International Journal of Electrical, Electronic and Communication Sciences
ISSN: 2517-9438
Vol:1, No:9, 2007

Complex-Valued Neural Networks for Blind Equalization of Time-Varying Channels

Rajoo Pandey

***Abstract*—**Most of the commonly used blind equalization algorithms are based on the minimization of a nonconvex and nonlinear cost function and a neural network gives smaller residual error as compared to a linear structure. The efficacy of complex valued feedforward neural networks for blind equalization of linear and nonlinear communication channels has been confirmed by many studies. In this paper we present two neural network models for blind equalization of time-varying channels, for M-ary QAM and PSK signals. The complex valued activation functions, suitable for these signal constellations in time-varying environment, are introduced and the learning algorithms based on the CMA cost function are derived. The improved performance of the proposed models is confirmed through computer simulations.

***Keywords*—**Blind Equalization, Neural Networks, Constant Modulus Algorithm, Time-varying channels.

I. Introduction

THE cancellation of inter-symbol interference (ISI) using an adaptive equalizer has been studied for several years by the signal processing community [1]. The classical methods of channel equalization rely on transmitting the training signal, known in advance by the receiver. The receiver adapts the equalizer so that its output closely matches the known reference-training signal. For time-varying situations, the training signals have to be transmitted repeatedly. Inclusion of such training signals, sacrifices valuable channel capacity. Therefore, to reduce the overhead of transmission of training signals, the equalization without using the training signals i.e. blind equalization is required. Thus, the term, blind equalization, means retrieving the information regarding the transmitted signal or the channel by analyzing the characteristics of its output, and some information about the system or the transmitted sequence but not the sequence itself [2].

Over the past few years many blind adaptive algorithms have been developed for the recovery of digitally modulated signals [3-9]. The blind equalization techniques can be broadly classified as based on higher order statistics or based on second order statistics. Some of the approaches aim at estimating the system function and then determining the input signals from these estimates, while others can directly estimate the input signals from the system outputs [2, 10]. For the implementation of blind adaptive algorithms, linear feedforward filter structures have been used. However, some of the studies are focused on decision feedback equalization [11, 12]. Higher order statistics-based blind equalization methods rely on optimization of some cost function. These cost functions of blind algorithms are non-convex and nonlinear functions of tap weights, when implemented using linear FIR filter structures. A linear FIR filter, however, has a convex decision region, and hence, is not adequate to optimize such cost functions. Therefore, a blind equalization scheme with a nonlinear structure that can form nonconvex decision regions is desirable [13].

Neural networks, often referred to as an emerging technology, have grown very rapidly on many fronts during the past few years. They have been used in many signal processing applications including nonlinear signal processing, real time signal processing, and adaptive signal processing [14-17]. The feedforward neural networks have been widely used for blind equalization [3, 4, 7-9, 13]. Application of recurrent neural networks for blind equalization can be found in [18-20]. However, most of these studies are limited to real valued signals and channel models. Therefore, the development of neural network-based equalization schemes is desirable for complex-valued channel models. One such study of blind equalization schemes is available in [13], but is limited to M-ary QAM signal in stationary channels only.

In general, complex data can be handled in two different ways. One way is to treat the real and imaginary parts of each complex data as two separate entities. In this case, the weights of two real valued neural networks are updated, independently. The other way is to assign complex values to the weights of neural network and update using a complex learning algorithm such as complex backpropagation algorithm (CBP). Some studies [13, 21] have shown that a complex valued MLP yields more efficient structure than two real valued MLP's.

The constant modulus algorithm (CMA) is considered to be the most successful among the HOS-based blind equalization algorithms. The cost function of the constant modulus algorithm is based solely on the amplitude of the received signals. Therefore, it is more robust than other Bussgang algorithms with respect to carrier phase offset [2, 22-24]. Thus, in this paper, the learning algorithms, to train the complex-valued multilayer feedforword neural networks for M-ary QAM and PSK signals, are based on minimization of

Manuscript received August 9, 2004. The Author is with the National Institute of Technology, Kurukshetra, India. (phone: +91-01744–238498-242; fax: +91-01744-238050; e-mail: rajoo_pandey@ rediffmail.com).

the CMA cost function.

In order to demonstrate the performance of the blind equalizers based on the proposed models in time-varying environment, the simulation of a time-varying channel is considered here. We present the models by introducing a modification to the activation function of [13] for M-ary QAM signals, and then defining new activation functions for M-ary PSK signals. A new algorithm to correct the arbitrary phase shift inherent in a blind algorithm, for the improved performance of the blind equalizer, in case of M-ary QAM signals, is also given. The paper is organized as follows. In section II the equalizer models are presented. Sections III-IV describe the training algorithms of the equalizers while simulation is presented in section V. Finally the conclusions are given in section VI.

II. Neural Network Models

A. Structure

The structure of a complex valued feedforward network for blind equalization with M-ary QAM and PSK signals is shown in Fig 1. The network has *N* input nodes, *H* hidden layer nodes and one output node. The complex valued weight $w_{kl}^{(1)}$ denotes the synaptic weight, connecting the node *l* of input layer to the input of neuron *k* in the hidden layer and $w_k^{(2)}$ refers to the synaptic weight connected between neuron *k* of hidden layer and the output neuron.

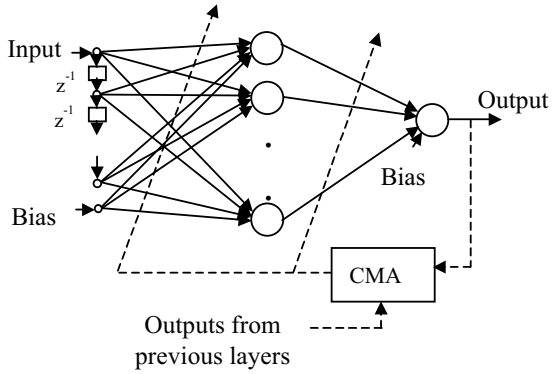

Figure1: Structure of the Neural Network Equalizer

A signal sequence of independent and identically distributed (iid) data is transmitted through a linear channel whose output along with the additive noise is denoted by $x(t)$. The input to the equalizer is formed by *N* samples of the received signal as

$$\mathbf{x}(n) = [x(n), x(n-1), \ldots . x(n-N+1)]^T \quad (1)$$

and represented as

$$\mathbf{x}(n) = [x_1(n), x_2(n), \ldots . . x_N(n)]^T \quad . \quad (2)$$

The activation sum $net_k^{(1)}(n)$ and the output $u_k(n)$ of neuron *k* in the hidden layer, are given as

$$net_k^{(1)}(n) = net_{k,R}^{(1)}(n) + j net_{k,I}^{(1)}(n)$$

$$= \sum_{l=1}^{N} w_{kl}^{(1)}(n) x_l(n) + \theta_k^{(1)}(n) \quad (3)$$

and $u_k(n) = \varphi^{(1)}(net_k^{(1)}(n)) \quad ; k = 1, 2, \ldots . . H \quad (4)$

where $net_{k,R}^{(1)}(n)$ and $net_{k,I}^{(1)}(n)$ are, respectively, the real and imaginary parts of the activation sum $net_k^{(1)}(n)$, at time *n*, $\varphi^{(1)}(.)$ represents the nonlinear activation function of neurons in hidden layer and $\theta_k^{(1)}(n)$ denotes the bias term for neuron *k* of the hidden layer.

