Decision Feedback Equalizer Using Artificial Neural Networks

Ekta Gurnani
Electronics & Communication Department
DIT School of Engineering, Greater Noida
Gautam Budh Nagar, UP, India
gurnaniekta@gmail.com

Sheena Gupta
Electronics & Communication Department
DIT School of Engineering, Greater Noida
Gautam Budh Nagar, UP, India
aries.sheen@gmail.com

***Abstract*— Our problem as communication engineers is to restore the transmitted sequence or, equivalently, to identify the inverse of the channel, given the observed sequence at the channel output. This task is accomplished by adaptive equalizers. Decision feedback equalizers are used extensively in practical communication systems. This paper addresses the techniques of channel equalization by decision feedback equalizer using artificial neural network. In this paper, radial basis function (RBF) network and multi layer perceptron net (MLP) is used to implement DFE. Advantages and problems of this system are discussed and its results are then compared accordingly.**

***Keywords*— Neural networks, radial basis function (RBF), decision feedback equalizers, Multi layer perceptron net (MLP)**

I. INTRODUCTION

It is an important technique to combat distortion and interference in communication links. The conventional approach to communication channel equalization is based on adaptive linear system theory. Channel equalization is an important subsystem in a communication receiver. Equalization is a technique used to remove inter-symbol interference (ISI) produced due to the limited bandwidth of the transmission channel [1]. When the channel is band limited, symbols transmitted through will be dispersed. This causes previous symbols to interfere with the next symbols, yielding the ISI. Also, multipath reception in wireless communications causes ISI at the receiver. Thus, equalizers are used to make the frequency response of the combined channel-equalizer system flat. Two classes of equalizers are known: linear and non-linear equalizers. In a Linear Equalizer, the current and the past values of the received signal are linearly weighted by equalizer coefficients and summed to produce the output and the other class of equalizers are non linear equalizers in which feedback is used whose one type is decision feedback equalizers which is discussed in this paper.

II. DECISION FEEDBACK EQUALIZER

It is a simple nonlinear equalizer used for channel with severe amplitude distortion, uses decision feedback to cancel the interference from symbols which have already have been detected. The equalized signal is the sum of the outputs of the forward and feedback parts of the equalizer as shown in Fig.1.

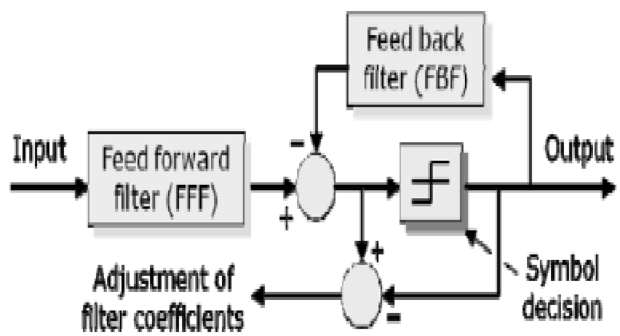

Figure 1. Decision Feedback Equalizer.

It consists of a linear feed forward filter (FFF) and a feedback filter (FBF) both are linear traversal filters, FFF suppresses the contribution of the pre-cursor ISI, i.e. the interference caused by the symbols transmitted after the symbol of interest. The FBF cancels the post-cursor ISI by subtracting a weighted linear combination of the previous symbol decisions, assumed to be correct. The result is then applied to a threshold device to determine the symbol of interest. The FFF enhances the noise, but the noise gain is not as severe as in the case of a linear equalizer. Both the forward and feedback filters may be adjusted simultaneously to minimize the Mean Square Error.

III. ARTIFICIAL NEURAL NETWORKS (ANN)

Artificial Neural Network (ANN) takes their name from the network of nerve cells in the brain. McCulloch and Pitts have developed the neural networks for different computing machines. There are extensive applications of ANN in the field of channel equalization, estimation of parameters of nonlinear systems, pattern recognition, etc. ANN is capable of performing nonlinear mapping between the input and output space due to its large parallel interconnection between different layers and the nonlinear processing characteristics.

