International Conference on Communication Technology and System Design 2011

Block based Partial update NLMS Algorithm for Adaptive Decision Feedback Equalization

Ch. Sumanth Kumar\textsuperscript{a}, D. Madhavi\textsuperscript{a}, N. Jyothi\textsuperscript{a}, K.V.V.S. Reddy\textsuperscript{b}, a*

\textsuperscript{a}Dept. Of ECE, GITAM Institute of Technology, GITAM University, Visakhapatnam, A.P., India
\textsuperscript{b}Dept. Of ECE, AU College of Engineering, Andhra University, Visakhapatnam, A.P., India

Abstract

Decision feedback equalizers are commonly employed to reduce the intersymbol interference that is caused by the time dispersive channel. In this paper a block based partial update normalized LMS algorithm is proposed, which significantly reduces the computational complexity over the other LMS based algorithms. The important characteristic of this algorithm is that only a part of the filter coefficients are updated in every iteration. 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 good performance characteristic in terms of convergence rate.

Keywords: Adaptive filtering; Mean Square error (MSE); Normalized least mean square (NLMS) algorithm.

1. Introduction

Modern digital communication techniques are used in many applications where an analog communication technique is previous used. This is because digital signals are more robust than analog signals in noise rejection and nonlinear immunity. Moreover, digital circuits have less dependence on process variations than analog circuits. At the same time, with the improvement of VLSI techniques and digital signal processing (DSP) techniques, digitally broadcasting technologies such as High Definition TV (HDTV), digital cable television systems (CATV) and satellite broadcasting, etc are able to be serve. It can provide not only multi-channel high quality video, but also variety of service, such as video conference,
interactive education and home shopping. Due to information and internet transmission and service are develop very quickly, both wide bandwidth network and high-speed digital modem are necessary in digital communication. In the high-speed digital transmission system, the noise which comes from subscriber line, coaxial cable or air may cause serious distortions on signal amplitude and phase. Since the subscriber line have different lengths, wire gauges, bridged taps, and its characteristics may change with temperature very slowly, signal transmit in multi-path propagation result in signal distortion. This is so called intersymbol interference (ISI). To compensate this distortion Adaptive Decision Feedback Equalizer (ADFE) is used. It consists of a feed forward filter (FFF) and a feedback filter (FBF). The FFF, working directly on the received data, tries to equalize the anticausal part of the channel impulse response. The residual ISI at the FFF output is then cancelled by passing the past decisions through an appropriately designed FBF and subtracting the FBF output from the FFF output. Both the FFF and the FBF coefficients are trained by some appropriate adaptive LMS algorithm. The overall structure of ADFE exhibits less noise enhancement compared with linear equalizers.

The normalized version of the LMS algorithm is known as the NLMS algorithm in which the correlation between the error and the reference input is normalized by a factor equal to the squared norm of the reference input signal vector. The advantage of the NLMS algorithm is that the step size can be chosen independent of the reference input signal power and the number of the tap weights, and it improves convergence speed in a non-static environment.

The variations of this LMS algorithm such as partial update block based normalized LMS algorithm greatly simplifies the complexity of the structure with good convergence rate.

In Section 2 the new normalized partial update block LMS based adaptive formulation is derived. In Section 3 some typical simulation results are discussed.

2. Proposed Implementation

Let us first formulate the conventional LMS algorithm as follows:

\[ y(n) = w^T(n)x(n) \]
\[ e(n) = d(n) - y(n) \]
\[ W(n + 1) = W(n) + \mu e(n)x(n) \]

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

\[ W(n) = [w_1(n), w_2(n), \ldots, w_L(n)]^T \]

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

\[ X(n) = [x(n), x(n-1), \ldots, x(n-L+1)]^T \]

