AN EXTENDED MULTIDELAY FILTER: FAST LOW-DELAY ALGORITHMS FOR VERY HIGH-ORDER ADAPTIVE SYSTEMS

Herbert Buchner, Walter Kellermann

Telecommunications Laboratory,
University of Erlangen-Nuremberg
Cauerstr. 7, D-91058 Erlangen, Germany
{buchner,wk}@LNT.de

Jacob Benesty

Bell Laboratories, Lucent Technologies,
700 Mountain Avenue
Murray Hill, NJ 07974, USA
jbenesty@bell-labs.com

ABSTRACT

We propose a novel class of efficient adaptive algorithms in the frequency domain that is tailored to very long adaptive filters and highly autocorrelated input signals as they arise, e.g., in high-quality full-duplex audio applications. The approach exhibits good tracking capabilities of the signal statistics and very low delay. Moreover, it is shown that the low order of computational complexity of the conventional frequency-domain adaptive algorithms can be maintained thanks to efficient realizations. The algorithm allows a tradeoff between the well-known multidelay filter (MDF) and the recursive least-squares (RLS) algorithm. It is also well suited for an efficient generalization to the multichannel case.

1. INTRODUCTION

Many signal processing applications require adaptive filters with very long impulse responses. In acoustic echo cancellation (Fig. 1), for example, thousands of FIR filter coefficients may be required to sufficiently model the echo path (EP). Moreover, the input data are often very highly correlated which causes slow convergence of most algorithms [1]. The requirements are particularly demanding for high-quality audio (with high sampling rates, e.g., $48kHz$ for both, recording and reproduction) and/or multichannel reproduction [2].

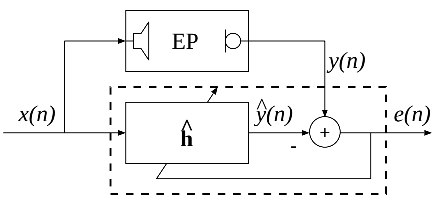

Fig. 1. Adaptive filter (dashed) in the AEC application.

An attractive solution to these problems is to use frequency-domain adaptive filters since, on the one hand, the computational complexity can be greatly reduced by exploiting the fast Fourier transformation (FFT). On the other hand, the discrete Fourier transform approximately decorrelates the input signals, which leads to very favorable convergence properties of the adaptive algorithms. Frequency-domain methods rely on block-processing. In early approaches, the block length was set to the number of filter taps. The associated processing delay, equal to the block length, and the resulting difficulty to follow time-varying statistics of non-stationary signals, are often considered to be a major handicap. Therefore, more flexible structures were introduced: the multidelay filter (MDF) [3], where the filter length L is partitioned into shorter length-N sub-filters, and the generalized multidelay filter (GMDF) [4] which foresees an additional overlap of the input data blocks for data reuse. While the processing delay can be significantly reduced with these structures, the major disadvantage of choosing a block length N that is much shorter than the filter length L is that the convergence speed is often severely degraded for highly correlated signals since the correlations between these shorter blocks are not taken into account. In this paper, we present an extended MDF (EMDF) to solve this problem. The EMDF algorithm follows directly from a generic partitioned frequency-domain adaptive algorithm which can be rigorously derived from an exponentially weighted least-squares criterion in the frequency-domain [5]. The generic frequency-domain framework has led to efficient implementations of multichannel acoustic echo cancellation systems by inherently taking all inter-channel correlations into account. In a similar way, the EMDF algorithm contains all inter-partition correlations. To keep the formal presentation short and accessible, we concentrate on the single-channel EMDF algorithm in this paper; the generalization to the multichannel version is obtained analogously as in [5]. In contrast to inter-channel correlations, the inter-partition correlations in the EMDF result from a shift-structure of the data. This structure is exploited in this paper to derive fast implementations. Using a fast implementation of the EMDF algorithm (FEMDF) the computational complexity can be kept on the same order as that of the classical MDF.