IV. MULTILAYER PERCEPTRONS

The basic element of the multilayer perceptron is the neuron, as shown in Fig. 2.

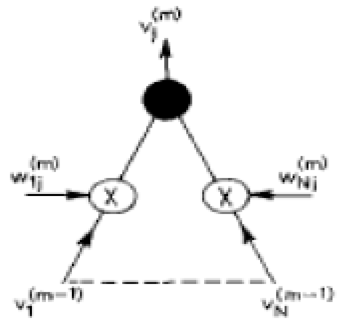

Figure 2. j^{th} neuron in m^{th} layer.

Each neuron has primarily local connections and is characterized by a set of real weights $[w_{ij},.....,w_{Nj}]$ applied to the previous layer to which it is connected and a real threshold level I_j. The j^{th} neuron in the m^{th} layer accepts inputs $V^{(m-1)} \in R^N$ from the $(m-1)^{th}$ layer and returns a scalar $v_j^{(m)} \in R$ given by

$$v_j^{(m)} = f_j\left(\sum_{i=1}^{N} w_{ij}^{(m)} v_i^{(m-1)} + I_j^{(m)}\right) \quad (1)$$

The output value $v_j^{(m)}$ serves as input to the $(m + 1)^{th}$ layer to which the neuron is connected. The nonlinearity commonly used in the perceptron is of the sigmoid type:

$$f(x) = \frac{1-e^{-x}}{1+e^{-x}} \quad (2)$$

Where f(x) lies in the interval [-1, 1] as shown in Fig. 3.

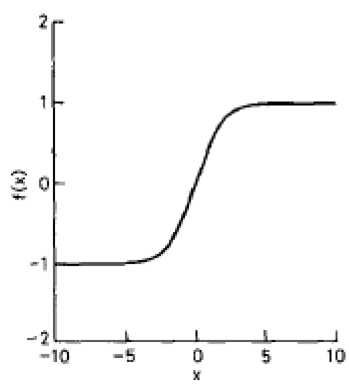

Figure 3. Activation Function.

The neurons store knowledge or information in the weights $\{w_{ij}\}$ and the weights are modified through experience or training. A multilayer perceptron (MLP) consists of several hidden layers of neurons which are capable of performing complex, nonlinear mappings between the input and the output layer as shown in Fig.4.

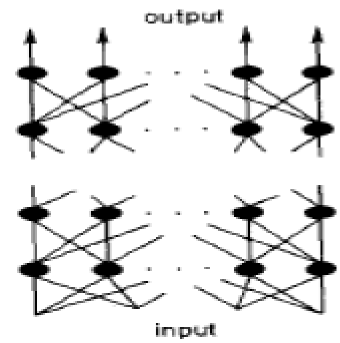

Figure 4. Multilayer Perceptron Architecture.

V. MLP BASED DECISION FEEDBACK EQUALIZER

A three-layer preceptron based decision feedback equalizer structure, as shown in Fig. 5. The input to the feed forward filter is the sequence of noisy received signal samples $\{y_n\}$. The input to the feedback filter is the output symbol decision sequence from a nonlinear symbol detector (quantizer) $\{\tilde{u}_{n-d}\}$

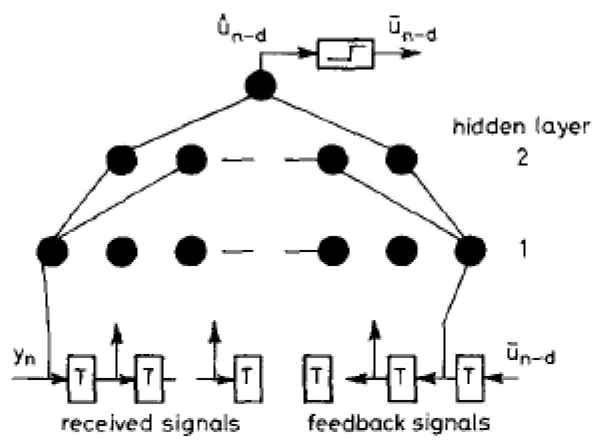

Figure 5. Multilayer Perceptron Decision Feedback Equalizer.

