Nonlinear Complex Channel Equalization
Using A Radial Basis Function Neural Network

Miclau Nicolae, Corina Botoca, Georgeta Budura
University Politehnica of Timișoara
corinab@etc.utt.ro

Abstract: The problem of equalization for complex signals is presented. It is proposed a competitive method for the estimation of the centers of a complex radial basis function neural network. Simulation results are presented from the point of view of mean square error and signals space partition. Concluding remarks and further developments are discussed.

Key Words: channel equalization, complex radial basis neural network

I. INTRODUCTION

In modern high-speed communications networks, the presence of symbol interference (ISI) is a major impediment of data transmission. Nonlinear active or passive devices and the transmission channels themselves introduce nonlinear distortions that affect the signals. Especially the signals with a variable envelope modulation, as for example the quadrature amplitude modulation (QAM) signals, more efficient in transmission from the spectral point of view, are affected, in phase and in amplitude. To compensate the distortions of QAM and phase shift keying (PSK) signals equalizers for complex signals are necessary. These equalizers are extensions of the real equalizers.

The problem of equalization may be treated as a problem of signals classification, so neural networks (NN) are quite promising candidates because they can produce arbitrarily complex decision regions. Studies performed during the last decade have established the superiority of neural equalizers comparative to the traditional equalizers, in conditions of high nonlinear distortions and rapidly varying signals.

Several different neural equalizers architectures have been developed, mostly combinations between a conventional linear transversal filter (LTE) and a neural network. The LTE eliminates the linear distortions, such as ISI, so the NN can be focused on compensating the nonlinearities. There have been studied the following structures: a LTE and a multilayer perceptron (MLP) [1], [7], a LTE and a radial basis function network (RBF) [2], [3], [4],[8] a LTE and a recurrent neural network [10],[12], a functional link equalizer [6] [9], and cellular neural network equalizer [11]. There have been introduced many different nonlinear devices models and channels models, so a unitary comparison between all known equalizers is difficult to be done. Since MLP networks are sometimes plagued by long training times and may be trapped at bad local minima, RBF networks often provide a faster and more robust solution to the equalization problem. In addition, the RBF neural network has a structure similar to the optimal Bayesian symbol decision equalizer. Note that the Bayesian equalizer does not necessarily yield a good minimum mean square error (MSE) performance but does provide the minimum average bit error rate (BER) achievable for symbol decision and indirect-modeling equalizer structures. Therefore, the RBF is an ideal processing structure to implement the optimal Bayesian equalizer. The RBF performances are better than the LTE and MLP equalizers [1], [4]. In conclusion the RBF network is an attractive alternative to the MLP neural network mainly due to its simple structure and more efficient learning.

Several learning algorithms have been proposed to update the RBF parameters. However, the most popular algorithm consists of an unsupervised learning rule for the centers of hidden neurons and a supervised learning rule for the weights of the output neurons. The centers are generally updated using the k-means clustering algorithm [7] which consists of computing the squared distance between the input vector and the centers, choosing a minimum squared distance, and moving the corresponding center closer to the input vector. The k mean algorithm has some potential problems: classification depend on the initials values of the centers of RBF, on the type of chosen distance, and on the number of classes. If a center is inappropriate chosen it may never be updated, so it may never represent a class.

In [8] is proposed a sequential learning algorithm referred as complex minimal resource allocation network (CMRAN). The studies proved that the equalizer performance is superior to the functional link equalizer of Patra and all [9] and the stochastic gradient RBF equalizer [2] of Cha and Kassam.