For the neuron of the output layer, the activation sum and the output are expressed as

$$net^{(2)}(n) = net_R^{(2)}(n) + j net_I^{(2)}(n)$$

$$= \sum_{k=1}^{H} w_k^{(2)}(n) u_k(n) + \theta^{(2)}(n) \quad (5)$$

and $y(n) = \varphi^{(2)}(net^{(2)}(n)) \quad (6)$

where $y(n)$ denotes the output of the equalizer, $net_R^{(2)}(n)$ and $net_I^{(2)}(n)$ are, respectively, the real and imaginary parts of the activation sum $net^{(2)}(n)$, at time *n*, $\varphi^{(2)}(.)$ is the activation function, and $\theta^{(2)}(n)$ denotes the bias term for the neuron in output layer.

B. Activation Functions

Although the structure of blind equalizer is same for both QAM and PSK signal constellations, the activation functions used for these signal constellations are different. The nonlinear activation functions perform the decorrelation of input signal and thus help in equalization [13]. The choice of activation functions, therefore, plays an important role in the performance of the blind equalizers. It has been observed that, the use of different nonlinear functions in different layers gives better results in some channel models [9]. In the present model of complex-valued neural blind equalizers, different nonlinear functions are chosen for hidden and output layer neurons. For the neurons of hidden layer, the activation function $\varphi^{(1)}$ is described as

$$\varphi^{(1)}(z) = \varphi^{(1)}(z_R) + j\varphi^{(1)}(z_I) \quad (7)$$

where z_R and z_I are the real and imaginary parts of the complex quantity *z*, and $\varphi^{(1)}(.)$ is a nonlinear function. For the neuron of output layer, the activation function is described as

$$\varphi^{(2)}(z) = \varphi^{(2)}(z_R) + j\varphi^{(2)}(z_I) \quad (8)$$

where $\varphi^{(2)}(.)$ is another nonlinear function.

With M-ary PSK signals, for the neurons in the hidden layer, following function can be used.

$$\varphi^{(1)}(x) = \alpha \tanh(\beta x) \quad (9)$$

while α and β are two real constants. For the node of the output layer , the activation function is given by

$$\varphi^{(2)}(z) = f_1(|z|).\exp(j f_2(\angle z))$$

$$= f_1(|z|).\cos(f_2(\angle z)) + j f_1(|z|).\sin(f_2(\angle z)) \quad (10)$$

where $|z|$ and $\angle z$ denote the modulus and the angle of a complex quantity z. The function $f_1(.)$ and $f_2(.)$ are defined as

$$f_1(|z|) = a\tanh(b.|z|) \tag{11}$$

and

$$f_2(\angle z) = \angle z - c\sin(n\angle z) \tag{12}$$

where a, b and c are real constants and m is the order of PSK signals.

In case of M-ary QAM Signals, for the hidden layer nodes, following function can be used.

$$\varphi^{(1)}(x) = x + d\sin(\pi x) \tag{13}$$

where d is a positive constant. The activation function of (13) is found to be suitable for M-ary QAM signals in stationary channels [13], and has been used in both hidden and output layers. However, these functions are most effective in the output layer because of their saturation characteristics. For nonstationary channels, we define a modified activation function for the output layer, as the function defined in (13) cannot correct the arbitrary phase of the equalizer's output signal in time-varying channels. In case of the output node, the function is defined as

$$\begin{aligned}\varphi^{(2)}(z) = {} & f(z_R\cos\theta(n) - z_I\sin\theta(n)) \\ & + jf(z_R\sin\theta(n) + z_I\cos\theta(n))\end{aligned} \tag{14}$$

where $z = z_R + jz_I$ is a complex quantity, the function $f(.)$ is defined as $f(x) = x + d\sin(\pi x)$, and d is a positive real number. The parameter $\theta(n)$, introduced in (14), is used for the correction of the phase of the output signal and is updated using the algorithm given in the next section.

III. Phase Correction

The activation functions defined by equations (11)-(12) and (13)-(14) have multisaturation characteristics in accordance with the M-ary signal constellations. Due to these characteristics of the nodes, the networks become robust to noise and can also correct the arbitrary phase shifts of the output symbols [9, 13]. However, since the neural blind equalizer cannot correct the arbitrary phase shift of the output symbols automatically, in case of the time varying channels with M-ary QAM signals, therefore, some external phase correction algorithm that can continually correct the arbitrary phase shift of output symbols is required. One such algorithm is mentioned in [12]. However, we propose a simpler algorithm, inspired by the CMA cost function, to correct the arbitrary phase shift as described below.

The phase angle $\theta(n)$ in the phase shift term $e^{-j\theta(n)}$ in the output of the equalizer is recursively updated by an adaptive algorithm. To take the full advantage of multi-saturated activation function of the neural network, the phase correction should be applied to the signal in the network, before it passes through the nonlinearity of the output node. For example, the arbitrary phase shift can be corrected at the output of the hidden layer nodes. However, to achieve this objective, we propose to use the modified activation function of the output layer neuron as described in (14). The parameter $\theta(n)$, introduced in (14) for phase correction is present in the phase shift term, and is adapted by the following update equation.

$$\begin{aligned}\theta(n+1) = {} & \theta(n) + \mu\,\mathrm{Im}[net^{(2)}(n)].\mathrm{Re}[net^{(2)}(n)] \\ & .\left((\mathrm{Re}[net^{(2)}(n)])^2 - \bar{R}_2\right)\end{aligned} \tag{15}$$

where

$$\bar{R}_2 = E\left[(\mathrm{Re}[net^{(2)}(n)])^4\right] / E\left[(\mathrm{Re}[net^{(2)}(n)])^2\right] \tag{16}$$

The constant $\bar{R}_2$ = 8.2 for 16 – QAM signal. The operators Re[.], Im[.] and E[.], respectively, denote the real part, the imaginary part and the expectation.