2. GENERIC PARTITIONED FREQUENCY-DOMAIN ADAPTIVE FILTERING

Here, we summarize the generic frequency-domain algorithm in its partitioned and constrained single-channel version. We follow the same notation as in [5], where a detailed derivation and an analysis can be found. Then, in the next section, the connection to the MDF [3] is established and the EMDF algorithm is introduced, based on the generic algorithm. In this paper, we do not introduce an overlap of input data blocks, i.e. the GMDF [4] is not considered in this paper.

From Fig. 1, it can be seen that the error signal at time n between the output of the adaptive filter $\hat{y}(n)$ and the desired output signal $y(n)$ is given by

$$e(n) = y(n) - \hat{y}(n), \tag{1}$$

with

$$\hat{y}(n) = \sum_{\kappa=0}^{L-1} x(n-\kappa)\hat{h}_\kappa, \quad (2)$$

where $\hat{h}_\kappa$ are the coefficients of the filter impulse response. By partitioning the impulse response $\hat{h}$ into segments of length N as in [3], (2) can be written as

$$\hat{y}(n) = \sum_{k=0}^{K-1}\sum_{\kappa=0}^{N-1} x(n-Nk-\kappa)\hat{h}_{Nk+\kappa}, \quad (3)$$

where we assume that the total filter length L is an integer multiple of N, so that $L = KN$. For convenient notation of the algorithm, we rewrite this equation in vectorized form

$$\hat{y}(n) = \sum_{k=0}^{K-1} \mathbf{x}_k^T(n)\hat{\mathbf{h}}_k = \mathbf{x}^T(n)\hat{\mathbf{h}}, \quad (4)$$

where

$$\begin{aligned}\mathbf{x}_k(n) &= [x(n-Nk), x(n-Nk-1), \dots \\ &\quad \dots, x(n-Nk-N+1)]^T, && (5)\\ \hat{\mathbf{h}}_k &= [\hat{h}_{Nk}, \hat{h}_{Nk+1}, \dots, \hat{h}_{Nk+N-1}]^T, && (6)\\ \mathbf{x}(n) &= [\mathbf{x}_0^T(n), \mathbf{x}_1^T(n), \dots, \mathbf{x}_{K-1}^T(n)]^T. && (7)\end{aligned}$$

Superscript ^{T} denotes transposition of a vector or a matrix. The length-N vectors $\hat{\mathbf{h}}_k$, $k = 0, \dots, K-1$ represent *sub-filters* of the partitioned tap-weight vector

$$\hat{\mathbf{h}} = [\hat{\mathbf{h}}_0, \dots, \hat{\mathbf{h}}_{K-1}]^T. \quad (8)$$

We now define the block error signal of length N. Based on (1) and (4) we write

$$\mathbf{e}(m) = \mathbf{y}(m) - \hat{\mathbf{y}}(m), \quad (9)$$

with m being the block time index, and

$$\hat{\mathbf{y}}(m) = \sum_{k=0}^{K-1} \mathbf{U}_k^T(m)\hat{\mathbf{h}}_k = \mathbf{U}^T(m)\hat{\mathbf{h}}, \quad (10)$$

where

$$\begin{aligned}\mathbf{e}(m) &= [e(mN), \dots, e(mN+N-1)]^T, && (11)\\ \mathbf{y}(m) &= [y(mN), \dots, y(mN+N-1)]^T, && (12)\\ \hat{\mathbf{y}}(m) &= [\hat{y}(mN), \dots, \hat{y}(mN+N-1)]^T, && (13)\\ \mathbf{U}_k(m) &= [\mathbf{x}_k(mN), \dots, \mathbf{x}_k(mN+N-1)], && (14)\\ \mathbf{U}(m) &= [\mathbf{U}_0^T(m), \dots, \mathbf{U}_{K-1}^T(m)]^T. && (15)\end{aligned}$$

