*IEEE-International Conference on Signal processing, Communications and Networking*
*Madras Institute of Technology, Anna University Chennai India, Jan 4-6, 2008. pp274-278*

# Design of Adaptive Channel Equaliser on Neural Framework Using Fuzzy Logic Based Multilevel Sigmoid Slope Adaptation

Susmita Das
*Member IEEE,*Department of Electrical Engineering
National Institute of Technology, Rourkela -769 008

***Abstract* - Adaptive equalisation in digital communication systems is a process of compensating the disruptive effects caused mainly by inter symbol interference in a band-limited channel and plays a vital role for enabling higher data rate in modern digital communication system. Designing efficient equalisers having low structural complexity and faster learning algorithms is also an area of much research interest in the present scenario. This research work proposes adaptive channel equalisation techniques on Recurrent Neural Network framework. Exhaustive simulation studies carried out prove that by replacing the conventional sigmoid activation functions in each of the processing nodes of recurrent neural network with multi-level sigmoid activation functions, the bit error rate performance have significantly improved. Further slopes of different levels of the multi-level sigmoid have been adapted using fuzzy logic control concept. Simulation results cosidering standard channel models show faster learning with less number of training samples and performance level comparable to the their conventional counterparts. Also there is scope for parallel implementation of slope adaptation technique in real-time implementation.**

## I. INTRODUCTION

Equalisation is a powerful techniques to combat Inter Symbol Interference and distortion in a band limited communication channel and is employed at the receiving end to compensate for such distortions. By viewing equalisation as a classification problem in which the optimal decision boundary is highly nonlinear, the solution offered by a linear equaliser is inherently sub-optimal. This drawback motivated the development of efficient non-linear equalisers for optimising the performance. In the recent past a lot of researches have been carried out using Artificial Neural Network (ANN) techniques since neural networks are well known for their ability of performing classification tasks. These include the development of many novel architectures and efficient training algorithms. The prime advantage of using neural networks for equalisation is their capability to model any non-linear decision boundaries and hence they are well suited for developing high performance adaptive equalisers[1, 2].

Different ANN architectures such as multilayer perceptron (MLP), radial basis function (RBF) and recurrent neural networks (RNN) for constructing adaptive equalisers have been suggested in the literatures [3]. Recurrent Neural Networks (RNN) store additional information about the past signals in the form of an internal state and incorporate feedback mechanism. As a result their architectures become inherently dynamic. Actually RNNs model non-linear IIR filters and can accurately realise the inverse of finite memory channels using relatively small number of neurons. Further RNNs are in some respect very similar to efficient Decision Feedback Equalisers (DFE) because in that outputs are fed back to the classifier to assist in subsequent decisions [4, 5, 6].

The most widely used Real–Time–Recurrent–Learning (RTRL) algorithm, proposed by Williams and Zipser [7], update the weights of the RNN by computing the gradient of the squared error with respect to the weights of the equaliser. Adaptive equalisation of linear and non-linear channels using Recurrent Neural Network Equalisers have shown superiority over traditional equalisation algorithms as reported in the literature [8]. The block diagram of a conventional RNN based adaptive equaliser is illustrated in Fig. 1.

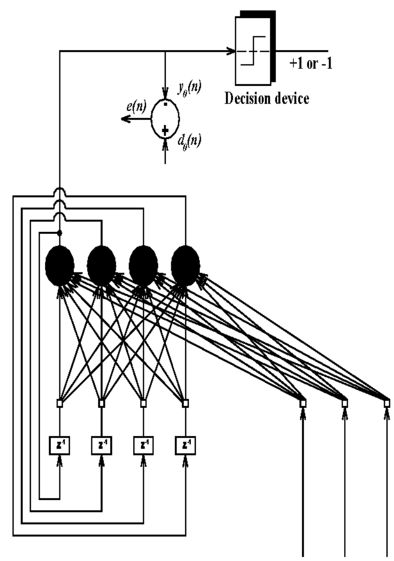


Fig. 1 A conventional RNN Equaliser

## II. DESCRIPTION OF PROPOSED RNN BASED NEURAL EQUALISERS

In this present work, processing units of conventional RNN equaliser are replaced by Multi-level sigmoid activation function [9]. For this purpose of varying number of levels, the sigmoidal activation function is taken as [9]

$$f_M(x) = \sum_{i=1}^{M-1} \frac{1 - e^{-\beta(x-\varphi_i)}}{1 + e^{-\beta(x-\varphi_i)}} \quad (1)$$

