Time- and Frequency-Domain Block LMS Adaptive Digital Filters: Part I - Realization Structures

시간영역 및 주파수영역 블럭적응 여파기에 관한 연구 : 제 1 부 - 구현방법

Jae Chon Lee*, Chong Kwan Un*

이 재 천, 은 종 관

ABSTRACT

In this work we study extensively the structures and performance characteristics of the block least mean-square (BLMS) adaptive digital filters (ADF's) that can be realized efficiently using the fast Fourier transform (FFT). The weights of a BLMS ADF realized using the FFT can be adjusted either in the time domain or in the frequency domain, leading to the time-domain BLMS (TBLMS) algorithm or the frequency-domain BLMS (FBLMS) algorithm, respectively. In Part I of the paper, we first present new results on the overlap-add realization and the number-theoretic transform realization of the FBLMS ADF's. Then, we study how we can incorporate the concept of different frequency-weighting on the error signals and the self-orthogonalization of weight adjustment in the FBLMS ADF's, and also in the TBLMS ADF's. As a result, we show that the TBLMS ADF can also be made to have the same fast convergence speed as that of the self-orthogonalizing FBLMS ADF. Next, based on the properties of the sectioning operations in weight adjustment, we discuss unconstrained FBLMS algorithms that can reduce two FFT operations both for the overlap-save and overlap-add realizations. Finally, we investigate by computer simulation the effects of different parameter values and different algorithms on the convergence behaviors of the FBLMS and TBLMS ADF's. In Part II of the paper, we will analyze the convergence characteristics of the TBLMS and FBLMS ADF's.

* Communications Research Laboratory Korea Advanced Institute of Science and Technology P.O. Box 150, Chongyangni
서울 청량리사서함150,
한국과학기술원 통신공학연구실

요 약

블럭적응 여파기는 구현시에 고속푸리에변환 기법을 이용하면 계산량을 대폭 줄일 수 있음이 밝혀져서 연구자들의 관심을 끌어 왔다. 본 논문은 2편으로 구성되어 있는데, 제1부에서는 블럭적응 여파기의 여러가지 구현방법을 연구하고 제2부에서는 성능분석의 결과들을 논의하고 있다. 블럭적응 여파기의 계수가 최적해를 추적하도록 하는 적응알고리즘은 시간영역 또는 주파수영역에서 동작하게 할 수 있는데 이를 각각 시간영역 및 주파수영역 블럭적응 여파기로 부른다. 특히 제1부에서는 이들 두 구조 사이의 공통점 및 대비 관계를 명확하게 할 것이다. 구체적으로, 제1부에서는 먼저 overlap-add 방식에 의한 고속 구현방법과 정수론적 변환 기법에 의한 효율적인 구현방법에 관한 새로운 결과들을 발표한다. 그리고 나서 계수 설계시에 주파수영역의 정보를 주파수대역마다 차등으로 적용하는 방법과 여파기 계수의 최적해 추정에 있어서 수렴시간 단축을 위한 self-orthogonalization 방법을 주파수영역은 물론 시간영역 블럭적응 여파기들에도 적용할 수 있음을 보인다. 다음으로 계수의 블럭적응을 위한 입출력데이터 블럭분할 방법의 성질에 근거해서 고속푸리에변환 연산을(부분적으로) 생략할 수 있는 비제약 주파수영역 블럭적응 여파기에 관해서 논의한다. 제1부의 마지막으로 여러가지 여파기 상수값들이 여파기 수렴특성에 미치는 영향을 컴퓨터 시뮬레이션을 통해서 조사한 결과를 발표한다.

I. INTRODUCTION

Since Widrow and Hoff proposed the least mean-square (LMS) algorithm [1], many researchers have studied various structures and adaptation algorithms for adaptive digital filters (ADF's). At present, applications of ADF's can be found in many diverse fields, and their application areas are being widened further with the rapid advance in the digital integrated circuit technology [1]-[6].

The performance characteristics of the LMS algorithm have been studied extensively, and thus are relatively well understood [7]. Among various structures and adaptation algorithms proposed so far, the finite impulse response (FIR) ADF using the LMS algorithm is being widely used due to its relative simplicity in realization. However, one drawback of the LMS algorithm is known to be its slow convergence speed when the input signals are highly correlated [1],[2].

So far, considerable research effort has been directed toward the realization of the FIR ADF's using the LMS-type algorithms in the frequency domain [8]-[35]. The major motivation for such frequency-domain ADF's is to develop an ADF which has improved convergence speed or is computationally more efficient in comparison to the conventional LMS ADF. In the frequency-domain adaptive filtering, the discrete Fourier transform (DFT) of the input signals provides a set of approximately uncorrelated signals, thereby making it possible to implement a self-orthogonalizing algorithm easily in the frequency domain for improving the convergence speed [20]-[23],[26],[34],[35] At the same time, by processing data on a block-by-block basis, the transversal ADF can be realized efficiently using the fast Fourier transform (FFT) and an appropriate sectioning method [13],[24],[33]. However, the computational savings cannot be expected in some of the frequency-domain ADF's that are operated on a sample-by-sample basis [14],[18],[23],[35].

One of the most significant work done in the frequency-domain adaptive filtering appears to be the work by Clark, Mitra, and Parker [12],[15],[16],[19],[24]. In [16], by introducing the concept of block Wiener filtering, they presented the basic theory and convergence properties of the FIR block LMS (BLMS) ADF that is obtained by minimizing the block mean-squared error (BMSE). As a result, they obtained a condition under which the convergence behaviors of the LMS and BLMS ADF's are the same for stationary inputs. In a subsequent paper [24], specific implementations of the BLMS ADF were discussed based on the FFT

and an appropriate sectioning method. It was shown that the weights of the BLMS ADF realized using the FFT can be adjusted either in the time domain or in the frequency domain, leading to the time-domain BLMS (TBLMS) algorithm or the frequency-domain BLMS (FBLMS) algorithm, respectively. Also, it was shown that all of the known frequency-domain ADF's in the literature are included in the realization structures derived in [24]. Another significant result is that the overlap-save implementation requires less computation for the BLMS ADF than the overlap-add implementation.

The self-orthogonalization method in the frequency domain was introduced as a frequency-domain LMS (FLMS) algorithm for the first time by Narayan and Peterson [14]. As for the FBLMS ADF, a similar concept was studied by Mansour and Gray [20] and by Picchi and Prati [26]. It is noted here that the FLMS ADF in [14] was proposed independently of the BLMS ADF. However, although the self-orthogonalizing FLMS and FBLMS ADF's were known to have significantly improved convergence speed, their performance, particularly the steady-state mean-squared error (MSE), has not been studied analytically until recently. The performance of the FLMS ADF has recently been analyzed by Lee and Un [35]. In this study it was shown that its convergence factor can be chosen such that the steady-state MSE satisfies the design specification. In another work by the same authors, it has been found that the FBLMS ADF reduces to the FLMS ADF when the block length is equal to one, and the convergence factors of the two ADF's can be related to each other [28]. Therefore, based on the results of [28] and [35], the steady-state performance of the self-orthogonalizing FBLMS ADF can be investigated. Although several structures of the BLMS ADF were studied in [24], no extensive results on the performance characteristics of the BLMS ADF's have been reported yet.

In this work, we extensively study the structures and performance characteristics of the BLMS ADF's. Part I is devoted largely to the systematic development of various BLMS ADF's, whereas Part II is concerned with a comprehensive performance analysis of those algorithms developed in Part I. In Part I of the paper, based on a unified matrix treatment, we first discuss the overlap-save and overlap-add sectioning methods for the fast convolution realization of the TBLMS ADF, and then extend our discussion to the FBLMS ADF. In particular, we present some new results of the overlap-add realization and the number-theoretic transform realization of the FBLMS ADF. We then study how we can incorporate in the BLMS ADF's the concept of different frequency-weighting on the error signals and the concept of the self-orthogonalization of the weight adjustment in the frequency domain for improving the convergence speed. These objectives are accomplished by formulating it directly in the frequency domain, thereby obtaining the frequency-weighted FBLMS (FWFBLMS) algorithm and the self-orthogonalizing FBLMS algorithm. It is shown that, when frequency weighting is uniform, the FWFBLMS algorithms realized using the overlap-save and overlap-add sectioning methods reduce to the FBLMS algorithms. Also, we investigate the effect of the convergence factor on the steady-state MSE of the self-orthogonalizing FBLMS ADF's. Then, it is shown that we can derive the TBLMS versions of those FBLMS algorithms utilizing the concepts of frequency-weighting and self-orthogonalization. Therefore, the TBLMS ADF can be made to have the same fast convergence speed as that of the self-orthogonalizing FBLMS ADF. Next, we investigate the possibility of removing some sectioning operations in the weight-adjustment algorithms of the FBLMS ADF's realized using the overlap-save and overlap-add sectioning methods. Specifically, based on the properties of the sectioning matrices used in representing the original FBLMS ADF's,

possible reduced forms are suggested both for the overlap-save and overlap-add realizations. One of these algorithms has been called the unconstrained FBLMS (UFBLMS) algorithm in the literature [21]. The performance characteristics of these UFBLMS ADF's will be analyzed extensively in Part II.

The organization of Part I of the paper is as follows. In Section II we introduce the matrix representations of the BLMS ADF and its variations realized using the fast convolution. In Section III we discuss the frequency-weighted and self-orthogonalizing algorithms for the FBLMS ADF, and then for the TBLMS ADF. Based on the properties of the sectioning matrices, we investigate the structures of the unconstrained FBLMS weight-adjustment algorithms both for the overlap-save and overlap-add sectioning methods in Section IV. In Section V we present and discuss computer simulation results demonstrating the performance characteristics of the BLMS ADF's. Finally, we draw conclusions in Section VI.

II. REALIZATION OF BLMS ADF BASED ON FAST CONVOLUTION

In this section, we first introduce the matrix representations of the BLMS ADF and its variations which will be used in our subsequent discussion. The fast convolution realization of the BLMS ADF using the two common sectioning methods, i.e., overlap-save and overlap-add sectioning, was discussed by Clark et al. [24]. Unlike for the case of overlap-save sectioning, they discussed the overlap-add realization method only when the block length and the number of weights are equal. In this paper, we present new results on the overlap-add realization method.

A. System Equations of the BLMS ADF

We consider an FIR ADF operated on the block-by-block basis, each block being L data samples long. We assume that the FIR ADF has M weights $\{w_m\}$ and that the filter produces its output $\{y_n\}$ from the input $\{x_n\}$ and the desired response $\{d_n\}$. All the signals are assumed to be stationary and real-valued. Throughout the paper, we use "k" for the block index.

We begin with the BMSE defined as[1]

$$\xi_B = \frac{1}{L} E[\mathbf{e}_k^t \mathbf{e}_k] \tag{1}$$

where

$$\mathbf{e}_k \triangleq [e_{kL}\ e_{kL+1} \cdots e_{kL+l} \cdots e_{kL+L-1}]^t,$$

$$e_{kL+l} \triangleq d_{kL+l} - y_{kL+l} \qquad 0 \le l \le L-1,$$

$$y_{kL+l} = \mathbf{w}_k^t \mathbf{x}_{kL+l} = \mathbf{x}_{kL+l}^t \mathbf{w}_k,$$

$$\mathbf{w}_k \triangleq [w_{k,0}\ w_{k,1} \cdots w_{k,M-2}\ w_{k,M-1}]^t,$$

and

$$\mathbf{x}_{kL+l} \triangleq [x_{kL+l}\ x_{kL+l-1} \cdots x_{kL+l-M+2}\ x_{kL+l-M+1}]^t.$$

In (1), $E[\cdot]$ and "t" denote statistical expectation and transpose of a matrix or a vector, respectively. Following the same approach used in derivation of the LMS algorithm, one can obtain the system equations of the BLMS ADF as the following [16],[22],[24]:

$$\mathbf{y}_k = \mathbf{X}_k \mathbf{w}_k, \tag{2a}$$

$$\mathbf{e}_k = \mathbf{d}_k - \mathbf{y}_k, \tag{2b}$$

and

$$\mathbf{w}_{k+1} = \mathbf{w}_k + \mu \mathbf{X}_k^t \mathbf{e}_k = \mathbf{w}_k + \mu \Delta \mathbf{w}_k, \tag{2c}$$

where

[1] In this paper, to represent vectors and matrices, we use boldface lower case and capital letters, respectively. Elements of vectors and matrices are represented by lower case letters. In addition, we use script letters to denote transform-domain variables.

$$\mathbf{X}_k \triangleq \begin{bmatrix} x_{kL} & x_{kL-1} & x_{kL-2} & \cdots & x_{kL-M+1} \\ x_{kL-1} & x_{kL} & x_{kL-1} & \cdots & x_{kL-M+2} \\ x_{kL+2} & x_{kL+1} & x_{kL} & \cdots & x_{kL-M+3} \\ \vdots & \vdots & \vdots & & \vdots \\ x_{kL+L-2} & x_{kL+L-3} & x_{kL+L-4} & \cdots & x_{kL-M+L-1} \\ x_{kL+L-1} & x_{kL+L-2} & x_{kL+L-3} & \cdots & x_{kL-M+L} \end{bmatrix},$$

$$\mathbf{y}_k \triangleq [y_{kL}\ y_{kL+1} \cdots y_{kL+L-1}]^t,$$

$$\mathbf{d}_k \triangleq [d_{kL}\ d_{kL+1} \cdots d_{kL+L-1}]^t,$$

and μ is a convergence factor. In (1) and (2), the sizes of $\mathbf{X}_k$ and $\mathbf{w}_k$ are (LxM) and (Mx1), respectively, and those of column vectors $\mathbf{y}_k$, $\mathbf{d}_k$ and $\mathbf{e}_k$ are all (Lx1). It is noted here that direct realization of (2) requires 2LM multiplication operations.