In this paper we propose a new competitive method to update the RBF centers, which recompenses the winning neuron and penalizes the second winner, named rival. The algorithm is quite simple and the performances are comparative to all the others reported equalizers. The classes a automatically generated at the output of the network. The RBF network has complex centers and connection weights, but the nonlinearity of its hidden nodes is a real-valued function. The RBF competitive equalizer is able to approximate an arbitrary nonlinear function in complex multi-dimensional space with a reduced calculus complexity comparative with other algorithms.
II. THE EQUALIZATION PROBLEM

The equalization problem is traditionally viewed as an inverse filter problem. Equalizers are designed to track the time-varying channel distortions by adjusting their coefficients and maintaining a prescribed signal to noise ratio (SNR). Tradeoffs between noise enhancement and channel inversion generally render these techniques suboptimal. An alternative viewpoint is to consider the equalization problem as a pattern classification problem.

The objective of equalization becomes the separation of the received symbols in the output signal space, whose optimal decision region boundaries are generally highly nonlinear. Since neural networks are well known for their ability of performing classification tasks by forming complex nonlinear decision boundaries, neural equalizers have been recently receiving considerable attention. Neural equalizers have shown the potential for significant performance improvements especially in severely nonlinear distorted and rapidly varying signals.

Fig. 1 represents a model of a communication system. If the signal \( x \) is a 4 QAM, the input constellation is given by:

\[
x(n) = x_R + jx_I = \begin{cases} 
  x^{(1)} = 1 + j \\
  x^{(2)} = -1 + j \\
  x^{(3)} = 1 - j \\
  x^{(4)} = -1 - j 
\end{cases} \quad (1)
\]

The input symbols sequence \( x(n) \) is passed through the nonlinear communication channel model and produces \( y(n) \) at its output sequence. The channel output signal is affected by an additive noise \( w(n) \), usually white Gaussian, and produces a corrupted signal \( o(n) \).

The problem of equalization is to determine an estimation of the input signal \( x(n) \) using the received signal \( o(n) \) and the desired delayed signal \( x(n-d) \). From the NN point of view, the equalizer has to classify the received signal in one of the four possible classes \( P_{m,d} \), according to the input signals:

\[
P_{m,d} = \bigcup_{1 \leq l \leq 4} P_{m,d}(l) \quad (2)
\]

or:

\[
P_{m,d}(l) = \{ y(n) | x(n-d) = x^{(l)} \}, \quad 1 \leq l \leq 4
\]

III. THE COMMUNICATION CHANNEL MODEL

Fig. 2 represents a model of the communication channel that introduces linear and nonlinear distortions. The linear complex part of the channel is often modeled by a transversal filter FIR whose output is given by:

\[
y(n) = \sum_{i=0}^{k-1} a_i x(n-i) \quad (4)
\]

where \( a_i \) are the filter coefficients and \( k \) is the order of the filter.

One of models suggested in [2] generates the output signal according to:

\[
y = (0.34 - 0.27j)x(n) + (0.87 + 0.43j)x(n-1) + (0.34 - 0.21j)x(n-2) \quad (5)
\]

The order of this filter is \( k=3 \).
The nonlinear part of the channel is a very strong one and produces at the output:
\[ y(n) = y(n) + 0.1[y(n)]^2 + 0.05[y(n)]^3 \]  \hspace{1cm} (6)

This signal is added with the Gaussian noise \( w(n) \), with a null mean and a dispersion of \( \sigma^2 \), and subsequently passed through the neural equalizer:
\[ o(n) = y(n) + w(n) \]  \hspace{1cm} (7)

IV. THE COMPLEX RADIAL BASIS FUNCTION EQUALIZER

An equalizer may be implemented with a LTE followed by a neural network, as in Fig.3.

If \( m \) is the FIR filter order, the received signal \( o = [o(n) o(n-1) ... o(n-m+1)] \) is the input for the RBF network. The number of possible states of the received QAM signal is \( n_o = 2^{4m-1} \).

The complex RBF is a straightforward extension from the real counterpart [8], obtained by replacing the relevant parameters with complex values. As depicted in Fig. 3, the RBF network has two layers—the hidden layer and the output layer.

