An Embedding Approach to Frequency-Domain and Subband Adaptive Filtering

Ricardo Merched, *Student Member, IEEE*, and Ali H. Sayed, *Senior Member, IEEE*

***Abstract*—Frequency-domain and subband implementations improve the computational efficiency and the convergence rate of adaptive schemes. The well-known multidelay adaptive filter (MDF) belongs to this class of block adaptive structures and is a DFT-based algorithm. In this paper, we develop adaptive structures that are based on the trigonometric transforms DCT and DST and on the discrete Hartley transform (DHT). As a result, these structures involve only real arithmetic and are attractive alternatives in cases where the traditional DFT-based scheme exhibits poor performance. The filters are derived by first presenting a derivation for the classical DFT-based filter that allows us to pursue these extensions immediately. The approach used in this paper also provides further insights into subband adaptive filtering.**

***Index Terms*—Circulant, DCT, DFT, DHT, DST, embedding, Hankel, real arithmetic, subband adaptive filter, Toeplitz.**

I. INTRODUCTION

COMPUTATIONAL complexity is a burden in applications that require long tapped-delay adaptive structures, such as echo cancellation, where filters with hundreds or even thousands of taps are necessary to model the echo path. Frequency-domain and subband adaptive filters have been proposed to reduce the computational requirements inherent to such applications (see, e.g., [1]–[4]). These techniques not only result in more efficient structures (due to the use of efficient block signal processing methods), but they also improve the convergence rate of an adaptive algorithm (due to a decrease in the eigenvalue spread of the correlation matrix of the transformed signals). A well-known example is the multidelay adaptive filter (MDF) [2], which relies on the use of the discrete-Fourier transform (DFT); it is a more general implementation than the original frequency-domain algorithm proposed in [1] since the adaptive filters are allowed to have more than one coefficient in the subbands.

The MDF structure has been derived in the literature in the DFT domain only. However, one would expect that different frequency domain transformations (other than the DFT) can result in different levels of performance (both computationally and otherwise) since performance is highly dependent on both the statistical properties of the input signals and on the nature of the frequency transformations. This fact motivates us to develop, in this paper, frequency-domain adaptive structures that are based on the trigonometric transforms DCT and DST, as well as on the discrete Hartley transform (DHT). It seems that the traditional derivations of frequency-domain adaptive filters cannot be directly extended to these new signal transformations without some effort. For this reason, we will first present a derivation for the classical DFT-based MDF structure using a so-called *embedding* approach. This approach will then allow us to pursue the new extensions rather immediately by exploiting different kinds of matrix structure (e.g., [5]–[7]).

The MDF schemes of this paper are attractive for applications where real arithmetic is required. Moreover, since efficient algorithms exist for computing the DCT, DST, and DHT (see, e.g., [8]), these schemes also lead to efficient adaptive filter structures. We will further present in Section IX examples where they can lead to better performance than the DFT-based scheme. We should mention that these new structures are distinct from the so-called transform domain algorithms (as, e.g., in [9]), which process the data on a *sample by sample* basis. The frequency domain structures, on the other hand, perform *block-by-block* processing, which is essential for efficient frequency-domain implementations.

In this paper, we also clarify the connection between the MDF structure and the more general subband adaptive filtering structure. The key point to note is that the derivation of a subband adaptive scheme can be carried out in much the same way as that of the MDF structure. We will further relate the MDF structure to the concept of delayless subband adaptive filtering proposed in [10]. In this reference, a mapping from subband to wideband filter coefficients was proposed with the intent of resolving the delay problem that is characteristic of subband adaptive filters. We will derive an alternative explicit mapping that guarantees optimal performance; we will also comment on the results in [11] for open-loop schemes. In particular, we will show that an adaptive subband scheme that is based on a maximally decimated DFT filter bank can be developed in a way analogous to the MDF.

The paper is organized as follows. We start by formulating a generic block estimation problem. We then show in Section III how the pseudocirculant structure of the optimal block filter can be exploited to provide a new embedding-based derivation of the MDF filter. In later sections, we show how the embedding approach can be extended to other classes of signal transformations (DCT, DST, DHT). In Section VI, we compare the computational requirements of the different algorithms, and in Section VII, we provide some simulations. The examples show

Manuscript received July 28, 1999; revised May 8, 2000. This work was supported in part by the National Science Foundation under Award CCR-9732376, the Army Research Office under Grant DAAH04-96-1-0176-P00005, and the University of California Core project CR 98-19. The work of R. Merched was also supported by a fellowship from CAPES, Brazil. The associate editor coordinating the review of this paper and approving it for publication was Dr. Vikram Krishnamurthy.

The authors are with the Department of Electrical Engineering, University of California, Los Angeles, CA 90095 USA (e-mail: sayed@ee.ucla.edu; URL: http://www.ee.ucla.edu/asl).

Publisher Item Identifier S 1053-587X(00)06659-9.

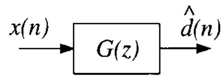


Fig. 1. Scalar linear optimal estimation.

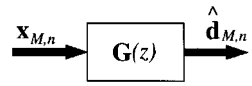


Fig. 2. Block linear optimal estimation.

situations where the new structures lead to better performance than the DFT-based MDF structure. We conclude the paper with a discussion of the connection of the MDF technique to subband adaptive filtering.

II. Block Estimation Problem

We start by formulating a basic estimation problem. Thus, consider two jointly wide-sense stationary (WSS) and zero-mean random sequences $\{x(n), d(n)\}$. Let $\hat{d}(n)$ denote the linear least mean squares estimator of $d(n)$ given the N observations

$$\boldsymbol{x}_n \triangleq \text{col}\{x(n), x(n-1), \ldots, x(n-N+1)\}.$$

Then, $\hat{d}(n) = \boldsymbol{g}\boldsymbol{x}_n$, where $\boldsymbol{g}$ is a row vector with N taps (entries) and is given by $\boldsymbol{g} = \boldsymbol{R_{dx}}\boldsymbol{R_x}^{-1}$. The variances $\boldsymbol{R_{dx}}$ and $\boldsymbol{R_x}$ are defined by $\boldsymbol{R_{dx}} = Ed(n)\boldsymbol{x}_n^T$ and $\boldsymbol{R_x} = E\boldsymbol{x}_n\boldsymbol{x}_n^T$. The stationarity of the processes $\{d(n), x(n)\}$ guarantees that $\boldsymbol{R_{dx}}$ and $\boldsymbol{R_x}$ are independent of n. Hence, we can regard $\boldsymbol{g}$ as the tap vector of a time-invariant FIR filter with transfer function (see Fig. 1)

$$G(z) \triangleq \sum_{i=0}^{N-1} g_i z^{-i} \tag{1}$$

so that $\hat{d}(z) = G(z)x(z)$, where $x(z)$ and $\hat{d}(z)$ denote the z-transforms of the scalar sequences $\{x(n), \hat{d}(n)\}$, and the $\{g_i\}$ denote the individual entries of $\boldsymbol{g}$.

The estimates $\{\hat{d}(n)\}$ can be alternatively computed on a block by block basis by using block digital filtering techniques. To this end, introduce the M-long data vectors

$$\boldsymbol{x}_{M,n} \triangleq \text{col}\{x(Mn), x(Mn-1), \ldots, x(Mn-M+1)\}$$
$$\boldsymbol{d}_{M,n} \triangleq \text{col}\{d(Mn), d(Mn-1), \ldots, d(Mn-M+1)\}$$

and let $\boldsymbol{x}_M(z)$ and $\boldsymbol{d}_M(z)$ denote their vector z-transforms:

$$\boldsymbol{x}_M(z) = \sum_{n=-\infty}^{\infty} \boldsymbol{x}_{M,n} z^{-n}, \quad \boldsymbol{d}_M(z) = \sum_{n=-\infty}^{\infty} \boldsymbol{d}_{M,n} z^{-n}.$$

Then, it can be verified that the block data $\{\boldsymbol{x}_{M,n}, \hat{\boldsymbol{d}}_{M,n}\}$ are related via (see Fig. 2) $\hat{\boldsymbol{d}}_M(z) = \boldsymbol{G}(z)\boldsymbol{x}_M(z)$, where the square transfer matrix $\boldsymbol{G}(z)$ has a pseudocirculant (PC) form,[1] i.e., it has the form (for $M = 3$)

$$\boldsymbol{G}(z) = \begin{bmatrix} g_0(z) & g_1(z) & g_2(z) \\ z^{-1}g_2(z) & g_0(z) & g_1(z) \\ z^{-1}g_1(z) & z^{-1}g_2(z) & g_0(z) \end{bmatrix}. \tag{2}$$

The functions $g_i(z)$, $i = 0, 1, \ldots, M-1$, of $\boldsymbol{G}(z)$ are the polyphase components of the wideband LTI filter $G(z)$ in (1), i.e., they correspond to N/M-long FIR filters that are defined by

$$\begin{cases} g_0(z) = g_0 + g_M z^{-1} + g_{2M} z^{-2} + \cdots \\ g_1(z) = g_1 + g_{M+1} z^{-1} + g_{2M+1} z^{-2} + \cdots \\ \qquad \vdots \end{cases}$$

Due to its pseudocirculant structure, the matrix $\boldsymbol{G}(z)$ can be factored as $\boldsymbol{G}(z) = \boldsymbol{P}(z)\boldsymbol{Q}(z)$, where $\boldsymbol{P}(z)$ is an $M \times (2M-1)$ matrix function with Toeplitz structure, e.g., for $M = 3$

$$\boldsymbol{P}(z) = \begin{bmatrix} g_0(z) & g_1(z) & g_2(z) & 0 & 0 \\ 0 & g_0(z) & g_1(z) & g_2(z) & 0 \\ 0 & 0 & g_0(z) & g_1(z) & g_2(z) \end{bmatrix} \tag{3}$$

and $\boldsymbol{Q}(z)$ is a $(2M-1) \times M$ matrix with a leading identity block and a lower block with shifts, say, for $M = 3$ again

$$\boldsymbol{Q}(z) = \left[\begin{array}{ccc} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ \hline z^{-1} & 0 & 0 \\ 0 & z^{-1} & 0 \end{array}\right]. \tag{4}$$

We will exploit the factorization $\boldsymbol{G}(z) = \boldsymbol{P}(z)\boldsymbol{Q}(z)$ heavily in the sequel.

III. DFT-Based Adaptive Structure

In this section, we show how the pseudocirculant structure of $\boldsymbol{G}(z)$ can be exploited to derive a well-known frequency-domain adaptive filter that relies on the DFT and is known in the literature as the multidelay adaptive filter [1]–[3]. The original derivation of this structure is different from the approach we present in this section. Our derivation is based on exploiting, in a direct way, the PC nature of $\boldsymbol{G}(z)$. As a fallout, the argument will suggest immediate extensions that rely on other signal transformations [such as the real trigonometric transforms DCT and DST, and the Hartley transform (DHT); see Sections VI and VII].

Since we deal with the DFT in this section, and since it is usually desirable to work with sequences whose lengths can be expressed as powers of 2, we find it convenient to redefine the above matrices $\boldsymbol{P}(z)$ and $\boldsymbol{Q}(z)$ as

$$\boldsymbol{P}(z) = \begin{bmatrix} g_0(z) & g_1(z) & g_2(z) & 0 & 0 & 0 \\ 0 & g_0(z) & g_1(z) & g_2(z) & 0 & 0 \\ 0 & 0 & g_0(z) & g_1(z) & g_2(z) & 0 \end{bmatrix} \tag{5}$$

and

$$\boldsymbol{Q}(z) = \left[\begin{array}{ccc} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ \hline z^{-1} & 0 & 0 \\ 0 & z^{-1} & 0 \\ 0 & 0 & z^{-1} \end{array}\right] \tag{6}$$

[1] A pseudocirculant matrix function $\boldsymbol{G}(z)$ is essentially a circulant matrix function with the exception that all the entries below the main diagonal are further multiplied by z^{-1}; see (2) and [12].

with an additional zero column added to $\boldsymbol{P}(z)$ and an additional row added to $\boldsymbol{Q}(z)$. The product $\boldsymbol{P}(z)\boldsymbol{Q}(z)$ is, of course, still equal to $\boldsymbol{G}(z)$. Now, however, $\boldsymbol{P}(z)$ is $M \times 2M$, and $\boldsymbol{Q}(z)$ is $2M \times M$ ($2M$ will be a power of 2 when M is).