IV. Update Rules

A. PSK Signals

The updating rules for the weights of the blind equalizer for PSK signals are described as follows.

For the weights, connected between hidden layer and output layer:

$$w_k^{(2)}(n+1) = w_k^{(2)}(n) + \eta\delta^{(2)}(n)u_k^*(n) \tag{17}$$

where, $\delta^{(2)}(n)$ is given as

$$\begin{aligned}\delta^{(2)}(n) = {} & (|y(n)|^2 - R_2)|y(n)|(ab - \frac{b}{a}|y(n)|^2) \\ & .(net^{(2)}(n)/|net^{(2)}(n)|)\end{aligned} \tag{18}$$

In (18), the parameter R_2 depends on the statistical characteristics of the signal sequence, as defined in the Appendix, whereas constants a and b are chosen according to the channel outputs.

For the weights connected between input and hidden layer:

$$w_{kl}^{(1)}(n+1) = w_{kl}^{(1)}(n) + \eta\delta_k^{(1)}(n)x_l^*(n) \tag{19}$$

where $\delta_k^{(1)}(n)$ is given by

$$\begin{aligned}\delta_k^{(1)}(n) = {} & \frac{\delta^{(2)}(n)}{net^{(2)}(n)}\{\varphi^{(1)'}(net_{k,R}^{(1)}(n))\,\mathrm{Re}(w_k^{(2)}(n)net^{(2)*}(n)) \\ & -\varphi^{(1)'}(net_{k,I}^{(1)}(n))\,\mathrm{Im}(w_k^{(2)}(n)net^{(2)*}(n))\}\end{aligned} \tag{20}$$

Here $u_k^*(n)$ and $x_l^*(n)$ denote the complex conjugate of kth and lth elements of $\mathbf{u}(n)$ and $\mathbf{x}(n)$ respectively, η is the learning rate parameter while $\varphi^{(1)'}(.)$ and $\varphi^{(2)'}(.)$ represent the derivatives of $\varphi^{(1)}(.)$ and $\varphi^{(2)}(.)$.

B. QAM Signals

The weights connected between hidden and output layer are updated by

$$w_k^{(2)}(n+1) = w_k^{(2)}(n) + \eta\delta^{(2)}(n)u_k^*(n)e^{-j\theta(n)} \tag{21}$$

where $\delta^{(2)}(n)$ is defined as follows.

$$\delta^{(2)}(n) = \{\varphi^{(2)}(n_R^{(2)}(n))\varphi^{(2)'}(n_R^{(2)}(n)) + j\varphi^{(2)}(n_I^{(2)}(n))\varphi^{(2)'}(n_I^{(2)}(n))\} .(R_2 - |y(n)|^2) \quad (22)$$

with

$$n_R^{(2)}(n) = net_R^{(2)}(n).\cos\theta(n) - net_I^{(2)}.\sin\theta(n) \quad (23)$$

and

$$n_I^{(2)}(n) = net_R^{(2)}(n).\sin\theta(n) + net_I^{(2)}.\cos\theta(n) \quad (24)$$

where $\theta(n)$ is the phase shift parameter of the node, and (21) is used along with (15).

The update rule for weights $\{w_{kl}^{(1)}\}$, connected between input and hidden layer, is given as,

$$w_{kl}^{(1)}(n+1) = w_{kl}^{(1)}(n) + \eta\delta_k^{(1)}(n)x_l^*(n) \quad (25)$$

where $\delta_k^{(1)}(n)$ is defined by

$$\delta_k^{(1)}(n) = \left[\varphi^{(1)'}(net_{k,R}^{(1)}(n).\mathrm{Re}\left[\delta^{(2)}(n).w_k^{(2)*}(n).e^{-j\theta(n)}\right] + j\varphi^{(1)'}(net_{k,I}^{(1)}(n)).\mathrm{Im}\left[\delta^{(2)}(n).w_k^{(2)*}(n).e^{-j\theta(n)}\right]\right] \quad (26)$$

The derivations of the phase correction algorithm given in (15) and the weight update equations (17) – (26) are given in the Appendix.

V. Simulation

It has been shown in [13] that a neural network equalizer gives better performance than a linear transversal filter for stationary channels, for M-ary QAM signals. In this section we present the simulation of the proposed models in the time-varying environment. As an example, a time-varying channel used for the simulation is shown in Fig 2.

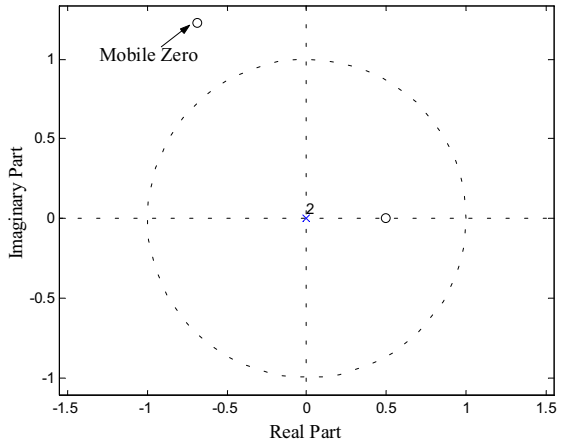

Figure 2: Zeros and poles of the nonstationary channel

This channel incorporates both, a sudden change and a gradual change in the environment. There is a fixed zero at $z_1 = 0.5$. After 3000 iterations another zero which is a mobile zero, appears as given below.

$$z_2(n) = 1.6\exp(j2\pi/3) + 0.2\exp(j\pi.(n-3000).10^{-4}) \quad (27)$$

The channel suddenly changes after $n = 3000$ and becomes a continuously varying medium.

Fig 3(a) and (b) show the 8-PSK, 16-QAM signal constellations, whereas the outputs of the channel and the equalizer for 8-PSK signal are shown in Fig 3(c) and (d), respectively, at 20 dB SNR. The structural details and the initializations for the blind equalizers used in the simulation are given in the Table 1.