B. Fast Convolution Realization

The convolution (2a) and the correlation (2c) may be computed efficiently using the fact that $\mathbf{X}_k$ is Toeplitz when it is square. That is, $\mathbf{X}_k$ and $\mathbf{X}_k^t$ have a structure that can be made a circulant matrix by appropriate modification, which is equivalent to the overlap-save or overlap-add sectioning procedure [36],[37]. As will be seen, the use of matrix representations in our discussion makes it possible to handle various algorithms effectively in studying the characteristics of the BLMS ADF.

1) Overlap-Save Sectioning: A fast convolution realization of the BLMS ADF using the FFT and the overlap-save sectioning method is well documented in [24]. Therefore, here we just summarize the results using our notations. First, we define three frequency-domain variables that are given by the transforms of the augmented time-domain variables as the following [22],[24]:

$$_s\mathbf{X}_k \triangleq \mathbf{F}\ {}_s\mathbf{X}_k\ \mathbf{F}^{-1}, \tag{3a}$$

$$_s\boldsymbol{\omega}_k \triangleq \mathbf{F}\ {}_s\mathbf{w}_k, \tag{3b}$$

and

$$_s\mathbf{e}_k \triangleq \mathbf{F}\ {}_s\mathbf{e}_k, \tag{3c}$$

where $\mathbf{F}$ is the (NxN) DFT matrix with $N \triangleq L + M - 1 + N_z$, and the two (Nx1) column vectors are defined by

$$_s\mathbf{w}_k \triangleq \begin{bmatrix} \mathbf{w}_k \\ \mathbf{0} \end{bmatrix}_{N-M}^{M}, \tag{3d}$$

$$_s\mathbf{e}_k \triangleq \begin{bmatrix} \mathbf{0} \\ \mathbf{e}_k \end{bmatrix}_{L}^{N-L}, \tag{3e}$$

and $\mathbf{0}$ denotes a zero vector or matrix. In (3a), $_s\mathbf{X}_k$ is an (NxN) circulant matrix whose first column for the case of the widely-used overlap-save sectioning is given as[2]

$$[\underbrace{0 \cdots 0}_{N_z}\ |\ \underbrace{x_{kL-M+1} \cdots x_{kL-M+2} \cdots x_{kL-1}}_{(M-1)\text{ previous data}}\ |\ \underbrace{x_{kL}x_{kL+1} \cdots x_{kL+L-2}x_{kL+L-1}}_{L\text{ data in the current block}}]^t \tag{4}$$

where N_z is the number of zero data, which allows to choose a suitable transform length of N. Thus, the circulant matrix $_s\mathbf{X}_k$ contains the (LxM) matrix $\mathbf{X}_k$ on the lower left corner as follows [22][3]:

$$_s\mathbf{X}_k \triangleq \begin{bmatrix} \mathbf{X}_a & \mathbf{X}_b \\ \mathbf{X}_k & \mathbf{X}_c \end{bmatrix}_{L}^{N-L} \tag{5}$$
$$M \quad N-M$$

Therefore, one can see from the discrete convolution theorem [37] that $_s\mathbf{X}_k$ of (3a) is an (NxN) diagonal matrix whose diagonal elements are given by the DFT of the first column of

---

[2] Assuming the circular down shift property, a circulant matrix is completely specified by any column or row [24], [37].

[3] It is noted here that in (5) the direct relation between $\mathbf{X}_k$ and $_s\mathbf{X}_k$ makes it possible to realize the BLMS ADF more efficiently using overlap-save sectioning than overlap-add sectioning. As will be seen later, since the effects of the matrices $\mathbf{X}_a$, $\mathbf{X}_b$ and $\mathbf{X}_c$ are removed by the sectioning operations, the N_z zero data in (4) can be replaced by the previous data $[x_{kL-M+N_z-1}\ x_{kL-M+N_z-2} \cdots x_{kL-M+1}\ x_{kL-M}]$.

$_sX_k$ in (4) [22],[24]. Consequently, by combining the equations of (2a),(2c), and (3)-(5), one can get the same results as those in [24]. That is,

$$y_k = P_{,L} F^{-1}(_sX_k \; _s\omega_k) \tag{6a}$$

and

$$w_{k+1} = w_k + \mu P_M, \; F^{-1}(_sX_k \; _s e_k) \tag{6b}$$

where the overbar denotes complex conjugate. In (6), $P_{,L}$ and P_M, are (LxN) and (MxN) sectioning matrices for obtaining the final results and they are defined as

$$P_{,L} \triangleq [\underset{N-L}{O} \mid \underset{L}{I_L}]\,L \quad \text{and} \quad P_M \triangleq [\underset{M}{I_M} \mid \underset{N-M}{O}]\,M \tag{7}$$

where I denotes an identity matrix and the subscript on I represents the dimension of I It is noted that no constraints on the values of L, M, and N have been imposed in our discussion of the overlap-save realization method.

So far, we have discussed the matrix representation of the BLMS ADF realized using the FFT and overlap-save sectioning method. Before proceeding further, we summarize the design procedure used which will be useful in deriving the overlap-add realization method later in the next subsection.

Overlap-save Sectioning Procedure for BLMS ADF

Step 1 : Construct the circulant matrix $_sX_k$ (In other words, choose the elements of the first column of $_sX_k$) --- Sectioning of input data [see (4) and (5)].

Step 2 : Augment the weight and error vectors according to the relation between X_k and $_sX_k$ (or X_k^t and $_sX_k^t$) such that y_k and $\triangle w_k$ can be contained in the products $_sX_k \, _sw_k$ and $_sX_k^t \, _se_k$, respectively. --- Augmentation of coefficients [see {(2a),(3a),(3d),(5)} and {(2c),(3c),(3e),(5)}].

Step 3 : Design appropriate output sectioning matrices so as to obtain y_k and $\triangle w_k$ from $_sX_k \; _sw_k$ and $_sX_k^t \; _se_k$, respectively. --- Sectioning of output data [see {(2a),(3d),(5),(6a),(7)} and {(2c),(3e),(5),(6b),(7)}].

2) Overlap-add Sectioning: Remembering the three design steps outlined above for overlap-save sectioning, let us consider the overlap-add sectioning procedure for the fast convolution realization of a BLMS ADF. In the first step, we construct an (NxN) circulant matrix $_aX_k$ for overlap-add sectioning. The most common choice is to design the first column of $_aX_k$ as

$$\underbrace{\{x_{kL}\, x_{kL+1} \cdots x_{kL+L-2}\, x_{kL+L-1}}_{\text{L data in the current block}} \quad \underbrace{0 \cdots\cdots\cdots\cdots 0\}}_{(M-1+N_z)\text{ zeros}} \tag{8}$$

It can easily be expected that, unlike the overlap-save sectioning case, some part of X_k in (2) will not appear in $_aX_k$ because (8) does not include the previous data $\{x_{kL-M+1}, x_{kL-M+2}, \cdots x_{kL-2}\, x_{kL-1}\}$. As discussed in [24], this aspect together with the change of the weights once in every block results in the increased number of FFT's needed in the overlap-add realization.

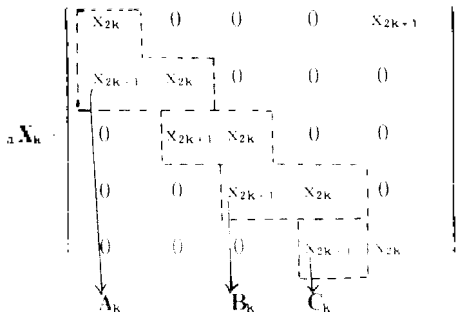
The number of the required FFT's in the overlap-add realization depends on the relative size of L with respect to that of M. In the following, we illustrate the relationship between X_k and $_aX_k$ for different L's given M. In these examples, the value of L is varied from 2 to 4 for M=4 and $N_z = 0$.

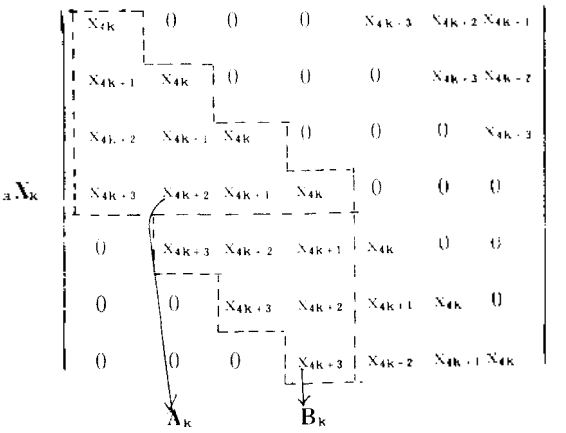
Examples of X_k and $_aX_k$

i) M = 4, L = 2 and N = 5

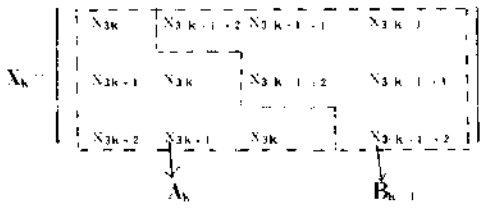
$$X_k = \left[\begin{array}{c|c|c} x_{2k} & x_{2k-1} \; x_{2k} & x_{2k-2} \; x_{2k-1} \\ \hline x_{2k+1} & x_{2k} \; x_{2k+1} & x_{2k-1} \; x_{2k} \end{array}\right]$$

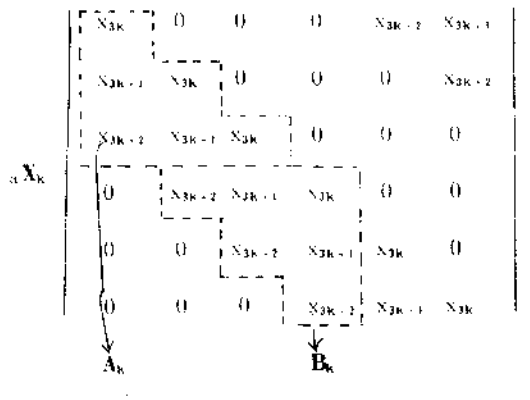
$A_k \qquad B_{k-1} \qquad C_{k-2}$



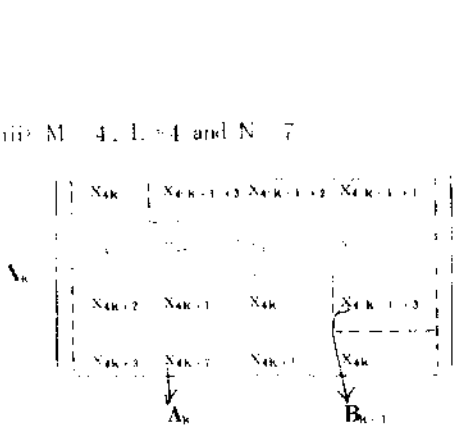


(ii) M = 4, L = 3 and N = 6





(iii) M = 4, L = 4 and N = 7



We can see from the abobe examples that, if we compute(2a)and(2c)using the overlap-add sectioning method, we need ${}_aX_k$, ${}_aX_{k-1}$, and ${}_aX_{k-2}$ for computing A_k, B_{k-1} and C_{k-2}, respectively. In general, to compute $X_k w_k$ and $X_k' e_k$, we need ${}_aX_k$, ${}_aX_{k-1}$, $\cdots$, ${}_aX_{k-r}$ where $r = \left\lceil \frac{M-1}{L} \right\rceil$, $\lceil \cdot \rceil$ indicating the least integer that is equal to or greater than the quantity inside the bracket. It is noted that, when $L \geq M-1$, the value of r becomes the minimum value of 1 in which case ${}_aX_k$ and ${}_aX_{k-1}$ will be used. For simplicity, we will consider the case of $L \geq M-1$ in our discussion of the overlap-add realization. For our discussion of overlap-add realization, we depict X_k and ${}_aX_k$ in Fig. 1 when $L \geq M-1$. In this figure, A_k' and A_k'' (or B_k' and B_k'') have the same elements, but are located in different places. It is noted that if $L \geq M-1$, A_k' and B_k' always exist in ${}_aX_k$ independently of the nonnegative integer N_z. However, the complete A_k'' exists only if $N_z - 1 \geq 0$ (i.e., $N_z \geq 1$), and B_k'' exists only if $L - M + 1 \geq 0$ (i.e., $L \geq M-1$). Therefore, the use of A_k' in ${}_aX_k$ and B_{k-1}' in ${}_aX_{k-1}$ is most the use of A_k in ${}_aX_k$ and B_{k-1} in ${}_aX_{k-1}$ is most desirable for computing (2a) and (2c). The second step is to augment the weight and error vectors so as to compute y_k and Δw_k in terms of A_k' and B_k' included in ${}_aX_k$ and ${}_aX_{k-1}$, respectively.