Let $[\boldsymbol{F}]_{kl} = (1/\sqrt{2M})e^{-2\pi jkl/2M}$ denote the DFT matrix of size $2M \times 2M$. We start by embedding the $M \times 2M$ Toeplitz matrix $\boldsymbol{P}(z)$ into a $2M \times 2M$ circulant matrix function $\boldsymbol{C}(z)$ (a similar technique was used in [13] to propose efficient structures for block digital filtering), say, for $M = 3$

$$\boldsymbol{C}(z) = \left[\begin{array}{cccccc} g_0(z) & g_1(z) & g_2(z) & 0 & 0 & 0 \\ 0 & g_0(z) & g_1(z) & g_2(z) & 0 & 0 \\ 0 & 0 & g_0(z) & g_1(z) & g_2(z) & 0 \\ \hline 0 & 0 & 0 & g_0(z) & g_1(z) & g_2(z) \\ g_2(z) & 0 & 0 & 0 & g_0(z) & g_1(z) \\ g_1(z) & g_2(z) & 0 & 0 & 0 & g_0(z) \end{array}\right] \tag{7}$$

so that $\boldsymbol{P}(z) = [\boldsymbol{I}_M\ \boldsymbol{0}]\boldsymbol{C}(z)$, where $\boldsymbol{I}_M$ is the $M \times M$ identity matrix, and $\boldsymbol{0}$ is the $M \times M$ null matrix.

Now, it is well known that a circulant matrix such as $\boldsymbol{C}(z)$ can be diagonalized by the DFT matrix $\boldsymbol{F}$, i.e., it always holds that

$$\boldsymbol{C}(z) = \boldsymbol{F}^* \boldsymbol{W}(z) \boldsymbol{F} \tag{8}$$

for some diagonal matrix

$$\boldsymbol{W}(z) = \text{diag}\{w_0(z), \ldots, w_{2M-1}(z)\}$$

and where $*$ denotes complex conjugate transposition. Each $w_i(z)$ has N/M taps. Using (8) and the fact that $\boldsymbol{F}$ is symmetric, it is easy to verify that the entries of the first row of $\boldsymbol{C}(z)$ (e.g., for $M = 3$) can be recovered from the $\{w_i(z)\}$ via

$$\begin{bmatrix} g_0(z) \\ g_1(z) \\ g_2(z) \\ 0 \\ 0 \\ 0 \end{bmatrix} = \boldsymbol{F} \begin{bmatrix} w_0(z) \\ w_1(z) \\ w_2(z) \\ w_3(z) \\ w_4(z) \\ w_5(z) \end{bmatrix}. \tag{9}$$

This relation shows that not every diagonal matrix $\boldsymbol{W}(z)$ in (8) will result in a circulant matrix $\boldsymbol{C}(z)$ of the form (7). This is because the transformation (9) requires that the $\{w_i(z)\}$ should be such that the last entries of the transformed vector are zero. We will use this constraint at the end of this section to derive the so-called *constrained* MDF adaptive structure.

We can now write $\boldsymbol{G}(z) = \boldsymbol{P}(z)\boldsymbol{Q}(z)$ in the form

$$\boldsymbol{G}(z) = [\boldsymbol{I}_M\ \boldsymbol{0}]\boldsymbol{C}(z)\boldsymbol{Q}(z) = [\boldsymbol{I}_M\ \boldsymbol{0}]\boldsymbol{F}^*\boldsymbol{W}(z)\boldsymbol{F}\boldsymbol{Q}(z). \tag{10}$$

The block estimation error $\boldsymbol{e}_{M,n} = \boldsymbol{d}_{M,n} - \hat{\boldsymbol{d}}_{M,n}$ is then, in the z-transform domain, given by

$$\begin{aligned} \boldsymbol{e}_M(z) &= \boldsymbol{d}_M(z) - \boldsymbol{G}(z)\boldsymbol{x}_M(z) \\ &= \boldsymbol{d}(z) - [\boldsymbol{I}_M\ \boldsymbol{0}]\boldsymbol{F}^*\boldsymbol{W}(z) \underbrace{\boldsymbol{F}\boldsymbol{Q}(z)\boldsymbol{x}_M(z)}_{\boldsymbol{x}'_M(z)}. \end{aligned} \tag{11}$$

Now, define the $2M \times 1$ signal $\boldsymbol{x}'_{M,n}$ with transform

$$\begin{aligned} \boldsymbol{x}'_M(z) &\triangleq \boldsymbol{F}\boldsymbol{Q}(z)\boldsymbol{x}_M(z) \\ &\triangleq \text{col}\{x'_0(z), x'_1(z), \ldots, x'_{2M-1}(z)\} \end{aligned}$$

and denote its individual entries by $x'_i(z)$. Then

$$\boldsymbol{W}(z)\boldsymbol{x}'_M(z) = \text{col}\{w_0(z)x'_0(z), \ldots, w_{2M-1}(z)x'_{2M-1}(z)\}.$$

Fig. 3 illustrates the decomposition in (11).

Let $\{\boldsymbol{w}_i\}$ denote the (column) tap vectors that correspond to the $\{w_i(z)\}$; each $\boldsymbol{w}_i$ has length N/M. Then, the output of each term $w_i(z)x'_i(z)$ at a certain time instant n can be obtained as the inner product $\boldsymbol{x}'_{n,i}\boldsymbol{w}_i$, where $\boldsymbol{x}'_{n,i}$ denotes the state (row) vector corresponding to $\boldsymbol{w}_i$ at time n and is given by

$$\boldsymbol{x}'_{n,i} = \left[x'_i(n) \quad x'_i(n-1) \quad \ldots \quad x'_i\left(n - \frac{N}{M} + 1\right) \right].$$

Here, $x'_i(n)$ denotes the ith entry of the vector $\boldsymbol{x}'_{M,n}$. Define the $2M \times 2N$ block diagonal matrix of regression vectors at time n

$$\overline{\boldsymbol{X}}(n) = \text{diag}\{\boldsymbol{x}'_{n,0}, \boldsymbol{x}'_{n,1}, \ldots, \boldsymbol{x}'_{n,2M-1}\}$$

and the following $2N \times 1$ column vector of unknown weight vectors that we wish to determine:

$$\overline{\boldsymbol{\mathcal{W}}} = \text{col}\{\boldsymbol{w}_0, \boldsymbol{w}_1, \ldots, \boldsymbol{w}_{2M-1}\}.$$

It then follows from the error equation (11) that in the time domain

$$\boldsymbol{e}_{M,n} = \boldsymbol{d}_{M,n} - [\boldsymbol{I}_M\ \boldsymbol{0}]\boldsymbol{F}^*\overline{\boldsymbol{X}}_n\overline{\boldsymbol{\mathcal{W}}}.$$

An LMS-based adaptive algorithm that recursively estimates the $\overline{\boldsymbol{\mathcal{W}}}$ is then given by

$$\overline{\boldsymbol{\mathcal{W}}}_{n+1} = \overline{\boldsymbol{\mathcal{W}}}_n + \mu\overline{\boldsymbol{X}}_n^*\boldsymbol{F}\begin{bmatrix}\boldsymbol{I}_M \\ \boldsymbol{0}\end{bmatrix}(\boldsymbol{d}_{M,n} - [\boldsymbol{I}_M\ \boldsymbol{0}]\boldsymbol{F}^*\overline{\boldsymbol{X}}_n\overline{\boldsymbol{\mathcal{W}}}_n)$$

where the regressor is taken as $[\boldsymbol{I}_M\ \boldsymbol{0}]\boldsymbol{F}^*\overline{\boldsymbol{X}}_n$. The above recursion can be rewritten more compactly as

$$\overline{\boldsymbol{\mathcal{W}}}_{n+1} = \overline{\boldsymbol{\mathcal{W}}}_n + \mu\overline{\boldsymbol{X}}_n^*\boldsymbol{e}'_{M,n} \tag{12}$$

where we introduced the $2M \times 1$ transformed error signal

$$\boldsymbol{e}'_{M,n} \triangleq \boldsymbol{F}\begin{bmatrix}\boldsymbol{I}_M \\ \boldsymbol{0}\end{bmatrix}(\boldsymbol{d}_{M,n} - [\boldsymbol{I}_M\ \boldsymbol{0}]\boldsymbol{F}^*\overline{\boldsymbol{X}}_n\overline{\boldsymbol{\mathcal{W}}}_n). \tag{13}$$

We will continue to write $\boldsymbol{e}_{M,n}$ to refer to the estimation error in the update equation, namely

$$\boldsymbol{e}_{M,n} = \boldsymbol{d}_{M,n} - [\boldsymbol{I}_M\ \boldsymbol{0}]\boldsymbol{F}^*\overline{\boldsymbol{X}}_n\overline{\boldsymbol{\mathcal{W}}}_n$$

with $\overline{\boldsymbol{\mathcal{W}}}$ replaced by $\overline{\boldsymbol{\mathcal{W}}}_n$.

Note, in particular, that the update for the estimate of the ith weight vector $\boldsymbol{w}_i$ is of the form (in terms of the ith entry of $\boldsymbol{e}'_{M,n}$ and the ith regression vector $\boldsymbol{x}'_{n,i}$)

$$\boldsymbol{w}_{n+1,i} = \boldsymbol{w}_{n,i} + \mu \cdot [\boldsymbol{x}'_{n,i}]^* e'_i(n).$$

This suggests an alternative way for rewriting the adaptive algorithm (12), where instead of collecting all unknown column weight vectors $\{\boldsymbol{w}_i\}$ into a single column vector $\overline{\boldsymbol{\mathcal{W}}}$, we collect their transposes into a block matrix of dimensions $2M \times \frac{N}{M}$. Thus, define

$$\boldsymbol{\mathcal{W}} = \begin{bmatrix} \boldsymbol{w}_0^T \\ \boldsymbol{w}_1^T \\ \vdots \\ \boldsymbol{w}_{2M-1}^T \end{bmatrix}$$

$$\boldsymbol{X}_n^T = [\,[\boldsymbol{x}'_{n,0}]^* \quad [\boldsymbol{x}'_{n,1}]^* \quad \cdots \quad [\boldsymbol{x}'_{n,2M-1}]^*\,]$$

and $\boldsymbol{E}_n = \text{diag}\{e'_0(n), \ldots, e'_{2M-1}(n)\}$. Here, T denotes matrix transposition. Then, the unconstrained frequency-domain adaptive filter becomes

$$\boldsymbol{\mathcal{W}}_{n+1} = \boldsymbol{\mathcal{W}}_n + \mu \boldsymbol{\Lambda}_n^{-1} \boldsymbol{E}_n \boldsymbol{X}_n \tag{14}$$

where we further introduced a $2M \times 2M$ diagonal weighting matrix $\boldsymbol{\Lambda}_n$; its entries consist of power estimates of the inputs of the individual subband channels

$$\boldsymbol{\Lambda}_n = \text{diag}\{\lambda_0(n), \ldots, \lambda_{2M-1}(n)\}$$

with each $\lambda_i(n)$ evaluated via

$$\lambda_i(n) = \beta\lambda_i(n-1) + (1-\beta)|x'_i(n)|^2, \qquad 0 < \beta < 1$$

with initial condition equal to 1.

