Relaxed Look-Ahead Pipelined Nonlinear Channel Equalizer

Daniel MASSICOTTE and Thierry DUFOUR

Department of Electrical and Computer Engineering, Université du Québec à Trois-Rivières
C.P. 500, Trois-Rivières, Québec, Canada, G9A 5H7
Tel. : (819)-376-5011 (3918), Fax : (819)-376-5219, Web: http://lssi.uqtr.ca
E-mail: {Daniel_Massicotte, Thierry_Dufour}@uqtr.ca

ABSTRACT

This paper presents a pipelined equalizer for nonlinear channel environment using the relaxed look ahead technique. Nonlinear channel is a known problem that can be treated by artificial neural network (ANN). But, the recursive nature of the adaptation algorithms found in ANN limits their operation speed and high throughput applications require pipelining. The standard look ahead technique is used to pipeline recursive type algorithms and the result often presents a large amount of hardware due to pipelining. In order to minimize this hardware, the relaxed look-ahead technique uses approximations of the look-ahead technique. The combination of these approximations in conjunction with parameter variation can result in a large variety of architectures. This paper will present the relaxed look-ahead technique in a general form and its application to the backpropagation algorithm found in ANN. Simulation results with linear and nonlinear channels will be shown for different pipeline depth.

1. INTRODUCTION

Popularity of digital communications takes modern communications system to operate at very high data rates (e.g. ADSL, WLAN). But, the effects of the communication channel at high data rates cannot be set aside and have to be corrected by the use of an equalizer. The original data sequence $s(n)$ is transmitted over the forward model of the nonlinear channel which provides the noisy signal $\tilde{y}(n)$

$$\{\tilde{y}(n)\} = \Im\left[\{s(n)\}, \Theta\right] + \eta(n) \qquad (1)$$

where the channel's output $\tilde{y}(n)$ is a signal corrupted with additive noise $\eta(n)$ and $\mathbf{\Theta}$ represents the vector of the parameters of the operator $\Im$. The problem of nonlinear channel equalization is to solve the reconstruction problem, which consists of a regularized inversion of the operator $\Im$, i.e., an operator of reconstruction $\Re$

$$\{\hat{x}(n)\} = \Re\left[\{\tilde{y}(n)\}, \mathbf{\Theta}\right] \qquad (2)$$

The received signal $\tilde{y}(n)$ leads us to deduce the value of the estimated entry $s(n)$ through a corrective processor called a channel equalizer. The equalizer produces $\hat{x}(n)$ which is saturated by the decision function to give $\hat{s}(n)$ as the estimate of $s(n)$. Transverse linear filter are often used as equalizer due to their simplicity but they do not perform well in nonlinear channel environment. The use of nonlinear equalizer like Artificial Neural Network (ANN) [4,5] or Fuzzy logic [6,7] makes nonlinear channel equalization a reality, at the expense of algorithm complexity. Usually, no information on the channel is available, the use of adaptive algorithm is imperative.

Channel equalization (see Fig. 1) consists of updating filter weights by the use of a linear algorithm on the reconstitution error $e(n)$. Common used adaptive algorithms are least mean square (LMS), recursive least square (RLS) applied to linear filter, Kalman filter and more specifically the backpropagation algorithm applied to ANN. A training data sequence known by the receiver is used to adapt the equalizer coefficients. In invariant or non stationary environment, periodic transmission of the adaptation sequence is done or blind adaptation procedure [3] can be used to track channel variations.

The recursiveness of the adaptation procedure limit the clock frequency of the equalizer VLSI structure and the throughput of the output data $\hat{s}(n)$. This limitation is due to the data dependency of one sample between the error signal $e(n)$ to adapt the equalizer parameters. A manner to increase the clock frequency and the throughput of the VLSI structure consist to apply the pipeline technique. In the equalizer case of linear transversal filter with LMS adaptation, the relaxed look-ahead approach is appropriate (e.g. [10-12])

This paper deal with the relaxed look-ahead technique applied to ANN to equalize the nonlinear channel. Section 2 summarizes the relaxed look-ahead technique. This technique is applied, in Section 3, to an ANN equalizer based on backpropagation. Section 4 presents the simulation results follows by the conclusion in Section 5.

