans. Commun.*, vol. 38, pp. 1199–1211, Aug. 1990.

[4] C.-Y. Huang and A. Polydoros, "Likelihood methods for MPSK modulation classification," *IEEE Trans. Commun.*, vol. 43, pp. 1144–1154, Feb. 1995.

[5] S.-Z. Hsue and S. Soliman, "Automatic modulation recognition of digitally modulated signals," in *IEEE Military Communicat. Conf.*, Boston, MA, Oct. 1989, pp. 645–650.

[6] ——, "Automatic modulation classification using zero crossing," *Proc. Inst. Elect. Eng.*, vol. 137, no. 6, pp. 459–464, Dec. 1990.

[7] S. Soliman and S.-Z. Hsue, "Signal classification using statistical moments," *IEEE Trans. Commun.*, vol. 40, pp. 908–916, May 1992.

[8] W. Wei, "Classification of digital modulations using constellation analyzes," Ph.D. dissertation, Univ. Southern California, Los Angeles, 1998.

[9] J. Mendel, "Fuzzy logic systems for engineering: A tutorial," *Proc. IEEE*, vol. 83, pp. 345–377, Mar. 1995.

[10] B. Kosko, *Neural Network And Fuzzy Systems, A Dynamical Systems Approach to Machine Intelligence.* Englewood Cliffs, NJ: Prentice-Hall, 1992.

[11] G. Mouzouris and J. Mendel, "Non-singleton fuzzy logic systems: Theory and application," *IEEE Trans. Fuzzy Syst.*, vol. 5, pp. 56–71, Feb. 1997.

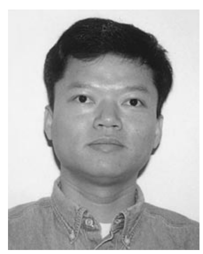
[12] C. Nikias and M. Shao, *Signal Processing with Alpha-Stable Distributions and Applications.* New York: Wiley, 1995.

[13] X. Ma and C. Nikias, "Parameter estimation and blind channel identification in impulsive signal environments," *IEEE Trans. Signal Processing*, vol. 43, pp. 2884–2897, Dec. 1995.

[14] L. Ljung, *System Identification: Theory for the User.* Englewood Cliffs, NJ: Prentice-Hall, 1987.

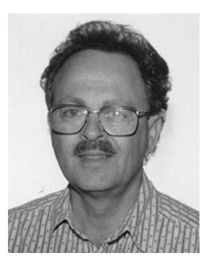
[15] B. Sadler, "Detection in correlated impulsive noise using fourth-order cumulants," *IEEE Trans. Signal Processing*, vol. 44, pp. 2793–2800, Nov. 1996.

[16] A. Swami, "Tde, doa, and related parameter estimation problems in impulsive noise," in *Higher Order Statistics: IEEE Signal Processing Workshop*, Banff, Canada, July 1997, pp. 273–279.



Wen Wei was born in Wenchang, China, on October 1, 1966. He received the B.S. and M.S. degrees in electronic engineering from the Tsinghua University, Beijing, China, in 1986 and 1989, respectively, and the Ph.D. degree in electrical engineering from the University of Southern California, Los Angeles, in 1998.

His research work was on fuzzy logic systems and their applications in modulation classification. He is now with Netscreen Technologies, Inc., Santa Clara, CA.



Jerry M. Mendel (S'59–M'61–SM'72–F'78) received the Ph.D. degree in electrical engineering from the Polytechnic Institute, Brooklyn, NY, in 1963.

Currently, he is a Professor of electrical engineering and an Associate Director of Education at the Integrated Media Systems Center, University of Southern California, Los Angeles, where he has been since 1974. He has published more than 370 technical papers and is the author and/or editor of seven books. His current research interests include type-2 fuzzy logic systems, higher order statistics, and neural networks and their applications to a wide range of signal processing problems.

Dr. Mendel is a Distinguished Member of the IEEE Control Systems Society. He was President of the IEEE Control Systems Society in 1986. Among his awards are the 1983 Best Transactions Paper Award of the IEEE Geoscience and Remote Sensing Society, the 1992 Signal Processing Society Paper Award, and a 1984 IEEE Centennial Medal.