# Complex-valued SRFNN with Decision Feedback for QAM signalling systems

Yao-Jen Chang*, Chia-Lu Ho, Guo-Tong Fang

*Department of Communication Engineering, National Central University*
Chung Li, Taoyuan 32054, Taiwan (R.O.C.)
*955403004@cc.ncu.edu.tw

***Abstract*—This paper proposes a novel adaptive decision feedback equalizer (DFE) based on self-constructing recurrent fuzzy neural network (SRFNN) for quadrature amplitude modulation systems. Without the prior knowledge of channel characteristics, a novel training scheme containing both self-constructing learning and back-propagation algorithms is derived for the SRFNN. The proposed DFE is compared with several neural network (NN)-based DFEs on a nonlinear complex-valued channel. The results show that the SRFNN DFE is superior to classical NN DFEs in terms of symbol-error rate and convergence speed.**

***Keywords*—Adaptive equalizers, ISI, MLP, TSK**

## 1. INTRODUCTION

To further enhance the performance of decision feedback equalizer (DFE), many kinds of neural networks (NNs), like multilayer perceptrons (MLPs) [1-4] and Gaussian basis functions (GBFs) [5-7], have been incorporated into the DFE. These NN DFEs give a greatly improvement on the original DFE for pulse amplitude modulation or quadrature amplitude modulation (QAM) signals [1-7]. Back-propagation (BP) algorithm [1-3,5] has been used to train the parameters of NNs. For GBFs, there is another training method, which is called clustering learning (CL) algorithm [6-7].

The advantage of BP algorithms over the CL algorithm is that an additional estimation for channel order is not necessary. However, since BPs are sensitive to the initial parameters of NNs, they are easily dropped into local minima [8] which may lead to poor symbol-error rate (SER) performance. By searching for the centers of signal clusters, CL algorithms [6-7] can make parameters of GBFs near to optimal Bayesian parameters [6]. The SER performance of GBF DFE based on the CL algorithm thus is very close to the optimal Bayesian DFE solution. But, the additional estimation procedure for the channel order [9] before the CL algorithm is a difficult part, especially when the channel is nonlinear. Moreover, as the channel order or the equalizer order increases, the number of nodes in GBFs trained with the CL algorithm [6-7] grows exponentially, and as a result, so does the computation and hardware complexity [9].

A new GBF termed self-constructing recurrent fuzzy NN (SRFNN) [10] has been recently applied to channel equalization. Specifically, the SRFNN performs both self-constructing learning (SL) algorithm and BP algorithm simultaneously in the training process without the knowledge of channel characteristics. Initially, there are no nodes (also called fuzzy rules hereinafter) in the SRFNN structure. All of the nodes are generated online by the SL algorithm, that not only helps automate structure modification but also locates good initial parameters for the subsequent BP algorithm [10]. The SER of the SRFNN TE thus is extremely superior to that of traditional NN TEs trained by simple BP algorithm. Moreover, the SL can limit the amount of nodes by employing a evaluation criterion and hence SRFNN results in lower computational costs compared to traditional GBFs.

The problem of traditional NN DFEs is the lack of an intelligent scheme for structure modification. Although the SRFNN in [10] has provided a scheme to automatic node generation, it doesn't take advantage of decision feedback symbols to counteract the distortion effects on communication systems. Furthermore, the drawbacks of classical NN DFEs trained by classical learning algorithms would be deteriorated sharply in QAM systems. In this paper we thus design a novel DFE incorporated with the SRFNN for QAM signalling systems. Without the prior knowledge of channel characteristics, the simulation results show that the performance of SRFNN DFE is much improved over classical NN DFEs.

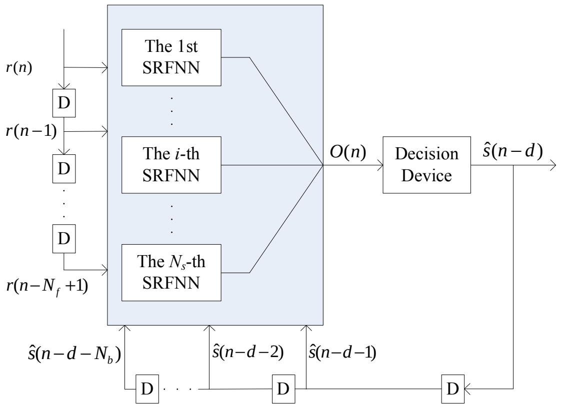


Fig. 1 Structure of SRFNN DFE

## 2. PROPOSED SRFNN DFE

The $M$-ary QAM signal can be expressed by a set of two-dimensional signals given by

