Proceedings of the World Congress on Engineering 2009 Vol I
WCE 2009, July 1 - 3, 2009, London, U.K.

A Novel Cascaded Nonlinear Equalizer Configuration on Recurrent Neural Network Framework for Communication Channel

Susmita Das, IEEE Member

Abstract— **Recurrent neural network (RNN) exhibits better performance in nonlinear channel equalization problem. In this present work a hybrid model of recurrent neural equalizer configuration has been proposed where a discrete cosine transform (DCT) block is embedded within the framework of a conventional RNN structure. The RNN module needs training and involves updation of the connection weights using the standard RTRL algorithm, which necessitates the determination of errors at the nodes of the RNN module. To circumvent this difficulty, an adhoc solution has been suggested to back propagate the output error through this heterogeneous configuration. Performance analysis of the proposed Recurrent Transform Cascaded (RTCS) equalizer for standard communication channel models show encouraging results**.

Index Terms— **Recurrent Neural Network, equalizer, Bit Error Rate, Discrete Cosine transform, Normalization**

I. Introduction

Channel equalization is powerful techniques for compensating inters symbol interference in a dispersive communication channel, the nonlinearities introduced by the modulation/demodulation process and the noise generated in the system. However, linear equalizers do not perform well on channels with deep spectral nulls or with nonlinear distortions. Researchers have shown that nonlinear equalizers based nonlinear theory exhibit better performance than linear equalizers in applications where the channel nonlinear distortions exist [1], [2]. When the channel itself has nonlinear characteristics or nonlinear channel distortions are too severe to ignore, even the Decision Feedback Equalizer cannot recover the corrupted signals effectively. Since neural networks (NN) [3] can perform complex mapping between its input and output space, and are capable of forming complex decision regions with nonlinear decision boundaries, many types of NNs have successfully applied in channel nonlinear equalization problem [2].The use of NN's is justified by noting that in most cases, the boundaries of the optimal decision regions are highly nonlinear, thus requiring the use of nonlinear classifiers, even with linear channels. Among the techniques based NN, recurrent neural network (RNN) [4][5] proposed to solve the nonlinear channel equalization problem, RNN have shown better performance than feed forward neural networks, because that RNN approximate infinite impulse response (IIR) filters while feed forward neural networks approximate FIR filters, which makes them attractive in the presence of channels with deep spectral nulls. In addition, RNN is more attractive for their small size [6]. Results from the simulations show that the RNE with simple size can yield a significant improvement in performance relative to the equalizers with linear filter, and outperform MLP equalizers of larger computational complexity in no minimum phase, partial response, and nonlinear channel equalizations cases. Complex versions of the RNE based on a real time current learning (RTRL) algorithm are developed to process complex signals [7]. Although various algorithms and hybrid structures [8] have improved the performance of the RNE, the computational burdens would become greater. In summary, the heavy computational load and low convergence speed have limited the practical applications of RNE.

In this paper, a hybrid configuration has been proposed where a Discrete Cosine Transform (DCT) block is embedded within the framework of a conventional RNE structure. A signal vector is mapped from a given domain to another when fed to a transform block, because basically the transform block performs a fixed filtering operation. The basic difference between the transform block and the neural block is that while adaptive weights are associated with the later, fixed weights are inherent in the former. Hence, this cascaded network representing a heterogeneous configuration has been proposed to solve the conventional RNE problem keeping the complexity of the weight adaptation less. It is obvious that the transform block does not require any weight adaptation, but the RNN module needs updation of the connection weights using the standard RTRL algorithm, which necessitates the determination of errors at the nodes of the RNN module. To circumvent this difficulty, an adhoc solution has been suggested. The primary objective of the proposed work is to design cascaded RNE on reduced structural framework with faster convergence keeping in mind real-time implementation issue.

The organization of this paper is as follows. In Section II, cascaded RNE equalizer based on the hybrid technique as well as the modified version of the RTRL algorithm used to train it are described in detail. In Section III, the performances of the proposed equalizer through various simulations for linear and nonlinear channels are illustrated. Finally, Section IV summaries this research works.