(6) where $\varphi_i, i = 1, 2, .....M-1$ are points at which a level change occurs. $\beta$ is slope parameter. $\varphi_1$ called as shifting parameter is set to a fixed negative value and, are calculated using the equation

$$\varphi_i = \varphi_{i-1} - \delta \text{ ,where } \delta = \frac{2\varphi_1}{M-2} \tag{2}$$

Now in we study the effect of level M and slope $\beta$ on the performance of equaliser. Number of levels in sigmoidal activation function can be changed by varying value of M. For M=2, $f_M$ becomes simple sigmoid function when all $\varphi_i$ are zeros. The four-level activation function is shown in Figure 2 for different values of $\varphi_1$ and $\beta$.

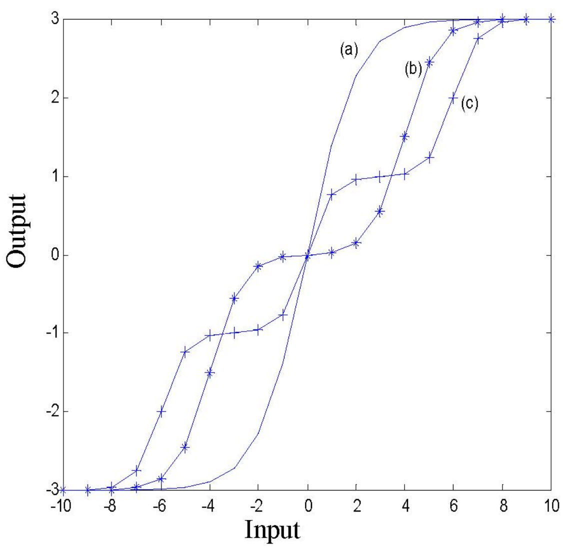


Fig. 2. Examples of 4-level sigmoid function
(a) $\beta$=1, $\varphi_1$=0, (b) $\beta$=1.5, $\varphi_1$ = -4, (c) $\beta$=2, $\varphi_1$ = -6

## III. MODIFIED RTRL ALGORITHM

The RTRL algorithm used for weight updation has been suitably modified due to the presence of the M-level activation function at each node. While the modified RTRL algorithm takes control by recursively updating the network weights and threshold values, the fuzzy controller approach adjusts the slope of each level of the activation function at the same time. The BER performances of these proposed RNN based equalisers are compared with that of a conventional recurrent neural equaliser. The learning algorithm for proposed structure is summarized below.

For every time step *n*, use the dynamic equations of the RNN network, the output values of the *n* neurons is computed. For the initial values of the weights are chosen from a set of uniformly distributed random numbers. The RNN chosen here has *nx* external inputs and *nr* fully interconnected recurrent processing units. Thus input of RNN is a vector *u(n)*, $l^{th}$ element of which $u_l(n)$ is define

$$u_1(n) = \begin{cases} y_k(n), & 1 \le k \le nr \\ x_k(n), & 1 \le k \le nx \end{cases} \quad \text{for } 1 \le l \le (nx+nr) \tag{3}$$

The output of $k^{th}$ neuron of RNN at time *n*

$$y_k(n) = \sum_{i=1}^{M-1} \frac{1 - e^{-\beta(C_k(n)-\varphi_i)}}{1 + e^{-\beta(C_k(n)-\varphi_i)}} \tag{4}$$

$$\text{where, } c_k(n) = \sum_{l=1}^{nx+nr} w_{kl}(n) \cdot u_l(n) \tag{5}$$

*w* denotes *nr* by *(nx + nr)* weight matrix of RNN. The final output of the proposed equaliser structure is taken from the output of $j^{th}$ neuron of RNN. By comparing $y_j(n)$ with the desired value *d(n),* the error $e_j(n)$ is calculated.

$$e_j(n) = d(n) - y_j(n) \tag{6}$$

Since $e_j(n)$ is known at all times, determination of the partial derivative is required to implement RTRL algorithm.