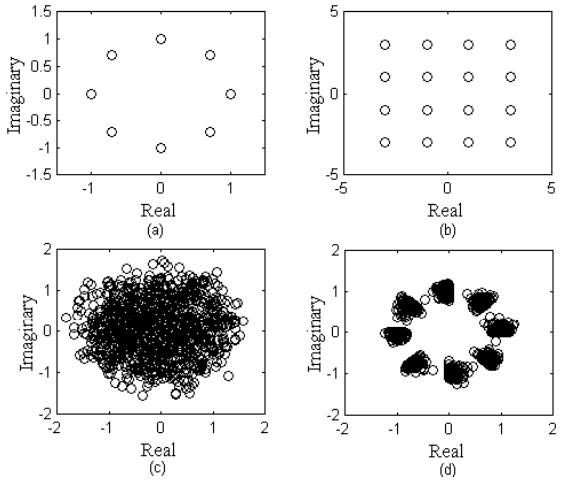

Figure 3. (a) 8-PSK signal constellation
(b) 16-QAM signal constellation
(c) Output of the channel for 8-PSK signal
(d) Output of the NN equalizer for 8-PSK signal

TABLE 1
STRUCTURAL DETAILS OF THE BLIND EQUALIZERS

Type of blind equalizer	No. of nodes in the input layer	No. of nodes in hidden layer	No. of taps	Initialization of equalizers
NN Equalizer for 8-PSK signal	15	9	-	$a = 2$, $b = 0.5$, c = 0.15, $\alpha = 4$, $\beta = 0.4$ $w_{59,R}^{(1)} = 1$ $w_{5,R}^{(2)} = 2.5$ other weights as small random numbers
Linear FIR Equalizer For 8-PSK signal	-	-	25	$w_{13} = 1$ other weights = 0
NN Equalizer for 16-QAM signal	15	9	-	$d = 0.15$ $w_{58,R}^{(1)} = 1.3$ $w_{5,R}^{(2)} = 1$.

The outputs of the equalizer of [13] and the proposed model for 16-QAM signal, are given in Fig 4 (a) and (b), respectively. The MSE plots obtained with and without phase correction in the output of the equalizer of [13] and the proposed model are shown in Fig 5(a) for 16-QAM signal. These plots are obtained by averaging 50 independent runs. The symbol error rate curves obtained from the proposed model (i.e. with phase correcting node) and the equalizer of [13] with external phase correction applied at the output of the equalizer, are shown in Fig 5(b). From these figures, it can be seen that the use of activation function of (14) gives lower MSE and SER than those obtained when the arbitrary phase shift is corrected at the output of the equalizer.

The MSE and SER plots for 8- PSK signal are shown in Fig 6 (a) and (b), respectively. The comparison with the linear blind equalizer shows that the neural network equalizer with the proposed activation function gives lower MSE and SER.

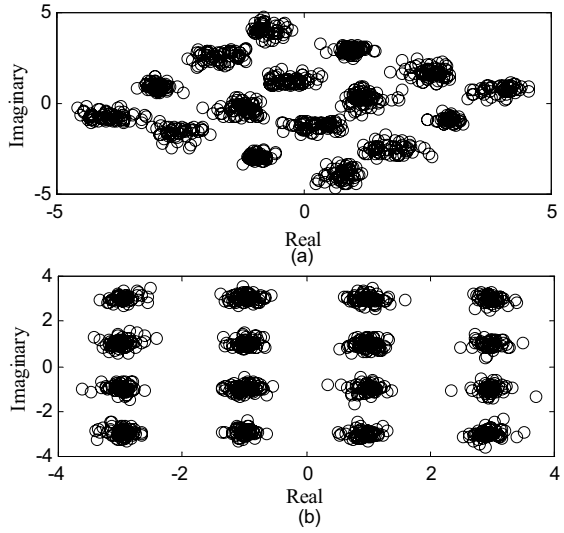

Figure 4: Output of NN blind equalizers for 16-QAM signal
(a) Equalizer of [13] (b) The proposed model

VI. CONCLUSION

We have presented two models of complex-valued feedforward neural networks for blind equalization in time-varying environment. Training algorithms are based on the CMA cost function. Activation functions are chosen in accordance with the signal constellations. In case of M-ary QAM signals, the arbitrary phase shift, inherent in a blind equalization scheme, is corrected by the output node of the neural network. Therefore, the proposed scheme does not require any external phase correction. The algorithm for the phase correction at the output nodes is based on the modified CMA cost function. The proposed models also give lower MSE and SER under time-varying environment. These benefits are, however, obtained at the cost of increased complexity.

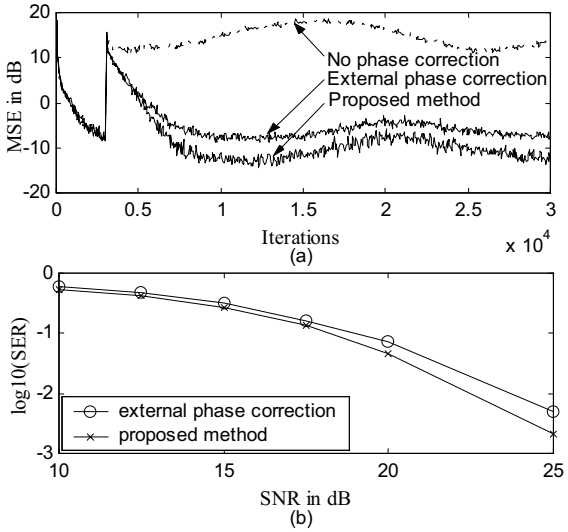

Figure 5: Performance of NN blind equalizer under nonstationary channel for 16-QAM signal
(a) MSE convergence
(b) Plot of symbol error rate

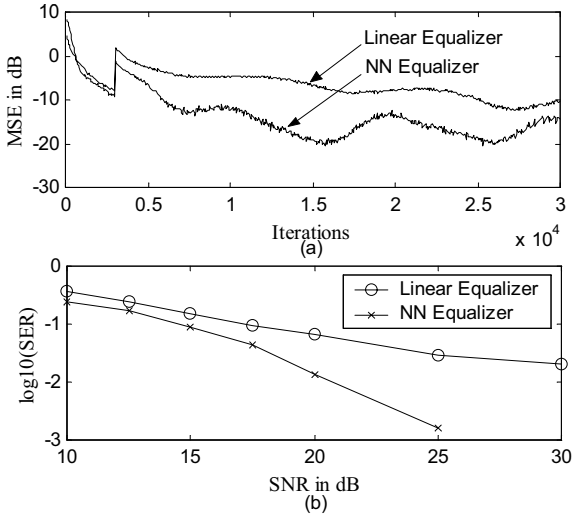

Figure 6: Performance of NN blind equalizer under nonstationary channel for 8-PSK signal
(a) MSE convergence (b) Plot of symbol error rate

APPENDIX

A. Derivations of Update Rules for QAM Signal

The CMA cost function is expressed as

$$J(n) = \frac{1}{4} E\left[\left(|y(n)|^2 - R_2\right)^2\right] \tag{28}$$

where $R_2 = E[|s(n)|^4] / E[|s(n)|^2]$, $E[.]$ is the expectation operator and $s(n)$ denotes the data sequence.

