EFFICIENT BLOCK IMPLEMENTATION OF THE LMS-BASED DFE

Kostas Berberidis * , *Thanasis Rontogiannis* † , *Sergios Theodoridis* †

* Computer Technology Institute (C.T.I.), P.O. Box 1122, 26110 Patras, GREECE
E-mail: berberid@cti.gr

† Dept. of Informatics, Univ. of Athens, TYPA Buildings, 15771 Athens, GREECE
E-mails: tronto@di.uoa.gr , stheodor@di.uoa.gr

ABSTRACT

In this paper an efficient block adaptive implementation of the widely used Decision Feedback Equalizer is developed. Both the feedforward and the feedback filters of the equalizer are updated once every K sample time intervals, with K being the block length. However this block adaptation is done in such a way that the resulting filters, as well as the decisions, are identical to those computed by the conventional sample by sample LMS based DFE (LMS-DFE). That is, the new block adaptive algorithm is mathematically equivalent to the LMS-DFE algorithm (and hence it features identical performance). At the same time the proposed algorithm offers substantial computational savings as compared to the sample by sample LMS-DFE. Thus the new block DFE turns out to be particularly suitable for applications in which long equalizers are required.

1 INTRODUCTION

A major cause of performance degradation in data transmission through dispersive communication channels is the introduced Intersymbol Interference (ISI). One of the most successful techniques used in practice in order to drastically reduce the ISI is the well known Decision Feedback Equalizer (DFE). However in a number of applications the computational complexity of the adaptive equalization part may be prohibitive. The excessive burden is mainly due to the long feedback part of the DFE, which is imposed by the nature of the problem. Indeed in many applications the tail portion (postcursor part) of the impulse response is much longer than the front portion (precursor part) and may last from almost one hundred up to several hundreds of baud intervals. Typical applications of the kind are high speed digital transmission over the digital subscriber loop [1], microwave communications via line-of-sight links [2] and digital TV terrestrial broadcasting [3].

A possible way to reduce complexity would be to develop a block adaptive filtering formulation of the conventional sample-by-sample DFE. However in order to obtain the decision symbol at a given time n the respective decisions at times $n-1, n-2, \ldots, n-N$ are required (where N is the length of the feedback filter). Therefore, it is not possible to obtain more than one decisions at a time. A block solution to the problem, implemented in the frequency domain, has recently been presented in [5]. However this technique, in order to retain its efficiency, imposes certain restrictions on the block length with respect to the lengths of the Feedforward (FF) and the Feedback (FB) filters of the DFE. These restrictions may be undesirable in some applications where we need more flexibility in the choice of the block length (and the processing delay accordingly).

A block adaptive filtering technique, suitable only for linear equalizers, was recently introduced in [7]. The unknown filter coefficients are updated in a manner which is mathematically equivalent to the conventional LMS and for this reason the new technique is called Fast Exact LMS (FELMS). Reduction of the computational complexity of the FELMS algorithm is achieved due to the application of a fast FIR filtering scheme that was originally described in [6]. It should be noted that the block size can be much smaller than the filter's order and therefore a relatively small processing delay is introduced. However this technique cannot be applied directly to a DFE structure since future decisions required by the blocks are not readily available.

In this paper, motivated by the algorithm in [7], we develop a new block DFE which is mathematically equivalent to the conventional LMS-based sample-by-sample DFE but with considerably reduced computational load. In order to compute efficiently the decisions at a given block we properly decompose the FB part and perform the internal computations in a specified manner. The derivation steps of the new algorithm are described in Section 2 while in Section 3 computational issues are discussed. Finally, in Section 4 the simulation results of a typical channel equalization experiment are given.

2 THE FE-DFE ALGORITHM

The new block DFE, called hereafter Fast Exact DFE (FE-DFE), consists of a filtering and an updating part. In the filtering part the decisions corresponding to the current block are produced, and in the updating part new estimates of the FF and FB filters are computed based on the respective estimates K time instants before. Although computed en block all the above quantities are identical to those obtained by the conventional sample by sample LMS-DFE, that is, the two algorithms are mathematically equivalent.