The reason for the qualification *unconstrained* is that the filters $w_i(z)$ that result from the weight estimates in $\boldsymbol{\mathcal{W}}_n$ do not necessarily satisfy the constraint (9). A *constrained* version of the algorithm is obtained as follows [as suggested by (9)]. We first premultiply $\boldsymbol{\mathcal{W}}_n$ by $\boldsymbol{F}$ followed by $(\boldsymbol{I}_M \oplus \boldsymbol{0})$ in order to zero out its last $M-1$ rows. We then return to the frequency domain by multiplying the result by $\boldsymbol{F}^*$. That is, the constrained estimate, which is denoted by $\boldsymbol{\mathcal{W}}_n^c$, is obtained via $\boldsymbol{\mathcal{W}}_n^c = \boldsymbol{F}^*(\boldsymbol{I}_M \oplus \boldsymbol{0})\boldsymbol{F}\boldsymbol{\mathcal{W}}_n$ so that the recursion for the *constrained* frequency-domain adaptive filter is

$$\boldsymbol{\mathcal{W}}_{n+1}^c = \boldsymbol{\mathcal{W}}_n^c + \mu \boldsymbol{F}^* \boldsymbol{U}_F \boldsymbol{F} \boldsymbol{\Lambda}_n^{-1} \boldsymbol{E}_n \boldsymbol{X}_n \tag{15}$$

with $\boldsymbol{U}_F = (\boldsymbol{I}_M \oplus \boldsymbol{0})$. We remark that in finite-precision implementations, this recursion for the constrained weight vectors can encounter numerical difficulties. This is because round-off errors can lead to weight estimates $\boldsymbol{\mathcal{W}}_n^c$ that violate (9). For this reason, we actually prefer to compute the constrained weight estimates as follows:

1) Run recursion (14) for the unconstrained weight estimates.
2) Then, set $\boldsymbol{\mathcal{W}}_{n+1}^c = \boldsymbol{F}^* \boldsymbol{U}_F \boldsymbol{F} \boldsymbol{\mathcal{W}}_{n+1}$.

IV. Delayless Implementation

Recursion (14) depends on the error matrix $\boldsymbol{E}_n$, and therefore, it requires that we determine the transformed error signals $\{e'_i(n)\}$ defined in (13). These can be evaluated in several dif-

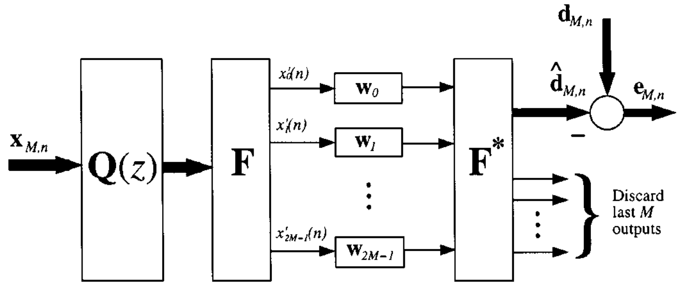


Fig. 3. Equivalent implementation of the block estimation problem of Fig. 2.

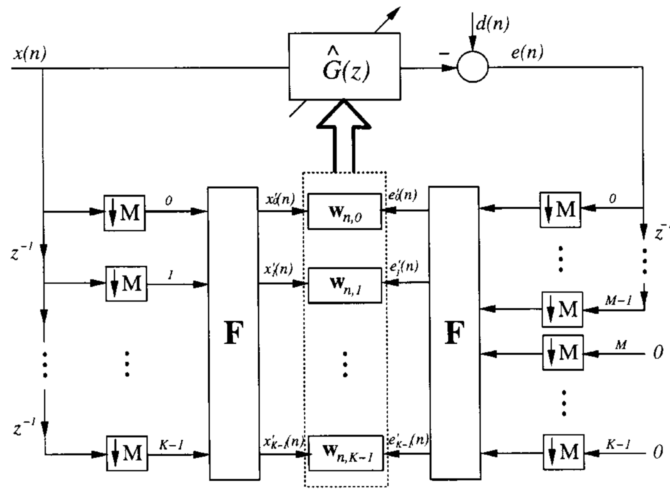


Fig. 4. Overlap-save DFT-MDF structure, where $K = 2M$.

ferent ways. One possibility is to first map the adaptive weights by using (9) or, equivalently, to compute

$$\begin{bmatrix} \hat{\boldsymbol{g}}_{n,0} \\ \hat{\boldsymbol{g}}_{n,1} \\ \vdots \\ \hat{\boldsymbol{g}}_{n,M-1} \end{bmatrix} = [\boldsymbol{I}_M \ \boldsymbol{0}]\boldsymbol{F} \begin{bmatrix} \boldsymbol{w}_{n,0}^T \\ \boldsymbol{w}_{n,1}^T \\ \vdots \\ \boldsymbol{w}_{n,2M-1}^T \end{bmatrix}. \tag{16}$$

Here, the $\hat{\boldsymbol{g}}_{n,i}$ are row vectors that contain the estimated polyphase components of $G(z)$ as in (2). Then, we map the $\hat{\boldsymbol{g}}_{n,i}$ back to subbands by using (9):

$$\begin{bmatrix} \hat{\boldsymbol{w}}_{n,0}^T \\ \hat{\boldsymbol{w}}_{n,1}^T \\ \vdots \\ \hat{\boldsymbol{w}}_{n,2M-1}^T \end{bmatrix} = \boldsymbol{F}^* \begin{bmatrix} \hat{\boldsymbol{g}}_{n,0} \\ \vdots \\ \hat{\boldsymbol{g}}_{n,M-1} \\ \boldsymbol{0} \end{bmatrix}. \tag{17}$$

The resulting $\{\hat{\boldsymbol{w}}_{n,i}^T\}$ are the rows of $\boldsymbol{\mathcal{W}}_{n+1}^c$, and therefore, they satisfy (9). With the $\{\hat{\boldsymbol{w}}_{n,i}\}$ so computed, we can proceed to evaluate the $\{e(n)\}$ as in Fig. 3 and, consequently, the $\{e'_i(n)\}$ as in Fig. 4. This implementation introduces a delay in the evaluation of the sequence $\{e(n)\}$ since its values are computed in block form. Alternatively, $e(n)$ can be computed as indicated in the top part of Fig. 4 by convolving $x(n)$ with $\hat{G}(z)$ in the time domain. Here, $\hat{G}(z)$ denotes the estimate for the wideband filter $G(z)$ that is constructed from the estimated polyphase components $\{\hat{\boldsymbol{g}}_{n,i}\}$. The inconvenience of this procedure is that it re-

TABLE I
SUMMARY OF THE DFT-MDF ALGORITHM

Filtering.

$$\mathbf{x}_n = [\ x(n) \quad \dots \quad x(n-N+1)\ ]$$
$$e(n) = d(n) - \hat{G}_n \mathbf{x}_n$$

Block input and error signals.

$$x(n) \longrightarrow \mathbf{x}_{M,n} = \text{col}\{x(Mn), \dots, x(Mn-M+1)\}$$
$$e(n) \longrightarrow \mathbf{e}_{M,n} = \text{col}\{e(Mn), \dots, e(Mn-M+1)\}$$
$$\mathbf{x}_{2M,n} = \text{col}\{\mathbf{x}_{M,n}, \mathbf{x}_{M,n-1}\}$$
$$\mathbf{x}'_{M,n} = \mathbf{F}\mathbf{x}_{2M,n}$$
$$\mathbf{e}'_{M,n} = \mathbf{F}\begin{bmatrix} \mathbf{I}_M \\ \mathbf{0} \end{bmatrix} \mathbf{e}_{M,n}$$

$$e'_i(n) = i\text{-th entry of } \mathbf{e}'_{M,n}$$

$$x'_i(n) = i\text{-th entry of } \mathbf{x}'_{M,n}$$

$$\mathbf{x}'_{n,i} = [\ x'_i(n) \quad x'_i(n-1) \quad \dots \quad x'_i(n - \tfrac{N}{M} + 1)\ ]$$

Adaptation. For each $i = 0, \dots, 2M-1$:

$$\mathbf{w}_{n+1,i} = \mathbf{w}_{n,i} + \tfrac{\mu}{\lambda_i(n)}[\mathbf{x}'_{n,i}]^* e'_i(n)$$
$$\lambda_i(n) = \beta\lambda_i(n-1) + (1-\beta)|x'_i(n)|^2$$

Subband/wideband mapping:

$$\begin{bmatrix} \hat{\mathbf{g}}_{n,0} \\ \vdots \\ \hat{\mathbf{g}}_{n,M-1} \end{bmatrix} = [\mathbf{I}_M \quad \mathbf{0}]\mathbf{F} \begin{bmatrix} \mathbf{w}^T_{n,0} \\ \vdots \\ \mathbf{w}^T_{n,2M-1} \end{bmatrix}$$

$\hat{G}_n$ is a row vector that denotes the impulse response of the FIR filter with polyphase components $\{\hat{\mathbf{g}}_{n,i}\}$.

quires convolution with a typically long filter $\hat{G}(z)$. Table I summarizes the main steps of this DFT-MDF algorithm.

A more efficient method for evaluating $\{e(n)\}$ is indicated in Fig. 5. A block size R is chosen and a direct convolution is performed only with the first R coefficients of $\hat{G}(z)$. The remaining convolution is performed in block form as follows. First, the polyphase components of size R for the transfer function that corresponds to the remaining coefficients of $\hat{G}(z)$ are obtained from $\hat{G}(z)$; the weight estimates $\boldsymbol{w}'_{n,\,i}$ in the figure are then obtained from these polyphase components according to a transformation similar to (17) with M replaced by R. The block size R can be chosen such that the overall computational complexity of this implementation is minimized. Moreover, the block delay in the signal path is eliminated due to the direct convolution (see [10] and [14] for details).

V. OVERLAP-ADD DFT-MDF STRUCTURE

The MDF in Fig. 4 is commonly referred to as the overlap-save MDF. However, the matrix $\boldsymbol{G}(z)$ can also be decomposed as

$$\boldsymbol{G}(z) = \boldsymbol{Q}_1(z)\boldsymbol{P}_1(z)$$

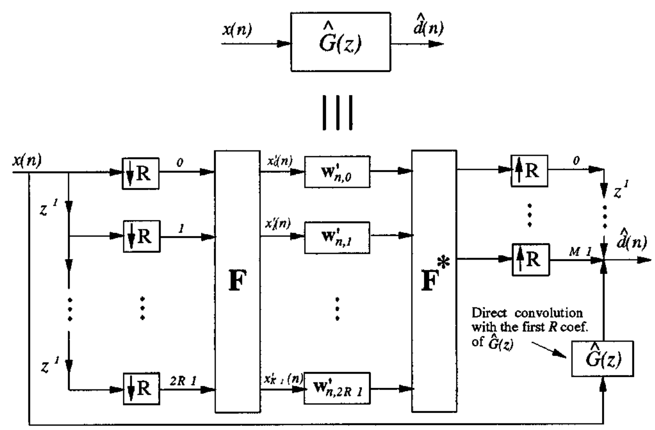


Fig. 5. Delayless block frequency-domain convolution.

where, for $M = 3$

$$\boldsymbol{Q}_1(z) = \begin{bmatrix} z^{-1} & 0 & 0 & 1 & 0 & 0 \\ 0 & z^{-1} & 0 & 0 & 1 & 0 \\ 0 & 0 & z^{-1} & 0 & 0 & 1 \end{bmatrix}$$

and

$$\boldsymbol{P}_1(z) = \begin{bmatrix} 0 & 0 & 0 \\ g_2(z) & 0 & 0 \\ g_1(z) & g_2(z) & 0 \\ g_0(z) & g_1(z) & g_2(z) \\ 0 & g_0(z) & g_1(z) \\ 0 & 0 & g_0(z) \end{bmatrix}. \tag{18}$$

