Performance Comparison of Blind and Non-Blind Channel Equalizers using Artificial Neural Networks

Sarvraj Singh Ranhotra, Atul Kumar, Maurizio Magarini
Dipartimento di Elettronica, Informazione e Bioingegneria
Politecnico di Milano, 20133 Milano, Italy
sarvrajsingh.ranhotra@mail.polimi.it

Amit Mishra
Thapar Institute of Engineering and Technology
Patiala, India
amit_mishra@thapar.edu

***Abstract*—In digital communication systems, multipath propagation induces Inter Symbol Interference (ISI). To reduce the effect of ISI different channel equalization algorithms are used. Complex equalization algorithms allow for achieving the best performance but they do not meet the requirements for implementation of real-time detection at low complexity, thus limiting their application. In this paper, we present different blind and non-blind equalization structures based on Artificial Neural Networks (ANNs) and, also, we analyze their complexity versus performance. The simulated network is based on multilayer feedforward perceptron ANN, which is trained by utilizing the error back-propagation algorithm. The weights of the network are updated in accordance with training of the network to improve the convergence speed. Simulation results demonstrate that the implementation of equalizers using ANN provides an upper hand over the performance and computational complexity with respect to conventional methods.**

Keywords—Blind Channel Equalization; Neural Networks; Multi-Layer Perceptron.

I. INTRODUCTION

In the recent years, mobility of communicators has added new challenges in the path to accomplish the goal of providing all the information asked for in any possible location. One of the new challenges is to conceive highly reliable and fast communication systems unaffected by the problems caused by the multipath in wireless fading channels [1]. Inter-symbol interference (ISI) is one of the major problems faced practically in digital communication.

To overcome these issues the design of new equalization technique is the one of main concerns in the case of frequency selective channels. This has lead to the development of more and more complex equalization techniques, with the problem that complex algorithms do not meet the requirement for real-time implementation at low complexity and, therefore, limiting their application. Therefore, we focus on the design of equalization techniques using artificial neural network (ANN) [2].

Many equalization techniques have been proposed and implemented. In some of them, a training sequence is transmitted prior the transmission of information data, while others are able to perform equalization without the need of such a training sequence [3]. This has motivated us to the use of ANNs, which have the advantage of accuracy and also provide with a faster response.

Linear equalizers generally employ linear filters with transversal or lattice structures using different adaptation algorithms such as recursive least square (RLS), least mean square (LMS), fast RLS, square-root RLS, gradient RLS [5]. However, linear equalizers do not perform well on channels with deep spectral nulls [6]. In contrast, ANNs can form arbitrarily nonlinear decision boundaries to take up complex classification tasks [7]. Model the nonlinear phenomenon in channel equalization based on ANNs is attractive for imitating the computational function of systems using simple computation in the biological domain [8].

In this work, we implement various linear and nonlinear channel equalizers using ANNs. We consider both the two cases with and without transmission of the training sequence. When a training sequence is not available for the equalizer, blind learning algorithms have to be used. The well-known decision-directed (DD) algorithm based on least mean square (LMS) is commonly selected for its computational efficiency. However, DD strictly requires a very low level of decision errors in initial acquisition state, which is a hard task for most applications, to prevent local convergence [9]. The main issue of local convergence is that the bit error rate (BER) of the equalizer output may be worse than that of the equalizer input. In other words, the symbol detection would even be degraded due to the employment of a blind equalizer. Trying to exploit more extra information, other than the output decisions, some excellent blind equalization algorithms have been developed using the former methods that avoid local convergence. Among these, we mention the stochastic quadratic distance (SQD) algorithm where the probability density function (pdf) of the equalizer output is forced to match the pdf of the estimated constellation. In [10] the entropy of the output error is considered and an error-entropy minimization algorithm is proposed. Also, the well-known Sato [11] and Godard [12] algorithms use high-order moments of the received information. Indeed, it is proved that a fractionally spaced equalizer (FSE) using the constant modulus algorithm (CMA) could be globally convergent under a finite-length channel satisfying a length-and-zero condition.

