Efficient Block Implementation of the Decision Feedback Equalizer

Kostas Berberidis, *Member, IEEE,* Thanasis A. Rontogiannis, and Sergios Theodoridis, *Member, IEEE*

***Abstract*—In this letter, a new block adaptive decision feedback equalizer is developed. Both the feedforward (FF) and the feedback (FB) filters are updated once every K sample time intervals, with K being the block length. It should be noted that this block adaptation is done in such a way that the resulting filters, and the decisions as well, are identical to those computed by the conventional sample by sample LMS-based decision feedback equalizer (LMS-DFE). The new algorithm offers substantial computational savings as compared to the sample-by-sample LMS-DFE with no loss in performance. The new block DFE turns out to be particularly suitable for applications requiring long equalizers.**

***Index Terms*—Adaptive filters, channel equalization.**

I. Introduction

In dispersive communication channels, introduced intersymbol interference (ISI) constitutes a major cause of performance degradation. The introduced ISI can be drastically reduced by using 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. 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 $n-1, n-2, \cdots, n-N$ are required (where N is the length of the feedback filter). Therefore, it is not possible to obtain more than one decision 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 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 that 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 finite impulse response (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 since future decisions required by the blocks are not readily available.

Motivated by the algorithm in [7], in this work we develop a new block DFE that 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, as described in the next section.

II. Derivation of the New Block DFE

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. Note that all the above quantities are identical to those obtained by the conventional sample by sample LMS-DFE. In other words, FE-DFE is mathematically equivalent to LMS-DFE.

To start our derivation, let us first formulate the conventional sample-by-sample LMS-based DFE algorithm as follows:

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

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

$$e(k) = d(k) - y(k) \tag{3}$$

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

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

where $\mathbf{x}_M(k+M-1) = [x(k+M-1)\cdots x(k)]^T$ consists of input samples and $\mathbf{d}_N(k-1) = [d(k-1)\cdots d(k-N)]^T$ consists of decision samples. Vectors $\mathbf{a}_M(k)$ and $\mathbf{b}_N(k)$ denote

Manuscript received February 19, 1997. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. G. W. Wornell.

K. Berberidis is with the Department of Computer Engineering and Informatics, University of Patras, 26500 Patras, Greece (e-mail: berberid@cti.gr).

T. A. Rontogiannis and S. Theodoridis are with the Department of Informatics, Division of Communications and Signal Processing, Panepistimioupolis, University of Athens, 157 71 Zografou, Greece.

Publisher Item Identifier S 1070-9908(98)04342-9.

the Mth order FF and the Nth order FB filter, respectively. $f\{\cdot\}$ in (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.

A. Filtering Part

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

$$\begin{aligned}\mathbf{a}_M(n-l+1) &= \mathbf{a}_M(n-K+1) \\ &\quad + \mu^a X(n;l)\tilde{\mathbf{e}}_{K-l}(n-l) \qquad (6)\\ \mathbf{b}_N(n-l+1) &= \mathbf{b}_N(n-K+1) \\ &\quad + \mu^b D(n;l)\tilde{\mathbf{e}}_{K-l}(n-l) \qquad (7)\end{aligned}$$

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 $\tilde{\mathbf{e}}_{K-l}(n-l) = [e(n-K+1) \cdots e(n-l)]^T$ consists of the respective filtering errors in reverse order. The above expressions are derived for $l = K-1, \cdots, 1$.

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

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

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

$$\begin{aligned}\{S_x(n)\}_{ij} &= \mu^a \mathbf{x}_M^T(n+M-K+i-1) \\ &\quad \cdot \mathbf{x}_M(n+M-K+j-1) \\ \{S_d(n)\}_{ij} &= \mu^b \mathbf{d}_N^T(n-K+i-1)\mathbf{d}_N(n-K+j-1)\end{aligned}$$

for $i = 2, 3, \cdots, K$ and $j = 1, 2, \cdots, i-1$. 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., $x(n+M-K)$, $x(n+M-K+1), \cdots, x(n+M-1)$], 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 $j = K, K-1, \cdots, 1$ [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

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

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 $i = 0, \cdots, 6$, and $j = 0, 1, \cdots, (N/4)-1$ 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 of (10), shown at the bottom of the page. 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 $\boldsymbol{d}_3$, $\boldsymbol{d}_4$, $\boldsymbol{d}_5$, $\boldsymbol{d}_6$ 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 $\boldsymbol{d}_2$), 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].

The above procedure, shown for $K = 4$, can be easily generalized for any K.

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

B. 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 (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 $d(n-K+1), \cdots, d(n)$ 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.

III. Computational Complexity

Due to lack of space, a detailed analysis of the computational complexity of the proposed scheme will not be provided. The operations required per decision is as follows ($K = 2^m$):

$$2\left(\frac{3}{4}\right)^m (M+N) + \frac{5\cdot 2^m - 9}{2} \quad \text{multiplications/decision}$$

$$2\left[2\left(\tfrac{3}{2}\right)^m - 1\right](M+N)/2^m + (4\cdot 2^m - 11) + 7\left(\tfrac{3}{2}\right)^m \quad \text{additions/decision.}$$

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 I, the number of operations required by the 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 the number of operations is achieved.

It has been verified through typical channel equalization experiments that the reduction in complexity is obtained with practically no loss in performance compared to the LMS-DFE. Indeed, as expected, the MSE curve of the FE-DFE coincides with the corresponding one of the LMS-DFE. Of course, in a real implementation, there may be small differences due to finite precision effects. For a discussion of the numerical properties of "fast exact" type block algorithms, we refer to [7].

TABLE I
Comparison of Required Operations

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

IV. Conclusion

We derived a new block DFE that is mathematically equivalent to the sample-by-sample LMS-DFE. Due to its computational efficiency, the new algorithm is particularly attractive in many practical cases and especially in applications that involve very long equalizers.

References

[1] P. Crespo and M. L. Honig, "Pole-zero decision feedback equalization for the digital subscriber loop," in *Proc. 1990 IEEE Int. Conf. Commun.,* 1990, pp. 1166–1171.

[2] M. C. S. Young, P. M. Grant, and C. F. N. Cowan, "Block LMS adaptive equalizer design for digital radio," in *Proc. EUSIPCO'88 Conf.,* pp. 1349–1353.

[3] W. F. Schreiber, "Advanced television systems for terrestrial broadcasting: Some problems and some proposed solutions," in *Proc. IEEE,* vol. 83, 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, NJ: Prentice-Hall, 1985, ch. 6, pp. 145–179.

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

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

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