At time n, the input N x 1 received signal vector

$$\Gamma(n)^T = [y_n, y_{n-1}, \dots, y_{n-N+1}] \quad (3)$$

and the decision l x 1signal vector

$$[\tilde{u}_{n-d-1}, \tilde{u}_{n-d-2}, \dots, \tilde{u}_{n-d-1}] \quad (4)$$

are in the feed forward filter and feedback filter of the decision feedback equalizer, respectively, where d is a delay parameter. The decision $\tilde{u}_{n-d}$ is formed by quantizing the estimate $\tilde{u}_{n-d}$ in the output layer to the nearest information symbol.

The signals at the input layer of the decision feedback equalizer can be represented by a (N+l)x1 vector as

$$V^{(0)} = [y_n, y_{n-1}, \ldots , y_{n-N+1}; \tilde{u}_{n-d-1}, \ldots , \tilde{u}_{n-d-1}]^T \qquad (5)$$

The N1x1 vector in the output of hidden layer 1 is

$$V^{(1)} = [v1^{(1)}, v2^{(1)}, \ldots , vj^{(1)}, \ldots , vN1^{(1)}]^T \qquad (6)$$

Where

$$v_j^{(1)} = f_j\left(\sum_{i=0}^{N-1} w_{ij}^{(1)} y_{n-1} + \sum_{p=1}^{l} w_{pj}^{b(1)} \tilde{u}_{n-d-p} + I_j^{(1)}\right) \qquad (7)$$

$j = 1, 2, \ldots, N_1$

Where b denotes the feedback tap weight.

The N_2x1 vector in the output of hidden layer 2 is

$$V^{(2)} = [v_1^{(2)}, v_2^{(2)}, \ldots v_k^{(2)}, \ldots, v_{N2}^{(2)}]^T \qquad (8)$$

Where

$$v_k^{(2)} = f_k\left(\sum_{j=1}^{N_1} w_{jk}^{(2)} v_j^{(1)} + I_k^{(2)}\right) \qquad (9)$$

$k = 1, 2, \ldots, N_2$

The final output is

$$v_0^{(3)} = \hat{u}_{n-d} = f_0\left(\sum_{k=1}^{N_2} w_{k0}^{(3)} v_k^{(2)} + I_0^{(3)}\right) \qquad (10)$$

Where $\breve{u}_{n-d}$ is the estimated signal at time n. Substituting eqns. 7 and 9 into eqn. 10, yields

$$\hat{u}_{n-d} = f_0\left(\sum_{k=1}^{N_2} w_{k0}^{(3)} f_k\left(\sum_{j=1}^{N_1} w_{jk}^{(2)} f_j\left(\sum_{i=0}^{N-1} w_{ij}^{(1)} y_{n-i} + \sum_{p=1}^{l} w_{pj}^{b(1)} \tilde{u}_{n-d-p} + I_j^{(1)}\right) + I_k^{(2)}\right) + I_0^{(3)}\right) \qquad (11)$$

The nonlinear detector can be modeled as a threshold function g(x) and is defined as

$$g(\hat{u}_{n-d}) = \tilde{u}_{n-d} = \begin{Bmatrix} 1 & if\ \tilde{u}_{n-d} \geq 0 \\ -1 & otherwise \end{Bmatrix} \qquad (12)$$

The ws (weights) and *Is* (threshold levels) in eqn. 11 are values specified by the training algorithm, so that after training is finished the equalizer will self-adapt to changes in channel characteristics occurring during transmission (decision directed mode).