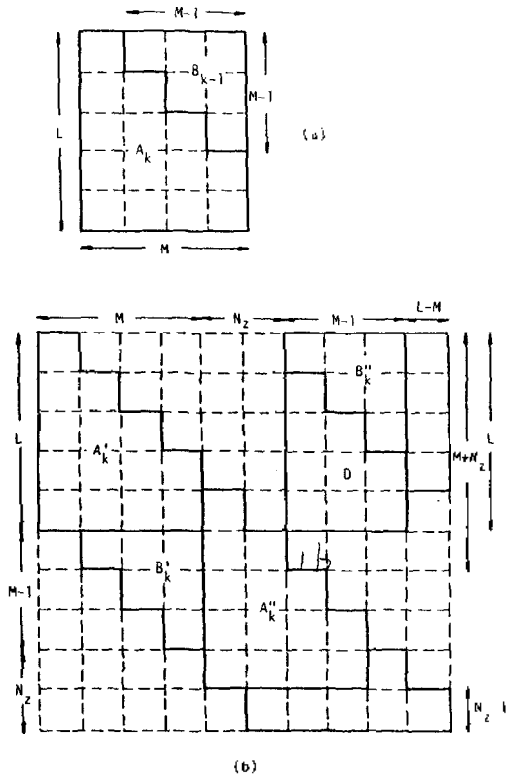


Fig. 1. A schematic representation of X_k and ${}_aX_k$ for overlap-add sectioning when $L \geq M-1$ (L=5, M=4, N_2=2).
(a) X_k (b) ${}_aX_k$

One can see from (2a), (2c), and Fig. 1 that in order to use A'_k and B'_{k-1}, we must define an augmented weight vector and two augmented error vectors as the following:

$$ {}_a\mathbf{w}_k \triangleq \begin{bmatrix} \mathbf{w}_k \\ \hline \mathbf{0} \end{bmatrix}_{N-M}^{M} \tag{9a} $$

$$ {}_a\mathbf{e}_k \triangleq \begin{bmatrix} \mathbf{e}_k \\ \hline \mathbf{0} \end{bmatrix}_{N-L}^{L} \tag{9b} $$

and

$$ {}_a\mathbf{e}_k \triangleq [\underbrace{0 \cdots 0}_{L} \quad \underbrace{e_{kL}, e_{kL-1}, \cdots, e_{kL-M+1}}_{M-1} \quad \underbrace{0 \cdots 0}_{N_2}]^t \tag{9c} $$

It is noted from (9b) and (9c) that ${}_a\mathbf{e}_k$ and ${}_a\mathbf{e}_k$ are related to each other via the (NxN) matrix $\mathbf{S}_{M-1,0}$ as

$$ {}_a\mathbf{e}_k = \mathbf{S}_{M-1,0}\, {}_a\mathbf{e}_k \tag{10} $$

whre

$$ \mathbf{S}_{M-1,0} \triangleq \begin{matrix} L \\ M-1 \\ N_2 \end{matrix} \left[\begin{array}{c|c} \mathbf{0} & \mathbf{0} \\ \hline \mathbf{I}_{M-1} & \mathbf{0} \\ \hline \mathbf{0} & \mathbf{0} \end{array}\right] \begin{matrix} \\ \\ \\ \end{matrix} \qquad \begin{matrix} \\ \\ \\ M-1 \quad L \end{matrix} \begin{matrix} \\ \\ \\ N_2 \end{matrix} $$

In the third step, by properly sectioning the outut, we can obtain y_k and w_k as

$$ y_k = \mathbf{P}_{L,\,a}\mathbf{X}_k\, {}_a\mathbf{w}_k + \mathbf{Q}_{1,\,a}\mathbf{X}_{k-1}\, {}_a\mathbf{w}_k \tag{11} $$

and

$$ \triangleq w_k = \mathbf{P}_{M,\,a}\mathbf{X}_k^t\, {}_a\mathbf{e}_k + \mathbf{U}_{M,\,a}\mathbf{X}_k^t\, {}_a\mathbf{e}_k \tag{12} $$

where the (LxN) sectioning matrices $\mathbf{P}_L$, and $\mathbf{Q}_1$, and the (MxN) matrix $\mathbf{U}_M$, are defined as

$$ \mathbf{P}_{L,} \triangleq \begin{matrix} \\ L \end{matrix} \left[\begin{array}{c|c} \mathbf{I}_L & \mathbf{0} \end{array}\right]_{L,\ N-L} \tag{13} $$

$$ \mathbf{Q}_{1,} \triangleq \begin{matrix} \\ L \end{matrix} \left[\begin{array}{ccc} \mathbf{0} & \mathbf{I}_{M-1} & \mathbf{0} \\ \hline \mathbf{0} & \mathbf{0} & \mathbf{0} \end{array}\right]_{L-M+1,} \begin{matrix} M-1 \\ \\ N_2 \end{matrix} \tag{14} $$

and

$$ \mathbf{U}_{M,} \triangleq \begin{matrix} \\ 1 \end{matrix} \left[\begin{array}{ccc} \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \hline \mathbf{0} & \mathbf{I}_{M-1} & \mathbf{0} \end{array}\right]_{M,} \begin{matrix} L \\ \\ N_2 \end{matrix} \tag{15} $$

Defining the frequency-domain variables as

$$ {}_a\mathbf{X}_k \triangleq \mathbf{F}_N\, \mathbf{X}_k\, \mathbf{F}^{-1} \tag{16} $$

$$ {}_a\boldsymbol{\omega}_k \triangleq \mathbf{F}_N \mathbf{w}_k, \tag{17} $$

$$ {}_a\boldsymbol{e}_k \triangleq \boldsymbol{F}\,{}_a\boldsymbol{e}_k \,. \tag{16c} $$

and

$$ {}_a\boldsymbol{e}_k \triangleq \boldsymbol{F}\,{}_a\boldsymbol{e}_k \,. \tag{16d} $$

we finally get the fast convolution realization of the BLMS ADF based on overlap-add sectioning as the following:

$$ \boldsymbol{y}_k = \boldsymbol{P}_{L,L}\,\boldsymbol{F}^{-1}\left({}_a\boldsymbol{X}_k\,{}_a\boldsymbol{w}_k\right) + \boldsymbol{Q}_{L,L}\,\boldsymbol{F}^{-1}\left({}_a\boldsymbol{X}_{k+1}\,{}_a\boldsymbol{w}_k\right) \tag{17} $$

and

$$ \boldsymbol{w}_{k+1} = \boldsymbol{w}_k + \mu\left[\boldsymbol{P}_{M,L}\,\boldsymbol{F}^{-1}\left({}_a\overline{\boldsymbol{X}}_k\,{}_a\boldsymbol{e}_k\right) + \boldsymbol{U}_{M,L}\,\boldsymbol{F}^{-1}\left({}_a\overline{\boldsymbol{X}}_k\,{}_a\boldsymbol{e}_k\right)\right] \tag{18} $$

To demonstrate the overlap-add realization method discussed above, let us consider a low-order filter example with M=2, L=3, and N=5 (i.e., $N_Z=1$), for which $\boldsymbol{Y}_k$ and $\Delta\boldsymbol{W}_k$ are given by

$$ \boldsymbol{y}_k = \begin{bmatrix} x_{3k} & x_{3k-1} \\ x_{3k+1} & x_{3k} \\ x_{3k+2} & x_{3k+1} \end{bmatrix} \begin{bmatrix} w_{k,0} \\ w_{k,1} \end{bmatrix} \tag{19a} $$

and

$$ \Delta\boldsymbol{W}_k = \begin{bmatrix} x_{3k} & x_{3k-1} & x_{3k-2} \\ x_{3k+1} & x_{3k} & x_{3k-1} \end{bmatrix} \begin{bmatrix} e_{3k} \\ e_{3k+1} \\ e_{3k+2} \end{bmatrix} \tag{19b} $$

Using the definitions of ${}_a\boldsymbol{X}_k$, ${}_a\boldsymbol{w}_k$, ${}_a\boldsymbol{e}_k$ and ${}_a\boldsymbol{e}_k$, we obtain four convolution and correlation terms as follows.

$$ {}_a\boldsymbol{X}_k\,{}_a\boldsymbol{w}_k = \begin{bmatrix} \boxed{\begin{matrix} x_{3k} & 0 \\ x_{3k+1} & x_{3k} \\ x_{3k+2} & x_{3k+1} \end{matrix}} & \begin{matrix} 0 & x_{3k+2} & x_{3k+1} \\ 0 & 0 & x_{3k+2} \end{matrix} \\ \begin{matrix} x_{3k} & 0 & 0 \\ 0 & x_{3k+2} & x_{3k+1} \\ 0 & 0 & x_{3k} \end{matrix} & \begin{matrix} 0 & 0 \\ x_{3k} & 0 \\ x_{3k+1} & x_{3k} \end{matrix} \end{bmatrix} \begin{bmatrix} \boxed{\begin{matrix} w_{k,0} \\ w_{k,1} \end{matrix}} \\ \begin{matrix} 0 \\ 0 \\ 0 \end{matrix} \end{bmatrix} \tag{20a} $$

$$ {}_a\boldsymbol{X}_{k-1}\,{}_a\boldsymbol{w}_k = \begin{bmatrix} \begin{matrix} x_{3k-3} & 0 \\ x_{3k-2} & x_{3k-3} \\ x_{3k-1} & x_{3k-2} \end{matrix} & \begin{matrix} 0 & x_{3k-1} & x_{3k-2} \\ 0 & x_{3k-1} & \end{matrix} \\ \begin{matrix} x_{3k-3} & 0 & 0 \\ 0 & x_{3k-1} & x_{3k-2} \\ 0 & 0 & x_{3k-3} \end{matrix} & \begin{matrix} 0 & 0 \\ 0 & 0 \\ x_{3k-1} & 0 \end{matrix} \end{bmatrix} \begin{bmatrix} \boxed{\begin{matrix} w_{k,0} \\ w_{k,1} \end{matrix}} \\ \begin{matrix} 0 \\ 0 \\ 0 \end{matrix} \end{bmatrix} \tag{20b} $$

$$ {}_a\boldsymbol{X}_k^t\,{}_a\boldsymbol{e}_k = \begin{bmatrix} \boxed{\begin{matrix} x_{3k} & x_{3k+1} & x_{3k+2} \\ 0 & x_{3k} & x_{3k-1} \end{matrix}} & \begin{matrix} 0 & 0 \\ x_{3k-2} & 0 \end{matrix} \\ \begin{matrix} 0 & 0 & x_{3k} \\ x_{3k-2} & 0 & 0 \\ x_{3k-1} & x_{3k-2} & 0 \end{matrix} & \begin{matrix} x_{3k+1} & x_{3k+2} \\ x_{3k} & x_{3k+1} \\ 0 & x_{3k} \end{matrix} \end{bmatrix} \begin{bmatrix} \boxed{\begin{matrix} e_{3k} \\ e_{3k+1} \\ e_{3k+2} \end{matrix}} \\ \begin{matrix} 0 \\ 0 \end{matrix} \end{bmatrix} \tag{20c} $$

and

$$ {}_a\boldsymbol{X}_{k+1}^t\,{}_a\boldsymbol{e}_k = \begin{bmatrix} \begin{matrix} x_{3k+3} & x_{3k+2} & x_{3k+1} \\ 0 & x_{3k+3} & x_{3k+2} \end{matrix} & \begin{matrix} 0 & 0 \\ \boxed{x_{3k+1}} & 0 \end{matrix} \\ \begin{matrix} 0 & 0 & x_{3k+3} \\ x_{3k+2} & 0 & 0 \\ x_{3k+1} & x_{3k+2} & 0 \end{matrix} & \begin{matrix} x_{3k+1} & x_{3k+2} \\ x_{3k} & x_{3k+1} \\ 0 & x_{3k} \end{matrix} \end{bmatrix} \begin{bmatrix} \begin{matrix} 0 \\ 0 \end{matrix} \\ \boxed{\begin{matrix} 0 \\ e_{3k+2} \end{matrix}} \end{bmatrix} \tag{20d} $$