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. \( \mu \) is the step size. By applying this conventional LMS algorithm to the ADFE which is shown in Fig.1,
The reformulated equations are,
\[ \hat{y}(n) = Q[y(n)] \]  \hspace{1cm} (6)
\[ y(n) = W^f(n)\phi(n) \]  \hspace{1cm} (7)
\[ W(n) = [W^f(n)W^b(n)]^T \]  \hspace{1cm} (8)
\[ \phi(n) = [x(n),.....x(n-p+1),v(n-1),....v(n-q)]^T \]  \hspace{1cm} (9)
where \(Q[.]\) represents quantization,
\[ W^f(n) = [w_1^f(n),w_2^f(n),....w_p^f(n)]^T \]  \hspace{1cm} (10)
is a pth order feed forward filter (FFF) and
\[ W^b(n) = [w_1^b(n),w_2^b(n),....w_q^b(n)]^T \]  \hspace{1cm} (11)
is a qth 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
\[ W(n+1) = W(n) + \mu \phi(n)e(n) \]  \hspace{1cm} (12)
where, \(e(n) = v(n) - \hat{y}(n)\) is the output error at index \(n\) and \(\mu\) is an appropriate step size.

To reduce the computational complexity, the partial updating the filter coefficients using LMS algorithm\[12\] is used. For the instant ‘n’, the filter coefficients are separated as even and odd indexed terms as,
\[ W_e(n) = [w_2(n), w_4(n), w_6(n), \ldots, w_L(n)]^T \]  
\[ W_o(n) = [w_1(n), w_3(n), w_5(n), \ldots, w_{L-1}(n)]^T \]  
(13)  
(14)  

The input sequence also divided as even and odd sequences as,
\[ X_e(n) = [x(n-1), x(n-3), \ldots, x(n-L+1)]^T \]  
\[ X_o(n) = [x(n), x(n-2), \ldots, x(n-L+2)]^T \]  
(15)  
(16)  

For odd \( n \), the filter coefficients updated using partial update LMS algorithm (PLMS) are given by,
\[ W_e(n+1) = W_e(n) + \mu e(n) X_e(n) \]  
\[ W_o(n+1) = W_o(n) \]  
(17)  
(18)  

For even \( n \), the filter coefficients are,
\[ W_e(n+1) = W_e(n) \]  
\[ W_o(n+1) = W_o(n) + \mu e(n) X_o(n) \]  
(19)  
(20)  

Define the coefficient error vectors as,
\[ V_e(n) = W_e(n) - W_e^{(opt)} \]  
\[ V_o(n) = W_o(n) - W_o^{(opt)} \]  
\[ V(n) = W(n) - W^{(opt)} \]  
\[ V^{\infty}(n) = [V_e(n), V_o(n)]^T \]  
(21)  
(22)  
(23)  
(24)  

Where \( W_e^{(opt)} = [w_2^{(opt)}, w_4^{(opt)}, w_6^{(opt)}, \ldots, w_L^{(opt)}] \)  
\( W_o^{(opt)} = [w_1^{(opt)}, w_3^{(opt)}, w_5^{(opt)}, \ldots, w_{L-1}^{(opt)}] \)  
\( W^{(opt)} = [w_1^{(opt)}, w_3^{(opt)}, w_5^{(opt)}, \ldots, w_L^{(opt)}] \)  
(25)  
(26)  
(27)  

For regular LMS algorithm, the recursion for mean coefficient error vector \( E[v(n)] \) is given by,
\[ E[v(n+1)] = (I - \mu R) E[v(n)] \]  
(28)  

Where \( I \) is an \( N \) dimensional identity matrix, and \( R = E[x(n)x'(n)] \) 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 \( \lambda_{\max} \) is the maximum eigen value of the input signal correlation matrix \( R \).

For odd \( n \)
\[ E[v(n+2)] = (I - \mu I_z R)(I - \mu I R) E[v(n)] \]  
(29)  

For even \( n \)
\[ E[v(n+2)] = (I - \mu I R)(I - \mu I_z R) E[v(n)] \]  
(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,

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

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

### 2.1. Implementation of Normalized Partial update block LMS ADFE Structure

Consider a signed LMS based adaptive filter that processes an input signal \( x(n) \) and generates the output \( y(n) \).

The weight update equation in the normalized LMS algorithm can be modified as,

\[
\hat{W}(n + 1) = \hat{W}(n) + \frac{\mu}{\|x(n)\|^2} x(n)e(n)
\]

Where \( \|x(n)\|^2 = x'(n)x(n) \) is the norm of the input signal, which eliminates the problem of gradient noise amplification.