To start our derivation let us first formulate the conventional sample-by-sample LMS-based DFE algorithm as below.

$$y(k) = \mathbf{a}_M^T(k)\mathbf{x}_M(k+M-1) + \mathbf{b}_N^T(k)\mathbf{d}_N(k-1) \quad (1)$$

$$d(k) = f\{y(k)\} \quad (2)$$

$$e(k) = d(k) - y(k) \quad (3)$$

$$\mathbf{a}_M(k+1) = \mathbf{a}_M(k) + \mu^a \mathbf{x}_M(k+M-1)e(k) \quad (4)$$

$$\mathbf{b}_N(k+1) = \mathbf{b}_N(k) + \mu^b \mathbf{d}_N(k-1)e(k) \quad (5)$$

where $\mathbf{x}_M(k+M-1) = [x(k+M-1) \ \ldots \ x(k)]^T$ consists of M input samples and $\mathbf{d}_N(k-1) = [d(k-1) \ldots d(k-N)]^T$ consists of N decision samples. Vectors $\mathbf{a}_M(k)$ and $\mathbf{b}_N(k)$ denote the M-th order FF and the N-th order FB filter respectively. $f\{\cdot\}$ in Eq. (2) stands for the decision device function.

Given the estimates of the FF and FB filters at time $n-K+1$ the aim is to compute the next K decisions as well as the new filter estimates at time $n+1$. The two parts of the new DFE are described below.

Filtering part

Successive application of Eqs. (4),(5) and a proper combination of the resulting formulae leads to the following expressions

$$\begin{aligned} \mathbf{a}_M(n-l+1) &= \mathbf{a}_M(n-K+1) \\ &+ \mu^a X(n;l)\tilde{\mathbf{e}}_{K-l}(n-l) \quad (6) \end{aligned}$$

$$\begin{aligned} \mathbf{b}_N(n-l+1) &= \mathbf{b}_N(n-K+1) \\ &+ \mu^b D(n;l)\tilde{\mathbf{e}}_{K-l}(n-l) \quad (7) \end{aligned}$$

where $X(n;l) = [\mathbf{x}_M(n+M-K) \cdots \mathbf{x}_M(n+M-l-1)]$ is an $M \times (K-l)$ matrix and $D(n;l) = [\mathbf{d}_N(n-K) \cdots \mathbf{d}_N(n-l-1)]$ is an $N \times (K-l)$ matrix. Vector $\tilde{\mathbf{e}}_{K-l}(n-l) = [e(n-K+1) \ \ldots \ e(n-l)]^T$ consists of the respective filtering errors in reverse order. The above expressions are derived for $l = K-1, \ldots, 1$.

If we now substitute Eqs. (1) and (6),(7) to Eq. (3) and group together the resulting error formulae, we end up with the following representation

$$\begin{aligned} \tilde{\mathbf{e}}_K(n) = \overbrace{\tilde{\mathbf{d}}_K(n)}^{A} &- \overbrace{X^T(n;0)\mathbf{a}_M(n-K+1)}^{B} \\ &- \overbrace{D^T(n;0)\mathbf{b}_N(n-K+1)}^{C} \\ &- \overbrace{(S_x(n) + S_d(n))\,\tilde{\mathbf{e}}_K(n)}^{D} \quad (8) \end{aligned}$$

where $\tilde{\mathbf{d}}_K(n) = [d(n-K+1) \ \ldots \ d(n)]^T$ and $S_x(n), S_d(n)$ are lower triangular matrices with zero diagonal elements. Their nonzero elements are given by

$$\{S_x(n)\}_{ij} = \\ \mu^a \mathbf{x}_M^T(n+M-K+i-1)\mathbf{x}_M(n+M-K+j-1)$$

$$\{S_d(n)\}_{ij} = \mu^b \mathbf{d}_N^T(n-K+i-1)\mathbf{d}_N(n-K+j-1)$$

for $i = 2, 3, \ldots, K$ and $j = 1, 2, \ldots, i-1$. The matrix-by-vector product of term B in (8) can be viewed as an FIR filtering problem. Therefore, having collected K new channel output samples (i.e. $x(n+M-K), x(n+M-K+1), \ldots, x(n+M-1)$) the fast FIR scheme of [6] can be applied. However, this is not the case for the term C since this involves future decision samples. Specifically, at time $n-K+1$, the first $j-1$ elements of vector $\mathbf{d}_N(n-j)$ are unknown, for $j = K, K-1, \ldots, 1$ (which means that only vector $\mathbf{d}_N(n-K)$ is completely known). Term D cannot be computed as well because matrix $S_d(n)$ involves future decision samples. Finally, term A is completely unknown.