II. Proposed Cascaded Neural Equalizer

Here, a real-valued discrete cosine transform followed by normalization block is cascaded with an RNN module at the output end as given in Figure 1. Power normalization technique [9] is applied to the transformed signals and the final output of the proposed structure is evaluated as a weighted sum of all normalised signals. In order to update the connection weights of this cascaded framework, a novel idea has been developed based on propagation of the output error through the network in the light of the conventional BP algorithm.

The transform block does not require any weight adaptation as it consists of fixed weights, but the RNN module needs updation of the connection weights using the standard RTRL algorithm, which necessitates the determination of errors at the nodes of the RNN module. But this estimate cannot be accomplished directly by using BP algorithm due to positioning of the transform block close to the output end, so problem is encountered here in propagating the final output error back into the network. To circumvent this difficulty, an adhoc solution has been evolved and error estimation at the input end of the transform block is done from the knowledge of the error at its output by considering its inverse transform. The mathematical expressions governing this concept is described in subsequent section

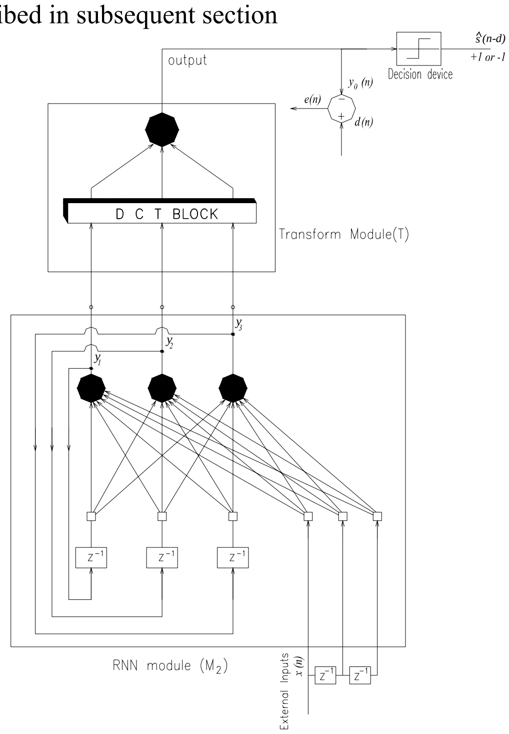

Figure 1 RNN−Transform cascaded equaliser structure

2.1 *Training algorithm of neural structure*

The proposed structure shown in Figure 1 consists of *nr* processing units in the RNN module with *nx* external inputs and a transform block. A step by step procedure has been adopted to update the weights of the neural network as mentioned below. Sensitivity parameters $\{p_{kl}^{j}\}$ of all RNN nodes are initialized to zero. The input signal to the proposed equalizer structure is represented by a *m* x 1 vector $\boldsymbol{x}(n) = \left[r(n), r(n-1), \ldots \; \ldots, r(n-m+1)\right]^T$.

Input signal vector to the RNN module is defined as $\boldsymbol{u}(n)$, l^{th} element of which is

$$u_l(n) = \begin{cases} y_j(n), & 1 \le j \le nr \\ x_i(n), & 1 \le i \le nx \end{cases} \quad \text{for } 1 \le l \le (nr+nx) \qquad (1)$$

The output of j^{th} neuron of the RNN module at time index *n* is given by

$$y_j(n) = \frac{1 - e^{-\phi . c_j(n)}}{1 + e^{-\phi . c_j(n)}} \qquad (2)$$

Where the net internal activity is described by

$$c_j(n) = \sum_{l=1}^{nx+nr} w_{kl}(n).u_l(n), \quad 1 \le k \le nr \qquad (3)$$

where **W** denotes *nr* by (*nx* + *nr*) weight matrix of the RNN module. Sigmoid activation functions (F) with slope parameter ϕ for neurons of the RNN module have been considered. Input signal vector to the transform block can be expressed as $\boldsymbol{z}(n)$**,** whose j^{th} element is denoted as,

$$z_j(n) = y_j(n), \qquad j = nr \qquad (4)$$

Here all the processing units of the RNN module act as visible units giving externally reachable outputs. The j^{th} element of the output from the transform block (DCT) is defined as

$$z_{\mathcal{T}j}(n) = DCT\left\{z_j(n)\right\} = \mathcal{T} z_j(n) \qquad (5)$$

