Efficient Realization of Multichannel Frequency-Domain Adaptive Filtering and Application to Hands-Free Speech Communication

Herbert Buchner, Jacob Benesty, and Walter Kellermann

Abstract

In unknown environments where we need to identify, model, or track unknown and time-varying channels, adaptive filtering has been proven to be an effective tool. In this contribution, we exclusively focus on multichannel algorithms in the frequency domain. For the single-channel case, there are many frequency-domain algorithms in the literature, where most of them are derived from existing time-domain algorithms and different heuristical considerations. Here, we propose a new theory for the rigorous derivation of a whole class of multichannel adaptive filtering algorithms in the frequency domain based on a recursive least-squares criterion. Then, from the so-called normal equation, we derive a generic adaptive algorithm in the frequency domain that we can write in different ways. An analysis of this algorithm shows that the mean-squared error convergence is independent of the input signal statistics. We suggest a very useful approximation, deduce some well-known algorithms, and give design rules for important parameters to optimize the performance in practice. Due to the rigorous approach, the proposed framework inherently takes the coherence between all input signal channels into account, while the computational complexity is kept low by introducing several new techniques, such as a robust recursive Kalman gain computation in the frequency domain and efficient fast Fourier transform (FFT) computation tailored for overlapping data blocks. Simulation results and real-time performance for applications such as multichannel acoustic echo cancellation on regular personal computers show the high efficiency of the approach.

H. Buchner and W. Kellermann are with the University of Erlangen-Nuremberg, Cauerstr. 7, 91058 Erlangen, Germany. E-mail: {buchner, wk}@LNT.de

J. Benesty is with Bell Laboratories, Lucent Technologies, 700 Mountain Avenue, 2D-518, Murray Hill, NJ 07974, USA. E-mail: jbenesty@bell-labs.com

I. Introduction

Adaptive filtering constitutes an important part of statistical signal processing. The ability of adaptive filters to operate satisfactorily in an unknown environment and track time variations of input statistics make it a powerful tool in such diverse fields as communications, acoustics, radar, sonar, seismology, and biomedical engineering. Despite of the large variety of applications, four basic classes of adaptive filtering applications may be distinguished [1]: system identification, inverse modeling, prediction, and interference cancelling.

Especially in speech and acoustics, where all those basic classes of adaptive filtering can be found, we often have to deal with very long filters (sometimes several thousand taps), highly time-variant environments, and highly non-stationary and auto-correlated signals.

In addition, the simultaneous processing of multiple input streams, i.e., multichannel adaptive filtering (MC ADF) is becoming more and more desirable for future products. Typical examples are multichannel acoustic echo cancellation (system identification) or adaptive beamforming microphone arrays (interference cancelling).

In this article, we present a rigorous approach to adaptive MIMO (multiple input and multiple output) systems that are updated in the frequency domain and show its high efficiency for the above mentioned applications. The resulting generalized multichannel frequency-domain adaptive filtering has led to successful real-time implementations of multichannel acoustic echo cancellers on regular personal computers [2], [3].

Generally, we distinguish two classes of adaptive algorithms. One class includes filters that are updated in the time domain, sample-by-sample in general, like the classical least-mean-square (LMS) [4] and recursive least-squares (RLS) [5] algorithms. The other class contains filters that are updated in the frequency domain, block-by-block in general, using the fast Fourier transform (FFT) as a powerful vehicle. As a result of this block processing, the arithmetic complexity of the algorithms that belong to the latter category is significantly reduced compared to time-domain adaptive algorithms. The possibility to exploit the efficiency of FFT algorithms is due to the Toeplitz structure of the matrices involved, which results from the time-shift properties of the filter input signal. Consequently, the key for deriving a frequency-domain adaptive algorithm is to rewrite the time-domain error criterion in a way that Toeplitz and circulant matrices are explicitly shown.

Another advantage of frequency-domain adaptive filtering is that the stepsize can be normalized independently for each frequency bin, which results in a more uniform convergence over the entire frequency range. It is well known [1] that the eigenvalue spread of the input signal correlation matrix determines the convergence speed of gradient-based algorithms such as the LMS algorithm. For speech signals, the eigenvalue spread can be relatively high. The eigenvalues approximately correspond to the power spectral density on equidistant frequency points [6]. Therefore, it is possible to compensate for this power variation by choosing stepsizes that are inversely proportional to the power spectral density in these frequency points.

The case of multichannel adaptive filtering (Fig. 1) has been found to be much more difficult in general, because of the often extremely ill-conditioned correlation matrix to be inverted. In typical scenarios, the input signals to the adaptive filter are not only auto-correlated but also highly cross-correlated which often results in very slow convergence. This problem becomes particularly severe in multichannel acoustic echo cancellation

[7], [8], [9].

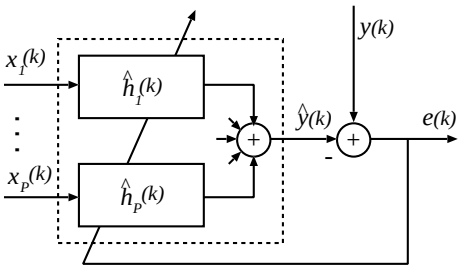


Fig. 1. Multichannel adaptive filtering.

Direct application of commonly used low-complexity algorithms, such as the LMS algorithm or conventional frequency-domain adaptive filtering, to the multichannel case often leads to disappointing results as the cross-correlations between the input channels are not taken into account [10]. In contrast to this, high-order affine projection algorithms and RLS algorithms do take the cross-correlations into account. Indeed, it can be shown that the RLS provides optimum convergence speed even in the multichannel case [10], but the complexity of these algorithms is prohibitively high and, e.g., will not allow real-time implementation of multichannel acoustic echo cancellation on regular PC hardware any time soon.

The rigorous derivation of frequency-domain adaptive filtering presented in the next section leads to a generic algorithm with RLS-like properties. We will also see that there is an efficient approximation of this algorithm taking the cross-correlations into account. The single-channel version of this algorithm provides a direct link to existing frequency-domain algorithms.

Single-channel frequency-domain adaptive filtering was first introduced by Dentino *et al.*, based on the least-mean-squares (LMS) algorithm in the time-domain [11]. Ferrara [13] was the first to elaborate an efficient frequency-domain adaptive filter algorithm (FLMS) that converges to the optimum (Wiener) solution. Mansour and Gray [14] derived an even more efficient algorithm, the *unconstrained* FLMS (UFLMS), using only three FFT operations per block instead of five for the FLMS, with comparable performance [15]. However, in some applications, a major handicap with these structures is the delay introduced between input and output. Indeed, this delay is equal to the length L of the adaptive filter, which is considerable for applications like acoustic echo cancellation. A new structure called *multi-delay filter* (MDF), using the classical overlap-save (OLS) method, was proposed in [16], [17] and generalized in [18] where the block processing N was made independent of the filter length L; N can be chosen as small as desired, with a delay equal to N. Although from a complexity point of view, the optimum choice is $N = L$, using smaller block sizes ($N < L$) in order to reduce the delay is still more efficient than time-domain algorithms. A more general scheme based on weighted overlap and add (WOLA) methods, the *generalized multidelay filter* (GMDFα) was proposed in [19], [20], where α is the overlap factor. The settings $\alpha > 1$ appear to be very useful in the context of adaptive filtering, since the filter coefficients can be adapted more frequently (every N/α samples instead of every N samples in the conventional OLS scheme). Thus, this structure introduces one more degree of freedom, but the complexity is increased roughly by a factor α. Taking the block size in the MDF as large as the delay

permits will increase the convergence rate of the algorithm, while choosing the overlap factor greater than 1 will increase the tracking abilities of the algorithm.

Two-channel frequency-domain adaptive filtering was first introduced in [21] in the context of stereophonic acoustic echo cancellation and derived from the extended least-mean-squares (ELMS) algorithm [22] in the time domain using similar considerations as for the single-channel case outlined above.

In this paper we provide a new and rigorous derivation for a class of multichannel frequency-domain adaptive algorithms based on a recursive least-squares criterion. It will be shown that the theory covers the above algorithms as special cases and leads the way to computationally efficient and fast converging practical approximations. For clarity, we will confine the detailed derivation to a block size $N = L$. A generalization to $N \leq L$ is straightforward; we will point to some interesting results along the lines.

The organization of this contribution is as follows. In Section II, we propose a frequency-domain recursive least-squares criterion from which the so-called normal equation is derived. Then, from this normal equation, we deduce a generic multichannel adaptive algorithm that we can write in different ways and introduce the so-called frequency-domain Kalman gain. In Section III, we study the convergence of this algorithm. In Section IV, we consider the general MIMO case and, in Section V, we give a very useful approximation, deduce some well-known algorithms as special cases, and give design rules for some important parameters such as the exponential window, regularization, and adaptation stepsize. A useful dynamical regularization method is discussed in more detail in Section VI. Section VII introduces several methods for increasing computational efficiency in the multi-input and MIMO cases, such as a robust recursive Kalman gain computation and FFT computation tailored for overlapping data blocks. Section VIII presents some simulations and multichannel real-world implementations for hands-free speech communications. Finally, conclusions are summarized in Section IX.

II. General Derivation of Multichannel Frequency-Domain Algorithms

In the first part of this section we formulate a block recursive least-squares criterion in the frequency domain. Once the criterion is rigorously defined, the adaptive algorithm follows immediately.

A. Optimization Criterion

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

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

with

$$\hat{y}(k) = \sum_{p=1}^{P} \boldsymbol{x}_p^T(k)\hat{\boldsymbol{h}}_p = \boldsymbol{x}^T(k)\hat{\boldsymbol{h}}, \tag{2}$$

where

$$\boldsymbol{x}_p(k) = [x_p(k), x_p(k-1), \cdots, x_p(k-L+1)]^T \tag{3}$$

is a vector containing the latest L samples of the input signal x_p of the p-th channel, and where

$$\hat{\boldsymbol{h}}_p = [\hat{h}_{p,0}, \hat{h}_{p,1}, \cdots, \hat{h}_{p,L-1}]^T \tag{4}$$

contains the current weights of the adaptive FIR filter taps for the p-th input channel. The vectors

$$\boldsymbol{x}(k) = [\boldsymbol{x}_1^T(k), \boldsymbol{x}_2^T(k), \cdots, \boldsymbol{x}_P^T(k)]^T \tag{5}$$

and

$$\hat{\boldsymbol{h}} = [\hat{\boldsymbol{h}}_1^T, \hat{\boldsymbol{h}}_2^T, \cdots, \hat{\boldsymbol{h}}_P^T]^T \tag{6}$$

allow a convenient notation of the multichannel algorithms. Superscript ^{T} denotes transposition of a vector or a matrix.

We now define the block error signal of length L. From (1) and (2) follows

$$\boldsymbol{e}(m) = \boldsymbol{y}(m) - \hat{\boldsymbol{y}}(m), \tag{7}$$

with m being the block time index, and

$$\hat{\boldsymbol{y}}(m) = \sum_{p=1}^{P} \boldsymbol{U}_p^T(m)\hat{\boldsymbol{h}}_p = \boldsymbol{U}^T(m)\hat{\boldsymbol{h}}, \tag{8}$$

where

$$\boldsymbol{e}(m) = [e(mL), \cdots, e(mL+L-1)]^T, \tag{9}$$
$$\boldsymbol{y}(m) = [y(mL), \cdots, y(mL+L-1)]^T, \tag{10}$$
$$\hat{\boldsymbol{y}}(m) = [\hat{y}(mL), \cdots, \hat{y}(mL+L-1)]^T, \tag{11}$$
$$\boldsymbol{U}_p(m) = [\boldsymbol{x}_p(mL), \cdots, \boldsymbol{x}_p(mL+L-1)], \tag{12}$$
$$\boldsymbol{U}(m) = [\boldsymbol{U}_1^T(m), \cdots, \boldsymbol{U}_P^T(m)]^T. \tag{13}$$

It can easily be verified that $\boldsymbol{U}_p$, $p = 1, \ldots, P$ are Toeplitz matrices of size $(L \times L)$:

$$\boldsymbol{U}_p^T(m) = \begin{bmatrix} x_p(mL) & \cdots & x_p(mL-L+1) \\ x_p(mL+1) & \ddots & \vdots \\ \vdots & \ddots & \vdots \\ x_p(mL+L-1) & \cdots & x_p(mL) \end{bmatrix}$$

These Toeplitz matrices are now diagonalized in two steps:

Step 1: Transformation of Toeplitz matrices into circulant matrices.

Any Toeplitz matrix $\boldsymbol{U}_p$ can be transformed, by doubling its size, to a circulant matrix

$$\boldsymbol{C}_p(m) = \begin{bmatrix} \boldsymbol{U}_p'^T(m) & \boldsymbol{U}_p^T(m) \\ \boldsymbol{U}_p^T(m) & \boldsymbol{U}_p'^T(m) \end{bmatrix}, \tag{14}$$

where the $\boldsymbol{U}_p'$ are also Toeplitz matrices and can be expressed in terms of the elements of $\boldsymbol{U}_p^T(m)$, except for an arbitrary diagonal, e.g.,

$$\boldsymbol{U}_p'^T(m) = \begin{bmatrix} x_p(mL+L) & \cdots & x_p(mL+1) \\ x_p(mL-L+1) & \ddots & \vdots \\ \vdots & \ddots & \vdots \\ x_p(mL-1) & \cdots & x_p(mL+L) \end{bmatrix}.$$

It follows

$$\boldsymbol{U}_p^T(m) = \boldsymbol{W}_{L\times 2L}^{01}\boldsymbol{C}_p(m)\boldsymbol{W}_{2L\times L}^{10}, \tag{15}$$

where we introduced the windowing matrices

$$\begin{aligned}\boldsymbol{W}_{L\times 2L}^{01} &= [\boldsymbol{0}_{L\times L}, \boldsymbol{I}_{L\times L}],\\ \boldsymbol{W}_{2L\times L}^{10} &= [\boldsymbol{I}_{L\times L}, \boldsymbol{0}_{L\times L}]^T.\end{aligned}$$

Step 2: Transformation of the circulant matrices into diagonal matrices.