The hidden layer is composed of an array of computing neurons, each having a parameter \( c_i \) vector called center. Each neuron computes a distance between its center and the network input vector. This distance may be of different types and it is subsequently divided by a parameter \( \rho_i \), called width, which is the spread of the corresponding center. The result is passed through a real, nonlinear activation function, \( \phi_i(c_i, \rho_i) \)
\[ \phi_i = \|o-c_i\|^\rho_i(o-c_i), \rho_i \text{, } 1 \leq i \leq n_h \]  \hspace{1cm} (8)

where \( o \) is the complex input vector of \( n_h \) dimension, \( c_i \) is the centers vector of the radial basis functions, which is also a complex vector of \( n_h \) dimension, \( \rho_i \) is the center spread parameter, \( n_h \) is the number of computing nodes.

The operator \( * \) is \( ((*)^T)^* \), where \( (*)^T \) is the transposition operator and \( (*)^* \) is the complex conjugation operator.

The nonlinear output function is usually the Gaussian function:
\[ \phi(z^2, \rho) = e^{-\frac{z^2}{\rho}} \]  \hspace{1cm} (9)
The number of hidden neurons $n_h$ is given by the number of possible states of the channel output $n_e$. A number $n_h$ greater than $n_e$ generates inutile computing. A number $n_h$ smaller than $n_e$ may degrade the performances of the network.

Similarity with the Bayesian equalizer impose that the spread parameter $\rho=2\sigma^2$ where $\sigma^2$ is the noise dispersion given by relation:

$$\sigma^2 = E\|o(n) - c_i\|^2$$  \hspace{1cm} (10)

where $E$ is the mean, the second order momentum.

The output layer of the network consists of eight neurons (two neurons for each class, one for the real part and the other for the imaginary part of each class) with a linear function:

$$f_{RBF}(o) = \sum_{i=1}^{n_h} \sum_{j=1}^{n_e} \phi_j w_{ij}$$  \hspace{1cm} (11)

where $w_{ij}$ are the complex weights. According to the relation (9), $f_{RBF}$ becomes:

$$f_{RBF}(o) = \sum_{i=1}^{n_h} w_{ij} e^{-\frac{(o-c_j)^2}{\rho_i}}$$  \hspace{1cm} (12)

IV.1. COMPETITIVE LEARNING ALGORITHM PENALIZING THE RIVAL

The competitive standard algorithm calculates the distance between the input vector and the RBF centers vector. The distance may be of different types, usually the Euclidian norm is used:

$$\|o(n) - c_i(n)\| = \sqrt{\|o(n) - c_i(n)\|^2 + \ldots + \|o(n-m+l) - c_i(n-m+l)\|^2}$$  \hspace{1cm} (13)

The neuron $j$ with a minimum distance is declared winner:

$$j = \text{argmin}_{i=1}^{n_h} \|o(n) - c_i(n)\|, \quad i = 1, n_h$$  \hspace{1cm} (14)

The winning neuron center is moved with a fraction $\eta$ towards the input.

The competitive algorithm penalizing the rival [5] determines not only the winning neuron but also the second winning neuron $r$:

$$r = \text{argmin}_{i \neq j} \|o(n) - c_i(n)\|, \quad i = 1, n_h$$  \hspace{1cm} (15)

The second winning neurons will move away from the input its center with a ratio $\gamma$. All the others neurons will not change their centers vector.

So the learning law can be synthesized in the following relation:

$$c_i(n+1) = c_i(n) + \frac{\eta}{n_h} [o(n) - c_i(n)] \quad \text{if } i = j$$
$$c_i(n+1) = c_i(n) + \frac{\gamma}{n_h} [o(n) - c_i(n)] \quad \text{if } i = r$$
$$c_i(n) \quad \text{if } i \neq j \text{ and } i \neq r$$

where $\eta$ and $\gamma$ are the learning constants with real values between 0 and 1.

