Optimized Adaptive equalizer for Wireless Communications

Ch.Sumanth kumar, D.Madhavi
Department of ECE
GITAM Institute of Technology
GITAM University
Visakhapatnam

N.Jyothi
Department of EIE
GITAM Institute of Technology
GITAM University
Visakhapatnam

KVVS Reddy
Department Of ECE
AU College of Engineering
Andhra University
Visakhapatnam

ABSTRACT

In this paper, a simple and efficient low complexity fast converging partial update normalized LMS (PNLMS) algorithm is proposed for the decision feedback equalization. The proposed implementation is suitable for applications requiring long adaptive equalizers, as is the case in several high-speed wireless communication systems. The proposed algorithm yields good bit error rate performance over a reasonable signal to noise ratio. In each iteration, without reducing the order of the filter, only a part of the filter coefficients are updated so that it reduces the computational complexity and improves speed of operation. The NLMS algorithm can be considered as a special case and slightly improved version of the LMS algorithm which takes into account the variation in the signal level at the filter output and selects a normalized step size parameter which results in a stable as well as fast converging adaptive algorithm. The frequency domain representation facilitates, easier to choose step size with which the proposed algorithm convergent in the mean squared sense, whereas in the time domain it requires the information on the largest eigen value of the correlation matrix of the input sequence. Simulation studies shows that the proposed realization gives better performance compared to existing realizations in terms of convergence rate.

Keywords: Adaptive filtering, Bit error rate(BER), Mean Square error (MSE), Normalized least mean square (NLMS) algorithm.

1. INTRODUCTION

The adaptive decision feedback equalizer (ADFE) [1] provides a very effective means for equalizing communication channels that exhibit spectral nulls and / or have very long impulse response spanning several symbol periods. The ADFE consists of a feed forward filter (FFF) and a feedback filter (FBF) and decision device. The received signal pass through the FFF first to cancel the pre-cursor ISI, and subtract with the result of FBF to cancel the post-cursor ISI. Both the FFF and the FBF coefficients are trained by some appropriate adaptive algorithm. Among the other algorithms, the stochastic gradient LMS algorithm for tap weight adaptation of the ADFE is most commonly used [4]. In high-speed applications, an ADFE with a large number of taps in the FFF and FBF is required. However, the implementation and the real-time operation of such an equalizer is a difficult task, due to increased complexity and the very small inter symbol period. Another important issue in adaptive equalization is the one of convergence speed. Fast converging equalizers are highly desirable since they require a reduced training sequence, thus offering a valuable saving in bandwidth. The issue of convergence is usually traded off with the issue of complexity. ADFEs based on the recursive least squares (RLS) algorithm exhibit very fast convergence, but unfortunately, they require a large number of operations per time step. On the other hand, the conventional ADFE, which is based on the LMS algorithm, has a much lower complexity as compared with the RLS-based ADFE, but its convergence is very slow, especially in channels that contain spectral nulls. Moreover, even the low computational burden of the LMS based ADFE turns out to be prohibitive in demanding applications. To overcome this difficulty, several efficient implementations of the LMS-based ADFE have been proposed in literature (e.g., [7]-[12]). The works [7]-[11] are mostly based on the assumption that the channel impulse response has the discrete sparse form. The algorithms in the referenced works are possibly applicable to other channels as well, but in such a case, their complexity would become quite high. Moreover, in some cases, the efficiency depends strongly on the constellation used. The NLMS algorithm has been developed from different viewpoints. Goodwin and Sin [15] formulated the NLMS algorithm as a constrained optimization problem. Nitzberg[13] obtained the recursion by running the conventional LMS algorithm many times, for each new input sample. Like the LMS, the NLMS is also a stochastic implementation of the steepest-descent algorithm where the mean value of the filter coefficients converge towards their optimal solution. Therefore, the filter coefficients will fluctuate about their optimum values given by the Wiener solution. The amplitude of the fluctuations is controlled by the step size. The smaller the step size, the smaller the fluctuations (less final misadjustment) but also the slower the adaptive coefficients converge to their optimal values. The NLMS algorithm estimates the energy of the input signal each sample and normalizes (divides) the step size by this estimate, therefore selecting a step size inversely proportion to the instantaneous input signal power. One of the advantages of the NLMS algorithm is that the step size can be chosen independent of the input signal power and the number of tap weights. Although this improves the convergence properties in comparison to the LMS, it does not solve the eigenvalue spread problem. The NLMS algorithm shows stable convergence behavior when the step size μ (convergence constant) takes a value between zero and an upper limit defined by the statistics of the filter's input signal. The fastest convergence will be achieved for a white noise input sequence with zero mean and unit variance.