The $T_{pq}^{\ th}$ element of the *N* X *N* transform matrix T is defined as

$$T_{pq} = \begin{cases} \frac{1}{\sqrt{N}}, & p=0, q=0,1,\ldots\ldots,N-1 \\ \left(\sqrt{\frac{2}{N}}\right)\cos\frac{\pi(2q+1)p}{2N}, & p=1,2,\ldots.N-1; q=0,1,\ldots.,N-1 \end{cases} \qquad (6)$$

The transformed signal $z_{\mathcal{T}j}(n)$ are then normalized by the square root of their power $\boldsymbol{B}_j(n)$, which can be estimated by filtering the $z_{\mathcal{T}j}(n)$ with an exponentially decaying window of scaling parameter $\gamma \in [\,0,1]$ as derived in the literature[102] and shown below. The j^{th} element of the normalized signal becomes

$$z_{\mathcal{N}j}(n) = \frac{z_{\mathcal{T}j}(n)}{\sqrt{\mathcal{B}_j(n) \quad + \varepsilon}} \qquad (7)$$

and

$$Bj\,(n) = \gamma \;\; \boldsymbol{B}_j(n\text{-}1) + (1 - \gamma)\, z_{\mathcal{T}j}^{2}(n) \qquad (8)$$

The scaling parameter $\gamma \in [\,0,1]$. The small constant ε is introduced to avoid numerical instabilities when signal power $\boldsymbol{B}_j(n)$ is close to zero.

The final output of the hybrid structure at time index *n*, $y_o(n)$ is expressed as the weighted sum of all normalized signals from the transform blocks.

$$y_o(n) = \sum_{j=1}^{nr} g_j(n).z_{\mathcal{N}j}(n) \qquad (9)$$

where **g** denotes the weight matrix at the output end of the proposed network.

The error at the equalizer output at time index *n* is en by,

$$e(n) = d_o(n) - y_o(n) \qquad (10)$$

With the knowledge of the output error, the errors at all the nodes of RNN module can be evaluated in order to facilitate the updation of weights using RTRL algorithm. But this is not possible directly as already explained before and hence a technique has been employed to tackle the situation.

At first the error *e(n)* is back propagated through various connection paths. Then the error at the j^{th} output of normalization block is computed as given by

$$e_{\mathcal{N}_j}(n) = e(n).g_j(n)\,, \; 1 \le j \le nr \qquad (11)$$

The error terms at the output of the transform block $\delta_{\mathcal{T}j}(n)$ can be calculated using the following approach. The power normalization can be considered as a process, whose operation is quite similar to the nonlinear transformation produced by sigmoid activation function of a neuron. This concept helps to calculate the error terms (i.e., local gradients) at the output of the transform block using the following equation

$$\delta_{\mathcal{T}j}(n) = e_{\mathcal{N}j}(n) \cdot \frac{\partial z_{\mathcal{N}j}(n)}{\partial z_{\mathcal{T}j}(n)}$$

$$= e_{Nj}(n) \quad \{z_{Nj}(n) \,/\, z_{Tj}(n)\} \quad \{1 - (1-\gamma)\; z^2_{\mathcal{T}j}(n)\} \tag{12}$$

Further, to propagate the error back through the transform block and to estimate the error magnitudes at the input side of the transform block, Inverse Discrete Cosine Transform (IDCT*)* is applied. This provides an estimate of the error at the input end of the transform block and the error at the j^{th} processing unit of the RNN module at time index n is given by

$$err_{rnn-node_j}(n) = IDCT\left\{\delta_{\mathcal{T}j}(n)\right\} \tag{13}$$

Application of RTRL algorithm involves primarily the evaluation of sensitivity parameter, a triply indexed set of variables $\{p^j_{kl}\}$ defined as[06].