As an alternative to linear equalizers, nonlinear equalizers have the potential to compensate distortions introduced by the channel [3]. A common nonlinear equalizer is the decision-feedback equalizer. Another class of nonlinear equalizers is based on artificial neural networks, *e.g.*, multilayer perceptron (MLP).

The main contribution of this paper is a comparison of the complexity of blind and non-blind channel equalizers with and without ANN using BER as performance index. Also, we analyse the hardware complexity by comparing CPU timing.

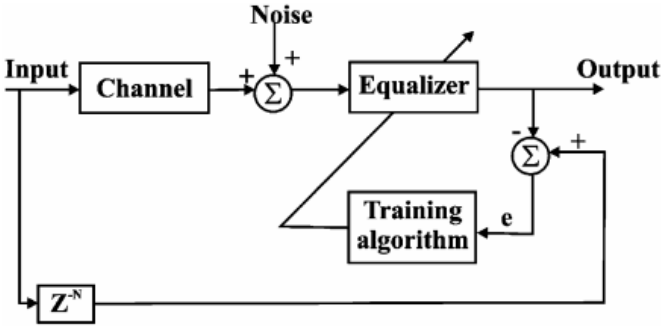

Figure 1: Block diagram of an adaptive equalizer

The remaining part of this paper is organized as follows. Section II introduces the feed forward ANN covering the MLP equalizer model and that based on radial basis function (RBF). Section III reviews the literature associated with various performance analysis of non-blind and blind based equalizers. Simulation results are provided in Sec. IV, where the BER performance of different equalizers is compared. Finally, Sec. V concludes the paper.

II. SYSTEM MODEL AND IMPLEMENTATION OF NEURAL NETWORKS

A. System Model

The discrete-time oversampled model of the received signal at the output of the time-invariant channel is written as

$$x\left(\frac{i}{m}T+(n-1)T\right)=\sum_{k=n-p}^{n-1}s_k h\left(\frac{i}{m}T-kT\right)+w\left[\frac{i}{m}T+(n-1)T\right],\qquad i=0,1\ldots,m-1,\qquad(1)$$

where $\{s_k\}$ is the sequence of symbols transmitted at baud rate T, m is the oversampling factor with respect to T, $\{w(kT/m)\}$ is the sampled zero mean additive white Gaussian noise with variance σ_w^2, and $h(iT/m)$ is the overall channel response, which is here modelled as finite impulse response (FIR) filter of length m given in [13].

By taking into account the oversampled channel output and by assuming that the length of the observation interval is L, the received signal in any interval $[iT, iT+L]$ can be expressed as [13]:

$$x(i)=Hs(i)+w(i),\quad i=0,1,2,\ldots\qquad(2)$$

where

$$x(i)=[x(iT),x(iT+\Delta),\ldots,\quad x(iT+(mq-1)\Delta)]^T,$$

Δ is the sampling period, $m=T/\Delta$, and $q=\left(\frac{L}{T}\right)\times\Delta$. The noise vector is

$$w(i)=[w(iT),w(iT+\Delta),\ldots,w(iT+(mq-1)\Delta)]^T,$$

$$s(i)=\left[s_{i-p+1},s_{i-p+2},\ldots.s_{i+p-1}\right]^T,$$

where $p=\left(\frac{L}{T}\right)$, and H is a $mq\times(p+q-1)$ matrix defined by:

$$H=\begin{bmatrix}h_1 & h_2 & . & . & . & h_n & 0 & 0 & 0 & 0 & 0 & 0\\ 0 & h_1 & h_2 & . & . & . & h_n & 0 & 0 & 0 & 0 & 0\\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots\\ 0 & 0 & 0 & 0 & 0 & 0 & h_1 & h_2 & . & . & . & h_n\end{bmatrix},\qquad(3)$$

where h_l, with $l=1,2,\ldots,p$, is a column vector defined as

$$h_l=\left[h[(p-l)T],h\left[\left(p-l+\tfrac{1}{m}\right)T\right],\ldots,h\left[\left(p-l+\tfrac{m-1}{m}\right)T\right]\right]^T.$$