VI. RADIAL BASIS FUNCTION

RBF consist of a set of input vectors $\{x_i\}$ and the corresponding output vectors $\{y_i\}$ that finds appropriate transfer function that can fit noisy input vectors to produce the most appropriate output according to the given input/output vector pairs

It consists of two layers with the activation functions in the first layer being radial, and in the output layer being linear as shown in figure 6. The activation function of the first layer is called the basis function. It is a radial function characterized by being monotonically increasing or decreasing from a centre value. Examples of radial function are the thin plate spline, multi-quadratic, inverse multi-quadratic and the Gaussian functions. The Gaussian function is most commonly used because of its smooth characteristics. It is given by eqn. 13:

$$h(x) = e^{-\frac{(x-c)^z}{r^2}} \qquad (13)$$

Where, c is the centre of the function and r is its spread constant. The centre and the spread constant control the location and the spread of the decision region of the radial function, respectively. The spread constants should be chosen such that the functions cover their areas and some of the adjacent areas in the space, increasing the ability of the ANNs to generalize for noisy patterns. The output of the RBF net is given by eqn. 14:

$$X = \sum_{j=1}^{p} x_j \qquad (14)$$

Where y = $\sum_{j=1}^{n} w_j h_j(X)$

The RBF net is trained by presenting the training data vectors and the corresponding output vectors to the net and it will compute its weight matrix that minimizes the cost function C given by:

$$C = \sum_{i=1}^{p} \{y_i - \sum_{j=1}^{n} w_j h_j(X_i)\}^2$$

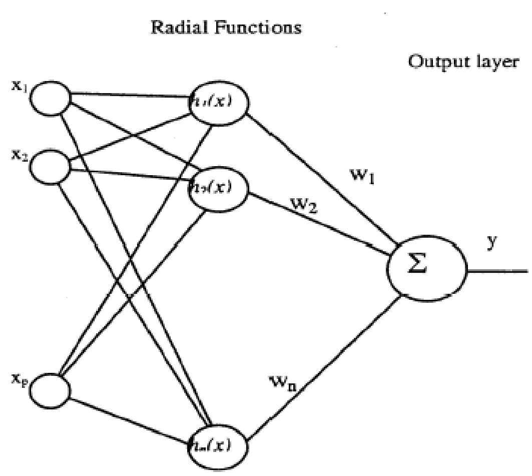

Figure 6. General Architecture of an RBF Net.

VII. THE IMPLEMENTED SYSTEM

The implemented RBF-based DFE consists of a tapped delay line that has 5 taps. At each sampling interval, the signals in the line are shifted by one location and a new received signal is put at the first tap. It is trained using 500 training samples with their corresponding outputs. It is initialized with one neuron whose activation function is Gaussian with a spread constant of 0.7. Each time, the RBF computes the weight matrix and adds one neuron if the MSE is still high. This process is repeated until the required MSE is obtained.

The hidden layer consists of 170 and 300 basis functions for the DFE and linear equalizers, respectively. This 9, 3 and 1 neuron, the nine input signals constitute a delay line of 9 taps. Both the hidden and the output layers have activation functions of the tan-sigmoid shape. The MLP net is initialized using the first training example from the channel. The training process then continues using the back propagation algorithm with a variable training rate. Upon receiving a new training example, it computes the MSE and updates its coefficients accordingly. This process is repeated recursively until the required MSE, which was set to 10^{-4}, is achieved.

VIII. SIMULATION RESULTS

The two RBF and MLP –based DFEs are used to equalize two channels that are of practical importance. The first is a linear channel having small distortions to its input. The second is a severe-ISI channel whose frequency response has a deep null. The latter type is faced often in practical communication systems and is very difficult to equalize using linear equalizers. However the two channels used are shown in Fig. 7.

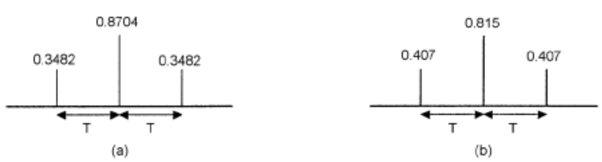

Figure 7. (a) Channel 1 (b) Channel 2.

