Adaptive Filtering in Subbands Using a Weighted Criterion

Marc de Courville and Pierre Duhamel, *Senior Member, IEEE*

***Abstract*— Transform-domain adaptive algorithms have been proposed to reduce the eigenvalue spread of the matrix governing their convergence, thus improving the convergence rate. However, a classical problem arises from the conflicting requirements between algorithm improvement requiring rather long transforms and the need to keep the input/output delay as small as possible, thus imposing short transforms. This dilemma has been alleviated by the so-called "short-block transform domain algorithms" but is still apparent.**

This paper proposes an adaptive algorithm compatible with the use of rectangular orthogonal transforms (e.g., critically subsampled, lossless, perfect reconstruction filter banks), thus allowing better tradeoffs between algorithm improvement, arithmetic complexity, and input/output delay.

The method proposed here makes a direct connection between the minimization of a specific weighted least squares criterion and the convergence rate of the corresponding stochastic gradient algorithm. This method leads to improvements in the convergence rate compared with both LMS and classical frequency domain algorithms.

I. Introduction and Related Work

ADAPTIVE filtering is a widespread technique in many applications. For acoustic echo cancellation (AEC) hands-free telephony, very large adaptive filters are used in a system identification context, whereas in digital communications, adaptive filters perform the channel distortion equalization. The present need for increased throughput in new systems also results in an increase of the equalizer length. In these two areas, there is a demand for efficient and low complexity algorithms. This paper builds on this approach.

• **Classical Approaches Using a Square Orthogonal Transform**

The least mean square (LMS) adaptive algorithm [1] is widely used since it provides both low complexity and robust performance. However, the relatively slow convergence is a major drawback in several applications. This is the motivation for searching for improved, yet simple, versions of the initial algorithm. Interestingly, many solutions make use of projections of the input signal on an orthogonal basis, allowing them to act "almost" separately on the various modes of the convergence.

One such approach builds on a frequency domain implementation of the block LMS (BLMS) algorithm [2], [3], which is a fast implementation using the discrete Fourier transform (DFT), known as the fast BLMS algorithm. This serves as a basis for an improved version denoted the frequency domain block LMS (FBLMS) algorithm, in which the DFT plays two roles. First, the DFT is used for complexity reduction, and second, the availability of the energy of the input signal frequency bins allows an individual normalization of the adaptation steps to be performed for convergence rate improvements.

The second approach (the transform domain adaptive filter (TDAF) [4], [5]) was originally proposed for convergence and residual error improvements (the fast convolution property of the orthogonal transform was not required). Compared with the FBLMS algorithm, more flexibility in the choice of this transform is provided.

In both cases, the orthogonal transform is used as a means for decomposing *the input signal* into approximately decorrelated components [5]–[7]. Moreover, in both approaches, the variables that are explicitly adapted are the transform domain coefficients of the adaptive filter. Therefore, in the initial versions of both algorithms, the transform size is strongly linked to the adaptive filter length. This constraint has been somewhat relaxed in further extensions such as the multidelay filter (MDF) [8] or the generalized MDF (GMDF) [9] but cannot be totally removed.

A common characteristic of these schemes is that the transform is primarily applied to the adaptive filter inputs as a means for factorizing the input signal autocorrelation matrix.

• **Filter Banks and Adaptive Filtering: The "Transform Domain" Approach**

For the TDAF as well as for the FBLMS algorithm, a better decorrelation of the input signal is feasible by increasing the square transform size (i.e., the number of basis functions). However, when the number of basis functions increases, so does their length. Thus, a specific procedure is required for constraining accordingly the time domain adaptive filter to remain of constant size. Nevertheless, the use of longer basis functions without increasing their number would also allow a better decorrelation to be performed in a simpler and more direct way, with maybe the additional advantages of a reduced I/O delay. Such longer, orthogonal, basis functions are provided by lossless (LL) perfect reconstruction (PR) filter banks (FB's) [10]; the coefficients of an LL PR FB can be interpreted as a nonsquare matrix of orthonormal vectors.

Indeed, it is well known that LL PR FB can achieve efficient decorrelation, even for a small number of components

Manuscript received March 7, 1997; revised December 19, 1997. This work was supported by the CCETT, Rennes, France. The associate editor coordinating the review of this paper and approving it for publication was Prof. M. H. Er.

M. de Courville was with Département Signal, ENST, Paris, France. He is now with Motorola CRM, Paris, France.

P. Duhamel is with Département Signal, ENST, Paris, France.

Publisher Item Identifier S 1053-587X(98)05243-X.

[11]. Since the discrete wavelet transform (DWT) can be implemented using LL PR FB, the DWT also belongs to this category and can be considered here as a special case.

A straightforward use of such basis functions applied to the input signal as done in a transform domain adaptive scheme would raise the same issues as the one met in [12]–[14]; some kind of periodization of the wavelet transform structure or of the input signal has to be performed in order to keep the orthogonality property on a finite length window [15], [16].

- **Proposed Algorithm Using a Subband Decomposition**

Instead of working directly on the input signal, in this section, a projection of *the modeling error* on an orthogonal basis [17] is considered. Indeed, intuitively, separating the error signal driving the convergence into decorrelated components should bring an improvement of convergence rate, as is the case when the input signal is decomposed.

Focusing on the previous considerations, this paper proposes a new subband adaptive algorithm: the weighted subband adaptive filter (WSAF). The main advantages of our algorithm are twofold. First, the required orthogonality is kept without changing the classical filter bank computational structure, and second, it provides full flexibility in the choice of the length of the filter bank. In addition, this approach shows some links between *appropriately weighted* least squares minimization and the improvement in convergence of the resulting adaptive filter.

- **Other Relevant Approaches Using a Subband Decomposition**

Another set of papers [18]–[21] deals with subband adaptive filtering in the context of AEC, mainly in order to allow computational savings. In these schemes, the input and reference signals are decomposed into subbands and decimated, and the adaptation is performed in each subband. Thus, transposing the adaptive filtering structure behind the analysis filter bank turns the problem of adapting a single long FIR filter into that of adapting several short filters operating at a lower rate [18].

Since the frequency response of the filters in the bank are overlapping (e.g. QMF banks), when critical subsampling is used [22], [23], the output of the FB contains undesirable aliasing components that impair the adaptation ability of the algorithm. These aliasing components can be canceled by introducing a full matrix of cross filters between the analysis and synthesis bank [11], [24]. When doing so, a time domain filtering equation is exactly implemented in subbands. However, the equivalence between filtering in the time domain and in the subband domain imposes many connections between these cross filters. Therefore, not all the filters of this polyphase matrix should be adapted independently if maximum convergence rate is desired. This is difficult to take into account while using a stochastic gradient algorithm.

An approximate approach has been proposed in [18] and [25], where only adjacent filters in the bank were supposed to overlap in order to reduce the complexity. This corresponds to the polyphase matrix of subband adaptive filters being tridiagonal.

Other authors [19] avoid this problem by using fewer overlapping filter banks (in the frequency domain). However, this tends to introduce spectral holes in the output signal of the adaptive filter (which are located at the edges of the subbands). Another approach, which was proposed in [26], consists of modifying the structure of the filter bank by introducing non-decimated auxiliary subbands. Finally, the use of oversampled filter banks was recommended in [20] and [21] in order to decrease the aliasing effect. In any of the above methods, it appears that a noticeable gain in computational complexity always comes with a degradation of the convergence properties of the subband adaptive process. Thus, the goal seems to be to try to offset this loss while increasing the computational complexity by the smallest possible amount.

One explanation of the problems encountered with these approaches comes from extending the use of subband decomposition to purposes other than convergence improvement: *fast* convolution implementation (which can only be achieved approximately in this way). In order to avoid this problem, our method separates the convergence improvement (which is obtained by a subband decomposition of the error) from the reduction in complexity achieved by any fast algorithm since our method is basically of a block type. This is explained in Section III, which presents the new multiband adaptive algorithm.

- **Scope of the Paper**

Section II describes the notation and introduces the new weighted criterion to be minimized. The proposed algorithm (the WSAF) is presented in Section III as well as an appropriate choice of the weights in the stationary case. The WSAF convergence behavior is studied in Section IV from both the convergence rate and residual error point of view. Further refinements that will improve the convergence of the WSAF are provided in Section V, including a time-varying strategy for the weights and the adaptation step size for a nonstationary input signal.

The use of a decimated FB forces our algorithm to be of a block type. Just like in all "improved" block algorithms, this has two opposing effects. On the one hand, increasing the block size improves the algorithm behavior, but on the other, this simultaneously decreases the stability region of the stepsizes. Hence, in order to provide fair comparisons, Section VI compares the TDAF and FBLMS algorithms with the WSAF using the same block sizes. This requires the description of small block realizations of the FBLMS and BTDAF, which is detailed in Section VI-A. Computational complexities are discussed in Section VI-B. Section VI-C finally compares convergence behaviors by simulation.

II. Notation Criterion

The whole study is undertaken in the context of adaptive identification. All variables are assumed to be of complex value, which corresponds to the most general case, and allows for easy use in a channel equalization context.

In the following, let $(\cdot)^t$ be the operator denoting transposition, let $(\cdot)^*$ be conjugation, and let $(\cdot)^H = ((\cdot)^t)^*$, whereas $\mathrm{Diag}(X)$ stands for the diagonal matrix whose diagonal elements are the components of vector X. I_L is the identity matrix of dimension L. $\odot$ indicates the component by component (Schur) product of two vectors. The FIR adaptive filter

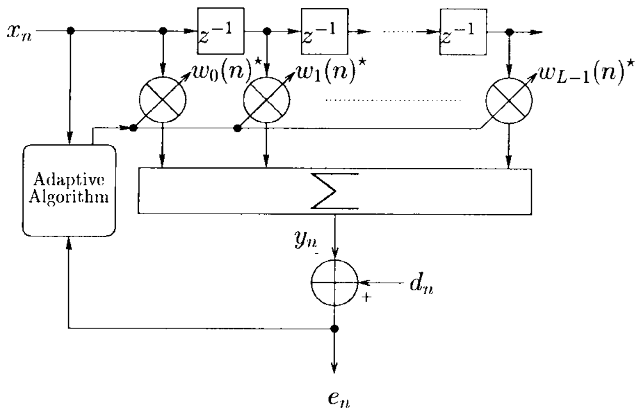

Fig. 1. General time domain adaptive digital filter.

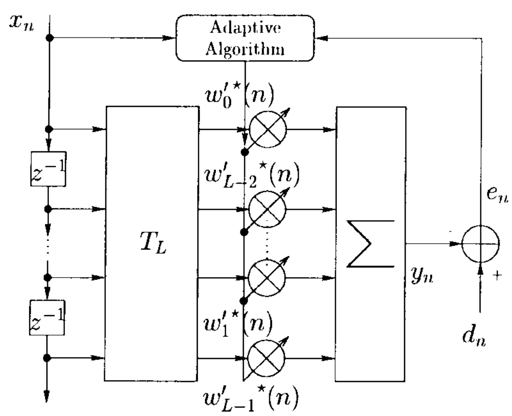

Fig. 2. General transform domain LMS adaptive digital filter.

