8

Adaptive Fuzzy Neural Filtering for Decision Feedback Equalization and Multi-Antenna Systems

Yao-Jen Chang and Chia-Lu Ho
Department of Communication Engineering, National Central University, Taiwan (R. O. C.)

1. Introduction

1.1 Background

In ordinary channel equalizer and multi-antenna system, many types of detecting methods have been proposed to compensate the distorted signals or recover the original symbols of the desired user [1]-[3]. For channel equalization, transversal equalizers (TEs) and decision feedback equalizers (DFEs) are commonly used as a detector to compensate the distorted signals [2]. It is well-known that a DFE performs significantly better than a TE of equivalent complexity [2]. As to a multi-user-multi-antenna system, adaptive beamforming (BF) detectors have provided practical methods to recover the symbols of the desired user [3]. Many classical optimization algorithms, such as minimum mean-squared error (MMSE) [1]-[4], minimum bit-error rate (MBER) [5]-[9], adaptive MMSE/MBER training methods [6], [10]-[12] and the bagging (BAG) adaptive training method [13], are proposed to adjust the parameters of the above mentioned classical detectors (i.e., TE, DFE and BF).

Due to the optimal nonlinear classification characteristics in the observed space, Bayesian decision theory derived from maximum likelihood detection [15] has been extensively exploited to design the so-called Bayesian TE (BTE) [14]-[15], Bayesian DFE (BDFE) [16]-[17] and Bayesian BF (BBF) [18]-[19]. The bit-error rate (BER) or symbol-error rate (SER) results of Bayesian-based detectors are often referred to as the optimal solutions, and are extremely superior to those of MMSE, MBER, adaptive MMSE (such as least mean square algorithm [1]), adaptive MBER (such as linear-MBER algorithm [6]) or BAG-optimized detector. The BTE, BDFE and BBF can be realized by the radial basis functions (RBFs) [14], [17], [19]-[23]. Classically, the RBF TE, RBF DFE or RBF BF is trained with a clustering algorithm, such as *k*-means [14], [17], [24] and rival penalized competitive learning (RPCL) [25]-[31]. These clustering techniques can help RBF detectors find the center vectors (also called center units or centers) associated with radial Gaussian functions.

1.2 Motivation of FSCFNN equalization with decision feedback

The number of radial Gaussian functions of a RBF TE, i.e., the number of hidden nodes or the number of RBF nodes, can be obtained from a prior knowledge. The mathematical operation with respect to the equalizer order and the channel order can readily determine the number of hidden nodes [14, 16, 20]. However, if the channel order or equalizer order increases linearly, the number of hidden nodes in RBF TE grows exponentially, so do the computation and hardware complexity [20]. Trial-and-error method is an alternative way to determine the number of hidden nodes of RBF.

Except the clustering RBF detectors, there are other types of nonlinear detectors, such as multilayer perceptrons (MLPs) [32]-[38], adaptive neuro fuzzy inference system (ANFIS) [39]-[41] and self-constructing recurrent fuzzy neural networks (SCRFNNs) [42]-[44]. Traditionally, MLP and ANFIS detectors are trained by the back-propagation (BP) learning [32], [34], [35], [38], [40]. However, due to the improper initial parameters of MLP and ANFIS detectors, the BP learning often results in an occurrence of local minima which can lead to bad performance [38]. Recently, evolution strategy (ES) has been also used to train the parameters of MLP and ANFIS detectors [36], [41]. Although the ES inherently is a global and parallel optimization learning algorithm, tremendous computational costs in the training process make it impractical in modern communication environments. In addition, the structures (i.e., the numbers of hidden nodes) of MLP and ANFIS detectors must be fixed and assigned in advance and determined by trial-and-error method.

In 2005, the SCRFNN detector and its another version, i.e., self-constructing fuzzy neural network (SCFNN), have been applied to the channel equalization problem [43]-[44]. Specifically, the SCRFNN or SCFNN equalizers perform both self-constructing process and BP learning process simultaneously in the training procedure without the knowledge of channel characteristics. Initially, there are no hidden nodes (also called fuzzy rules hereinafter) in the SCRFNN or SCFNN structure. All of the nodes are flexibly generated online during the self-constructing process that not only helps automate structure modification (i.e., the number of hidden nodes is automatically determined by the self-constructing algorithm instead of the trial-and-error method) but also locates good initial parameters for the subsequent BP algorithm. The BER or SER of the SCRFNN TE and SCFNN TE thus is extremely superior to that of the classical BP-trained MLP and ANFIS TEs, and is close to the optimal Bayesian solution. Moreover, the self-constructing process of SCRFNN and SCFNN can construct a more compact structure due to setting conditions to restrict the generation of a new hidden node, and hence SCRFNN and SCFNN TEs results in lower computational costs compared to traditional RBF and ANFIS TEs.

Although the SCRFNN TE and SCFNN TE in [43-44] have provided a scheme to obtain satisfactory BER and SER performance with low computational complexity, it doesn't take advantage of decision feedback signals to improve the detecting capability. In Section 2, a novel DFE structure incorporated with a fast SCFNN learning algorithm is presented. We term it as fast SCFNN (FSCFNN) DFE [58]. FSCFNN DFE is composed of several FSCFNN TEs, each of which corresponding to one feedback input vector. Because the feedback input vector occurs independently, only one FSCFNN TE is activated to decide the estimated symbol at each time instant. Without knowledge of channel characteristics, the improvement over the classical SCRFNN or SCFNN TE can be achieved by FSCFNN DFE in terms of BER, computational cost and hardware complexity.

In modern communication channels, a time-varying fading caused by Doppler effect [33], [37], [49] and a frequency offset casued by Doppler effect and/or mismatch between the frequencies of the transmitter and receiver oscillators are usually unavoidable [45]. Moreover, a phase noise [45] also may exist due to distorted transmission environment and/or imperfect oscillators. Therefore, these distortions need to be compensated at the receiver to avoid a serious degradation. To the best of our knowledge, most of the work in the area of nonlinear TE or DFE over the past few decades focuses on the time-invariant channels. Therefore, the simulations of the FSCFNN DFE and the other nonlinear equalizing methods will be investigated in Section 2.3 under the linear and nonlinear channels with time-invariant or time-varying environment.

1.3 Motivation of adaptive RS-SCFNN beamformer

As mentioned in Section 1.1, for multi-antenna systems, classical adaptive BFs are designed based on the MMSE or MBER algorithm [1], [3], [6], [8], [11], [19]. This classical MMSE or MBER beamforming design requires that the number of users supported is no more than the number of receiving antenna elements [19], [46]. If this condition is not met, the multi-antenna system is referred to as overloaded or rank-deficient. Moreover, BER performance of MMSE and MBER beamformers in the rank-deficient system will be very poor. Due to the nonlinear classification ability as mentioned in Section 1.1, the BBF realized by a RBF detector has shown a significant improvement over the MMSE and MBER ones, especially in the rank-deficient multi-antenna system [19], [47], [48]. Recently, a symmetric property of BBF [8] is exploited to design a novel symmetric RBF (SRBF) BF [47]-[48]. This SRBF BF can obtain better BER performance and simpler training procedure than the classical RBF one. Differing from the clustering method, the MBER method [47] based on a stochastic approximation of Parzen window density estimation also can be used to train the parameters of RBF as demonstrated in [47]. Unfortunately, RBF BF trained by an enhanced *k*-means clustering [48] or the MBER algorithm still needs large amounts of hidden nodes and training data to achieve satisfactory BER performance.

To the best of our knowledge, all existing SCFNN detectors are designed for single-user single-antenna assisted systems. In Section 3, we thus propose to incorporate the SCFNN structure into multi-antenna assisted beamforming systems with the aid of a symmetric property of array input signal space. This novel BF is called symmetric SCFNN (S-SCFNN) BF. The training procedure of this S-SCFNN also contains self-constructing and parameter training phases. Although S-SCFNN BF has better BER performance and lower BF complexity than the standard SCFNN one, the BF complexity is still huge at low signal-to-noise (SNR) ratios. Thus, a simple inhibition criterion is added to the self-constructing training phase to greatly reduce the BF complexity, and this low-complexity S-SCFNN is called reduced S-SCFNN (RS-SCFNN). The simulation results have shown that the RS-SCFNN BF extremely outperforms the BFs incorporated with MMSE, MBER, SRBF and the classical SCFNN detectors in the rank-deficient multi-antenna assisted systems. Besides, the proposed SCFNN BF can flexibly and automatically determine different numbers of hidden nodes for various SNR environments, but, as discussed in Section 3.3, the RBF detector must assign hidden node's numbers as a fix constant for various SNR environments before training. Although the RBF BF can also assign the various numbers of hidden nodes for different SNRs, it needs huge manpower to achieve this goal.

2. Self-constructing fuzzy neural filtering for decision feedback equalizer

Classical equalizers, such as a transversal equalizer (TE) and a decision feedback equalizer (DFE), usually employ linear filters to equalize distorted signals. It has been shown that the mean square error (MSE) for a DFE is always smaller than that of a TE, especially when the channel has a deep spectral null in its bandwidth [2]. However, if the channel has severely nonlinear distortions, classical TE and DFE perform poorly. Generally speaking, the nonlinear equalization techniques proposed to address the nonlinear channel equalization problem are those presented in [14], [16], [17], [22], [32], [35], [39], [44], [54]. Chen *et al.* have derived a Bayesian DFE (BDFE) solution [16], which not only improves performance but also reduces computational cost compared to the Bayesian transversal equalizer (BTE). Based on the assumption that the channel order n_h has been known, i.e., the channel order n_h has been successfully estimated before detection process, a radial basis function (RBF) detection can realize the optimal BTE and BDFE solutions [14], [16]. However, as the channel order or/and the equalizer order increases, the computational cost and memory requirement will grow exponentially as mentioned in Section 1.2.

A powerful nonlinear detecting technique called fuzzy neural network (FNN) can make effective use of both easy interpretability of fuzzy logics and superior learning ability of neural networks, hence it has been adopted for equalization problems, e.g. an adaptive neuro fuzzy inference system (ANFIS)-based equalizer [39] and a self-constructing recurrent FNN (SCRFNN)-based equalizer [44]. Multilayer perceptron (MLP)-based equalizers [32], [35] are another kind of detection. Both FNN and MLP equalizers do not have to know the channel characteristics including the channel order and channel coefficients. For ANFIS and MLP nonlinear equalizers, the structure size must be fixed by trial-and-error method in advance, and all parameters are tuned by a gradient descent method. As to SCRFNN equalizer, it can simultaneously tune both the structure size and the parameters during its online learning procedure. Although the SCRFNN equalizer has provided a scheme to automatically tune the structure size, it doesn’t derive an algorithm to improve the performance with the aid of decision feedback symbols. Thus, a novel adaptive filtering based on fast self-constructing neural network (FSCFNN) algorithm has been proposed with the aid of decision feedback symbols [58].