B. Implementing Neural Networks

The received symbol sequence $x(i)$ at the output of the channel is applied at the input of network. By following the same model given in [20], $x(i)$ is the vector of length mq applied at the input layer. The estimated vector $\hat{x}(i)$ is compared with $x(i)$ and the resulting error vector is written as

$$e(i)=x(i)-\hat{x}(i),\qquad(4)$$

As given in [20], symbol *i* denote the *i*-th vector of the error whose *j*-th is given by

$$e_j(i)=x_j(i)-\sum_{k=1}^{M}\hat{H}_{jk}\,q_k(i),\qquad(5)$$

where $\hat{H}_{jk}$ is the element (j, k) of the estimated channel matrix $\hat{H}$. By following the same procedure described in [20], the error is given by

$$e_j=x_j-\sum_{k=1}^{M}\hat{H}_{jk}\sum_{l=1}^{N}W_{kl}x_l(i),\qquad(6)$$

where $W_{kl}=x_{kl}+jy_{kl}$ is the corresponding weight of the each neurons. The block diagram of the feedforward neural network is given in Fig. 2. Each node, representing a basic element of the neural network, is called neuron. As given in [20], the output of a neuron is given by

$$y=f\left(\sum_{i=1}^{N}w_i x_i\right),\qquad(7)$$

where x_i is the *i*-th input to a neuron, w_i is the weight associated with the *i*-th input, and $f(\cdot)$ is the activation function.

C. Multi-Layer Perceptron

Possible activation functions for multilayer perceptron's are given in [21]

- linear: $f(v)=kv$
- sigmoid: $f(v)=\frac{1}{1+e^{-v}}$
- hyperbolic tangent: $f(v)=\tanh(v)=\frac{1-e^{-v}}{1+e^{-v}}$.

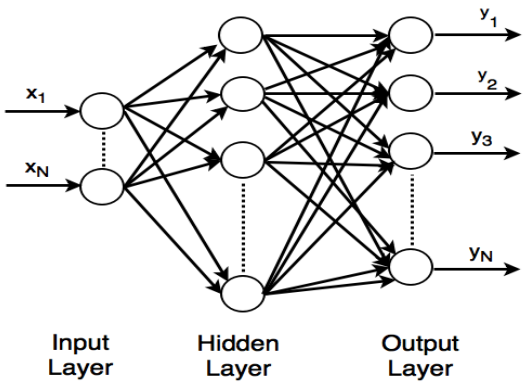

Figure 2: A feedforward neural network. Each circle represents a neuron which sums the inputs and passes the sum through an. activation function. Each arc represents multiplication by a scalar weight.

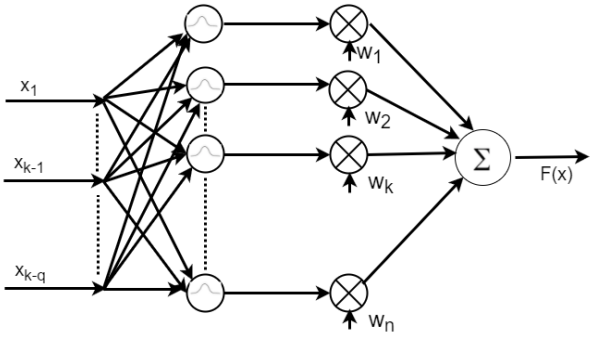

Figure 3: RBF Network

An MLP may have more than one hidden layer. The neurons in the hidden layer may use either sigmoid or hyperbolic tangent activation functions. The activation function for the output layer may be any one of the above.