To derive the frequency-domain algorithm, the block error signal (9), together with (10) is transformed by a DFT matrix to its frequency-domain counterpart. The matrices $\mathbf{U}_k(m)$, $k = 0, \dots, K-1$ are Toeplitz matrices of size $(N \times N)$. Since a Toeplitz matrix $\mathbf{U}_k(m)$ can be transformed, by doubling its size, to a circulant matrix of size $(2N \times 2N)$, and a circulant matrix can be diagonalized using the $(2N \times 2N)$-DFT matrix $\mathbf{F}$ with elements $e^{-j2\pi\nu n/(2N)}$ $(\nu, n = 0, \dots, 2N-1)$, we have

$$\mathbf{U}_k^T(m) = [\mathbf{0}_{N\times N}, \mathbf{I}_{N\times N}]\mathbf{F}^{-1}\mathbf{X}_k(m)\mathbf{F}[\mathbf{I}_{N\times N}, \mathbf{0}_{N\times N}]^T \quad (16)$$

with the diagonal matrices

$$\begin{aligned}\mathbf{X}_k(m) &= \mathrm{diag}\{\mathbf{F}[x(mN-Nk-N), \dots \\ &\quad \dots, x(mN-Nk+N-1)]^T\}. && (17)\end{aligned}$$

By minimizing the resulting frequency-domain error signal by an exponentially weighted least-squares criterion [5], we obtain the corresponding normal equation in the frequency domain. The *exact* recursive solution of this normal equation is given by the generic partitioned block frequency-domain algorithm, which is [5]

$$\begin{aligned}\mathbf{S}_{xx}(m) &= \lambda\mathbf{S}_{xx}(m-1) \\ &\quad +(1-\lambda)\mathbf{X}^H(m)\mathbf{G}_1\mathbf{X}(m), && (18)\\ \mathbf{K}(m) &= \mathbf{S}_{xx}^{-1}(m)\mathbf{X}^H(m), && (19)\\ \underline{\mathbf{e}}(m) &= \underline{\mathbf{y}}(m) - \mathbf{G}_1\mathbf{X}(m)\underline{\hat{\mathbf{h}}}(m-1), && (20)\\ \underline{\hat{\mathbf{h}}}(m) &= \underline{\hat{\mathbf{h}}}(m-1) + (1-\lambda)\mathbf{G}_2\mathbf{K}(m)\underline{\mathbf{e}}(m). && (21)\end{aligned}$$

$\mathbf{S}_{xx}(m)$ denotes the input power spectral density matrix, which is calculated from

$$\mathbf{X}(m) = [\mathbf{X}_0(m), \mathbf{X}_1(m), \dots, \mathbf{X}_{K-1}(m)]. \quad (22)$$

λ $(0 < \lambda < 1)$ is an exponential forgetting factor, and ^{H} denotes conjugate transposition. The underlined quantities $\underline{\mathbf{e}}(m)$, $\underline{\mathbf{y}}(m)$, and $\underline{\hat{\mathbf{h}}}(m)$ denote the frequency-domain counterparts of the respective block representations of the signals in the time domain, i.e.,

$$\begin{aligned}\underline{\mathbf{e}}(m) &= \mathbf{F}\begin{bmatrix}\mathbf{0}_{N\times 1}\\ \mathbf{e}(m)\end{bmatrix}, && (23)\\ \underline{\mathbf{y}}(m) &= \mathbf{F}\begin{bmatrix}\mathbf{0}_{N\times 1}\\ \mathbf{y}(m)\end{bmatrix}, && (24)\\ \underline{\hat{\mathbf{h}}}(m) &= [\underline{\hat{\mathbf{h}}}_0^T(m), \underline{\hat{\mathbf{h}}}_1^T(m), \dots, \underline{\hat{\mathbf{h}}}_{K-1}^T(m)]^T, && (25)\\ \underline{\hat{\mathbf{h}}}_k(m) &= \mathbf{F}\begin{bmatrix}\hat{\mathbf{h}}_k\\ \mathbf{0}_{N\times 1}\end{bmatrix}. && (26)\end{aligned}$$