2.1 Equalization model with decision feedback

A general DFE model in a digital communication system is displayed in Figure 2.1 [2]. A sequence, {$s(n)$}, extracted from a source of information is transmitted and the transmitted symbols are then corrupted by channel distortion and buried in additive white Gaussian noise (AWGN). Then, the channel with nonlinear distortion is modeled as

$$r(n) = g(\hat{r}(n)) + v(n) = g\left(\sum_{i=0}^{n_h} h_i s(n-i)\right) + v(n)\,, \tag{2.1}$$

where $g(\cdot)$ is a nonlinear distortion, h_i is the channel coefficient of the linear FIR channel $\hat{r}(n)$ with length $n_h + 1$ (n_h is also called channel order), $s(n)$ is the transmitted symbol at the time instant n, and $v(n)$ is the AWGN with zero mean and variance σ_v^2 . The standard DFE is characterized by the three integers N_f, N_b and d known as the feedforward order, feedback order, and decision delay, respectively. We define the feedforward input vector at the time instant n as the sequence of the noisy received signals {$r(n)$} inputting to the DFE, i.e.,

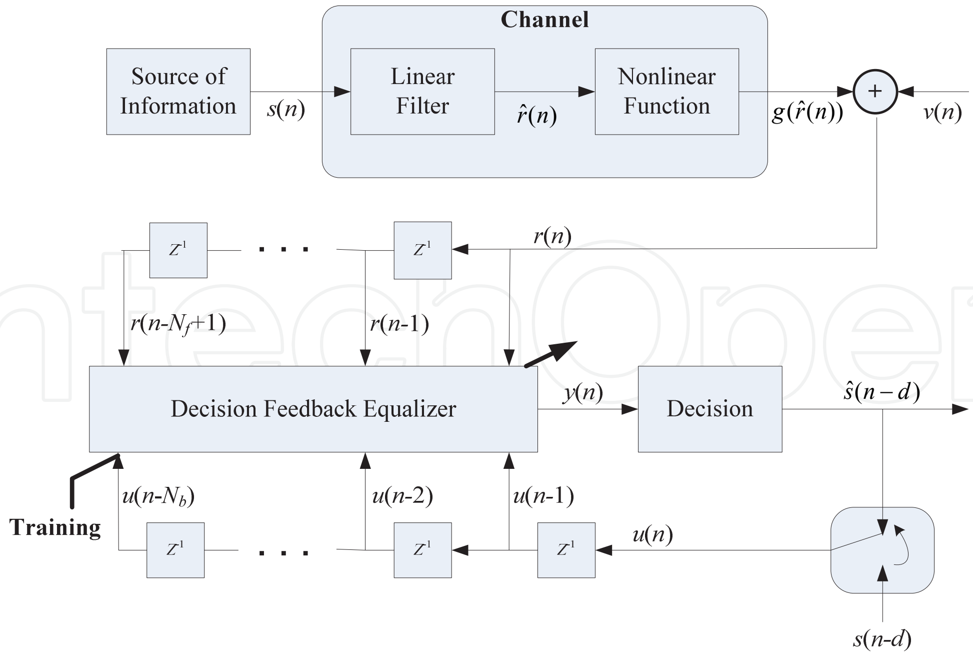

Fig. 2.1 General equalization model with decision feedback

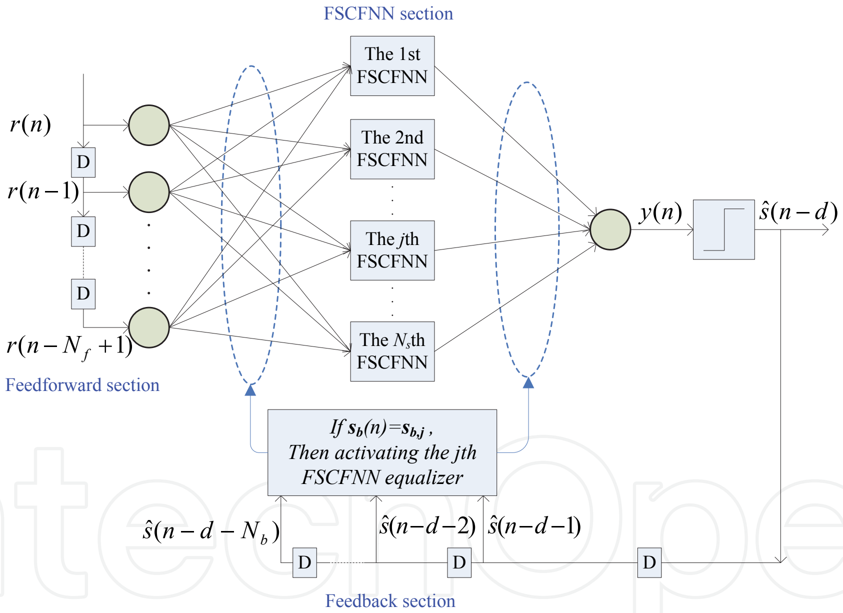

Fig. 2.2 FSCFNN DFE structure

$$\mathbf{s}_f(n) = [r(n),r(n\text{-}1),\ldots,r(n\text{-}N_f\text{+}1)]^T. \qquad (2.2)$$

The feedback input vector that inputs into the DFE at the time instant n can be defined as the decision sequence, i.e.,

$$\mathbf{s}_b(n) = [u(n),u(n\text{-}1),\ldots,u(n\text{-}N_b\text{+}1)]^T = [\,\hat{s}\ (n\text{-}d\text{-}1), \hat{s}\ (n\text{-}d\text{-}2),\ldots, \hat{s}\ (n\text{-}d\text{-}N_b)]^T. \qquad (2.3)$$

The output of the DFE is $y(n)$ and it is passed through a decision device to determine the estimated symbol $\hat{s}$ $(n\text{-}d)$ of the desired symbol $s(n\text{-}d)$.

2.2 Adaptive FSCFNN decision feedback equalizer

A. Novel DFE design

The FSCFNN DFE structure shown in Figure 2.2 [58] consists of feedforward section, feedback section and FSCFNN section. The feedforward and feedback sections contain the signal vectors $\mathbf{s}_f(n)$ and $\mathbf{s}_b(n)$, where the notations N_f and N_b have been defined in Section 2.1. We assume that the FSCFNN section contains N_s FSCFNN equalizers. The transmitted sequence $\{s(n)\}$ is assumed to be an equiprobable and independent binary sequence taking +1 or –1 in this section. Thus, the estimated symbol can be easily determined by

$$\hat{s}(n-d)=\begin{cases} s_1 \equiv +1, if\ \ y(n)\geq 0 \\ s_2 \equiv -1, if\ \ y(n)<0 \end{cases} \tag{2.4}$$

Usually, the feedback input vector $s_b(n)$ in the training mode is formed by the known training symbols, i.e.,

$$\mathbf{s}_b(n) = [s(n\text{-}d\text{-}1),s(n\text{-}d\text{-}2),\ldots,s(n\text{-}d\text{-}N_b)]^T. \tag{2.5}$$

Without loss of generality, we can select $N_f = d + 1$, where d is chosen by a designer. Increasing d may improve performance, but reducing d reduces equalizer complexity. In this section, we set $d = 1$.

It is clear that the channel equalization process can be viewed as a classification problem, which seeks to classify observation vectors into one of the classes. Thus, we apply the principle of classification to designing the FSCFNN DFE. Suppose, at each time instant n, there are N_t transmitted symbols that will influence the decision output $y(n)$ of FSCFNN DFE:

$$\mathbf{s}_t(n) = [s(n),\ldots,s(n\text{-}d\text{-}1),\ldots,s(n\text{-}d\text{-}N_b),\ldots,s(n\text{-}N_t\text{+}1)]^T, \tag{2.6}$$

where the value $N_t \geq d + N_b + 1$ is determined by the channel order n_h. Since we assume that the FSCFNN DFE doesn't estimate the channel order n_h in advance, the value N_t will be unknown. Obviously, the sequence $\mathbf{s}_t(n)$ at the time instant n contains the correct feedback input vector $s_b(n)$. Moreover, as $s_t(n)$ sequentially going through a channel, the feedforward input vector $\mathbf{s}_f(n)$ is then generated. Clearly, the set of $\mathbf{s}_t(n)$ can be partitioned into 2^{N_b} subsets due to $\mathbf{s}_b(n)$ involving 2^{N_b} feedback states, denoted as $\mathbf{s}_{b,j}$, $j = 1 \sim 2^{N_b}$. Therefore, the set $R_d = \{\mathbf{s}_f(n)\}$ associated with feedforward input vectors can be also divided into 2^{N_b} subsets according to the feedback states:

$$R_d = \bigcup\nolimits_{1\leq j\leq 2^{N_b}} R_{d,j} \tag{2.7}$$

where $R_{d,j} = \{\mathbf{s}_f(n)\,|\,\mathbf{s}_b(n)\text{=}\mathbf{s}_{b,j}\}$, . Since each feedback state $\mathbf{s}_{b,j}$ occurs independently, the FSCFNN DFE uses $N_s = 2^{N_b}$ FSCFNN detectors to separately classify the N_s feedforward input subsets $R_{d,j}$ into 2 classes. Thus, for the feedforward input vectors belonging to $R_{d,j}$, the jth FSCFNN detector corresponding to $R_{d,j}$ will be exploited as shown in Figure 2.2 to further classify subset $R_{d,j}$ into 2 subsets according to the value of $s(n\text{-}d)$, i.e.,

$$R_{d,j} = \bigcup\nolimits_{1\leq i\leq 2} R_{d,j}^{(i)} \tag{2.8}$$

where $R_{d,j}^{(i)} = \{\boldsymbol{s_f}(n) \,|\, (\boldsymbol{s_b}(n) = \boldsymbol{s_{b,j}}) \wedge (s(n-d) = s_i)\}$, i = 1, 2. Thus, a feedfoward input vector with $\boldsymbol{s}_{b,j}$ being its feedback state can be equalized by solely observing subset $R_{d,j}$ corresponding to the jth FSCFNN detector.

B. Learning of the FSCFNN with decision feedback

If the FSCFNN DFE (Figure 2.2) receives a feedforward input vector $s_f(n)$ with $\boldsymbol{s}_b(n)$=$\boldsymbol{s}_{b,j}$ at n, the j-th FSCFNN detection will be activated as mentioned above. The structure of this j-th FSCFNN detection is shown in Figure 2.3. The output of the j-th FSCFNN detector is defined as

$$O_j(n) = \sum\nolimits_{k=1}^{K_j(n)} \omega_{k,j}(n) O_{k,j}^{(3)}(n) \tag{2.9}$$

with

$$O_{k,j}^{(3)}(n) = \prod\nolimits_{p=1}^{N_f} \exp\left(-\frac{(O_p^{(1)}(n) - m_{kp,j}(n))^2}{2\sigma_{kp,j}^2(n)}\right) \tag{2.10}$$

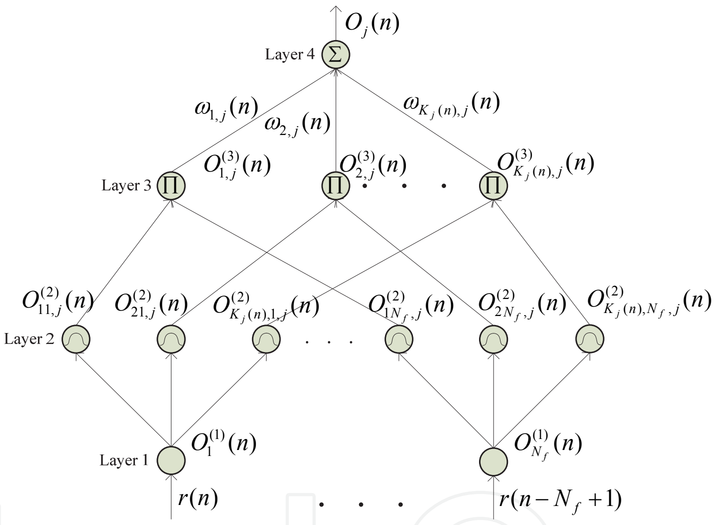

Fig. 2.3 Structure of the j-th FSCFNN

where $K_j(n)$ is the number of rule in the j-th FSCFNN detector, $w_{k,j}(n)$ is the consequent weight of the k-th rule in the j-th FSCFNN detector, and $m_{kp,j}(n)$ and $\sigma_{kp,j}(n)$ are the mean and standard deviation of the Gaussian membership function $O_{k,j}^{(3)}(n)$ corresponding to k-th rule in the j-th FSCFNN detector. Finally, the output value of FSCFNN DFE (Figure 2.2) at n is expressed as $y(n) = O_j(n)$.