Consequently, according to (11)-(15) and (20), we get

$$y_k = \begin{bmatrix} x_{3k} W_{k,0} \\ x_{3k+1} W_{k,0} + x_{3k} W_{k,1} \\ x_{3k+2} W_{k,0} + x_{3k+1} W_{k,1} \end{bmatrix} + \begin{bmatrix} x_{3k-1} W_{k,1} \\ 0 \\ 0 \end{bmatrix}$$

$$= \begin{bmatrix} x_{3k} W_{k,0} + x_{3k-1} W_{k,1} \\ x_{3k+1} W_{k,0} + x_{3k} W_{k,1} \\ x_{3k+2} W_{k,0} + x_{3k+1} W_{k,1} \end{bmatrix} \quad (21a)$$

and

$$\Delta \mathbf{w}_k = \begin{bmatrix} x_{3k} e_{3k} + x_{3k+1} e_{3k+1} + x_{3k+2} e_{3k+2} \\ x_{3k} e_{3k+1} + x_{3k+1} e_{3k+2} \end{bmatrix} + \begin{bmatrix} 0 \\ x_{3k-1} e_{3k} \end{bmatrix}$$

$$= \begin{bmatrix} x_{3k} e_{3k} + x_{3k+1} e_{3k+1} + x_{3k-2} e_{3k+2} \\ x_{3k-1} e_{3k} + x_{3k} e_{3k+1} + x_{3k-1} e_{3k+2} \end{bmatrix} \quad (21b)$$

It is noted that in [24], when L=M, Δw_k was computed using A_k' and B_k'', and y_k was computed using A_k' and B_{k-1}'. However, in that case, more restrictions must be imposed on the parameter values. Specifically, to use A_k' and B_{k-1}'', it is required that $L \geq M$ and $N_z \geq L - M$. The latter condition comes from the fact that to obtain a correct w_k, A_k'' and D in Fig. 1 should not be overlapped and thus $M - N_z - L$. Consequently, when $L > M$ and $N_z > L - M$, one can obtain from (2c),(9),(16) and Fig. 1 an alternative realization method for weight adjustment as

$$\mathbf{w}_{k+1} = \mathbf{w}_k + \mu \mathbf{P}_M \cdot \mathbf{F}^{-1}(_a\bar{\mathbf{X}}_{k,a} e_k) - \mathbf{V}_M \cdot \mathbf{F}^{-1}(_a\bar{\mathbf{X}}_{k+1,a} e_k) \quad (22a)$$

where the (MxN) sectioning matrix $\mathbf{V}_M$ is defined as

$$\mathbf{V}_M \triangleq \begin{bmatrix} \underbrace{\mathbf{0}}_{M+N_z-M-1} & \underbrace{\mathbf{0}}_{1} & \underbrace{\mathbf{0}}_{L-M} \\ \mathbf{0} & \mathbf{I}_{M-1} & \mathbf{0} \\ \mathbf{0} & 0 & \mathbf{0} \end{bmatrix}_{M} \quad (22b)$$

It is noted that, when L=M and $N_z = 1$, the results of (17) and (22) become the same as those in [24]. Comparing (18) and (22), it appears that realization of the BLMS ADF based on (18) requires one more FFT in comparison to that based on (22). However, one can note from (7), (9c),(15), and Fig. 1 that the first element of the product $_aX_{k-1}^{1} \, _a\tilde{e}_k$ is always zero, and thus V_M in (18) can be replaced by P_M. Therefore, we can realize (18) with one FFT operation eliminated as the following:

$$\mathbf{w}_{k+1} = \mathbf{w}_k + \mu \mathbf{P}_M \cdot \mathbf{F}^{-1}(_a\bar{\mathbf{X}}_{k,a} e_k + {}_a\bar{\mathbf{X}}_{k+1,a} e_k) \quad (2$$

Note that the two weight-adjustment algorithms in (18) and (23) yield the same performance.

C. Frequency-domain BLMS Algorithms

Up to now, we have discussed efficient realization of the BLMS ADF in (2) using the fast convolution. As a result, the weight adjustment algorithm in (2c), which is described in terms of the time-domain variables, can be realized using the FFT's as in (6b),(18),(22), and (23). However, in those algorithms the weights are adjusted still in the time domain. Hereafter, we shall call (6b), (18), and (22) the time-domain BLMS (TBLMS) algorithms. The weights of the BLMS algorithm realized using the FFT's can also be adjusted in the frequency domain. Using the definition of the frequency-domain weight vectors, we obtain the frequency-domain BLMS (FBLMS) algorithms corresponding to (6b),(18),(22), and (23), respectively, as the following:

Overlap-Save :

$$_a\boldsymbol{\omega}_{k+1} = {}_a\boldsymbol{\omega}_k + \mu \boldsymbol{P}_{M,a} \cdot {}_a\bar{\mathbf{X}}_{k,a} e_k \quad (2$$

Overlap-Add I :

$$_a\boldsymbol{\omega}_{k+1} = {}_a\boldsymbol{\omega}_k + \mu \boldsymbol{P}_{M,a} \cdot {}_a\bar{\mathbf{X}}_{k+1} e_k + \boldsymbol{U}_{M,a} \cdot {}_a\bar{\mathbf{X}}_{k+1} e_k$$

(2

or

$$_{s}\boldsymbol{\omega}_{k+1} = {}_{a}\boldsymbol{\omega}_{k} + \mu P_{M,0}({}_{a}\bar{X}_{k,a}e_{k} + {}_{a}\bar{X}_{k,a}e_{k}') \quad (25b)$$

Overlap-Add II :

$${}_{a}\boldsymbol{\omega}_{k+1} = {}_{a}\boldsymbol{\omega}_{k} + \mu\{P_{M,0}({}_{a}\bar{X}_{k,a}e_{k}) + V_{M,0}({}_{a}\bar{X}_{k-1,a}e_{k})\} \quad (26)$$

In (24)-(26), the (NxN) matrices $P_{M,0}$, $U_{M,0}$ and $V_{M,0}$ are defined as

$$P_{M,0} \triangleq FP_{M,0}F^{-1},\ U_{M,0} \triangleq FU_{M,0}F^{-1},\ \text{and}\ V_{M,0} \triangleq FV_{M,0}F^{-1} \quad (27)$$

where

$$P_{M,0} \triangleq \begin{bmatrix} P_{M,0} & M \\ O & L \\ N & \end{bmatrix},\quad U_{M,0} \triangleq \begin{bmatrix} U_{M,0} & M \\ O & L \\ N & \end{bmatrix}$$

$$\text{and}\ V_{M,0} \triangleq \begin{bmatrix} V_{M,0} & M \\ O & L \\ N & \end{bmatrix} \quad (27')$$

As will be seen in Section IV, the FBLMS algorithms have some special structures resulting from the unique properties of $P_{M,0}$, $U_{M,0}$ and $V_{M,0}$. In particular, the FBLMS algorithms appear to be working well without the sectioning operations represented by $P_{M,0}$ and $U_{M,0}$ in most applications. However, $V_{M,0}$ cannot be removed without serious performance degradation.

D. Alternative Realization Structures

It is known that the number-theoretic transforms (NTT's) have several desirable properties in computing convolutions and correlations in comparison to the FFT [38]. However, for the NTT realization, the complex conjugate operation in computing $_{s}w_{i}$ should be avoided. To eliminate the complex conjugate, we apply the result in [39] to the TBLMS and FBLMS algorithms realized using overlap-save sectioning. As a result, for (6b) and (24) we obtain alternative adjustment algorithms, respectively, as the following [22],[33]:

$$\mathbf{w}_{k+1} = \mathbf{w}_{k} + \mu P_{M,0}F^{-1}{}_{s}X_{k}^{*}(F^{-1}{}_{s}\mathbf{e}_{k}) \quad (28)$$

and

$${}_{s}\boldsymbol{\omega}_{k+1} = {}_{s}\boldsymbol{\omega}_{k} + \mu FP_{M,0}F^{-1}{}_{s}X_{k}^{*}(F^{-1}{}_{s}\mathbf{e}_{k}) \quad (29)$$

It is noted here that, unlike (6b) and (24), the algorithms shown in (28) and (29) can be realized using either the NTT or the FFT. Using the relation $F^{2} = NE$[39], the FBLMS algorithm of (29) can be realized alternatively as

$${}_{s}\boldsymbol{\omega}_{k+1} = {}_{s}\boldsymbol{\omega}_{k} + \mu N P_{M,0}E\{{}_{s}X_{k}^{*}(F^{-1}{}_{s}\mathbf{e}_{k})\} \quad (30)$$

In (30), the (NxN) matrix E is defined by

$$E \triangleq (e_{i,j}),\ e_{i,j} = \delta\{(i+j)_{N}\},\ \text{and}\ 1 \leq i,j \leq N, \quad (31)$$

where $\delta\{\cdot\}$ and $\{\cdot\}_{N}$ denote the delta function and the modulo N operation, respectively. It can be seen from (30) and (31) that E does not involve multiplication operations and is actually a matrix representation of reversing (modulo N) operations on the N rows of a matrix or a vector that follows E. A similar discussion can be made for overlap-add sectioning.

III. INCORPORATION OF FREQUENCY-DOMAIN INFORMATION IN BLMS ADF's

In Section II we have discussed the TBLMS and FBLMS algorithms as efficient realization methods for the BLMS ADF that we derived by minimizing the BMSE defined originally in the time domain. Of course, we can consider minimization of the BMSE defined alternatively in the frequency domain. Thus, the FBLMS algorithm can also be derived directly by minimizing the frequency-domain BMSE (FBMSE) [21],[24].

Our main concern here is how we can incorporate in BLMS ADF's some desirable information which originates from the frequency-domain consideration. It is conceptually natural to design those algorithms directly in the frequency domain. Different weighting on the frequency-domain errors and self-orthogonalization of the frequency-domain weights will be discussed in this context. Then, it will be shown that these frequency-domain modifications done in the FBLMS algorithms can be realized equally in the TBLMS algorithms.

A. Frequency-Weighted FBLMS Algorithms

In some applications, it is desirable to design the BLMS ADF with different frequency-weighting on the error signals depending on the relative significance of various frequency bands. For example, in speech enhancement it is more important perceptually to reduce noise in the frequency range of 1 to 3 kHz [31],[40]. Another example is designing of fixed-coefficient digital filters using the LMS algorithm, in which frequency weighting can be utilized effectively to satisfy the more critical design requirement at certain frequencies than at other [1, Chap. 9]. As a performance criterion satisfying this need, we define the frequency-weighted BMSE(FWBMSE) as[4]

$$\xi_k^{fw} \triangleq E[\, e_k^* \, \Gamma \, e_k] \tag{32}$$

where the asterisk denotes complex-conjugate transpose of a vector or a matrix. In (32), Γ is an (NxN) diagonal matrix whose diagonal elements are nonnegative and their relative magnitudes represent the relative significance of each frequency component. It should be noted from (32) that the frequency-domain error vector e_k has not been specified yet. The frequency-domain error vector e_k is obtained by transforming the augmented error vector in which zero data are added to the time-domain error vector *in an appropriate way*. On the other hand, the time-domain error vector is also related directly to the filter output vector through the relation $e_k = d_k - y_k$ Therefore, the frequency-domain error vector e_k can be of different form depending on how the augmented error vector and the output vector are computed. This implies that different weight-adjustment algorithms can result from different computation methods for y_k. In other words, unlike in the minimization of the time-domain BMSE, some information on the filter realization must be specified in the frequency-domain formulation.