2. RELAXED LOOK-AHEAD TECHNIQUE

We will now introduce the relaxed look-ahead technique to pipeline recursive algorithms. The technique consists of approximating (relaxing) the result obtained by the standard look-ahead technique (e.g. [10]). Many types of relaxations can be used and their application depends of the variations in the input data and the reconstitution error *e(n)* [11-14]. There

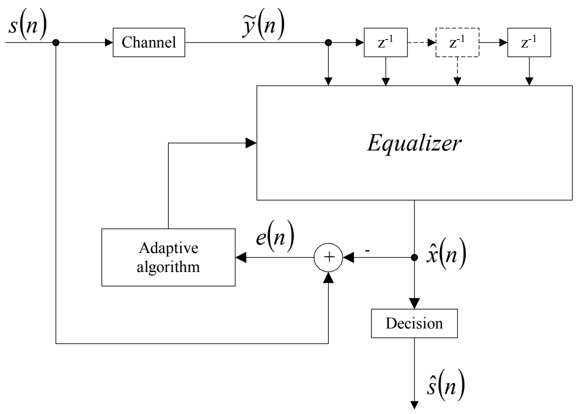


Fig. 1: Channel equalization technique

are no rules in applying relaxations but care has to be taken since every case is unique. Sum and delay are the most common relaxations. Let's start with a first order recursive equation

$$x(n) = x(n-1) + a(n)u(n) \tag{3}$$

Applying look-ahead pipelining [10] of a length K leads to

$$x(n)=x(n-K)+\sum_{i=0}^{K-1} a(n-i)u(n-i) \tag{4}$$

When the number of filter coefficient (N) and the value of K are large, the amount of hardware overhead due to pipelining increases and VLSI implementation are not very efficient [11]. The idea is to get the summation minimized by limiting its length. This is known as sum relaxation.

Sum Relaxation

The sum relaxation can be applied if the product $a(n)u(n)$ varies slowly over K samples. We then limit the sum to a value of LA where $1 \le LA \le K$. We obtain the following result

$$x(n)=x(n-K)+\frac{K}{LA}\sum_{i=0}^{LA-1} a(n-i)u(n-i) \tag{5}$$

The factor K/LA is introduced to keep the profile of (5) identical to (4) for stationary $a(n)$ and $b(n)$. To help pipeline the summation itself, we need delay relaxation.

Delay Relaxation

The delay relaxation inserts delays in the terms used for summation. This approximation is valid only if the gradient of $a(n)u(n)$ is stable over D_1 samples. We then get

$$x(n)=x(n-K)+\sum_{i=0}^{K-1} a\left(n-D_1-i\right)u\left(n-D_1-i\right) \tag{6}$$

The application of sum and delay relaxations can be combined and the resulting architectures can take many forms depending of the pipelining depth and the parameter values. The clock frequency of the equalizer VLSI structure and throughput of the data output increase with the parameters D_1 and LA.

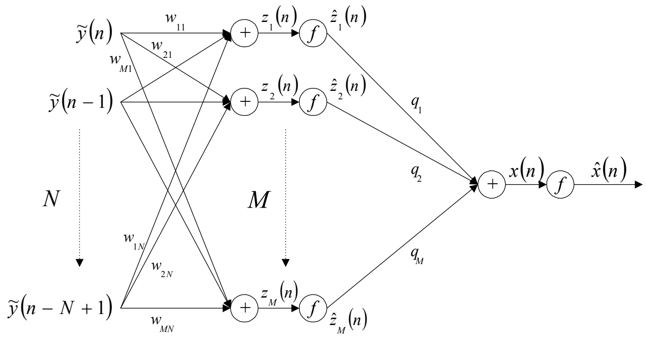


Fig. 2: ANN general form.

3. Artificial Neural Network

Fig. 2 shows the ANN in its general equalizer form where N and M denotes respectively the number of inputs and the number of neurons on the hidden layer. It has been shown that one hidden layer is sufficient for channel equalization [15,16]. The ANN hidden and output layers are calculated in a matrix form by

$$\hat{\mathbf{z}}(n) = f\left\{\mathbf{W}(n)\tilde{\mathbf{y}}(n)\right\} \tag{7}$$

$$\hat{x}(n) = f\left\{\mathbf{q}^T\hat{\mathbf{z}}(n)\right\} \tag{8}$$

where $f\{\bullet\}$ is the hyperbolic tangent activation function.

The ANN hidden and output weights are updated using the backpropagation algorithm given by

$$e(n)=\left[s(n-d)-\hat{x}(n)\right] \tag{9}$$

$$q(n)=q(n-1)+e(n)\hat{\mathbf{z}}(n)^T \tag{10}$$

$$\mathbf{W}(n)=\mathbf{W}(n-1)+e(n)\left[\left(\mathbf{q}(n)^T f'\{\mathbf{z}(n)\}\right)\tilde{\mathbf{y}}(n)^T\right] \tag{11}$$

where $s(n-d)$ is the known sequence for the adaptation. The delay d introduced is used to compensate the communication channel delay and $f'\{\bullet\}$ is the derivative of the activation function.