In Section II a new normalized partial update LMS based adaptive formulation is derived. In Section III some typical simulation results are discussed.

2. PROPOSED IMPLEMENTATION

Let us first formulate the conventional LMS algorithm as follows:

$$y(n) = w^t(n)x(n) \tag{1}$$

$$e(n) = d(n) - y(n) \tag{2}$$

$$W(n+1) = w(n) + \mu e(n)x(n) \tag{3}$$

Where, y(n) is the filter output and w(n) is the weight vector, which can be expressed as,

$$W(n) = [w_1(n), w_2(n).....w_L(n)]^t \tag{4}$$

Here, x(n) is the input vector, which can be expressed as,

$$X(n) = [x(n), x(n-1)........x(n-L+1)]^t \tag{5}$$

Where L is the length of the FIR filter, $e(n)$ is the error signal and $d(n)$ is the desired response during the initial training phase and decision directed during subsequent phase. μ is the step size. By applying this conventional LMS algorithm to the ADFE which is shown in Fig.1,

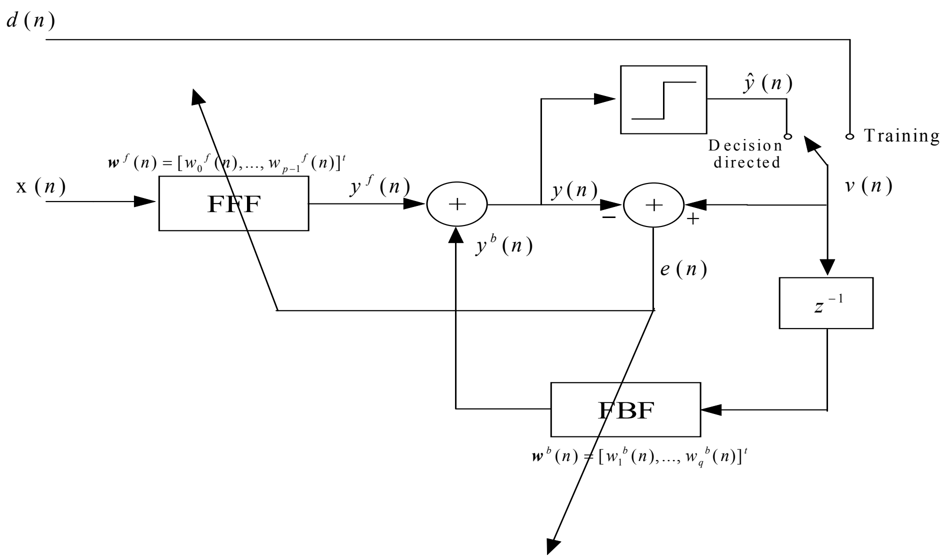

Fig.1. Front end of an Adaptive Decision Feedback Equalizer.

The reformulated equations are,

$$\hat{y}(n) = Q[y(n)] \tag{6}$$

$$y(n) = W^t(n)\phi(n) \tag{7}$$

$$W(n) = [W^{ft}(n)W^{bt}(n)]^t \tag{8}$$

$$\phi(n) = [x(n),.....x(n-p+1), v(n-1),....v(n-q)]^t \tag{9}$$

where Q[.] represents quantization,

$$W^f(n) = [w_1^f(n), w_2^f(n),....w_p^f(n)]^t \tag{10}$$

is a p[th] order feed forward filter (FFF) and

$$W^b(n) = [w_1^b(n), w_2^b(n),....w_q^b(n)]^t \tag{11}$$

is a q[th] order feedback filter (FBF) at index n. The signal v(n) is given by a desired response d(n) during the initial training phase and by $\hat{y}(n)$ during the subsequent decision directed phase.

The weight updating equation is

$$W(n+1) = W(n) + \mu\phi(n)e(n) \tag{12}$$

Where, $e(n) = v(n)-y(n)$ is the output error at index n and μ is an appropriate step size.