Let us consider the overlap-save and overlap-add cases in oder. The frequency-domain error vector ${}_s e_k$ for the overlap-save realization can be obtained from (2b),(3c),(3e), and (6a). Since we are going to compute $\triangle {}_s\omega_k = \mu^2 \partial_s e_k / \partial_s \omega_k$, there must be some constraint on ${}_s\omega_k$ [see (24)]. As done in [20] and $P_{M,0}\, {}_s\omega_k = {}_s\omega_k$, we can implement this constraint as the following:

$${}_s e_k = {}_s d_k - P_{0,L}({}_s X_k \, P_{M,0} \, {}_s\omega_k) \tag{33}$$

where

$${}_s d_k \triangleq F \begin{bmatrix} 0 \\ \vdots \\ d_k \end{bmatrix}_{L}^{N-L}, \quad P_{0,L} \triangleq F P_{0,L} F^{-1}$$

$$\text{and } P_{0,L} \triangleq \begin{bmatrix} 0 & & \\ & \ddots & \\ & & P_{\perp} \end{bmatrix}_{L}^{N-L} \tag{34}$$

Noting from (32) that $\xi_k^{fw} = E[\, \Gamma^{1/2} {}_s e_k]^* \, (\Gamma^{1/2} {}_s e$ one can obtain $\triangle {}_s\omega_k$ as

$$\triangle {}_s\omega_k = E[\, \Gamma^{1/2} \, P_{0,L} \, {}_s X_k \, P_{M,0}]^* \, (\Gamma^{1/2} {}_s e_k] \tag{35}$$

Thus, using an instantaneously estimated gradient, we get a frequency-weighted FBLMS

[4] It is noted that there is no distance concept in the NTT case. Therefore, the NTT realization will not be considered whenever the concept of magnitude is necessary in our discussion.

(FWFBLMS) algorithm as the following:

$$ {}_{s}\boldsymbol{\omega}_{k+1} = {}_{s}\boldsymbol{\omega}_{k} - \mu\, \boldsymbol{P}_{M,0}\, {}_{s}\bar{\boldsymbol{X}}_{k}^{*}\, \boldsymbol{P}_{0,L}\, \Gamma\, {}_{s}\boldsymbol{e}_{k}, \tag{36} $$

In obtaining (36), we used the relations $\boldsymbol{P}_{M,0}^{*} = \boldsymbol{P}_{M,0}$ and $\boldsymbol{P}_{0,L}^{*} = \boldsymbol{P}_{0,L}$ [see (27) and (34)]. It is noted from (36) that, when Γ is an identity matrix, the FWFBLMS algorithm reduces to the FBLMS algorithm in (24) since $\boldsymbol{P}_{0,L}\, {}_{s}\boldsymbol{e}_{k} = {}_{a}\boldsymbol{e}_{k}$ from (3c),(3e) and (34).

Similarly, as for the case of overlap-add realization, the frequency-domain error vector ${}_{a}\boldsymbol{e}_{k}$ can be represented from (2b),(9b),(17) and (25b) by

$$ {}_{a}\boldsymbol{e}_{k} = {}_{a}\boldsymbol{d}_{k} - [\boldsymbol{P}_{L,0}\, {}_{a}\boldsymbol{X}_{k} + \boldsymbol{Q}_{L,0}\, {}_{a}\boldsymbol{X}_{k-1}]\, \boldsymbol{P}_{M,0}\, {}_{a}\boldsymbol{\omega}_{k}, \tag{37} $$

where $\boldsymbol{P}_{L,0} \triangleq \mathbf{F}\mathbf{P}_{L,0}\mathbf{F}^{-1}$, $\boldsymbol{Q}_{L,0} \triangleq \mathbf{F}\mathbf{Q}_{L,0}\mathbf{F}^{-1}$,

$$ {}_{a}\boldsymbol{d}_{k} \triangleq \mathbf{F}\begin{bmatrix} \mathbf{d}_{k} \\ \mathbf{0} \end{bmatrix}_{N-L}^{L}, \quad \mathbf{P}_{L,0} \triangleq \begin{bmatrix} \mathbf{P}_{L,0} \\ \mathbf{0} \end{bmatrix}_{N-L}^{L} $$

$$ \text{and,} \quad \mathbf{Q}_{L,0} \triangleq \begin{bmatrix} \mathbf{Q}_{L,0} \\ \mathbf{0} \end{bmatrix}_{N-L}^{L}. $$

Thus, we obtain ${}_{a}\boldsymbol{\omega}_{k}$ as

$$ \triangleq {}_{a}\boldsymbol{\omega}_{k} = [\mathbf{I}_{M}^{1/2}\, (\boldsymbol{P}_{L,0}\, {}_{a}\boldsymbol{X}_{k} + \boldsymbol{Q}_{L,0}\, {}_{a}\boldsymbol{X}_{k-1})\, \boldsymbol{P}_{M,0}]^{*} $$
$$ [\Gamma^{1/2}\, {}_{a}\boldsymbol{e}_{k}]. \tag{38} $$

As done in derivation of (36), we finally obtain an FWFBLMS algorithm based on overlap-add sectioning as the following:

$$ {}_{a}\boldsymbol{\omega}_{k+1} = {}_{a}\boldsymbol{\omega}_{k} + \mu\, \boldsymbol{P}_{M,0}\, [{}_{a}\boldsymbol{X}_{k}^{*}\, \boldsymbol{P}_{L,0}\, \Gamma\, {}_{a}\boldsymbol{e}_{k} + {}_{a}\boldsymbol{X}_{k-1}^{*}\, \boldsymbol{Q}_{L,0}^{*}\, \Gamma\, {}_{a}\boldsymbol{e}_{k}]. \tag{39} $$

Even when Γ is an identity matrix, the adjustment algorithm (39) in its present form appears to be somewhat different from (25b) though $\boldsymbol{P}_{L,0}\, {}_{a}\boldsymbol{e}_{k} = {}_{a}\boldsymbol{e}_{k}$. However, it can be shown that using the definitions of $\boldsymbol{Q}_{L,0}$ and $\mathbf{S}_{M-L,0}$ in (10),(14), and (37) leads to

$$ \boldsymbol{Q}_{L,0}^{*} = \mathbf{S}_{M-L,0}, \tag{40} $$

Therefore, when $\Gamma = \mathbf{I}_{N}$, we get for the second term of (39)

$$ {}_{a}\boldsymbol{X}_{k-1}^{*}\, \boldsymbol{Q}_{L,0}^{*}\, {}_{a}\boldsymbol{e}_{k} = {}_{a}\bar{\boldsymbol{X}}_{k}^{*} + \mathbf{F}\mathbf{S}_{M-L,0}\, {}_{a}\boldsymbol{e}_{k} = {}_{a}\bar{\boldsymbol{X}}_{k-1}^{*}\, {}_{a}\boldsymbol{e}_{k}, \tag{41} $$

Consequently, we can see that, unlike (25a), the FBLMS algorithm (25b) can be formulated directly in the frequency domain. This is because, when computing the correlation in (25a), we did not use the same A_k' and B_{k-1}' as in computation of the convolution (11).

B. Self-orthogonalizing FBLMS Algorithms

Several researchers proposed the so-called self-orthogonalizing algorithms to improve the convergence speed of the LMS-type algorithm [41],[42]. The self-orthogonalizing algorithm uses a matrix convergence factor that is obtained by multiplying the inverse of the autocorrelation matrix by a scalar constant. It can easily be shown from (1) and (2) that the mean of the weight-error vector which characterizes the convergence behavior of the BLMS algorithm in (2) is given as

$$ E[\mathbf{v}_{k+1}] = [\mathbf{I}_{M} - \mu\, {}_{B}\mathbf{R}_{X}]\, E[\mathbf{v}_{k}] \tag{42} $$

where $\mathbf{v}_{k} \triangleq \mathbf{w}_{k} - \mathbf{w}_{opt}$ and ${}_{B}\mathbf{R}_{X} \triangleq E[\mathbf{X}_{k}^{t}\, \mathbf{X}_{k}]$.

Therefore, a self-orthogonalizing algorithm for (2) can be of the following form:

$$ \mathbf{w}_{k+1} = \mathbf{w}_{k} + \gamma\, \mathbf{R}_{X}^{-1}\, \mathbf{X}_{k}^{t}\, \mathbf{e}_{k} \tag{43} $$

where γ is a constant that affects the convergence behavior of the algorithm. Since the FBLMS algorithm is an exact implementation of the BLMS algorithm, a self-orthogonalizing

overlap-save FBLMS algorithm which is equivalent to (43) becomes from (3d) and (6b)

$$ {}_{s}\boldsymbol{\omega}_{k+1} = {}_{s}\boldsymbol{\omega}_{k} + \gamma \, \mathrm{F} \begin{bmatrix} {}_{B}\mathbf{R}_{x}^{-1} & \vdots & \mathbf{0} \\ \cdots & \cdots & \cdots \\ \mathbf{0} & \vdots & \mathbf{0} \end{bmatrix}_{\substack{M \\ N \\ M}} \mathrm{F}^{-1} \, P_{M,0} \, {}_{s}\bar{X}_{k} \, {}_{s}e_{k}, \tag{44} $$

However, in practice the self-orthogonalizing algorithms in the frequency domain have been usually realized by using the frequency-domain information (i.e., frequency-domain power) [14],[21]. The self-orthogonalizing overlap-save FBLMS algorithm that has attracted attention recently can be represented as the following [22],[26],[34]:

$$ {}_{s}\boldsymbol{\omega}_{k+1} = {}_{s}\boldsymbol{\omega}_{k} + \gamma \, P_{M,0} \, {}_{s}\hat{R}_{k}^{-1} \, {}_{s}\bar{X}_{k} \, {}_{s}e_{k}. \tag{45} $$

So far, the use of an (NxN) diagonal matrix for ${}_{s}\hat{R}_{k}$ in (45) has been successful according to simulation results, and its diagonal elements are given as the following [21],[35]:

$$ {}_{s}\hat{R}_{k} \triangleq \mathrm{diag}\,({}_{s}\hat{P}_{k,0}, {}_{s}\hat{P}_{k,1}, \cdots, {}_{s}\hat{P}_{k,i}, \cdots, {}_{s}\hat{P}_{k,N-1}) \tag{46a} $$

where

$$ {}_{s}\hat{P}_{k,i} \triangleq \beta \, {}_{s}\hat{P}_{k-1,i} + (1-\beta)\,({}_{s}\chi_{k,i} \, {}_{s}\bar{\chi}_{k,i}), \quad 0 \leq i \leq N-1, \tag{46b} $$

and ${}_{s}\chi_{k,i}$ is the i_{th} diagonal element of ${}_{s}X_{k}$. We can see from (46) that each diagonal element of ${}_{s}\mathbf{R}_{k}$ is an estimated power spectrum for each frequency component of the input using a one-pole low-pass filter (LPF). In (46b), the smoothing constant β controls the accuracy and the time constant in estimating the frequency-domain signal power [35]. It is noted that in [26],the frequency-domain signal power ${}_{s}\hat{P}_{k,i}$ was estimated alternatively using the cumulative average up to the present time as the following:

$$ {}_{s}\hat{P}_{k,i} = \frac{1}{k+1} \sum_{j=0}^{k} {}_{s}\chi_{j,i} \, {}_{s}\bar{\chi}_{j,i}, \quad 0 \leq i \leq N-1. \tag{47} $$

Comparing the two self-orthogonalizing FBLMS algorithms of (44) and (45), we can see that realization of (44) requires far more computations than that of (45) [29]. Particularly, estimation of ${}_{B}R_{x}$ and inversion of the estimated matrix are required for (44). Also, it is seen from (44) and (45) that the self-orthogonalization operations are done in a different order in two cases, that is, after and before the sectioning operation $P_{M,0}$, respectively. Therefore, noting that (44) is an exact implementation of (43), one may raise a question as to how the self-orthogonalizing FBLMS algorithm in (45) works. This aspect will be discussed in Part II.