Using the gradient descent technique, the weights of the neural network are updated as

$$w_k^{(2)}(n+1) = w_k^{(2)}(n) - \eta \nabla_{w_k^{(2)}} J(n) \tag{29}$$

and

$$w_{kl}^{(1)}(n+1) = w_{kl}^{(1)}(n) - \eta \nabla_{w_{kl}^{(1)}} J(n) \tag{30}$$

where η is the learning rate parameter and the terms $\nabla_{w_k^{(2)}} J(n)$ and $\nabla_{w_{kl}^{(1)}} J(n)$ represent the gradients of the cost function $J(n)$ defined by (28), with respect to the weights $w_k^{(2)}$ and $w_{kl}^{(1)}$ respectively. These gradients are given as

$$\nabla_{w_k^{(2)}} J(n) = \left(|y(n)|^2 - R_2\right).|y(n)|.\left(\frac{\partial |y(n)|}{\partial w_{k,R}^{(2)}(n)} + j\frac{\partial |y(n)|}{\partial w_{k,I}^{(2)}(n)}\right) \tag{31}$$

and

$$\nabla_{w_{kl}^{(1)}} J(n) = \left(|y(n)|^2 - R_2\right).|y(n)|.\left(\frac{\partial |y(n)|}{\partial w_{kl,R}^{(1)}(n)} + j\frac{\partial |y(n)|}{\partial w_{kl,I}^{(1)}(n)}\right). \tag{32}$$

Now, in order to compute these gradients, let us define as given by (23) and (24):

$$n_R^{(2)}(n) = net_R^{(2)}(n).\cos\theta(n) - net_I^{(2)}.\sin\theta(n)$$

and

$$n_I^{(2)}(n) = net_R^{(2)}(n).\sin\theta(n) + net_I^{(2)}.\cos\theta(n)$$

where $\theta(n)$ is the phase shift parameter of the node. The partial derivative terms required in (31), are expressed as

$$\frac{\partial |y(n)|}{\partial w_{k,R}^{(2)}(n)} = \frac{1}{|y(n)|}\Big[\varphi^{(2)}(n_R^{(2)}(n)).\varphi^{(2)'}(n_R^{(2)}(n)).$$
$$(u_{k,R}(n).\cos\theta - u_{k,I}(n).\sin\theta(n)) + \varphi^{(2)}(n_I^{(2)}(n)).\varphi^{(2)'}(n_I^{(2)}(n)).$$
$$(u_{k,I}(n).\cos\theta + u_{k,R}(n).\sin\theta(n))\Big] \tag{33}$$

and

$$\frac{\partial |y(n)|}{\partial w_{k,I}^{(2)}(n)} = -\frac{1}{|y(n)|}\Big[\varphi^{(2)}(n_R^{(2)}(n)).\varphi^{(2)'}(n_R^{(2)}(n)).$$
$$(u_{k,I}(n).\cos\theta + u_{k,R}(n).\sin\theta(n)) - \varphi^{(2)}(n_I^{(2)}(n)).\varphi^{(2)'}(n_I^{(2)}(n)).$$
$$(u_{k,R}(n).\cos\theta - u_{k,I}(n).\sin\theta(n))\Big]. \tag{34}$$

On substituting (33) and (34) in (31), we obtain the expression of gradient $\nabla_{w_k^{(2)}} J(n)$ after some simplification as

$$\nabla_{w_k^{(2)}} J(n) = \left(|y(n)|^2 - R_2\right).\Big[\varphi^{(2)}(n_R^{(2)}(n)).\varphi^{(2)'}(n_R^{(2)}(n)) + j\varphi^{(2)}(n_I^{(2)}(n)).\varphi^{(2)'}(n_I^{(2)}(n))\Big].u_k^*(n).e^{-j\theta(n)}. \tag{35}$$

By substituting (35) in (29), the update equation (21) follows. To obtain the update equation for the hidden layer weights, $\{w_{kl}^{(1)}\}$, the gradient of (32) is used. The partial derivative terms of (32) can be expressed as

$$\frac{\partial |y(n)|}{\partial w_{kl,R}^{(1)}(n)} = \frac{1}{|y(n)|}\Big[\varphi^{(2)}(n_R^{(2)}(n)).\varphi^{(2)'}(n_R^{(2)}(n)).\frac{\partial n_R^{(2)}(n)}{\partial w_{kl,R}^{(1)}(n)} + \varphi^{(2)}(n_I^{(2)}(n)).\varphi^{(2)'}(n_I^{(2)}(n)).\frac{\partial n_I^{(2)}(n)}{\partial w_{kl,R}^{(1)}(n)}\Big] \tag{36}$$

and

$$\frac{\partial |y(n)|}{\partial w_{kl,I}^{(1)}(n)} = \frac{1}{|y(n)|}\Big[\varphi^{(2)}(n_R^{(2)}(n)).\varphi^{(2)'}(n_R^{(2)}(n)).\frac{\partial n_R^{(2)}(n)}{\partial w_{kl,I}^{(1)}(n)} + \varphi^{(2)}(n_I^{(2)}(n)).\varphi^{(2)'}(n_I^{(2)}(n)).\frac{\partial n_I^{(2)}(n)}{\partial w_{kl,I}^{(1)}(n)}\Big] \tag{37}$$