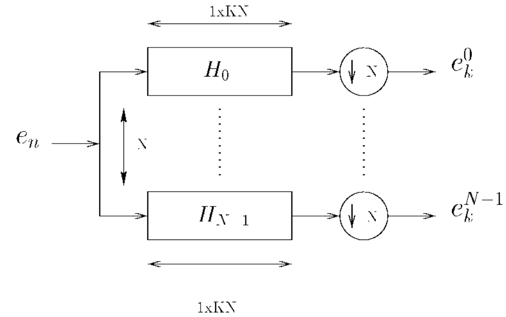

Fig. 3. Notation for the analysis filter bank.

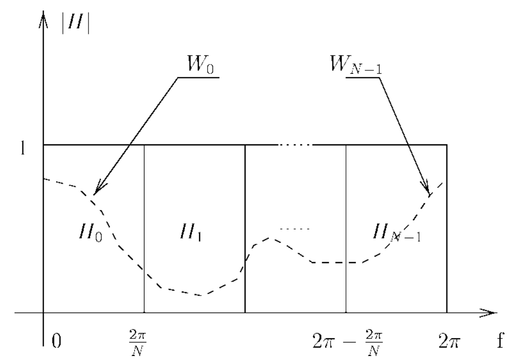

Fig. 4. Frequency division of the update of the adaptive filter.

has length L. In this paper, data are filtered by the conjugate of filter W for a more convenient expression of the convergence equations. Notation x_n and d_n denote the complex valued input and reference signal, respectively, as depicted in Fig. 1. N is the block size (number of computations that are grouped together, which is equal to the number of subbands), whereas the orthogonal filter bank is assumed to be of length KN for simplicity. Uppercase letters stand for vectors or matrices of appropriate sizes as in

$$\begin{aligned}
X_n &= (x_n, x_{n-1}, \cdots, x_{n-KN+1})^t \\
\mathcal{X}_n &=, (X_n, X_{n-1}, \cdots, X_{n-L+1})_{KN\times L} \\
W_n &= (w_0(n), w_1(n), \cdots, w_{L-1}(n))^t \\
D_n &= (d_n, d_{n-1}, \cdots, d_{n-KN+1})^t \\
E_n &= (e_n, e_{n-1}, \cdots, e_{n-KN+1})^t = D_n - \mathcal{X}_n W_n^*.
\end{aligned}$$

The row vector of the ith analysis filter coefficients is denoted by H_i, where all of them are gathered in an $N \times KN$ matrix $H = (H_0^t, \cdots, H_{N-1}^t)^t_{N\times KN}$. Note that when $K = 1$, H reduces to an orthonormal transform, whereas larger values of K are able to increase the decorrelation properties of the transform without increasing the number of subbands. The efficiency of this procedure will be apparent in Table I in Section IV-A.

Suppose now that the error $e_{kN+n}, 0 \le n \le N-1, k \ge 0$ has been passed through a LL PR FB (cf., Fig. 3), thus being decomposed in N subbands $(e_k^0, \cdots, e_k^{N-1})^t, k \ge 0$ by H, which is the analysis bank.

If the input signal is ergodic and wide sense stationary, the orthogonality property of the LL PR FB ensures that both formulations of the block criterion below are equivalent (where $\mathcal{E}(\cdot)$ denotes mathematical expectation)

$$J^{\text{Block}} = \mathcal{E}\left(\sum_{n=0}^{N-1} |e_{kN+n}|^2\right) = \mathcal{E}\left(\sum_{i=0}^{N-1} |e_k^i|^2\right). \quad (1)$$

Of course, minimizing both versions of the criterion would result in the same algorithm, i.e., a BLMS algorithm. However, consider now the minimization of the following weighted mean square criterion, where the quadratic errors in each subband are weighted by some constants

$$J^{\text{WSAF}} = \sum_{i=0}^{N-1} \lambda_i \mathcal{E}(|e_k^i|^2). \quad (2)$$

Here, the size of the transform (the number of subbands) is independent of the filter length and depends only on the block size. Further, since this approach relies on orthogonality, and since orthogonality of the LL PR FB requires the presence of a subsampling by N, this method is restricted to block algorithms. Although it is described in the context of LL PR FB, this method still holds for any orthogonal transform (as a special case of LL PR FB in which $K = 1$).

III. The Proposed Algorithm

Assume that the set of λ_i is constant. An LMS-like adaptative minimization of the criterion (2) is easily obtained by

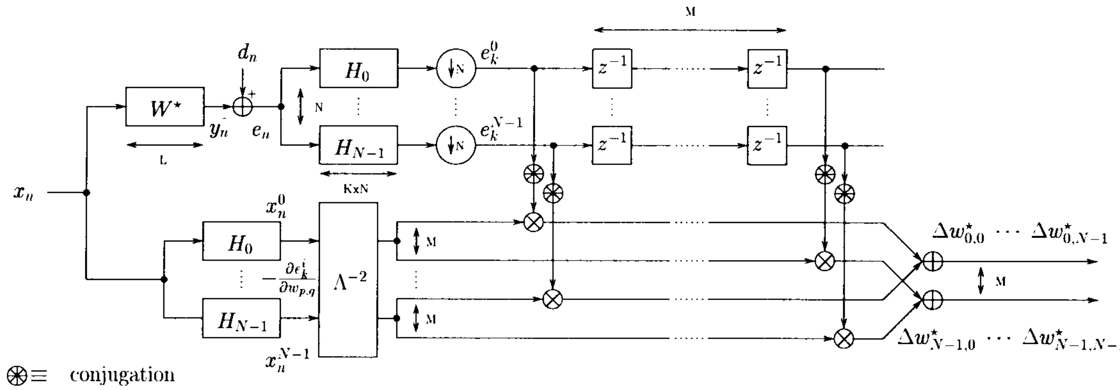

Fig. 5. Multirate adaptive filter scheme for $L = MN$.

computing its instantaneous gradient estimate $\hat{J}_k(W)$ relative to W

$$W^*_{(k+1)N} = W^*_{kN} - \mu \frac{\partial \hat{J}_k}{\partial W} = W^*_{kN} - \mu \sum_{i=0}^{N-1} e_k^i \left(\frac{\partial e_k^i}{\partial W^*} \right)^* \quad (3)$$

where μ is the positive scalar step size controlling the convergence of the adaptive process.

In order to calculate each partial derivative of the subband error, define

$$\begin{aligned} d_k^i &= H_i D_{kN} \\ X_n^{i^t} &= (x_n^i, \cdots, x_{n-L+1}^i) = H_i \mathcal{X}_n \end{aligned} \quad (4)$$

where X_n^i contains L nondecimated outputs of the ith analysis filter H_i excited by x_n, and d_k^i is the ith subband decomposition of the reference signal by the analysis bank.

This is also denoted by $e_k^i = H_i E_{kN}$ and

$$E_{kN} = D_{kN} - \mathcal{X}_{kN} W^*_{kN} \quad (5)$$

such that $e_k^i = d_k^i - X_{kN}^{i^t} W^*_{kN}$. Thus

$$\frac{\partial e_k^i}{\partial W^*} = -X_{kN}^i. \quad (6)$$

Hence, the tap update (3) reads

$$W^*_{(k+1)N} = W^*_{kN} + \mu \sum_{i=0}^{N-1} \lambda_i X_{kN}^{i^*} e_k^i \quad (7)$$

or, in a more compact form (matrix notation)

$$\begin{aligned} W^*_{(k+1)N} &= W^*_{kN} + \mu \Delta W^*_{kN} \\ \Delta W^*_{kN} &= (H\mathcal{X}_{kN})^H \Lambda^{-2} H E_{kN} \\ E_{kN} &= D_{kN} - \mathcal{X}_{kN} W^*_{kN} \end{aligned} \quad (8)$$

where $\Lambda^{-2} = \mathrm{Diag}(\lambda_0, \cdots, \lambda_{N-1})$ is the diagonal matrix of the weights λ_i.

The actual algorithm described by (8) is denoted as the WSAF.

Note that when $K = 1$ (i.e., H is an orthonormal transform) and $\lambda_i = 1\ \forall i$, the proposed algorithm reduces exactly to the BLMS algorithm.

A. Computational Structure

Equation (8) defines the generic version of the WSAF. However, careful consideration of (6) shows that some redundancy exists in the computations when $L = MN, M \in \mathbb{N}$ (i.e., the adaptive filter is larger than the number of subbands). This paragraph proposes a reduced complexity implementation scheme for the weight update.

If $w^*_{p,q}$ $(0 \le p \le N-1, 0 \le q \le M-1)$ denotes the $z^{-(p+Nq)}$ coefficient of W^* [i.e., qth coefficient of the pth polyphase component of $W^*(z)$], we can observe that

$$\frac{\partial e_k^i}{\partial w^*_{p,q+1}} = -x^i_{kN-p-(q+1)N} = -x^i_{(k-1)N-p-qN} = \frac{\partial e^i_{k-1}}{\partial w^*_{p,q}}. \quad (9)$$

Thus, the full vector $(\partial e_k^i / \partial W^*)_{0 \le i \le N-1}$ is obtained by collecting the different output vectors generated by the analysis filter bank whose input is (x_n) in which the subsampler has been removed. The resulting scheme is provided on Fig. 5, where Λ^{-2} represents the weighting of each subband by the corresponding factor λ_i.

B. Appropriate Choice of the Weights

At this point, it is not clear how the use of weighted mean square errors instead of regular ones could improve the adaptive algorithm convergence rate. In order to understand this mechanism, it is first necessary to rewrite (7) in terms of the contributions of each subband error to the total update increment

$$\begin{aligned} W^*_{kN} &= \sum_{i=0}^{N-1} W^{i^*}_{kN} \\ W^{i^*}_{(k+1)N} &= W^{i^*}_{kN} + \mu \lambda_i X^{i^*}_{kN} e_k^i. \end{aligned} \quad (10)$$

Consider the ideal case where the filter bank is composed of N ideal Nyquist filters that are adjacent and do not overlap (i.e., K tends to infinity $K \to +\infty$). Under this assumption, as the spectra of X_n^i are nonoverlapping, each term $\lambda_i X^{i^*}_{kN} e_k^i$ of the sum in (7) deals with adapting a different part $W^{i^*}_{(k+1)N}$ of the spectrum of W_n^* (cf., Fig. 4). Thus, the algorithm minimizes the quadratic errors in each subband independently with a different step size for each of them. Without any drawback, our algorithm is merely equivalent to *several LMS algorithms* working *separately* in *each subband*.