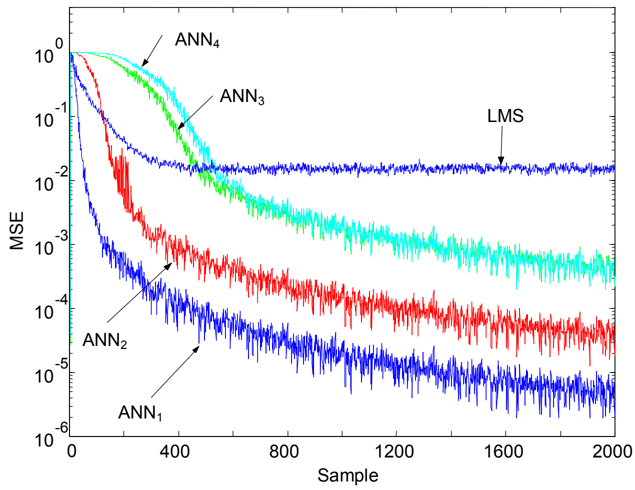
Equations (10) and (11) are recursive type and relaxed look ahead technique can be applied resulting in (12) and (13).

$$\mathbf{q}(n)=\mathbf{q}\left(n-D_2\right)+\frac{\mu}{LA}\sum_{k=0}^{LA-1} e\left(n-D_1-k\right)\left[\hat{\mathbf{z}}\left(n-D_1-k\right)\right]^T \tag{12}$$

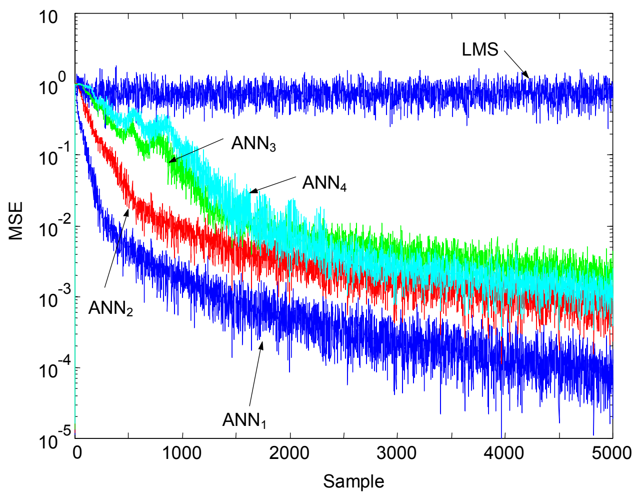
$$\mathbf{W}(n)=\mathbf{W}\left(n-D_2\right)+\frac{\mu}{LA}\sum_{k=0}^{LA-1}\Big\{e\left(n-D_1-k\right)\bullet$$
$$\left[\left\{\mathbf{q}\left(n-D_1-D_2-k\right)^T f'\left(\mathbf{z}\left(n-D_1-k\right)\right)\right\}\tilde{\mathbf{y}}\left(n-D_1-k\right)^T\right]\Big\} \tag{13}$$

Matrix and vector sizes are given by:

$$\begin{aligned}&\dim[\mathbf{W}]=M\times N\\&\dim[\mathbf{z}]=\dim[\mathbf{q}]=\dim[\tilde{\mathbf{y}}]=M\times 1\end{aligned} \tag{14}$$

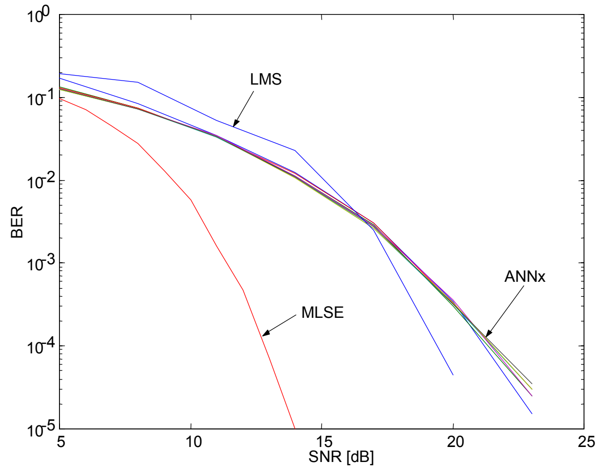


a)