where

$$\frac{\partial n_R^{(2)}(n)}{\partial w_{kl,R}^{(1)}(n)} = \cos\theta(n).\Big(w_{k,R}^{(2)}(n).\varphi^{(1)'}(net_{k,R}^{(1)}(n)).x_{l,R}(n) - w_{k,I}^{(2)}(n).\varphi^{(1)'}(net_{k,I}^{(1)}(n)).x_{l,I}(n)\Big) - \sin\theta(n).\Big(w_{k,R}^{(2)}(n).\varphi^{(1)'}(net_{k,I}^{(1)}(n)).x_{l,I}(n) + w_{k,I}^{(2)}(n).\varphi^{(1)'}(net_{k,R}^{(1)}(n)).x_{l,R}(n)\Big)$$
$$\frac{\partial n_I^{(2)}(n)}{\partial w_{kl,R}^{(1)}(n)} = \cos\theta(n).\Big(w_{k,R}^{(2)}(n).\varphi^{(1)'}(net_{k,I}^{(1)}(n)).x_{l,I}(n) + w_{k,I}^{(2)}(n).\varphi^{(1)'}(net_{k,R}^{(1)}(n)).x_{l,R}(n)\Big) + \sin\theta(n).\Big(w_{k,R}^{(2)}(n).\varphi^{(1)'}(net_{k,R}^{(1)}(n)).x_{l,R}(n) - w_{k,I}^{(2)}(n).\varphi^{(1)'}(net_{k,I}^{(1)}(n)).x_{l,I}(n)\Big) \tag{38}$$

$$\frac{\partial n_R^{(2)}(n)}{\partial w_{kl,I}^{(1)}(n)} = -\cos\theta(n).\Big(w_{k,R}^{(2)}(n).\varphi^{(1)'}(net_{k,R}^{(1)}(n)).x_{l,I}(n) + w_{k,I}^{(2)}(n).\varphi^{(1)'}(net_{k,I}^{(1)}(n)).x_{l,R}(n)\Big) - \sin\theta(n).\Big(w_{k,R}^{(2)}(n).\varphi^{(1)'}(net_{k,I}^{(1)}(n)).x_{l,R}(n) - w_{k,I}^{(2)}(n).\varphi^{(1)'}(net_{k,R}^{(1)}(n).x_{l,I}(n)\Big)$$
$$\frac{\partial n_I^{(2)}(n)}{\partial w_{kl,I}^{(1)}(n)} = \cos\theta(n).\Big(w_{k,R}^{(2)}(n).\varphi^{(1)'}(net_{k,I}^{(1)}(n)).x_{l,R}(n) - w_{k,I}^{(2)}(n).\varphi^{(1)'}(net_{k,R}^{(1)}(n)).x_{l,I}(n)\Big) - \sin\theta(n).\Big(w_{k,R}^{(2)}(n).\varphi^{(1)'}(net_{k,R}^{(1)}(n)).x_{l,I}(n) + w_{k,I}^{(2)}(n).\varphi^{(1)'}(net_{k,I}^{(1)}(n)).x_{l,R}(n)\Big) \tag{39}$$

$$\frac{\partial n_R^{(2)}(n)}{\partial w_{kl,I}^{(1)}(n)} = -\cos\theta(n).\Big(w_{k,R}^{(2)}(n).\varphi^{(1)'}(net_{k,R}^{(1)}(n)).x_{l,I}(n) + w_{k,I}^{(2)}(n).\varphi^{(1)'}(net_{k,I}^{(1)}(n)).x_{l,R}(n)\Big) - \sin\theta(n).\Big(w_{k,R}^{(2)}(n).\varphi^{(1)'}(net_{k,I}^{(1)}(n)).x_{l,R}(n) - w_{k,I}^{(2)}(n).\varphi^{(1)'}(net_{k,R}^{(1)}(n).x_{l,I}(n)\Big)$$
$$\frac{\partial n_I^{(2)}(n)}{\partial w_{kl,I}^{(1)}(n)} = \cos\theta(n).\Big(w_{k,R}^{(2)}(n).\varphi^{(1)'}(net_{k,I}^{(1)}(n)).x_{l,R}(n) - w_{k,I}^{(2)}(n).\varphi^{(1)'}(net_{k,R}^{(1)}(n)).x_{l,I}(n)\Big) - \sin\theta(n).\Big(w_{k,R}^{(2)}(n).\varphi^{(1)'}(net_{k,R}^{(1)}(n)).x_{l,I}(n) + w_{k,I}^{(2)}(n).\varphi^{(1)'}(net_{k,I}^{(1)}(n)).x_{l,R}(n)\Big) \tag{41}$$

Now by substituting the partial derivative terms from (38) and (39) in (36), and from (40) and (41) in (37) and then making use of (32), we get the expression for the gradient after some simplification as

$$\nabla_{w_{kl}^{(1)}} J(n) = -\Big[\Big(\varphi^{(1)'}(net_{k,R}^{(1)}(n)).\mathrm{Re}\Big[\delta^{(2)}(n).w_k^{(2)*}(n)\Big]. + j\varphi^{(1)'}(net_{k,R}^{(1)}(n)).\mathrm{Im}\Big[\delta^{(2)}(n).w_k^{(2)*}(n)\Big]\Big).x_l^*(n).\cos\theta(n) + \Big(\varphi^{(1)'}(net_{k,R}^{(1)}(n)).\mathrm{Im}\Big[\delta^{(2)}(n).w_k^{(2)*}(n)\Big] - j\varphi^{(1)'}(net_{k,I}^{(1)}(n)).\mathrm{Re}\Big[\delta^{(2)}(n).w_k^{(2)*}(n)\Big]\Big).x_l^*(n).\sin\theta(n)\Big].$$

Further simplification leads to

$$\nabla_{w_{kl}^{(1)}} J(n) = \Big[\varphi^{(1)'}(net_{k,R}^{(1)}(n).\mathrm{Re}\Big[\delta^{(2)}(n).w_k^{(2)*}(n).e^{-j\theta(n)}\Big] + j\varphi^{(1)'}(net_{k,I}^{(1)}(n)).\mathrm{Im}\Big[\delta^{(2)}(n).w_k^{(2)*}(n).e^{-j\theta(n)}\Big]\Big].x_l^*(n) = \delta_k^{(1)}(n).x_l^*(n) \quad (42)$$

where

$$\delta_k^{(1)}(n) = \Big[\varphi^{(1)'}(net_{k,R}^{(1)}(n).\mathrm{Re}\Big[\delta^{(2)}(n).w_k^{(2)*}(n).e^{-j\theta(n)}\Big] + j\varphi^{(1)'}(net_{k,I}^{(1)}(n)).\mathrm{Im}\Big[\delta^{(2)}(n).w_k^{(2)*}(n).e^{-j\theta(n)}\Big]\Big] \quad (43)$$

Substitution of (42) in (30) gives update equation for the weights $\{w_{k,l}^{(1)}\}$ given in (25).