If the learning speed $\eta$ is chosen much greater than $\gamma$, the RBF network will find automatically the number of signal output classes. In other words, suppose that the number of classes is unknown and the number $n_h$ is greater than the number of the classes. The RBF centers will converge towards the centers of the input signals clusters. The penalizing competitive algorithm will move away the rival, in each iteration. If the $n_h$ is smaller than the number of the classes, the network will oscillate during training, indicating that the number of hidden neurons must be increased.

IV.2. LMS ALGORITHM

A supervised algorithm may be used to update the output neurons weights, for instance the LMS algorithm given by the following relations:

$$w_i(n+1) = w_i(n) + \alpha e(n) \phi(n)$$  \hspace{1cm} (17)

where $\alpha$ is the learning constant.

LMS minimizes the mean square error:

$$\text{MSE} = \frac{1}{N} \sum_{i=1}^{N} e_i(n)^2$$  \hspace{1cm} (18)

where $N$ is the number of input sequences and the complex error $e(n)$ is determined with:

$$e(n) = x(n-d) - f_{RBF}(o)$$  \hspace{1cm} (19)

V. SIMULATION RESULTS

QAM input signals were generated, using an uniform distribution, independently for the real part from the imaginary part. Simulations were done using the channel model presented in section III. The output channel $y(n)$ had one of 64 possible states. A white noise $w(n)$ was generated and added to $y(n)$. The FIR filter used in front of the neural equalizer had the order $m=1$. The number of the hidden neurons was chosen $n_h=64$ and of the output neurons 8. The RBF centers were randomly initialized to a subset of channel output values. The centers spread was chosen 0.28. The best results were obtained for the following learning constants: $\eta=0.09$, $\gamma=0.03$ and $\alpha=0.01$. There were applied $N=1000$ input signal sequences $x(n)$, $x(n)^2=[x(n) x(n-1) x(n-2)]$ to train the equalizer.
Fig. 4 Bidimensional representation of output channel states without noise, inputs of RBF network, initial and final positions of the RBF centers for N=1000 sequences.

Fig. 5 Evolution of mean square error during 5000 iterations for $\sigma^2=0.01$.

The equalizer was tested in different conditions of noise, with a noise dispersion $\sigma^2=0.01$, 0.1 and 0.5. In all the situations the equalizer succeeded to find a correct solution. The competitive algorithm trained the centers of the RBF network. For each sequence it has been calculated the error $e(n)$ and then the MSE. The output weights were modified according to the relation (17), in order to minimize the MSE.

In Fig. 4 are represented the output channel states $y(n)$, the corrupted received signal $o(n)$, the initial and final positions of the RBF centers in case of $\sigma^2=0.01$. This operation was repeated a number of times, in the aim of minimizing the MSE. Fig. 5 depicts the MSE evolution during 5000 iterations for $\sigma^2=0.01$ and a delay of $d=1$.

The complex space was divided in points using a sampling pas of $\delta=0.02$ to represent the decision regions of the RBF complex equalizer. Fig. 6 represents the partition signals space for a delay of $d=1$, in the worst conditions of noise, $\sigma^2=0.5$, which had strong nonlinear decisions boundaries.

Fig. 6 The output signals space partition.
VI. CONCLUSIONS

The main drawback of the neural network equalizers is the computational complexity and the extensive training. Our competitive algorithm, that recompenses the winner and penalizes the rival to train the centers of the RBF network, is rather simple and has a fast convergence to a solution. It generates strong nonlinear regions of decision in the signal space. So this algorithm is adequate to the adaptive equalization of fast varying signals corrupted with strong linear and nonlinear distortions. Because of its structure similar to the Bayesian equalizer the performance of the RBF equalizer is superior to the LTE and MLP equalizers. The MSE performance of our equalizer is comparative to others RBF equalisers reported in literature, tested in the same conditions. To improve the performances it might be increased the order of the LTE filter coupled with the RBF neural network.

References