D. Radial Basis function

Since there is no guarantee that an MLP would converge to a global minimum, RBF networks are a key alternative. RBFs have only three layers (one input, one hidden, and one output) [14]. The *k*-th output is given by

$$y_k = \sum_{i=1}^{N_h} w_{ki} \emptyset(x), \qquad (8)$$

where N_h is the number of neurons in hidden layer and $\emptyset_i(x)$ is a radially symmetric scalar function with N_h centers of the radial basis function. A commonly used radial basis function $\emptyset_i(x)$ is the Gaussian function

$$\emptyset_{i(x)} = \exp\left(-\frac{\|x - c_i\|^2}{2\sigma_i^2}\right), \qquad (9)$$

where $\|.\|$ is a norm, *e.g.*, Euclidean. A radial basis function is local in its characteristic response to the input x and it drops off quickly for input values that are away from the center of the activation function's receptive field, c_i.

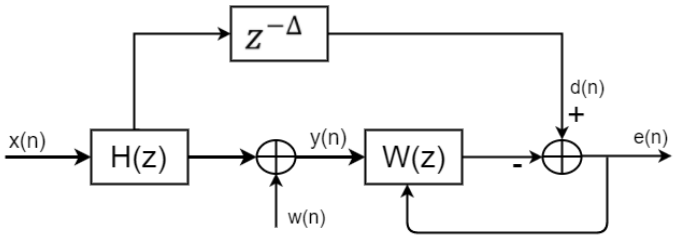

Figure 4: Adaptive channel equalization

III. BLIND AND NON BLIND EQUALIZERS

In this Sec. performance analysis of various equalizers is considered by a broad classification of them into Non-blind and Blind equalizers [15]. Non-blind algorithms implement supervised learning approaches where training sequences are used. In supervised learning, a machine can infer a function from labelled training data. The main issue is the amount of training data available relative to the complexity of the "true" function (classifier or regression function). If the true function is simple, then an "inflexible" learning algorithm with high bias and low variance will be able to learn it from a small amount of data. However, if the true function is highly complex because, for example, it involves complex interactions among many different input features and behaves differently in different parts of the input space, then the function will only be learnable from a very large amount of training data and using a "flexible" learning algorithm with low bias and high variance. A different scenario occurs when the process of parameter optimization and/or adaptation cannot be guided because a reference signal is by no means available. Channel equalization is an unsupervised problem by nature. In addition to that, such a problem is characterized by the requirement of real-time and low computational burden, due to the practical operation conditions of a communication system.

III.1. NON BLIND EQUALIZERS –SUPERVIZED

A. Zero Forcing (ZF)

ZF Equalizer [16] refers to a form of linear equalization algorithm used in communication systems which applies the inverse of the frequency response of the channel. The name Zero Forcing corresponds to bringing down the inter-symbol interference (ISI) to zero in a noise free case. This will be useful when ISI is significant compared to noise. Thus the combination of channel and equalizer gives a flat frequency response and linear phase. Apart from it being lucrative choice for an equalizer it suffers from drawbacks such as infinite channel impulse response.

B. Adaptive Minimum Mean Square Equalizer (MMSE)

The adaptive MMSE equalizer is a classic approach that has been widely used in digital communication systems [17]. The MMSE equalizer is obtained by minimizing the cost function

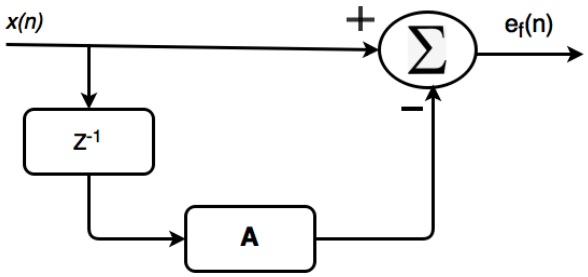

Figure 5: Structure of Forward Linear Predictor.

$$\xi = E|d(n) - w^T(n)y(n)|^2 \quad (10)$$

with respect to the tap vector $w(n)$. The input vector is given by

$$y(n) = [y(n) \dots y(n-N+1)], \quad (11)$$

where N is the tap-length and $d(n) = x(n-\Delta)$. The decision delay Δ determines which symbol is being detected at the current time n, or the current equalization output $z(n)$ is an estimate of $x(n-\Delta)$. The maximum potential performance of an MMSE equalizer is achieved by the ideal equalizer with $N \rightarrow \infty$ extending from -∞ to ∞.