As for the case of the overlap-add sectioning realization, we propose the self-orthogonalizing FBLMS algorithm of the following form:

$$ {}_{a}\boldsymbol{\omega}_{k+1} = {}_{a}\boldsymbol{\omega}_{k} + \gamma P_{M,0} \, ({}_{a}\hat{R}_{k}^{-1} \, {}_{a}\bar{X}_{k} \, {}_{a}e_{k} + {}_{a}\hat{R}_{k-1}^{-1} \, {}_{a}\bar{X}_{k-1} \, {}_{a}e_{k}) \tag{48} $$

where

$$ {}_{a}\hat{R}_{k} \triangleq \mathrm{diag}\,({}_{a}\hat{P}_{k,0}, {}_{a}\hat{P}_{k,1}, \cdots, {}_{a}\hat{P}_{k,i}, \cdots, {}_{a}\hat{P}_{k,N-1}) $$

$$ {}_{a}\hat{P}_{k,i} \triangleq \beta \, {}_{a}\hat{P}_{k-1,i} + (1-\beta) \, {}_{a}\chi_{k,i} \, {}_{a}\bar{\chi}_{k,i}, \quad 0 \leq i \leq N-1, $$

and ${}_{a}\chi_{k,i}$ is the i_{th} diagonal element of ${}_{a}X_{k}$.

We now investigate the relation between the convergence factors μ and γ based on the results published recently. Detailed convergence analyses of the self-orthogonalizing FBLMS algorithms both for overlap-save and overlap-add sectioning will be done in Part II. Our discussion in the following applies to the overlap-save case only. It is known that, given the values of M and μ, the TBLMS and FBLMS ADF's described in (6),(17),(18),(22a),(23)-(26), and (28)-(30) have the same steady-state MSE (in the region of convergence) regardless of L, including the LMS case for L=1 [15],[16], [22]. Of course, the steady-state performances of the self-orthogonalizing FBLMS algorithms realized using the overlap-save sectioning method

will be affected significantly by the choice of the convergence factor γ in (45).

Recently, it has been found that the FBLMS ADF reduces to the FLMS ADF regardless of self-orthogonalization when the block length is equal to one [22], [28]. Also, it has been shown that, for the same steady-state MSE, the convergence factor of the FBLMS ADF must be N times that of the FLMS ADF. If we denote γ as the convergence factor of the self-orthogonalizing FLMS ADF, the relation can be expressed as [28]

$$\gamma = \alpha N. \tag{49}$$

On the other hand, the performance of the FLMS ADF has recently been analyzed in connection with that of the LMS ADF by the same authors [35]. The result obtained for the same steady-state MSE of the self-orthogonalizing FLMS ADF and the LMS ADF is given as the following [35]:

$$\mu = \frac{\alpha}{\sigma_x^2} \tag{50}$$

where σ_x^2 is the input signal power. In (50), it is noted that, as mentioned before, the FBLMS ADF and the LMS ADF can have the same steady-state MSE for the same μ. Consequently, we can see from (49) and (50) that, given the values of M and σ_x^2, the overlap-save FBLMS ADF's with and without self-orthogonalization can have the same steady-state MSE regardless of L under the following condition:

$$\gamma = \mu N \sigma_x^2. \tag{51}$$

Various results of computer simulations verifying (49)–(51) will be given in Section V.

C. Utilization of Frequency-domain Information in TBLMS Algorithms

In the previous subsections, we have discussed how we can incorporate some additional frequency-domain information in BLMS ADF's. Due to the inherent frequency-domain nature of these problems, we considered the frequency-weighted BLMS ADF and the self-orthogonalizing BLMS ADF directly in the frequency domain. However, it can be seen from (36), (39),(45) and (48) that the resulting modification of the frequncy-domain information occurs before the frequency-domain sectioning operation $P_{M,0}$ in both cases. Therefore, one can expect that the frequency-domain modification done in the FBLMS algorithms can be applied equally in the TBLMS cases. This is because the TBLMS algorithm has the same term used for realizing the correlation in the frequency domain.

Let us consider the overlap-save sectioning realization.[5] We can note from (3b) and (3d) that W_k is related to ${}_sW_k$ via the following relation:

$$w_k = P_{M,1} F^{-1} {}_s\omega_k. \tag{52}$$

Therefore, multiplying both sides of (36) and (45) by $P_{M,1} F^{-1}$ and noting $P_{M,1} P_{M,0} = P_M$, we obtain the frequency-weighted TBLMS (FWTBLMS) algorithm and the self-orthogonalizing TBLMS algorithm, respectively, as the following [34]:

$$w_{k+1} = w_k + \mu P_{M,1} F^{-1} {}_s\Lambda_k P_{0,1} F e_k \tag{53}$$

and

$$w_{k+1} = w_k + \gamma P_{M,1} F^{-1} {}_sR_k^{-1} {}_s\Lambda_k {}_se_k. \tag{54}$$

As a result, we can conclude that there is no difference in the performances of the TBLMS

[5] A similar discussion can be made on the overlap-add sectioning case.

and FBLMS ADF's. Particularly, the TBLMS ADF can be made to have the same fast convergence speed as that of the self-orthogonalizing FBLMS ADF. The concept of different frequency-weighting on the error signal can also be applied to the TBLMS ADF. On the other hand, it can be expected that both TBLMS and FBLMS ADF's have similar complexities [24].

IV. ELIMINATION OF THE CONSTRAINTS IN WEIGHT ADJUSTMENT OF THE FBLMS ADF's

It is known that under proper conditions, the FBLMS algorithm of (24), which is realized using the overlap-save sectioning method, can work without the constraint $P_{M,0}$ on the weights [20],[21]. In this section we discuss which sectioning matrices (or which constraints) in the weight adjustment of the FBLMS ADF's can be removed, thereby reducing two FFT operations in realization. In doing so, we treat the overlap-save and overlap-add realization methods in a unified way based on the properties of the sectioning matrices. It should be noted that there is no similar concept in the TBLMS ADF's.

A. Filter-bank Representation

Unlike the overlap-save realization, the FBLMS algorithms derived using overlap-add sectioning in Section II-C have two terms in the estimated gradient. In the following we show that the overlap-add FBLMS algorithms I and II in (25) and (26) can be expressed as a bank of two adjustment algorithms, each resembling the overlap-save FBLMS algorithm. As an example, we consider the overlap-add algorithm II. We obtain from (26)

$$ {}_a\boldsymbol{\omega}_{k+1} = {}_a\boldsymbol{\omega}_0 + \sum_{j=0}^{k} \mu P_{M,0}({}_a\bar{X}_{j,a}e_j) + \sum_{j=0}^{k} \mu V_{M,0} ({}_aX_{j,a}e_j), \tag{55} $$

Then, defining

$$ {}_i\boldsymbol{\omega}_{k+1} \triangleq {}_a\boldsymbol{\omega}_0 + \sum_{j=0}^{k} \mu P_{M,0}({}_a\bar{X}_{j,a}e_j) \tag{56a} $$

and

$$ {}_{ii}\boldsymbol{\omega}_{k+1} \triangleq \sum_{j=0}^{k} \mu V_{M,0}({}_a\bar{X}_{j,a}e_j) \tag{56b} $$

and from (55), we can realize the overlap-add algorithm II as the following:

$$ {}_a\boldsymbol{\omega}_{k+1} = {}_i\boldsymbol{\omega}_{k+1} + {}_{ii}\boldsymbol{\omega}_{k+1}, \tag{57a} $$

$$ {}_i\boldsymbol{\omega}_{k+1} = {}_i\boldsymbol{\omega}_k + \mu P_{M,0}({}_a\bar{X}_{k,a}e_k), \tag{57b} $$

and

$$ {}_{ii}\boldsymbol{\omega}_{k+1} = {}_{ii}\boldsymbol{\omega}_k + \mu V_{M,0}({}_a\bar{X}_{k-1,a}e_k), \tag{57c} $$

where

$$ {}_i\boldsymbol{\omega}_0 \equiv {}_a\boldsymbol{\omega}_0 \text{ and } {}_{ii}\boldsymbol{\omega}_0 \equiv 0. \tag{57d} $$

A similar result can be obtained for the overlap-add algorithm I in (25). This filter-bank representation will be useful in understanding the UFBLMS algorithms for overlap-add sectioning.

B. Properties of the Sectioning Matrices and Alternative Adjustment Algorithms

It is interesting to note that the sectioning matrices $P_{M,0}$, $U_{M,0}$, and $V_{M,0}$ have different properties. Specifically, we can show from (7),(15),(22b), and (27) that

$$ P^2_{M,0} = P_{M,0}, \quad U^2_{M,0} = U_{M,0}, \quad V^2_{M,0} = 0. \tag{58} $$

Therefore, we have from (27) and (58)

$$ P^n_{M,0} = P_{M,0}, \quad U^n_{M,0} = U_{M,0}, \text{ and } V^n_{M,0} = 0 $$

where n = 2,3,4,5,···.

Let us discuss alternative realization methods for (24)-(26), in which some sectioning

operations are reordered. For the case of the overlap-save sectioning, we can consider the following [20]:

$$ {}_{s}\omega_{k+1} = P_{M,0}({}_{s}\omega_{k} + \mu_{s}\,\bar{X}_{k\,s}e_{k}) \tag{60} $$

Using the identity $P_{M,0}^{n} = P_{M,0}$ in (59), we obtain from (60)

$$ {}_{s}\omega_{k+1} = P_{M,0}\,{}_{s}\omega_{0} + \sum_{j=0}^{k} \mu\, P_{M,0}({}_{s}\bar{X}_{j\,s}e_{j}), \tag{61} $$

Thus, we can see that (24) and (61) are identical when $P_{M,0}\,{}_{s}\omega_{0} \equiv {}_{s}\omega_{0}$. This condition can be met by choosing

$$ {}_{s}\omega_{0} = 0 \quad \text{or} \quad {}_{s}\omega_{0} = F\begin{bmatrix} w_{0} \\ \cdots \\ 0 \end{bmatrix}_{N-M}^{M}. \tag{62} $$

It is noted that unlike (24), the algorithm of (60) can converge (to the optimal solution) without the condition in (62). A block diagram for the realization of the FBLMS ADF based on the algorithm of (60) is shown in [34, Fig. 1].

Similarly, noting that ${}_{i}\omega_{0} \equiv {}_{a}\omega_{0}$, ${}_{i}\omega_{0} \equiv 0$ and $P_{M,0}\,{}_{a}\omega_{0} = {}_{a}\omega_{0}$, and from (25),(26),(57), and (59), we obtain alternative adjustment algorithms for the overlap-add algorithms I and II, respectively, as

$$ {}_{a}\omega_{k+1} = {}_{i}\omega_{k+1} + {}_{ii}\omega_{k+1} \tag{63} $$

where

$$ {}_{i}\omega_{k+1} = P_{M,0}({}_{i}\omega_{k} + \mu_{a}\bar{X}_{k\,a}e_{k}) \tag{64a} $$

$$ {}_{ii}\omega_{k+1} = U_{M,0}({}_{ii}\omega_{k} + \mu_{a}\bar{X}_{k-1\,a}e_{k}) \quad \left.\begin{matrix}\\ \end{matrix}\right\} \text{overlap-add I} \tag{64b} $$

$$ {}_{i}\omega_{k+1} = P_{M,0}({}_{i}\omega_{k} + \mu_{a}\bar{X}_{k\,a}e_{k}) \tag{65a} $$

$$ {}_{ii}\omega_{k+1} = {}_{ii}\omega_{k} + \mu\, V_{M,0}({}_{a}\bar{Y}_{k-1\,a}e_{k}) \quad \left.\begin{matrix}\\ \end{matrix}\right\} \text{overlap-add II} \tag{65b} $$

It is noted in (65b) that the algorithm of the form ${}_{ii}\omega_{k+1} = V_{M,0}({}_{ii}\omega_{k} + \mu_{a}X_{k-1\,a}e_{k})$ is not possible, in which case ${}_{ii}\omega_{k+1} = \mu V_{M,0}({}_{a}X_{k-1\,a}e_{k})$ since $V_{M,0}^{n} \equiv 0$ for $n \geq 2$.

C. Unconstrained FBLMS (UFBLMS) Algorithms

Based on the representation shown in (63)-(65), some reduced forms, in which $P_{M,0}$ and $U_{M,0}$ are eliminated, can be considered first. Then, recombining the results according to the filter-bank property of Section IV-A, we obtain the UFBLMS algorithms as follows.