$$s(n) \equiv s^R(n) + js^I(n), \quad (1)$$

where $j \equiv \sqrt{-1}$. $s^R(n)$ and $s^I(n)$ exist only for a finite number of discrete values. A 4-QAM signal sequence, for example, is composed of $\{s_1, s_2, s_3, s_4\}$, where $s_1 = s_0 + js_0$, $s_2 = s_0 - js_0$, $s_3 = -s_0 + js_0$, $s_4 = -s_0 - js_0$ and $s_0$ is a constant representing the amplitude of the symbol. Due to the presence of distortions such as linear channel ISI, nonlinear effect of demodulator and AWGN, the resulted complex-valued received signal can be written as follows:

$$r(n) = g\left(\sum_a h(a)s(n-a)\right) + v(n), \quad (2)$$

where the linear channel is assumed to be a finite impulse response filter, $h(a)$ is the complex-valued channel coefficients and $g(\cdot)$ is some nonlinear function. Both real and imaginary parts of AWGN, i.e., $\mathrm{Re}[v(n)]$ and $\mathrm{Im}[v(n)]$, have zero means and equal variances $\sigma_v^2$. The purpose of DFE [1-7] is to recover the transmitted symbol $s(n-d)$ from received signals $\boldsymbol{s_f}(n) \equiv [r(n), ..., r(n-N_f+1)]^T$ with the aid of decision feedback symbols $\hat{\boldsymbol{s}}_{\boldsymbol{b}}(n) \equiv [\hat{s}(n-d-1),..., \hat{s}(n-d-N_b)]^T$, where integers $d$, $N_f$ and $N_b$ are known as the decision delay, feedforward order and feedback order, respectively. The notation $\hat{s}(n)$ represents the estimated symbol of $s(n)$.

Suppose there are $N_t$ transmitted symbols that influence the decision output of DFE at $n$:

$$\begin{aligned}\boldsymbol{s_t}(n) \equiv [s(n),..., s(n-d-1),..., \\ s(n-d-N_b),..., s(n-N_t+1)]^T\end{aligned}, \quad (3)$$

where the value $N_t \geq d + N_b + 1$ is associated with the channel order. If we don't estimate the channel order, the value $N_t$ will be unknown to DFE. As $\boldsymbol{s_t}(n)$ sequentially goes through the channel defined in (2), the feedforward input vector $\boldsymbol{s_f}(n)$ is generated. The sequence $\boldsymbol{s_t}(n)$ apparently includes the correct feedback symbols $\boldsymbol{s_b}(n) \equiv [s(n-d-1),...,s(n-d-N_b)]^T$. It is well-known that the equalization process can be viewed as a classification problem in which the feedforward input vector set $R_d$ is divided into one of the symbol points $s_m$, $m = 1 \sim M$. By means of the concept of the classification, the set of $\boldsymbol{s_t}(n)$ can be partitioned into $M^{N_b}$ subsets due to $\boldsymbol{s_b}(n)$ involving $M^{N_b}$ feedback states $\boldsymbol{s_{b,i}}$, $i$ = 1, …, $M^{N_b}$. The set $R_d$ thus can be divided into $M^{N_b}$ subsets accordingly:

$$R_d = \bigcup_{1\leq i \leq N_s} R_{d,i}, \quad (4)$$

where $R_{d,i} = \{\boldsymbol{s_f}(n) | \boldsymbol{s_b}(n) = \boldsymbol{s_{b,i}}\}$ and $N_s = M^{N_b}$. Since each feedback state $\boldsymbol{s_{b,i}}$ occurs independently, we propose the DFE established with $N_s$ SRFNN-based equalizers as shown in Fig. 1. For feedforward input vectors with feedback state $\boldsymbol{s_{b,i}}$, the $i$-th SRFNN can further classify subset $R_{d,i}$ into $M$ subsets based on $s(n-d)$:

$$R_{d,i} = \bigcup_{1\leq m \leq M} R_{d,i}^{(m)}, \quad (5)$$

where $R_{d,i}^{(m)} = \{\boldsymbol{s_f}(n) | (\boldsymbol{s_b}(n) = \boldsymbol{s_{b,i}}) \wedge (s(n-d) = s_m)\}$, $m = 1 \sim M$. From the viewpoint of the symbol-by-symbol equalization, as receiving a feedfoward input vector $\boldsymbol{s_f}(n)$ with $\boldsymbol{s_{b,i}}$ being its feedback state at $n$, the SRFNN DFE only activates the $i$-th SRFNN to equalize this feedfoward input vector.