To reduce the computational complexity we use [14] , the partial updating the filter coefficients using LMS algorithm,

For the instant 'n', the filter coefficients are separated as even and odd indexed terms as,

$$W_e(n) = [w_2(n), w_4(n), w_6(n),.......w_L(n)]^t \tag{13}$$

$$W_o(n) = [w_1(n), w_3(n), w_5(n), ...w_{L-1}(n)]^t \quad (14)$$

The input sequence also divided as even and odd sequences as,

$$X_e(n) = [x(n-1), x(n-3)........x(n-L+1)]^t \quad (15)$$

$$X_o(n) = [x(n), x(n-2)........x(n-L+2)]^t \quad (16)$$

For odd n, the filter coefficients updated using partial update LMS algorithm (PLMS) are given by,

$$W_e(n+1) = W_e(n) + \mu e(n) X_e(n) \quad (17)$$

$$W_o(n+1) = W_o(n) \quad (18)$$

For even n, the filter coefficients are,

$$W_e(n+1) = W_e(n) \quad (19)$$

$$W_o(n+1) = W_o(n) + \mu e(n) X_o(n) \quad (20)$$

Define the coefficient error vectors as,

$$V_e(n) = W_e(n) - W_e(opt) \quad (21)$$

$$V_o(n) = W_o(n) - W_o(opt) \quad (22)$$

$$V(n) = W(n) - W(opt) \quad (23)$$

$$V^{eo}(n) = [V_e(n), V_o(n)]^t \quad (24)$$

Where,

$$W_e(opt) = [w_2(opt), w_4(opt), w_6(opt), ...w_L(opt)] \quad (25)$$

$$W_o(opt) = [w_1(opt), w_3(opt), w_5(opt), ...w_{L-1}(opt)] \quad (26)$$

$$W(opt) = [w_1(opt), w_2(opt), w_3(opt), ...w_L(opt)]^t \quad (27)$$

For regular LMS algorithm, the recursion for mean coefficient error vector $E[v(n)]$ is given by,

$$E[v(n+1)] = (I - \mu R) E[v(n)] \quad (28)$$

Where I is an N dimensional identity matrix, and $R = E[x(n)x^t(n)]$ is the input signal correlation matrix.

The necessary and sufficient condition for stability of the recursion is given by

$0 < \mu < \frac{2}{\lambda_{max}}$, where λ_{max} is the maximum eigen value of the input signal correlation matrix R.

For odd n,

$$E[v(n+2)] = (I - \mu I_2 R)(I - \mu I_1 R) E[v(n)] \quad (29)$$

For even n,

$$E[v(n+2)] = (I - \mu I_1 R)(I - \mu I_2 R) E[v(n)] \quad (30)$$

For stability, the eigen values of $(I - \mu I_1 R)(I - \mu I_2 R)$ should lie inside the unit circle. Instead of just two partitions of even and odd coefficients (P=2), we have any number of arbitrary partitions $(p \geq 2)$ then the update equations can be similarly as above with p>2.

Namely,

$$E[v(n+P)] = \pi_{i=1}^{p} (I - \mu I_i R) E[v(k)] \quad (31)$$

If R is a positive definite matrix of dimension $N \times N$ with eigen values lying in the open interval (0, 2) then, $\pi_{i=1}^{p}(I - I_i R)$ has eigen values inside the unit circle.

I_i, i=1,2,…..p is obtained from I, the identity matrix of dimension $N \times N$, by zeroing out some rows in I such that $\sum_{i=1}^{M} I_i$ is positive definite.

A. Implementation of Normalized Partial update LMS ADFE Structure

The weight update equation for the NLMS algorithm is generally given by

$$w(n+1) = w(n) + \mu(n) e(n) x(n) \quad (32)$$

Where , $w(n) = [w_0(n), w_1(n), ...w_{L-1}(n)]^t$ is the tap vector at the n[th] index,

$X(n) = [x(n), x(n-1)...., x(n-L+1)]^t$ is the tap input vector,

$e(n) = d(n) - w^t(n) X(n)$ is the error signal with d(n) being the desired response available during the initial training period. The variable $\mu(n)$ denotes the so-called time varying step size parameter. In the most commonly used form of the NLMS algorithm, $\mu(n)$ is taken as

$\frac{\mu}{\|x(n)\|^2 + \theta} e(n)$, where $\|x(n)\|^2 = x^t(n)x(n)$, is the norm of the input signal, which eliminates the problem of *gradient noise amplification*.

μ is a step size control parameter, used to control the speed of convergence and chosen

Here the adaptation constant μ is with in the range 0 to 2 for convergence and θ is an appropriate positive number introduced to avoid divide-by-zero like situations which may arise when the norm of the input signal becomes very small.