A *sensitivity parameter* is described by a triply indexed set of variables $\{p^j_{kl}\}$, where

$$p^j_{kl}(n) = \frac{\partial y_j(n)}{\partial w_{kl}(n)}, \quad 1 \le k \le nr \text{ and } 1 \le l \le (nr+nx) \tag{7}$$

The evaluation of this partial derivative is carried out as follows: $p^j_{kl}(0) = 0$, (8)

$\partial_{kj}$ is a *Kronecker delta* defined as

$\partial_{kj}=1$ for $k = j$

$= 0$, otherwise. (9)

The variables $p^j_{kl}$ for all appropriate *j, k,* and *l* is

$$p^j_{kl}(n+1) = F'(c_j(n))\left[\sum_{i=1}^{nr} w_{ji}(n) \cdot p^i_{kl}(n) + \partial_{kj} u_l(n)\right] \tag{10}$$

taking initial condition $p^j_{kl}(0) = 0$ and the derivative

$$F'(c_j(n)) = \sum_{i=1}^{M-1}\left(\beta * 2 * \frac{1-e^{-\beta(C_k(n)-\varphi_i)}}{\left(1+e^{-\beta(C_k(n)-\varphi_i)}\right)^2}\right) \tag{11}$$

Use the values of $p^j_{kl}$ obtained and error signal $e_j(n)$ to compute the corresponding weight changes

$$\Delta w_{kl}(n) = \eta \sum_{j=\zeta} e_j(n) p^j_{kl} \tag{12}$$

where $\eta$ is the learning-rate parameter.

Update the weight $w_{kl}$ in accordance with

$$w_{kl}(n+1) = w_{kl}(n) + \Delta w_{kl}(n) \tag{13}$$

and repeat the computation till the error is minimized.

The output of the decision device can be defined as

$$\hat{s}(n-d) = \begin{cases} 1 & \text{if } \quad y(n) \ge 0 \\ -1 & \text{otherwise} \end{cases} \tag{14}$$

The *w*s (weights) in Equation (13) are values specified by the training algorithm, so that after training is completed the equaliser will self-adapt to the changes in the channel characteristics occurring during transmission (decision directed mode).

## IV. PROPOSED RNN BASED EQUALISER WITH SLOPE ADAPTATION

The error term at individual (neuron) of a neural structure is to be minimised to get a pseudo-optimal solution. In a neural network paradigm the synaptic weights and threshold values are generally considered as free parameters in conventional sense, which are adapted using appropriate learning algorithms in order to train the network. The proposed structure is built around the concept of tuning the slope parameter of each level of the multilevel sigmoid activation function to enhance the adaptability of the network. In the present research work attempt has been made to adapt the slope of the sigmoid activation function only using the fuzzy logic controller approach. The fuzzy logic controller technique [10,11] is applied to determine the amount of correction needed for the slope of the sigmoidal activation function at each node of the network. Basically a fuzzy controller evaluates the change in the control action based on the information regarding error and rate of change of error at the process output. The same concept is adopted in the present work.

The node error term is known as $e_j(n)$ and its rate of change $\Delta e_j(n)$ is described by are fed into the fuzzy controller block as shown in Figure 3.

$$\Delta e_j(n) = e_j(n) - e_j(n-1) \tag{15}$$

The output generated from the control block $\Delta\,\phi(n)$, as shown in Figure 3, is used to obtain the changed slope at the $(n+1)^{th}$ time index $\phi(n+1)$ of the sigmoidal activation function using the relation

$$\phi\ (n+1) = \phi\ (n) + \Delta\phi\ (n) \tag{16}$$

In the present investigation, seven categories of linguistic variables {Large Positive (LP), Medium Positive (MP), Small Positive (SP), Zero (ZE), Small Negative (SN), Medium Negative (MN) and Large Negative (LN)} are employed to describe both the input variables and the output . The membership functions of fuzzy controller structure are assumed to have Gaussian type distribution [11] and have fixed centres and widths. The fuzzified inputs are used to construct the rule base. Taking into account the linguistic information of the inputs and with a-priori knowledge about the bounds of the sigmoid slope variation, the controller output is decided.