Furthermore, it is a well-known result [27] that under white noise input, the fastest convergence rate of the LMS algorithm is achieved for an adaptive step size given by $\mu_i = \mu\lambda_i = 1/(L\sigma_{x^i}^2)$, where $\sigma_{x^i}^2$ denotes the variance of the input signal (x_n^i in the present situation). The normalized LMS algorithm is usually defined with reference to this value of the adaptation step. Since the parameters μ and λ_i need to be tuned, we shall also choose λ_i so that under white noise input, the fastest convergence occurs independently in each subband when $\mu = 1$. Such a choice corresponds to

$$\lambda_i = 1/(L\sigma_{x^i}^2). \tag{11}$$

In practice, this set of weights can be used for parameterizing the WSAF but with a smaller μ: $\mu \leq 1$ in order to obtain various tradeoffs between convergence rate and residual error.

Interestingly, the choice of the set of λ_i determines the criterion to be minimized. Thus, this establishes a link between the speeding up of the convergence rate of the proposed algorithm and the formal statement of the criterion to be minimized.

This contrasts with classical frequency-domain or transform-domain adaptive approaches. Despite intuition, these do not minimize the error in each frequency band independently: instead of adapting each adaptive filter coefficient in the transform domain with respect to the full time domain error generated as any classical transform domain approach, the WSAF rather adapts each part of the spectrum of W^* according to the error produced in the same frequency subband (cf., Fig. 4).

Moreover, in the context of AEC, the WSAF enables perceptual considerations to be taken into account. Indeed, each subband error in the criterion can be weighted according the sensitivity of the ear applicable in that subband so that large errors can be tolerated without affecting the subjective quality of the AEC. In an extreme case, if no adaptation is required in some subbands, this can also result in complexity savings in the adaptation process by either including some decision functions to determine when to switch off the adaptation or preselecting the relevant subbands according to a perceptual knowledge.

IV. Algorithm Behavior

Without trying to determine precisely the adaptive behavior, this subsection intends to provide a more accurate understanding of the underlying mechanism, allowing a faster convergence for the WASF.

A. Convergence Rate

Let $R_{X^iX^i} = \mathcal{E}(X_n^i X_n^{i^H})$ denote the size L autocorrelation matrix of the nonsubsampled outputs of the ith filter H_i. Since the filters in the bank are selective, x_n^i has a narrow spectrum. Thus, $R_{X^iX^i}$ is expected to be badly conditioned. If the WSAF were to be strictly considered as several LMS algorithms working independently in each subband, we would expect a slow convergence rate for each LMS algorithm. This apparently contradicts the expected good convergence behavior of the WSAF. The following subsection shows that it is only an apparent contradiction.

Let W_o be the L-tap filter to be identified (of same length as W^*). The tap-error vector of the WSAF δW_{kN} is defined as $\delta W_{kN} = W_{kN} - W_o$. Suppose that the reference signal d_n is the sum of the true linear filtering of x_n by W_o and of a zero mean white Gaussian noise b_n. In vector notation, we have $D_n = \mathcal{X}_n W_o^* + B_n$, where $B_n = (b_n, \cdots, b_{n-KN+1})^t$. Under this assumption, the error can be expressed as$E_{kN} = D_{kN} - \mathcal{X}_{kN} W_{kN}^* = B_{kN} - \mathcal{X}_{kN}\delta W_{kN}^*$, and the WSAF adaptation equation reads

$$\begin{aligned}\delta W_{(k+1)N}^* =& [I_L - \mu(H\mathcal{X}_{kN})^H \Lambda^{-2} H\mathcal{X}_{kN}]\delta W_{kN}^* \\ &+ \mu(H\mathcal{X}_{kN})^H\Lambda^{-2}HB_{kN}.\end{aligned} \tag{12}$$

Under the usual so-called "independence" assumptions (i.e., the adaptive filter taps are uncorrelated from the input samples x_n, and x_n is not correlated to b_n), the above equation yields

$$\mathcal{E}(\delta W_{(k+1)N}^*) = \mathcal{E}[I_L - \mu(H\mathcal{X}_{kN})^H\Lambda^{-2}H\mathcal{X}_{kN}]\mathcal{E}(\delta W_{kN}^*). \tag{13}$$

The convergence rate of the WSAF is thus driven by the eigenvalues of matrix $I_L - \mu(H\mathcal{X}_{kN})^H\Lambda^{-2}H\mathcal{X}_{kN}$, which should be close to zero for faster convergence. Obviously, this could be feasible by some tuning of μ if the eigenvalue spread of matrix $M = \mathcal{E}[(H\mathcal{X}_{kN})^H\Lambda^{-2}H\mathcal{X}_{kN}]$ is close to 1. It can be shown in a straightforward way that $M = \mathcal{E}[(H\mathcal{X}_{kN})^H\Lambda^{-2}H\mathcal{X}_{kN}] = \Sigma_{i=0}^{N-1}\lambda_i R_{X^iX^i}$.

If the analysis bank is composed of perfect Nyquist brick filters (i.e., $K \to +\infty$), each individual matrix $R_{X^iX^i}$ is singular. However, since the filters in the bank are nonoverlapping and adjacent, a given eigenvalue will not be zero in all subbands. Therefore, the summation of all matrices $R_{X^iX^i}$ weighted by λ_i is not singular, and for each matrix $R_{X^iX^i}$, the weight λ_i plays the role of a normalization coefficient such that the eigenvalue spread of the whole summation is reduced.

This somewhat intuitive development has been checked by simulation by computing the eigenvalue spread of the matrices determining the convergence rate of the algorithm in the case of an autoregressive matrix of an order 2 (AR2) highly correlated input signal. This signal is obtained by filtering white Gaussian noise by the inverse of $G(z) = 1 - 1.6z^{-1} + 0.81z^{-2}$ (with zeroes such that $|z_i| = 0.9$). The WSAF with weights $\lambda_i = 1/(L\sigma_{x^i}^2)$ and increasing filter lengths in the bank is compared with the LMS algorithm. The number of subbands is set to $N = 10$, and the filter to be modeled has $L = 20$ taps (as does the adaptive filter). The length of the filters in the bank are 20 $(K = 1)$ for a discrete cosine transform (DCT$_{IV}$), 40 $(K = 2)$ for a modulated lapped transform (MLT), and 80 $(K = 4)$ for an extended lapped transform (ELT) [10], respectively. We can observe in Table I that the eigenvalue spread reduces when the filter lengths in the bank increase. In fact, $\Sigma_{i=0}^{N-1}\lambda_i R_{X^iX^i}$ has a much smaller condition number than R_{XX}; it has been reduced from 1400 to 2.2. This explains the better convergence behavior of the WSAF compared with the LMS algorithm, as well as the influence of the weights (in the definition of the

TABLE I
EIGENVALUE SPREAD OF THE MATRIX DETERMINING THE CONVERGENCE RATE

algorithm	*eigenvalue spread:* $\frac{\lambda_{\max}}{\lambda_{\min}}$
LMS ou WSAF(I_N)	1.4×10^3
WSAF(DCT$_{IV}$) $K = 1$	51.2
WSAF(MLT) $K = 2$	2.7
WSAF(ELT) $K = 4$	2.2

criterion) on the convergence rate of the associated stochastic gradient algorithm.

B. Residual Error

It is well known that the misadjustment of LMS-like algorithms depends on the adaptation step and on the amount of noise found in the reference signal. If the noise is white, and if the input signal is not, the relative noise level is different in each subband. Since, in some sense, the WSAF behaves like several LMS algorithms running in parallel, it is therefore suitable to have different tunings of the adaptation step in each subband. It turns out, as shown below, that a good choice for the weights makes use of the SNR in each subband.

First, decompose the error E_n as the sum of the modeling error $\varepsilon_n = (\epsilon_0, \cdots, \epsilon_{n-KN-1})^t$ plus the noise. Since $D_n = B_n + \mathcal{X}_n W_o^*$, we have $\varepsilon_n = \mathcal{X}_n \delta W_n^*$ yielding [cf., (8)]

$$W^*_{(k+1)N} = W^*_{kN} + \mu (H\mathcal{X}_{kN})^H \Lambda^{-2} H (B_{kN} - \mathcal{X}_{kN}\delta W^*_{kN}). \tag{14}$$

All signals x_n, b_n, d_n, and e_n are assumed to be ergodic and wide sense stationary. The filters in the bank are once again supposed to be nonoverlapping perfect Nyquist filters. Consequently, the subband outputs are uncorrelated. In the following, the notation d_k^i (signal d with superscript i and indexed by k) is used for the ith subband sample associated with block k, resulting from decomposition of d_n by H. Let us calculate the expectation of the squared norm of $\delta W_{(k+1)N}$ ($\|\delta W_n\|^2 = \delta W_n^H \delta W_n$). Under the assumption that $x_n \perp b_n$ and $b_n \perp \epsilon_n$ ($\perp$ stands for "is uncorrelated with") and designating by A_n the matrix $A_n = \Lambda^{-2} H \mathcal{X}_n \mathcal{X}_n^H H^H \Lambda^{-2}$, we have

$$\begin{aligned}\mathcal{E}(\|\delta W_{(k+1)N}\|^2) =& \mathcal{E}(\|\delta W_{kN}\|^2) + \mu^2 \mathcal{E}[(HB_n)^H A_{kN} H B_{kN}] \\ &+ \mu^2 \mathcal{E}[(H\varepsilon_{kN})^H A_{kN} H \varepsilon_{kN}] \\ &- 2\mu \mathcal{E}[(H\varepsilon_{kN})^H \Lambda^{-2} H \varepsilon_{kN}].\end{aligned} \tag{15}$$

It is well known that $\mathcal{E}(|\varepsilon_k^i|^2 |x_{n-k}^i|^2) \le \sigma_{\varepsilon^i}^2 \sigma_{x^i}^2$. Hence, define $a_i \ge 0$ as $\mathcal{E}(|\varepsilon_k^i|^2 |x_{n-k}^i|^2) = a_i \sigma_{\varepsilon^i}^2 \sigma_{x^i}^2$. The case $a_i = 1$ corresponds to the classical (but unrealistic unless the input is white noise) hypothesis that the modeling error ε_k^i is uncorrelated with the input signal, whereas smaller values $a_i < 1$ represent larger correlations.

Since, in this section, the filters of the bank are assumed to be ideal, for any $\{i, j\}$ $i \ne j$, we have $\varepsilon_k^i \perp \varepsilon_k^j, x_k^i \perp x_k^j$, and $x_k^i \perp \varepsilon_k^j$.