By embedding $\boldsymbol{P}_1(z)$ into the same circulant matrix defined in (7) and proceeding in the same manner as we have done for the overlap-save method, we obtain the following estimation error vector

$$\boldsymbol{e}_M(z) = \boldsymbol{d}_M(z) - \boldsymbol{Q}_1(z)\boldsymbol{F}^*\boldsymbol{W}(z)\underbrace{\boldsymbol{F}\begin{bmatrix} \mathbf{0} \\ \boldsymbol{I}_M \end{bmatrix}\boldsymbol{x}_M(z)}_{\boldsymbol{x}'_M(z)}$$

where we now introduce the $2M \times 1$ signal $\boldsymbol{x}'_{M,\,n}$ with transform

$$\boldsymbol{x}'_M(z) \triangleq \boldsymbol{F}\begin{bmatrix} \mathbf{0} \\ \boldsymbol{I}_M \end{bmatrix}\boldsymbol{x}_M(z).$$

By writing $\boldsymbol{Q}_1(z) = [\mathbf{0}_M\ \boldsymbol{I}] + [\boldsymbol{I}_M\ \mathbf{0}]z^{-1}$, we get

$$\begin{aligned}\boldsymbol{e}_M(z) = \boldsymbol{d}_M(z) - ([\mathbf{0}_M\ \boldsymbol{I}]\boldsymbol{F}^*\boldsymbol{W}(z)\boldsymbol{x}'_M(z) \\ + [\boldsymbol{I}_M\ \mathbf{0}]\boldsymbol{F}^*\boldsymbol{W}(z)\boldsymbol{x}'_M(z)z^{-1}).\end{aligned} \tag{19}$$

In the time domain, this equation becomes

$$\boldsymbol{e}_{M,\,n} = \boldsymbol{d}_{M,\,n} - ([\mathbf{0}_M\ \boldsymbol{I}]\boldsymbol{F}^*\overline{\boldsymbol{X}}_n + [\boldsymbol{I}_M\ \mathbf{0}]\boldsymbol{F}^*\overline{\boldsymbol{X}}_{n-1})\overline{\boldsymbol{W}}_n$$

and the LMS recursion in this case is therefore given by

$$\begin{aligned}\overline{\boldsymbol{W}}_{n+1} &= \overline{\boldsymbol{W}}_n + \mu\left(\overline{\boldsymbol{X}}^*_n\boldsymbol{F}\begin{bmatrix} \mathbf{0} \\ \boldsymbol{I}_M \end{bmatrix} + \overline{\boldsymbol{X}}^*_{n-1}\boldsymbol{F}\begin{bmatrix} \boldsymbol{I} \\ \mathbf{0}_M \end{bmatrix}\right)\boldsymbol{e}_{M,\,n} \\ &= \overline{\boldsymbol{W}}_n + \mu(\overline{\boldsymbol{X}}^*_n + \boldsymbol{J}\overline{\boldsymbol{X}}^*_{n-1})\boldsymbol{e}'_{M,\,n}\end{aligned}$$

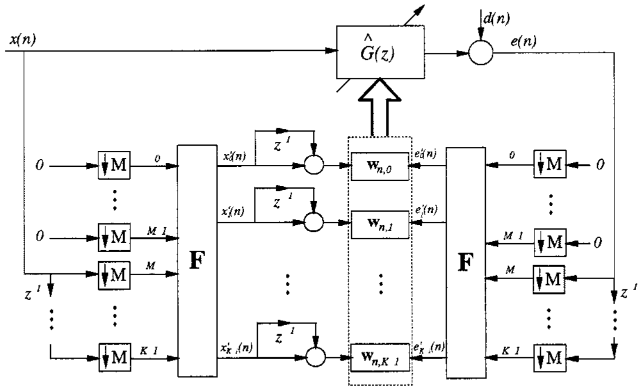


Fig. 6. Overlap-add DFT-MDF structure, where $K = 2M$.

where we defined the transformed block error vector

$$e'_{M,n} \triangleq \boldsymbol{F}\begin{bmatrix}\boldsymbol{0}\\ \boldsymbol{I}_M\end{bmatrix}\boldsymbol{e}_{M,n}$$

and the $2M \times 2M$ diagonal matrix $\boldsymbol{J} = \mathrm{diag}\{1, -1, 1, -1, \cdots, -1\}$. Fig. 6 illustrates the resulting overlap-add MDF structure.

VI. DCT-Based Adaptive Structure

The DFT-based adaptive structure was thus rederived by embedding the matrix $\boldsymbol{P}(z)$ in (5) into the larger circulant matrix $\boldsymbol{C}(z)$ in (7), which was then diagonalized by the DFT matrix. Now, one could embed $\boldsymbol{P}(z)$ into other larger matrices that are not necessarily circulant but that could still be diagonalized by other orthogonal transforms, say, by trigonometric transforms. In this section, we focus on the DCT transform and, in particular, consider the following so-called DCT-III matrix, say, of dimensions $K \times K$,[2]

$$\mathcal{C}_{III} = \sqrt{\frac{2}{K}}\left[\eta_j \cos\frac{j(2i+1)\pi}{2K}\right]_{i,j=0}^{K-1}$$

where $\eta_j = 1/\sqrt{2}$ for $j = 0$ and $j = K$ and $\eta_j = 1$ otherwise. In addition, i indicates the row index and j the column index.

It is known that $\mathcal{C}_{III}$ diagonalizes any $K \times K$ structured matrix functions $\boldsymbol{A}(z)$ that can be expressed as the sum of Toeplitz-plus-Hankel matrix functions in the following form (this fact is developed in [5] in the context of constant matrices with so-called displacement structure [6], [7]):

$$\boldsymbol{A}(z) = \boldsymbol{T}(z) + \boldsymbol{H}(z) + \boldsymbol{B}(z) \tag{20}$$

where

- $\boldsymbol{T}(z)$ *symmetric* Toeplitz matrix;
- $\boldsymbol{H}(z)$ Hankel matrix related to $\boldsymbol{T}(z)$;
- $\boldsymbol{B}(z)$ "border" matrix that is also related to $\boldsymbol{T}(z)$.

[2]The derivation applies equally well to other trigonometric transforms such as DCT-I, DCT-II, DCT-IV, and DST-I to DST-IV. These transforms are also known to diagonalize matrices $\boldsymbol{A}(z)$ of the form (20) for different choices of the Hankel and border matrices $\boldsymbol{H}(z)$ and $\boldsymbol{B}(z)$. We omit the details for brevity.

For example, for $K = 4$, $\{\boldsymbol{A}(z), \boldsymbol{B}(z), \boldsymbol{H}(z)\}$ have the forms

$$\boldsymbol{T}(z) = \begin{bmatrix} t_0(z) & t_1(z) & t_2(z) & t_3(z)\\ t_1(z) & t_0(z) & t_1(z) & t_2(z)\\ t_2(z) & t_1(z) & t_0(z) & t_1(z)\\ t_3(z) & t_2(z) & t_1(z) & t_0(z)\end{bmatrix}$$

$$\boldsymbol{H}(z) = \begin{bmatrix} t_0(z) & t_1(z) & t_2(z) & t_3(z)\\ t_1(z) & t_2(z) & t_3(z) & 0\\ t_2(z) & t_3(z) & 0 & -t_3(z)\\ t_3(z) & 0 & -t_3(z) & -t_2(z)\end{bmatrix}$$

$$\boldsymbol{B}(z) = \begin{bmatrix} -\frac{t_0(z)}{\sqrt{2}-2} & t_1(z) & t_2(z) & t_3(z)\\ t_1(z) & & & \\ t_2(z) & & & \\ t_3(z) & & & \end{bmatrix}(\sqrt{2}-2).$$

Returning to the Toeplitz matrix $\boldsymbol{P}(z)$ in (3), which arises from the representation $\boldsymbol{G}(z) = \boldsymbol{P}(z)\boldsymbol{Q}(z)$, we now embed it into a matrix $\boldsymbol{A}(z)$ that can be diagonalized by $\mathcal{C}_{III}$ [in contrast to the earlier embedding into the circulant matrix $\boldsymbol{C}(z)$]. We do so as follows. Assume, for simplicity, that $M = 2$. Then

$$\boldsymbol{P}(z) = \begin{bmatrix} g_0(z) & g_1(z) & 0\\ 0 & g_0(z) & g_1(z)\end{bmatrix}. \tag{21}$$

We first embed $\boldsymbol{P}(z)$ into a symmetric matrix $\boldsymbol{T}(z)$

$$\boldsymbol{T}(z) \triangleq \begin{bmatrix} 0 & g_0(z) & g_1(z) & 0 & 0\\ g_0(z) & 0 & \boxed{g_0(z)} & \boxed{g_1(z)} & \boxed{0}\\ g_1(z) & g_0(z) & \boxed{0} & \boxed{g_0(z)} & \boxed{g_1(z)}\\ 0 & g_1(z) & g_0(z) & 0 & g_0(z)\\ 0 & 0 & g_1(z) & g_0(z) & 0\end{bmatrix}$$

where the framed entries correspond to $\boldsymbol{P}(z)$. Then, the corresponding matrix $\boldsymbol{A}(z)$ is (we now drop the argument z from the $g_i(z)$ for compactness of notation)

$$\boldsymbol{A}(z) = \begin{bmatrix} 0 & \sqrt{2}g_0 & \sqrt{2}g_1 & 0 & 0\\ \sqrt{2}g_0 & g_1 & \boxed{g_0} & \boxed{g_1} & \boxed{0}\\ \sqrt{2}g_1 & g_0 & \boxed{0} & \boxed{g_0} & \boxed{g_1}\\ 0 & g_1 & g_0 & 0 & g_0\\ 0 & 0 & g_1 & g_0 & 0\end{bmatrix}.$$

We can thus recover $\boldsymbol{P}(z)$ from $\boldsymbol{A}(z)$ as

$$\boldsymbol{P}(z) = [\boldsymbol{0}\ \boldsymbol{I}_M\ \boldsymbol{0}]\boldsymbol{A}(z)\begin{bmatrix}\boldsymbol{0}\\ \boldsymbol{I}_{2M-1}\end{bmatrix} \tag{22}$$

where the column dimension of the square matrix $\boldsymbol{A}(z)$ is $(7M-4)/2$ when M is even and $(7M-3)/2$ when M is odd. We will denote the dimensions of $\boldsymbol{A}(z)$ generically by $K \times K$. That is

$$K = \begin{cases}(7M-4)/2, & M \text{ even}\\ (7M-3)/2, & M \text{ odd.}\end{cases}$$

The matrix $\boldsymbol{A}(z)$ can now be diagonalized by $\mathcal{C}_{III}$ as

$$\boldsymbol{A}(z) = \mathcal{C}_{III}^T\boldsymbol{W}(z)\mathcal{C}_{III} \tag{23}$$

where $\boldsymbol{W}(z) = \text{diag}\{w_i(z)\}$ has K entries. Moreover, as in (9), and for the case $M = 2$, the $\{w_i(z), g_i(z)\}$ are related as follows:

$$\sqrt{10}\begin{bmatrix} 0 \\ g_0(z) \\ g_1(z) \\ 0 \\ 0 \end{bmatrix} = \mathcal{C}_{III}^T \begin{bmatrix} w_0(z) \\ w_1(z) \\ w_2(z) \\ w_3(z) \\ w_4(z) \end{bmatrix}. \tag{24}$$