III.2. BLIND EQUALIZERS-NON SUPERVISED

A. Blind Constant Modulus Algorithm

The most commonly used adaptive algorithm for blind channel equalization is the Constant Modulus Algorithm (CMA), which uses the constant modularity of the signal as the desired property [18]. CMA assumes that the input to the channel is a modulated signal that has constant amplitude at every instant in time. CMA attempts to accomplish this objective by forcing the output of the equalizer to have constant amplitude. It can also be used for QAM signals where the amplitude of the modulated signal is not the same at every time instant. The error is then determined by considering the nearest valid amplitude level of the modulated signal at the desired value as

$$J_{CM} = E(|x(n)|^2 - R_2)^2, \quad (12)$$

where $x(n)$ is the received signal and $R_2$is Godard dispersion constant given by

$$R_2 = \frac{E[x(n)^4]}{E[x(n)^2]}. \quad (13)$$

The constant R_2 depends on a priori statistical information about transmitted signal. Equalizer coefficients' update equation in CMA uses a gradient descent to minimize JCM by as illustrated in Fig. 4, that is given by

$$w(n+1) = w(n) - \mu y(n)x(n)\left[abs\left(x(n)\right)^2 - R_2\right], \quad (14)$$

where $y(n)$ is the output, $w(n)$ is current equalizer coefficient, $w(n+1)$ is next equalizer coefficient, μ is the step size and R_2 is the Godard constant.

TABLE 1: SETTINGS FOR THE NEURAL NETWORK

Tap Length of equalizer(L)	20
Number of input Neuron	L+1
Number of output Neuron	1
Number of hidden Neuron	15
Number of training patterns	1000

B. Blind Linear Predictive Equalizer

Blind equalization based on linear prediction is one of the methods for blind multiuser case [19]. The idea is to obtain an estimate for the received vector $x(n)$ as a linear combination of the vectors $x(n-1), \dots, x(n-K+1)$, *i.e.*, the components of $x(n-1)$. The estimate can be expressed as

$$x(n) = AH(1)x(n-1) + \cdots + AH(K-1)x(k-K+1) \quad (15)$$

where A is a $P(K-1) \times P$ matrix composed by the$(K-1)$ $P \times P$ matrices of prediction coefficients.

$$A = \left[A^H(1) \dots A^{H(K-1)}\right]^H. \quad (16)$$

The forward prediction error is given by:

$$e_f(n)|_{x_{(n-1)}} = x(n) - \hat{x}(n)|_{x(n-1)} = [I - A^H]x(n) \quad (17)$$

The operation of a forward linear multichannel predictor is illustrated in Figure 5. A $P \times P$ matrix with the forward prediction-error variance is defined by

$$\sigma_{e_f}^2 = [I - A^H]\mathbb{R}_x^{(K)}(n)[I - A^H]^H, \quad (18)$$

where $\mathbb{R}_x^{(K)}(n) = E[\mathbb{x}(n)\mathbb{x}(n)^H]$ and K indicates the number of time instances taken into account. The minimization of the variance of the prediction error, $\sigma_{e_f}^2$, leads to the following optimization problem:

$$\min_A \left[I_p - A^H\right] \mathbb{R}_x^{(K)}(n)[I - A^H]^H = \sigma_{e_f}^2 \quad (19)$$

Upon solving the minimization results according to equation

$$\left[I_p - A^H\right]\mathbb{R}_x^{(K)}(n) = \left[\sigma_{e_f}^2 \;\; 0_p \dots . 0_p\right] \quad (20)$$