Two constraint matrices, $\mathbf{G}_1$ of size $(2N \times 2N)$, and $\mathbf{G}_2$ of size $(2L \times 2L)$ appear in the above algorithm (18)-(21). They are defined as

$$\begin{aligned}\mathbf{G}_1 &= \mathbf{F}\mathbf{W}_1\mathbf{F}^{-1}, && (27)\\ \mathbf{W}_1 &= \begin{bmatrix}\mathbf{0}_{N\times N} & \mathbf{0}_{N\times N}\\ \mathbf{0}_{N\times N} & \mathbf{I}_{N\times N}\end{bmatrix}, && (28)\end{aligned}$$

and

$$\begin{aligned}\mathbf{G}_2 &= \mathrm{diag}\{\tilde{\mathbf{G}}_2, \dots, \tilde{\mathbf{G}}_2\}, && (29)\\ \tilde{\mathbf{G}}_2 &= \mathbf{F}\mathbf{W}_2\mathbf{F}^{-1}, && (30)\\ \mathbf{W}_2 &= \begin{bmatrix}\mathbf{I}_{N\times N} & \mathbf{0}_{N\times N}\\ \mathbf{0}_{N\times N} & \mathbf{0}_{N\times N}\end{bmatrix}. && (31)\end{aligned}$$

Due to the formal similarity of Eqs. (18)-(21) to the RLS algorithm [1] in the time domain, we call the matrix $\mathbf{K}(m)$ the frequency-domain Kalman gain. The Kalman gain plays a key role in the following sections.

3. EXTENDED MULTIDELAY FILTER (EMDF)

The Algorithm (18)-(21) is strictly equivalent to the RLS algorithm in the time domain for a block length $N = 1$. Unfortunately, the matrix $\mathbf{S}_{xx}(m)$ in (18) is not sparse (or even diagonal), so the above generic algorithm still has a high computational complexity due to the matrix inversion in (19). However, as shown in e.g., [5], matrix $\mathbf{G}_1$ can very well be approximated by $\mathbf{G}_1 = \mathbf{I}/2$ in (18) if N is sufficiently large. This approximation leads to a block-diagonal structure of matrix $\mathbf{S}_{xx}(m)$ with the diagonal submatrices ($i, j = 0, \ldots, K-1$)

$$\mathbf{S}_{i,j}(m) = \lambda \mathbf{S}_{i,j}(m-1) + (1-\lambda)\mathbf{X}_i^*(m)\mathbf{X}_j(m). \quad (32)$$

Figure 2 illustrates the block structure for the example of 5 partitions. The classical multidelay filter (MDF) is obtained by further approximating $\mathbf{S}_{xx}(m)$ by dropping the off-diagonal components, i.e. the inter-partition correlations (grey diagonals in Fig. 2). This leads to the low computational complexity per output sample, which is linear in K.

The extended multidelay filter (EMDF) proposed here takes the inter-partition correlations into account and thus provides a better approximation to the exact solution of the normal equation. However, a straightforward implementation leads to a computational complexity, which increases quadratically with the number K of partitions. Fast schemes, as discussed in the next section, provide a solution with a complexity that is comparable to that of the classical MDF.

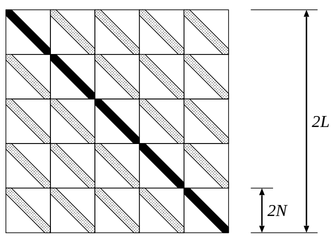

Fig. 2. Structure of matrix $\mathbf{S}_{xx}(m)$.

4. FAST EMDF ALGORITHMS

In order to reduce the computational complexity of the EMDF algorithm, it is interesting that the data among the partitions are not independent. Due to the formal similarity of Eqs. (32), (19)-(21) with the RLS algorithm in the time domain [1, 6], corresponding *fast* implementations can be expected. The necessary and sufficient condition for the existence of fast versions of the RLS algorithms is – apart from the form of the equations for the calculation of the Kalman gain – the shift-structure of the input signal vector [1, 6]. Let $\mathbf{u}(n) = [u(n), u(n-1), \ldots, u(n-L+1)]^T$ be the length-L input signal vector of the RLS algorithm. It is easy to see that the k-th component $u_k(n)$ of this tap input vector can be expressed as

$$u_k(n) = \begin{cases} u(n) & k = 0 \\ u_{k-1}(n-1) & k = 1, \ldots, L-1 \end{cases} \quad (33)$$