Using the $2L \times 2L$ DFT matrix $\boldsymbol{F}_{2L\times 2L}$, the circulant matrices are diagonalized as follows:

$$\boldsymbol{C}_p(m) = \boldsymbol{F}_{2L\times 2L}^{-1}\boldsymbol{X}_p(m)\boldsymbol{F}_{2L\times 2L}, \tag{16}$$

where the diagonal matrices $\boldsymbol{X}_p(m)$ can be expressed by the first columns of $\boldsymbol{C}_p(m)$,

$$\boldsymbol{X}_p(m) = \mathrm{diag}\{\boldsymbol{F}_{2L\times 2L}[x_p(mL-L+1),\cdots,x_p(mL+L)]^T\}. \tag{17}$$

Now, (15) can be rewritten equivalently as

$$\boldsymbol{U}_p^T(m) = \boldsymbol{W}_{L\times 2L}^{01}\boldsymbol{F}_{2L\times 2L}^{-1}\boldsymbol{X}_p(m)\boldsymbol{F}_{2L\times 2L}\boldsymbol{W}_{2L\times L}^{10}. \tag{18}$$

Since

$$[\boldsymbol{A}\boldsymbol{X}_1\boldsymbol{B},\cdots,\boldsymbol{A}\boldsymbol{X}_P\boldsymbol{B}] = \boldsymbol{A}[\boldsymbol{X}_1,\cdots,\boldsymbol{X}_P]\mathrm{diag}\{\boldsymbol{B},\cdots,\boldsymbol{B}\}$$

for any matrices $\boldsymbol{A}$,$\boldsymbol{B}$,$\boldsymbol{X}_p$ with compatible dimensions, it follows for the error vector using (13) and (18):

$$\boldsymbol{e}(m) = \boldsymbol{y}(m) - \boldsymbol{W}_{L\times 2L}^{01}\boldsymbol{F}_{2L\times 2L}^{-1}[\boldsymbol{X}_1(m),\cdots,\boldsymbol{X}_P(m)]\mathrm{diag}\{\boldsymbol{F}_{2L\times 2L}\boldsymbol{W}_{2L\times L}^{10},\cdots,\boldsymbol{F}_{2L\times 2L}\boldsymbol{W}_{2L\times L}^{10}\}\hat{\boldsymbol{h}}. \tag{19}$$

If we multiply (19) by the DFT matrix $\boldsymbol{F}_{L\times L}$ of size $L \times L$, we get the error signal in the frequency domain:

$$\underline{\boldsymbol{e}}(m) = \underline{\boldsymbol{y}}(m) - \boldsymbol{G}_{L\times 2L}^{01}\boldsymbol{X}(m)\boldsymbol{G}_{2LP\times LP}^{10}\underline{\hat{\boldsymbol{h}}}, \tag{20}$$

where

$$\begin{aligned}\underline{\boldsymbol{e}}(m) &= \boldsymbol{F}_{L\times L}\boldsymbol{e}(m), &(21)\\ \underline{\boldsymbol{y}}(m) &= \boldsymbol{F}_{L\times L}\boldsymbol{y}(m), &(22)\\ \boldsymbol{G}_{L\times 2L}^{01} &= \boldsymbol{F}_{L\times L}\boldsymbol{W}_{L\times 2L}^{01}\boldsymbol{F}_{2L\times 2L}^{-1}, &(23)\\ \boldsymbol{G}_{2LP\times LP}^{10} &= \mathrm{diag}\{\boldsymbol{G}_{2L\times L}^{10},\cdots,\boldsymbol{G}_{2L\times L}^{10}\}, &(24)\\ \boldsymbol{G}_{2L\times L}^{10} &= \boldsymbol{F}_{2L\times 2L}\boldsymbol{W}_{2L\times L}^{10}\boldsymbol{F}_{L\times L}^{-1}, &(25)\\ \boldsymbol{X}(m) &= [\boldsymbol{X}_1(m),\boldsymbol{X}_2(m),\cdots,\boldsymbol{X}_P(m)], &(26)\\ \underline{\hat{\boldsymbol{h}}}_p &= \boldsymbol{F}_{L\times L}\hat{\boldsymbol{h}}_p, &(27)\\ \underline{\hat{\boldsymbol{h}}} &= [\underline{\hat{\boldsymbol{h}}}_1^T,\underline{\hat{\boldsymbol{h}}}_2^T,\cdots,\underline{\hat{\boldsymbol{h}}}_P^T]^T. &(28)\end{aligned}$$

Optimization Criterion:

Having derived a frequency-domain error signal, we now define a frequency-domain criterion for optimizing the coefficient vector $\underline{\hat{\boldsymbol{h}}} = \underline{\hat{\boldsymbol{h}}}(m)$:

$$J_{\mathrm{f}}(m) = (1-\lambda)\sum_{i=0}^{m}\lambda^{m-i}\underline{\boldsymbol{e}}^H(i)\underline{\boldsymbol{e}}(i), \tag{29}$$

where ^{H} denotes conjugate transpose and λ ($0 < \lambda < 1$) is an exponential forgetting factor. The criterion (29) is very similar to the one leading to the well-known RLS algorithm [5]. The main advantage of using (29) is to take advantage of the FFT in order to have low-complexity adaptive filters.

B. Normal Equation

Let $\nabla_{\underline{\hat{\boldsymbol{h}}}}$ be the gradient operator with respect to $\underline{\hat{\boldsymbol{h}}}$. Applying the operator $\nabla_{\underline{\hat{\boldsymbol{h}}}}$ to the cost function J_{f}, we obtain [1], [23] the complex gradient vector:

$$\begin{aligned}\nabla_{\underline{\hat{\boldsymbol{h}}}} J_{\mathrm{f}}(m) = \frac{\partial J_{\mathrm{f}}(m)}{\partial \underline{\hat{\boldsymbol{h}}}(m)} &= -(1-\lambda)\sum_{i=0}^{m}\lambda^{m-i}(\boldsymbol{G}_{2LP\times LP}^{10})^T\boldsymbol{X}^T(i)(\boldsymbol{G}_{L\times 2L}^{01})^T\underline{\boldsymbol{y}}^*(i) \\ &\quad + (1-\lambda)\left[\sum_{i=0}^{m}\lambda^{m-i}(\boldsymbol{G}_{2LP\times LP}^{10})^T\boldsymbol{X}^T(i)(\boldsymbol{G}_{2L\times 2L}^{01})^T\boldsymbol{X}^*(i)(\boldsymbol{G}_{2LP\times LP}^{10})^*\right]\underline{\hat{\boldsymbol{h}}}^*(m),\end{aligned} \tag{30}$$

where * denotes complex conjugate,

$$\begin{aligned}\boldsymbol{G}_{2L\times 2L}^{01} &= (\boldsymbol{G}_{L\times 2L}^{01})^H\boldsymbol{G}_{L\times 2L}^{01} \\ &= \boldsymbol{F}_{2L\times 2L}\boldsymbol{W}_{2L\times 2L}^{01}\boldsymbol{F}_{2L\times 2L}^{-1},\end{aligned} \tag{31}$$

and

$$\boldsymbol{W}_{2L\times 2L}^{01} = \begin{bmatrix}\boldsymbol{0}_{L\times L} & \boldsymbol{0}_{L\times L}\\ \boldsymbol{0}_{L\times L} & \boldsymbol{I}_{L\times L}\end{bmatrix}. \tag{32}$$

By setting the gradient of the cost function equal to zero, conjugating, noting that $(\boldsymbol{G}_{2L\times 2L}^{01})^H = \boldsymbol{G}_{2L\times 2L}^{01}$ and defining

$$\begin{aligned}\underline{\boldsymbol{y}}_{2L}(m) &= (\boldsymbol{G}_{L\times 2L}^{01})^H\underline{\boldsymbol{y}}(m) \\ &= \boldsymbol{F}_{2L\times 2L}\begin{bmatrix}\boldsymbol{0}_{L\times 1}\\ \boldsymbol{y}(m)\end{bmatrix},\end{aligned} \tag{33}$$

we obtain the so-called *normal* equation:

$$\boldsymbol{S}(m)\underline{\hat{\boldsymbol{h}}}(m) = \boldsymbol{s}(m), \tag{34}$$

where

$$\begin{aligned}\boldsymbol{S}(m) &= (1-\lambda)\sum_{i=0}^{m}\lambda^{m-i}(\boldsymbol{G}_{2LP\times LP}^{10})^H\boldsymbol{X}^H(i)\boldsymbol{G}_{2L\times 2L}^{01}\boldsymbol{X}(i)\boldsymbol{G}_{2LP\times LP}^{10} \\ &= \lambda\boldsymbol{S}(m-1) + (1-\lambda)(\boldsymbol{G}_{2LP\times LP}^{10})^H\boldsymbol{X}^H(m)\boldsymbol{G}_{2L\times 2L}^{01}\boldsymbol{X}(m)\boldsymbol{G}_{2LP\times LP}^{10}\end{aligned} \tag{35}$$

and

$$\begin{aligned}\boldsymbol{s}(m) &= (1-\lambda)\sum_{i=0}^{m}\lambda^{m-i}(\boldsymbol{G}_{2LP\times LP}^{10})^H\boldsymbol{X}^H(i)\underline{\boldsymbol{y}}_{2L}(i) \\ &= \lambda\boldsymbol{s}(m-1) + (1-\lambda)(\boldsymbol{G}_{2LP\times LP}^{10})^H\boldsymbol{X}^H(m)\underline{\boldsymbol{y}}_{2L}(m) \\ &= \lambda\boldsymbol{s}(m-1) + (1-\lambda)(\boldsymbol{G}_{2LP\times LP}^{10})^H\boldsymbol{X}^H(m)(\boldsymbol{G}_{L\times 2L}^{01})^H\underline{\boldsymbol{y}}(m).\end{aligned} \tag{36}$$

If the input signal is well-conditioned, matrix $\boldsymbol{S}(m)$ is nonsingular. In this case, the normal equation has a unique solution which is the optimum Wiener solution.

C. Adaptive Algorithm

There are many different ways to write the adaptive algorithm, i.e., a recursive update of $\underline{\hat{\boldsymbol{h}}}(m)$. In any case, it is derived directly from the normal equation (34) and associated equations (35) and (36).

In the recursive equation (36), we replace $\boldsymbol{s}(m)$ and $\boldsymbol{s}(m-1)$ by formulating (34) in terms of block time indices m and $m-1$, respectively. We then eliminate $\boldsymbol{S}(m-1)$ from the resulting equation using (35). Reintroducing the error signal vector (20), we obtain the following exact adaptive algorithm:

$$\begin{aligned}
\underline{\boldsymbol{e}}(m) &= \underline{\boldsymbol{y}}(m) - \boldsymbol{G}^{01}_{L\times 2L}\boldsymbol{X}(m)\boldsymbol{G}^{10}_{2LP\times LP}\underline{\hat{\boldsymbol{h}}}(m-1) && (37)\\
\underline{\hat{\boldsymbol{h}}}(m) &= \underline{\hat{\boldsymbol{h}}}(m-1) + (1-\lambda)\boldsymbol{S}^{-1}(m)(\boldsymbol{G}^{10}_{2LP\times LP})^H\boldsymbol{X}^H(m)(\boldsymbol{G}^{01}_{L\times 2L})^H\underline{\boldsymbol{e}}(m), && (38)
\end{aligned}$$

For practical purposes, it is useful to reformulate this algorithm equivalently. First, we multiply (37) by $(\boldsymbol{G}^{01}_{L\times 2L})^H$,

$$\begin{aligned}
\underline{\boldsymbol{e}}_{2L}(m) &= \underline{\boldsymbol{y}}_{2L}(m) - \boldsymbol{G}^{01}_{2L\times 2L}\boldsymbol{X}(m)\boldsymbol{G}^{10}_{2LP\times LP}\underline{\hat{\boldsymbol{h}}}(m-1) && (39)\\
\underline{\hat{\boldsymbol{h}}}(m) &= \underline{\hat{\boldsymbol{h}}}(m-1) + (1-\lambda)\boldsymbol{S}^{-1}(m)(\boldsymbol{G}^{10}_{2LP\times LP})^H\boldsymbol{X}^H(m)\underline{\boldsymbol{e}}_{2L}(m), && (40)
\end{aligned}$$

where we defined analogously to (33)

$$\begin{aligned}
\underline{\boldsymbol{e}}_{2L}(m) &= (\boldsymbol{G}^{01}_{L\times 2L})^H\underline{\boldsymbol{e}}(m)\\
&= \boldsymbol{F}_{2L\times 2L}\begin{bmatrix}\boldsymbol{0}_{L\times 1}\\ \boldsymbol{e}(m)\end{bmatrix}. && (41)
\end{aligned}$$