Based on the self-constructing and parameter learning phases in SCRFNN structure [44], a fast learning version [58] has been proposed for FSCFNN DFE to further reduce the computational cost in the training period. Similarly, there are no fuzzy rules initially in each FSCFNN detector. As $\boldsymbol{s}_b(n)$=$\boldsymbol{s}_{b,j}$ at n, the proposed fast self-constructing and parameter learning phases are performed simultaneously in the j-th FSCFNN structure. In the self-constructing learning phase, we use two measures to judge whether to generate the hidden node or not. The first is the measure of the system error $\varepsilon = s(n-d) - \hat{s}(n-d)$ for considering the generalization performance of the overall network. The second is the measure of the maximum membership degree $\mu_{\max} = \max_k O_{k,j}^{(3)}(n)$. Consequently, for a feedforward input vector $s_f(n)$ with $\boldsymbol{s}_b(n)=\boldsymbol{s}_{b,j}$, the fast learning algorithm contains three possible scenarios to perform the self-constructing and parameter learning phases:

a. $\varepsilon \neq 0$ and $\mu_{\max} \leq \mu_{\min}$: It shows that the network obtains an incorrect estimated symbol and no fuzzy rule can geometrically accommodate the current feedforward input vector $\boldsymbol{s}_f(n)$. Our strategy for this case is to try improving the entire performance of the current network by adding a fuzzy rule to cover the vector $s_f(n)$, i.e., $K_j(n+1) = K_j(n) + 1$. The parameters associated with the new fuzzy rule in the antecedent part of the j-th FSCFNN are initialized the same as those of SCRFNN:

$$\boldsymbol{m}_{new,j}(n) = \boldsymbol{s}_{\boldsymbol{f}}(n), \quad \boldsymbol{\sigma}_{new,j}(n) = \sigma \cdot \boldsymbol{I} \tag{2.11}$$

where σ is set as 0.5 in this chapter.

b. $\varepsilon \neq 0$ and $\mu_{\max} > \mu_{\min}$: This means that the network obtains an incorrect estimated symbol but at least one fuzzy rule can accommodate the vector $\boldsymbol{s}_f(n)$. Thus the parameter learning can be used here to improve the performance of the network and no fuzzy rule should be added.

c. $\varepsilon = 0$ This means that the network has obtained a correct estimated symbol. Thus, it is unnecessary to add a rule, but the parameter learning is still performed to optimize the parameters.

As to the parameter learning used in the above scenarios (a)-(c), any kind of gradient descent algorithms can be used to update the parameters.

2.3 Simulation results

The performance of the FSCFNN DFE will be examined in time-invariant and time-varying channels in this sub-section. Table 2.1 shows the transfer functions of the simulated time-invariant channels. For comparisons, SCRFNN [44], ANFIS DFE with 16 rules [39], RBF DFE [16] and BDFE [16], [17] are added in the experiments. The parameters N_f = 2 and N_b = 2 are chosen for the equalizers with decision feedback. The SCRFNN equalizer with 2 taps is performed without decision feedback as mentioned above. The RBF DFE with the k-means algorithm works under the assumption of the perfect knowledge of the channel order [16], [20]. The performance is determined by taking an average of 1000 individual runs, each of which involves a different random sequence for training and testing. The testing period for each individual run has a length of 1000. The size of training data will be discussed later.

Channel	Transfer function
Channel A [21]	$H(z) = 0.348z^0 + 0.870z^{-1} + 0.348z^{-2}$ $H_A(z) = H(z) + 0.2H^2(z)$
Channel B [14] [37]	$H_B(z) = 0.348z^0 + 0.870z^{-1} + 0.348z^{-2}$

Table 2.1 Transfer functions of the simulated channels

A. Time-invariant channel

Several comparisons are made with various methods for the nonlinear time-invariant channel A. Figure 2.4 shows the BER performance and average numbers of fuzzy rules needed in computation for FSCFNN DFE under various values $\mu_{\min}$ in a different length of training. Clearly, the results of BER performance are similar if $\mu_{\min} > 0.05$, but the numbers of rules are increased as $\mu_{\min}$ grows. Moreover, it shows that the needed training data size for FSCFNN DFE is about 300. Figure 2.5 demonstrates the BER performance and average numbers of rules for various methods. The SCRFNN with $\mu_{\min} = 0.00003$ is used in this plot. The FSCFNN DFEs with $\mu_{\min} = 0.5$ and $\mu_{\min} = 0.05$ are respectively denoted as FSCFNN DFE(A) and FSCFNN DFE(B) in this plot. Obviously, the FSCFNN DFEs are superior to the other methods. Because we want to obtain satisfactory BER performance, both 400 training data size for various methods and $\mu_{\min} = 0.05$ for FSCFNN DFE will be set in the following simulations.

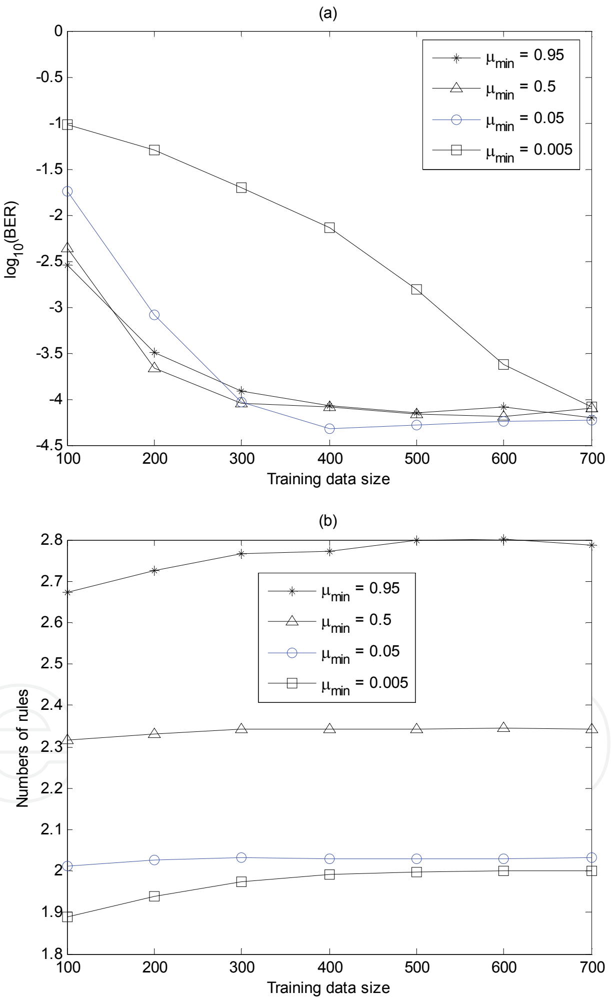

Fig. 2.4 Performance of FSCFNN DFE for various values $\mu_{\min}$ with a different length of training in the time-invariant channel A as SNR = 18 dB: (a) BER (b) Numbers of fuzzy rules

Figure 2.6 illustrates the performance of various methods at different SNRs. Note that the BERs in SNR = 20 dB are gained by averaging 10000 runs for accurate consideration.

Without knowledge of the channel, FSCFNN DFE improves the BER performance close to optimal BDFE solutions in satisfactory low numbers of rules.

Figures 2.7 & 2.8 show examples of the fuzzy rules generated by SCRFNN equalizer and FSCFNN DFE as SNR = 18 dB. The channel states and decision boundaries of the optimal solution are also plotted. The *j*-th FSCFNN detector can geometrically cluster the feedforward input vectors associated with $s_b(n)=s_{b,j}$, and in Figure 2.8, only 2 fuzzy rules in each FSCFNN are generated. Because the SCRFNN equalizer needs to cluster the whole input vectors, 4 fuzzy rules are created to attain this purpose (Figure 2.7). Therefore, FSCFNN DFE requires lower computational cost than SCRFNN in the learning or equalization period. In Figure 2.8, the optimal decision boundaries for four types of feedforward input vector subsets $R_{d,j}$ are almost linear, but the optimal decision boundary in SCRFNN is nonlinear. It also implies that classifying the distorted received signals into 2 classes in FSCFNN DFE is easier than that in SCRFNN equalizer. This is the main reason that the BER performance of FSCFNN DFE is superior to that of the classical SCRFNN equalizer.

B. Time-varying channel

The FSCFNN DFE is tested on time-varying channel environments. The following linear multipath time-varying channel model is used:

$$r(n) = h_1(n)s(n) + h_2(n)s(n\text{-}1) + h_3(n)s(n\text{-}2) + v(n) \tag{2.15}$$

where $h_i(n)$ represents the time-varying channel coefficients. We use a second-order low-pass digital Butterworth filter with cutoff frequency f_d to generate a time-varying channel [49], [55], where the value f_d determines the relative bandwidth (fade rate) of the channel time variation. The input to the Butterworth filter is a white Gaussian sequence with standard deviation ξ = 0.1. Then, a colored Gaussian output sequence is generated by this Butterworth filter, and is regarded as a time-varying channel coefficient. These time-varying coefficients can be further processed by centering the $h_1(n)$ at 0.348, $h_2(n)$ at 0.87 and $h_3(n)$ at 0.348. The linear time-varying channel B then is made.

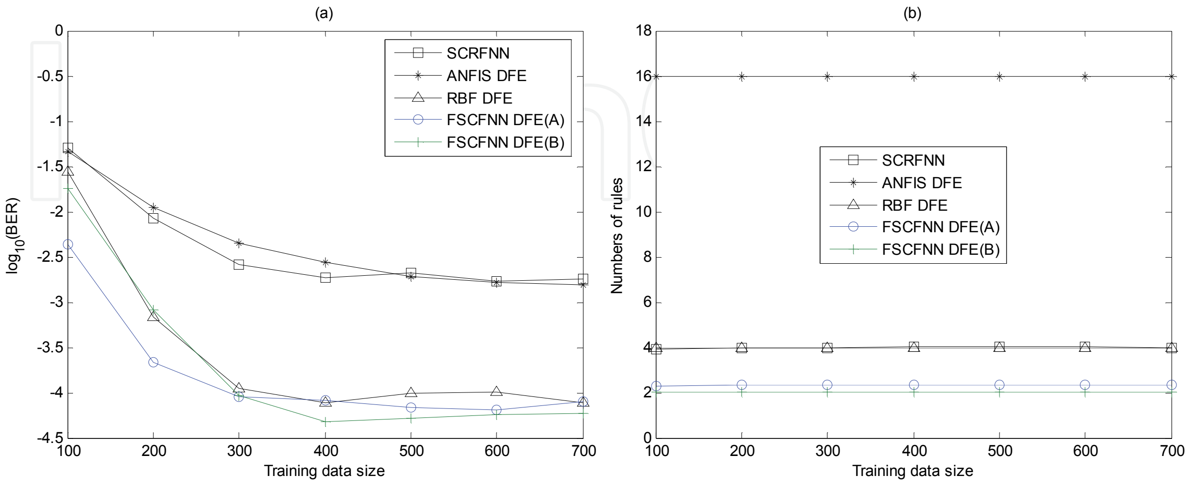

Fig. 2.5 Performance of various methods with a different length of training in the time-invariant channel A as SNR = 18 dB: (a) BER (b) Numbers of fuzzy rules