$$p^j_{kl}(n) = \frac{\partial y_j(n)}{\partial w_{kl}(n)}, \quad k \in \mathrm{A} \text{ and } l \in \mathrm{A} \cup \mathrm{B}$$

Where, $\mathrm{A} = \{1, 2, ..., nr\}$ and $\mathrm{B} = \{1, 2, ..., nf\}$

The sensitivity parameters $\{p^j_{kl}\}$ are updated as follows

$$p^j_{kl}(n+1) = \mathrm{F}'\{c_j(n)\}\left[\sum_{i=1}^{nr} w_{ji}(n).p^i_{kl}(n) + \partial_{kj}u_l(n)\right]$$

$$1 \le j \le nr,\; 1 \le k \le nr \text{ and } 1 \le l \le (nr + nx) \tag{14}$$

Where $\mathrm{F}'\{c_j(n)\}$ is given in Equation (5.50) and ∂_{kj} is defined in Equation (11).

While the incremental weight change $\Delta g_j(n)$ is calculated using BP algorithm, RTRL algorithm computes the incremental weight change $\Delta w_{kl}(n)$.

$$\Delta g_j(n) = \theta\,.e(n).z_{Nj}(n), \qquad 1 \le j \le nr \tag{15}$$

$$\Delta w_{kl}(n) = \lambda.\sum_{j=1}^{nr} err_{rnn-node_j}(n).p^j_{kl}(n),$$

$$1 \le k \le nr \text{ and } 1 \le l \le (nr+nx) \tag{16}$$

Where λ and θ are the learning-rate parameters of the RNN module and the output layer respectively. The connection weights are updated as given below.

$$g_j(n+1) = g_j(n) + \Delta g_j(n) \tag{17}$$

$$w_{kl}(n+1) = w_{kl}(n) + \Delta w_{kl}(n) \tag{18}$$

The recursion process of updating weights of the cascaded network continues till a predefined condition is achieved as have been mentioned earlier.

III. SIMULATION RESULTS AND DISCUSSION

An exhaustive computer simulation study has been undertaken for evaluating the performance of all the proposed neural equalizer structures based on FNN topologies for a variety of linear and non-linear real communication channels models. The simulation model of an adaptive equalizer considered is illustrated in Figure 2. In the simulation study the channel under investigation is excited with a 2-PAM signal, where the symbols are extracted from uniformly distributed bipolar random numbers {-1,1}. The channel output is then contaminated by an AWGN (Additive White Gaussian Noise). The pseudo-random input and noise sequences are generated with different seeds for the random number generators. For mathematical convenience, the received signal power is normalized to unity. Thus the received signal to noise ratio (SNR) is simply the reciprocal of the noise variance at the input of the equalizer. The power of additive noise has been taken as 0.01, representing a SNR of 20dB.

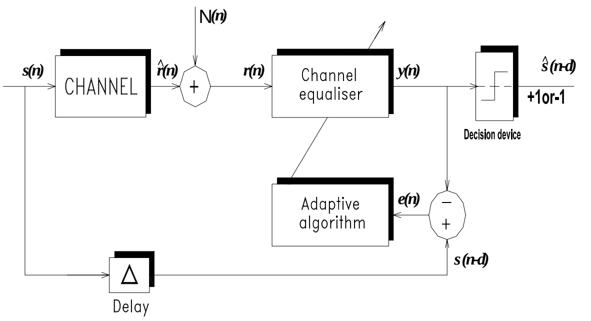

Figure.2. Simulation model of Channel Equalizer in Training Phase

Equalization of different types of channel models (both linear and non-linear type) are attempted in order to establish the efficacy of the proposed equalizer structures based on RNN topology and to prove their robustness. It has been already reported in the literatures [6][7], that a two-unit, one input, one output RNN is a non-linear IIR model which is sufficient to model many communication channels. Considering this aspect, all the proposed cascaded equalizers in RNN framework are compared with a conventional RNN equalizer (CRNN) with two recurrent units and one external input sample from the channel output. Further the TDRNN structure has two nodes in RNN module followed by a 2 x 2 DCT block with power normalization and a summing unit at the output end. For a comparative study and analysis purpose the number of training samples presented to the proposed equalizer considered here are restricted to 200 samples only as it is observed that their performances are quite satisfactory. The BER performance comparison of the proposed equalizer structures based on RNN topology has been carried out after all the structures has undergone a training phase(200 samples) The weight vectors of the equalizers are frozen after the training stage is over and then the performance test is continued. The BER performances for each SNR are evaluated, based on 10^7 more received symbols (test samples) and averaged over 20 independent realizations.