In order to reflect this concept a fuzzy rule base is constructed as given in Table 1. The fuzzy control rules are expressed in the form of 'IF…THEN' statements and some of the interpretations of the fuzzy rules for sigmoidal slope adjustments have been listed below for clarification.

- *If $\delta(n)$ is SN and $\Delta\delta(n)$ is SP, then $\Delta\phi\ (n)$ is ZE.*
- *If $\delta(n)$ is SP and $\Delta\delta(n)$ is LP, then $\Delta\phi\ (n)$ is LP.*
- ………………………………………………………
- *If $\delta(n)$ is MP and $\Delta\delta(n)$ is LN, then $\Delta\phi\ (n)$ is SN.*

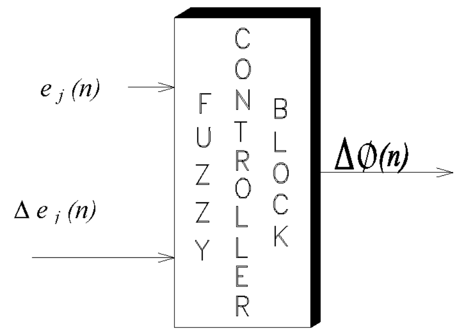


Fig. 3 Fuzzy logic control block

Table 1: Fuzzy Rule Base

| $\delta(n)$ \ $\Delta\delta(n)$ | | ← Error term change → | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | | LN | MN | SN | ZE | SP | MP | LP |
| ← Error term → | LN | LN | LN | LN | LN | MN | SN | ZE |
| | MN | LN | LN | LN | MN | SN | ZE | SP |
| | SN | LN | LN | MN | SN | ZE | SP | MP |
| | ZE | LN | MN | SN | ZE | SP | MP | LP |
| | SP | MN | SN | ZE | SP | MP | LP | LP |
| | MP | SN | ZE | SP | MP | LP | LP | LP |
| | LP | ZE | SP | MP | LP | LP | LP | LP |

## V. SIMULATION STUDY

The simulation model of an adaptive equaliser considered is illustrated in Figure. 4. 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. The power of additive noise has been taken as 0.01, representing a SNR of 20dB. In the training phase the proposed equalisers are trained with 50 training samples and an ensemble average of 10 independent trials have been taken.

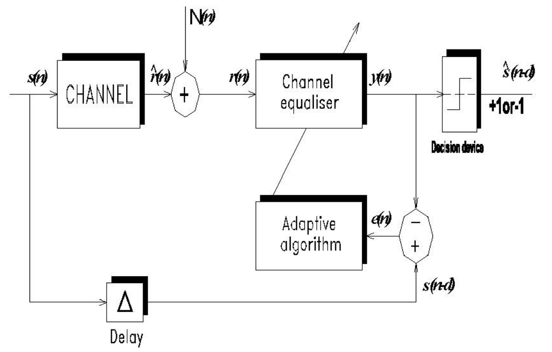


Fig. 3 Simulation model of adaptive equaliser in training phase

For the performance evaluation in testing phase, bit error rate (BER) of each of proposed equalisers is observed over $10^5$ test samples and 50 ensembles. Four-level sigmoid activation functions with intial slopes as discussed in Section 2 are used in the proposed equalisers. While the basic RTRL algorithm with suitable modification explained in Section III updates the network weights, the slope of each level of the multilevel sigmoid at the same time is tuned following the technique discussed in Section IV.

## VI RESULTS AND DISCUSSIONS

The first example used is two-tap minimum-phase channel [12] with transfer function defined as

$$H_1(z) = 0.5 + 1\,z^{-1} \qquad (17)$$

A Recurrent Neural Equaliser with a single recurrent unit with 2 inputs is used here and the decision is delayed by one sample. The conventional RNN equaliser results a poor performance (it yields a BER level below $10^{-2.5}$ at 18 dB SNR condition), where as the proposed RNN with Multi sigmoid activation function (RNNMS) and further RNN with sigmoid slope tunimg (RNNST) equalisers exhibit significant improvement in BER performance at 18dB SNR (a practical condition) as shown in Fig. 4. A 4-level sigmoid function before and after alope adaptation by FLC approach used in the proposed neural equaliser is illustrated in Fig. 5.