Overlap-save :

$$ {}_{s}\omega_{k+1} = {}_{s}\omega_{k} + \mu_{s}\bar{X}_{k\,s}e_{k} \tag{66} $$

Overlap-add I :

$$ {}_{a}\omega_{k+1} = {}_{a}\omega_{k} + \mu\,({}_{a}x_{k\,a}e_{k} + {}_{a}\bar{X}_{k-1\,a}e_{k}) \tag{67} $$

Overlap-add II :

$$ {}_{a}\omega_{k+1} = {}_{a}\omega_{k} + \mu\,[{}_{a}\bar{X}_{k\,a}e_{k} + \mu\, V_{M,0}({}_{a}\bar{X}_{k-1\,a}e_{k})] \tag{68} $$

In a similar way, the self-orthogonalizing versions for the UFBLMS algorithms in (66)-(68) can be obtained from (45) and (48). Based on the results obtained so far, it can also be shown that the UFBLMS version of (30) becomes

$$ {}_{s}\omega_{k+1} = {}_{s}\omega_{k} + \mu N\,E[{}_{s}X_{k}(F^{-1}\,{}_{s}e_{k})]. \tag{69} $$

The use of the NTT's in realization of (69) yields a BLMS algorithm with further reduced complexity.

As a result of removing the constraints, we can reduce two FFT operations for each sectioning matrix in the fast convolution realization of the FBLMS ADF's. However, one may claim that these computational savings will be traded for less accurate performance. One of the objectives of Part II is to investigate this aspect. It will be seen in Part II that in many applications, the use of the UFBLMS algorithms yields no significant degradation in performance.

V. COMPUTER SIMULATION RESULTS AND DISCUSSION

In this section we present the results of computer simulation, showing the effects of different parameter values and different algorithms on the convergence behaviors of the BLMS ADF's. Our discussion will focus on the convergence characteristics of the self-orthogonalizing FBLMS algorithms. The unconstrained algorithms will be discussed in detail in Part II.

For the computer simulation, we have simulated the same adaptive equalizer and channal characteristics used in our earlier work [35] (see also [43]). Several parameter values chosen for the simulation are as follows: The input signal-to-noise ratio was 30 dB; the eigenvalue ratio of the input correlation matrix was 21; the input signal power (σ_x^2) was 1.001, and the number of squared-error data used for computing MSE was 500.

Let us first discuss choosing appropriate values of the smoothing constant β and the initial values of the power estimates $\{{}_s\hat{P}_{o,i}\}_{i=0}^{N-1}$ To see the effects of different parameter values on the convergence behavior, we did computer simulation of the self-orthogonalizing FBLMS algorithm given in (45) and (46). In the simulation, we chose the values of $\{{}_s\hat{P}_{o,i}\}_{i=0}^{N-1}$ to be P_o for all values of i. As can be seen in Fig. 2 and Figs. 4 and 5 in [35], the use of proper values for β and P_o is even more important in the FBLMS ADF than in the FLMS ADF. According to the results of the FLMS ADF, the values of P_o in a relatively broad range were acceptable. However, it appears from Fig. 2(a) and (c) that the use of $\beta=0.9$ does not produce good transient behavior. When the block length is large, this problem becomes more serious, and thus, the transient behaviors deviate significantly depending on P_o. The major reason for this phenomenon is that, unlike in the FLMS ADF, the frequency-domain power of the FBLMS ADF is estimated on the block-by-block basis. One solution to mitigate this block effect is to use a reduced value of β in order to have faster estimation. According to our simulation results, $\beta=0.8$ would be a good choice. (We use this value throughout Parts I and II.) However, a more difficult problem appears to be how to choose P_o. We found that given β, an appropriate value of P_o depends on the values of γ, N and σ_x^2. Since it is difficult to design analytically, we use an empirical value of $P_o=\frac{1}{4}\gamma N^2\sigma_x^2$ which can be a good choice for the case of L=M. The values of 2.5 and 10.0 in Fig. 2(b) and (d), respectively, correspond to the empirical P_o we mentioned.

We now discuss the results of computer simulation of different algorithms whose convergence factors are chosen according to (49)-(51). Fig. 3 verifies the relation between the convergence factors α and γ of the frequency-domain self-orthogonalizing algorithms. It is interesting to see that the convergence speed of the FLMS algorithm has been improved by using the block-type FBLMS algorithm. (It should be noted that the learning curve of the FLMS ADF in Fig. 3 was the best one obtainable by varying β and P_o.) This improvement will be explained theoretically in Part II. Fig. 4 compares the convergence behaviors of the two self-orthogonalizing FBLMS ADF's using overlap-save and overlap-add of the two self-orthogonalizing FBLMS ADF's using overlap-save and overlap-add sectioning methods, respectively. The overlap-add algorithm is based on (48) and uses the same convergence factor γ as the overlap-save algorithm does. It is noted from Fig. 4 that the steady-state MSE of the overlap-add algorithm is slightly worse than that of the overlap-save algorithm for the same parameter values. (This aspect will also be analyzed in Part II.)

Fig. 5 verifies the convergence factor relations in (49)-(51). Also, it can be seen from this figure that the self-orthogonalization in the frequency domain significantly improves the convergence speed of the LMS algorithm. We also did computer simulation to see the dif-

ference in the convergence behaviors of the FBLMS ADF's using the two different power estimation methods in (46b) and (47). In general, the FBLMS ADF using the sample average method in (47) showed very unstable convergence behavior even for moderate values of α. For example, it does not converge for α=0.01 for which value the FBLMS ADF using a one-pole LPF converges. The convergence behaviors for a decreased α =0.001 are shown in Fig. 6. It should be noted from this figure that the sample average method still shows large variation (i.e., spurious peaks) in its learning curve. (Note that both learning curves were obtained by ensemble averaging the same number of squared-error data.) Finally, it is noted that although the BLMS ADF can be implemented using the FFT, the stability region of the BLMS ADF is narrower than the LMS ADF. Specifically, the convergence factor μ of the BLMS ADF must be less than $1/(L\lambda_{max})$ for stability where λ_{max} is the largest eigenvalue of the autocorrelation matrix. λ_{max} can be, in general, far greater than 1 when $\sigma_x^2 \gg 1$. As seen in Fig. 7, the self-orthogonalizing algorithm which reduces λ_{max} significantly improves the stability of the BLMS ADF.

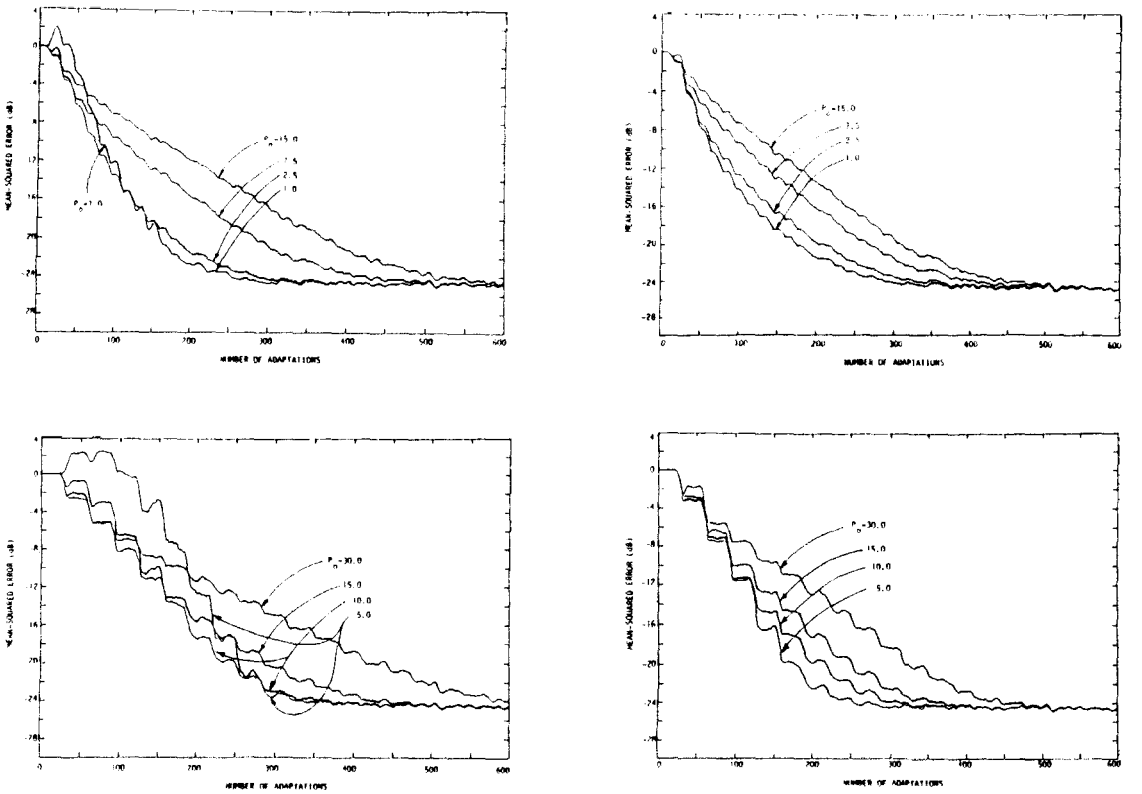


Fig. 2. Effects of different values of β, P_0 and L on the transient behavior of the self-orthogonalizing FBLMS ADF (M=16, [illegible] N and α = 0.01).
(a) β =0.9, L=16 and N=32.
(b) β =0.8 L=16 and N=32.
(c) β =0.9, L=32 and N=64.
(d) β =0.8, L=32 and N=64.

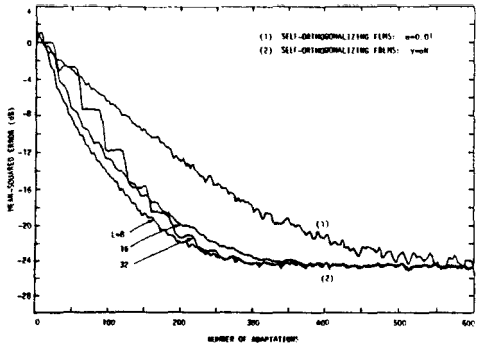


Fig. 3. Convergence behaviors of the self-orthogonalizing FLMS and FBLMS ADF's with different block lengths (M=16).

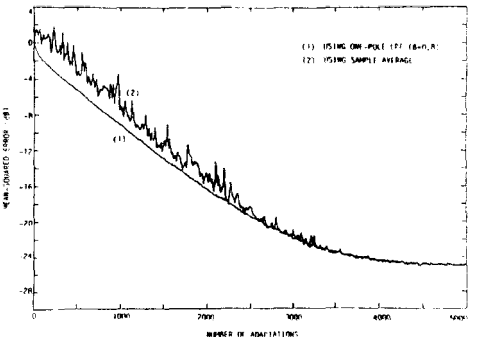


Fig. 6. Comparison of the convergence behaviors of the self-orthogonalizing FBLMS ADF's for different power estimation methods (M=16, L=16, N=32, $\gamma = \alpha N$ and α=0.001).

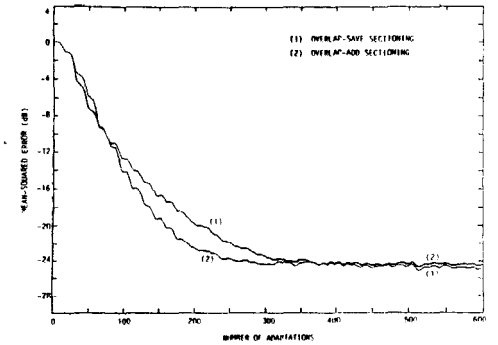


Fig. 4. Comparison of the convergence behaviors of the self-orthogonalizing FBLMS ADF's for different sectioning methods (M=16, L=16, N=32 and α=0.01).

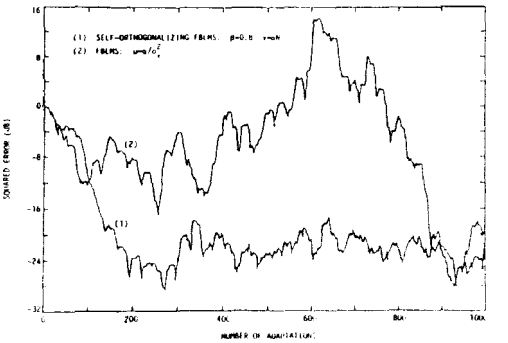


Fig. 7. A stabilized operation of the self-orthogonalizing FBLMS ADF(M=16, L=16, N=32, α=0.04 and σ_x^2 =1.001).