With all previous assumptions taken into account, relations

$$\begin{aligned}\sigma_b^2 &= \sigma_{b^i}^2, \\ \mathcal{E}[(H\varepsilon_{kN})^H \Lambda^{-2} H \varepsilon_{kN}] &= \sum_{i=0}^{L-1} \lambda_i \sigma_{\varepsilon^i}^2 \\ \mathcal{E}[(HB_{kN})^H A_{kN} H B_{kN}] &= L \sum_{i=0}^{L-1} \lambda_i^2 \sigma_{b^i}^2 \sigma_{x^i}^2, \\ \mathcal{E}[(H\varepsilon_{kN})^H A_{kN} H \varepsilon_{kN}] &= L \sum_{i=0}^{L-1} a_i \lambda_i^2 \sigma_{x^i}^2 \sigma_{\varepsilon^i}^2\end{aligned}$$

hold. At convergence, $\mathcal{E}(\|\delta W_{(k+1)N}\|^2) = \mathcal{E}(\|\delta W_{kN}\|^2)$, and (15) yields

$$\sum_{i=0}^{L-1} L\lambda_i \mu \left[\mu \lambda_i \sigma_{x^i}^2 (a_i \sigma_{\varepsilon^i}^2 + \sigma_{b^i}^2) - \frac{2}{L} \sigma_{\varepsilon^i}^2 \right] = 0. \tag{16}$$

As it is, this equation is not very tractable. However, due to the assumption of an ideal filter bank, it is reasonable to suppose that the part of the residual error in the ith subband ($\sigma_{\varepsilon^i}^2$) is due only to the signals x_k^i and b_k^i in this same subband. In this case, each individual term of the sum (16) is zero, and we have

$$\sigma_{\varepsilon^i}^2 \left(\frac{2}{L\sigma_{x^i}^2} - \mu \lambda_i a_i \right) = \mu \lambda_i \sigma_{b^i}^2, \qquad 0 \le i \le N-1 \tag{17}$$

which can be rewritten in terms of the modeling error variances $\sigma_{\varepsilon^i}^2 = (\mu \lambda_i \sigma_{b^i}^2 / (2/L\sigma_{x^i}^2) - \mu \lambda_i a_i)$.

In AEC applications, the adaptive filter is run in order to subtract an estimate of the echo (the amount of signal in the microphone that is a linear filtering of the loudspeaker signal) from the microphone signal. The error signal is, thus, what is sent back to the distant speaker. Reasonable specifications are given in terms of echo return loss enhancement (ERLE), which should be larger than some fixed quantity (say 20 dB).

In terms of the variables of the adaptive algorithm, this corresponds to the requirement that the asymptotic variance of the total misadjustment should be ρ times smaller than that of the total echo.

The flexibility of the WSAF allows the possibility of different requirements to be set in each subband. The tunings such that $\sigma_{\varepsilon^i}^2$ in each subband is asymptotically smaller than the variance of the actual echo $\sigma_{d^i}^2$ in the same subband (say, ρ_i times lower) can now be derived. This condition reads

$$\sigma_{\varepsilon^i}^2 \le \rho_i \sigma_{d^i}^2, \qquad 0 \le i \le N-1 \tag{18}$$

and, applying this inequality in the previous expression of $\sigma_{\varepsilon^i}^2$, results in $\rho_i \sigma_{d^i}^2 \ge (\mu \lambda_i \sigma_{b^i}^2 / (2/L\sigma_{x^i}^2) - \mu \lambda_i a_i)$.

An upper bound on the adaptation step sizes $\mu_i = \mu \lambda_i$ can thus be obtained from the above equation

$$\mu_i = \mu \lambda_i \le \frac{2\rho_i}{L\sigma_{x^i}^2 \left(a_i \rho_i + \frac{\sigma_{b^i}^2}{\sigma_{d^i}^2} \right)}, \qquad 0 \le i \le N-1. \tag{19}$$

The following are a few comments on (19).

- When the requirement described in (18) is loose (i.e., ρ_i is large), the adaptation step sizes tend to the regular ones given in the previous subsection.

- The smaller the SNR $\sigma^2_{d^i}/\sigma^2_{b^i}$, the smaller the adaptation step size should be in order to meet the requirement.
- With excellent SNR, the adaptation steps (19) reduce also to the regular ones

$$\mu_i = \mu\lambda_i \leq 2/(L\sigma^2_{x^i} a_i) \qquad 0 \leq i \leq L-1. \tag{20}$$

- The upper bound for the step sizes still holds when the noise is not assumed to be uncorrelated (white). The only difference in this case is that $\sigma^2_{b^i}$ are not equal. This is obtained under the assumption of perfect Nyquist filter banks.
- This strategy can be applied in a straightforward way to the BLMS or the LMS algorithm as a particular case of the WSAF, where all λ_i are equal: $\lambda_i = 1/\sigma^2_x$.
- This normalization aims at a given echo rejection level requiring $\sigma^2_{\varepsilon^i} \leq \rho_i \sigma^2_{d^i}$.
- The use of these new step-size formulas can only result in decreasing the adaptation step size, which cannot lead to instability problems.

This result shows another advantage brought by the algorithm: Each adaptation step size $\mu_i = \mu\lambda_i$ can be tuned according to the signal variance and noise variance in the same subband. This technique is of great interest when nonstationary signals such as speech are processed. This possibility is further worked out in the next subsection. Moreover, such different requirements (ρ_i) can be chosen according to the perceptual importance of each subband.

V. Further Refinements

This section proposes two possible refinements (by no means compulsory) that can be applied to the WSAF or any other adaptive algorithm in order to improve the convergence behavior.

A. A Time-Varying Strategy for the Weights and Step Size

Practical evaluation of all proposed expressions for the weights λ_i requires the computation of an average variance estimate of some signals. Moreover, since the signals (as well as the system to be identified) can be nonstationary, large variations of the variance of the observed signals can be expected. More than a 60-dB dynamic range is common for speech. This justifies the need for a time-varying strategy in the calculation of the WSAF weights λ_i, as proposed below.

Classically, as for the normalized version of the LMS algorithm [27], an exponential variance averaging of the signal with smoothing constant f is used. For instance, this amounts to the recursion for the time estimate Λ_k^{-2} of $\Lambda^{-2} = \mathrm{Diag}(\lambda_0, \cdots, \lambda_{N-1})$ in $\Lambda^2_{k+1} = f\Lambda^2_k + (1-f)\ \mathrm{Diag}[HX_{kN} \odot (HX_{kN})^*]$, where λ_i are the weights proposed in (11), and $f, 0 \leq f \leq 1$ is the forgetting factor. In that case, the update equation of the WSAF becomes

$$W^*_{(k+1)N} = W^*_{kN} + \mu(H\mathcal{X}_{kN})^H \Lambda_k^{-2} HE_{kN}. \tag{21}$$

The same technique can be applied for the proposed normalization of Section IV-B, aiming at ensuring a certain amount of echo rejection level. Denoting by $\hat{\sigma}^2_{x^i}(k), \hat{\sigma}^2_{b^i}(k)$ and $\hat{\sigma}^2_{d^i}(k)$ the estimated variances of signals x^i_k, b^i_k and d^i_k during block k, the weights $\Lambda_k^{-2} = \mathrm{Diag}[\lambda_0(k), \cdots, \lambda_{N-1}(k)]$ are given by [cf., (19)] $\lambda_i(k) = (2\rho_i/L\hat{\sigma}^2_{x^i}(k)(a_i\rho_i + (\hat{\sigma}^2_{b^i}(k)/\hat{\sigma}^2_{d^i}(k))))\ 0 \leq i \leq N-1$ and updated by

$$\hat{\sigma}^2_{X^i}(k+1) = f\hat{\sigma}^2_{X^i}(k) + (1-f)|X^i(k)|^2 \quad \text{where } X \in \{d^i, b^i, x^i\} \tag{22}$$

leading to the same adaptive equation for W^*_n as (21).

When a poor SNR occurs, the classical adaptation rule results in a somewhat erratic variation of the adaptive filter taps since the filter is driven mainly by noise. This can generate large modeling errors that are even larger than the actual echo. The adaptive filter must then converge again when the SNR improves. Of course, this phenomenon does not happen in stationary situations where the adaptation step size is tuned in order to obtain a given residual error. However, when the input signal variance varies a lot, the SNR also exhibits large variations and often reaches values for which the adaptation step is too large and temporarily amplifies the echo.

The time-varying strategy provided in this section allows the correction of this problem, even in the case of nonstationary signals. Indeed, if the SNR is low in a subband, the weight in that specific subband is also small. Thus, the adaptation process is slowed down in that very subband until a good SNR is encountered. It is illustrated in the simulations Section VI-C that in a context of AEC for nonstationary signals such as speech, time variable weights $\lambda_i(k)$ noticeably improve the convergence behavior of the WSAF in a noticeable manner.

B. Faster Update: The Refreshed WSAF

An improvement concerning the convergence rate of the WSAF can be obtained at the cost of additional computations with the same technique as used in the GMDF algorithm [9]. Indeed, the error samples present in the FB filters (for computing the decomposition of the error) can be re-evaluated at each update of the adaptive filter (if $K > 1$). In fact, if the filter size of the FB is large and if a big error is made at a particular moment, it will influence the FB output for a long time. Thus, in order to minimize this problem, it is possible to recompute at each iteration the elements present in the delay line of the filters as if they had resulted from the output of the last updated adaptive filter. A drawback of this method is the increase of the overall complexity. On the other hand, as illustrated later, it improves the convergence rate. In the following, this algorithm is denoted as the "refreshed WSAF" (RWSAF).

TABLE II
COMPLEXITY OF THE TRANSFORMS USED IN THE PAPER

$\mathbb{C}$ *transform*	$\mathbb{R}$ *multiplications*	$\mathbb{R}$ *additions*
DFT(L)	$L\log_2 L - 3L + 4$	$3L\log_2 L - 3L + 4$
DCT(L)	$L\log_2 L + 2$	$3L\log_2 L - 2L + 2$
DCT$_{IV}$(L)	$L\log_2 L + 2L$	$3L\log_2 L$
ELT$_{N\times KN}$ (K even)	$N(K + \log_2 N + 3)$	$N(K + 3\log_2 N + 1)$

TABLE III
COMPLEXITY OF VARIOUS ELEMENTARY TASKS

$\mathbb{C}$ *task*	$\mathbb{R}$ *multiplications*	$\mathbb{R}$ *additions*
modulus	2	1
$\mu_{\mathbb{C}}$ complex multiplication	3	3
$\alpha_{\mathbb{C}}$ complex addition	–	2
$\mu_{\mathbb{R}}$ real multiplication	–	1
$\alpha_{\mathbb{R}}$ real addition	1	–
$\mu_{\mathbb{RC}}$ a real times a complex	2	–
$N\|L$ $\mathcal{C}(N,L)$	$4+\frac{4L}{N}-6N+2L\log_2(2N)+2N\log_2(2N)$	$4+\frac{4L}{N}-6N+6L\log_2(2N)+6N\log_2(2N)$
$L\|N$ $\mathcal{C}(N,L)$	$-9N+\frac{8N}{L}+4N\log_2(2L)$	$-2L-7N+\frac{8N}{L}+12N\log_2(2L)$
$\mathcal{L}_L$ update of Λ^{+2}	$4L$	$2L$
$\mathcal{D}_L$ division by Λ^2	$2L$	–
$\mathcal{A}_L$ update of W	–	$2L$

VI. EVALUATION AND COMPARISON WITH OTHER ALGORITHMS

A. The Set of Algorithms to Be Compared

Since the WSAF uses a critically subsampled analysis filter bank, this algorithm is obviously bound to be of block type. However, one of the major drawbacks of block algorithms (at least in their initial version) is avoided since the block size is fully independent of the filter length. This is all the more important in AEC since we are working with very large filters (typically 1000 taps) and that the application limits the processing delay to at most 128 samples. Thus, working with (relatively) small blocks is a requirement.