More generally, the first row of $\boldsymbol{T}(z)$ will have the form

$$[\mathbf{0}_{1\times M-1} \quad g_0 \cdots \quad g_{M-1} \quad \mathbf{0}_{1\times\alpha}]$$

with $M-1$ leading zeros and α trailing zeros so that $\boldsymbol{P}(z)$ is recovered from $\boldsymbol{A}(z)$ as

$$\boldsymbol{P}(z) = [\mathbf{0}_{M\times\beta}\, \boldsymbol{I}_M\, \mathbf{0}]\boldsymbol{A}(z)\begin{bmatrix} \mathbf{0}_{\alpha\times(2M-1)} \\ \boldsymbol{I}_{2M-1} \end{bmatrix}.$$

Here

$$\alpha = \begin{cases} (3M-2)/2, & M \text{ even} \\ (3M-1)/2, & M \text{ odd} \end{cases}$$

and

$$\beta = \begin{cases} M/2, & M \text{ even} \\ (M+1)/2, & M \text{ odd} \end{cases}$$

With this notation, the constraint (24) takes the general form

$$\sqrt{2K}\begin{bmatrix} \mathbf{0}_{(M-1)\times 1} \\ g_0(z) \\ \vdots \\ g_{M-1}(z) \\ \mathbf{0}_{\alpha\times 1} \end{bmatrix} = \mathcal{C}_{III}^T \begin{bmatrix} w_0(z) \\ w_1(z) \\ \vdots \\ w_{K-1}(z) \end{bmatrix}. \tag{25}$$

We now have K FIR filters to adapt, with weight vectors $\{\boldsymbol{w}_i\}$ and regression vectors $\{\boldsymbol{x}'_{n,i}\}$, where

$$\boldsymbol{x}'_M(z) = \mathcal{C}_{III}\begin{bmatrix} \mathbf{0}_{\alpha\times(2M-1)} \\ \boldsymbol{I}_{2M-1} \end{bmatrix}\boldsymbol{Q}(z)\boldsymbol{x}_M(z).$$

If we define, as before

$$\mathcal{W} = \begin{bmatrix} \boldsymbol{w}_0^T \\ \boldsymbol{w}_1^T \\ \vdots \\ \boldsymbol{w}_{K-1}^T \end{bmatrix}$$

$$\boldsymbol{X}_n^T = [[\boldsymbol{x}'_{n,0}]^* \quad [\boldsymbol{x}'_{n,1}]^* \quad \ldots \quad [\boldsymbol{x}'_{n,K-1}]^*]$$

and let $\boldsymbol{E}_n = \text{diag}\{e'_0(n), \ldots, e'_{K-1}(n)\}$, where

$$\boldsymbol{e}'_{M,n} \triangleq \mathcal{C}_{III}\begin{bmatrix} \mathbf{0}_{\beta\times M} \\ \boldsymbol{I}_M \\ \mathbf{0} \end{bmatrix}\boldsymbol{e}_{M,n} \tag{26}$$

we then obtain the following *unconstrained* adaptive version

$$\mathcal{W}_{n+1} = \mathcal{W}_n + \mu\boldsymbol{\Lambda}_n^{-1}\boldsymbol{E}_n\boldsymbol{X}_n. \tag{27}$$

The *constrained* version of the algorithm is obtained as follows [as suggested by the relation (24)]. We first premultiply $\mathcal{W}_n$ by $\mathcal{C}_{III}^T$ followed by $(\mathbf{0} \oplus \boldsymbol{I}_M \oplus \mathbf{0})$ in order to introduce the zero pattern shown in (24). We then return to the frequency domain by multiplying the result by $\mathcal{C}_{III}$. That is, the constrained estimate, which is denoted by $\mathcal{W}_n^c$, is obtained via $\mathcal{W}_n^c = \mathcal{C}_{III}(\mathbf{0}\oplus\boldsymbol{I}_M\oplus\mathbf{0})\mathcal{C}_{III}^T\mathcal{W}_n$ so that the recursion for the *constrained* frequency-domain adaptive filter is

$$\mathcal{W}_{n+1}^c = \mathcal{W}_n^c + \mu\mathcal{C}_{III}\boldsymbol{U}_C\mathcal{C}_{III}^T\boldsymbol{\Lambda}_n^{-1}\boldsymbol{E}_n\boldsymbol{X}_n \tag{28}$$

where

$$\boldsymbol{U}_C = \begin{bmatrix} \mathbf{0}_{(M-1)\times(M-1)} & & \\ & \boldsymbol{I}_M & \\ & & \mathbf{0}_{\alpha\times\alpha} \end{bmatrix}.$$

Again, in order to avoid difficulties with round-off errors, it is preferred to compute the constrained weight estimates as follows:

1) Run recursion (27) for the unconstrained weight estimates.
2) Then, set $\mathcal{W}_{n+1}^c = \mathcal{C}_{III}\boldsymbol{U}_C\mathcal{C}_{III}^T\mathcal{W}_{n+1}$..

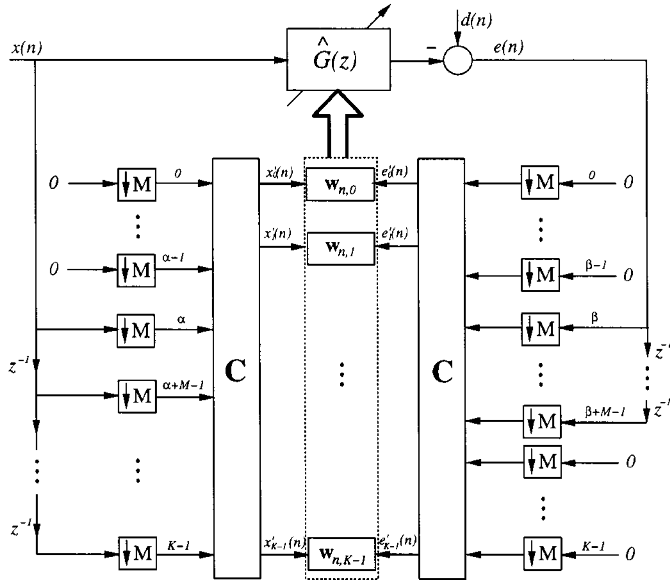


Fig. 7. Delayless DCT-MDF adaptive structure for even M.

Fig. 7 illustrates the DCT-MDF structure, which is analogous to the DFT-MDF, for the overlap-save configuration. The computation of $e(n)$ is similar to the DFT case, as discussed in Section IV. The main steps in the algorithm are listed in Table II.

VII. DHT-Based Adaptive Structure

The DHT matrix of dimensions $K \times K$ is defined as

$$\mathcal{H} = \left[\cos\frac{2ij\pi}{K}\right]_{i,j=0}^{K-1}$$

or

$$\mathcal{H} = \text{Re}(\boldsymbol{F}) + Im(\boldsymbol{F}) \tag{29}$$

where $\boldsymbol{F}$ is the DFT matrix. Note that $\mathcal{H}\mathcal{H}^T = \mathcal{H}^2 = I$. It can be verified (see, e.g., [16]) that $\mathcal{H}$ diagonalizes *symmetric circulant* matrices of the form, say, for $K = 5$ (odd)

$$\boldsymbol{S}(z) = \begin{bmatrix} a_0(z) & a_1(z) & a_2(z) & a_2(z) & a_1(z) \\ a_1(z) & a_0(z) & a_1(z) & a_2(z) & a_2(z) \\ a_2(z) & a_1(z) & a_0(z) & a_1(z) & a_2(z) \\ a_2(z) & a_2(z) & a_1(z) & a_0(z) & a_1(z) \\ a_1(z) & a_2(z) & a_2(z) & a_1(z) & a_0(z) \end{bmatrix} \tag{30}$$

TABLE II
SUMMARY OF THE DCT-III MDF ALGORITHM

Filtering.

$$\mathbf{x}_n = [\; x(n) \quad \dots \quad x(n-N+1) \;]$$
$$e(n) = d(n) - \hat{G}_n \mathbf{x}_n$$

Block input and error signals.

$$x(n) \longrightarrow \mathbf{x}_{M,n} = \mathrm{col}\{x(Mn), \dots, x(Mn-M+1)\}$$
$$e(n) \longrightarrow \mathbf{e}_{M,n} = \mathrm{col}\{e(Mn), \dots, e(Mn-M+1)\}$$
$$\mathbf{x}_{2M,n} = \mathrm{col}\{\mathbf{x}_{M,n}, \mathbf{x}_{M,n-1}\}$$
$$\mathbf{x}'_{M,n} = \mathcal{C}_{III} \begin{bmatrix} \mathbf{0}_{\alpha\times(2M-1)} \\ \mathbf{I}_{2M-1} \end{bmatrix} \mathbf{x}_{2M,n}$$
$$\mathbf{e}'_{M,n} = \mathcal{C}_{III} \begin{bmatrix} \mathbf{0}_{\beta\times M} \\ \mathbf{I}_M \\ \mathbf{0} \end{bmatrix} \mathbf{e}_{M,n}$$

$e'_i(n) = i$-th entry of $\mathbf{e}'_{M,n}$

$x'_i(n) = i$-th entry of $\mathbf{x}'_{M,n}$

$$\mathbf{x}'_{n,i} = [\; x'_i(n) \quad x'_i(n-1) \quad \dots \quad x'_i(n-\tfrac{N}{M}+1) \;]$$

Adaptation. For each $i = 0, \dots, K-1$:

$$\mathbf{w}_{n+1,i} = \mathbf{w}_{n,i} + \tfrac{\mu}{\lambda_i(n)}[\mathbf{x}'_{n,i}]^* e'_i(n)$$
$$\lambda_i(n) = \beta\lambda_i(n-1) + (1-\beta)|x'_i(n)|^2$$

Subband/wideband mapping:

$$\begin{bmatrix} \hat{\mathbf{g}}_{n,0} \\ \vdots \\ \hat{\mathbf{g}}_{n,M-1} \end{bmatrix} = \tfrac{1}{\sqrt{2K}} [\; \mathbf{0}_{M\times(M-1)} \quad \mathbf{I}_M \quad \mathbf{0}_{M\times\alpha} \;] \mathcal{C}_{III}^T \begin{bmatrix} \mathbf{w}_{n,0}^T \\ \vdots \\ \mathbf{w}_{n,K-1}^T \end{bmatrix}$$

$\hat{G}_n$ is a row vector that denotes the impulse response of the FIR filter with polyphase components $\{\hat{\mathbf{g}}_{n,i}\}$.

and for $K = 6$ (even)

$$S(z) = \begin{bmatrix} a_0(z) & a_1(z) & a_2(z) & 0 & a_2(z) & a_1(z) \\ a_1(z) & a_0(z) & a_1(z) & a_2(z) & 0 & a_2(z) \\ a_2(z) & a_1(z) & a_0(z) & a_1(z) & a_2(z) & 0 \\ 0 & a_2(z) & a_1(z) & a_0(z) & a_1(z) & a_2(z) \\ a_2(z) & 0 & a_2(z) & a_1(z) & a_0(z) & a_1(z) \end{bmatrix}. \tag{31}$$

Now, proceeding similarly to the DFT and DCT cases, consider the same matrix $P(z)$ given by (21) for $M = 2$. We embed it into a symmetric circulant matrix $S(z)$ as

$$S(z) \triangleq \begin{bmatrix} 0 & g_0 & g_1 & 0 & g_1 & g_0 \\ g_0 & 0 & g_0 & g_1 & 0 & g_1 \\ g_1 & g_0 & 0 & g_0 & g_1 & 0 \\ 0 & g_1 & g_0 & 0 & g_0 & g_1 \\ g_1 & 0 & g_1 & g_0 & 0 & g_0 \\ g_0 & g_1 & 0 & g_1 & g_0 & 0 \end{bmatrix}.$$