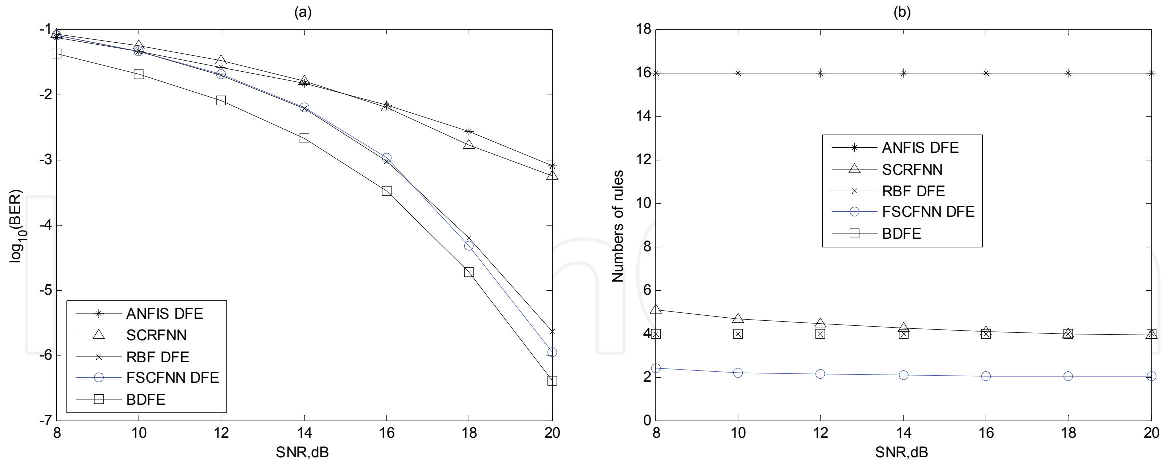

Fig. 2.6 Performance of various methods for different SNRs in the time-invariant channel A: (a) BER (b) Numbers of fuzzy rules

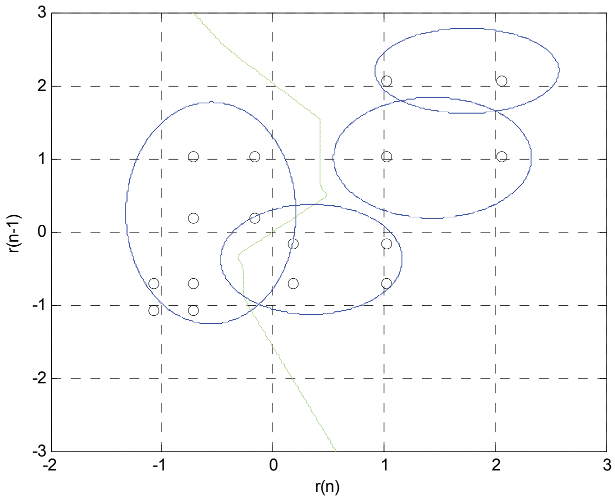

Fig. 2.7 Fuzzy rules generated by trained SCRFNN (ellipse), channel states (small circles) and optimal decision boundary (line) in the time-invariant channel A as SNR = 18dB

Figure 2.9 shows the performance of various methods for different f_d in time-varying channel B. Because the fade rate f_d in a real world is usually no larger than 0.1, thus we run the simulations from f_d = 0.02 (slowly time-varying) to f_d = 0.18 (fast time-varying). The FSCFNN DFEs with $\mu_{\min} = 0.95$ and $\mu_{\min} = 0.05$ are, respectively, denoted as FSCFNN DFE(A) and FSCFNN DFE(B) here. Besides, the values $\mu_{\min}$ in SCRFNN(A) and SCRFNN(B) are set as 0.00003 and 0.003, respectively. When the value of $\mu_{\min}$ is large enough, the BER performance of FSCFNN DFE for various time-varying environments may be satisfactory. Also, numbers of rules in FSCFNN DFE are increased as $\mu_{\min}$ grows. Because the FSCFNN DFE(B) has better performance in both time-varying channels B and C than the classical equalizers, the value $\mu_{\min} = 0.05$ is used in the following simulations of this paper.

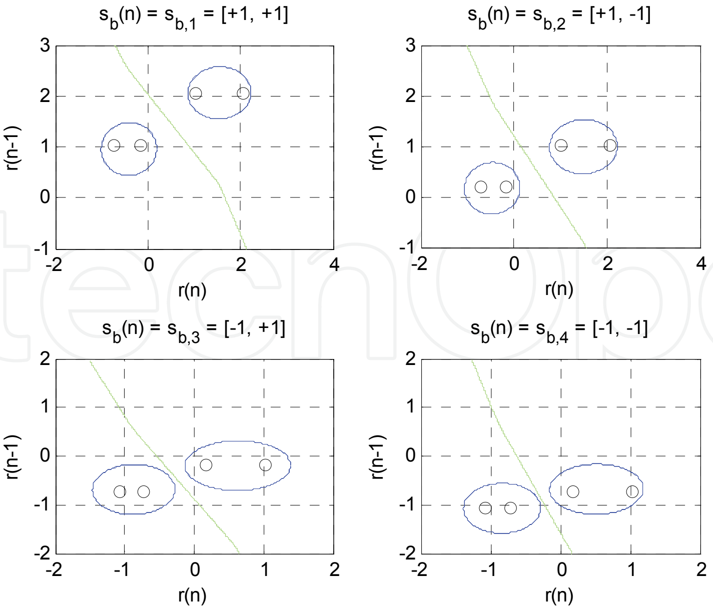

Fig. 2.8 Fuzzy rules generated by trained FSCFNN DFE (ellipse), channel states (small circles) and optimal decision boundaries (lines) for four feedback input vectors in the time-invariant channel A as SNR = 18dB

Figure 2.10 shows the performance of various methods for different SNRs in time-varying channel B. The SCRFNN equalizer with $\mu_{\min} = 0.003$ is used here for obtaining satisfactory performance. Note that the BER results as SNR = 18 dB in Figure 2.10(a) are gained by averaging 10000 runs for accurate consideration. The BER performance of FSCFNN DFE is slightly better than that of RBF DFE. However, the RBF DFE is assumed that the perfect knowledge of the channel order is acquired in advance for simulations. Similarly, numbers of rules needed in computation for FSCFNN DFE are the best.

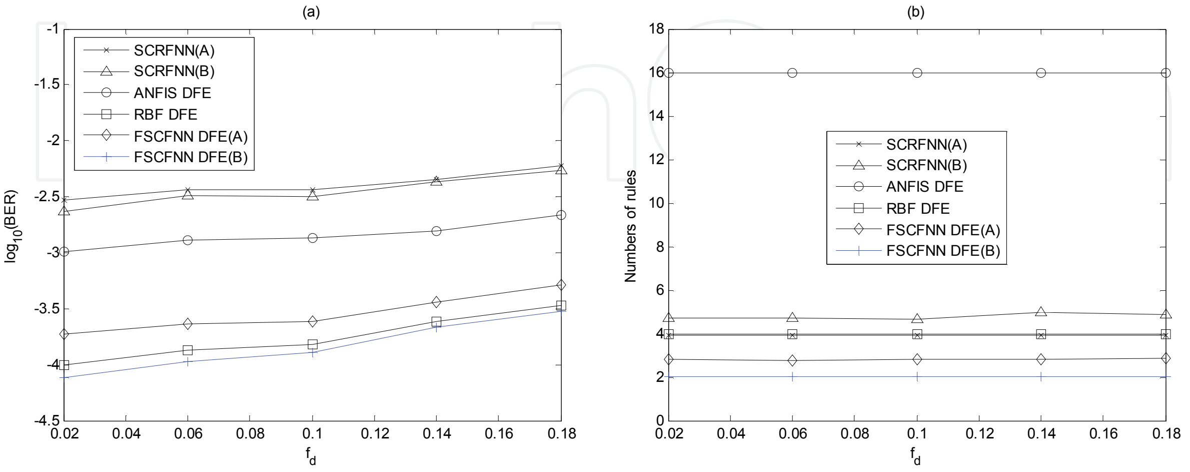

Fig. 2.9 Performance of various methods for different f_d in the time-varying channel B as SNR = 16 dB: (a) BER (b) Numbers of rules

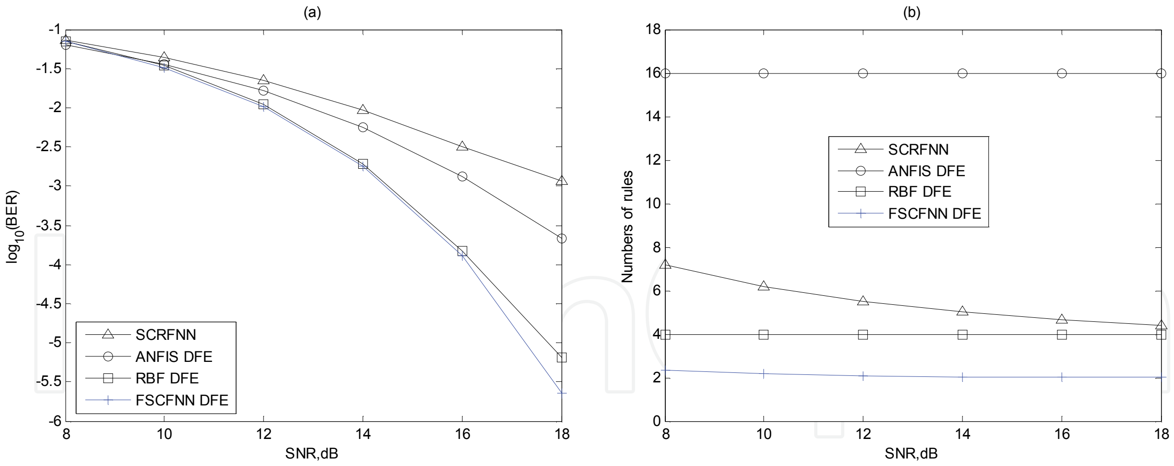

Fig. 2.10 Performance of various methods for different SNRs as f_d = 0.1 in the time-varying channel B (a) BER (b) Numbers of rules

3. Self-constructing fuzzy neural filtering for multi-antenna systems

Adaptive beamforming technology [3], [6], [8], [11], [18], [19], [46]-[48], [56] has been widely applied in smart antenna systems that can increase user's capacity and coverage in modern communication products. In this section, a powerful reduced symmetric self-constructing fuzzy neural network (RS-SCFNN) beamformer is presented for multi-antenna assisted systems. A novel training algorithm for the RS-SCFNN beamformer is proposed based on clustering of array input vectors and an adaptive minimum bit-error rate (MBER) method. An inherent symmetric property of the array input signal space is exploited to make training procedure of RS-SCFNN more efficient than that of standard SCFNN. In addition, the required amount of fuzzy rules can be greatly reduced in the RS-SCFNN structure. Simulation results demonstrate that RS-SCFNN beamformer provides superior performance to the classical linear ones and the other nonlinear ones, including symmetric radial basis function (SRBF), SCFNN and S-SCFNN, especially when supporting a large amount of users in the rank-deficient multi-antenna assisted system.