Now since NLMS was obtained as a stochastic-gradient approximation to Newton's method, and given the superior convergence speed of Newton's recursion as compared to the standard steepest-descent recursion, we expect NLMS to exhibit a faster convergence behavior than LMS.

For high speed digital communications, the input sequence **x**(n), which is partitioned into non-overlapping blocks of length p, is applied to an FIR filter of length *L*, one block at a time. The tap weights of the filter are updated using normalized tap update equation, after the collection of each block of data samples, so that the adaptation of the filter proceeds on a block-by-block basis rather than on a sample-by-sample basis as in conventional LMS algorithm. With the *j*-th block, the filter coefficients are updated from block to block as,

$$W(j+1) = W(j) + \mu \sum_{r=0}^{p-1} X(jp+r)e(jp+r) \qquad (33)$$

Where $W(j) = [w_0(j), w_1(j), \ldots. w_{L-1}(j)]^t$ is the tap weight vector corresponding to the *j*-th block,

$$X(jp+r)=[x(jp+r),x(jp+r-1),\ldots x(jp+r-L+1)]^t \qquad (34)$$

and $e(jP + r)$ is the output error at $n = jP + r$, given by,

$$e(jp+r) = d(jp+r) - y(jp+r) \qquad (35)$$

The sequence $d(jP + r)$ is the so-called desired response available during the initial training period and $y(jP + r)$ is the filter output at $n = jP + r$, given as,

$$y(jp+r) = W^t(j)X(jp+r) \qquad (36)$$

The parameter μ popularly called the step size parameter is to be chosen as

$$0 < \mu < \frac{2}{trace\{R\}}$$, for convergence of the algorithm.

However, deriving a block-adaptive DFE is not a straightforward task. This is due to an inherent "causality" problem appearing in the block formulation of the DFE equations. Specifically, in order to obtain the decision symbol at a given time n, the respective decisions at times n-1, n-2…..n-N are required (where N is the length of the Feedback filter). However by implementing the block based DFE in frequency domain, it provides low complexity and faster convergence. The inherent "causality" problem appearing in the block formulation of the DFE is overcome by replacing the unknown decisions with properly derived tentative decisions. An initial estimation of these tentative decisions is taken via a minimization criterion that exploits all the available information. Then, these initial decisions are improved by applying a nonlinear iterative procedure, which is executed at each block. This procedure converges to the optimum, in the MMSE sense, decisions within a few steps. The whole algorithm is implemented in the frequency domain, thus offering all the advantages of such an implementation. The algorithm has a steady-state performance that is identical to that of the conventional symbol-by-symbol DFE and remarkably faster convergence rate. The simulations have shown that its overall performance is practically insensitive to the choice of the block length. Additionally, the complexity of the new algorithm is substantially lower, as compared with that of the conventional DFE.

The both feedforward and feedback filter coefficients are trained by the weight update equations of Normalized partial update block LMS algorithm. Initially the training is imparted by a pilot sequence d(n) (Known transmitted sequence) during initial training mode and by the output decision $\hat{y}(n)$ during the subsequent decision directed mode.

The output v(n) = d(n) or $\hat{y}(n)$ depending on whether it is the initial training period or subsequent decision directed phase.

The feed forward filter output, $y^f(n)$ is,

$$y^f(n) = w^f(n)x(n) \qquad (37)$$

Where $W^f(n) = [w_0^f(n), \ldots\ldots. w_{p-1}^f(n)]^t$

The feed back filter output, $y^b(n)$ is,

$$y^b(n) = w^b(n)v(n-1) \qquad (38)$$

Now the overall output, which is the input to the decision device, y(n) is,

$$y(n) = y^f(n) + y^b(n) \qquad (39)$$

3. SIMULATION RESULTS

The random number generator provides the test signal and in the channel additive white gaussion noise is added. The experimental plots for the proposed algorithms are tested. The ensemble averaging was performed over 100 independent trials of the experiment. Significant improvement in the steady state performance and the convergence characteristics are identified. The transmitted signals are taken as simple QPSK signals. The MSE curves for the proposed ADFE is shown for different values of step size μ are shown in Fig.2. Significant improvement in the convergence characteristics are observed with fewer computations as compared with the traditional LMS algorithm for normalized partial update LMS(NPLMS) based ADFE structure.