Similarly to (22), we can recover $P(z)$ from $S(z)$ as

$$P(z) = [I_M \; \mathbf{0}] S(z) \begin{bmatrix} \mathbf{0}_{1\times 2M-1} \\ I_{2M-1} \\ \mathbf{0} \end{bmatrix} \tag{32}$$

where the dimension of the $K \times K$ square matrix $S(z)$ is $3M \times 3M$.[3] The matrix $S(z)$ can now be diagonalized by $\mathcal{H}$, say

$$S(z) = \mathcal{H} W(z) \mathcal{H} \tag{33}$$

and, as in (9) and (24), we have for $M = 2$

$$\begin{bmatrix} 0 \\ g_0(z) \\ g_1(z) \\ 0 \\ g_1(z) \\ g_0(z) \end{bmatrix} = \mathcal{H} \begin{bmatrix} w_0(z) \\ w_1(z) \\ w_2(z) \\ w_3(z) \\ w_4(z) \\ w_5(z) \end{bmatrix}. \tag{34}$$

Observe that the $\{g_i(z)\}$ appear now repeated twice after the transformation by $\mathcal{H}$. In order to recover the $\{g_i(z)\}$, we will propose further ahead to average the identical entries in the transformed vector. More generally, we get

$$\begin{bmatrix} 0 \\ g_0(z) \\ \vdots \\ g_{M-1}(z) \\ \mathbf{0}_{M-1\times 1} \\ g_{M-1}(z) \\ \vdots \\ g_0(z) \end{bmatrix} = \mathcal{H} \begin{bmatrix} w_0(z) \\ w_1(z) \\ \vdots \\ w_{K-1}(z) \end{bmatrix}. \tag{35}$$

The transformed input and error vectors in this case are given by

$$x'_M(z) = \mathcal{H} \begin{bmatrix} \mathbf{0}_{1\times 2M-1} \\ I_{2M-1} \\ \mathbf{0} \end{bmatrix} Q(z) x_M(z)$$

and

$$e'_{M,n} \triangleq \mathcal{H} \begin{bmatrix} I_M \\ \mathbf{0} \end{bmatrix} e_{M,n}. \tag{36}$$

The corresponding *unconstrained* adaptive recursion becomes

$$\mathcal{W}_{n+1} = \mathcal{W}_n + \mu \Lambda_n^{-1} E_n X_n \tag{37}$$

whereas the *constrained* version is obtained as follows. We first premultiply $\mathcal{W}_n$ by $\mathcal{H}$ followed by

$$U_H = \tfrac{1}{2} \begin{bmatrix} 0 & & & \\ & I_M & & I_M^{\#} \\ & & \mathbf{0}_{M-1\times M-1} & \\ & I_M^{\#} & & I_M \end{bmatrix}$$

where $I^{\#}$ denotes the reversed identity matrix (it has ones on the antidiagonal). The multiplication by U_H amounts to averaging the repeated entries of the transformed vector. We then return to the frequency domain by multiplying the result by $\mathcal{H}$. That is, the constrained estimate, which is denoted by $\mathcal{W}_n^c$, is obtained via $\mathcal{W}_n^c = \mathcal{H} U_H \mathcal{H} \mathcal{W}_n$ so that the recursion for the *constrained* frequency-domain adaptive filter is

$$\mathcal{W}_{n+1}^c = \mathcal{W}_n^c + \mu \mathcal{H} U_H \mathcal{H} \Lambda_n^{-1} E_n X_n. \tag{38}$$

[3] In general, the top row of $S(z)$ will have the form

$$[0 \; g_0(z) \; \dots \; g_{M-1}(z) \; \mathbf{0}_{1\times M-1} \; g_{M-1}(z) \; \dots \; g_0(z)].$$

It has a single leading zero and $M-1$ zeros in the middle.

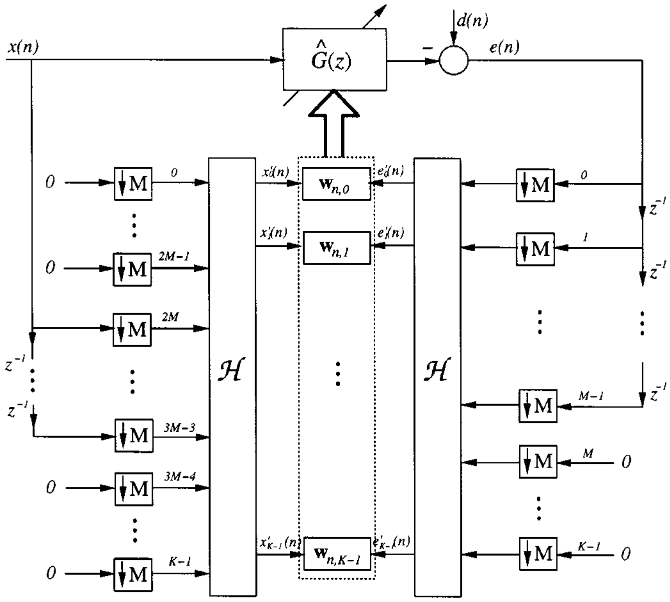


Fig. 8. Delayless DHT-MDF adaptive structure.

Again, in order to avoid difficulties with round-off errors, it is preferred to compute the constrained weight estimates as follows.

1) Run recursion (37) for the unconstrained weight estimates.
2) Then, set $\boldsymbol{\mathcal{W}}^c_{n+1} = \boldsymbol{\mathcal{H}} U_H \boldsymbol{\mathcal{H}} \boldsymbol{\mathcal{W}}_{n+1}$.

Fig. 8 illustrates the DHT-MDF adaptive structure. The algorithm is summarized in Table III.

TABLE III
SUMMARY OF THE DHT-MDF ALGORITHM

Filtering.

$$\mathbf{x}_n = [\ x(n) \quad \ldots \quad x(n-N+1)\ ]$$
$$e(n) = d(n) - \hat{G}_n \mathbf{x}_n$$

Block input and error signals.

$$x(n) \longrightarrow \mathbf{x}_{M,n} = \text{col}\{x(Mn), \ldots, x(Mn-M+1)\}$$
$$e(n) \longrightarrow \mathbf{e}_{M,n} = \text{col}\{e(Mn), \ldots, e(Mn-M+1)\}$$
$$\mathbf{x}_{2M,n} = \text{col}\{\mathbf{x}_{M,n}, \mathbf{x}_{M,n-1}\}$$
$$\mathbf{x}'_{M,n} = \boldsymbol{\mathcal{H}} \begin{bmatrix} \mathbf{0}_{1\times(2M-1)} \\ \mathbf{I}_{2M-1} \\ \mathbf{0} \end{bmatrix} \mathbf{x}_{2M,n}$$
$$\mathbf{e}'_{M,n} = \boldsymbol{\mathcal{H}} \begin{bmatrix} \mathbf{I}_M \\ \mathbf{0} \end{bmatrix} \mathbf{e}_{M,n}$$
$$e'_i(n) = i\text{-th entry of } \mathbf{e}'_{M,n}$$
$$x'_i(n) = i\text{-th entry of } \mathbf{x}'_{M,n}$$
$$\mathbf{x}'_{n,i} = [\ x'_i(n) \quad x'_i(n-1) \quad \ldots \quad x'_i(n - \tfrac{N}{M} + 1)\ ]$$

Adaptation. For each $i = 0, \ldots, K-1$:

$$\mathbf{w}_{n+1,i} = \mathbf{w}_{n,i} + \tfrac{\mu}{\lambda_i(n)} [\mathbf{x}'_{n,i}]^* e'_i(n)$$
$$\lambda_i(n) = \beta \lambda_i(n-1) + (1-\beta)|x'_i(n)|^2$$

Subband/wideband mapping:

$$\begin{bmatrix} \hat{\mathbf{g}}_{n,0} \\ \vdots \\ \hat{\mathbf{g}}_{n,M-1} \end{bmatrix} = \tfrac{1}{2} [\ \mathbf{0}_{M\times 1} \quad \mathbf{I}_M \quad \mathbf{0}_{M\times M-1} \quad \mathbf{I}^{\#}_M\ ] \boldsymbol{\mathcal{H}} \begin{bmatrix} \mathbf{w}^T_{n,0} \\ \vdots \\ \mathbf{w}^T_{n,K-1} \end{bmatrix}$$

$\hat{G}_n$ is a row vector that denotes the impulse response of the FIR filter with polyphase components $\{\hat{\mathbf{g}}_{n,i}\}$.

VIII. COMPUTATIONAL ASPECTS

The proposed algorithms can be implemented with the same efficiency as the DFT-MDF algorithm. In order to get an approximate idea of the computational complexity, we will rely on the algorithm of [8] for computing the DCT and DFT. This algorithm has the advantage of reducing the number of additions by about 25–30% for the DFT's and DCT's on real data. Moreover, the number of multiply counts for a K-length DCT is given by $K/2 \cdot \log_2 K$ and for the DFT by $K/2 \cdot (\log_2(K) - 3) + 2$. Note further that the DHT can be easily evaluated through the DFT [as suggested by (29)]. The overall complexity can be divided into four parts.

1) *Subband decomposition of $x(n)$ and $e(n)$.* This requires a transform of size K for each block of M input samples, which therefore amounts to approximately $(K/M) \log_2 K$ multiplies per sample. Noting that we have $K = 2M$ for the DFT, $K \approx 3.5$ M for the DCT, and $K \approx 3$ M for the DHT, we see that the complexity for this part is similar.
2) *Updating of KN/M-length adaptive filters for each block of M samples.* This requires $K \cdot (N/M)$ computations for each block of M samples or, equivalently, $K \cdot (N/M^2)$ computations per sample. Using $K = 2M$ for the DFT, $K \approx 3.5$ M for the DCT, and $K \approx 3$ M for the DHT, we get approximately 4 N/M[4] real multiplications for the DFT, ≈ 3.5 N/M for the DCT, and 3 N/M for the DHT. One could expect that the complexity for the real arithmetic algorithms would be reduced when compared with the DFT method. However, the size of the transforms involved are greater, which keeps the complexity level approximately the same.
3) *Subband/wideband mapping (constraint).* This requires N/M transforms of size K for each block of M samples. The complexity of this part is similar to the first one, except that here, we need to compute N/M transforms. The computational burden of this part can be further reduced if we apply the constraint less often than every M samples, say, every MI samples, where I is an integer. The same idea was suggested in [10] and [11] without degrading the convergence performance significantly.
4) *Wideband convolution.* As we have mentioned, the convolution part can be realized separately, with no delay and with an optimized block size, in order to reduce complexity. The computational burden for this part is given by $C_{\text{conv}} = R + (N/R+1)[\log_2(2R) - 1] + 4(N/R - 1)$

[4]For the DFT structure, only half of the subband adaptive filters need to be updated.

TABLE IV
COMPUTATIONAL COMPLEXITY FOR THE VARIOUS MDF IMPLEMENTATIONS

MDF	Computational Complexity in mult./sample
DFT	$(2+\frac{N}{M})(\log_2(2M)-3)+\frac{4N+4}{M}+\frac{2N}{M^2}+C_{\text{conv}}$
DCT	$(\frac{NK}{2M^2}+\frac{K}{M})\log_2 K+C_{\text{conv}},\ K\approx 3.5M$
DHT	$(\frac{NK}{2M^2}+\frac{K}{M})(\log_2 K-3)+\frac{N(K+2)}{M^2}+C_{\text{conv}},\ K\approx 3M$

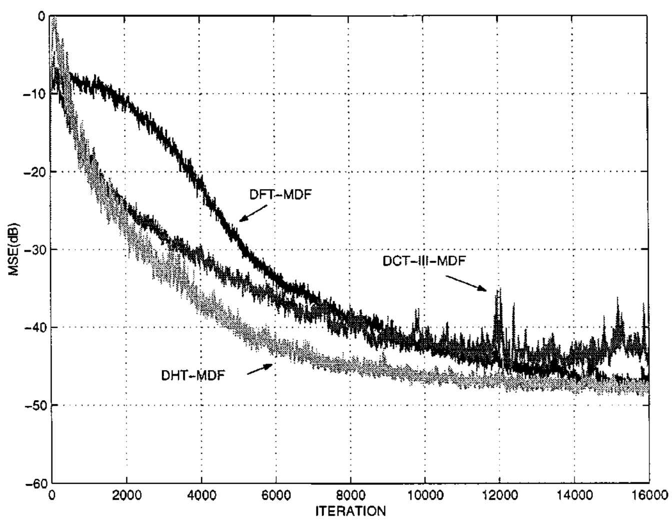


Fig. 9. MSE decay for block sizes $M = 64$, 64, and 32 for the DFT, DCT-III, and DHT-MDF-based structures, respectively.