Moreover, a fair algorithm evaluation requires the comparison of similar versions of the various algorithms of interest [frequency domain block LMS (FBLMS), transform domain adaptive filter (TDAF)]. Plenty of possibilities exist: small blocks, faster adaptation rate, etc. For completeness, some of the corresponding versions are recalled below.

• **Block TDAF**

Classically, in the TDAF, the update of the adaptive filter is performed at the sample rate. Thus, in order to compare the WSAF with various orthogonal transforms to the classical algorithms, a block version of the TDAF has to be derived. This is simply performed by constraining the taps of the adaptive filter to be updated once per N samples and minimizing the common block criterion J^{Block} [cf. (1)].

The classical time domain update equation of the TDAF $(N = L)$ is

$$\begin{aligned} W_{n+1} &= W_n + \mu T_L^{-1}\Lambda_n^{-2}T_L X_n e_n^* \\ e_n &= d_n - W_n^H X_n \end{aligned} \tag{23}$$

where μ denotes the positive scalar step size, and Λ_n^2 estimates the diagonal matrix of the average variance of the components of $T_L X_n$. When transformed into a block algorithm, this results in

$$\begin{aligned} W_{(k+1)N} &= W_{kN} + \frac{\mu}{N}T_L^{-1}\Lambda_{kN}^{-2}T_L\mathcal{X}_{kN}E_{kN}^* \\ E_{kN} &= D_{kN} - \mathcal{X}_{kN}^t W_{kN}^* \end{aligned} \tag{24}$$

for the block TDAF (BTDAF).

Small block versions are outlined below by similarity with already available versions of small block FDLMS (SBFDLMS) algorithms.

• **Small Block Algorithms**

Existing versions of small block LMS algorithms can be found in [8]. The corresponding scheme, which is called the multidelay block frequency adaptive filter (MDFAF), allows for the use of smaller blocks, hence introducing a smaller delay in the processing. Due to application requirements, such a constraint is a prerequisite for time varying systems such as acoustic path modeling.

The method used in [8] consists of dividing the time domain response of the adaptive filter in several portions of same length N (the block size), each one being adapted by the BLMS algorithm with the global time domain block error. Any fast convolution scheme [e.g., using fast Fourier transforms (FFT's)] can thus be used to perform the convolution for each subfilter as well as the correlations required for the update of coefficients.

The same technique can readily be applied to the FBLMS algorithm, resulting in the small block (SB) FBLMS (SBFBLMS) algorithm (which is a simpler version of the GMDF [9]).

When applied to the BTDAF, the same procedure results in the small block BTDAF (SBBTDAF). Sections VI-C and IV-B provide an evaluation of the performance and the required complexity of these algorithms compared to the WSAF.

B. Arithmetic Complexity and Processing Delay

A first criterion for algorithm comparison is arithmetic complexity, which allows the evaluation of one of the terms of the complexity/efficiency tradeoff. The complexity is here evaluated in terms of the number of real multiplications and of real additions ($\mu_{\mathbb{R}}$ and $\alpha_{\mathbb{R}}$). All algorithms are considered under their complex valued versions (the more general one is usable in an equalization context). In an AEC context, where all variables are real, the given values will thus need to be approximately halved.

This complexity is conveniently expressed in terms of basic operations (transforms, filters, and correlations), and we first recall the corresponding number of multiplications and additions.

Basic Operators: In the following $\alpha_{\mathbb{C}}, \mu_{\mathbb{C}}$, and $\mu_{\mathbb{RC}}$ denote the complexity of the complex addition, multiplication, and of a real times a complex, respectively. Based on the techniques described in [28], the computational cost of these basic operations is evaluated in Table III.

Transforms: Table II gives the arithmetic complexity of the complex and real transforms used in the simulations, as given in [10]. In the following and for the simulations, the filter bank H used as an orthogonal transform when the number of analysis filters taps is larger than the number of subbands, is an ELT [10].

More generally, $\mathcal{H}_{N\times KN}$ stands for the arithmetic complexity needed to compute one output vector of an N-band filter bank with filters of size KN for a given input sequence. $\mathcal{H}^H_{KN\times N}$ represents the product complexity of H^H by a KN components vector.

In the special case, where H is an ELT, the output vector of H^H can be calculated very efficiently using a DCT_{IV} [10]. The corresponding complexity is

$$\text{complexity}(\mathrm{ELT}^H_{KN\times N}, \text{K even}) = \mathrm{DCT}_{IV}(N) + KN\mu_{\mathbb{RC}}. \tag{25}$$

Correlation and Convolution: In order to evaluate the complexity of the various algorithms, it is assumed that fast convolution and correlation is achieved using the fast Fourier transform (FFT)-based algorithms. Both operations have the same cost $\mathcal{C}(N,L)$, where $\mathcal{C}(N,L)$ stands for the complexity of the fast product of a L-dimension vector Y by a $N\times L$, not necessarily square matrix $\mathcal{X}$ made of Hankel blocks (each of the form $(x_{i+j})_{(i,j)\in\mathbb{N}^2_{N-1}}$): $\mathcal{X}Y$. Two cases have to be considered:

- If $L|N$ (i.e., the remainder of the integer division of N by L is zero), the matrix is divided in $(N/L)L\times L$ square Hankel submatrices, as depicted in Fig. 6. Thus, the result of the multiplication is the concatenation of several products of these Hankel matrices by the same original vector. This can be achieved in a very efficient way with an overlap save (OLS) [29], [30] technique and results in the following: For $L|N, \mathcal{C}(N,L) = ((N/L)+1)\ \mathrm{DFT}_{\mathbb{C}}(2L) + 2N\mu_{\mathbb{C}}$.
- If $N|L$, the vector and the matrix are split into $(L/N)N$-dimension vectors and $N\times N$ square Hankel matrices, respectively, as illustrated in Fig. 6. The result of the multiplication is the summation of the products of these Hankel matrices by the corresponding vectors. Again, each term of the summation is computed applying an OLS technique, yielding

$$N|L, \mathcal{C}(N,L) = \frac{2L}{N}\ \mathrm{DFT}_{\mathbb{C}}(2N) + 2L\mu_{\mathbb{C}} + \left(\frac{L}{N}-1\right)N\alpha_{\mathbb{C}}. \tag{26}$$

Note that since one $\mathrm{DFT}_{\mathbb{C}}$ is often shared by the fast correlation and convolution, this DFT has been counted only once.

Exponential Weighting Update: The update $\Lambda^2_{n+1} = f\Lambda^2_n + (1-f)\ \mathrm{Diag}[T_L X_n \odot (T_L X_n)^*]$ of the exponential window estimate of the mean square components of the input transformed vector has a complexity of $\mathcal{L}_L = L$ modulus $+2L\mu_{\mathbb{R}} + L\alpha_{\mathbb{R}}$. $\mathcal{A}_L = L\alpha_{\mathbb{C}}$ is the number of adds required for adding the precomputed update δW^*_n to the L-tap adaptive filter W^*_n, whereas $\mathcal{D}_L = L\mu_{\mathbb{RC}}$ denotes the cost of the multiplication by Λ^{-2}_n.

The resulting numbers of real additions and multiplications are summarized in Table III.

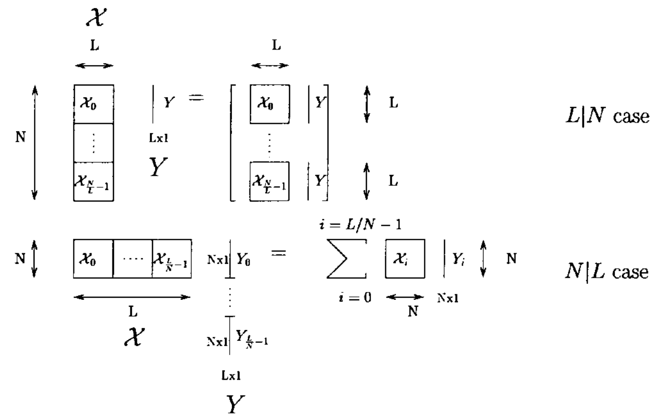

Fig. 6. Decomposition of the product of a block Hankel matrix by a vector for fast computation.

Adaptive Algorithms: The complexity of the adaptive algorithms using an orthogonal transform can often be reduced in the case of the DFT. Therefore, particular attention will be paid to this special case in the following.

Since most considered algorithms are of block type, a useful figure of merit for comparing them is the complexity *per input sample*.

The arithmetic complexity of each algorithm per input sample in terms of the elementary tasks detailed above is given in Table IV ($\mathcal{T}_L$ denotes the complexity of a square transform of dimension L). The corresponding analytical expressions are detailed in Table V, whereas Table VI gathers the numerical values of these formulae for an $L = 1024$ tap adaptive filter: $N = 32$ and $K = 2$. In these tables, for plain block transform domain adaptive filters, the block length is assumed to be equal to the filter size (L), whereas in the WSAF case, the block length is the number of subbands (N). Note that, for simplicity, the complexity of the RWSAF is given only in the case where $L \geq N$ (i.e., the adaptive filter length is larger than the number of subbands) and $K > 1$ (i.e., the filterbank does not reduce to a square transform).

We can observe [cf., (6)] that for an MLT filter bank of $N = 32$ subbands (i.e., $K = 2$ ELT case), $\mu_{\mathbb{R}} = 1013.38$, and $\alpha_{\mathbb{R}} = 3725.38$. In comparison with a small block version of the FBLMS and the SBFBLMS (where the adaptive filter is split into equal sized vectors whose size is equal to the block length, each of them being adapted by an FBLMS), the complexities of the WSAF and the RWSAF are 16% and 6% smaller, respectively. The additional cost of employing the refreshed version of the WSAF is quite small (about 10%). Moreover, it can be checked that the WSAF based on square transforms has the same complexity as the BTDAF based on the same transform.