If we multiply (40) by $\boldsymbol{G}^{10}_{2LP\times LP}$, we obtain the algorithm:

$$\begin{aligned}
\boldsymbol{S}(m) &= \lambda\boldsymbol{S}(m-1) + (1-\lambda)(\boldsymbol{G}^{10}_{2LP\times LP})^H\boldsymbol{X}^H(m)\boldsymbol{G}^{01}_{2L\times 2L}\boldsymbol{X}(m)\boldsymbol{G}^{10}_{2LP\times LP} && (42)\\
\underline{\boldsymbol{e}}_{2L}(m) &= \underline{\boldsymbol{y}}_{2L}(m) - \boldsymbol{G}^{01}_{2L\times 2L}\boldsymbol{X}(m)\underline{\hat{\boldsymbol{h}}}_{2LP}(m-1) && (43)\\
\underline{\hat{\boldsymbol{h}}}_{2LP}(m) &= \underline{\hat{\boldsymbol{h}}}_{2LP}(m-1) + (1-\lambda)\boldsymbol{G}^{10}_{2LP\times LP}\boldsymbol{S}^{-1}(m)(\boldsymbol{G}^{10}_{2LP\times LP})^H\boldsymbol{X}^H(m)\underline{\boldsymbol{e}}_{2L}(m), && (44)
\end{aligned}$$

where

$$\begin{aligned}
\underline{\hat{\boldsymbol{h}}}_{2LP}(m) &= \boldsymbol{G}^{10}_{2LP\times LP}\underline{\hat{\boldsymbol{h}}}(m)\\
&= \left[\underline{\hat{\boldsymbol{h}}}^T_{2LP,1}(m),\cdots,\underline{\hat{\boldsymbol{h}}}^T_{2LP,P}(m)\right]^T,\\
\underline{\hat{\boldsymbol{h}}}_{2LP,p}(m) &= \boldsymbol{F}_{2L\times 2L}\begin{bmatrix}\hat{\boldsymbol{h}}_p(m)\\ \boldsymbol{0}_{L\times 1}\end{bmatrix}. && (45)
\end{aligned}$$

The rank of the matrix $\boldsymbol{G}^{10}_{2LP\times LP}$ is equal to LP. Since we have to adapt LP unknowns, in principle (44) is equivalent to (40). Indeed, if we multiply (44) by $(\boldsymbol{G}^{10}_{2LP\times LP})^H$, we obtain exactly (40) since $(\boldsymbol{G}^{10}_{2LP\times LP})^H\boldsymbol{G}^{10}_{2LP\times LP} = \boldsymbol{I}_{LP\times LP}$. It is interesting to see how naturally we have ended up using blocks of length $2L$ (especially for the error signal) even though we have used an error criterion with blocks of length L. We can do even better than that and rewrite the algorithm exclusively using FFTs of size $2L$. This formulation

is by far the most interesting one because an explicit link with existing frequency-domain algorithms can be established. Let us first define the $(2LP \times 2LP)$ matrix

$$\begin{aligned}\boldsymbol{S}_{\mathrm{d}}(m) &= (1-\lambda)\sum_{i=0}^{m}\lambda^{m-i}\boldsymbol{X}^H(i)\boldsymbol{G}_{2L\times 2L}^{01}\boldsymbol{X}(i) \\ &= \lambda\boldsymbol{S}_{\mathrm{d}}(m-1)+(1-\lambda)\boldsymbol{X}^H(m)\boldsymbol{G}_{2L\times 2L}^{01}\boldsymbol{X}(m). \end{aligned} \tag{46}$$

The relationship with $\boldsymbol{S}(m)$ is obviously given by:

$$\boldsymbol{S}(m) = (\boldsymbol{G}_{2LP\times LP}^{10})^H\boldsymbol{S}_{\mathrm{d}}(m)\boldsymbol{G}_{2LP\times LP}^{10}. \tag{47}$$

Next, we define

$$\begin{aligned}\boldsymbol{G}_{2L\times 2L}^{10} &= \boldsymbol{G}_{2L\times L}^{10}(\boldsymbol{G}_{2L\times L}^{10})^H \\ &= \boldsymbol{F}_{2L\times 2L}\boldsymbol{W}_{2L\times 2L}^{10}\boldsymbol{F}_{2L\times 2L}^{-1}\end{aligned}$$

and

$$\boldsymbol{G}_{2LP\times 2LP}^{10} = \mathrm{diag}\{\boldsymbol{G}_{2L\times 2L}^{10}\cdots\boldsymbol{G}_{2L\times 2L}^{10}\}, \tag{48}$$

where

$$\boldsymbol{W}_{2L\times 2L}^{10} = \begin{bmatrix}\boldsymbol{I}_{L\times L} & \boldsymbol{0}_{L\times L}\\ \boldsymbol{0}_{L\times L} & \boldsymbol{0}_{L\times L}\end{bmatrix}. \tag{49}$$

Now, we have an interesting relation between the inverse of the two matrices $\boldsymbol{S}$ and $\boldsymbol{S}_{\mathrm{d}}$:

$$\boldsymbol{G}_{2LP\times 2LP}^{10}\boldsymbol{S}_{\mathrm{d}}^{-1}(m) = \boldsymbol{G}_{2LP\times LP}^{10}\boldsymbol{S}^{-1}(m)(\boldsymbol{G}_{2LP\times LP}^{10})^H. \tag{50}$$

This can be verified by post-multiplying both sides of (50) by $\boldsymbol{S}_{\mathrm{d}}(m)\boldsymbol{G}_{2LP\times LP}^{10}$ and noting that $\boldsymbol{G}_{2LP\times 2LP}^{10}\boldsymbol{G}_{2LP\times LP}^{10} = \boldsymbol{G}_{2LP\times LP}^{10}$.

Using (50), the adaptive algorithm (42)-(44) can now be written more conveniently:

$$\boldsymbol{S}_{\mathrm{d}}(m) = \lambda\boldsymbol{S}_{\mathrm{d}}(m-1)+(1-\lambda)\boldsymbol{X}^H(m)\boldsymbol{G}_{2L\times 2L}^{01}\boldsymbol{X}(m) \tag{51}$$

$$\underline{\boldsymbol{e}}_{2L}(m) = \underline{\boldsymbol{y}}_{2L}(m)-\boldsymbol{G}_{2L\times 2L}^{01}\boldsymbol{X}(m)\underline{\hat{\boldsymbol{h}}}_{2LP}(m-1) \tag{52}$$

$$\underline{\hat{\boldsymbol{h}}}_{2LP}(m) = \underline{\hat{\boldsymbol{h}}}_{2LP}(m-1)+(1-\lambda)\boldsymbol{G}_{2LP\times 2LP}^{10}\boldsymbol{S}_{\mathrm{d}}^{-1}(m)\boldsymbol{X}^H(m)\underline{\boldsymbol{e}}_{2L}(m). \tag{53}$$

Due to the structure of the update equations, we introduce a frequency-domain Kalman gain matrix in analogy to the RLS algorithm [1]:

$$\boldsymbol{K}(m) = (1-\lambda)\boldsymbol{S}_{\mathrm{d}}^{-1}(m)\boldsymbol{X}^H(m). \tag{54}$$

This $2LP \times 2L$ matrix includes the inverse in (53) and plays an important role in practical aspects as shown later. Figure 2 summarizes the general steps in multichannel frequency-domain adaptive filtering.

A note concerning block sizes $N < L$:

Analogously to (51)-(53), an algorithm can be derived straightforwardly using $K = L/N$ sub-filters per channel and block convolution. Using DFTs of length N and length $2N$, respectively, the error criterion (29)

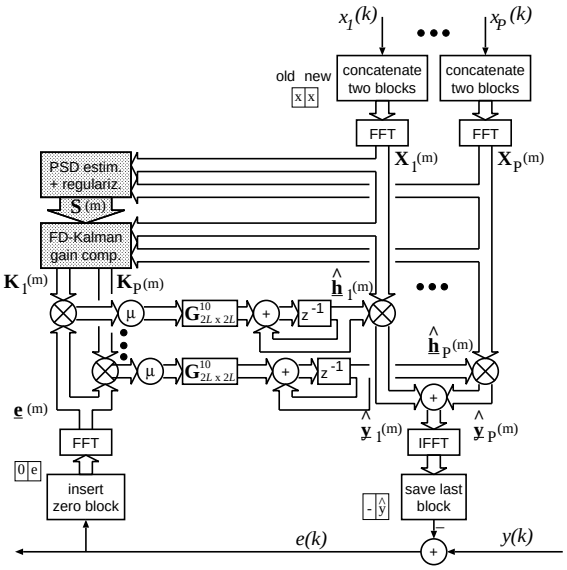


Fig. 2. Multichannel frequency-domain adaptive filtering.

is then applied to length-N error vectors. In the resulting algorithm, the matrices $\boldsymbol{G}^{01}_{2L\times 2L}$ in (51) and (52) are replaced by the corresponding $2N \times 2N$-matrices $\boldsymbol{G}^{01}_{2N\times 2N}$. Equation (53) remains in the same form, where $\boldsymbol{G}^{10}_{2LP\times 2LP}$, defined in (48), is then composed from $K \cdot P$ sub-matrices of size $2N \times 2N$; $\hat{\underline{\boldsymbol{h}}}_{2LP}(m)$ and $\boldsymbol{X}(m)$ are the concatenations of the sub-filters and diagonal sub-matrices, respectively.

III. Convergence Analysis

In this section, we analyze the convergence behaviour of the algorithm in a stationary environment using (37) and (38).

Due to the assumed stationarity of the filter input signals, we obtain, after taking the expected value of (35):

$$E\{\boldsymbol{S}(m)\} = (1-\lambda)\sum_{i=0}^{m}\lambda^{m-i}\boldsymbol{S}_E, \tag{55}$$

where

$$\boldsymbol{S}_E = E\left\{(\boldsymbol{G}^{10}_{2LP\times LP})^H \boldsymbol{X}^H(m)\boldsymbol{G}^{01}_{2L\times 2L}\boldsymbol{X}(m)\boldsymbol{G}^{10}_{2LP\times LP}\right\} \tag{56}$$

denotes the time-independent ensemble average. Noting that in (55) we have a sum of a finite geometric series, it can be simplified to

$$E\{\boldsymbol{S}(m)\} = (1-\lambda^{m+1})\boldsymbol{S}_E. \tag{57}$$

For a single realization, we assume that

$$\boldsymbol{S}(m) \approx (1-\lambda^{m+1})\boldsymbol{S}_E, \tag{58}$$

and for the steady state we see that

$$\boldsymbol{S}(m) \approx \boldsymbol{S}_E \quad \text{for large } m. \tag{59}$$

A. Analysis Model

To proceed with the analysis, we assume that the desired response $y(k)$ and the tap-input vector $x(k)$ are related by the multiple linear regression model [1]

$$y(k) = \boldsymbol{x}^T(k)\boldsymbol{h} + n(k), \tag{60}$$

where the $LP \times 1$ vector $\boldsymbol{h}$ denotes the fixed regression parameter vector of the model and the measurement error $n(k)$ is assumed to be a zero-mean white noise that is independent of $x(k)$. The equivalent expression in the frequency domain reads

$$\underline{\boldsymbol{y}}(m) = \boldsymbol{G}^{01}_{L\times 2L}\boldsymbol{X}(m)\boldsymbol{G}^{10}_{2LP\times LP}\underline{\boldsymbol{h}} + \underline{\boldsymbol{n}}(m), \tag{61}$$

where $\underline{\boldsymbol{h}}$ and $\underline{\boldsymbol{n}}(m)$ are defined in the same way as $\hat{\underline{\boldsymbol{h}}}$ in (28) and $\underline{\boldsymbol{y}}(m)$ in (22), respectively.

B. Convergence in the Mean

By noting that

$$(\boldsymbol{G}^{01}_{L\times 2L})^H\boldsymbol{G}^{01}_{L\times 2L} = \boldsymbol{G}^{01}_{2L\times 2L} \tag{62}$$

from (31), the coefficient update (38) can be written as

$$\begin{aligned}\underline{\boldsymbol{h}} - \hat{\underline{\boldsymbol{h}}}(m) &= \underline{\boldsymbol{h}} - \hat{\underline{\boldsymbol{h}}}(m-1) - (1-\lambda)\boldsymbol{S}^{-1}(m)(\boldsymbol{G}^{10}_{2LP\times LP})^H\boldsymbol{X}^H(m)\boldsymbol{G}^{01}_{2L\times 2L}\boldsymbol{X}(m)\\ &\quad\cdot\boldsymbol{G}^{10}_{2LP\times LP}[\underline{\boldsymbol{h}} - \hat{\underline{\boldsymbol{h}}}(m-1)] - (1-\lambda)\boldsymbol{S}^{-1}(m)(\boldsymbol{G}^{10}_{2LP\times LP})^H\boldsymbol{X}^H(m)\underline{\boldsymbol{n}}(m).\end{aligned} \tag{63}$$

$\underline{\boldsymbol{\epsilon}}(m) = \underline{\boldsymbol{h}} - \hat{\underline{\boldsymbol{h}}}(m)$ is the misalignment vector. Taking mathematical expectation of expression (63), using the independence theory [1], and (56) together with (59), we deduce for large m that

$$\begin{aligned}E\{\underline{\boldsymbol{\epsilon}}(m)\} &= \lambda E\{\underline{\boldsymbol{\epsilon}}(m-1)\}\\ &= \lambda^m E\{\underline{\boldsymbol{\epsilon}}(0)\}.\end{aligned} \tag{64}$$

Equation (64) says that the convergence rate of the algorithm is governed by λ. Most importantly, the rate of convergence is completely independent of the input statistics. Finally, we have

$$\lim_{m\to\infty} E\{\underline{\boldsymbol{\epsilon}}(m)\} = \boldsymbol{0}_{LP\times 1} \Rightarrow \lim_{m\to\infty} E\{\hat{\boldsymbol{h}}(m)\} = \boldsymbol{h}. \tag{65}$$

Now, suppose that λ_{t} is the forgetting factor of a sample-by-sample adaptive algorithm (operating in the time domain). To have the same effective window length for the sample-by-sample and block-by-block algorithms, we should choose $\lambda = \lambda_{\mathrm{t}}^L$. For example, the usual choice for the RLS algorithm is $\lambda_{\mathrm{t}} = 1 - 1/(3L)$. In this case, a good choice for the frequency-domain algorithm is $\lambda = [1 - 1/(3L)]^L$.