To see how to overcome this problem, let us take as an example the case $K = 4$. By properly rearranging the involved elements, term C can be written as

$$\begin{bmatrix} \boldsymbol{d}_3^T & \boldsymbol{d}_4^T & \boldsymbol{d}_5^T & \boldsymbol{d}_6^T \\ \boldsymbol{d}_2^T & \boldsymbol{d}_3^T & \boldsymbol{d}_4^T & \boldsymbol{d}_5^T \\ \boldsymbol{d}_1^T & \boldsymbol{d}_2^T & \boldsymbol{d}_3^T & \boldsymbol{d}_4^T \\ \boldsymbol{d}_0^T & \boldsymbol{d}_1^T & \boldsymbol{d}_2^T & \boldsymbol{d}_3^T \end{bmatrix} \begin{bmatrix} \boldsymbol{b}_0 \\ \boldsymbol{b}_1 \\ \boldsymbol{b}_2 \\ \boldsymbol{b}_3 \end{bmatrix} \quad (9)$$

where $\boldsymbol{d}_i$ and $\boldsymbol{b}_i$ are the corresponding polyphase components of the decision vectors and the FB filter respectively, that is $\boldsymbol{d}_i = [d(n-i-1) \ \cdots \ d(n-i-1-4j) \ \cdots \ d(n-i-N+3)]^T$ for $i = 0, \ldots, 6$, and $j = 0, 1, \ldots, (N/4)-1$ and $\boldsymbol{b}_i = [b_i \, b_{i+4} \cdots b_{i+4j} \cdots b_{i+N-4}]^T(n-3)$ for $i = 0, 1, 2, 3$. Note that the polyphase vectors are denoted as bold slanted with a subscript indicating their polyphase index. Following the scheme suggested in [6] the above matrix-vector multiplication takes the form

$$\begin{bmatrix} \boldsymbol{d}_3^T(\boldsymbol{b}_0 + \boldsymbol{b}_1 + \boldsymbol{b}_2 + \boldsymbol{b}_3) + (\boldsymbol{d}_4 - \boldsymbol{d}_3)^T(\boldsymbol{b}_1 + \boldsymbol{b}_3) \\ \quad + (\boldsymbol{d}_5 - \boldsymbol{d}_3)^T(\boldsymbol{b}_2 + \boldsymbol{b}_3) \\ \qquad + [(\boldsymbol{d}_6 - \boldsymbol{d}_4) - (\boldsymbol{d}_5 - \boldsymbol{d}_3)]^T \boldsymbol{b}_3 \\ \boldsymbol{d}_3^T(\boldsymbol{b}_0 + \boldsymbol{b}_1 + \boldsymbol{b}_2 + \boldsymbol{b}_3) - (\boldsymbol{d}_3 - \boldsymbol{d}_2)^T(\boldsymbol{b}_0 + \boldsymbol{b}_2) \\ \quad + (\boldsymbol{d}_5 - \boldsymbol{d}_3)^T(\boldsymbol{b}_2 + \boldsymbol{b}_3) \\ \qquad - [(\boldsymbol{d}_5 - \boldsymbol{d}_3) - (\boldsymbol{d}_4 - \boldsymbol{d}_2)]^T \boldsymbol{b}_2 \\ \boldsymbol{d}_3^T(\boldsymbol{b}_0 + \boldsymbol{b}_1 + \boldsymbol{b}_2 + \boldsymbol{b}_3) + (\boldsymbol{d}_4 - \boldsymbol{d}_3)^T(\boldsymbol{b}_1 + \boldsymbol{b}_3) \\ \quad + (\boldsymbol{d}_3 - \boldsymbol{d}_1)^T(\boldsymbol{b}_0 + \boldsymbol{b}_1) \\ \qquad - [(\boldsymbol{d}_4 - \boldsymbol{d}_2) - (\boldsymbol{d}_3 - \boldsymbol{d}_1)]^T \boldsymbol{b}_1 \\ \boldsymbol{d}_3^T(\boldsymbol{b}_0 + \boldsymbol{b}_1 + \boldsymbol{b}_2 + \boldsymbol{b}_3) - (\boldsymbol{d}_3 - \boldsymbol{d}_2)^T(\boldsymbol{b}_0 + \boldsymbol{b}_2) \\ \quad + (\boldsymbol{d}_3 - \boldsymbol{d}_1)^T(\boldsymbol{b}_0 + \boldsymbol{b}_1) \\ \qquad + [(\boldsymbol{d}_3 - \boldsymbol{d}_1) - (\boldsymbol{d}_2 - \boldsymbol{d}_0)]^T \boldsymbol{b}_0 \end{bmatrix} \quad (10)$$