3.1 Multi-antenna array model

A uniformly spaced linear array is studied with L identical isotropic elements in this section, and the distance between the elements is represented by d. The plane waves impinge on the array in a θ angle in relation to the normal to the array and the difference of distance along one of the two adjacent ways is $d\sin\theta$. Note that the multi-antenna array model is completely the same as that in [6], [19], [46], [47], [57]. It is assumed that the system supports M users, and each user transmits a modulated signal on the same carrier frequency of $\omega = 2\pi f$. Then, the complex-valued signals received by the L-element antenna array are given by

$$x_l(n) = \sum_{i=1}^{M} A_i b_i(n) \exp(j\omega t_l(\theta_i)) + v_l(n) \tag{3.1}$$

where $1 \le l \le L$, $x_l(n) = x_l^R(n) + jx_l^I(n)$ is the complex-valued array input signal of the lth linear array element, n denotes the bit instance, the ith user's signal $b_i(n)$ is assumed to be a binary signal taking from the set {±1} with equal probability, A_i^2 denotes the signal power of user *i*, $t_l(\theta_i) = [(l-1)d\sin\theta_i]/c$ [57] is the relative time delay at element *l* for user *i*, θ_i is the direction of arrival (DOA) for user *i*, *c* is speed of light and $v_l(n)$ is the complex-valued white Gaussian noise having a zero mean and a variance of $2\sigma_v^2$. Without loss of generality, user 1 is assumed to be a desired user and the rest of the users are interfering users. The desired user's SNR is defined as $SNR = A_1^2/2\sigma_v^2$. We can rewrite (3.1) in a vector form:

$$\boldsymbol{x}(n) = \boldsymbol{Pb}(n) + \boldsymbol{v}(n)\,, \tag{3.2}$$

Where $\boldsymbol{x}(n) = [x_1(n),...,x_L(n)]^T$, $\boldsymbol{v}(n) = [v_1(n),...,v_L(n)]^T$, the system matrix $\boldsymbol{P}$ is given by $[A_1\boldsymbol{s}_1, A_2\boldsymbol{s}_2,...,A_M\boldsymbol{s}_M]$ with the *i*-th user's steering vector $\boldsymbol{s}_i = [\exp(j\omega t_1(\theta_i)),\exp(j\omega t_2(\theta_i)),...,\exp(j\omega t_L(\theta_i))]^T$ (i.e., the spatial transfer function between the *i*-th emitting source and the array), and the transmitted bit vector is $\boldsymbol{b}(n) = [b_1(n),...,b_M(n)]^T$. The purpose of an adaptive beamformer is to reconstruct the desired user's signal $b_1(n)$ based on the array input vector $\boldsymbol{x}(n)$.

3.2 Adaptive beamformer based on SCFNN-related detection

A. Adaptive SCFNN beamformer

Because the detection process in any digital communication systems can be viewed as a classification problem, which seeks to classify the observed vectors into one of the classes. Thus, the SCFNN-based classifiers shown in Section 2 can also be applied to multi-antenna assisted beamforming systems, and the SCFNN beamformer can classify the array input signal space $\chi = \{\boldsymbol{x}(n)\}$ into two classes, i.e., $\chi^{(+)}=\{\boldsymbol{x}(n)\,|\,b_1(n)=+1\}$ and $\chi^{(-)}=\{\boldsymbol{x}(n)\,|\,b_1(n)=-1\}$. At the *n*-th time instant, the adaptive beamformer's output based on a standard SCFNN is then expressed by

$$y_{\mathrm{s}}(\boldsymbol{x}(n)) = \sum_{k=1}^{K} w_k G_k(n)\,, \tag{3.3}$$

where *K* is the number of fuzzy rules, w_k is the real-valued consequent weight of the *k*-th fuzzy rule, and $G_k(n)$ is the Gaussian membership function (GMF) of the *k*-th fuzzy rule, which is associated with the current array input vector $\boldsymbol{x}(n)$:

$$G_k(n) = \exp\left\{-\sum_{l=1}^{L}\left(\frac{(x_l^R(n)-c_{kl}^R)^2}{2(\sigma_{kl}^R)^2} + \frac{(x_l^I(n)-c_{kl}^I)^2}{2(\sigma_{kl}^I)^2}\right)\right\} \tag{3.4}$$

$$\equiv G(c_k, \sigma_k; x(n))$$

where $c_{kl} = c_{kl}^R + jc_{kl}^I$ and $\sigma_{kl} = \sigma_{kl}^R + j\sigma_{kl}^I$, respectively, are the complex-valued center and complex-valued width of the *k*-th fuzzy rule for the *l*-th array input signal, and we define the center vector and width vector of the *k*-th fuzzy rule as $\boldsymbol{c}_k \equiv [c_{k1},...,c_{kL}]^T$ and $\boldsymbol{\sigma}_k \equiv [\sigma_{k1},..., \sigma_{kL}]^T$. The major difference between the equation (4.4) and a standard RBF [19] is that the ellipsoid GMFs are designed for the former, but the radial GMFs are used for the latter. To accommodate all geometric locations of $\boldsymbol{x}(n)$ belonging to χ by little geometric clusters corresponding to GMFs (i.e., classify all observed vectors $\boldsymbol{x}(n)$ with a small number K), the widths of the SCFNN classifier will be thus designed to be trainable to attain this purpose. The estimation of $b_1(n)$ is obtained by $\hat{b}_1(n) = \mathrm{sgn}\{y_\mathrm{s}(n)\}$.

As demonstrated in Section 2.2, the learning algorithm of a standard SCFNN detection involves two phases: self-constructing learning and parameter learning. Given a series of training data ($\boldsymbol{x}(n)$, $b_1(n)$), n = 0, 1, 2, …, the SCFNN training algorithm is performed at each time instant n. Note that there are no fuzzy rules in adaptive SCFNN beamformer initially, too. In the self-constructing learning phase, the maximum membership degree $\mu_{\max} = \mathrm{Max}_{1\le k\le K} G_k(n)$ is also adopted to judge whether to generate a fuzzy rule or not, and the parameters of the fuzzy rule generated are then initialized properly. Consequently, the growth criterion that must be met before a new fuzzy rule is added is:

$$\mu_{\max} \le \mu_{\min} \tag{3.5}$$

where $\mu_{\min}$ is a pre-specified threshold ($0 < \mu_{\min} < 1$). This growth criterion implies that the geometric clusters corresponding to the existing fuzzy rules are far from the geometric location of the current array input vector $\boldsymbol{x}(n)$. Hence, a new fuzzy rule should be generated to cover $\boldsymbol{x}(n)$, i.e., $K \leftarrow K + 1$. Once a new fuzzy rule is added, its initial geometric cluster is assigned accordingly:

$$\boldsymbol{c}_{new} = \boldsymbol{x}(n) \text{ and } \sigma_{new,l} = \sigma,\ 1 \le l \le L, \tag{3.6}$$

where σ is an empirical pre-specified value and set as 1+j1 in this section. By setting $\boldsymbol{c}_{new}$ as $\boldsymbol{x}(n)$, the current vector $\boldsymbol{x}(n)$ can be surely covered by this new fuzzy rule, and this design also satisfies the basic strategy of SCFNN, i.e., aiming to accommodate all geometric locations of the observed vectors. When the growth criterion defined in (3.5) doesn't be met at n, i.e., $\mu_{\max} > \mu_{\min}$, no fuzzy rule should be added and the parameter learning phase is performed to optimize the parameters of SCFNN beamformer, i.e., c_{kl}, σ_{kl} and w_k.

Traditionally, MMSE-based gradient descent methods are used for optimizing the parameters of a nonlinear detection [35], [38], [43], [44]. However, minimizing the MSE does not necessarily produce a low BER [5]-[9], [47] and hence an adaptive MBER-based gradient descent method recently has been proposed for a nonlinear structure [47]. In this chapter, we slightly modify the adaptive MBER method for the proposed SCFNN-related beamformers, which is summarized as follows. First, the decision variable $z(n) = b_1(n) \cdot y_S(\boldsymbol{x}(n))$ is defined [47] and the probability density function of $z(n)$ can be adaptively estimated by [47]

$$\tilde{p}(z,n) = \frac{1}{\sqrt{2\pi}\rho} \mathrm{e}^{-\frac{[z-b_1(n)y_S(x(n))]^2}{2\rho^2}}, \tag{3.7}$$

where ρ is the chosen kernel width [47]. Then the estimated error probability of an SCFNN-related beamformer at the time instant n can be given by [47]

$$P_E(n) = \int_{-\infty}^{0} \tilde{p}(z,n)\mathrm{d}z. \tag{3.8}$$

The objective of the standard MBER method is to minimize $P_E(n)$ subject to the SCFNN-related beamformer's parameters. Namely, all of parameters of SCFNN are adapted by MBER-based gradient descent method. Because the criterion $\mu_{\max} > \mu_{\min}$ implies that the array input vector $x(n)$ should be a member of the geometric cluster corresponding to $G_q(n)$, where $q = \arg\mathrm{Max}_{1\le k\le K} G_k(n)$, we propose to only optimize the parameters corresponding to the q-th fuzzy rule during MBER-based parameter training phase. By adopting this method, the time cost can be significantly reduced. Consequently, this modified MBER method (called C-MBER hereinafter for convenience) optimizes the parameters of the proposed SCFNN beamformer by the updating amount in (3.9)-(3.11):

$$\Delta w_q = -\gamma \frac{\partial P_E(n)}{\partial w_q} = \gamma \xi(n) G_q(n) \tag{3.9}$$

$$\Delta c_{ql}^Z = -\alpha \frac{\partial P_E(n)}{\partial c_{ql}^Z} = \alpha \xi(n) w_q G_q(n) \frac{x_l^Z(n) - c_{ql}^Z}{(\sigma_{ql}^Z)^2} \tag{3.10}$$

$$\Delta \sigma_{ql}^Z = -\beta \frac{\partial P_E(n)}{\partial \sigma_{ql}^Z} = \beta \xi(n) w_q G_q(n) \frac{(x_l^Z(n) - c_{ql}^Z)^2}{(\sigma_{ql}^Z)^3} \tag{3.11}$$

with

$$\xi(n) = \frac{\partial P_E(n)}{\partial y_{\mathrm{S}}(\boldsymbol{x}(n))} = \frac{-1}{\sqrt{2\pi}\rho} \mathrm{e}^{-\frac{[y_{\mathrm{S}}(\boldsymbol{x}(n))]^2}{2\rho^2}} \cdot b_1(n) \tag{3.12}$$

where $Z \in \{R, I\}$, α, β and γ are learning rates.

B. Adaptive S-SCFNN beamformer

In this sub-section, an inherent symmetry of the array input signal space is investigated and a novel S-SCFNN detection is designed based on this symmetry. First, let us denote $N_b = 2^{M-1}$ legitimate sequences of $\boldsymbol{b}(n)$ with $b_1(n) = +1$ as $\boldsymbol{b}_m^{(+)}$, $1 \le m \le N_b$, and denote 2^{M-1} legitimate sequences of $\boldsymbol{b}(n)$ with $b_1(n) = -1$ as $\boldsymbol{b}_m^{(-)} = -\boldsymbol{b}_m^{(+)}$, $1 \le m \le N_b$. Clearly, both of signal spaces $\chi^{(+)}$ and $\chi^{(-)}$ can be partitioned into N_b subspaces:

$$\chi^{(+)} = \cup_{1\le m\le N_b} \chi_m^{(+)} \text{ and } \chi^{(-)} = \cup_{1\le m\le N_b} \chi_m^{(-)}, \tag{3.13}$$

where $\chi_m^{(+)} = \{\boldsymbol{x}(n) | \boldsymbol{b}(n) = \boldsymbol{b}_m^{(+)}\}$ and $\chi_m^{(-)} = \{\boldsymbol{x}(n) | \boldsymbol{b}(n) = \boldsymbol{b}_m^{(-)}\}$. It can be easily seen that $\chi_m^{(+)} = \{\boldsymbol{x}(n) = \boldsymbol{P}\boldsymbol{b}_m^{(+)} + \boldsymbol{v}(n)\} \in \chi^{(+)}$ and $-\chi_m^{(+)} \equiv \left\{\boldsymbol{x}(n) = \boldsymbol{P}(-\boldsymbol{b}_m^{(+)}) - \boldsymbol{v}(n) = \boldsymbol{P}\boldsymbol{b}_m^{(-)} + \boldsymbol{v}(n)\right\} = \chi_m^{(-)} \in \chi^{(-)}$. Therefore, the two spaces $\chi^{(+)}$ and $\chi^{(-)}$ are distributed symmetrically, namely, for any subspace $\chi_m^{(-)} \in \chi^{(-)}$ the subspace $\chi_m^{(+)} \in \chi^{(+)}$ satisfies $\chi_m^{(-)} = -\chi_m^{(+)}$.