C. Convergence of the Mean-Squared Error

The convergence of the algorithm in the mean is not sufficient for mean-squared error (MSE) convergence [1] as it only assures a bias-free estimate $\hat{\underline{\boldsymbol{h}}}(m)$.

The algorithm converges in the mean square if

$$\lim_{m\to\infty} J'_{\mathrm{f}}(m) = J'_{\mathrm{f,min}} < \infty, \tag{66}$$

where

$$J'_{\mathrm{f}}(m) = \frac{1}{L} E\left\{\underline{\boldsymbol{e}}^H(m)\underline{\boldsymbol{e}}(m)\right\}. \tag{67}$$

From (37), the error signal $\underline{\boldsymbol{e}}(m)$ can be written in terms of $\underline{\boldsymbol{\epsilon}}(m)$ as

$$\underline{\boldsymbol{e}}(m) = \boldsymbol{G}^{01}_{L\times 2L}\boldsymbol{X}(m)\boldsymbol{G}^{10}_{2LP\times LP}\underline{\boldsymbol{\epsilon}}(m-1) + \underline{\boldsymbol{n}}(m). \tag{68}$$

Expression (67) becomes

$$J'_{\mathrm{f}}(m) = \frac{1}{L} J_{\mathrm{ex}}(m) + \sigma_n^2, \tag{69}$$

where

$$J_{\mathrm{ex}}(m) = E\left\{\underline{\boldsymbol{\epsilon}}^H(m-1)(\boldsymbol{G}^{10}_{2LP\times LP})^H\boldsymbol{X}^H(m)\boldsymbol{G}^{01}_{2L\times 2L}\boldsymbol{X}(m)\boldsymbol{G}^{10}_{2LP\times LP}\underline{\boldsymbol{\epsilon}}(m-1)\right\} \tag{70}$$

is the excess mean-square error and σ_n^2 is the variance of the noise signal $n(k)$. Furthermore

$$\begin{aligned} J_{\mathrm{ex}}(m) &= E\left\{\mathrm{tr}\left[\underline{\boldsymbol{\epsilon}}^H(m-1)(\boldsymbol{G}^{10}_{2LP\times LP})^H\boldsymbol{X}^H(m)\boldsymbol{G}^{01}_{2L\times 2L}\boldsymbol{X}(m)\boldsymbol{G}^{10}_{2LP\times LP}\underline{\boldsymbol{\epsilon}}(m-1)\right]\right\} \\ &= E\left\{\mathrm{tr}\left[(\boldsymbol{G}^{10}_{2LP\times LP})^H\boldsymbol{X}^H(m)\boldsymbol{G}^{01}_{2L\times 2L}\boldsymbol{X}(m)\boldsymbol{G}^{10}_{2LP\times LP}\underline{\boldsymbol{\epsilon}}(m-1)\underline{\boldsymbol{\epsilon}}^H(m-1)\right]\right\} \\ &= \mathrm{tr}\left[E\left\{(\boldsymbol{G}^{10}_{2LP\times LP})^H\boldsymbol{X}^H(m)\boldsymbol{G}^{01}_{2L\times 2L}\boldsymbol{X}(m)\boldsymbol{G}^{10}_{2LP\times LP}\underline{\boldsymbol{\epsilon}}(m-1)\underline{\boldsymbol{\epsilon}}^H(m-1)\right\}\right]. \end{aligned}$$

Invoking the independence assumption and using (56), we may reduce this expectation to

$$J_{\mathrm{ex}}(m) \approx \mathrm{tr}[\boldsymbol{S}_E\boldsymbol{M}(m-1)], \tag{71}$$

where

$$\boldsymbol{M}(m) = E\left\{\underline{\boldsymbol{\epsilon}}(m)\underline{\boldsymbol{\epsilon}}^H(m)\right\} \tag{72}$$

is the misalignment correlation matrix.

We derive an expression for the misalignment vector $\underline{\boldsymbol{\epsilon}}(m)$ using the normal equation (34), and (36):

$$\begin{aligned} \underline{\boldsymbol{\epsilon}}(m) &= \underline{\boldsymbol{h}} - \hat{\underline{\boldsymbol{h}}}(m) \\ &= \underline{\boldsymbol{h}} - \boldsymbol{S}^{-1}(m)\boldsymbol{s}(m) \\ &= \underline{\boldsymbol{h}} - (1-\lambda)\boldsymbol{S}^{-1}(m)\sum_{i=0}^{m}\lambda^{m-i}(\boldsymbol{G}^{10}_{2LP\times LP})^H\boldsymbol{X}^H(i)(\boldsymbol{G}^{01}_{L\times 2L})^H\underline{\boldsymbol{y}}(i). \end{aligned} \tag{73}$$

Using $\underline{\boldsymbol{y}}(m)$ from the model (61), we obtain with (62) and (35):

$$\underline{\boldsymbol{\epsilon}}(m) = -(1-\lambda)\boldsymbol{S}^{-1}(m)\sum_{i=0}^{m}\lambda^{m-i}(\boldsymbol{G}^{10}_{2LP\times LP})^H\boldsymbol{X}^H(i)(\boldsymbol{G}^{01}_{2L\times 2L})^H\underline{\boldsymbol{n}}(i). \tag{74}$$

If we plug this equation into (72), we obtain, after taking the expectations, and noting that for a given input sequence, the only random variable is the white measurement noise $\underline{\boldsymbol{n}}(m)$:

$$\boldsymbol{M}(m) = \sigma_n^2(1-\lambda)^2\boldsymbol{S}^{-1}(m)\left[\sum_{i=0}^{m}\lambda^{2(m-i)}(\boldsymbol{G}^{10}_{2LP\times LP})^H\boldsymbol{X}^H(i)\boldsymbol{G}^{01}_{2L\times 2L}\boldsymbol{X}(i)\boldsymbol{G}^{10}_{2LP\times LP}\right]\boldsymbol{S}^{-1}(m), \tag{75}$$

where $E\{\underline{\boldsymbol{n}}(m)\underline{\boldsymbol{n}}^H(m)\} = \sigma_n^2\boldsymbol{I}$. Analogously to (58), we find for the term in brackets in (75):

$$\sum_{i=0}^{m}\lambda^{2(m-i)}(\boldsymbol{G}_{2LP\times LP}^{10})^H\boldsymbol{X}^H(i)\boldsymbol{G}_{2L\times 2L}^{01}\boldsymbol{X}(i)\boldsymbol{G}_{2LP\times LP}^{10} \approx (1-\lambda^{2(m+1)})\boldsymbol{S}_E. \tag{76}$$

Using (58), (76), and $1-\lambda^{2(m+1)} = (1-\lambda^{m+1})(1+\lambda^{m+1})$, this leads to

$$\boldsymbol{M}(m) = \sigma_n^2(1-\lambda)^2\frac{1+\lambda^{m+1}}{1-\lambda^{m+1}}\boldsymbol{S}_E^{-1}. \tag{77}$$

Finally, we obtain for (69) with (71)

$$J'_{\mathrm{f}}(m) = \left[(1-\lambda)^2\frac{1+\lambda^m}{1-\lambda^m}+1\right]\sigma_n^2. \tag{78}$$

This equation describes the convergence curve of the mean-squared error. One can see that in the steady state, i.e., for large m, the mean-squared error converges to a constant value as desired in (66):

$$J'_{\mathrm{f}}(m\to\infty) = J'_{\mathrm{f,min}} = \left[(1-\lambda)^2+1\right]\sigma_n^2. \tag{79}$$

Moreover, we see from (78) that the convergence behaviour of the mean-squared error is independent of the eigenvalues of the ensemble-averaged matrix $\boldsymbol{S}_E$.

The scalar

$$J_{\mathrm{mis}}(m) = E\left\{\underline{\boldsymbol{\epsilon}}^H(m)\underline{\boldsymbol{\epsilon}}(m)\right\} \tag{80}$$

describes of the convergence of the misalignment, i.e. the coefficient convergence. Using (77), we deduce that

$$\begin{aligned} J_{\mathrm{mis}}(m) &= \mathrm{tr}[\boldsymbol{M}(m)] \\ &= \sigma_n^2(1-\lambda)^2\frac{1+\lambda^{m+1}}{1-\lambda^{m+1}}\mathrm{tr}[\boldsymbol{S}^{-1}] \\ &= \sigma_n^2(1-\lambda)^2\frac{1+\lambda^{m+1}}{1-\lambda^{m+1}}\sum_{i=0}^{L-1}\frac{1}{\lambda_{\mathrm{s},i}}, \end{aligned} \tag{81}$$

where the $\lambda_{\mathrm{s},i}$ denote the eigenvalues of the ensemble-averaged matrix $\boldsymbol{S}_E$. It is important to notice the difference between the convergence of the mean-squared error and the misalignment. While the mean-squared error does not depend on the eigenvalues of $\boldsymbol{S}_E$, the misalignment is magnified by the inverse of the smallest eigenvalue $\lambda_{\mathrm{s,min}}$ of $\boldsymbol{S}_E$ (and thus of $\boldsymbol{S}(m)$). The situation is worsened when the variance of the noise σ_n^2 is large. So in practice, at some frequencies, where the signal is poorly excited, we may have a very large misalignment. In order to avoid this problem and to keep the misalignment low, the adaptive algorithm should be regularized by adding small values to the diagonal of $\boldsymbol{S}(m)$. In Section VI, this important topic is discussed in more detail.

IV. Generalized Frequency-Domain Adaptive MIMO Filtering

In this section, we consider the extension of the algorithm proposed in Section II to the general MIMO case, i.e., we have P input signals $x_p(k)$, $p = 1,\ldots,P$, and Q desired signals $y_q(k)$, output signals $\hat{y}_q(k)$, and error signals $e_q(k)$, $q = 1,\ldots,Q$, respectively (Fig. 3).

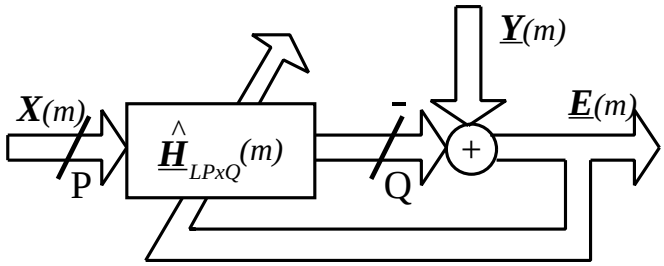


Fig. 3. Adaptive MIMO filtering in the frequency domain.

Now, the following questions are discussed: What is the optimum solution? Can correlation between the error signals $e_q(k)$ be exploited and how do the resulting update equations look like?

Let us define signal block vectors $\boldsymbol{y}_q(m)$, $\boldsymbol{e}_q(m)$, $\underline{\boldsymbol{y}}_q(m)$, $\underline{\boldsymbol{e}}_q(m)$ for each output channel in the same way as shown in (10), (9), (22), and (21), respectively. These quantities can be combined in the $(L \times Q)$ matrices

$$\begin{aligned}
\boldsymbol{E}(m) &= [\boldsymbol{e}_1(m), \cdots, \boldsymbol{e}_Q(m)], \\
\boldsymbol{Y}(m) &= [\boldsymbol{y}_1(m), \cdots, \boldsymbol{y}_Q(m)], \\
\underline{\boldsymbol{E}}(m) &= [\underline{\boldsymbol{e}}_1(m), \cdots, \underline{\boldsymbol{e}}_Q(m)], \\
\underline{\boldsymbol{Y}}(m) &= [\underline{\boldsymbol{y}}_1(m), \cdots, \underline{\boldsymbol{y}}_Q(m)].
\end{aligned}$$

We consider three conceivable generalizations of the recursive least-squares error criterion proposed in (29):

Error criterion 1: Separate optimization

The most obvious approach to the problem is to treat each of the Q desired signal channels separately by the algorithm proposed above:

$$J_{\mathrm{f1},q}(m) = (1-\lambda)\sum_{i=0}^{m}\lambda^{m-i}\underline{\boldsymbol{e}}_q^H(i)\underline{\boldsymbol{e}}_q(i) \tag{82}$$

for $q = 1, \ldots, Q$. This criterion has been traditionally used in all approaches for multichannel echo cancellation, i.e. system identification.