IV. SIMULATION RESULTS AND CONFIGURATION SETTINGS

In this section, to have a collective performance index, we consider BER as the performance figure of merit. The following minimum phase channel with five taps is considered [20]:

$$h = [0.0545 + 0.05i \;\; 0.2832 - 0.11971i \;\; -0.7676 \;\; 0.2788i \;\; -0.0641 - 0.0576i \;\; 0.0466 - 0.02275i]$$

TABLE 2: COMPARISON OF A HARDWARE AND TIME COMPLEXITY

Method	Number of Additions	Number of Multiplications	Time Complexity
Adaptive CMA	346	349	0.085675sec
Non Blind Neural Network	330	330	0.043085sec
Blind, Neural Network	330	330	0.060391sec

Random data generation at the input account for the presence of $3 \cdot 10^5$ symbols drawn from a QPSK constellation, 10^5 being total number of data symbols and $2 \cdot 10^5$ training symbols. The SNR is varied from 0 to 30 dB with an increment of 2 dB. The input signal is passed through the complex multipath channel h. Implementation settings of Blind and Non Blind equalizers with ANN are reported in Table 1 [15].

Two activation functions have been used in the equalizers implemented by ANN: the hyperbolic tangent function for the hidden layer and the identity linear function for the output layers. The complexity of the addition and multiplication when bias is used for all neurons can be calculated in neural networks by a general formula given in [22] as

$$\#\text{ of multiplication} = n \times \rho + m \times \rho \qquad (21)$$

$$\#\text{ of addition} = n \times \rho + m \times \rho, \qquad (22)$$

where n is number of neuron in input layer, m is the number of neurons in output layer, and ρ is the number of hidden layer.

Figure 6 reports BER versus E_b/N_o performance for QPSK modulation for ZF and MMSE equalizers. The theoretical BER is also reported as a reference. In Fig. 7 the BER performance of the linear predictive equalizer with and without ANN is shown. From the results it is clearly visible that we have same performance with and without ANN. In Fig. 8 similar behavior is observed for the BER performance of Non-Blind equalizers of ZF with and without ANN. From the results it is clearly visible that by using Neural network a better performance is achieved. Finally, Fig. 9 depicts the overall performance of the both blind and non-blind equalizers, with and without neural networks. The equalizers implemented using ANN outperform the conventional ones both in terms of BER performance and and in terms of number of additions and multiplications required for the hardware implementation, as reported in Table 2.

V. CONCLUSION

From the presented results it is clear that BER performance is better for channel equalizers implementing conventional method based on ANN as compared to equalizers that do not implement it. Also in terms of hardware and time complexity (CPU time), channel equalizers implemented by neural network performs better than conventional method. There would be an undermining deteriorating effect in case of blind approaches, since it is not trained. We were able to successfully mimic the BER performance blind channel equalizer namely blind linear

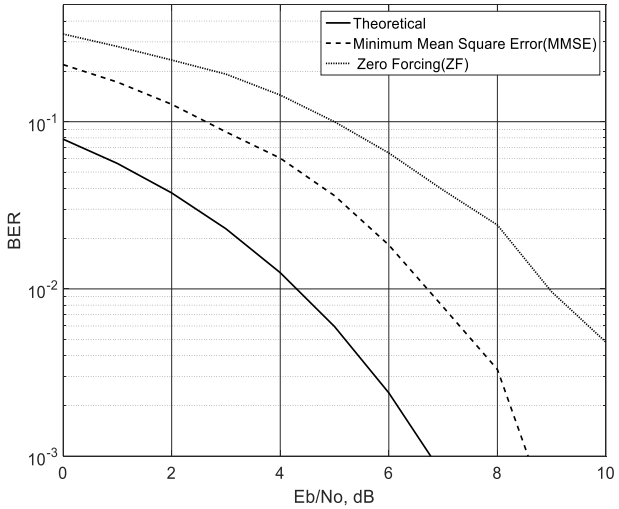

Figure 6: Comparison of performance analysis of Zero Forcing and Minimum Mean Square Equalizer with theoretical BER of QPSK.