The next channel under study is a 5 tap channel [12] with transfer function defined as

$$H_2(z) = -\,0.2052 - 0.5131\,z^{-1} + 0.7183\,z^{-2} + 0.3695\,z^{-3} + 0.2052\,z^{-4} \qquad (18)$$

A conventional RNN equaliser with 2 processing units,2 external inputs and decision delay 4 is used. It is observed that in Fig.6, the application of the RNNMS and RNNST equalisers demonstrate SNR gains of more than 3 dB and 5 dB at a prefixed BER level of $10^{-3}$ over the conventional RNN Further the result of slope adaptation all levels in a 4-level sigmoid function is shown in Fig. 7.

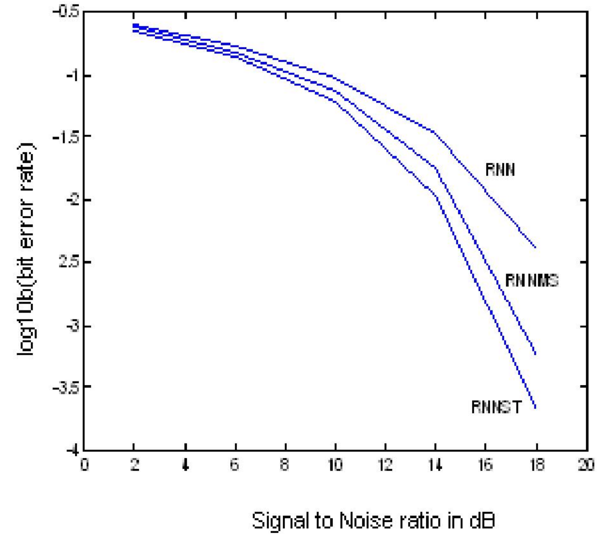


Fig. 4 BER performance comparison for channel $H_1(z)$

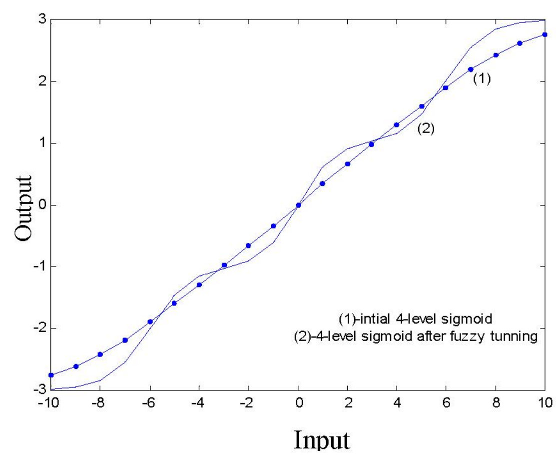


Fig. 5 Slope adaptation of a 4-level sigmoid for channel $H_1(z)$

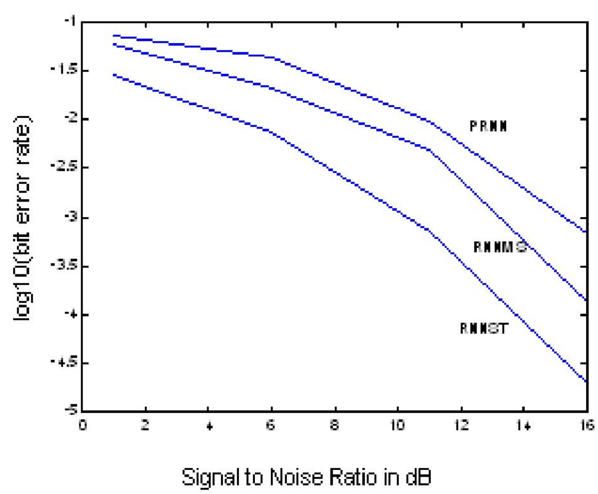


Fig. 6. BER performance comparison for channel $H_2(z)$

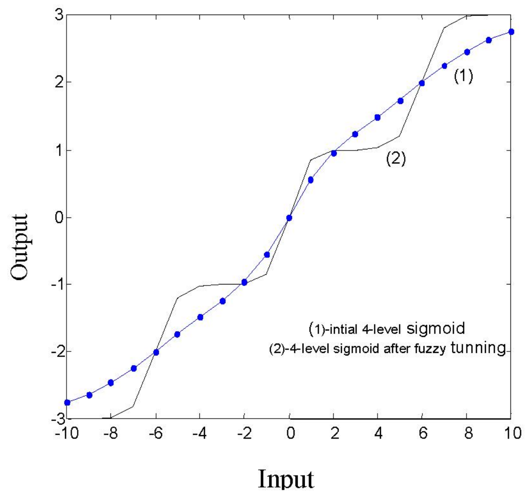


Fig. 7 Slope adaptation of a 4-level sigmoid for channel $H_2(z)$

## VII. CONCLUSION

In the present work adaptive equalisers are proposed on the backbone of Recurrent Neural Network and their efficacy are verified by considering various standard channel models. It is observed in the simulation study that by replacing the conventional sigmoid activation functions in the RNN processing units with Multi-level sigmoids, the bit error rate performances have significant improvement. The adaptation of the slope parameter increases the degrees of freedom in the weight space of the conventional Recurrent Neural Network configuration Further, tuning of slope parameters of different levels of the M-level activation function has been incorporated using fuzzy logic concept which has enhanced the RNN equaliser performance to a higher limit. These new approaches offer better adaptability and also result faster learning and performance improvement Further parallel implementation of sigmoid slope adaptation along with the weight adaptaton during training mode will be beneficial in realtime implementation using DSP, FPGA processors.