The basic idea of SCFNN learning is to accommodate all array input vectors $\boldsymbol{x}(n) \in \chi$ by adjusting the geometric clusters corresponding to $G_k(n)$, $k = 1, \dots, K$. Since the signal spaces $\chi^{(+)}$ and $\chi^{(-)}$ are distributed symmetrically in the multi-antenna beamforming system, we propose to create symmetric geometric clusters to accommodate all of $\boldsymbol{x}(n) \in \chi$. Thus, the output of the proposed S-SCFNN beamformer is defined by

$$y_{\mathrm{S}}(\boldsymbol{x}(n)) = \sum_{k=1}^{K} w_k (G_k^+(n) - G_k^-(n)) \tag{3.14}$$

with

$$G_k^+(n) = G(\boldsymbol{c}_k, \boldsymbol{\sigma}_k; \boldsymbol{x}(n)) \text{ and } G_k^-(n) = G(-\boldsymbol{c}_k, \boldsymbol{\sigma}_k; \boldsymbol{x}(n)), \tag{3.15}$$

where the geometric clusters corresponding to the two GMFs $G_k^+(n)$ and $G_k^-(n)$ are symmetric with each other. Moreover, all of geometric clusters corresponding to membership functions $G_k^+(n)$, $k = 1, \dots, K$, are assumed to be capable of accommodating the vectors $\boldsymbol{x}(n) \in \chi^{(+)}$, and naturally all of geometric clusters corresponding to $G_k^-(n)$, $k = 1, \dots, K$, can accommodate the vectors $\boldsymbol{x}(n) \in \chi^{(-)}$.

The learning procedure of SCFNN beamformer presented in the last section is modified here to make it suitable for the S-SCFNN beamformer. Similarly, there are no fuzzy rules initially in adaptive S-SCFNN beamformer. Because only the center vectors $\boldsymbol{c}_k \in \chi^{(+)}$, $k = 1 \sim K$, are needed in the S-SCFNN beamformer as defined in (3.14)-(3.15), we can focus on observing the array input vectors $\boldsymbol{x}(n) \in \chi^{(+)}$ during learning. Thus, the vector $\boldsymbol{x}(n) \in \chi^{(-)}$ with $b_1(n) = -1$ can be modified as $-\boldsymbol{x}(n) \in \chi^{(+)}$ before training:

$$\hat{\boldsymbol{x}}(n) = \begin{cases} \boldsymbol{x}(n), & if\ b_1(n) = 1 \\ -\boldsymbol{x}(n), & if\ b_1(n) = -1. \end{cases} \tag{3.16}$$

In the self-constructing learning phase, the maximum membership degree is also adopted. Because the k-th fuzzy rule of S-SCFNN detector is strongly related with the geometric clusters corresponding to both $G_k^+(n)$ and $G_k^-(n)$, the output value of $\{G_k^+(n) - G_k^-(n)\}$ is regarded as the membership degree that the current array input vector belongs to the k-th fuzzy rule. Thus the maximum membership degree is defined as

$$\mu_{\max} = \mathrm{Max}_{1 \le k \le K}\{G(c_k, \sigma_k; \hat{x}(n)) - G(-c_k, \sigma_k; \hat{x}(n))\}. \tag{3.17}$$

Consequently, the growth criterion that must be met before a new fuzzy rule is added is: $\mu_{\max} \le \mu_{\min}$, where $\mu_{\min}$ is as defined in (3.5). This criterion implies that the existing fuzzy rules cannot simultaneously satisfy the following two conditions:

(a) At least one of geometric clusters of $G_k^+(n)$ is close to the geometric location of $\hat{\mathrm{x}}(\mathrm{n})$.

(b) The geometric cluster of $G_k^-(n)$ should be relatively far from the geometric location of $\hat{\boldsymbol{x}}(n)$ compared to that of $G_k^+(n)$.

Hence, a new fuzzy rule should be generated to accommodate $\hat{\boldsymbol{x}}(n)$, i.e., $K \leftarrow K + 1$. Then, its initial symmetric clusters are assigned by

$$\boldsymbol{c}_{new} = \hat{\boldsymbol{x}}(n) \text{ and } \sigma_{new,l} = \sigma,\ 1 \le l \le L, \tag{3.18}$$

where σ is specified as in the last section.

When the growth criterion doesn't be met at n, no fuzzy rule should be added and the C-MBER method is performed to optimize the parameters of the S-SCFNN beamformer. The estimated error probability of S-SCFNN beamformer is the same as (3.8). Consequently, the updating amount of the S-SCFNN beamformer's parameters is derived as done in (3.9)- (3.12):

$$\Delta w_q = \gamma \xi(n)\left(G_q^+(n) + G_q^-(n)\right) \tag{3.19}$$

$$\Delta c_{ql}^Z = \alpha \xi(n) w_q \left(G_q^+(n) \frac{x_l^Z(n) - c_{ql}^Z}{(\sigma_{ql}^Z)^2} + G_q^-(n) \frac{x_l^Z(n) + c_{ql}^Z}{(\sigma_{ql}^Z)^2}\right) \tag{3.20}$$

$$\Delta \sigma_{ql}^Z = \beta \xi(n) w_q \left(G_q^+(n) \frac{(x_l^Z(n) - c_{ql}^Z)^2}{(\sigma_{ql}^Z)^3} - G_q^-(n) \frac{(x_l^Z(n) + c_{ql}^Z)^2}{(\sigma_{ql}^Z)^3}\right), \tag{3.21}$$

where

$$q = \arg\mathrm{Max}_{1\le k\le K}\{G(\boldsymbol{c}_k, \boldsymbol{\sigma}_k; \hat{\boldsymbol{x}}(n)) - G(-\boldsymbol{c}_k, \boldsymbol{\sigma}_k; \hat{\boldsymbol{x}}(n))\}. \tag{3.22}$$

C. Adaptive RS-SCFNN beamformer

Because SCFNN or S-SCFNN learning procedure would generate too many fuzzy rules at low SNRs in applications, we further propose a reduced S-SCFNN (RS-SCFNN) beamformer here. The RS-SCFNN beamformer's output is completely the same as the S-SCFNN one, i.e., (3.14)-(3.15).

Like the S-SCFNN learning procedure, there are no fuzzy rules initially in the RS-SCFNN beamformer, and we focus on observing the vectors $\boldsymbol{x}(n) \in \chi^{(+)}$ during learning. The learning procedure also contains the self-constructing and parameter learning phases. It can be easily seen from (3.13) that the reasonable number of the Gaussian clusters inside the subspace $\chi^{(+)}$ is $N_b = 2^{M-1}$, so, actually, it is enough to accommodate all the geometric locations of $\boldsymbol{x}(n) \in \chi^{(+)}$ by creating N_b geometric Gaussian clusters, i.e., the maximum number of K should be 2^{M-1}. In the application of multi-antenna assisted systems, the user's number M supported usually is regarded as a known number [20]-[21]. Therefore, a simple inhibition criterion $K \le N_b$ can be added in the self-constructing learning phase. Then, the criteria that must be met before a new fuzzy rule is added are:

$$\mu_{\max} \le \mu_{\min} \text{ and } K \le N \tag{3.23}$$

where $\mu_{\max}$ and $\mu_{\min}$ are as defined in the SCFNN or S-SCFNN learning, and the integer N is pre-assigned to be no larger than N_b. With the aid of the latter criterion in (3.23), the number of fuzzy rules generated during learning can be greatly reduced at low SNRs compared with that of the S-SCFNN beamformer. The initial cluster setting of a new fuzzy rule and the parameter learning phase are the same as described in (3.18)-(3.22).

The center vectors $\boldsymbol{c}_k$ of the adaptive SRBF beamformer in the literature have been trained by several algorithms, such as classical k-means clustering, improved enchanced k-means clustering [48] and MBER method [47]. However, a serious slow convergence easily occurs in these classical clustering due to the improper initialization. Moreover, if the number of the center vectors is really huge, the classical algorithms will even need a longer training sequence to achieve convergence. Compared to the adaptive SRBF beamformer, the proposed RS-SCFNN beamformer has a faster convergence capability. By using the criterion $\mu_{\max} < \mu_{\min}$ during the self-constructing learning phase, the initial location of the new center vector $\boldsymbol{c}_{new} = \hat{\boldsymbol{x}}(n)$ can be far from the locations of the other existing center vectors $\boldsymbol{c}_k$. This feature avoids two or more center vectors initially located around the same ideal center.

Although in this section we design the adaptive RS-SCFNN beamformer under the BPSK beamforming system, its extension to high-order quadrature amplitude modulation (QAM) schemes is also achievable. For example, for the 4-QAM modulation, the array input signal space χ can be partitioned into the four subsets $\chi^{(\pm 1 \pm j)}$ depending on the value of $b_1(n)$ as the above derivation. Besides, for the four subspaces $\chi^{(\pm 1 \pm j)}$, the following symmetric relationship can be easily verified by using the same derivation in S-SCFNN learning: $\chi^{(-1+j)} = +j \cdot \chi^{(+1+j)}$, $\chi^{(-1-j)} = -1 \cdot \chi^{(+1+j)}$ and $\chi^{(+1-j)} = -j \cdot \chi^{(+1+j)}$. Then, the idea of creating symmetric geometric clusters to accommodate all $\boldsymbol{x}(n) \in \chi$ can be exploited to modify the 4-QAM FNN detector in [41] [44]. The derivation for the high-order QAM RS-SCFNN beamformer is much more complex and is beyond the scope of this work.

3.3 Simulation results

In this sub-section, a rank-deficient multi-antenna assisted beamforming systems is used to demonstrate the efficiency of the proposed adaptive RS-SCFNN beamformer. The beamforming systems considered is summarized in Table 3.1. As done in [1], [6], [11], [18], [19], [47], [48], DOAs for each user are fixed, and the distance between linear array elements is $d = \lambda/2$. The simulated narrowband channels are $A_i = 1 + j0$ as done in [19], [47]. The threshold $\mu_{\min}$ for various SCFNN related beamformers is set as 0.06 in the three systems. For comparisons, the linear MMSE [1], linear adaptive MBER [8], adaptive SRBF [47] and optimal Bayesian solution [19] are added in the simulations. The MBER learning rates for different nonlinear adaptive beamformers are listed in Table 3.2, and are used in the following simulations. These MBER learning rates were found on conditions of 1000 training data and 8-dB SNR, and are the best choices satisfying the three systems for each nonlinear adaptive beamformer. Note that SCFNN, S-SCFNN and RS-SCFNN are trained by the C-MBER method during parameter learning phase, and SRBF is trained by the MBER method [10]. The RS-SCFNN-M in Table 3.2 denotes RS-SCFNN trained with the standard MBER method. The MBER learning rate for linear beamformer [6] is set as 0.1 in the simulations. The simulation results are obtained by averaging 10^4 individual runs, each of which involves a different random sequence for training and testing. The testing data size is 10^3, and the training data size will be given later.