All the proposed equalizers in RNN domain require fewer samples in training phase for satisfactory BER performance. Simulation results demonstrate this advantages offered by these structures. Figure 5.18 shows the effect of change of length of training sequence on the BER performance obtained using the conventional RNN equalizer. It is shown for channels *H1*(z) and $H_8(z)$ that increasing the length of learning phase from 200 to 1000 samples, the CRNN equalizer still could not achieve the BER performance level

ISBN: 978-988-17012-5-1

of the HKFRCS equalizer. For the RTCS structure, the number of processing units remains the same as the CRNN equalizer. After the input signal is preprocessed in the RNN module, it is fed to the DCT transform block for further processing. As expected, such a proposed structure performs better than a CRNN due to the further signal de-correlation in the transform block followed by power normalization. Another three tap channel characterized by

$$H_1(z)=0.407 - 0.815\, z^{-1} - 0.407\, z^{-2} \qquad (19)$$

RTCS equalizer show distinct SNR gains of about 4.4 dB at a prefixed BER level of 10^{-4} over a pure RNN equalizer which is quite encouraging.

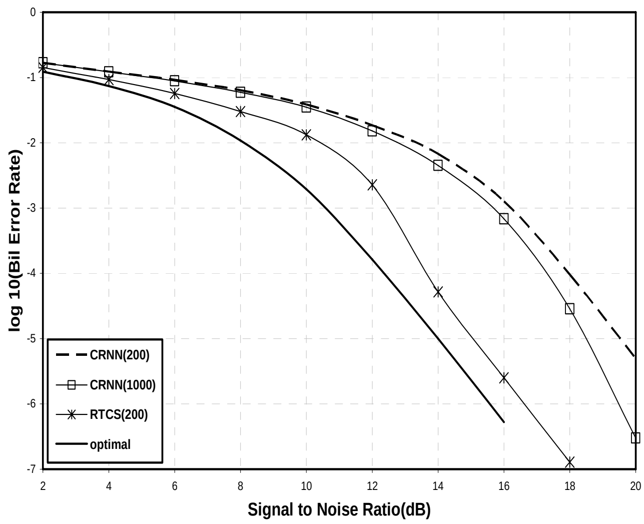

Figure 3 BER performance of the proposed hybrid equalizers for channel $H_1(z)$

Of processing units remains the same as the CRNN equalizer. After the input signal is preprocessed in the RNN module, it is fed to the DCT transform block for further processing. As expected, such a proposed structure performs better than a CRNN due to the further signal de-correlation in the transform block followed by power normalization. Another three tap channel characterized by

$$H_1(z)=0.407 - 0.815\, z^{-1} - 0.407\, z^{-2} \qquad (19)$$

RTCS equalizer show distinct SNR gains of about 4.4 dB at a prefixed BER level of 10^{-4} over a pure RNN equalizer which is quite encouraging.

In order to prove the robustness and consistency in performance of all the proposed neural structures, equalization of nonlinear channels is simulated. Such nonlinear channels are frequently encountered in several places like the telephone channel, in data transmission over digital satellite links, especially when the signal amplifiers operate in their high gain limits and in mobile communication where the signal may become non-linear because of atmospheric nonlinearities. These typical channels encountered in real scenario and commonly referred to in technical literatures [4] are described by following transfer functions.

$$H_2(z)= (1+ 0.5\, z^{-1}) - 0.9\, (1+ 0.5\, z^{-1})^3 \qquad [20]$$

For this example the proposed RTCS equalizers result significant 2dB gain in SNR level at a prefixed BER of 10^{-4} over the CRNN equalizer which clearly justifies their application for such type of channel.