The results of using linear equalization for channels 1 and 2 are shown in Figs. 8(a) and 8(b) respectively. The RBF based equalizer performance is better than that of the MLP based by 5 and 4 dBs, for channels 1 and 2, respectively at 10-2 bit error rate (BER). It is clear that channel 2 was not equalized well using linear equalization because of its severe ISI.

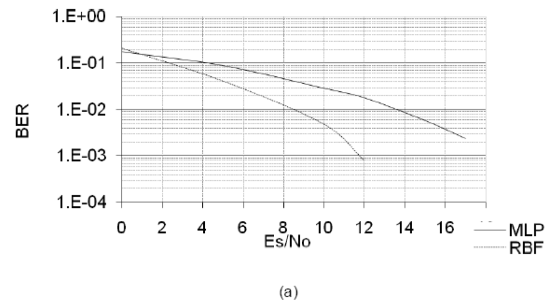

Figure 8(a) Performance of linear equalization of Channel 1.

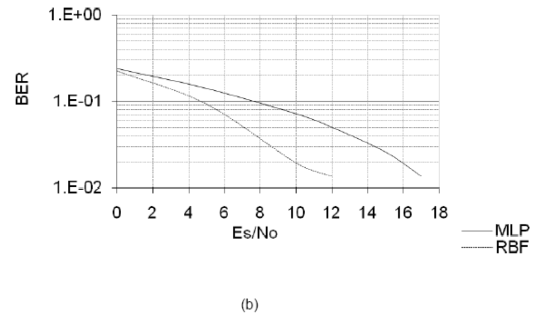

Figure 8(b) Performance of linear equalization of Channel 2.

Fig. 9(a) shows the performance of both MLP and RBF based (4, 1) DFEs for channel1. It is clear that the RBF based equalizer outperforms the MLP based one by about 4 dBs at 10-3 BER. Fig. 9(b) shows the same information as part (a) for channel 2. Also, the RBF based DFE outperforms the DFE based on MLP by about 2dB. Of course, the overall performance for channel 2 is worse than that of channel 1 because channel 2 is more severe. In both channels, the DFE based on RBF is better than the one based on MLP even when the correct symbol is fed back in the MLP and the detected one is fed back in the RBF. This means the former DFE is better than the latter always, since feeding back the correct symbol is the most ideal case.

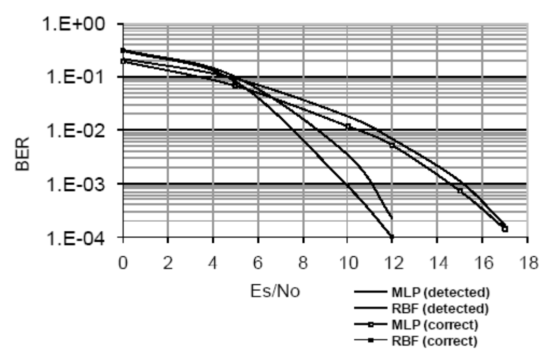

Figure 9(a) Performance of DFE of Channel 1.

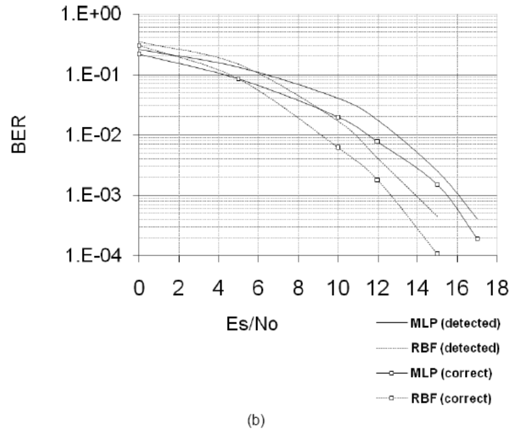

Figure 9(b) Performance of DFE of Channel 2.