Processing Delay: Since the WSAF is intrinsically a block algorithm, it introduces the same processing delay as any other block algorithm with same block size (N). Moreover, one of the WSAF advantages is that this processing delay is formally linked to neither the adaptive filter size (a property shared by the small block versions of any block algorithm) nor to the filterbank decorrelation efficiency (i.e., the length of the subband filters).

TABLE IV
META-COMPLEXITIES

algorithm	*complexity*
LMS	$2L\mu_{\mathbb{C}} + \mu_{\mathbb{RC}} + 2L\alpha_{\mathbb{C}}$
hline BLMS, $N = L$	$[\mathcal{A}_L + \mathcal{C}(L,L) + \mathrm{DFT}_{\mathbb{C}}(2L) + \mathcal{C}(L,L) + L\alpha_{\mathbb{C}} + L\mu_{\mathbb{RC}}]/L$
FBLMS, $N = L$	$[\mathcal{A}_L + \mathcal{L}_{2L} + \mathcal{D}_{2L} + \mathcal{C}(L,L) + \mathrm{DFT}_{\mathbb{C}}(2L) + \mathcal{C}(L,L) + L\alpha_{\mathbb{C}}]/L$
BTDAF, $N = L$, T_L	$[\mathcal{A}_L + \mathcal{L}_L + \mathcal{D}_L + \mathcal{C}(L,L) + \mathrm{DFT}_{\mathbb{C}}(2L) + 3\mathcal{T}_L + \mathcal{C}(L,L) + L\alpha_{\mathbb{C}}]/L$
BTDAF, $N = L$, T_{2L}	$[\mathcal{A}_L + \mathcal{L}_{2L} + \mathcal{D}_{2L} + \mathcal{C}(2L,L) + \mathrm{DFT}_{\mathbb{C}}(2L) + 3\mathcal{T}_{2L} + \mathcal{C}(L,L) + L\alpha_{\mathbb{C}}]/L$
BTDAF, $N = L$, DFT_{2L}	$[\mathcal{A}_L + \mathcal{L}_{2L} + \mathcal{D}_{2L} + \mathcal{C}(2L,L) + \mathrm{DFT}_{\mathbb{C}}(2L) + 2\mathrm{DFT}_{\mathbb{C}}(2L) + \mathcal{C}(L,L) + L\alpha_{\mathbb{C}}]/L$
SBBLMS, $N\|L$	$[\mathcal{A}_L + \mathcal{C}(L,N) + \mathrm{DFT}_{\mathbb{C}}(2N) + \mathcal{C}(N,L) + N\alpha_{\mathbb{C}} + L\mu_{\mathbb{RC}}]/N$
SBFBLMS, $N\|L$	$[\mathcal{A}_L + \mathcal{L}_{2N} + \frac{L}{N}(\mathcal{D}_{2N} + \mathcal{C}(N,N)) + \mathrm{DFT}_{\mathbb{C}}(2N) + \mathcal{C}(N,L) + N\alpha_{\mathbb{C}}]/N$
SBBTDAF, $N\|L$, T_L	$[\mathcal{A}_L + \mathcal{L}_N + \mathcal{T}_N + \frac{L}{N}(\mathcal{D}_N + \mathcal{C}(N,N) + 2\mathcal{T}_N) + \mathrm{DFT}_{\mathbb{C}}(2N) + \mathcal{C}(N,L) + N\alpha_{\mathbb{C}}]/N$
SBBTDAF, $N\|L$, T_{2L}	$[\mathcal{A}_L + \mathcal{L}_{2N} + \mathcal{T}_{2N} + \frac{L}{N}(\mathcal{D}_{2N} + \mathcal{C}(2N,N) + 2\mathcal{T}_{2N}) + \mathrm{DFT}_{\mathbb{C}}(2N) + \mathcal{C}(N,L) + N\alpha_{\mathbb{C}}]/N$
SBBTDAF, $N\|L$, $\mathrm{DFT}_{\mathbb{C}}(2L)$	$[\mathcal{A}_L + \mathcal{L}_{2N} + \frac{L}{N}(\mathcal{D}_{2N} + \mathcal{C}(2N,N) + 2\mathrm{DFT}_{\mathbb{C}}(2N)) + \mathrm{DFT}_{\mathbb{C}}(2N) + \mathcal{C}(N,L) + N\alpha_{\mathbb{C}}]/N$
WSAF, $KN\|L$, $H_{N\times KN}$	$[\mathcal{A}_L + \mathcal{L}_N + \mathcal{H} + \mathcal{D}_N + \mathcal{C}(L,KN) + \mathrm{DFT}_{\mathbb{C}}(2KN) + \mathcal{H}^H + \mathcal{C}(N,L) + \mathrm{DFT}_{\mathbb{C}}(2N) + N\alpha_{\mathbb{C}} + \mathcal{H}]/N$
WSAF, $L\|KN$, $H_{N\times KN}$	$[\mathcal{A}_L + \mathcal{L}_N + \mathcal{H} + \mathcal{D}_N + \mathcal{C}(L,KN) + \mathrm{DFT}_{\mathbb{C}}(2L) + \mathcal{H}^H + \mathcal{C}(N,N) + \mathrm{DFT}_{\mathbb{C}}(2N) + N\alpha_{\mathbb{C}} + \mathcal{H}]/N$
RWSAF, $KN\|L$, $K > 1$, $L \geq N$, $H_{N\times KN}$	$[\mathcal{A}_L + \mathcal{L}_N + \mathcal{H} + \mathcal{D}_N + \mathcal{C}(L,KN) + \mathrm{DFT}_{\mathbb{C}}(2KN) + \mathcal{H}^H + \mathcal{C}(KN,L) + KN\alpha_{\mathbb{C}} + \mathcal{H}]/N$
RWSAF, $L\|KN$, $K > 1$, $L \geq N$, $H_{N\times KN}$	$[\mathcal{A}_L + \mathcal{L}_N + \mathcal{H} + \mathcal{D}_N + \mathcal{C}(L,KN) + \mathrm{DFT}_{\mathbb{C}}(2L) + \mathcal{H}^H + \mathcal{C}(KN,KN) + \mathrm{DFT}_{\mathbb{C}}(2KN) + KN\alpha_{\mathbb{C}} + \mathcal{H}]/N$

TABLE V
COMPLEXITY OF THE VARIOUS SIMULATED ALGORITHMS

algorithm	$\mathbb{R}$ *multiplications*	$\mathbb{R}$ *additions*
LMS	$2 + 6L$	$10L$
BLMS	$-16 + \frac{20}{L} + 10\log_2(2L)$	$-14 + \frac{20}{L} + 30\log_2(2L)$
FBLMS	$-6 + \frac{20}{L} + 10\log_2(2L)$	$-10 + \frac{20}{L} + 30\log_2(2L)$
BTDAF, $T_L = \mathrm{DFT}_L$	$-21 + \frac{32}{L} + 3\log_2(L) + 10\log_2(2L)$	$-21 + \frac{32}{L} + 9\log_2(L) + 30\log_2(2L)$
BTDAF, $T_L = \mathrm{DCT}_L$	$-12 + \frac{26}{L} + 3\log_2(L) + 10\log_2(2L)$	$-18 + \frac{26}{L} + 9\log_2(L) + 30\log_2(2L)$
SBBLMS	$-12 + \frac{4}{L} + \frac{4L}{N^2} + \frac{12}{N} - \frac{4L}{N} + 2\log_2(2L) + \frac{2L\log_2(2L)}{N} + 4\log_2(2N) + \frac{2L\log_2(2N)}{N}$	$-10 + \frac{4}{L} + \frac{4L}{N^2} + \frac{12}{N}\frac{4L}{N} + 6\log_2(2L) + \frac{6L\log_2(2L)}{N} + 12\log_2(2N) + \frac{6L\log_2(2N)}{N}$
SBFBLMS	$2 + \frac{16L}{N^2} + \frac{4}{N} - \frac{11L}{N} + 2\log_2(2N) + \frac{8L\log_2(2N)}{N}$	$-2 + \frac{16L}{N^2} + \frac{4}{N} - \frac{11L}{N} + 6\log_2(2N) + \frac{24L\log_2(2N)}{N}$
SBBTDAF, $T_N = \mathrm{DFT}_N$	$-5 + \frac{24L}{N^2} + \frac{8}{N} - \frac{19L}{N} + \log_2(N) + \frac{2L\log_2(N)}{N} + 2\log_2(2N) + \frac{8L\log_2(2N)}{N}$	$-7 + \frac{24L}{N^2} + \frac{8}{N} - \frac{17L}{N} + 3\log_2(N) + \frac{6L\log_2(N)}{N} + 6\log_2(2N) + \frac{24L\log_2(2N)}{N}$
SBBTDAF, $T_N = \mathrm{DCT}_N$	$-2 + \frac{20L}{N^2} + \frac{6}{N} - \frac{13L}{N} + \log_2(N) + \frac{2L\log_2(N)}{N} + 2\log_2(2N) + \frac{8L\log_2(2N)}{N}$	$-6 + \frac{20L}{N^2} + \frac{6}{N} - \frac{15L}{N} + 3\log_2(N) + \frac{6L\log_2(N)}{N} + 6\log_2(2N) + \frac{24L\log_2(2N)}{N}$
WSAF, $KN\|L$, $H_{N\times KN} = \mathrm{ELT}$	$8 - 8K + \frac{8L}{N^2} + \frac{4L}{KN^2} + \frac{12}{N} - \frac{9L}{N} + 3\log_2(N) + 2\log_2(2N) + \frac{4L\log_2(2N)}{N} + 4K\log_2(2KN) + \frac{2L\log_2(2KN)}{N}$	$-2 - 10K + \frac{8L}{N^2} + \frac{4L}{KN^2} + \frac{12}{N} - \frac{5L}{N} + 9\log_2(N) + 6\log_2(2N) + \frac{12L\log_2(2N)}{N} + 12K\log_2(2KN) + \frac{6L\log_2(2KN)}{N}$
WSAF, $L\|KN$, $H_{N\times KN} = \mathrm{ELT}$	$2 - 5K + \frac{8K}{L} + \frac{16}{N} - \frac{6L}{N} + 4K\log_2(2L) + \frac{2L\log_2(2L)}{N} + 3\log_2(N) + 6\log_2(2N)$	$-6 - 5K + \frac{8K}{L} + \frac{16}{N} - \frac{6L}{N} + 12K\log_2(2L) + \frac{6L\log_2(2L)}{N} + 9\log_2(N) + 18\log_2(2N)$
RWSAF, $KN\|L$, $H_{N\times KN} = \mathrm{ELT}$	$14 - 8K + \frac{12L}{KN^2} + \frac{8}{N} - \frac{9L}{N} + 3\log_2(N) + 4K\log_2(2KN) + \frac{6L\log_2(2KN)}{N}$	$4 - 10K + \frac{12L}{KN^2} + \frac{8}{N} - \frac{5L}{N} + 9\log_2(N) + 12K\log_2(2KN) + \frac{18L\log_2(2KN)}{N}$
RWSAF, $L\|KN$, $H_{N\times KN} = \mathrm{ELT}$	$14 - 17K + \frac{8K}{L} + \frac{16}{N} - \frac{6L}{N} + 4K\log_2(2L) + \frac{2L\log_2(2L)}{N} + 3\log_2(N) + 6K\log_2(2KN)$	$4 - 15K + \frac{8K}{L} + \frac{16}{N} - \frac{6L}{N} + 12K\log_2(2L) + \frac{6L\log_2(2L)}{N} + 9\log_2(N) + 18K\log_2(2KN)$
WSAF, $N = L$, $K = 1$, $H_{L\times L} = \mathrm{DFT}_L$	same complexity as the BTDAF, $T_L = \mathrm{DFT}_L$	
WSAF, $N = L$, $K = 1$, $H_{L\times L} = \mathrm{DCT}_L$	same complexity as the BTDAF, $T_L = \mathrm{DCT}_L$	

C. Simulations

This section presents some simulations that are aimed at comparing the convergence behavior of the various algorithms. All simulations are run in a context of adaptive modelization.