Simulated system $M = 6$, $L = 4$	User i	1	2	3	4	5	6
	DOA θ_i	0°	10°	-17°	15°	20°	-15°

Table 3.1 Multi-antenna assisted beamforming system simulated in Section 3.3

Beamformers	α	β	γ	ρ
SCFNN	0.25	0.01	0.005	0.7
S-SCFNN	2.5	0.55	0.5	0.25
RS-SCFNN	2.5	0.55	0.5	0.25
RS-SCFNN-M	0.3	0.25	0.05	0.2
SRBF	0.3	0.3	0.3	0.2

Table 3.2 MBER learning rates for different nonlinear adaptive beamformers used in Section 3.3

Due to the fact of $N \le N_b = 2^{6-1} = 32$, we choose $N = 23$ and $N = 28$ for RS-SCFNN beamformers in the simulations. The chosen training data size for all adaptive beamformers is 400 in the simulated System A. Figure 3.1 depicts the BER performance for adaptive SCFNN related beamformers. Figure 3.2 shows the average numbers of fuzzy rules for adaptive SCFNN related beamformers. Since adaptive S-SCFNN beamformer only observes half the array input signal space χ during training, S-SCFNN can generate half as many fuzzy rules as SCFNN. Namely, the S-SCFNN beamformer only needs to train half as many parameters ($\boldsymbol{c}_k$, $\boldsymbol{\sigma}_k$ and $\boldsymbol{w}_k$) as the SCFNN one. As a result, the parameters of S-SCFNN beamformer can quickly converge, and thus the SCFNN exploiting symmetry has better BER performance than the standard SCFNN. When SNR is larger than 2 dB, the average numbers of fuzzy rules for S-SCFNN and RS-SCFNN beamformers are almost the same. Thus they also have similar BER performance at SNR = 4 ~ 8 dB. However, the numbers of fuzzy rules for S-SCFNN beamformer are sharply increased at low SNRs. With tiny sacrifice for BER performance, the numbers of fuzzy rules for RS-SCFNN can be effectively limited within the number *N* as mentioned in Section 3.2. In order to check the performance of C-MBER and MBER methods, the RS-SCFNN-M beamformer is also plotted. The results indicate that the RS-SCFNN with C-MBER has the similar performance to the RS-SCFNN with MBER (RS-SCFNN-M). Note that the C-MBER method only needs to train parameters associated with one of fuzzy rules as mentioned in Section 3.2, but the standard MBER method [47] have to train parameters associated with all of fuzzy rules.

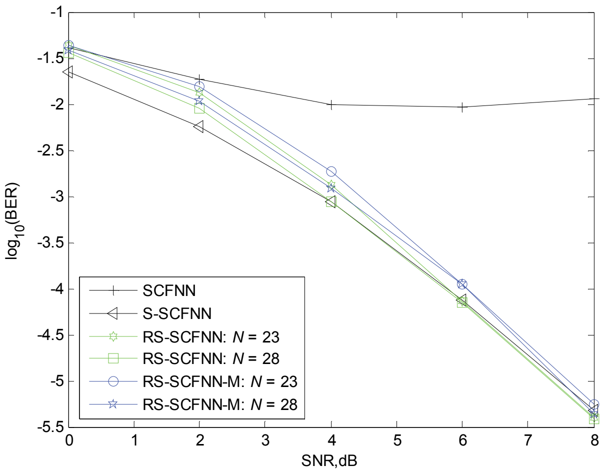

Fig. 3.1 BER performance for adaptive SCFNN related beamformers

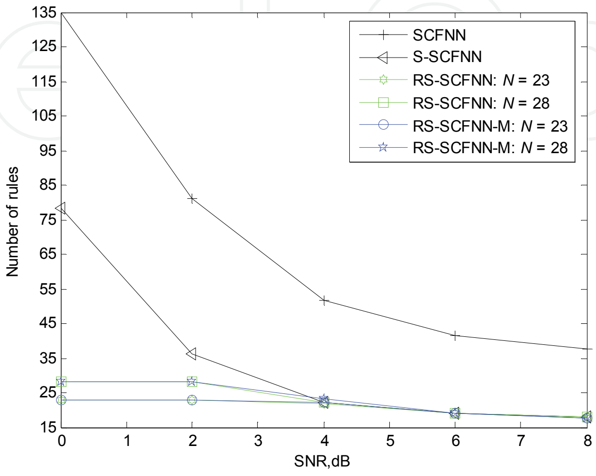

Fig. 3.2 Numbers of hidden nodes for adaptive SCFNN related beamformers

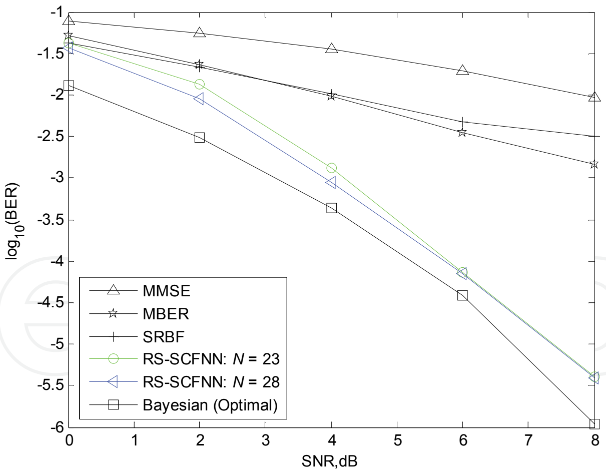

Fig. 3.3 BER performance for various adaptive beamformers

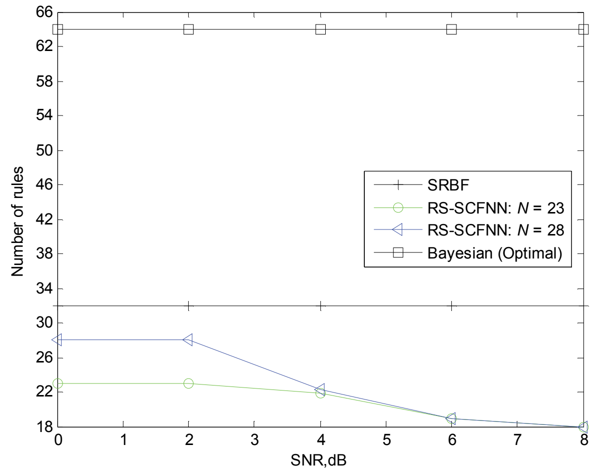

Fig. 3.4 Numbers of hidden nodes for adaptive SRBF and RS-SCFNN beamformers

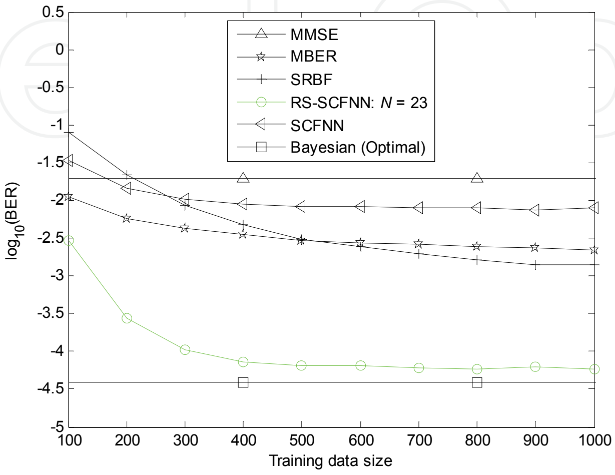

Fig. 3.5 Convergence rates for different adaptive beamformers at SNR = 6 dB

The number of rules for SRBF beamformer is fixed before training and specified by $2^{M-1} = 32$ as done in [47]. Then the BER performance for various kinds of adaptive beamformers is illustrated in Figure 3.3, and the average numbers of rules for various nonlinear adaptive beamformers are plotted in Fig. 3.4. In such rank-deficient system and low amount of training data, the proposed RS-SCFNN beamformers can provide excellent BER performance compared to the classical linear and nonlinear ones. As shown in Figure 3.4, the numbers of rules for adaptive RS-SCFNN beamformer can be determined flexibly at different SNRs, but those for adaptive SRBF one must be fixed to a constant at every SNR. Of course, the SRBF beamformer also can assign different numbers of hidden nodes for various SNRs, but it will need huge manpower to achieve this purpose. The relatively large numbers of rules and bad initial parameters for SRBF easily lead to a slow convergence or poor BER results. However, the proposed RS-SCFNN beamformer can set good initial parameters as specified in Section 3.2 with little fuzzy rules, so the RS-SCFNN has the BER results close to Bayesian solutions. The BER convergence rates for different beamformers are demonstrated in Figure 3.5. We can see that the proposed RS-SCFNN beamformer can obtain satisfactory performance close to Bayesian solutions if the training data size up to 300.

4. Conclusion

This chapter provides different versions of SCFNN-based detectors for various system models to improve the performance of classical nonlinear detectors. For improving the classical RBF, FNN and SCRFNN equalizers in both time-invariant and time-varying channels, a novel FSCFNN DFE has been demonstrated in Section 2. Specifically, FSCFNN DFE is composed of several FSCFNN detectors, each of which corresponds to one feedback input vectors. The fast learning processes, i.e., self-constructing and parameter learning, are adopted in FSCFNN DFE to make it suitable in time-varying environments. The fast learning algorithm of FSCFNN DFE has set conditions on the increase demand of fuzzy rules during the self-constructing algorithm and FSCFNN DFE only activates only one FSCFNN detector at each time instant. Therefore, the computational complexity of FSCFNN DFE is less than that of traditional equalizers. In multi-antenna systems, adaptive beamformers based on SCFNN detectors are also presented in Section 3. By adopting the symmetric property of array input signal space, the RS-SCFNN learning algorithm can be easier than the standard SCFNN one. From the simulations, we can see that the SCFNN-based adaptive beamformers can flexibly and automatically determine structure size themselves for various SNRs. Therefore, we conclude that the adaptive RS-SCFNN beamformer is potentially a better scheme than the SRBF and SCFNN ones. Because the competitors of SCFNN-based detectors, such as RBF and FNN, have been successfully applied to the space-time equalization, turbo equalization, DOA estimation, high-level QAM systems, OOK optical communications and CDMA or OFDM communication systems in the literature (IEEE/IET Journals etc.), the future work for SCFNN detectors could be the extension to the above mentioned systems or the improvement of SCFNN-based detectors based on the issues of determination of threshold $\mu_{\min}$ and the method for simplifying the complexity.

5. References

[1] S. Haykin, Adaptive filter theory (4th Edition), Prentice Hall, 2002.

[2] T.S. Rappaport, Wireless communications: principles and practice (2nd Edition), Prentice Hall, 2002.

[3] J. Litva, T.K.Y. Lo, Digital beamforming in wireless communications, Artech House, 1996.

[4] Y. Gong, X. Hong, "OFDM joint data detection and phase noise cancellation based on minimum mean square prediction error," Signal Process., vol. 89, pp. 502-509, 2009.

[5] M.Y. Alias, A.K. Samingan, S. Chen, L. Hanzo, "Multiple antenna aided OFDM employing minimum bit error rate multiuser detection," Electron. Lett., vol. 39, no. 24, pp. 1769-1770, 2003.

[6] S. Chen, N.N. Ahmad, L. Hanzo, "Adaptive minimum bit error rate beamforming," IEEE Trans. Wirel. Commun., vol. 4, no.2, pp. 341-348, 2005.

[7] J. Li, G. Wei, F. Chen, "On minimum-BER linear multiuser detection for DS-CDMA channels," IEEE Trans. Signal Process., vol. 55, no.3, pp. 1093-1103, 2007.

[8] T.A. Samir, S. Elnoubi, A. Elnashar, "Block-Shannon minimum bit error rate beamforming," IEEE Trans. Veh. Technol., vol. 57, no.5, pp. 2981-2990, 2008.

[9] W. Yao, S. Chen, S. Tan, L. Hanzo, "Minimum bit error rate multiuser Transmission designs using particle swarm optimisation," IEEE Trans. Wirel. Commun., vol. 8, no.10, pp. 5012-5017, 2009.

[10] S. Gollamudi, S. Nagaraj, S. Kapoor, Y.F. Huang, "Set-membership filtering and a set-membership normalized LMS algorithm with an adaptive step size," IEEE Signal Process. Lett., vol. 5, no. 5, pp. 111-114, 1998.