In the frequency-domain, we have the following diagonal input matrices to the k-th sub-filter (partition):

$$\begin{aligned} \mathbf{X}_k(m) &= \operatorname{diag}\{\mathbf{F}[x(mN - Nk - N), \ldots \\ &\quad \ldots, x(mN - Nk + N - 1)]^T\}. \quad (34) \end{aligned}$$

Since we assume $\mathbf{G}_1 = \mathbf{I}/2$, the matrices $\mathbf{S}_{xx}(m)$, and thus $\mathbf{K}(m)$ are also block-diagonal. This allows us to perform the calculations separately for each frequency bin ν ($\nu = 0, \ldots, 2N-1$). The ν-th element on the diagonal $\mathbf{X}_k(m)$ is

$$X_k^{(\nu)}(m) = \sum_{n=0}^{2N-1} x(mN - Nk - N + n) e^{-j2\pi\nu n/(2N)}. \quad (35)$$

By substitution of k and m, we find that the following relation holds:

$$X_k^{(\nu)}(m) = \begin{cases} X_0^{(\nu)}(m) & k = 0 \\ X_{k-1}^{(\nu)}(m-1) & k = 1, \ldots, K-1. \end{cases} \quad (36)$$

This relation has the same structure as (33), i.e., in each frequency-bin there is a corresponding shift structure among the partitions. This allows us to apply any fast RLS algorithm to the EMDF case. Note that due to the relatively low number K of partitions (compared to the number L of filter taps, which is relevant in the RLS), the algorithms exhibit a very stable behaviour. The complexity increases only linearly (instead of quadratically as with the ordinary EMDF algorithm) with the number of partitions. Therefore the complexitiy is on the same order as in the classical MDF.

As an example, we consider the so-called fast transversal filter (FTF) structure [6] for efficient calculation of the Kalman gain. Table 1 summarizes the corresponding fast EMDF algorithm. The FTF can be derived by using the *a priori* Kalman gain $\mathbf{K}^{(\nu)}(m) = (\mathbf{S}^{(\nu)})^{-1}(m-1)\mathbf{X}^{(\nu)H}(m)$, where $\mathbf{X}^{(\nu)}(m)$ is a length K row *vector*. This *a priori* Kalman gain can be computed recursively by $5N$ multiplications (for N output values). "Stabilized" versions of FRLS (with L, resp. N more multiplications) exist in the literature but with non-stationary signals like speech, they are not much more stable than their non-stabilized counterparts. A simple remedy is to re-initialize the predictor-based variables when instability is detected with the use of the maximum likelihood variable $\varphi^{(\nu)}$ which is an inherent variable of the fast algorithm [6].

5. EVALUATION FOR ACOUSTIC ECHO CANCELLATION

We demonstrate the performance of the algorithm by an example for acoustic echo cancellation. We apply the (single-channel) EMDF algorithm for (single-channel) AEC with $K = 50$ partitions, a block length (each partition) $N = 64$, and a high sampling rate of $48kHz$. As input signal, we chose classical music (*Air* by Bach). The signal sequence is highly auto-correlated (tonal sounds, which are known as worst case for the adaptation). An echo-to-background noise ratio (EBR) of $45dB$ on the microphone was chosen. The dashed lines in Fig. 3 show the *echo return loss enhancement* $ERLE$ and the misalignment achieved by the conventional MDF. For the solid lines, the same data and the same parameters are used with the EMDF algorithm. It is important to note that the regularization is adjusted in each case. Several simulations have confirmed that the EMDF shows a significantly more stable behaviour than the classical MDF due to the more accurate approximation to the exact recursive solution of the normal equation.