- **Convergence Behavior for an AR2 Highly Correlated Input Signal**

Context: In Figs. 7 and 8, the input x_n is a highly correlated AR2 signal obtained by filtering a white Gaussian noise by

TABLE VI
NUMERICAL EVALUATION OF THE COMPLEXITIES FOR $L = 1024, N = 32$ AND $K = 2$

algorithm	$\mathbb{R}$ *multiplications*	$\mathbb{R}$ *additions*
LMS	6146	10240
BLMS	94.0195	316.02
FBLMS	104.02	320.02
BTDAF, $T_L = \mathrm{DFT}_L$	119.031	399.031
BTDAF, $T_L = \mathrm{DCT}_L$	128.025	402.025
SBBLMS	998.379	3268.38
SBFBLMS	1214.12	4306.12
SBBTDAF, $T_N = \mathrm{DFT}_N$	1284.25	5092.25
SBBTDAF, $T_N = \mathrm{DCT}_N$	1475.19	5153.19
WSAF, $H_{N\times KN} = \mathrm{ELT}$	1013.38	3725.38
RWSAF, $H_{N\times KN} = \mathrm{ELT}$	1131.25	4075.25
WSAF, $N = L$, $K = 1$, $H_{L\times L} = \mathrm{DFT}_L$	same complexity as the BTDAF, $T_L = \mathrm{DFT}_L$	
WSAF, $N = L$, $K = 1$, $H_{L\times L} = \mathrm{DCT}_L$	same complexity as the BTDAF, $T_L = \mathrm{DCT}_L$	

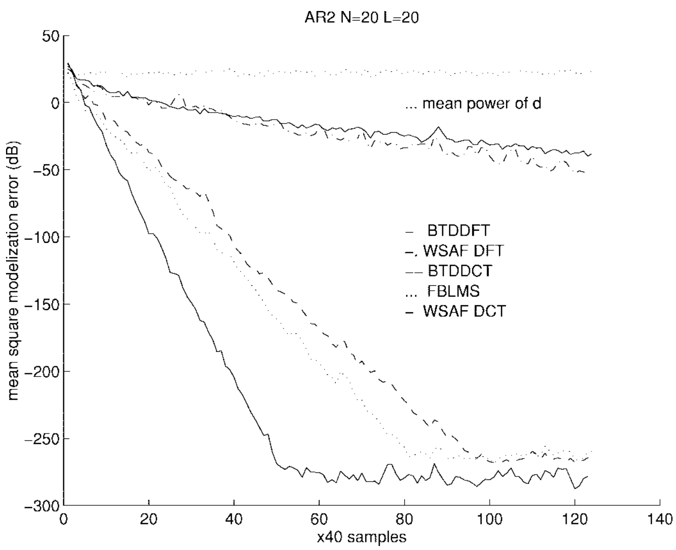

Fig. 7. Performance of some exhibited algorithms for step sizes μ tuned for fastest convergence. All algorithms are similarly tuned. Same block size and filter length L.

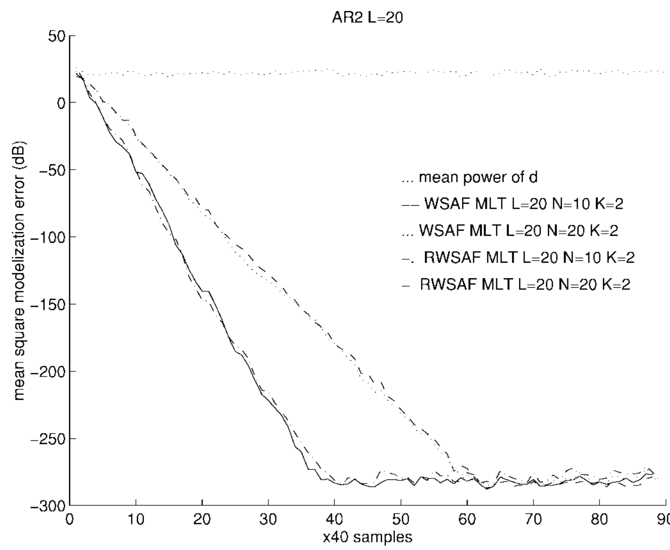

Fig. 8. Performance of the WSAF and RWSAF using an MLT $(K = 2)$ as analysis bank and a step size μ tuned for fastest convergence.

$1/G(z) = 1/(1 - 1.6z^{-1} + 0.81z^{-2})$. The adaptive filter as well as the filter to be identified both have $L = 20$ taps. The simulations are performed in a non-noisy context (no modeling noise), and the step size μ is tuned in order to achieve the fastest convergence rate.

Classical Square Transforms, Block Size Equal to the Adaptive Filter Length: Fig. 7 compares the various convergence behaviors of the FBLMS algorithm, the BTDAF, and the WSAF with the same block size (set to the adaptive filter length L) and using two square transforms: the DFT and the DCT.

Note that for the FBLMS algorithm, the DFT_{2L} dimension is two times the block size and thus performs a better decorrelation. This explains its better convergence rate compared with both the WSAF and the BTDAF with a DFT_L as transform. However, in all the cases, the WSAF has a better behavior than the BTDAF for comparable transforms. Moreover, when replacing the DFT_L with the DCT_L as a filter bank in order to decorrelate further the outputs of the transform, the WSAF convergence rate is even faster than that of the FBLMS with its transform two times larger.

RWSAF versus WSAF with Various Block Lengths: In Fig. 8, the WSAF makes use of a modulated lapped transform (MLT) as the filter bank. This case corresponds to a filter length being equal to twice the number of subbands (i.e., $K = 2$) [10]. The aim of this simulation is twofold. First, it intends to present the convergence rate improvement when refreshing the error samples present in the delay line of the analysis bank at each of the adaptive filter updates (as described in Section V-B). Second, it illustrates the convergence behavior of the WSAF in the case of a nonsquare transform.

In these simulations, the fastest convergence rate is reached by the RWSAF, where the filter bank is an MLT with $N = 20, K = 2$, and $L = 20$. The expected improvement of the convergence rate between the RWSAF and WSAF is clearly noticeable and would appear to be worth the only 10% increase in the complexity. Increasing the lengths of the filters in the bank for a constant number of subbands (from an DCT_L in Fig. 7 to the MLT $L = N = 20$ and $K = 2$ in Fig. 8) can lead to significant improvement in convergence rate but only for the refreshed version of the WSAF (RWSAF).

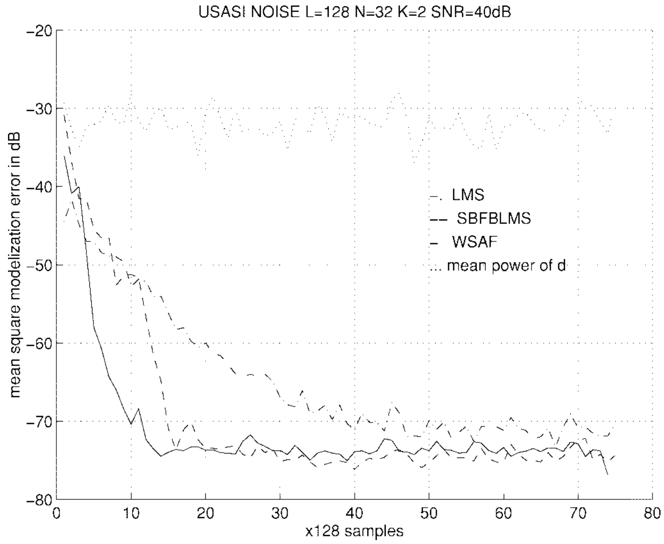

Fig. 9. Comparison of the convergence curves of the WSAF, SBFBLMS, and LMS algorithms.

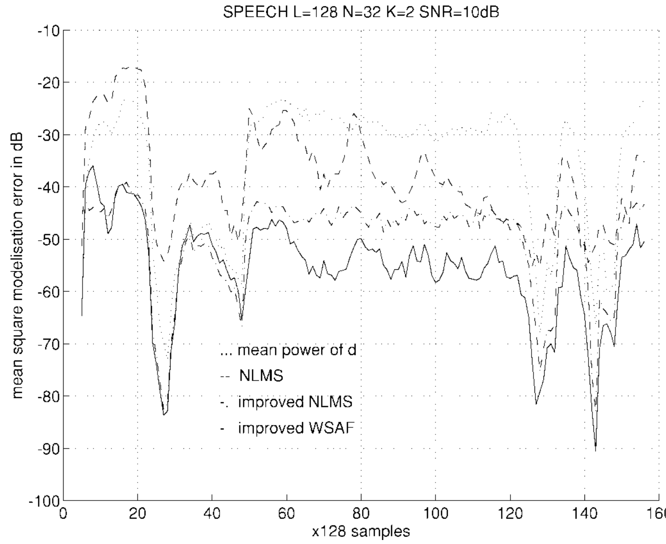

Fig. 10. Comparison of the convergence curves of the LMS, improved LMS and improved WSAF algorithms for speech data.

- **Acoustic Echo Cancellation Framework**

In practice, it is interesting to use small transforms (i.e., few subbands) and larger adaptive filters (e.g., in the context of AEC). This is readily achievable for the WSAF because no link between the adaptive filter length and the number of subbands is required in its original version (see Section VI-A). The comparison is made with small block versions of the FBLMS algorithm and the BTDAF.

Context: The filter to be identified has 128 taps (it is the truncation of the acoustic response of a room), and so does the adaptive filter ($L = 128$). The parameters of the WSAF algorithm are $N = 32$ and $K = 2$. The filter bank is again an MLT, and once again, the step size of each algorithm is also chosen to obtain the fastest convergence rate. The noise is subtracted from the error before computing its mean squared value.