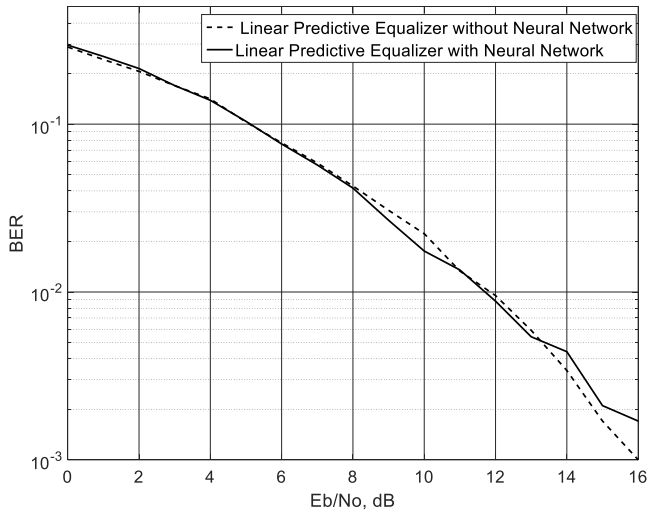

Figure 7: Comparison of Performance of Blind Linear Predictive Equalizers with and without Neural Networks.

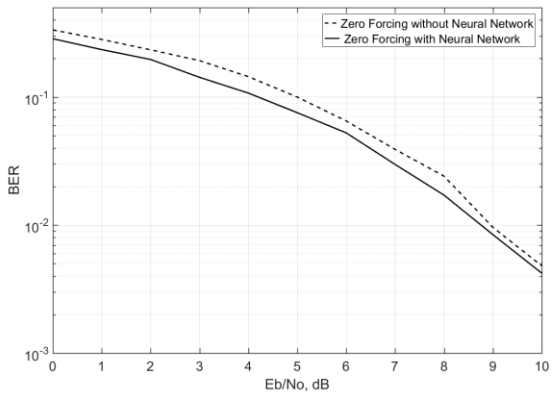

Figure 8: Comparison of Neural Network implemented equalizers for Blind and Non Blind.

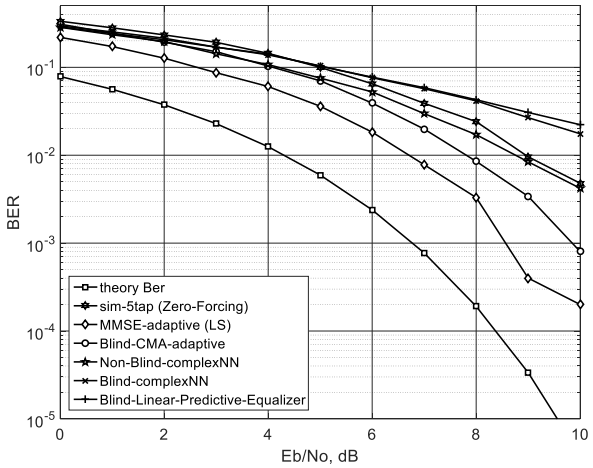

Figure 9: Overall Performance evaluation of various equalizers

predictive equalizer using ANN. A significant improvement of the performance is observed for the conventional equalizers, namely non-blind ZF when ANN is used.

REFERENCES

[1] S. Bang, S. H. Sheu, and J. Bing, "Neural network for detection of signals in communication," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 43, no. 8, pp. 644–655, Aug. 1996.

[2] J.M Cioffi., "When do I use an RLS adaptive filter?," in Proc. of *Asilomar Conference on Circuits, Systems and Computers*, 1985, pp. 636—639.

[3] J. John Hopfield, "Neural networks and physical systems with emergent collective computational abilities," *Proceedings of the national academy of sciences*, vol. 79, no. 8 pp. 2554-2558, 1982.