[10], [11], and it is the same for all the algorithms. This convolution could be implemented using the DCT or the DHT, but as we have seen, the transform sizes would be greater, resulting in a higher computational complexity. Table IV shows the computational complexity of the MDF for each type of transform.

IX. SIMULATIONS

In Fig. 9, we compare the performance of the DFT, DCT-III, and DHT structures for a second-order AR input signal, with z-spectrum given by $S(z) = 1/(1 - 1.9z^{-1} + 0.99z^{-2})$. In each experiment, the block sizes were adjusted for each structure so that the corresponding algorithm exhibited the best performance. We observed that the block size has a different effect in each algorithm. The length of the impulse response of the unknown system was $N = 64$, and the step size used for the adaptive algorithms was chosen as $\mu = M/N$ (which is the inverse of the length of the filter). This choice is within the stability region and guarantees faster convergence. The DFT and DCT-based filters were tested with a block size $M = 64$ (corresponding to subband filters of single tap each). For the DHT-MDF, we used $M = 32$ (corresponding to adaptive filters with two coefficients).

Fig. 10 illustrates the performance of the MDF's where the block sizes were changed to $M = 16$, $M = 32$, and $M = 8$ for the DFT, DCT-III, and DHT schemes, respectively. Note that these block sizes are only for the adaptation process because, as we have mentioned, the convolution can be performed efficiently and without delay with a different optimized block size. For this specific input, we observe faster convergence for the DHT-MDF. In both figures, the curves were generated by averaging over 100 experiments.

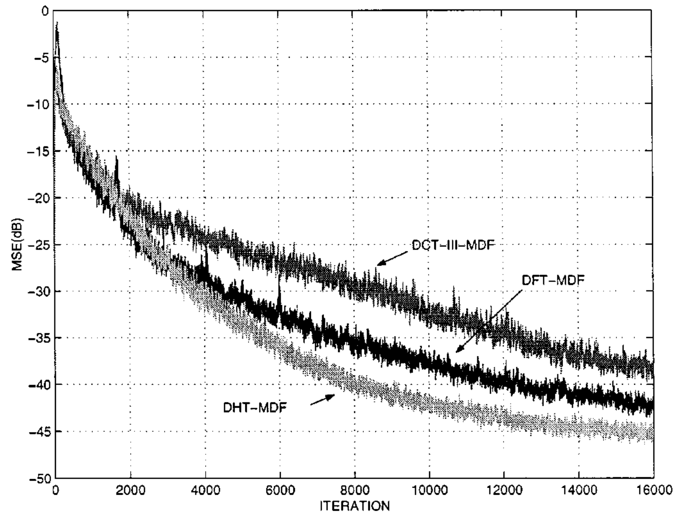


Fig. 10. MSE decay for block sizes $M = 16, 32$, and 8 for the DFT, DCT-III, and DHT-MDF-based structures, respectively.

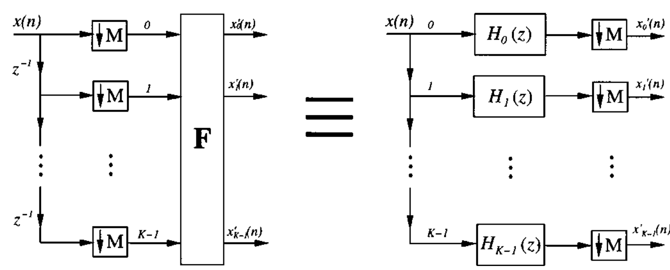


Fig. 11. Equivalent representations of an oversampled DFT filter bank.

The performance of each algorithm depends on the statistics of the input signal applied to the adaptive filters. Different trigonometric transforms perform unequal decorrelations in subbands, which lead to differences in performance. Actually, there is no substantial work on the convergence properties of subband adaptive filters in the literature. We leave a more detailed study of the convergence behavior of the proposed structures for future work. Our simulations are meant to show that there is merit to the proposed schemes; there are clear instances where they perform better.

X. SUBBAND ADAPTIVE FILTERING

We now clarify the relation of the more general subband adaptive filtering structures to the MDF schemes of the earlier sections.

Fig. 11 shows two equivalent representations of the DFT block that comprises the DFT-based MDF structure. The figure on the right is in terms of the DFT modulated bandpass filters, which are known to have poor frequency characteristics (the impulse response of the prototype filter H_0 is a rectangular window).

This fact motivated works on more general adaptive structures, known as subband adaptive filters, which employ different choices for the subband filters $H_k(z)$. These works focus on designing sharp filters $H_k(z)$ that have good attenuation outside passband and are flatter in the passband. However, as in the DFT case, certain constraints [say similar to (9)] need to be developed in order for such designs to correspond to optimal implementations (as in Fig. 3).

We can motivate one such optimal subband adaptive implementation by starting with a different factorization for $\boldsymbol{G}(z)$. Indeed, it can be verified that the pseudocirculant matrix $\boldsymbol{G}(z)$ can also be decomposed as

$$\boldsymbol{G}(z) = \boldsymbol{\Gamma}^*(1/z^*)\boldsymbol{F}^*\boldsymbol{W}(z)\boldsymbol{F}\boldsymbol{\Gamma}(z) \tag{39}$$

where $\boldsymbol{\Gamma}(z)$ is an $M \times M$ diagonal matrix with fractional powers of z

$$\boldsymbol{\Gamma}(z) = \begin{bmatrix} z^{-(M-1)/M} & & & \\ & z^{-(M-2)/M} & & \\ & & \ddots & \\ & & & 1 \end{bmatrix} \tag{40}$$

and $\boldsymbol{W}(z)$ is some diagonal matrix with entries $\{w_i(z)\}$. From (39), it is easy to see that the entries of $\boldsymbol{W}(z)$ and the polyphase components $\{g_i(z)\}$ of the wideband filter $G(z)$ are related as follows:

$$\begin{bmatrix} g_0(z) \\ z^{-1}g_1(z) \\ \vdots \\ z^{-1}g_{M-1}(z) \end{bmatrix} = \begin{bmatrix} 1 & & & \\ & z^{(-(M-1)/M)} & & \\ & & \ddots & \\ & & & z^{-1/M} \end{bmatrix} \cdot \boldsymbol{F} \begin{bmatrix} w_0(z) \\ w_1(z) \\ \vdots \\ w_{M-1}(z) \end{bmatrix}. \tag{41}$$

That is

$$w_i(z) = \frac{1}{\sqrt{M}} \sum_{k=0}^{M-1} z^{-k/M} g_k(z) e^{-(j2\pi ki/M)}.$$

Although $g_k(z)$ are FIR filters of length N/M each, we assign $(N/M)+1$ coefficients to each $w_i(z)$ in order to account for the additional fractional delay terms $\{z^{-k/M}\}$ that appear in the above expression.

We can now proceed similarly to the DFT-based MDF derivation and introduce the estimation error vector

$$\boldsymbol{e}_M(z) = \boldsymbol{d}_M(z) - \boldsymbol{\Gamma}^*(1/z^*)\boldsymbol{F}^*\boldsymbol{W}(z)\underbrace{\boldsymbol{F}\boldsymbol{\Gamma}(z)\boldsymbol{x}_M(z)}_{\boldsymbol{x}'_M(z)}$$

where we defined the transformed input vector

$$\boldsymbol{x}'_M(z) = \boldsymbol{F}\boldsymbol{\Gamma}(z)\boldsymbol{x}_M(z). \tag{42}$$

If we expand $\boldsymbol{\Gamma}(z)$ as

$$\boldsymbol{\Gamma}(z) = \sum_{i=0}^{M-1} \boldsymbol{B}_i z^{-(M-1-i)/M}$$

where the $\boldsymbol{B}_i$ are $M \times M$ diagonal matrices with a single unity entry at the ith diagonal position

$$\boldsymbol{B}_i = (\boldsymbol{0}_i \oplus 1 \oplus \boldsymbol{0}_{M-1-i})$$

then we can express $\boldsymbol{e}_M(z)$ in the form

$$\boldsymbol{e}_M(z) = \boldsymbol{d}_M(z) - \left(\sum_{i=0}^{M-1} \boldsymbol{B}_i z^{(M-1-i)/M}\right)\boldsymbol{F}^*\boldsymbol{W}(z)\boldsymbol{x}'_M(z).$$

In the time domain, this equation becomes

$$\boldsymbol{e}_{M,n} = \boldsymbol{d}_{M,n} - \left(\sum_{i=0}^{M-1} \boldsymbol{B}_i\boldsymbol{F}^*\overline{\boldsymbol{X}}_{n+(M-1-i)/M}\right)\overline{\boldsymbol{W}}.$$

An LMS-based adaptive algorithm that recursively estimates the $\overline{\boldsymbol{W}}$ is then given by

$$\overline{\boldsymbol{W}}_{n+1} = \overline{\boldsymbol{W}}_n + \mu\left(\sum_{i=0}^{M-1} \overline{\boldsymbol{X}}^*_{n+(M-1-i)/M}\boldsymbol{F}\boldsymbol{B}_i\right)\boldsymbol{e}_{M,n}$$

where we continue to write $\boldsymbol{e}_{M,n}$ to denote the estimation error with $\overline{\boldsymbol{W}}$ replaced by $\overline{\boldsymbol{W}}_n$.