Adaptive Filter Length Greater than the Block Size: Fig. 9 compares the new algorithm WSAF to the LMS and SBFBLMS algorithms. The simulation is run in the context of adaptive modeling. The input of the adaptive filter is stationary noise with on average the same spectrum as speech. White noise is added to the reference signal and the output signal-to-noise ratio (SNR) is 40 dB.

In all the simulations we have run, the proposed algorithm is clearly an improvement over the LMS and the SBFBLMS algorithms for which the parameters were similarly tuned: We have the same number of row vectors in the transform and same block sizes.

Time-Varying Strategy for the Weights for Nonstationary Signals: Finally, Fig. 10 shows a classical normalized LMS (NLMS) algorithm, an improved NLMS algorithm (with the same time-varying strategy as we propose for the WSAF), and an improved WSAF in the same context as previously described (AEC) but with a real speech signal as input and a SNR of 10 dB. It clearly appears that while the improvement provided by the time-varying strategy on the NLMS is important both in terms of convergence and residual error, the same strategy applied to the WSAF allows a further improvement of about 7 dB on the residual error without loss in terms of convergence rate.

VII. Conclusion

In this paper, a new subband adaptive scheme using LL PR FB is presented: the weighted subband adaptive filter. Like classical approaches (BTDAF or the FBLMS algorithm), the WSAF uses an orthonormal projector for convergence rate improvement, but it is used in a different manner. In the WSAF, it is the modeling error e_n that is projected in place of x_n (classical schemes). We can highlight the following distinctions between both types of algorithms:

- **BTDAF-like algorithms, including the FBLMS**: These algorithms use the common time domain block criterion J^{Block} for the adaptive filter tap update. The convergence behavior improvement is obtained by introducing different step sizes for the adaptation of each of the adaptive filter coefficients in the transform domain.
- **The WSAF algorithm**: Here, the different weights are introduced through a block weighted criterion and the use of critically subsampled lossless perfect reconstruction filter banks. Hence, the weights are applied on the spectral contents.

The WSAF offers more flexibility for the choice of the different parameters involved in the adaptive process. Indeed, for the WSAF, the size of the transform (FB) is independent of the adaptive filter length and is only linked to the block size. This gives the required degrees of freedom, compared with classical transform domain-based approaches. Moreover, in the WSAF, the size of the transform is not necessarily square, and this allows a better decorrelation property without increasing the block size, which is not easy to achieve in the original versions of the classical algorithms. Finally, this approach presents some links between *appropriately weighted* least square minimization and the improvement of the adaptive process convergence rate.

In the same conditions, the WSAF is shown to perform a better convergence rate than the small block implementation of the FBLMS (SBFBLMS) while offering a similar (and even slightly lower) complexity.

Furthermore, a time-varying strategy for the step size of the adaptation that replaces the natural normalization of the WSAF is discussed. This technique is relevant for signals having large variance variations such as speech in an AEC context. In this case, the technique is shown to further improve the adaptive behavior of the algorithm.

REFERENCES

[1] B. Widrow, "Adaptive filters," in *Aspects of Network and System Theory*, R. Kalman and N. DeClaris, Eds. New York: Holt, Rinehart, and Winston, 1971, pp. 563–587.

[2] M. Dentino, J. McCool, and B. Widrow, "Adaptive filtering in the frequency domain," *Proc. IEEE*, vol. 66, pp. 1658–1659, Dec. 1978.

[3] E. R. Ferrara, "Fast implementation of LMS adaptive filters," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-28, pp. 474–475, Aug. 1980.

[4] 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.

[5] 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.

[6] ——, "Performance analysis of frequency-domain block LMS adaptive digital filters," *IEEE Trans. Circuits Syst.*, vol. 36, pp. 173–189, Feb. 1989.

[7] D. F. Marshall, W. K. Jenkins, and J. J Murphy, "The use of orthogonal transform for improving performances of adaptive filters," *IEEE Trans. Circuits Syst.*, vol. 36, pp 474–484, Apr. 1989.

[8] J.-S. Soo and K. K. Pang, "Multi block frequency domain adaptive filter," *IEEE Trans. Signal Processing*, vol. 38, pp. 373–376, Feb. 1990.

[9] E. Moulines, O. AitAmrane, and Y. Grenier, "The generalized multidelay filter: Structure and convergence analysis," *IEEE Trans. Signal Processing*, vol. 43, pp. 14–28, Jan. 1995.

[10] H. S. Malvar, *Signal Processing with Lapped Transforms*. Norwood, MA: Artech House, 1992.

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

[12] N. Erdol and F. Basbug, "Performance of wavelet transform based adaptive filters," in *Proc. Int. Conf. Acoust., Speech, Signal Process.*, vol. 3, pp. 500–503, Minneapolis, MN, Apr. 1993.

[13] S. Hosur and A. H. Tewfik, "Wavelet transform domain LMS algorithm," in *Proc. Int. Conf. Acoust., Speech, Signal Process.*, Minneapolis, MN, Apr. 1993, vol. 3, pp. 508–510.

[14] M. Doroslovački and H. Fan, "Wavelet-based adaptive filtering," in *Proc. Int. Conf. Acoust., Speech, Signal Process.*, Minneapolis, MN, Apr. 1993, vol. 3, pp. 488–491.

[15] S. Attallah and M. Najim, "O the convergence enhancement of the wavelet transform based LMS," in *Proc. Int. Conf. Acoust., Speech, Signal Process.*, Detroit, MI, May 1995, vol. 2, pp. 973–976.

[16] ——, "A fast wavelet transform-domain LMS algorithm," in *Proc. Int. Conf. Acoust., Speech, Signal Process.*, Atlanta, GA, May 1996, vol. 3, pp. 1343–1346.

[17] M. de Courville and P. Duhamel, "Adaptive filtering in subbands using a weighted criterion," in *Proc. Int. Conf. Acoust., Speech, Signal Process.*, Detroit, MI, May 1995, vol. 2, pp. 985–988.

[18] A. Gilloire and M. Vetterli, "Adaptive filtering in subbands with critical sampling: Analysis, experiments, and application to acoustic echo cancellation," *IEEE Trans. Signal Processing*, vol. 40, pp. 1862–1875, Aug. 1992.

[19] H. Yasukawa, S. Shimada, and I. Furukawa, "Acoustic echo canceller with high speech quality," in *Proc. Int. Conf. Acoust., Speech, Signal Process.*, Dallas, TX, Apr. 1987, vol. 4, pp. 2125–2128.

[20] W. Kellermann, "Kompensation Akustischer Echos in Frequenzteilbandern," *Frequenz*, vol. 39, pp. 209–215, July 1985.

[21] M. R. Petraglia and S. K. Mitra, "Performance analysis of adaptive filter structures based on subband decomposition," in *Proc. Int. Symp. Circuits Syst.*, Chicago, IL, June 1993, vol. 1, pp. 60–63.

[22] A. Gilloire, "Experiments with sub-band acoustic echo cancellers for teleconferencing," in *Proc. Int. Conf. Acoust., Speech, Signal Process.*, Dallas, TX, Apr. 1987, vol. 4, pp. 2141–2144.

[23] Y. Ono and H. Kiya, "Performance analysis of subband adaptive systems using an equivalent model," in *Proc. Int. Conf. Acoust., Speech, Signal Process.*, Adelaide, Australia, Apr. 1994, vol. 3, pp. 53–56.

[24] M. Vetterli, "Running FIR and IIR filtering using multirate filter banks," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 36, pp. 730–738, May 1988.

[25] S. Zimmermann and G. A. Williamson, "Subband adaptive filters with zero alias component constraints," in *Proc. Int. Conf. Acoust., Speech, Signal Process.*, Atlanta, GA, May 1996, vol. 3, pp. 1736–1739.

[26] V. S. Somayazulu, S. K. Mitra, and J. J. Shynk, "Adaptive line enhancement using multirate techniques," in *Proc. Int. Conf. Acoust., Speech, Signal Process.*, Glasgow, U.K., May 1989, vol. 2, pp. 928–931.

[27] B. Widrow, J. M. McCool, M. G. Larimore, and C. R. Johnson, Jr., "Stationary and nonstationary learning characteristics of the LMS adaptive filter," *Proc. IEEE*, vol. 64, pp. 1151–1162, Aug. 1976.

[28] Z. J. Mou, P. Duhamel, and J. Benesty, "Fast complex FIR filtering algorithms with applications to real FIR and complex LMS filters," in *Proc. EUSIPCO*, Sept. 1990, pp. 549–552.

[29] E. O. Brigham, *The Fast Fourier Transform*. Englewood Cliffs, NJ: Prentice-Hall, 1974.

[30] A. V. Oppenheim and R. W. Schafer, *Discrete-Time Signal Processing*. Englewood Cliffs, NJ: Prentice-Hall, 1989.

Marc de Courville was born in Paris, France, on April 21, 1969. He graduated from the Ecole Nationale Supérieure des Télécommunications (ENST), Paris, in 1993 and received the Ph.D. degree, also from the ENST, in 1996.

His research interests include multicarrier systems, adaptive algorithms, and multirate filtering. Since 1996, he has been with the Motorola Research Center (CRM), Paris, as a Research Engineer.

Pierre Duhamel (SM'87) was born in France in 1953. He received the Ing. degree in electrical engineering from the National Institute for Applied Sciences (INSA), Rennes, France, in 1975, the Dr.Ing. degree in 1978, and the Doctorat ès sciences degree in 1986, both from Orsay University, Orsay, France.

From 1975 to 1980, he was with Thomson-CSF, Paris, France, where his research interests were in circuit theory and signal processing, including digital filtering and analog fault diagnosis. In 1980, he joined the National Research Center in Telecommunications (CNET), Issy les Moulineaux, France, where his research activities were first concerned with the design of recursive CCD filters. Later, he worked on fast Fourier transforms and convolution algorithms and applied similar techniques to adaptive filtering, spectral analysis, and wavelet transforms. He is now developing studies in channel equalization (including multicarrier systems) and source coding (including joint source/channel coding). Since June 1993, he has been Professor with the Ecole National Supérieure des Télécommunications (ENST), Paris, with research activities focused on signal processing for communications. He was recently nominated Head of the Signal and Image Processing Department.

Dr. Duhamel is chairman of the IEEE Signal Processing Society's DSP Committee, was an Associate Editor of the IEEE TRANSACTIONS ON SIGNAL PROCESSING from 1989 to 1991, and was an Associate Editor for the IEEE SIGNAL PROCESSING LETTERS. He was a Guest Editor of the IEEE TRANSACTIONS ON SIGNAL PROCESSING for the Special Issue on Wavelets.