B. Update equation for phase angle

To obtain the update equation for parameter θ, either the real or the imaginary version of CMA cost function can be used i.e.

$$J = \frac{1}{4}E\Big[(z_I^2(n) - \bar{R}_2)^2\Big]$$

or

$$J = \frac{1}{4}E\Big[(z_R^2(n) - \bar{R}_2)^2\Big] \quad (44)$$

where

$$z(n) = y(n).e^{j\theta(n)} = z_R(n) + jz_I(n) \quad (45)$$

represents the output of the equalizer after the phase rotation, $z_R(n)$ and $z_I(n)$ are real and imaginary parts of $z(n)$ at time n, respectively. The parameter $\bar{R}_2$ is given by

$$\bar{R}_2 = \frac{E\Big[z_R^4\Big]}{E\Big[z_R^2\Big]}. \quad (46)$$

The update equation for θ is expressed by using the gradient descent approach, as

$$\theta(n+1) = \theta(n) - \mu.\nabla_\theta J(n) \quad (47)$$

where μ is a constant that governs the convergence rate. The gradient of the cost function of (44) is expressed as

$$\nabla_\theta J(n) = (z_R^2(n) - \bar{R}_2).z_R(n).\frac{\partial z_R(n)}{\partial\theta(n)}. \quad (48)$$

Now the partial derivative term in (48) is obtained from (45), as

$$\frac{\partial z_R(n)}{\partial\theta(n)} = \frac{\partial}{\partial\theta(n)}\big[y_R(n).\cos(\theta(n)) - y_I(n).\sin(\theta(n))\big] = -z_I(n)\ . \quad (49)$$

By substituting (49) in (48), we get

$$\nabla_\theta J(n) = -(z_R^2(n) - \bar{R}_2).z_R(n).z_I(n)\,. \quad (50)$$

Finally, by the substitution of $\nabla_\theta J(n)$ from (50) in (47) and replacing $z(n)$ by $net^{(2)}(n)$, (15) is obtained. Then (46) gives (16).

C. Derivations of update rules for PSK signal

Since the activation function of the output layer neuron for M-ary PSK signal is defined in terms of modulus and angle of the activation sum, the gradient of the CMA cost function with respect to the output layer weight $w_k^{(2)}(n)$ is expressed as

$$\nabla_{w_k^{(2)}} J(n) = (|y(n)|^2 - R_2).|y(n)|.\left(ab - \frac{b}{a}.|y(n)|^2\right).\frac{\partial net^{(2)}(n)}{\partial w_k^{(2)}(n)}. \quad (51)$$

To obtain an expression for the partial derivative of (51), we use the relationship

$$\left|net^{(2)}(n)\right|^2 = \left(net_R^{(2)}(n)\right)^2 + \left(net_I^{(2)}(n)\right)^2. \quad (52)$$

On differentiating (52) with respect to $w_k^{(2)}$ we get

$$\frac{\partial\left|net^{(2)}(n)\right|}{\partial w_k^{(2)}(n)} = \frac{1}{\left|net^{(2)}(n)\right|}.\left[net_R^{(2)}(n).\frac{\partial net_R^{(2)}(n)}{\partial w_k^{(2)}(n)} + net_I^{(2)}(n)\frac{\partial net_I^{(2)}(n)}{\partial w_k^{(2)}(n)}\right] = \frac{\varphi^{(1)*}(net_k^{(1)}(n)).net^{(2)}(n)}{\left|net^{(2)}(n)\right|}. \quad (53)$$

On substituting (53) in (51), the expression for the gradient becomes

$$\nabla_{w_k^{(2)}} J(n) = \delta^{(2)}(n).u_k^*(n) \quad (54)$$

where

$$\delta^{(2)}(n) = \left[\left(|y(n)|^2 - R_2\right).|y(n)|.\left(ab - \frac{b}{a}|y(n)|^2\right)\right].\frac{net^{(2)}(n)}{\left|net^{(2)}(n)\right|}. \quad (55)$$

Substitution of (54) in (29) along with (55) gives the update equation (17) with (18) for M-ary PSK signal.

In order to obtain the update equation for the weights $\{w_{kl}^{(1)}\}$, we need the following gradient.

$$\nabla_{w_{kl}^{(1)}} J(n) = \left(|y(n)|^2 - R_2\right).|y(n)|.\left(ab - \frac{b}{a}|y(n)|^2\right).\frac{\partial\left|net^{(2)}(n)\right|}{\partial w_{kl}^{(1)}(n)} \quad (56)$$

The partial derivative terms in (56) can be obtained by using (52) as described below.

$$\frac{\partial\left|net^{(2)}(n)\right|}{\partial w_{kl}^{(1)}(n)} = \frac{1}{\left|net^{(2)}(n)\right|}\left[net_R^{(2)}(n).\frac{\partial net_R^{(2)}(n)}{\partial w_{kl}^{(1)}(n)} + net_I^{(2)}(n)\frac{\partial net_I^{(2)}(n)}{\partial w_{kl}^{(1)}(n)}\right] \quad (57)$$

where

$$\frac{\partial net_R^{(2)}(n)}{\partial w_{kl}^{(1)}(n)} = w_{k,R}^{(2)}(n).\varphi^{(1)'}(net_{k,R}^{(1)}(n)).(x_{l,R}(n) - jx_{l,I}(n)) - w_{k,I}^{(2)}(n).\varphi^{(1)'}(net_{k,I}^{(1)}(n)).(x_{l,I}(n) + jx_{l,R}(n)) \quad (58)$$

and

$$\frac{\partial net_I^{(2)}(n)}{\partial w_{kl}^{(1)}(n)} = w_{k,R}^{(2)}(n).\varphi^{(1)'}(net_{k,I}^{(1)}(n)).(x_{l,I}(n) + jx_{l,R}(n)) + w_{k,I}^{(2)}(n).\varphi^{(1)'}(net_{k,R}^{(1)}(n)).(x_{l,R}(n) - jx_{l,I}(n)) \quad (59)$$

Now the substitution of (58) and (59) in (57) and some simplification lead to

$$\frac{\partial\left|net^{(2)}(n)\right|}{\partial w_{kl}^{(1)}(n)} = \frac{x_l^*(n)}{\left|net^{(2)}(n)\right|}\left(\varphi^{(1)'}(net_{k,R}^{(1)}(n)).\mathrm{Re}[w_k^{(2)}(n).net^{(2)*}(n)] - j\varphi^{(1)'}(net_{k,I}^{(1)}(n)).\mathrm{Im}[w_k^{(2)}(n).net^{(2)*}(n)]\right). \quad (60)$$