Error criterion 2: Joint-optimization

A more general approach foresees to jointly optimize the MIMO filter by the following criterion:

$$\begin{aligned}
J_{\mathrm{f2}}(m) &= \sum_{q=1}^{Q} J_{\mathrm{f1},q}(m) \\
&= (1-\lambda)\sum_{i=0}^{m}\lambda^{m-i}\sum_{q=1}^{Q}\underline{\boldsymbol{e}}_q^H(i)\underline{\boldsymbol{e}}_q(i) \\
&= (1-\lambda)\sum_{i=0}^{m}\lambda^{m-i}\mathrm{tr}[\underline{\boldsymbol{E}}^H(i)\underline{\boldsymbol{E}}(i)] \\
&= (1-\lambda)\sum_{i=0}^{m}\lambda^{m-i}\|\mathrm{diag}\{\underline{\boldsymbol{E}}^H(i)\underline{\boldsymbol{E}}(i)\}\|_1,
\end{aligned}$$

where the matrix norm $\|\cdot\|_1$ sums up the absolute values of all matrix elements. Introducing the $(LP \times Q)$

coefficient matrix in the frequency domain

$$\underline{\hat{\boldsymbol{H}}}_{LP\times Q} = \begin{bmatrix} \underline{\hat{\boldsymbol{h}}}_{1,1} & \cdots & \underline{\hat{\boldsymbol{h}}}_{1,Q} \\ \vdots & \ddots & \vdots \\ \underline{\hat{\boldsymbol{h}}}_{P,1} & \cdots & \underline{\hat{\boldsymbol{h}}}_{P,Q} \end{bmatrix}, \tag{83}$$

and using the same approach as in Section II, we obtain the following normal equation:

$$\boldsymbol{S}(m)\underline{\hat{\boldsymbol{H}}}_{LP\times Q} = \boldsymbol{s}_{LP\times Q}(m). \tag{84}$$

Fortunately, this matrix equation can be easily decomposed into Q equations (34). Therefore, criteria 1 and 2 are strictly equivalent for the behaviour of the adaptation. We note, however, that the compact formulation (84) of the normal equation can be used, e.g. to obtain a generalized control of the adaptation for the echo cancellation application [24].

Error criterion 3: Joint-Optimization, considering cross-correlations between error signals

The last formulation of Criterion 2 reveals an interesting possibility to take the cross-correlations between the error signals into account by optimizing

$$J_{\mathrm{f3}}(m) = (1-\lambda)\sum_{i=0}^{m}\lambda^{m-i}\|\underline{\boldsymbol{E}}^H(i)\underline{\boldsymbol{E}}(i)\|_1. \tag{85}$$

Let us consider the optimization of the additional off-diagonal elements $\underline{\boldsymbol{e}}_q^H(i)\underline{\boldsymbol{e}}_r(i)$ $(q \neq r)$ of $\underline{\boldsymbol{E}}^H(i)\underline{\boldsymbol{E}}(i)$. According to [1], [23], we obtain

$$\frac{\partial}{\partial\underline{\hat{\boldsymbol{h}}}_q(i)}\underline{\boldsymbol{e}}_q^H(i)\underline{\boldsymbol{e}}_r(i) = 0, \tag{86}$$

and from

$$\frac{\partial}{\partial\underline{\hat{\boldsymbol{h}}}_r(i)}\underline{\boldsymbol{e}}_q^H(i)\underline{\boldsymbol{e}}_r(i), \tag{87}$$

we obtain the well-known normal equations (34) for $\underline{\hat{\boldsymbol{h}}}_q$.

Therefore, for all criteria, the generalized frequency-domain adaptive MIMO filter can be summarized as

$$\begin{aligned}
\boldsymbol{S}_{\mathrm{d}}(m) &= \lambda\boldsymbol{S}_{\mathrm{d}}(m-1) + (1-\lambda)\boldsymbol{X}^H(m)\boldsymbol{G}_{2L\times 2L}^{01}\boldsymbol{X}(m) && (88)\\
\boldsymbol{K}(m) &= (1-\lambda)\boldsymbol{S}_{\mathrm{d}}^{-1}(m)\boldsymbol{X}^H(m) && (89)\\
\underline{\boldsymbol{E}}_{2L\times Q}(m) &= \underline{\boldsymbol{Y}}_{2L\times Q}(m) - \boldsymbol{G}_{2L\times 2L}^{01}\boldsymbol{X}(m)\underline{\hat{\boldsymbol{H}}}_{2LP\times Q}(m-1) && (90)\\
\underline{\hat{\boldsymbol{H}}}_{2LP\times Q}(m) &= \underline{\hat{\boldsymbol{H}}}_{2LP\times Q}(m-1) + \boldsymbol{G}_{2LP\times 2LP}^{10}\boldsymbol{K}(m)\underline{\boldsymbol{E}}_{2L\times Q}(m) && (91)
\end{aligned}$$

in analogy to equations (51) to (54).

Note that for block size $N < L$, an algorithm is obtained in the same way as mentioned in section II.

V. Approximation and Special Cases

We start this section by giving a very useful approximation of the algorithm proposed in the preceding Section. We then derive some examples of classical and efficient algorithms. This list is not exhaustive and several other algorithms may also be derived.

A. Approximation of the Frequency-Domain Kalman Gain

Frequency-domain adaptive filters were first introduced to reduce the arithmetic complexity of the (single-channel) LMS algorithm [13]. Unfortunately, the matrix $\boldsymbol{S}_{\mathrm{d}}$ is generally not diagonal, so its inversion in (89) has a high complexity and the algorithm may not be very useful in practice. Since $\boldsymbol{S}_{\mathrm{d}}$ is composed of P^2 sub-matrices

$$\boldsymbol{S}_{i,j} = \lambda \boldsymbol{S}_{i,j}(m-1) + (1-\lambda)\boldsymbol{X}_i^*(m)\boldsymbol{G}_{2L\times 2L}^{01}\boldsymbol{X}_j(m), \tag{92}$$

it is desirable that each of those sub-matrices be a diagonal matrix. In the next paragraph, we will argue that $\boldsymbol{G}_{2L\times 2L}^{01}$ can be well approximated by the identity matrix with weight1/2; accordingly, after introducing the positive factor $\mu \leq 2$ in (91), we then obtain the following approximate algorithm:

$$\begin{aligned}
\boldsymbol{S}'(m) &= \lambda \boldsymbol{S}'(m-1) + (1-\lambda)\boldsymbol{X}^H(m)\boldsymbol{X}(m) && (93)\\
\boldsymbol{K}(m) &= (1-\lambda)\boldsymbol{S}'^{-1}(m)\boldsymbol{X}^H(m) && (94)\\
\underline{\boldsymbol{E}}_{2L\times Q}(m) &= \underline{\boldsymbol{Y}}_{2L\times Q}(m) - \boldsymbol{G}_{2L\times 2L}^{01}\boldsymbol{X}(m)\underline{\hat{\boldsymbol{H}}}_{2LP\times Q}(m-1) && (95)\\
\underline{\hat{\boldsymbol{H}}}_{2LP\times Q}(m) &= \underline{\hat{\boldsymbol{H}}}_{2LP\times Q}(m-1) + \mu \boldsymbol{G}_{2LP\times 2LP}^{10}\boldsymbol{K}(m)\underline{\boldsymbol{E}}_{2L\times Q}(m), && (96)
\end{aligned}$$

where each sub-matrix of $\boldsymbol{S}'$ and $\boldsymbol{K}$ is now a diagonal matrix and $\mu \leq 2$ is a positive number. Note that the imprecision introduced by the approximation in (93) and thus in the Kalman gain (94) will only affect the convergence rate. Obviously, we can not permit the same kind of approximation in (95), because that would result in approximating a linear convolution by a circular one, which of course can have a disastrous impact in our adaptive filtering problem.

To justify the above approximation, let us examine the structure of the matrix $\boldsymbol{G}_{2L\times 2L}^{01}$. We have

$$(\boldsymbol{G}_{2L\times 2L}^{01})^* = \boldsymbol{F}_{2L\times 2L}^{-1}\boldsymbol{W}_{2L\times 2L}^{01}\boldsymbol{F}_{2L\times 2L}. \tag{97}$$

Since $\boldsymbol{W}_{2L\times 2L}^{01}$ is a diagonal matrix, $(\boldsymbol{G}_{2L\times 2L}^{01})^*$ is a circulant matrix. Therefore, inverse transformation of the diagonal of $\boldsymbol{W}_{2L\times 2L}^{01}$ gives the first column of $(\boldsymbol{G}_{2L\times 2L}^{01})^*$,

$$\begin{aligned}
\boldsymbol{g}^* &= [g_0^*, g_1^*, \cdots, g_{2L-1}^*]^T\\
&= \boldsymbol{F}_{2L\times 2L}^{-1}[0, \cdots, 0, 1, \cdots, 1]^T.
\end{aligned}$$

The elements of vector $\boldsymbol{g}$ can be written explicitly as:

$$\begin{aligned}
g_k &= \frac{1}{2L}\sum_{l=L}^{2L-1}\exp(-j2\pi kl/2L)\\
&= \frac{(-1)^k}{2L}\sum_{l=0}^{L-1}\exp(-j\pi kl/L), && (98)
\end{aligned}$$

where $j^2 = -1$. Since g_k is the sum of a finite geometric series, we have:

$$g_k = \begin{cases} 0.5 & k = 0\\ \frac{(-1)^k}{2L}\frac{1-\exp(-j\pi k)}{1-\exp(-j\pi k/L)} & k \neq 0 \end{cases}$$

$$= \begin{cases} 0.5 & k = 0 \\ 0 & k \text{ even} \\ -\frac{1}{2L}\left[1 - j\cot\left(\frac{\pi k}{2L}\right)\right] & k \text{ odd}, \end{cases} \tag{99}$$

where $L-1$ elements of vector $\boldsymbol{g}$ are equal to zero. Moreover, since $(\boldsymbol{G}^{01}_{2L\times 2L})^H \boldsymbol{G}^{01}_{2L\times 2L} = \boldsymbol{G}^{01}_{2L\times 2L}$, then $\boldsymbol{g}^H\boldsymbol{g} = g_0 = 0.5$ and we have

$$\boldsymbol{g}^H\boldsymbol{g} - g_0^2 = \sum_{l=1}^{2L-1} |g_l|^2 = 2\sum_{l=1}^{L-1} |g_l|^2 = \frac{1}{4}. \tag{100}$$

We can see from (100) that the first element of vector $\boldsymbol{g}$, i.e., g_0, is dominant in a mean-square sense, and from (99) that the absolute values of the L first elements of $\boldsymbol{g}$ decrease rapidly to zero as k increases. Because of the conjugate symmetry, i.e. $|g_k| = |g_{2L-k}|$ for $k = 1, \ldots, L-1$, the last few elements of $\boldsymbol{g}$ are not negligible, but this affects only the first and last columns of $\boldsymbol{G}^{01}_{2L\times 2L}$ since this matrix is circulant with $\boldsymbol{g}$ as its first column. All other columns have those non-negligible elements wrapped around in such a way that they are concentrated around the main diagonal. To summarize, we can say that for L large, only the very first (few) off-diagonals of $\boldsymbol{G}^{01}_{2L\times 2L}$ will be non-negligible while the others can be completely neglected. We also neglect the influence of the two isolated peaks $|g_{2L-1}| = |g_1| < g_0$ on the lower left corner and the upper right corner, respectively. Thus, approximating $\boldsymbol{G}^{01}_{2L\times 2L}$ by a diagonal matrix, i.e., $\boldsymbol{G}^{01}_{2L\times 2L} \approx g_0\boldsymbol{I} = \boldsymbol{I}/2$, is reasonable, and in this case we will have $\mu \approx 1/g_0 = 2$ for an optimum convergence rate. For the rest of this paper, we suppose that $0 < \mu \leq 2$.

B. Special Cases

In the single-channel case $P = Q = 1$, $\boldsymbol{S}'$ and $\boldsymbol{K}$ are diagonal matrices and the classical constrained FLMS [13] follows immediately from (93)-(96). This algorithm requires the computation of 5 FFTs of length $2L$ per block. By approximating $\boldsymbol{G}^{10}_{2LP\times 2LP}$ in (96) to the identity matrix, we obtain the unconstrained FLMS (UFLMS) algorithm [14] which requires only 3 FFTs per block. Many simulations show that the two algorithms have virtually the same performance.

Note that for $N < L$, $\boldsymbol{S}_{\mathrm{d}}(m)$ in (88) consists of $(K \cdot P)^2$ sub-matrices that can be approximated as shown above. It is interesting that for $N = 1$, the algorithm is strictly equivalent to the RLS algorithm in the time domain. After the approximation, we obtain an *extended multidelay filter* (EMDF) for $1 < N < L$ that takes the auto-correlations between the blocks into account. Finally, the classical multidelay filter is obtained by further approximating $\boldsymbol{S}'(m)$ in (93) by

$$\boldsymbol{S}''(m) = \mathrm{diag}\{\boldsymbol{S}_{MDF}(m), \cdots, \boldsymbol{S}_{MDF}(m)\}, \tag{101}$$

where

$$\boldsymbol{S}_{MDF}(m) \;=\; \lambda\boldsymbol{S}_{MDF}(m-1) + (1-\lambda)\boldsymbol{X}^*_{2N\times 2N}(m)\boldsymbol{X}_{2N\times 2N}(m)$$

is a $(2N \times 2N)$ diagonal matrix.