6. CONCLUSIONS

We presented a new class of algorithms within the general framework of frequency-domain adaptive filtering. This class exhibits

Table 1 An FTF-based Fast EMDF algorithm

Definitions:

$$
\begin{aligned}
\mathbf{W}_1 &= \begin{bmatrix} \mathbf{0}_{N\times N} & \mathbf{0}_{N\times N} \\ \mathbf{0}_{N\times N} & \mathbf{I}_{N\times N} \end{bmatrix} \\
\mathbf{G}_1 &= \mathbf{F}\mathbf{W}_1\mathbf{F}^{-1} \\
\mathbf{W}_2 &= \begin{bmatrix} \mathbf{I}_{N\times N} & \mathbf{0}_{N\times N} \\ \mathbf{0}_{N\times N} & \mathbf{0}_{N\times N} \end{bmatrix} \\
\tilde{\mathbf{G}}_2 &= \mathbf{F}\mathbf{W}_2\mathbf{F}^{-1} \\
\mathbf{G}_2 &= \mathrm{diag}\{\tilde{\mathbf{G}}_2, \ldots, \tilde{\mathbf{G}}_2\}
\end{aligned}
$$

Input:

$$
\begin{aligned}
\mathbf{X}_k(m) &= \mathrm{diag}\{\mathbf{F}[x(mN - Ni - N), \ldots \\
&\quad \ldots, x(mN - Ni + N - 1)]^T\}, \\
&\quad k = 0, \ldots, K-1
\end{aligned}
$$

Prediction:

$$
\begin{aligned}
\mathbf{X}^{(\nu)}(m) &\leftarrow \mathbf{X}_k(m), \nu = 0, \ldots, 2N-1, \\
&\quad k = 0, \ldots, K-1 \\
\underline{e}_a^{(\nu)}(m) &= X_0^{(\nu)*}(m) \\
&\quad -\underline{\mathbf{a}}^{(\nu)H}(m-1)\mathbf{X}^{(\nu)H}(m-1) \\
\varphi_1^{(\nu)}(m) &= \varphi^{(\nu)}(m-1) + \frac{|\underline{e}_a^{(\nu)}(m)|^2}{E_a^{(\nu)}(m-1)} \\
\begin{bmatrix} \mathbf{t}^{(\nu)}(m) \\ M^{(\nu)}(m) \end{bmatrix} &= \begin{bmatrix} 0 \\ \mathbf{K}_1^{(\nu)}(m-1) \end{bmatrix} \\
&\quad + \begin{bmatrix} 1 \\ -\underline{\mathbf{a}}^{(\nu)}(m-1) \end{bmatrix} \frac{\underline{e}_a^{(\nu)}(m)}{E_a^{(\nu)}(m-1)} \\
E_a^{(\nu)}(m) &= \lambda\left(E_a^{(\nu)}(m-1) + \frac{|\underline{e}_a^{(\nu)}(m)|^2}{\varphi^{(\nu)}(m-1)}\right) \\
\underline{\mathbf{a}}^{(\nu)}(m) &= \underline{\mathbf{a}}^{(\nu)}(m-1) \\
&\quad + \mathbf{K}_1^{(\nu)}(m-1)\frac{\underline{e}_a^{(\nu)*}(m)}{\varphi^{(\nu)}(m-1)} \\
\underline{e}_b^{(\nu)}(m) &= E_b^{(\nu)}(m-1)M^{(\nu)}(m) \\
\mathbf{K}_1^{(\nu)}(m) &= \mathbf{t}^{(\nu)}(m) + \underline{\mathbf{b}}^{(\nu)}(m-1)M^{(\nu)}(m) \\
\varphi^{(\nu)}(m) &= \varphi_1^{(\nu)}(m) - \underline{e}_b^{(\nu)*}(m)M^{(\nu)}(m) \\
E_b^{(\nu)}(m) &= \lambda\left(E_b^{(\nu)}(m-1) + \frac{|\underline{e}_b^{(\nu)}(m)|^2}{\varphi^{(\nu)}(m)}\right) \\
\underline{\mathbf{b}}^{(\nu)}(m) &= \underline{\mathbf{b}}^{(\nu)}(m-1) + \mathbf{K}_1^{(\nu)}(m)\frac{\underline{e}_b^{(\nu)*}(m)}{\varphi^{(\nu)}(m)} \\
\mathbf{K}^{(\nu)}(m) &= \frac{\mathbf{K}_1^{(\nu)}(m)}{\varphi^{(\nu)}(m)} \\
\mathbf{K}(m) &\leftarrow \mathbf{K}^{(\nu)}(m), \nu = 0, \ldots, 2N-1
\end{aligned}
$$