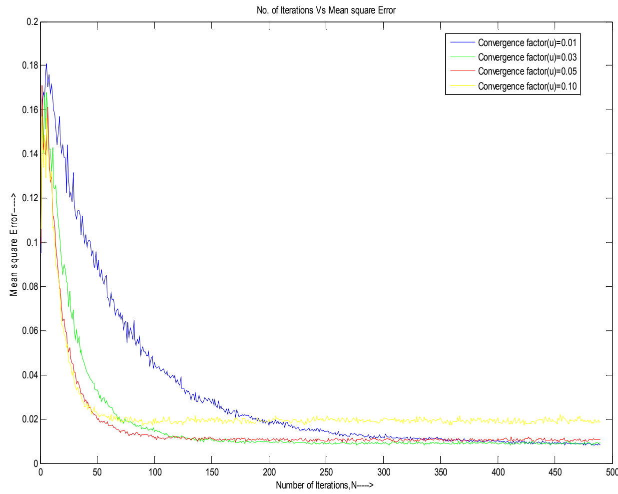

Fig.2. MSE curves for normalized partial update LMS based ADFE for different μ values.

4. CONCLUSION

Adaptive Decision feedback equalizers with high convergence and low complexity are highly desirable in mobile and wireless communication systems. In this paper a new normalized partial update block LMS based ADFE'S has been developed, which exhibits good steady-state performance and convergence characteristics with less computations as compared to traditional LMS based ADFE. The performances of the designed algorithms are verified by plotting the MSE curves.

5. REFERENCES

[1] Qureshi, S. U. H., "Adaptive Equalization," *Proc. IEEE*, vol. 73, no. 9, pp. 1349-1387, Sept. 1985.

[2] B. Farhang- Boroujeny, *Adaptive Filters – Theory and Application*, John Wiley and Sons, Chichester, UK, 1998.

[3] Simon Haykin., *Adaptive Filter Theory*, Prentice Hall, NJ.Second Edition.1991.

[4] Wu, W. R and D. G. Tsuie, "An LMS- based decision Feedback equalizer for IS-136 receivers ," *IEEE Trans. Commun.*, vol. 51, pp. 130-143, 2002.

[5] J. G. Proakis, "Adaptive equalization for TDMA digital mobile radio," *IEEE Trans. Veh. Technol.*, vol. 40, pp. 333- 341, May 1991.

[6] Davidson, G. W *et al*., "An investigation of block adaptive decision feedback equalization for frequency selective fading," *IEEE Int. Conf.Commun.*, May 1988.

[7] Ariyavisitakul, S, N. R. Sollenberger and L. J. Greenstein, "Tap- selectable decision feedback equalization," *IEEE Trans. Commun.*, vol. 45, pp. 1498-1500, Dec. 1997.

[8] S. A. Raghavan *et al*., "Nonuniformly selectable tapped delay line equalizers," *IEEE Trans. Commun.*, vol. 41, pp. 1290-1295, Sept. 1993.

[9] Berberidis, K., T. A. Rontogiannis and S. Theodoridis, "Efficient block implementation of the DFE," *IEEE Signal Processing Letters*, vol. 5, pp.129-131, June 1998.

[10] I. J. Fevrier, S. B. Gelfand and M. P. Fitz, "Reduced complexity decision feedback equalization for multipath channels with large delay spreads,"*IEEE Trans. Commun.*, vol. 47, pp. 927-937, June 1999.

[11] Berberidis, K., and P. Karaivazoglou, " An efficient block adaptive decision feedback equalizer implemented in the frequency domain," *IEEE Trans. Signal Processing*, vol. 50, no. 9, pp. 2273-2285, Sept. 2002.

[12] Rontogiannis, A. A. and K. Berberidis, "Efficient decision feedback equalization for sparse wireless channels," *IEEE Trans. Wireless Communications*, vol. 2, no. 3, pp. 570-581, May 2003.

[13] Nitzberg.R, "Application of the normalized LMS Algorithm to MSLC," ,"*IEEE Trans.Aerospace and Electronic Syst.*,vol.AES-21,no.1,pp79-91,Jan.1985.

[14] Mahesh Godavarti, Alfred O. Hero, III "Partial Update LMS Algorithms" *IEEE Trans.Signal Processing*, vol. 53, no.7, July2005.

[15] Goodwin,G.C. and K.S.Sin, Adaptive Filtering,Prediction and Control.Englewood Cliffs,NJ:Prentice- Hall,1984.