In the multichannel case, (94) is the solution of a $P \times P$ system of linear equations of block matrices:

$$\boldsymbol{K}(m) = [\boldsymbol{K}_1^T(m), \cdots, \boldsymbol{K}_P^T(m)]^T. \tag{102}$$

This allows decomposition of the update equation (96) into PQ single-channel update equations

$$\underline{\hat{\boldsymbol{h}}}_{p,q}(m) = \underline{\hat{\boldsymbol{h}}}_{p,q}(m-1) + \mu \boldsymbol{G}^{10}_{2L\times 2L} \boldsymbol{K}_p \underline{\boldsymbol{e}}_q(m) \tag{103}$$

($p = 1, \ldots, P,\ q = 1, \ldots, Q$) with the sub-matrices $\boldsymbol{K}_p(m)$ taking the cross-correlations between the input channels into account. These decomposed update equations can then be calculated element-wise and the (cross) power spectra are estimated recursively:

$$\boldsymbol{S}_{i,j}(m) = \lambda \boldsymbol{S}_{i,j}(m-1) + (1-\lambda)\boldsymbol{X}^*_i(m)\boldsymbol{X}_j(m), \tag{104}$$

where $\boldsymbol{S}_{j,i}(\cdot) = \boldsymbol{S}^*_{i,j}(\cdot)$.

It is important to note that the calculation of the Kalman gain (equations (88) and (88)), which is the computationally most demanding part, is completely independent of the number Q of output channels and thus, has to be calculated only once, while the remaining update equations (103) corresponds to simple (U)FLMS algorithms.

In the case of two input channels $P = 2$, the Kalman gain can be written in an explicit form by block-inversion:

$$\boldsymbol{K}_1 = \boldsymbol{D}(m)\boldsymbol{S}^{-1}_{1,1}(m)[\boldsymbol{X}^*_1(m) - \boldsymbol{S}_{1,2}(m)\boldsymbol{S}^{-1}_{2,2}(m)\boldsymbol{X}^*_2(m)] \tag{105}$$

$$\boldsymbol{K}_2 = \boldsymbol{D}(m)\boldsymbol{S}^{-1}_{2,2}(m)[\boldsymbol{X}^*_2(m) - \boldsymbol{S}_{2,1}(m)\boldsymbol{S}^{-1}_{1,1}(m)\boldsymbol{X}^*_1(m)], \tag{106}$$

with the abbreviation

$$\boldsymbol{D}(m) = (1-\lambda)[\boldsymbol{I}_{2L\times 2L} - \boldsymbol{S}^*_{1,2}(m)\boldsymbol{S}_{1,2}(m)\{\boldsymbol{S}_{1,1}(m)\boldsymbol{S}_{2,2}(m)\}^{-1}]^{-1}.$$

The solutions of (94) for more than two input channels may be formulated similarly to the corresponding part of the stereo update equations (105) and (106) (e.g. using Cramer's rule). These representations allow an intuitive interpretation: as a correction of the interchannel-correlations in $\boldsymbol{K}_i$ between $\boldsymbol{X}^*_i$ and the other input signals $\boldsymbol{X}^*_j, j \neq i$.

For three channels, we have (omitting, for simplicity, the time index m of all matrices)

$$\begin{aligned}\boldsymbol{K}_1 &= (1-\lambda)\boldsymbol{D}^{-1}[\boldsymbol{X}^*_1(\boldsymbol{S}_{2,2}\boldsymbol{S}_{3,3} - \boldsymbol{S}_{3,2}\boldsymbol{S}_{2,3}) - \boldsymbol{X}^*_2(\boldsymbol{S}_{1,2}\boldsymbol{S}_{3,3} - \boldsymbol{S}_{1,3}\boldsymbol{S}_{3,1}) - \boldsymbol{X}^*_3(\boldsymbol{S}_{1,3}\boldsymbol{S}_{2,2} - \boldsymbol{S}_{1,2}\boldsymbol{S}_{2,3})],\\ \boldsymbol{D} &:= \boldsymbol{S}_{1,1}(\boldsymbol{S}_{2,2}\boldsymbol{S}_{3,3} - \boldsymbol{S}_{3,2}\boldsymbol{S}_{2,3}) - \boldsymbol{S}_{2,1}(\boldsymbol{S}_{1,2}\boldsymbol{S}_{3,3} - \boldsymbol{S}_{1,3}\boldsymbol{S}_{3,1}) - \boldsymbol{S}_{3,1}(\boldsymbol{S}_{1,3}\boldsymbol{S}_{2,2} - \boldsymbol{S}_{1,2}\boldsymbol{S}_{2,3})\end{aligned}$$

as the first of the three Kalman gain components with the common factor $\boldsymbol{D}$.

Unfortunately, for a higher number of channels, the number of update terms increases rapidly, and the equations become too complicated for practial use. Therefore, a more efficient scheme for these cases will be proposed in section VII.

VI. A Dynamical Regularization Strategy

In most practical scenarios, the desired signal $y(k)$ is disturbed, e.g., by some acoustic background noise. As shown above (c.f. (81)), the parameter estimation (i.e., misalignment) is very sensitive in poorly excited

frequency bins. For robust adaptation the power spectral densities $\boldsymbol{S}_{i,i}$ are replaced by regularized versions according to $\tilde{\boldsymbol{S}}_{i,i} = \boldsymbol{S}_{i,i} + \mathrm{diag}\{\boldsymbol{\delta}_i\}$ prior to inversion in (89). The basic feature of the regularization is a compromise between fidelity to data and fidelity to some prior information about the solution [25]. The latter increases the robustness, but leads to biased solutions. Therefore, we propose here a *bin-selective dynamical regularization vector*

$$\boldsymbol{\delta}_i(m) = \delta_{\max} \cdot [e^{-S_{i,i}^{(0)}(m)/S_0}, \cdots, e^{-S_{i,i}^{(2L-1)}(m)/S_0}]^T \tag{107}$$

with two scalar parameters $\delta_{\max}$ and S_0. $S_{i,i}^{(\nu)}$ denotes the ν-th frequency component ($\nu = 0, \cdots, 2L-1$) on the main diagonal of $S_{i,i}$. Note that for efficient implementation, e in (107) may be replaced by a basis 2 and modified S_0. $\delta_{\max}$ should be chosen according to the (estimated) disturbing noise level in the desired signal $y(k)$.

This exponential method provides a smooth transition between regularization for low input power and data fidelity whenever the input power is high enough, and yields improved results compared to fixed regularization and to the popular approach of choosing the maximum out of the respective component $\boldsymbol{S}_{i,i}^{(\nu)}$ and a fixed threshold δ_{th} (Fig. 4). It copes well with unbalanced excitation of the input channels, and most importantly, it can be easily extended for the efficient Kalman gain calculation introduced in the next section.

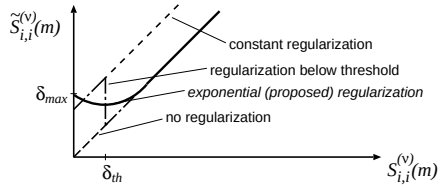


Fig. 4. Different regularization methods (channel i, bin ν).

VII. Efficient Multichannel Realization

As will be demonstrated by simulation results and real-world applications in Section VIII, the presented algorithm copes well with multichannel input. The case of a larger number of filter input channels (P larger than 2 or 3) calls for further improvement of the computational efficiency. In this section, we propose efficient and stable recursive calculation schemes for the frequency-domain Kalman gain for and the DFTs of the overlapping input data blocks for the case of a large number of filter input channels. Overlapping input data blocks result from an overlap factor $\alpha > 1$, originally proposed in [19]. Incorporating this extension in the proposed algorithm is very simple. Essentially, only the way the input data matrices (17) are calculated, is modified to

$$\boldsymbol{X}_p(m) = \mathrm{diag}\{\boldsymbol{F}_{2L\times 2L}[x_p(m\frac{L}{\alpha} - L + 1), \cdots, x_p(m\frac{L}{\alpha} + L)]^T\}. \tag{108}$$

Simulations show that increased overlap factors α are particularly useful in the multichannel case.

A. Efficient Calculation of the Frequency-Domain Kalman Gain

For a *practical implementation* of a system with $P > 2$ channels, we propose computationally more efficient methods to calculate (94) as follows.

Due to the block diagonal structure of (94), it can be simply decomposed into $2L$ equations

$$\boldsymbol{K}^{(\nu)}(m) = (1-\lambda)\left(\boldsymbol{S}^{(\nu)}(m)\right)^{-1}(\boldsymbol{X}^{(\nu)}(m))^H \tag{109}$$

with (small) $P \times P$ unitary and positive definite matrices $\boldsymbol{S}^{(\nu)}$ for the components $\nu = 0, \ldots, 2L-1$ on the diagonals. Both $\boldsymbol{K}^{(\nu)}$ and $\boldsymbol{X}^{(\nu)}$ are *vectors* of length P. Note that for real input signals x_i we need to solve (109) only for $L+1$ bins.

A well-known and numerically stable method for this type of problems is the Cholesky decomposition of $\boldsymbol{S}^{(\nu)}$ followed by solution via backsubstitution, see [26]. The resulting total complexity for one output value is then

$$O(P \cdot \log_2(2L)) + O(P^3), \tag{110}$$

where in the two-channel (stereo) case the second term $O(P^3)$ is much smaller than the share due to the first term.

For a large number (≥ 3) of input channels (see, e.g., the applications in Section VIII) we introduce a recursive solution of (109) that jointly estimates the *inverse* power spectra $(\boldsymbol{S}^{(\nu)})^{-1}$ in (93) using the matrix-inversion lemma, e.g. [1]. This lemma relates a matrix

$$\boldsymbol{A} = \boldsymbol{B}^{-1} + \boldsymbol{C}\boldsymbol{D}^{-1}\boldsymbol{C}^H \tag{111}$$

to its inverse according to

$$\boldsymbol{A}^{-1} = \boldsymbol{B} - \boldsymbol{B}\boldsymbol{C}(\boldsymbol{D} + \boldsymbol{C}^H\boldsymbol{B}\boldsymbol{C})^{-1}\boldsymbol{C}^H\boldsymbol{B}, \tag{112}$$

as long as $\boldsymbol{A}$ and $\boldsymbol{B}$ are positive definite. Comparing (93) to (111), we immediately obtain from (93) an update equation for the *inverse* matrices

$$(\boldsymbol{S}^{(\nu)}(m))^{-1} = \lambda^{-1}\left[(\boldsymbol{S}^{(\nu)}(m-1))^{-1} - \frac{(\boldsymbol{S}^{(\nu)}(m-1))^{-1}(\boldsymbol{X}^{(\nu)}(m))^H\boldsymbol{X}^{(\nu)}(m)(\boldsymbol{S}^{(\nu)}(m-1))^{-1}}{\lambda(1-\lambda)^{-1} + \boldsymbol{X}^{(\nu)}(m)(\boldsymbol{S}^{(\nu)}(m-1))^{-1}(\boldsymbol{X}^{(\nu)}(m))^H}\right]$$

using the decomposition (109) (making the denominator a scalar value).

Introduction of the common vector

$$\boldsymbol{T}_1^{(\nu)}(m) = (\boldsymbol{S}^{(\nu)}(m-1))^{-1}(\boldsymbol{X}^{(\nu)}(m))^H \tag{113}$$

in the numerator and the denominator leads to

$$(\boldsymbol{S}^{(\nu)}(m))^{-1} = \lambda^{-1}(\boldsymbol{S}^{(\nu)}(m-1))^{-1} - \frac{\boldsymbol{T}_1^{(\nu)}(m)(\boldsymbol{T}_1^{(\nu)}(m))^H}{\lambda^2(1-\lambda)^{-1} + \lambda\boldsymbol{X}^{(\nu)}(m)\boldsymbol{T}_1^{(\nu)}(m)}. \tag{114}$$

The Kalman gain (109) can then be efficiently calculated (using (114)) by

$$\boldsymbol{K}^{(\nu)}(m) = \frac{1-\lambda}{\lambda}\boldsymbol{T}_1^{(\nu)}(m)\left[1 - \frac{(\boldsymbol{T}_1^{(\nu)}(m))^H(\boldsymbol{X}^{(\nu)}(m))^H}{\lambda(1-\lambda)^{-1} + \boldsymbol{X}^{(\nu)}(m)\boldsymbol{T}_1^{(\nu)}(m)}\right]. \tag{115}$$

Again, there are common factors

$$T_2^{(\nu)}(m) = \boldsymbol{X}^{(\nu)}(m)\boldsymbol{T}_1^{(\nu)}(m) \tag{116}$$

in (115) and (114).

Note that our approach should not be confused with the classical RLS approach [5] which also makes use of the matrix-inversion lemma. As we apply the lemma independently to small $P \times P$ systems (109), it is numerically much less critical than in the RLS algorithm. Moreover, there is no analogon to a more efficient *fast* RLS [27] due to the different matrix structures (vector $\boldsymbol{X}^{(\nu)}(m)$ does not reflect a tapped delay line).

The complexity of the different computation methods for the Kalman gains (for one output value $e(k)$) are compared in Fig. 5. Note that our approach is particularly efficient for the extended multidelay filter (block

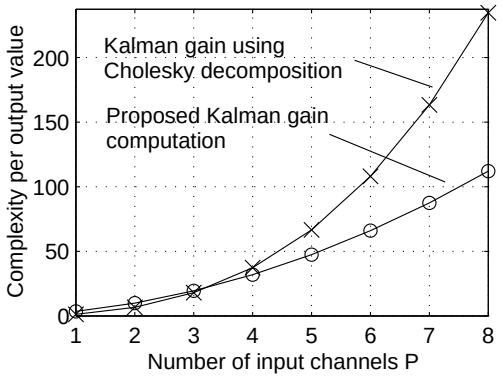


Fig. 5. Complexity of Kalman gain.

size $N < L$) introduced in section V.