The detailed formulas in the $i$-th SRFNN [10] for the proposed DFE are described here. We redefine the complex vector $\boldsymbol{s_f}(n)$ as

$$\begin{aligned}&[r(n),..., r(n-N_f+1)]^T \\ &= [x_1(n),..., x_{N_f}(n)]^T\end{aligned}. \quad (6)$$

For convenience, we define a notation $\Lambda$ as:

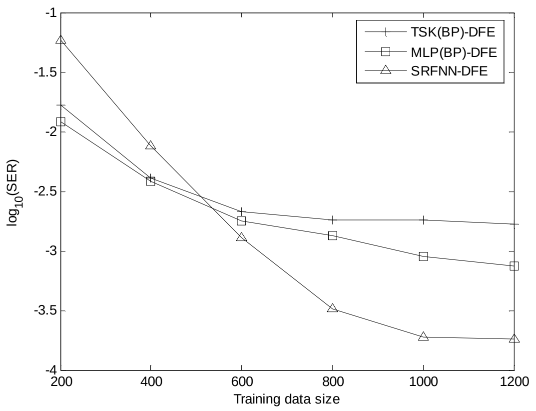


Fig. 2 Influence of the training data size on SER performance for various methods at SNR = 26 dB

$$\Lambda^C(n) \equiv \begin{cases} \mathrm{Re}[\Lambda(n)], & C = R \\ \mathrm{Im}[\Lambda(n)], & C = I \end{cases}, \qquad (7)$$

where $\Lambda$ can be changed to any kind of notation which is complex value of this paper. For a feedfoward input vector $\boldsymbol{s}_f(n)$ with $\boldsymbol{s}_{b,i}$ , the complex-valued output of the SRFNN DFE equals to the $i$-th SRFNN output:

$$O(n) = \sum_{k=1}^{K_i(n)} w_{ik}(n) O_{ik}^{(2)}(n) \,. \qquad (8)$$

where $w_{i,k}(n)$ is the $k$-th rule output, $K_i(n)$ is the number of existing rules, $O_{ik}^{(2)}(n) = \prod_k O_{ikp}^{(1)}(n)$ is the firing strength, $O_{i,k,p}^{(1)}(n)$ is

$$\exp\left\{-\left(\frac{(x_p^R(n) + \varpi_{ikp}^R(n) O_{ikp}^{(1)}(n-1) - m_{ikp}^R(n))^2}{(\sigma_{ikp}^R(n))^2} + \frac{(x_p^I(n) + \varpi_{ikp}^I(n) O_{ikp}^{(1)}(n-1) - m_{ikp}^I(n))^2}{(\sigma_{ikp}^I(n))^2}\right)\right\},$$

and $\varpi_{i,k,p}(n)$ is the recurrent coefficient [10].

The learning scheme in [10] is adopted in the proposed DFE. There are no rules initially in each SRFNN. As a feedforward input vector $\boldsymbol{s}_f(n)$ with a feedback state $\boldsymbol{s}_{b,i}$ is received at $n$, the SL and BP algorithms are performed simultaneously in the $i$-th SRFNN. The SL algorithm adopts a measure to help decide the rule generation. It is the rule evaluation measure $\lambda_{\max}$ defined as $\max_k\{O_{ik}^{(2)}(n)\}$ [10]. Then the evaluation criterion that must be met before a new rule is added is

$$\lambda_{\max} \le \lambda_{\min} \,, \qquad (9)$$

where $0 < \lambda_{\min} < 1$ and $\lambda_{\min}$ is a pre-specified threshold. This indicates that no rule in the $i$-th SRFNN can cluster $\boldsymbol{s}_f(n)$, and then a new rule must be added to improve the entire performance and cover the vector $\boldsymbol{s}_f(n)$. Once a new cluster is generated, its initial shape should be assigned:

$$m_{i,new,p}(n) = x_{\boldsymbol{p}}(n) \text{ and } \sigma_{i,new,p}(n) = \sigma \,, \qquad (10)$$

where $p = 1 \sim N_f$ , $\sigma$ is an empirically pre-specified complex value and set as 0.7 + $j$ 0.7 in this paper. The parameters $\varpi_{ikp}(n)$ and $w_{ik}(n)$ are initialized randomly.