## REFERENCES

[1] G.J. Gibson, S. Siu and C.F.N. Cowan, "Multilayer Perceptron Structures Applied to Adaptive Equalisers for Data Communications", *International Conference on ASSP, ICASSP 89*, vol. 2, pp. 1183-1186, May 1989.

[2] P. Power, F. Sweeney and C.F.N. Cowan,"Non-Linear MLP Channel Equalisation", *IEE Colloquium on Statistical Signal Processing*,pp. 3/1-3/6, 6 January 1999.

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

[4] M.J. Bradley and P. Mars, "Application of Recurrent Neural Networks to Communication Channel Equalisation", *International Conference on Acoustics, Speech and Signal Processing, ICASSP-95*, vol. 5, pp. 3399-3402, 9-12 May 1995.

[5] 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.

[6] R. Parishi, E.D.D Claudio, G. Orlandi and B.D. Rao, "Fast Adaptive Digital Equalisation by Recurrent Neural Networks", *IEEE Transactions Neural Networks*, vol. 45, no. 1, pp. 2731-2739, November 1997.

[ [7] R.J. Williams and D. Zipser, "A Learning Algorithm for Ckontinually Running Fully Recurrent Neural Networs", *Neural Computation*, vol. 1, pp. 270-280, 1989.

[8] 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.

[9] Kadri Hacioglu, "An improved Recurrent Neural Network for M-PAM Symbol Detection", *IEEE Transactions on Neural networks,* Vol. 8, No. 3, May 1997.

[10] A.F. Stronach, P. Vas and M. Neuroth, "Implementation of Intelligent Self-organising Controllers in DSP Controlled Electromechanical drives", IEE Proceeedings, Control Theory Application, Vol.144, No.4, pp.324-330,July 1997.

[11] J.K. Satapathy and C.J. Harris, "Application of Fuzzy-Tuned Adaptive Feedforward Neural Networks for Accelerating Convergence in Identification", *3rd International Conference on Industrial Automation*, pp. 6.1, June 1999.

[12] S. Chen, G.J. Gibson, and C.F.N. Cowan, "Adaptive Channel Equalisation using a polynomial–perceptron structure", *Proceeding - I of the IEE*, Vol. 137, pp. 257-264, October 1996.