B. Dynamical Regularization for Proposed Kalman Gain Approach

Due to the recursion (114) the regularization according to (107) is not immediately applicable. Therefore, an equivalent modification is applied directly to the data matrices $\boldsymbol{X}^{(\nu)}(m)$ by addition of mutually uncorrelated white noise sequences to each channel and frequency bin, respectively.

Using the modified signal vectors, denoted by

$$\tilde{\boldsymbol{X}}^{(\nu)}(m) = \boldsymbol{X}^{(\nu)}(m) + \boldsymbol{N}^{(\nu)}(m), \tag{117}$$

where $\boldsymbol{N}^{(\nu)}(m)$ are the vectors of the white noise signals, we obtain the modified power spectral density matrices (c.f. Eq. (93))

$$\tilde{\boldsymbol{S}}^{(\nu)}(m) \approx (1-\lambda)\sum_{q=0}^{m}\lambda^{m-q}\boldsymbol{X}^{(\nu)H}(q)\boldsymbol{X}^{(\nu)}(q) + (1-\lambda)\sum_{q=0}^{m}\lambda^{m-q}\mathrm{diag}\{[|N_1^{(\nu)}(q)|^2,\cdots,|N_P^{(\nu)}(q)|^2]^T\}. \tag{118}$$

The diagonal elements of the second term can be interpreted as a bin-selective dynamical regularization vector $\boldsymbol{\delta}^{(\nu)}(m)$ with elements (for channel i and bin ν)

$$\begin{aligned}\delta_i^{(\nu)}(m) &= (1-\lambda)\sum_{q=0}^{m}\lambda^{m-q}|N_i^{(\nu)}(q)|^2, \\ &= \lambda\delta_i^{(\nu)}(m-1) + (1-\lambda)|N_i^{(\nu)}(m)|^2. \end{aligned}\tag{119}$$

Thus, in order to update the regularization from $\delta_i^{(\nu)}(m-1)$ to $\delta_i^{(\nu)}(m)$ with the appropriate speed (determined by λ), we need to add noise with power

$$|N_i^{(\nu)}(m)|^2 = \frac{\delta_i^{(\nu)}(m) - \lambda\delta_i^{(\nu)}(m-1)}{1-\lambda}. \tag{120}$$

On the other hand, according to (107), the regularization should be chosen according to

$$\begin{aligned}\delta_i^{(\nu)}(m) &= \delta_{\max} \cdot \exp\left(-\frac{S_{i,i}^{(\nu)}(m)}{S_0}\right) \\ &= \delta_{\max} \cdot \exp\left(-\frac{\lambda S_{i,i}^{(\nu)}(m-1) + (1-\lambda)|X_i^{(\nu)}(m)|^2}{S_0}\right). \end{aligned} \tag{121}$$

Now, unlike other dynamical regularization methods, the exponential regularization allows simple elimination of the elements $S_{i,i}^{(\nu)}(m-1)$ of the non-inverted matrix (which need not be computed at all due to the matrix-inversion lemma (112)), since

$$\begin{aligned}\delta_i^{(\nu)}(m) &= \delta_{\max}\left[\exp\left(-\frac{S_{i,i}^{(\nu)}(m-1)}{S_0}\right)\right]^{\lambda} \cdot \exp\left(-\frac{(1-\lambda)|X_i^{(\nu)}(m)|^2}{S_0}\right) \\ &= \delta_{\max}^{1-\lambda}(\delta_i^{(\nu)}(m-1))^{\lambda} \cdot \exp\left(-\frac{(1-\lambda)|X_i^{(\nu)}(m)|^2}{S_0}\right). \end{aligned} \tag{122}$$

C. Efficient DFT calculation of overlapping data blocks

In this section we address the first term of the computational cost given in (110) which is mainly determined by the DFTs of the frequency-domain adaptive filtering scheme (Fig. 2). The $2L$-point DFT calculation in (108) has to be carried out for each of the P loudspeaker signals and is therefore most costly. Moreover, as will be discussed in Section VIII, an increased overlap factor α is often desirable in the multichannel case. Therefore, we aim at exploiting the overlap of the input data blocks by implementing (108) recursively as well. Note that a similar idea using a different approach was suggested in [28] for the single-channel case.

For the following derivation,

$$x_i^{(k)}(m) = x_i\left(m\frac{L}{\alpha} - L + 1 + k\right) \tag{123}$$

denotes the k-th component ($k = 0, \ldots, 2L-1$) of the time domain vector (block index m) to be transformed in (108). Let us now consider the ν-th element on the diagonal of $\boldsymbol{X}_i(m)$ where $w = e^{-j2\pi/2L}$:

$$X_i^{(\nu)}(m) = \sum_{k=0}^{2L-1} x_i^{(k)}(m) w^{\nu k}. \tag{124}$$

Separating the summation into one for previous and one for new input values (Fig. 6), followed by the introduction of the previous vector elements $x_i^{(k)}(m-1)$ leads to

$$\begin{aligned}X_i^{(\nu)}(m) &= \sum_{k=0}^{2L-L/\alpha-1} x_i^{(k)}(m) w^{\nu k} + \sum_{k=2L-L/\alpha}^{2L-1} x_i^{(k)}(m) w^{\nu k} \\ &= \sum_{k=L/\alpha}^{2L-1} x_i^{(k)}(m-1) w^{\nu(k-L/\alpha-1)} + \Delta X_i^{(\nu)}(m), \end{aligned} \tag{125}$$

where

$$\Delta X_i^{(\nu)}(m) = \sum_{k=2L-L/\alpha}^{2L-1} x_i^{(k)}(m) w^{\nu k} \tag{126}$$

contains the new input values and will be the update term in our recursive scheme. Next, we introduce the

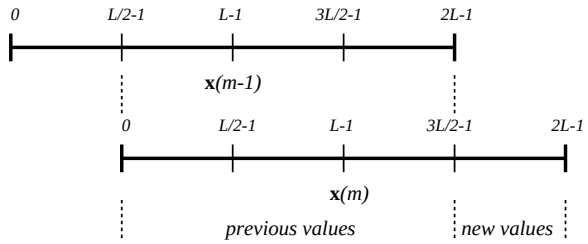


Fig. 6. Example: overlapping data blocks, $\alpha = 2$.

previous DFT output values $X_i^{(\nu)}(m-1)$ by subtracting the vector elements of $\boldsymbol{x}_i^{(k)}(m-1)$ of the previous data vector shifted out of the DFT length $2L$:

$$\begin{aligned} X_i^{(\nu)}(m) &= w^{-\nu L/\alpha}\left[\sum_{k=0}^{2L-1} x_i^{(k)}(m-1)w^{\nu k} - \sum_{k=0}^{L/\alpha-1} x_i^{(k)}(m-1)w^{\nu k}\right] + \Delta X_i^{(\nu)}(m) \\ &= w^{\nu L/\alpha} X_i^{(\nu)}(m-1) - w^{\nu L/\alpha} \sum_{k=2L-L/\alpha}^{2L-1} x_i^{(k-2L+L/\alpha)}(m) w^{\nu(k-2L+L/\alpha)} + \Delta X_i^{(\nu)}(m). \quad (127) \end{aligned}$$

Using (123), we can show that

$$x_i^{(k-2L+L/\alpha)}(m) = x_i^{(k)}(m-2\alpha+1). \quad (128)$$

Finally, be obtain

$$X_i^{(\nu)}(m) = w^{-\nu L/\alpha} X_i^{(\nu)}(m-1) - w^{-\nu 2L} \Delta X_i^{(\nu)}(m-2\alpha+1) + \Delta X_i^{(\nu)}(m).$$

Again, this recursive update needs to be carried out only for the bins $\nu = 0, \ldots, L$ if $x_i^{(k)}(m)$ is real-valued. Only the update $\Delta X_i^{(\nu)}(m)$ in this equation has to be calculated explicitly using the L/α new values of the input vector.

With the truncation of the time-domain input vector for calculating $\Delta X_i^{(\nu)}(m)$ in mind, we consider now the decimation-in-frequency FFT algorithm. Figure 7 shows a simple example for $2L = 8$ and $\alpha = 2$. $2L - L/\alpha$ inputs (thin lines) always carry zero value. As can be seen from the figure, the first $\log_2(\alpha)$ stages do not contain any summations while for the following stages any FFT algorithm (e.g. from highly optimized software libraries) can be employed. Note that for the decimation-in-time approach one would need a special FFT

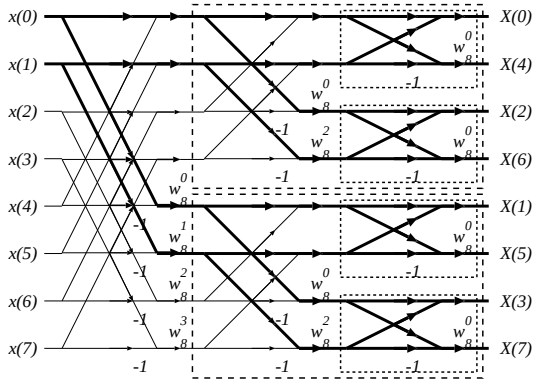


Fig. 7. Illustration of decimation-in-frequency FFT with windowed input.

implementation for all stages in order to take advantage from high overlapping factors α. In summary, the

recursive DFT approach reduces the first term of the complexity in (110) to $O(P \cdot \log_2(L/\alpha))$ for each output point.

VIII. Simulations and Real-World Applications

As mentioned in the introduction, there are many areas of applications for multichannel adaptive filtering. In the following, we demonstrate the performance of our approach in a few examples for hands-free speech communication.

A. Multichannel Acoustic Echo Cancellation

For applications such as home entertainment, virtual reality (e.g., games, training), or advanced teleconferencing, there is a growing interest in multimedia terminals with an increased number of audio channels for sound reproduction (e.g., stereo or 5.1 channel - surround systems). In such applications, multichannel acoustic echo cancellation is a key technology whenever hands-free and full-duplex communication is desired (Fig. 8).

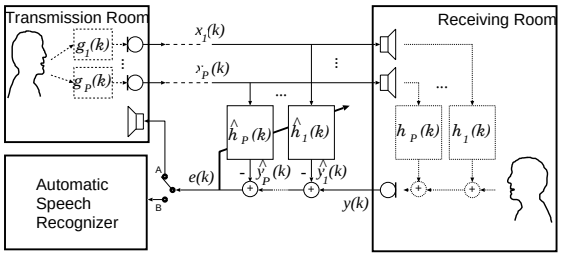
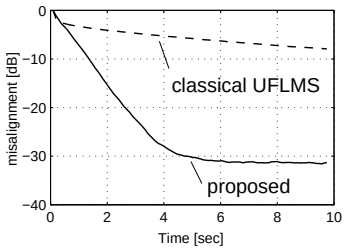


Fig. 8. Multichannel acoustic echo cancellation.

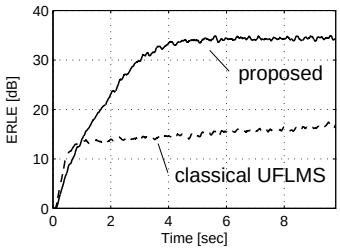
The fundamental problem is that the multiple channels may carry linearly related signals which in turn may make the normal equation to be solved by the adaptive algorithm singular. This implies that there is no unique solution to the equation but an infinite number of solutions and it can be shown that all but the true one depend on the impulse responses of the transmission room [7], [8]. It is shown in [9] that the only solution to the nonuniqueness problem is to reduce the correlation between the different signals. Three methods of preprocessing can be distinguished: nonlinear processing, e.g., [9], additive noise (below the masking threshold of human hearing), e.g., [29], and time-varying filtering, e.g., [30]. For the following simulations, a signal from a common source (in the transmission room) was convolved by P different room impulse responses and nonlinearly, but inaudibly preprocessed according to [9] (P different nonlinearities with factor 0.5). In this subsection we consider only one microphone in the receiving room. The convergence behaviour is shown both in terms of system misalignment (ratio of the squared norms of (63) and the desired response), and in terms of echo return loss enhancement ($ERLE$) which describes the ratio of the short-term powers of the echo $y(k) - n(k)$ and the residual echo $e(k) - n(k)$. For smoothing the $ERLE$ curves, a moving average filter of length 256 was used.

Figure 9 illustrates the effect of taking the cross-correlations in (105) and (106) into account. As input $x_p(k)$, a common white noise signal was convolved by the room impulse responses in the transmission room. Another

white noise signal was added to the echo on the microphone for $SNR = 35dB$. Here, both the receiving room impulse responses and the modeling filter lengths were chosen to be 1024 (solid lines: proposed, dashed lines: classical UFLMS algorithm).



(a)

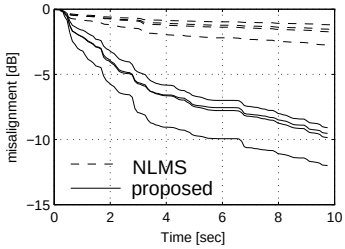


(b)

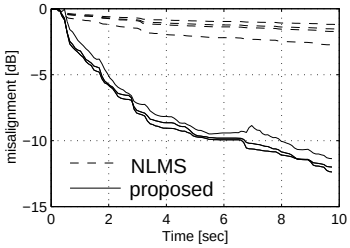
Fig. 9. Effect of taking cross-correlation into account (P=2 channels, $\alpha = 4$). (a) Misalignment, (b) $ERLE$.