Note that the delayed versions of $\overline{\boldsymbol{X}}^*_n$ in the above equation are all in terms of fractions of the unit delay. Now, by invoking stationarity, it becomes justified to make the substitution[5]

$$\overline{\boldsymbol{X}}^*_{n+\delta}\boldsymbol{F}\boldsymbol{B}_i\boldsymbol{e}_{M,n} \approx \overline{\boldsymbol{X}}^*_n\boldsymbol{F}\boldsymbol{B}_i\boldsymbol{e}_{M,n-\delta}$$

for any fractional δ and for all i so that the LMS recursion becomes

$$\begin{aligned}\overline{\boldsymbol{W}}_{n+1} &= \overline{\boldsymbol{W}}_n + \mu\overline{\boldsymbol{X}}^*_n\boldsymbol{F}\sum_{i=0}^{M-1}\boldsymbol{B}_i\boldsymbol{e}_{M,n-(M-1-i)/M} \\ &= \overline{\boldsymbol{W}}_n + \mu\overline{\boldsymbol{X}}^*_n\boldsymbol{e}'_{M,n}\end{aligned} \tag{43}$$

where we defined, in a manner similar to (42)

$$\boldsymbol{e}'_M(z) = \boldsymbol{F}\boldsymbol{\Gamma}(z)\boldsymbol{e}_M(z).$$

In the above expressions, fractional delays appear, and it is known that these can be approximated by a special class of FIR filters (see, e.g., [15]), say

$$z^{-(D_{\text{int}}+(M-i-1)/M)} \approx E_i(z)$$

where D_{int} represents the integer delay associated with the FIR filter $E_i(z)$. Fig. 12 illustrates the subband structure that results from this factorization for $\boldsymbol{G}(z)$. The transformation applied to $\boldsymbol{x}_M(z)$ and $\boldsymbol{e}_M(z)$ can be readily recognized as a maximally decimated[6] DFT filter bank in its polyphase form, where the

[5]Recall that, in general, an LMS update is obtained from a steepest-descent update by replacing the true gradient vector with an instantaneous approximation for it, which simply amounts to dropping the expectation operator. Now, the steepest-descent update that estimates the $\overline{\boldsymbol{W}}$ is given by

$$\overline{W}_{n+1} = \overline{W}_n + \mu E\left[\left(\sum_{i=0}^{M-1}\overline{\boldsymbol{X}}^*_{n+(M-1-i)/M}\boldsymbol{F}\boldsymbol{B}_i\right)\boldsymbol{e}_{M,n}\right]$$

with the expectation operator E. By invoking the stationarity of the data $\{\boldsymbol{d}_{M,n}, \overline{\boldsymbol{X}}_n\}$, we can assume

$$\begin{aligned} E\overline{\boldsymbol{X}}^*_{n+\delta}\boldsymbol{d}_{M,n} &= E\overline{\boldsymbol{X}}^*_n\boldsymbol{d}_{M,n-\delta}, \\ E\overline{\boldsymbol{X}}^*_{n+\delta}\overline{\boldsymbol{X}}_{n+\delta} &= E\overline{\boldsymbol{X}}^*_n\overline{\boldsymbol{X}}_n \end{aligned}$$

for any δ so that we can replace the above update by the alternative update

$$\overline{W}_{n+1} = \overline{W}_n + \mu E\left[\overline{\boldsymbol{X}}^*_n\boldsymbol{F}\sum_{i=0}^{M-1}\boldsymbol{B}_i\boldsymbol{e}_{M,n-(M-1-i)/M}\right].$$

If we now drop the expectation, we arrive again at the desired relation (43).

[6]That is, the number of channels is equal to the decimation factor M.

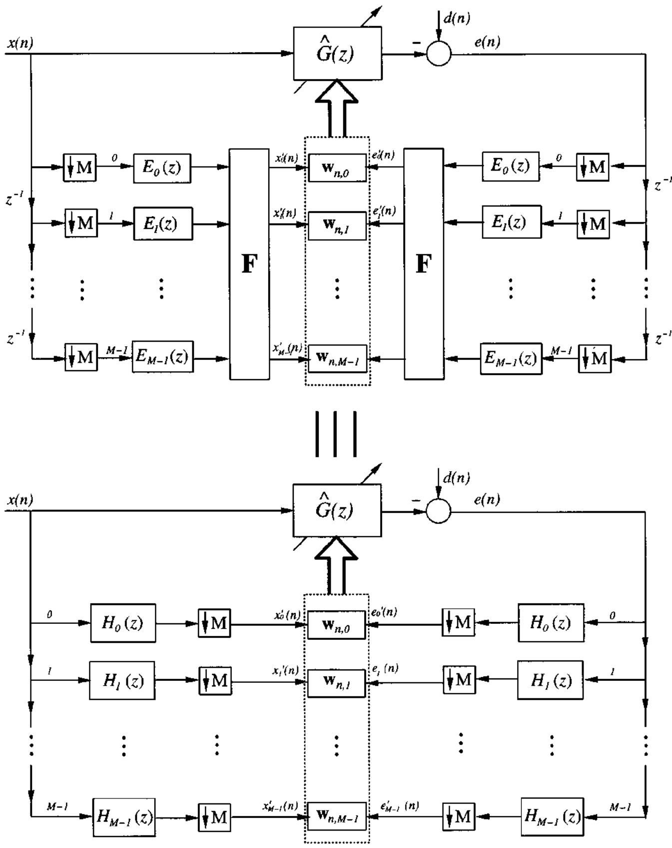


Fig. 12. Subband adaptive filter structure with constraint.

filters $E_i(z)$ represent the polyphase components of a certain prototype filter $H_0(z)$. In addition, note that the last polyphase component is simply a delay, that is

$$E_{M-1}(z) = z^{-D_{\text{int}}}. \tag{44}$$

This means that the prototype filter $H_0(z)$ must be a *Nyquist* (M) filter, which by definition has its $(M-1)$th polyphase component of the form given by (44). Actually, it can be shown that one technique for approximating a fractional delay is to design a symmetric Nyquist (M) filter and pick its ith polyphase component to represent the delay $D_{\text{int}} + (M-i-1)/M$.

Now, similarly to (9) and (24), we can recover the polyphase components of the wideband filter $G(z)$ from the adaptive filters via (41). That is, in a way analogous to the DFT-MDF, we can obtain an estimate of the polyphase components by computing the DFT of the adaptive filters and delaying them by fractional delays. This corresponds to multiplying the DFT of the adaptive filters by the already existing fractional delay filters $\{1, E_0(z), \ldots, E_{M-2}(z)\}$, thus leading to

$$\begin{bmatrix} g_0(z) \\ z^{-1-D_{\text{int}}} g_1(z) \\ \vdots \\ z^{-1-D_{\text{int}}} g_{M-1}(z) \end{bmatrix} = \begin{bmatrix} 1 & & & \\ & E_0(z) & & \\ & & \ddots & \\ & & & E_{M-2}(z) \end{bmatrix} \cdot \boldsymbol{F} \begin{bmatrix} w_0(z) \\ w_1(z) \\ \vdots \\ w_{M-1}(z) \end{bmatrix}$$

where the trailing polyphase components appear delayed by $D_{\text{int}} + 1$. This same construction was used in [11] for an open-loop subband adaptive structure.

Due to the inherent delay D_{int} that is introduced by the filters $E_i(z)$, the LMS update (before the constraint) for the derived structure will ultimately be a delayed LMS version of the form [17]

$$\boldsymbol{\mathcal{W}}_{n+1} = \boldsymbol{\mathcal{W}}_n + \mu \boldsymbol{\Lambda}_n^{-1} \boldsymbol{E}_{n-D_{\text{int}}} \boldsymbol{X}_{n-D_{\text{int}}}. \tag{45}$$

For this reason, smaller step sizes should be used to ensure stability, and the performance of such a scheme can degrade in comparison with the MDF schemes of the earlier sections. Actually, such delayed updates are typical of closed-loop subband implementations.

XI. Conclusions

Using an embedding approach to derive the DFT-based MDF, we have proposed new extensions of the MDF that are based on trigonometric transforms and the discrete Hartley transform. We have also verified that the computational complexity required in the proposed structures is approximately the same as in the DFT-based case, and it involves only real data. Finally, we have shown that the philosophy of the MDF structures can be extended to the more general case of subband adaptive filters.

References

[1] G. A. Clark, S. K. Mitra, and S. R. Parker, "Block implementation of adaptive digital filters," *IEEE Trans. Circuits Syst.*, vol. CAS-28, pp. 584–592, 1981.

[2] K. P. J. Soo, "Multidelay block frequency domain adaptive filters," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 38, pp. 373–376, 1990.

[3] J. Shynk, "Frequency-domain and multirate adaptive filtering," *IEEE Signal Processing Mag.*, pp. 15–37, 1992.

[4] M. R. Petraglia and S. K. Mitra, "Performance analysis of adaptive filter structures based on subband decomposition," in *Proc. ISCAS*, 1993, pp. 60–63.

[5] G. Heinig and K. Rost, "Representations of Toeplitz-plus-Hankel matrices using trigonometric transformations with application to fast matrix-vector multiplication," *Linear Algebra Appl.*, vol. 275–276, pp. 225–248, 1998.

[6] T. Kailath and A. H. Sayed, "Displacement structure: Theory and applications," *SIAM Rev.*, vol. 37, no. 3, pp. 297–386, Sept. 1995.

[7] T. Kailath and A. H. Sayed, Eds., *Fast Reliable Algorithms for Matrices with Structure*. Philadelphia, PA: SIAM, 1999.

[8] M. Vetterli and H. J. Nussbaumer, "Simple FFT and DCT algorithms with reduced number of operations," *Signal Process.*, pp. 267–278, 1984.

[9] F. Beaufays, "Transform-domain adaptive filters: An analytical approach," *IEEE Trans. Signal Processing*, vol. 43, pp. 422–431, Jan. 1995.

[10] D. R. Morgan and M. J. C. Thi, "A delayless subband adaptive filter architecture," *IEEE Trans. Signal Processing*, vol. 43, pp. 1818–1830, Aug. 1995.

[11] R. Merched, P. S. R. Diniz, and M. R. Petraglia, "A new delayless subband adaptive filter structure," *IEEE Trans. Signal Processing*, vol. 47, pp. 1580–1591, Jun. 1999.

[12] P. P. Vaidyanathan, *Multirate Systems and Filter Banks*. Englewood Cliffs, NJ: Prentice-Hall, 1993.

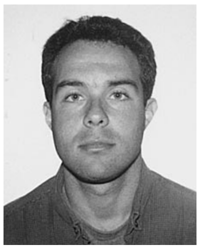
[13] I.-S. Lin and S. K. Mitra, "Overlapped block digital filtering," *IEEE Trans. Circuits Syst. II*, vol. 43, pp. 586–596, Aug. 1996.

[14] M. Xu and Y. Grenier, "Time-frequency domain adaptive filter," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 1989, pp. 1154–1157.

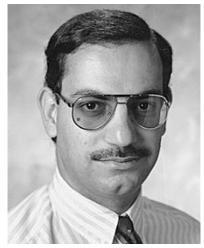
[15] T. I. Laakso, V. Välimäki, M. Karjalainen, and U. K. Laine, "Splitting the unit delay," *IEEE Signal Processing Mag.*, vol. 13, pp. 30–60, Jan. 1996.

[16] D. Bini and P. Favati, "On a matrix algebra related to the discrete Hartley transform," *SIAM J. Matrix Anal. Appl.*, vol. 14, no. 2, pp. 500–507, Apr. 1993.

[17] G. Long, F. Ling, and J. G. Proakis, "The LMS algorithm with delayed coefficient adaptation," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 37, pp. 1397–1405, Sept. 1989.



Ricardo Merched (S'97) was born in Rio de Janeiro, Brazil. He received the B.S. and M.S. degrees in electrical engineering from Universidade Federal do Rio de Janeiro in 1995 and 1997, respectively. Since September 1997, he has been pursuing the Ph.D. degree with the Electrical Engineering Department, University of California, Los Angeles, working on different aspects of adaptive filter design, especially multirate adaptive filters and fast RLS Laguerre adaptive filters.



Ali H. Sayed (SM'99) received the Ph.D. degree in electrical engineering in 1992 from Stanford University, Stanford, CA.

He is Associate Professor of electrical engineering at the University of California, Los Angeles (UCLA). He has over 135 journal and conference publications, is co-author of the research monograph *Indefinite Quadratic Estimation and Control* (Philadelphia, PA: SIAM, 1999) and of the graduate-level textbook *Linear Estimation* (Englewood Cliffs, NJ: Prentice-Hall, 2000). He is also co-editor of the volume *Fast Reliable Algorithms for Matrices with Structure* (Philadelphia, PA: SIAM, 1999). He is a member of the editorial boards of the *SIAM Journal on Matrix Analysis and Its Applications* and the *International Journal of Adaptive Control and Signal Processing*, has served as Co-Editor of special issues of the journal *Linear Algebra and Its Applications*. He has contributed several articles to engineering and mathematical encyclopedias and handbooks. His research interests span several areas including adaptive and statistical signal processing, filtering and estimation theories, equalization techniques for communications, interplays between signal processing and control methodologies, and fast algorithms for large-scale problems.

Dr. Sayed has served on the program committees of several international meetings. He is also a recipient of the 1996 IEEE Donald G. Fink Award. He is Associate Editor of the IEEE TRANSACTIONS ON SIGNAL PROCESSING.