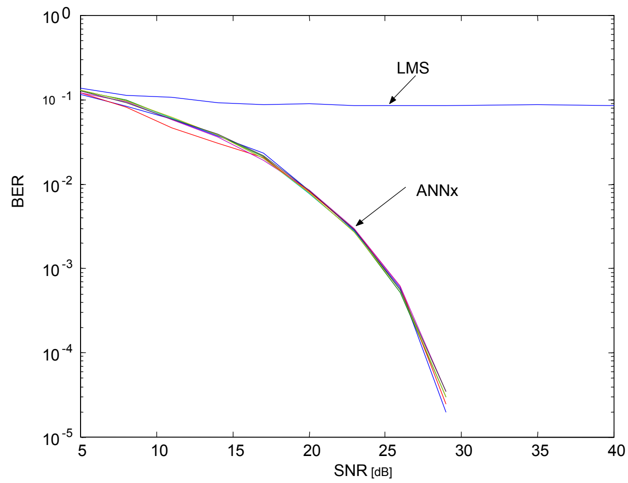


b)

Fig. 3: Mean Square Error for a) linear channel and b) non linear channel.



a)



b)

Fig. 4: BER vs SNR for a) linear channel and b) non linear channel.

4. Simulations Results

A simulation study with a conventional linear equalizer based on transversal filter using a simple LMS adaptation procedure [2] and the ANN using the proposed adaptation structure has been realized for both linear and nonlinear channels. Channel equalization simulations were realized with non return to zero (NRZ) on off keying (OOK) random synthetic data. The linear channel is given by

$$\tilde{y}(n)=\mathbf{h}(n)^T\,\mathbf{s}(n)+\eta(n) \tag{15}$$

where the channel impulse response is $\mathbf{h}=[h_0\ h_1\ h_2\ \ldots\ h_L]^T$. The input vector of the channel is original data $\mathbf{s}(n)=[s(n)\ s(n\text{-}1)\ s(n\text{-}2)\ \ldots\ s(n-L+1)]^T$, $\eta(n)$ is an additive Gaussian white noise and L is the length of the channel.

The nonlinear channel is given by

$$\begin{aligned}\tilde{y}(n)&=0.5v(n)+v(n)^3+\eta(n)\\ v(n)&=\mathbf{h}(n)^T\,\mathbf{s}(n)\end{aligned} \tag{16}$$

Coefficients for both linear and nonlinear invariant and stationary channel are

$$\mathbf{h}=\begin{bmatrix}h_0 & h_1 & h_2\end{bmatrix}^T=[0.3482\quad 0.8704\quad 0.3482]^T \tag{17}$$

The input signal $s(n)$ is assumed to be an independent sequence taking values from {-1,1} with equal probability and $\eta(n)$ denotes the zero mean white Gaussian noise to obtain a signal noise ratio (SNR) of 5 dB to 40 dB. The quality of correction is assessed using the bit error rate (BER). Each algorithm used 5 000 data for the adaptation of its parameters for a SNR from 5 dB to 40 dB.

Table 1 shows the simulations pipeline parameters for linear and nonlinear channel. For ANN_x (x=1,2,3,4) structures, the

Table 1: Pipeline parameters

Case	D_1	D_2	LA	LC, μ	NLC, μ
ANN_1	0	1	1	0.3	0.12
ANN_2	20	1	1	0.1	0.04
ANN_3	60	1	1	0.03	0.02
ANN_4	60	4	4	0.125	0.1

parameters are set to M=3, N=3, which correspond to the minimum to ensure the algorithm convergence, and the step size μ is set to optimize the convergence for linear channel (LC) and nonlinear channel (NLC). Mean Square Error (MSE) results of the ANN_x are compared with 7-tap and 23-tap LMS transverse filter for the linear channel and nonlinear channel respectively, the step size μ is set to 0.1.

Fig 3 shows the mean of 200 repetitions of MSE for both linear and nonlinear channels with SNR to 35 dB. We can see that convergence properties are altered by the pipeline depth. Also, residual error is higher as the pipeline gets deeper but the ANN_x always converge with lower residual MSE than the LMS algorithm. In the nonlinear channel case, the LMS algorithm fails to converge which clearly show its limitation to linear problems. Simulations with $LA<D_2$ has shown that the value of he sum relaxation parameter LA has to be made equal to parameter D_2 to maintain convergence. This important characteristic makes the hardware overhead an issue for large number of inputs N.