Figs.10(a) and 10(b) shows the convergence of both MLP and RBF- based DFEs, respectively. Both equalizers were able to reach the required MSE but the RBF is faster. On the other hand, the RBF based DFE needs more computations in the decision directed modes.

This is due to the high no. of basis functions in the hidden layer of the RBF system compared to the MLP system.

Simulation results showed that increasing the no. of neurons in the hidden layer of the MLP will not improve the convergence time or the BER performance. So, the price paid for reducing the BER and speeding up the training process by using the RBF based DFE, is the more computations required in the decision directed mode.

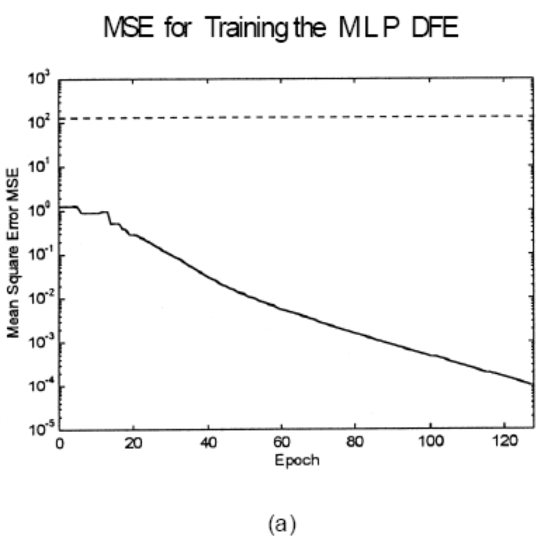

Figure 10(a) Convergence of MLP-based DFE.

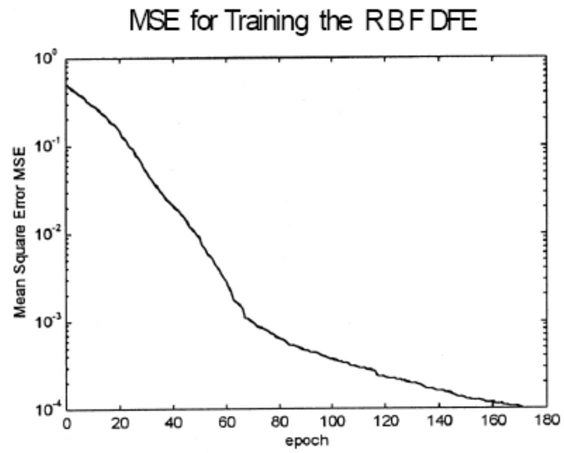

Figure 10(b) Convergence of RBF-based DFE.

IX. CONCLUSION

In this paper DFE equalizers were implemented using both MLP and RBF nets. The above systems were tested for two different channels. it is seen that the RBF based equalizers perform better than the MLP based one, especially at high SNR. Moreover, the RBF equalizer converges faster than the MLP in the training mode but need more computational time in the decision directed mode, because of its large no. of neurons compared with the MLP. Trade off between fast convergence and performance on one side and the on line computational time on the other side should be taken into consideration upon designing such systems in practice.

The DFE performs better when the correct symbol is the feedback signal that is an ideal case. They also are efficient in reducing the effect of the deep frequency null of channel 2.

REFERENCES

[1] K.C. Ho and Y.T. Chan, "Bias removal in equation-error adaptive iir filters," IEEE Trans. Signal Processing, vol. 43, no. 1, Jan 1995.

[2] J.E. Cousseau and P.S.R. Diniz, "New adaptive iir filtering algorithms based on the steiglitz-mcbride method," IEEE Trans. Signal Processing, vol. 45, no. 5, May 1997.

[3] I. M. W. K.L. Blackmore, R.C. Williamson, "Online learning via congregational gradient descent," Mathematics of Control, Signals, and Systems, vol. 10, pp. 331-363, 1997.

[4] J. M. F.Albu, A. Mateescu and B. Dorizzi, "Adaptive channel equalization using neural network".