Filtering:

$$
\begin{aligned}
\underline{\mathbf{e}}(m) &= \underline{\mathbf{y}}(m) - \mathbf{G}_1\mathbf{X}(m)\hat{\underline{\mathbf{h}}}(m-1) \\
\hat{\underline{\mathbf{h}}}(m) &= \hat{\underline{\mathbf{h}}}(m-1) + \mu\mathbf{G}_2\mathbf{K}(m)\underline{\mathbf{e}}(m)
\end{aligned}
$$

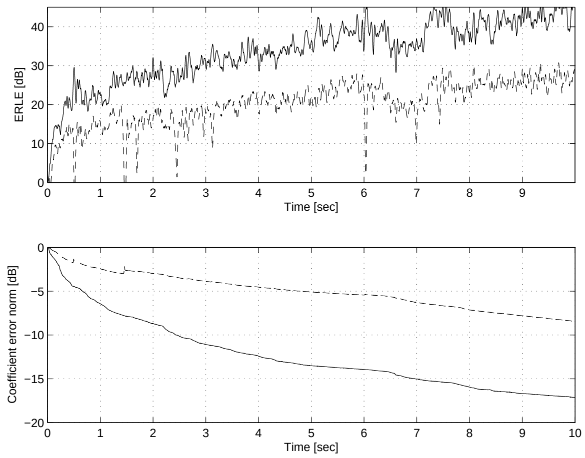

Fig. 3. Comparison between classical MDF (dashed) and EMDF (solid lines).

several very desirable properties, particularly for very long adaptive filters and high sampling rates. Due to the rigorous derivation of the new algorithm with a block size $N \leq L$, we found a natural way of efficiently taking all cross-correlations between the partitions into account. As shown by way of simulations, the algorithm can lead to a significant improvement of the convergence speed over the multidelay filter, i.e., the conventional frequency-domain algorithm with partitioned blocks. Moreover, by introducing fast calculation schemes for the frequency-domain Kalman gain in the extended multidelay filter, the computational complexity can be kept on the same order as that of the conventional multidelay filter.

7. REFERENCES

[1] S. Haykin, *Adaptive Filter Theory*, 3rd ed., Prentice Hall Inc., Englewood Cliffs, NJ, 1996.

[2] H. Buchner, S. Spors, W. Kellermann, and R. Rabenstein, “Full-Duplex Communication Systems with Loudspeaker Arrays and Microphone Arrays,” *Proc. IEEE Int. Conference on Multimedia and Expo (ICME)*, Lausanne, Switzerland, Aug. 2002.

[3] J.-S. Soo and K.K. Pang, “Multidelay block frequency domain adaptive filter,” *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-38, pp. 373-376, Feb. 1990.

[4] E. Moulines, O. Ait Amrane, and Y. Grenier, “The generalized multidelay adaptive filter: structure and convergence analysis,” *IEEE Trans. Signal Processing*, vol. 43, pp. 14-28, Jan. 1995.

[5] H. Buchner, J. Benesty, and W. Kellermann, “Multichannel Frequency-Domain Adaptive Algorithms with Application to Acoustic Echo Cancellation,” in J.Benesty and Y.Huang (eds.), *Adaptive signal processing: Application to real-world problems*, Springer-Verlag, Berlin/Heidelberg, Jan. 2003.

[6] M.G. Bellanger, *Adaptive Digital Filters and Signal Analysis*, Marcel Dekker, 1987.