From the first row of the matrix in (10) it is readily observed that for the computation of the first element of term C only the polyphase components d_3, d_4, d_5, d_6 of $\mathbf{d}_N(N-4)$ are required, which are already known. The first element of term B is calculated in a similar manner and the sum of these two elements, say $y(n-3)$, provides the next decision sample as $d(n-3) = f\{y(n-3)\}$. Note that in this first step of the recursive procedure there is no contribution from term D, due to the structure of matrix $S_x(n) + S_d(n)$. The error term $e(n-3)$ is obtained as $e(n-3) = d(n-3) - y(n-3)$.

Having calculated $d(n-3)$ (which is also the first element of $\mathbf{d}_2$), the second element of term C can be computed as in (10). Moreover, the unknown (2,1)-element of matrix $S_d(n)$ can now be obtained. In the sequel, $d(n-2)$ and $e(n-2)$, to be used in the next step, can be computed (note that from now on there is a contribution from term D). Proceeding in a similar manner, all the errors can be obtained.

It must be noted that having calculated a particular decision sample, all the elements of the respective row of $S_d(n)$ can subsequently be computed. Also, due to the lower triangular structure of matrix $S_d(n) + S_x(n)$ only the errors that have already been calculated in previous steps are involved in the computation of elements of the term D. Finally, due to the row by row computation of matrix $S_d(n)$, the elements of each row can be recursively obtained in a way similar to that also used for the corresponding elements of $S_x(n)$, as proposed in [7].

Recall that the above procedure was shown for $K = 4$; however it can be easily generalized for any block length.

Updating part

In the updating part of the FE-DFE the FF and FB filter estimates are computed from their respective values K time instants before. Indeed, if we write equations (6) and (7) for $l = 0$, we get

$$\mathbf{a}_M(n+1) = \mathbf{a}_M(n-K+1) + \mu^a X(n;0)\tilde{\mathbf{e}}_K(n) \quad (11)$$

$$\mathbf{b}_N(n+1) = \mathbf{b}_N(n-K+1) + \mu^b D(n;0)\tilde{\mathbf{e}}_K(n) \quad (12)$$

Having calculated the decisions $d(n-K+1), \ldots, d(n)$ as described above, the fast scheme of [6] can be directly applied to (11) and (12) leading to a reduction in the number of operations required.

3 COMPUTATIONAL COMPLEXITY

Due to lack of space a detailed analysis of the computational complexity of the proposed scheme will not be provided. Assuming that K is a power of two (i.e. $K = 2^m$) it can be shown that the multiplications and additions required by FE-DFE are given by

$$M_d = 2(\frac{3}{4})^m(M+N) + \frac{5 \cdot 2^m - 9}{2}$$

$$A_d = 2[2(\frac{3}{2})^m - 1](M+N)/2^m + (4 \cdot 2^m - 11) + 7(\frac{3}{2})^m$$

where M_d is the number of multiplications per decision and A_d is the corresponding number of additions. The "optimum" m which provides the smallest number of multiplications is given by

$$m_{opt} = -1.125 + 0.7\log_2(M+N) \quad (13)$$

In Table 1, the number of operations required by the

Parameters			LMS-DFE		FE-DFE	
M	N	K	Mults.	Adds.	Mults.	Adds.
16	64	8	160	160	82	160
32	128	16	320	320	137	271
32	256	32	576	576	212	425
64	512	32	1152	1152	349	681
128	1024	64	2304	2304	565	1109

Table 1: Comparison of Computational Complexities (multiplications and additions required per decision)

FE-DFE is compared with that of the classical LMS-DFE, for different values of M,N and for m chosen according to (13). We observe that, using FE-DFE, a significant reduction in computational complexity is achieved. As it it will be verified experimentally in next section, the substantial improvement in complexity is obtained with no loss in performance.