[4] D. W. Tank and J. J. Hopfield, "Simple 'neural' optimization networks: an A/D converter, signal decision circuit, and a linear programming circuit," *IEEE Transactions on Circuits and Systems*, vol.33 no.5, pp. 533-541, 1986.

[5] Ali H. Sayed and K. Thomas, "A state-space approach to adaptive RLS filtering," *IEEE Signal Processing Magazine,* vol. 11, no. 3, pp. 18-60 1994.

[6] S. Bang, S. H. Sheu, and J. Bing, "Neural network for detection of signals in communication," *IEEE Trans. Circuits Syst. I*, vol. 43, no. 8, pp. 644–655, Aug. 1996.

[7] B. Widrow and M. A. Lehr, "30 years of adaptive neural networks: perceptron, madaline and backpropagation," *Proc. IEEE*, vol. 78, no. 9, pp. 1415–1442, Sep. 1990.

[8] Magarini, Maurizio, Arnaldo Spalvieri, and Guido Tartara. "The mean-square delayed decision feedback sequence detector." IEEE transactions on communications 50.9 (2002): 1462-1470.

[9] E. Eleftheriou and D. Falconer, "Adaptive equalization techniques for HF channels," *IEEE Journal on Selected Areas in Communications,* vol, 5, no. 2, pp. 238-247, 1987.

[10] J. Shore and J. Rodney, "Properties of cross-entropy minimization," *IEEE Transactions on Information Theory,* vol. 27, no. 4 pp. 472-482, 1981.

[11] G. Picchi and G. Prati, "Blind equalization and carrier recovery using a stop-and-go decision-directed algorithm," *IEEE Trans. Commun,* vol. 35, no. 9, pp. 877-887, 1987.

[12] D. Godard, "Self-recovering equalization and carrier tracking in two-dimensional data communication systems," *IEEE Trans. Commun,* vol. 28, no. 11, pp. 1867-1875, 1980.

[13] Y. Fang and T.W.S. Chow, "Blind equalization of a noisy channel by linear neural network." *IEEE Trans. Neural Netw., v*ol. 10, no. 4, pp. 918-924, 1999.

[14] J. Lee, C. Beach, and N. Tepedelenlioglu, "A practical radial basis function equalizer," *IEEE Trans. Neural Netw.*, vol. 10, no. 2, pp. 450–455, Mar. 1999.

[15] D. Godard and P. E. Thirion, "Method and device for training an adaptive equalizer by means of an unknown data signal in a quadrature amplitude modulation transmission system," U.S. Patent No. 4,227,152. 7 Oct. 1980.

[16] M. Goursat and A. Benveniste, "Blind equalizers," *IEEE Trans. Commun.,* vol. 28., no. 11, pp. 871-883, 1984.

[17] Y. Gong, X. Hong, and K. F. Abu-Salim, "Adaptive MMSE equalizer with optimum tap-length and decision delay," *IET Sensor Signal Processing for Defence*, pp. 1-5, 2010.

[18] S. Abrar and A. K. Nandi, "An adaptive constant modulus blind equalization algorithm and its stochastic stability analysis," *IEEE Signal Proc. Lett.*, vol. 17, no. 1 pp. 55-58, 2010.

[19] G. Kechriotis, E. Zervas, and E. S. Manolakos. "Using recurrent neural networks for adaptive communication channel equalization," *IEEE Trans. Neural Netw.,* vol. 5, pp. 267-278, 1994.

[20] A. Naveed *et al*, "Blind equalization and estimation of channel using artificial neural networks," in Proc. *of Intern. Multitopic Conf.,* 2004.

[21] W. D. Ruck *et al*, "The multilayer perceptron as an approximation to a Bayes optimal discriminant function," *IEEE Trans. Neural Netw.*, vol. 1, no. 4, pp. 296-298, 1990.

[22] K.-S. Oh and J. Keechul, "GPU implementation of neural networks," *Pattern Recognition*, vol.37, no. 6, pp. 1311-1314 37.6, 2004.