Robustness to noise of the equalizer for linear and nonlinear environment is depicted in Fig. 4. For these simulations, 10 000 data are generated to test the algorithms, the BER was calculated after each learning on linear and nonlinear channels. Fig. 4 illustrates the mean of 20 repetitions. In the linear case, the result of Maximum Likelihood Sequence Estimator (MLSE) (Viterbi algorithm) with perfect knowledge of channel taps is shown as a reference point. We can see that all ANN_x have the same equalization characteristics for both cases which demonstrate that pipelining the backpropagation algorithm preserve the equalizer performance.

5. Conclusion

In this paper, we have presented an algorithm approximation method based on the standard look-ahead technique. This relaxed look-ahead technique is used to pipeline recursive type algorithm with minimal hardware overhead. ANN based channel equalizer has been pipelined using the relaxed look-ahead technique and simulation results has shown that convergence properties are affected by the pipeline depth but not the BER results. ANN structure with a depth pipeline gives an appropriate solution to equalize a nonlinear channel. Comparison was made with conventional LMS transverse filter and we saw that ANN equalization performances are kept at the same level even in a nonlinear channel application.

Acknowledgement

This project was supported by the Natural Sciences and Engineering Research Council of Canada (NSERC).

References

[1] S. Haykin, Adaptive Filter Theory, Prentice-Hall, 1996

[2] J.G. Proakis, and M. Salehi, Communication systems Engineering, Prentice Hall, 1994.

[3] O. Macchi, "L'égalisation numérique en communication", *Ann. Télécomm.*, Vol 53, No 1-2, 1998.

[4] G. Kechtiotis, E. Zervas, E. S. Manolakos, "Using Recurrent Neural Networks for Adaptive Communication Channel Equalization", *IEEE Transactions on Neural Networks*, Vol. 5, No. 2, pp. 267-278, March 1994.

[5] R. Parisi et al., "Fast Adaptive Digital Equalisation by Recurrent Neural Networks", *IEEE Trans. on Signal Processing*, Vol. 45, No. 11, pp 2731-2739, Nov. 1997.

[6] L.-X. Wang and J. M. Mendel, "Fuzzy Adaptative Filter With Application to Nonlinear Channel Equalization", *IEEE Transactions on Fuzzy System*, Vol. 1, No. 3, August 1993, pp. 161-170.

[7] M. Zakhama et D. Massicotte, "A Systolic Architecture for Channel Equalization Based on a Fuzzy Logic Algorithm", *IEEE Canadian Conference on Electrical and Computer Engineering (CCECE'99),* Calgary, 1999, pp. 1098-1101.

[8] G. Long, F. Ling, and J. G. Proakis, "The LMS algorithm with delayed Coefficient adaptation", *IEEE Trans. Acoust., Speech, Signal Processing,* vol. 37, Sept. 1989, pp. 1397–1405.

[9] K.K. Parhi, and D.G. Messerschmitt, "Pipeline interleaving and parallelism in recursive digital filter – Parts I-II", *IEEE Transactions on acoustics, speech and signal processing*, vol 37, no 7, pp. 1099-1134.

[10] N. Shanbag, "Algorithm Transformation Techniques for Low-Power Wireless VLSI Systems Design", *International Journal of Wireless Information Networks*, vol. 5, no 2, 1998.

[11] N.R. Shanbag and K.K. Parhi "Relaxed Look Ahead Pipelined LMS Adaptative Filters and their Application to ADPCM Coder", *IEEE Transaction on circuits and systems*, vol. 40, no 12, 1993.

[12] N.R. Shanbag and G.H. Im, "VLSI Design of 51.84Mb/s Transceivers for ATM-LAN and Broadband Access", *IEEE Transaction on signal processing*, vol. 46 no 5 1998.

[13] A.K. Shaw and M.I. Ahmed, "Pipelined recursive digital filters: a general look-ahead scheme and optimal approximation", *IEEE Trans. on circuits and systems – II*, vol 46, no 11, 1999.

[14] K. Matsubara, K. Nishikawa, H. Kiya, "Pipelined LMS adaptive filter using a new look-ahead transformation", *IEEE Trans. on circuits and systems*, vol 46, no 1, 1999.

[15] M. Vidal and D. Massicotte, "A Parallel Architecture of a Piecewise Linear Neural Network for Nonlinear Channel Equalization", *IEEE Instrumentation Measurement Technology Conf. (IMTC/99)*, Italy, 24-26 May 1999, pp. 1629-1634.

[16] M. Vidal and D. Massicotte, "QPSK Equalizer Based on Piecewise Neural Network", *Int. Conf. on Wireless Communications (Wireless'99)*, Calgary, 12-14 July 1999, pp. 499-502.