[11] Y.C. Liang, F.P.C. Chin, "Coherent LMS algorithms," IEEE Commun. Lett., vol. 4, no. 3, pp. 92-94, 2000.

[12] S. Choi, T.W. Lee, D. Hong, "Adaptive error-constrained method for LMS algorithms and applications," Signal Process., vol. 85, pp. 1875-1897, 2005.

[13] E.F. Harrington, "A BPSK decision-feedback equalization method robust to phase and timing errors," IEEE Signal Process. Lett., vol. 12, pp. 313-316, 2005.

[14] S. Chen, B. Mulgrew, P.M. Grant, "A clustering technique for digital communications channel equalization using radial basis function networks," IEEE Trans. Neural Netw., vol. 4, no. 4, pp. 570-579, 1993.

[15] J. Montalvão, B. Dorizzi, J. Cesar M. Mota, "Why use Bayesian equalization based on finite data blocks," Signal Process., vol. 81, pp. 137-147, 2001.

[16] S. Chen, B. Mulgrew, S. McLaughlin, "Adaptive Bayesian equalizer with decision feedback," IEEE Trans. Signal Process., vol. 41, no. 9, pp. 2918-2927, 1993.

[17] S. Chen, S. McLaughlin, B. Mulgrew, P.M. Grant, "Bayesian decision feedback equalizer for overcoming co-channel interference," IEE Proc.-Commun., vol. 143, no. 4, pp. 219-225, 1996.

[18] S. Chen, L. Hanzo, A. Wolfgang, "Kernel-based nonlinear beamforming construction using orthogonal forward selection with the fisher ratio class separability measure," IEEE Signal Process. Lett., vol. 11, no. 5, pp. 478-481, 2004.

[19] S. Chen, L. Hanzo, A. Wolfgang, "Nonlinear multiantenna detection methods," EURASIP J Appl. Signal Process., vol. 9, pp. 1225-1237, 2004.

[20] J. Lee, C. Beach, N. Tepedelenlioglu, "A practical radial basis function equalizer," IEEE Trans. Neural Netw., vol. 10, pp. 450-455, 1999.

[21] P.C. Kumar, P. Saratchandran, N. Sundararajan, "Minimal radial basis function neural networks for nonlinear channel equalization," IEE Proc.-Vis. Image Signal Process., vol. 147, no. 5, pp. 428-435, 2000.

[22] M.S. Yee, B.L. Yeap, L. Hanzo, "Radial basis function-assisted turbo equalization," IEEE Trans. Commun., vol. 51, pp. 664-675, 2003.

[23] Wolfgang, S. Chen, L. Hanzo, "Radial basis function network assisted space-time equalisation for dispersive fading environments," Electron. Lett., vol. 40, no. 16, pp. 1006-1007, 2004.

[24] J.B. MacQueen, "Some methods for classification and analysis of multivariate observations," Proceedings of the 5th Berkeley Symposium on Mathematics Statistics and Probability, Berkeley, U.S.A., pp. 281-297, 1967.

[25] R. Assaf, S.E. Assad, Y. Harkouss, "Adaptive equalization for digital channels RBF neural network," Proceedings of the European Conference on Wireless Technology, Paris, France, pp. 347-350, 2005.

[26] R. Assaf, S.E. Assad, Y. Harkouss, "Adaptive equalization of nonlinear time varying-channel using radial basis network," Proceedings of the 2006 International Conference on Information and Communication Technologies, Damascus, Syria, pp. 1866-1871, 2006.

[27] L. Xu, A. Krzyzak, E. Oja, "Rival penalized competitive learning for clustering analysis, RBF net, and curve detection," IEEE Trans. Neural Netw., vol. 4, no. 4, pp. 636-649, 1993.

[28] Y.M. Cheung, "On rival penalization controlled competitive learning for clustering with automatic cluster *number* selection," IEEE Trans. Knowl. Data Eng., vol. 17, no. 11, pp. 1583-1588, 2005.

[29] J. Ma, T. Wang, "A cost-function approach to rival penalized competitive learning (RPCL)," IEEE Trans. Syst. Man Cybern. Part B-Cybern., vol. 36, no. 4, pp. 722-737, 2006.

[30] S. Chen, T. Mei, M. Luo, H. Liang, "Study on a new RPCCL clustering algorithm," Proceedings of the 2007 IEEE International Conference on Mechatronics and Automation, Harbin, China, pp. 299-303, 2007.

[31] X. Qiao, G. Ji, H. Zheng, "An improved rival penalized competitive learning algorithm based on fractal dimension of algae image," Proceedings of the 2008 IEEE Control and Decision Conference, Yantai, China, pp. 199-202, 2008.

[32] S. Siu, G.J. Gibson, C.F.N. Cowan, "Decision feedback equalization using neural network structures and performance comparison with standard architecture," IEE Proc.-Commun., vol. 137, pp. 221-225, 1990.

[33] J. Coloma, R.A. Carrasco, "MLP equaliser for frequency selective time-varying channels," Electron. Lett., vol. 30, pp. 503-504, 1994.

[34] C.H. Chang, S. Siu, C.H. Wei, "Decision feedback equalization using complex backpropagation algorithm," Proceedings of 1997 IEEE International Symposium on Circuits and Systems, Hong Kong, China, pp. 589-592, 1997.

[35] S.S. Yang, C.L. Ho, C.M. Lee, "HBP: improvement in BP algorithm for an adaptive MLP decision feedback equalizer," IEEE Trans. Circuits Syst. II-Express Briefs, vol. 53, no. 3, pp. 240-244, 2006.

[36] S. Siu, S.S. Yang, C.M. Lee, C.L. Ho, "Improving the Back-propagation algorithm using evolutionary strategy," IEEE Trans. Circuits Syst. II-Express Briefs, vol. 54, no. 2, pp. 171-175, 2007.

[37] K. Mahmood, A. Zidouri, A. Zerquine, "Performance analysis of a RLS-based MLP-DFE in time-invariant and time-varying channels," Digit. Signal Prog., vol. 18, no. 3, pp. 307-320, 2008.

[38] S.S. Yang, S. Siu, C.L. Ho, "Analysis of the initial values in split-complex backpropagation algorithm," IEEE Trans. Neural Netw., vol. 19, pp. 1564-1573, 2008.

[39] J.S.R. Jang, C.T. Sun, E. Mizutani, Neuro-fuzzy and soft computing - a computational approach to learning and machine intelligence, Prentice Hall, 1997.

[40] C.H. Lee, Y.C. Lin, "An adaptive neuro-fuzzy filter design via periodic fuzzy neural network," Signal Process., vol. 85, pp. 401-411, 2005.

[41] S. Siu, C.L. Ho, C.M. Lee, "TSK-based decision feedback equalizer using an evolutionary algorithm applied to QAM communication systems," IEEE Trans. Circuits Syst. II-Express Briefs,vol. 52, pp. 596-600, 2005.

[42] F.J. Lin, C.H. Lin, P.H. Shen, "Self-constructing fuzzy neural network speed controller for permanent-magnet synchronous motor drive," IEEE Trans. Fuzzy Syst., vol. 9, no. 5, pp. 751-759, 2001.

[43] W.D. Weng, R.C. Lin, C.T. Hsueh, "The design of an SCFNN based nonlinear channel equalizer," J. Inf. Sci. Eng., vol. 21, pp. 695-709, 2005.

[44] R.C. Lin, W.D. Weng, C.T. Hsueh, "Design of an SCRFNN-based nonlinear channel equaliser," IEE Proc.-Commun., vol. 152, no. 6, pp. 771-779, 2005.

[45] A. Bateman, A. Bateman, Digital communications: design for the real world, Addison Wesley, 1999.

[46] M. Martínez-Ramón, J.L. Rojo-Álvarez, G. Camps-Valls, C.G. Christodoulou, "Kernel antenna array processing," IEEE Trans. Antennas Propag., vol. 55, no. 3, pp. 642-650, 2007.

[47] S. Chen, A. Wolfgang, C.J. Harris, L. Hanzo, "Adaptive nonlinear least bit error-rate detection for symmetrical RBF beamforming," Neural Netw., vol. 21, pp. 358-367, 2008.

[48] S. Chen, A. Wolfgang, C.J. Harris, L. Hanzo, "Symmetric complex-valued RBF receiver for multiple-antenna-aided wireless systems," IEEE Trans. Neural Netw., vol. 19, no. 9, pp. 1657-1663, 2008.

[49] Q. Liang, J.M. Mendel, "Overcoming time-varying co-channel interference using type-2 fuzzy adaptive filter," IEEE Trans. Circuits Syst. II-Express Briefs, vol. 47, pp. 1419-1428, 2000.

[50] Q. Liang, J.M. Mendel, "Equalization of nonlinear time-varying channels using type-2 fuzzy adaptive filters," IEEE Trans. Fuzzy Syst., vol. 8, no. 5, pp. 551-563, 2000.

[51] Y.J. Chang, C.L. Ho, "Improving the BP algorithm using RPCL for FNN-based adaptive equalizers," Proceedings of 2008 National Symposium on Telecommunications, Yunlin, Taiwan, pp. 1385-1388, 2008.

[52] Y.J. Chang, C.L. Ho, "SOFNN-based equalization using rival penalized controlled competitive learning for time-varying environments," Proceedings of 2009

International Conference on Wireless Communication and Signal Processing, Nanjian, China, 2009.

[53] S. Han, I. Lee, W. Pedrycz, "Modified fuzzy c-means and Bayesian equalizer for nonlinear blind channel," Appl. Soft. Comput., vol. 9, pp. 1090-1096, 2009.

[54] J. Choi, A.C. de C. Lima, S. Haykin, "Kalman filter-trained recurrent neural equalizers for time-varying channels," IEEE Trans. Commun., vol. 53, no. 3, pp. 472-480, 2005.

[55] C.F.N. Cowan, S. Semnani, "Time-variant equalization using a novel nonlinear adaptive structure," Int. J. Adapt. Control Signal Process., vol. 12, no. 2, pp. 195-206, 1998.

[56] R. Kohno, "Spatial and temporal communication theory using adaptive antenna array," IEEE Pers. Commun., vol. 5, no. 1, pp. 28-35, 1998.

[57] Ju´nior, A.D.D. Neto, W. Mata, "Determination of multiple direction of arrival in antennas arrays with radial basis functions," Neurocomputing, vol. 70, pp. 55-61, 2006.

[58] Y.J. Chang, S.S. Yang, C.L. Ho, " Fast Self-constructing Fuzzy Neural Network-based Decision Feedback Equaliser in Time-invariant and Time-varying Channels," IET Commun., vol. 4, no. 4, pp. 463-471, 2010.

Adaptive Filtering

Edited by Dr Lino Garcia

ISBN 978-953-307-158-9
Hard cover, 398 pages
Publisher InTech
Published online 06, September, 2011
Published in print edition September, 2011

Adaptive filtering is useful in any application where the signals or the modeled system vary over time. The configuration of the system and, in particular, the position where the adaptive processor is placed generate different areas or application fields such as prediction, system identification and modeling, equalization, cancellation of interference, etc., which are very important in many disciplines such as control systems, communications, signal processing, acoustics, voice, sound and image, etc. The book consists of noise and echo cancellation, medical applications, communications systems and others hardly joined by their heterogeneity. Each application is a case study with rigor that shows weakness/strength of the method used, assesses its suitability and suggests new forms and areas of use. The problems are becoming increasingly complex and applications must be adapted to solve them. The adaptive filters have proven to be useful in these environments of multiple input/output, variant-time behaviors, and long and complex transfer functions effectively, but fundamentally they still have to evolve. This book is a demonstration of this and a small illustration of everything that is to come.

InTech Europe
University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China
Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821