4 EXPERIMENTAL RESULTS

In order to verify the correctness of the proposed scheme, a typical experiment in the context of multipath echo cancellation in digital TV terrestial transmission was carried out. The impulse response of the multipath channel consisted of 6 echoes with amplitudes -10dB,-7dB,-6dB,-10dB,-14dB,-4.5dB and delays $-20T_s$,$-12T_s$,$10T_s$,$40T_s$,$70T_s$,$140T_s$ respectively where T_s is the symbol time interval. The echo phases were chosen randomly. Notice the existence of a very strong far echo of -4.5dB. The impulse response of the channel used in our experiment is shown in Figure 1. The input to the channel is taken from a binary alphabet (± 1), while its output is corrupted by additive white Gaussian noise. The variance of the noise is such that the channel output SNR equals 15dB.

In Figure 2 two mean squared error curves are depicted, although they are not distinguished. One corresponds to the LMS-DFE and the other to the proposed FE-DFE. The orders of the FF and FB filters were taken equal to 32 and 256 respectively and the step size parameters were chosen equal to 0.0015. In the case of the FE-DFE, the block size K is 32, chosen according to (13) (see also Table 1). The experiment verifies that the performance of the proposed algorithm is identical to that of the LMS-type scheme as expected from the analysis of the previous section.

5 CONCLUSION

In this paper a block adaptive version of the LMS based Decision Feedback Equalizer was derived. The new algorithm is mathematically equivalent to its sample-by-sample counterpart offering at the same time considerable computational savings. Due to its computational efficiency the new algorithm is particularly attractive in many practical cases and especially in applications which involve very long equalizers.

References

[1] P. Crespo, M.L. Honig, "Pole-zero decision feedback equalization for the digital subscriber loop", *Proc. of the 1990 IEEE International Conference on Communications*, pp. 1166-1171, 1990.

[2] M.C.S. Young, P.M. Grant, C.F.N. Cowan, "Block LMS adaptive equalizer design for digital radio", *Proc. of the EUSIPCO-88 conference*, pp. 1349-1353, 1988.

[3] W.F. Schreiber, "Advanced television systems for terrestrial broadcasting: some problems and some proposed solutions", *Proceedings of the IEEE*, vol. 83, no. 6, pp. 958-981, June 1995.

[4] E.R. Ferrara, "Frequency-domain adaptive filtering", in *Adaptive Filters*, C.F.N. Cowan and P.M. Grant, Eds., Englewood Cliffs, Prentice Hall, 1985, Ch. 6, pp. 145-179.

[5] K. Berberidis, J. Palicot, "A frequency-domain decision feedback equalizer for multipath echo cancellation", *Proc. of the 1995 IEEE CLOBECOM Conference*, Singapore, Nov. 1995.

[6] Z.J. Mou, P. Duhamel, "Fast FIR filtering: algorithms and implementation", *Signal Processing*, vol. 13, pp. 377-384, Dec. 1987.

[7] J. Benesty, P. Duhamel, "A fast exact least mean square adaptive algorithm", *IEEE Trans. on Signal Processing*, vol. 40, pp. 2904-2920, Dec. 1992.

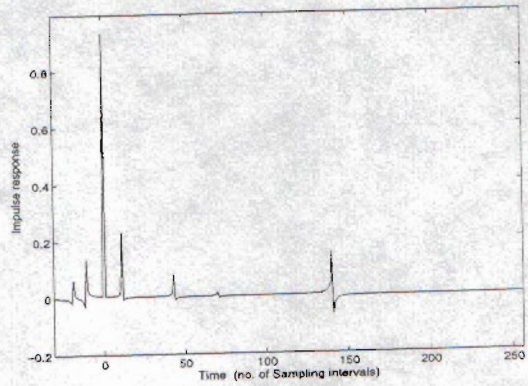

Figure 1: The impulse response of the multipath echo channel.

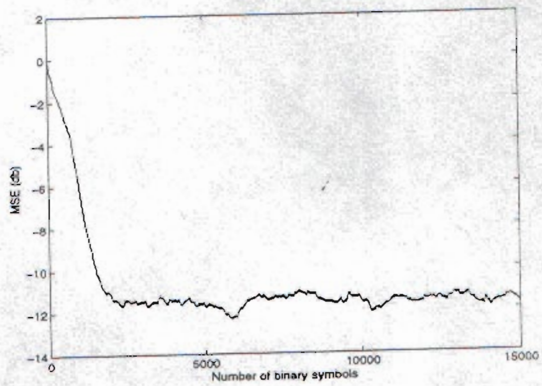

Figure 2: Mean squared error curves.