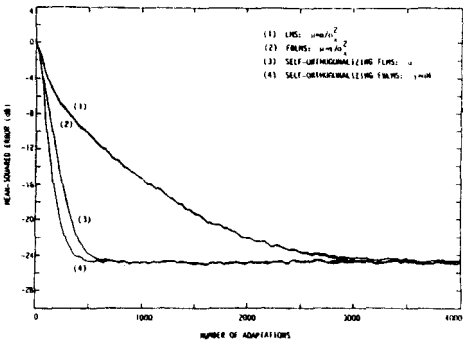


Fig. 5. Comparison of the convergence behaviors of the (overlap-save) FBLMS ADF's with and without self-orthogonalization in the frequency domain (M=16, L=16, N=32 and σ_x^2 =1.001).

VI. CONCLUSIONS

Based on a unified matrix treatment, we have discussed the overlap-save and overlap-add sectioning methods for the fast convolution realization of the TBLMS and FBLMS ADF's. As a result, we have obtained new results of the overlap-add realization and the number-theoretic transform realization of the FBLMS ADF's. We hav also studied the FWBLMS and self-orthogonalizing FBLMS algorithms by formulating them directly in the frequency domain. Then, it has been shown that the TBLMS ADF can be made to have the same convergence behavior as that of the FBLMS ADF. Next,

we have discussed possible UFBLMS algorithms in a unified treatment of the overlap-save and overlap-add sectioning methods. Finally, based on the results of simulation, we have discussed choosing the value of β and the initial value of the power estimates. Our simulation results indicate that if a proper convergence factor is used, the self-orthogonalizing FBLMS ADF can converge to the same steady-state MSE regardless of block length and the FBLMS ADF has improved convergence speed compared to the FLMS ADF.

REFERENCES

1. B. Widrow and S.D. Stearns, *Adaptive Signal Processing*. Englewood Cliffs, NJ: Prentice-Hall, 1985.
2. M.L. Honig and D.G. Messerschmitt, *Adaptive Filters*. Boston, MA: Kluwer, 1984.
3. S. Haykin, *Adaptive Filter Theory*. Englewood Cliffs, NJ: Prentice-Hall, 1986.
4. C.F.N. Cowan and P.M. Grant, Eds. *Adaptive Filters*. Englewood Cliffs, NJ: Prentice-Hall, 1985.
5. R.W. Lucky, J. Salz, and E.J. Weldon, Jr. *Principles of Data Communication*. New York: McGraw-Hill, 1968.
6. R.A. Monzingo and T.W. Miller, *Introduction to Adaptive Arrays*. New York: Wiley, 1980.
7. B. Widrow et al., "Stationary and nonstationary learning characteristics of the LMS adaptive filter," Proc. IEEE, vol. 64, pp.1151-1162, Aug. 1976.
8. T. Walzman and M. Schwartz, "Automatic equalization using the discrete frequency domain," IEEE Trans. Inform. Theory, vol. IT-19, pp.59-68, Jan. 1973.
9. D. Maiwald, H.P. Kaeser, and F. Closs, "An adaptive equalizer with significantly reduced number of operations," in Proc. 1978 IEEE Int. Conf. Acoust., Speech, Signal Processing, Apr. 2-4, 1978, pp.100-104.
10. M.J. Dentino, J. McCool, and B. Widrow, "Adaptive filtering in the frequency domain," Proc IEEE, vol. 66, pp.1658-1659, Dec 1978
11. M.J. Dentino, J. McCool, and B. Widrow, "Adaptive filtering in the frequency domain," Proc. IEEE, vol. 66, pp.1658-1659, Dec. 1978.
11. N.J. Bershad and P.L. Feintuch, "Analysis of the frequency domain adaptive filter," Proc. IEEE, vol. 67, pp.1658-1659, Dec. 1979.
12. G.A. Clark, S.K. Mitra, and S.R. Parker, "Block adaptive filtering," in Proc. 1980 IEEE Int. Symp. Circuits Syst., Houston, TX, Apr. 1980, pp.384-387.
13. E.R. Ferrara, Jr., "Fast implementation of LMS adaptive filter," IEEE Trans. Acoust., Speech, Signal Processing, vol. ASSP-28, pp.474-475, Aug. 1980.
14. S.S. Narayan and A.M. Peterson, "Frequency domain least-mean-square algorithm," Proc. IEEE, vol. 60, 69, pp.124-126, Jan. 1981.
15. G.A. Clark, "Block adaptive filtering and its application to seismic event detection," Ph.D. dissertation, Univ. California, Santa Babara, CA, Apr. 1981.
16. G.A. Clark, S.K. Mitra, and S.R. Parker, "Block implementation of adaptive digital filters," IEEE Trans. Acoust., Speech, Signal Processing, vol. ASSP-29, pp.744-752, June 1981.
17. F.A. Reed and P.L. Feintuch, "A comparison of LMS adaptive cancellers implemented in the frequency domain and the time domain," IEEE Trans. Acoust., Speech, Signal Processing, vol. ASSP-29, pp.770-775, June 1981.
18. R.R. Bitmead and B.D.O. Anderson, "Adaptive frequency sampling filters," IEEE Trans. Acoust., Speech, Signal Processing, vol. ASSP-29, pp.684-694, June 1981.
19. G.A. Clark, S.R. Parker, and S.K. Mitra, "Efficient realization of adaptive digital dilters in the time and frequency domains," in Proc. 1982 IEEE Int. Conf. Acoust., Speech, Signal Processing, Paris, France, May 3-5, 1982, pp.1345-1348.
20. D. Mansour and A.H. Gray, Jr., "Performance characteristics of the unconstrained frequency-domain adaptive filter," in Proc. 1982 IEEE Int. Symp. Circuits Syst., May 1982, pp.695-698.
21. D. Mansour and A.H. Gray, Jr., "Unconstrained frequency-domain adaptive filter," IEEE Trans. Acoust., Speech, Signal Processing, vol. ASSP-30, pp.726-734, Oct. 1982.
22. J.C. Lee, "A class of adaptive digital filters and their applications," Ph.D. dissertation, Dept. Elec. Eng., Korea Advanced Inst. Sci. Technol., Seoul, Korea, June 1983.
23. S.S. Narayan, A.M. Peterson, and M.J. Narasimha, "Transform domain LMS algorithm," IEEE Trans Acoust., Speech, Signal Processing, vol. ASSP-31, pp.609-615, June 1983.
24. G.A. Clark, S.R. Parker, and S.K. Mitra, "A unified approach to time- and frequency-domain realiza-

tion of FIR adaptive digital filters," IEEE Trans. Acoust., Speech, Signal Processing, vol. ASSP-31, pp.1073-1083, Oct. 1983.

25. J.C. Ogue, T. Saito, and Y. Hoshiko, "A fast convergence frequency domain adaptive filter," IEEE Trans. Acoust., Speech, Signal Processing, vol. ASSP-31, pp.1312-1314, Oct. 1983.
26. G. Picchi and G. Prati, "Self-orthogonalizing adaptive equalization in the discrete frequency domain," IEEE Trans. Commun., vol. COM-32, pp.371-379, Apr. 1984.
27. A. Morgül, P.M. Grant, and C.F.N. Cowan, "Wideband hybrid analog/digital frequency domain adaptive filter," IEEE Trans. Acoust., Speech, Signal Processing, vol. ASSP-32, pp.762-769, Aug. 1984.
28. J.C. Lee and C.K. Un, "A reduced structure of the frequency-domain block LMS adaptive digital filter," Proc. IEEE, vol. 72, pp.1816-1818, Dec. 1984.
29. E.R. Ferrara, Jr., "Frequency domain adaptive filtering," in *Adaptive Filters*, C.F.N. Cowan and P.M. Grant, Eds. Englewood Cliffs, NJ: Prentice-Hall, 1985.
30. F.A. Reed, P.L. Feintuch, and N.J. Bershad, "The application of the frequency domain LMS adaptive filer to split array bearing estimation with a sinusoidal signal," IEEE Trans. Acoust., Speech, Signal Processing, vol. ASSP-33, pp.61-69, Feb. 1985.
31. J.C. Lee, C.K. Un, and D.H. Cho, "A frequency-weighted block LMS algorithm and its application to speech processing," Proc. IEEE, vol. 73, pp. 1137-1138, June 1985.
32. E.R. Ferrara, Jr., "Frequency-domain implementations of periodically time-varying filters," IEEE Trans. Acoust., Speech, Signal Processing, vol. ASSP-33, pp.883-892, Aug. 1985.
33. J.C. Lee, B.K. Min, and M. Suk, "Realization of adaptive digital filters using the Fermat number transform," IEEE Trans. Acoust., Speech, Signal Processing, vol. ASSP-33, pp.1036-1039, Aug. 1985.
34. J.C. Lee and C.K. Un, "Block realization of multirate adaptive digital filters," IEEE Trans. Acoust., Speech, Signal Processing, vol. ASSP-34, pp.105-117, Feb. 1986.
35. J.C. Lee and C.K. Un, "Performance of transform-domain LMS adaptive digital filters," IEEE Trans. Acoust., Speech, Signal Processing, vol. ASSP-34, pp.499-510, June 1986.
36. A.V. Oppenheim and R.W. Schafer, *Digital Signal Processing*. Englewood Cliffs, NJ: Prentice-Hall, 1975.
37. H.C. Andrews and B.R. Hunt, *Digital Image Restoration*. Englewood Cliffs, NJ: Prentice-Hall, 1977.
38. R.C. Agarwal and C.S. Burrus, "Number theoretic transforms to implement fast digital convolution," Proc. IEEE, vol. 63, pp.550-560, Apr., 1975.
39. J.C. Lee and C.K. Un, "On the interrelationships among a class of convolutions," IEEE Trans. Acoust., Speech, Signal Processing, vol. ASSP-32, pp.1245-1247, Dec. 1984.
40. J.S. Lim, Ed., *Speech Enhancement*. Englewood Cliffs, NJ: Prentice-Hall, 1983.
41. R.D. Gitlin and F.R. Magee, Jr., "Self-orthogonalizing adaptive equalization algorithms," IEEE Trans. Commun., vol. COM-25, pp.666-672, July 1977.
42. B. Widrow and E. Walach, "On the statistical efficiency of the LMS algorithm with nonstationary inputs," IEEE Trans. Inform. Theroy, vol. IT-30, pp.211-221, March 1984.
43. E.H. Satorius and S.T. Alexander, "Channel equalization using adaptive lattice algorithms," IEEE Trans. Commun., vol. COM-27, pp.899-905, June 1979.

▲ **Jae Chon Lee** was born in Seoul, Korea, in 1954. He received the B.S. degree with honors in electronics engineering from Seoul National University (SNU), Seoul, in 1977, and the M.S. and Ph.D. degrees in electrical engineering from the Korea Advanced Institute of Science and Technology (KAIST), Seoul, in 1979 and 1983, respectively.

Since August 1983, he has been with KAIST as a senior research scientist. During the academic years 1984-1985 and 1985-1986, he was on leave at the Massachusetts Institute of Technology (MIT) and the University of California at Santa Barbara (UCSB), working with Prof. Jae Lim and Prof. Sanjit Mitra, respectively. His research interest is in the area of digital signal processing and digital communication systems including adaptive filtering and neural computing.

▲ **Dr. Chong Kwan Un** received the B.S., M.S., and Ph.D. degrees in electrical engineering from the University of Delaware, Newark, Delaware, in 1964, and 1969, respectively. From 1969 to 1973 he was Assistant Professor of Electrical Engineering at the University of Maine, Portland, where he taught communications and did research on synchronization problems. In May 1973, he joined the Staff o the Telecommunication Sciences Center, SRI International, Menlo Park, CA, where he did research on voice digitization and bandwidth compression systems. Since June 1977 he has been with KAIST, where he is Professor of Electrical Engineering and Head of the Communications Research Laboratory, teaching and doing research in the areas of digital communications and digital signal processing. He has authored or coauthored over 140 papers and 50 reports in speech coding and processing, packet voice/data transmission, synchronization, and digital filtering. Also, he holds 6 patents granted or pending. From February 1982 to June 1983 he served as Dean of Engineering at KAIST.

Dr. Un has received a number of awards including the 1976 Best Paper Award from the IEEE Communications Society, the National Order of Merits (Dong Baik Jang) from the Government of Korea, and Achievement Awards from KITE, KICS and ASK. He is a Fellow of IEEE and a member of Tau Beta Pi and Eta Kappa Nu Honor Societies.