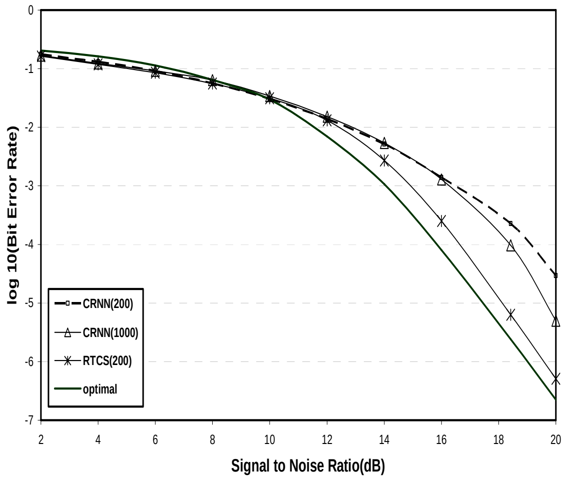

Figure 4 BER performance of the proposed hybrid equalizers for channel $H_2(z)$

IV. CONCLUSION

A real-valued transform is a powerful signal decorrelator which performs whitening of the signal by causing the Eigen value spread of an auto-correlation matrix to reduce. The proposed neural equalizers with hybrid structures have outperformed their conventional counterparts to a large limit and require less number of samples in training phase simultaneously As BER performance is a significant measure of channel equalizer and proposed hybrid neural structure has an edge over conventional ones and close to the theoretically optimal Bayesian equalizers. Further a reduced structure with low computational complexity can be utilized. These hybrids ANN architecture has opened up new directions in designing efficient adaptive nonlinear equalizers and can be implemented in DSP processors for real - time applications.

REFERENCES

[1] G.J. Gibson, S. Siu, S. Chen, C.F.N. Cowan and P.M. Grant, “The Application of Nonlinear Architectures to Adaptive Channel Equalisation”, *Proceeding ICC’90 (Atlanta*, USA), pp312.8.1-312.8.5, April 1990.

[2] S. Theodoridis, C.F.N. Cowan, C.P. Callender and C.M.S. See, “Schemes for Equalisation of Communication Channels with Non-linear Impairments”, *IEE Proceedings – Communication*, Vol.142, No.3, pp. 165-171, June 1995

[3] S. Haykin, Neural networks – A Comprehensive Foundation. New York : Macmillan, 1994.

[4] G.Kechriotis, E. Zervas and E.S. Manolakos, “Using Recurrent Neural Networks for Adaptive Communication Channel Equalisation”, *IEEE Transaction on Neural Networks*, Vol.5, No.2, pp.267-278, March 1994.

[5] J. Choi, M. Bouchard, and T. H. Yeap, “Decision feedback recurrent neural equalization with fast convergence rate,” *IEEE Trans. Neural Netw.*,Vol.16, no.3, pp. 699-708, 2005.

[6] R. Parisi, E.D. Di Claudio, G. Orlandi and B.D. Rao, "Fast Adaptive Digital Equalisation by Recurrent Neural Networks", *IEEE Transactions Signal Processing*, Vol. 45, No. 11, pp. 2731-2739, November 1997.

[7] J.D. Ortiz-Fuentes and M.L. Forcada, "A Comparison between Recurrent Neural Network Architectures for Digital Equalisation", *IEEE International Conference on Acoustics, Speech and Signal Processing*, ICASSP-97, Vol. 4, pp. 3281-3284, 21-24 April 1997.

[8] H. R. Jiang and K. S. Kwak, "On modified complex recurrent neural network adaptive equalizer," *Journal of Circ. Syst. Comput.*, Vol. 11, No. 1, pp. 93-101, 2002

[9] X. Wang, H. Lin, J. Lu, and T. Yahagi, "Combining recurrent neural networks with self-organizing map for channel equalization," *IEICE Tranaction in. Commun.*, Vol. E85-B, pp. 2227-2235, Oct. 2002.

[10] F.Beaufays, "Transform-Domain Adaptive Filters: An Analytical Approach", *IEEE Transactions on Signal Processing*, Vol. 43, no. 2, pp. 422-431, February 1995.