Finally, by substituting (60) in (56), we get

$$\nabla_{w_{kl}^{(1)}}J(n) = \delta^{(2)}(n).x_l^*(n).\left(\varphi^{(1)'}(net_{k,R}^{(1)}(n)).\mathrm{Re}[w_k^{(2)}(n).net^{(2)*}(n)] - j\varphi^{(1)'}(net_{k,I}^{(1)}(n)).\mathrm{Im}[w_k^{(2)}(n).net^{(2)*}(n)]\right) = \delta_k^{(1)}(n).x_l^*(n) \quad (61)$$

where

$$\delta_k^{(1)}(n) = \frac{\delta^{(2)}(n)}{net^{(2)}(n)}.\left(\varphi^{(1)'}(net_{k,R}^{(1)}(n)).\mathrm{Re}[w_k^{(2)}(n).net^{(2)*}(n)] - j\varphi^{(1)'}(net_{k,I}^{(1)}(n)).\mathrm{Im}[w_k^{(2)}(n).net^{(2)*}(n)]\right).$$

Using (61) and (30), we get the update rule of (19) for M-ary PSK signal.

REFERENCES

[1] J. G. Proakis, *Digital Communications.* Third Edition, Singapore, McGraw Hill, 1995.
[2] S. Haykin (Editor). *Blind Deconvolution*. Englewood Cliffs, New Jersey: Prentice-Hall, 1994.
[3] S. I. Amari and A. Cichocki, "Adaptive blind signal processing – Neural network approaches," *Proceedings of IEEE*, vol. 86, no. 10, pp. 2026 – 2048, Oct. 1998.
[4] T. W. S. Chow and Y. Fang, "Neural blind deconvolution of MIMO noisy channels," *IEEE Trans. Circuits and Systems-I*, vol. 48, no. 1, pp. 116 – 120, Jan. 2001.
[5] H. Lin and M. Amin, "A dual mode technique for improved blind equalization for QAM signals," *IEEE Signal Processing Letters*, vol. 10, no. 2, pp. 29 – 31, Feb. 2003.
[6] N. Thirion Moreau and E. Moreau, "Generalized criterion for blind multivariate signal equalization," *IEEE Signal Processing Letters*, vol. 9, no. 2, pp. 72 – 74, Feb. 2002.
[7] J. B. Destro Filho, G. Favier and J. M. Travassos Romano, "Neural networks for blind equalization," *Proceedings of IEEE Globcom*, 1996, pp. 196 – 200.
[8] Y. Fang and T. W. S. Chow, "Blind equalization of a noisy channel by linear neural network," *IEEE Trans. Neural Networks*, vol. 10, no. 4, pp. 925 – 929, July 1999.
[9] R. Pandey, "Blind equalization and signal separation using neural networks," Ph. D. thesis, I.I.T. Roorkee, India, 2001.
[10] S. Haykin, *Adaptive Filter Theory*, Third Edition. Upper Saddel River, New Jersey: Prentice Hall, 1996.
[11] M. Ghosh, "Blind decision feedback equalization for terrestrial television receivers," *Proceedings of IEEE*, vol. 86, no. 10, pp. 2070 – 2081, Oct. 1998.
[12] J. Labat, O. Macchi and C. Laot, "Adaptive decision feedback equalization: Can you skip the training period?," *IEEE Trans. Communications*, vol. 46, no. 7, pp. 921 – 930, July 1998.
[13] C. You and D. Hong, "Nonlinear blind equalization schemes using complex valued multilayer feedforward neural networks," *IEEE Trans. Neural Networks*, vol. 9, no. 6, pp. 1442 – 1455, Nov. 1998.
[14] A. Cichocki, R. Unbehauen, *Neural Networks for Optimization and Signal Processing*, Chichester: John Wiley & Sons, 1994.
[15] S. Haykin, *Neural Networks: A Comprehensive Foundation*, Upper Saddle River, New Jersey: Prentice –Hall, 1994.
[16] F. L. Luo, and R. Unbehauen, *Applied Neural Networks for Signal Processing*, Cambridge: University Press, 1997.
[17] S. Haykin, Neural networks expand SP's horizons, *IEEE Signal Processing Magazine*, issue 3, pp. 24 – 49, 1996.
[18] G. Kechriotis, E. Zervas and E. S. Manolakos, "Using recurrent neural networks for adaptive communication channel equalization," *IEEE Trans. Neural Networks*, vol. 5, no. 2, pp. 267 – 278, March 1994.
[19] R. Parisi, C. D. Di Elio, G. Orlandi and B. D. Rao, "Fast adaptive digital equalization by recurrent neural network," *IEEE Trans. Signal Processing*, vol. 45, no. 11, pp. 2731 – 2739, Nov. 1997.
[20] S. Ong, C. You, S. Choi and D. Hong, "A decision feedback recurrent neural equalizer as an infinite impulse response filter," *IEEE Trans. Signal Processing*, vol. 45, no. 11, pp. 2851 – 2858, Nov. 1997.
[21] N. Benvenuto and F.Piazza, "On the complex backpropagation algorithm," *IEEE Trans. Signal Processing*, vol. 40, pp. 967 – 969, Apr. 1992.
[22] C. R. Johnson Jr. *et al*., "Blind equalization using the constant modulus criterion: A review," *Proceedings of IEEE*, vol. 86, no. 10, pp. 1927 – 1950, Oct. 1998.
[23] T. J. Endres, B. D. O. Anderson, C. R. Johnson Jr. and M. Green, "Robustness to fractionally-spaced equalizer length using the Constant Modulus Criterion," *IEEE Trans. Signal Processing*, vol. 47, no. 2, pp. 544 – 548, Feb. 1999.
[24] P. Schniter and C. R. Johnson Jr., "Dithered signed error CMA: Robust, computationally efficient blind adaptive equalization," *IEEE Trans. Signal Processing*, vol. 47, no. 6, pp. 1592 – 1603, June 1999.

Rajoo Pandey received the B. E degree in 1989 from Govt. Engineering College, Jabalpur, India, M. Tech. degree in 1991 from R. E. C. Kurukshetra, India and Ph. D. degree in 2002 from Indian Institute of Technology, Roorkee, India in Electronics and Communication Engineering. He joined as a lecturer in the Department of Electronics, Communication & Computer Engineering, R. E. C. Kurukshetra in 1991 and is currently working as Assistant Professor in the Department of Electronics and Communication Engineering at N.I.T. Kurukshetra. His research interests include signal processing, communication systems and neural networks.