After the SL algorithm is over in the $i$-th SRFNN, the BP algorithm is adopted subsequently to train the parameters of this $i$-th SRFNN. For more detailed derivatives, readers may refer to the reference [10].

## 3. SIMULATIONS

The performance for various DFEs is examined by using computer simulations over a 4-QAM signalling system, which is given by [2,5]

$$z(n) = (0.34 - j0.27)s(n) + (0.87 + j0.43) s(n-1) + (0.34 - j0.27)s(n-2), \qquad (11)$$

$$r(n) = z(n) + 0.1z^2(n) + 0.05z^3(n) + v(n). \qquad (12)$$

The parameters $N_f$ = 2, $N_b$ = 1 and $d$ = 1 are used in the following experiments. The experimental results are obtained by averaging 500 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 length for the training period will be given below. Two NN DFEs trained with BP algorithm, i.e., MLP DFE [2] and Takagi-Sugeno-Kang (TSK) DFE with 16 fuzzy rules [5], are simulated for comparisons. TSK is one type of GBF. Like the proposed DFE, these two NN DFEs trained by BP work without the prior knowledge of the channel order. The learning rates for various methods are determined to achieve their best performances. For MLP DFEs, the number of neurons in the first hidden layer, the number of neurons in the second hidden layer, and the number of neurons in the output layer are chosen as 10, 5 and 1, respectively.

Fig. 2 shows the influence of the training data size on the SER performance for various methods considered. The proposed DFE using SRFNN performs better than the classical NN DFEs trained by BP algorithm as iterations are larger than 500. To obtain satisfactory SER results, 1000 training data size are set hereafter for various DFEs in the simulations.

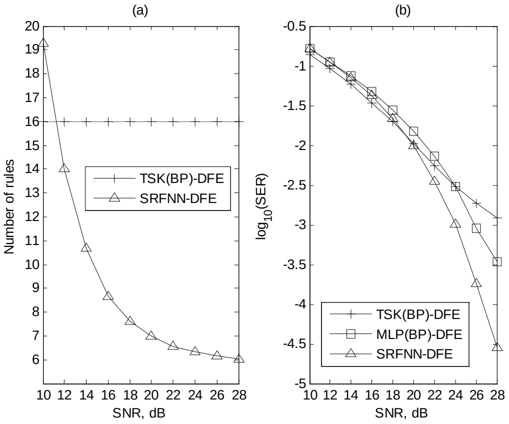


Fig. 3 (a) Average numbers of fuzzy rules for different GBF DFEs at various SNRs; (b) SER for different NN DFEs at various SNRs

Average numbers of fuzzy rules for different GBF DFEs at various SNRs are shown in Fig. 3-(a). SRFNN can intelligently determine the numbers of rules needed in computation at various SNRs. The comparison of SER curves is demonstrated in Fig. 3-(b). The SRFNN DFE performs better than classical DFEs at high SNRs.

## 4. CONCLUSIONS

This paper has presented the performance of the proposed DFE for QAM signals. When SRFNN is applied to the DFE, the DFE structure can be intelligently built during the learning procedure. Besides, the SRFNN DFE without pre-knowing channel characteristics acquires better SER performance in a higher convergence rate than classical NN DFEs, which also don't need to estimate channel characteristics in advance.

## REFERENCES

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

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

[3] 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 53 (2006) 240-244.

[4] 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 54 (2007) 171-175.

[5] 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 52 (2005) 596-600.

[6] S. Chen, B. Mulgrew, S. McLaughlin, Adaptive Bayesian equalizer with decision feedback, IEEE Trans. Signal Process. 41 (1993) 2918-2927.

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

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

[9] J. Lee, C. Beach, N. Tepedelenloglu, A practical radial basis function equalizer, IEEE Trans. Neural Netw., 10 (1999) 450-455.

[10] R.C. Lin, W.D. Weng, C.T. Hsueh, Design of an SCRFNN-based nonlinear channel equaliser, IEE Proc.-Commun., 152 (2005) 771-779.