Here the normalization by \( \|x(n)\|^2 \) does not change the direction of the estimated gradient vector, it alters only the magnitude. Therefore with statistical assumptions it may be shown that the proposed algorithm converges for the adaptation constant \( \mu \) is with in the range 0 to 2. Further in the traditional LMS algorithm, the correction applied to the filter weight vector is proportional to the input vector \( x(n) \). Therefore, when \( x(n) \) is large, it experiences a problem with gradient noise amplification. Where as in NLMS algorithm, normalization by \( \|x(n)\|^2 \), diminishes the noise amplification problem. Further sometimes if \( \|x(n)\|^2 \) is too small, similar type of problem that occur, hence a small positive number \( \varepsilon \) is added to the normalization term. The additional computational term in the NLMS algorithm, however is computed recursively as,

\[
\|x(n + 1)\|^2 = \|x(n)\|^2 + \|x(n + 1)\|^2 - \|x(n - p)\|^2,
\]

where \( p \) is the order of the filter. Therefore the extra computation involves only two squaring operations, one addition and one subtraction.

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

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

$$X(jp+r)=[x(jp+r),x(jp+r-1),\ldots,x(jp+r-L+1)]^T$$  \hspace{1cm} (34)

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

$$e(jp + r) = d(jp + r) - y(jp + r)$$  \hspace{1cm} (35)

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

$$y(jp + r) = W'(j)X(jp + r)$$  \hspace{1cm} (36)

The parameter $\mu$ popularly called the step size parameter is to be chosen as

$$0 < \mu < \frac{2}{\text{trace } R},$$

for convergence of the algorithm.

However, block-adaptive DFE implementation is not a straightforward task, 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 previous instants 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 [2,6]. 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, and offers 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.

To reduce the sensitivity of the LMS convergence speed on the input signal power, gain control has been suggested. This can be realized by dividing the input data samples by an amount proportional to the estimated mean signal power, thereby adjusting the input signal power to within some predetermined range. However, the estimation of the received signal power, from the input data samples, imposes a delay before equalization takes place and can be unresponsive to rapid variations in the input signal statistics. For these reason, the NLMS algorithm has been adapted which is able to achieve the same effect as AGC. In this algorithm the step size is adjusted in proportion to the inverse of the sum of the squares of the input signal to the adaptive filter. The coefficient update for the NLMS algorithm with variable step size is,
The scaling factor $\mu$ controls the speed of the adjustment and is chosen in proportion to the length of the FFF and FBF’s. It will be shown below that $\mu(n)$ can be computed order recursively in the same manner as error signal. Note that during the startup, when the FFF is not completely full of input data samples, the variable step size will be undetermined and, hence, must be initialized to some fixed value during this period. The coefficient updates must, however, be modified to account for the time-varying step size. For the FFF, the $i^{th}$ coefficient update is given by,

$$w_f^i(n+1) = w_f^i(n) + \mu(n-i)x_f(n-i)e(n-i)$$

The both feedforward and feedback filter coefficients are trained by the weight update equations of normalized block based partial update LMS based 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,

$$y^f(n) = w^f(n)x(n)$$

where

$$W^f(n) = [w^f_0(n),...,w^f_p(n)]^T$$

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

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

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

$$y(n) = y^f(n) + y^b(n)$$

| Table I |
| Sketch of The Proposed Algorithm |
| 1) Initially transmit the known sequence. |
| 2) Assume, initially both the FFF, FBF weights to be zero. |
| 3) Find the output vector, which is the sum of the outputs of FFF, FBF. |
| 4) Estimate the tap weight vector at each instant of time using normalized modified block LMS algorithm. |
| 5) Update the filter coefficients. |
3. Simulation Results

Fig. 2 shows the PSK modulated signal generated by randin(100:1) function having a input code word length of 2 bits for every input data bit. Fig. 3 shows the signal at the receiver which is corrupted with a Gaussian noise in the channel. Fig. 4 shows the output of the ADFE. It is observed that the output signal is almost equal to the input signal.

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 simulating the algorithm.

References