For simulations with real-world signals, the lengths of the measured receiving room impulse responses were 4096 and the modeling filters were 1024, respectively. One common speech signal from the transmission room serves as input signal. Figure 10 shows the misalignment convergence of the described algorithm (solid) for the multichannel cases $P = 2, 3, 4, 5$, and the basic NLMS [1] (dashed) for comparison. In (a) the overlap factor α was set to 4 in all cases, while in (b) the overlap factor α was set to 4 for $P = 2$, and adjusted to 8 for $P = 3, 4$, and to 16 for $P = 5$. Using these parameters, the convergence curves for the different numbers of channels are almost indistinguishable. Figure 11 (a) shows the corresponding $ERLE$ curves.

Figure 11 (b) compares different regularization methods (white noise distortion as above): no regularization (uppermost curve), constant regularization (dotted), threshold (dashed), exponential with original algorithm (dash-dot), proposed Kalman gain (lower solid line).



(a)



(b)

Fig. 10. Misalignment convergence for the multichannel cases P=2,3,4,5. (a) Overlap $\alpha = 4$, (b) overlap α adjusted.

We note that for both, stereophonic teleconferencing and hands-free speech recognition applications, real-time systems could be successfully implemented on regular personal computers [2], [3].

B. Adaptive MIMO Filtering for Hands-Free Speech Communication

In applications such as hands-free speech recognition, it is very important to reduce interfering noise or competing speech signals, and reverberation of the target speech signal, in addition to the acoustic echo cancellation (Fig. 12).

(a) (b)

Fig. 11. (a) *ERLE* convergence for the multichannel cases P=2,3,4,5 and adjusted α and (b) comparison of regularization methods, P=5, α=16.

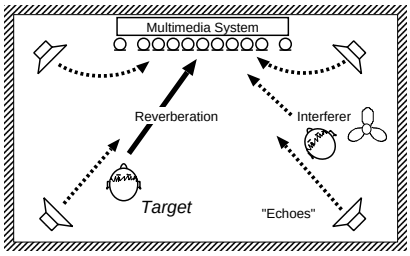


Fig. 12. Hands-free speech recognition in multimedia systems.

An efficient approach to address these problems is to replace the single microphone by a microphone array directing a beam of increased sensitivity at the active talker [31]. In any practical system, this scenario presents a MIMO system identification problem for the acoustic echo canceller [31], [3]. Fortunately, as noted in Section IV, the costly calculation of the Kalman gain is necessary only once, i.e., it is independent of the number of microphones. Figure 13 gives an example of a low-complexity structure. Echo cancellation is applied to several beamformer (BF) output signals. The fixed beamformers do not disturb the convergence of the echo cancellation and direct beams to all directions of interest [31]. Due to the efficient frequency-domain approach, e.g., stereo echo cancellation ($L = 4096$) for 5 beams is possible on an Intel-based PC (1GHz).

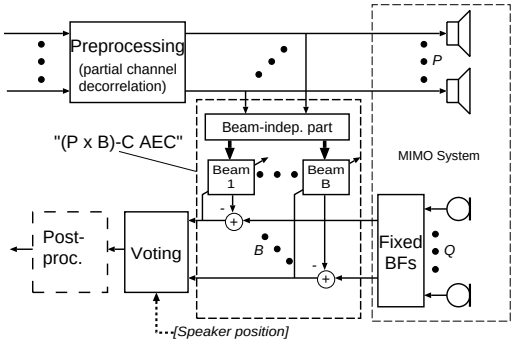


Fig. 13. A human/machine interface for hands-free speech recognition.

It is interesting to note that our *divide and conquer* approach (i.e. common Kalman gain calculation) is very efficient for the frequency-domain framework, despite of the much lower complexity compared to the basic LMS algorithm.

C. Adaptive Beamforming

Next, we show that the proposed generalized multichannel frequency-domain adaptive filter is also an interesting option for adaptive beamforming.

A simple and very effective structure for adaptive beamforming is the generalized sidelobe canceller (GSC) after Griffith and Jim [32] (Fig. 14). The fixed beamformer (FBF) enhances target signal components, and is used as reference for the adaptation of the adaptive sidelobe cancelling path, which consists of a blocking matrix (BM) and an adaptive interference canceller (AIC). For our considerations, the AIC, a multichannel adaptive filter as shown in Fig. 1, is of particular interest. It is driven by the interferer signals, while the target signal is blocked by the BM.

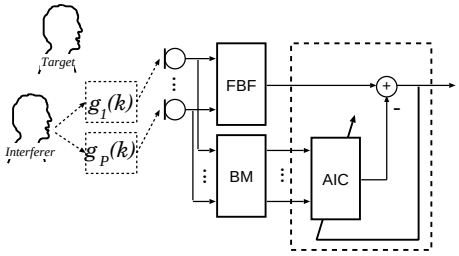


Fig. 14. Generalized Sidelobe Canceler.

To ensure robust operation (i.e., to avoid distortion of the target signal), the BM should be adaptive as well [33]. However, for simplicity, we assumed in the simulations this matrix to be fixed, as originally proposed in [32].

Often, if there is a dominant interferer, the underlying normal equation of the AIC is very ill-conditioned as in the case of multichannel acoustic echo cancellation. A low level of background noise usually ensures that there is a unique solution, but the convergence may be slowed down considerably. Figure 15 shows the Interference Rejection (IR) of a GSC with conventional UFLMS adaption. For the simulations, $P = 5$ microphone signals were used. The filter lengths were $L = 128$ and the overlap factor was set to $\alpha = 1$. In Fig. 16 (same parameters), the cross-correlations between the microphone signals were taken into account leading to significant improvement of the interference rejection.

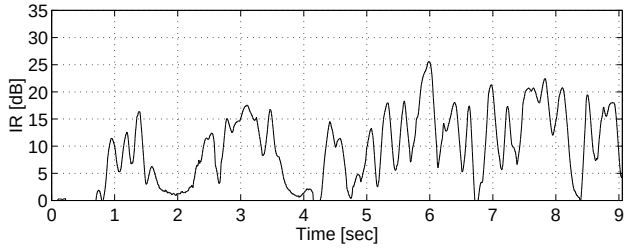


Fig. 15. Interference rejection: Generalized Sidelobe Canceller with classical UFLMS.

Note that due to the exploitation of the cross-correlations between the channels, the proposed generalized multichannel frequency-domain adaptive filter even allows to feed any other possibly highly crosscorrelated interference signals other than the BM outputs into the AIC module. The most obvious example would be

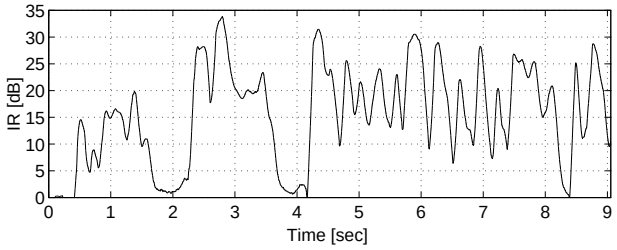


Fig. 16. Interference rejection: Generalized Sidelobe Canceller taking cross-correlations into account.

known loudspeaker signals as in the case of conventional acoustic echo cancellation. However, compared to conventional echo cancellation, this will generally lead to a suboptimum solution [12] since in that case the (time-varying) beamformer will then appear within the echo pathes to be identified.

IX. Conclusions

In many applications where an adaptive filter is required, frequency-domain algorithms are an attractive alternative to time-domain algorithms, expecially for the multichannel case. First, the computational complexity can be low by utilizing the efficiency of the FFT. Second, the convergence is improved if crucial parameters of these algorithms such as the exponential window, regularization, and adaptation step are properly chosen.

In this article a general framework for multichannel frequency-domain adaptive filtering was presented and its efficiency in actual applications was demonstrated. We have shown that a generic algorithm with an MSE convergence that is independent of the input signal statistics can be derived from the normal equation after minimizing a block least-squares criterion in the frequency domain. We analyzed the convergence of this algorithm and discussed some approximations that lead to well-known algorithms in the single-channel case, such as the FLMS and UFLMS. For the multichannel case the framework is highly interesting as the cross-correlations between all input signals are efficiently taken into account. Several simulations and real-time implementations confirm the benefits of the multichannel algorithm. We have also presented strategies to improve the computational efficiency further by introducing stable schemes for recursive DFT and Kalman gain computation.

Acknowledgment

The authors would like to thank W. Herbordt of the telecommunications institute, University of Erlangen-Nuremberg, for fruitful discussions on adaptive beamforming.

References

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

[2] V. Fischer, T. Gänsler, E. J. Diethorn, and J. Benesty, "A Software Stereo Acoustic Echo Canceler for Microsoft Windows," in *Proc. Int. Workshop on Acoustic Echo and Noise Control*, Darmstadt, Germany, pp. 87-90, Sept. 2001.

[3] H. Buchner, W. Herbordt, and W. Kellermann, "An Efficient Combination of Multichannel Acoustic Echo Cancellation With a Beamforming Microphone Array," in *Proc. Int. Workshop on Hands-Free Speech Communication*, Kyoto, Japan, pp. 55-58, April 2001.

[4] B. Widrow and S.D. Stearns, *Adaptive Signal Processing,* Prentice-Hall, Inc., Englewood Cliffs, N.J., 1985.

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

[6] R.M. Gray, "On the Asymptotic Eigenvalue Distribution of Toeplitz Matrices," *IEEE Trans. on Information Theory*, vol.18, no.6, pp. 725–730, Nov. 1972.

[7] M.M. Sondhi and D.R. Morgan, "Stereophonic Acoustic Echo Cancellation - An Overview of the Fundamental Problem," *IEEE SP Letters*, vol.2, no.8, pp. 148–151, Aug. 1995.

[8] S. Shimauchi and S. Makino, "Stereo projection echo canceller with true echo path estimation," in *Proc. ICASSP,* pp. 3059-3062, May 1995.

[9] J. Benesty, D.R. Morgan, and M.M. Sondhi, "A better understanding and an improved solution to the specific problems of stereophonic acoustic echo cancellation," *IEEE Trans. on Speech and Audio Processing*, vol. 6, no.2, March 1998.

[10] T. Gänsler and J. Benesty, "Stereophonic acoustic echo cancellation and two-channel adaptive filtering: an overview," *International Journal of adaptive control and signal processing*, Feb. 2000.

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

[12] W. Kellermann, "Strategies for combining acoustic echo cancellation and adaptive beamforming microphone arrays," Proc. ICASSP 1997, pp. 219–222.

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

[14] D. Mansour and A.H. Gray, "Unconstrained Frequency-Domain Adaptive Filter," *IEEE Trans. on Acoustics, Speech, and Signal Processing*, vol.30, no.5, Oct. 1982.

[15] J.C.Lee and C.K.Un, "Performance analysis of frequency-domain block LMS adaptive digital filters," *IEEE Trans. Circuits Syst.,* vol. CAS-36, pp. 173-189, Feb. 1989.

[16] 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.

[17] J. Benesty and P. Duhamel, "Fast constant modulus adaptive algorithm," *IEE Proc.-F,* vol. 138, pp. 379-387, Aug. 1991.

[18] J. Benesty and P. Duhamel, "A fast exact least mean square adaptive algorithm," *IEEE Trans. Signal Processing,* vol. 40, no. 12, pp. 2904-2920, Dec. 1992.

[19] 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.

[20] J. Prado and E. Moulines, "Frequency-domain adaptive filtering with applications to acoustic echo cancellation," *Ann. Télécommun.,* vol. 49, pp. 414-428, 1994.

[21] J. Benesty, A. Gilloire, and Y. Grenier, "A frequency-domain stereophonic acoustic echo canceler exploiting the coherence between the channels," *J. Acoust. Soc. Am.*, vol. 106, pp. L30-L35, Sept. 1999.

[22] J. Benesty, F. Amand, A. Gilloire, and Y. Grenier, "Adaptive filtering algorithms for stereophonic acoustic echo cancellation," in *Proc. Int. Conf. on Acoustics, Speech, and Signal Processing*, pp. 3099-3102, May 1995.

[23] D.H. Brandwood, "A complex gradient operator and its application in adaptive array theory," *Proc. IEE,* vol. 130, Pts. F and H, pp. 11-16, Feb. 1983.

[24] J. Benesty and T. Gänsler, "A multichannel acoustic echo canceller double-talk detector based on a normalized cross-correlation matrix," in *Proc. Int. Workshop on Acoustic Echo and Noise Control*, Darmstadt, Germany, pp. 63-66, Sept. 2001.

[25] V.A. Morozov, *Regularization Methods for Ill-posed Problems*, CRC Press, Boca Raton, FL, 1993.

[26] G.H. Golub and C.F. Van Loan, *Matrix Computations*, 2nd ed., Johns Hopkins, Baltimore, MD, 1989.

[27] J.M. Cioffi, and T. Kailath, "Fast, recursive-least-squares transversal filters for adaptive filtering," *IEEE Trans. Acoustic Speech Signal Processing,* vol. ASSP-32, pp. 304-337.

[28] D.W.E. Schobben, G.P.M. Egelmeers, and P.C.W. Sommen, "Efficient Realization of the Block Frequency Domain Adaptive Filter," in *Proc. IEEE ICASSP*, Munich, Germany, pp. 2257–2260, April 1997.

[29] T. Gänsler and P. Eneroth, "Influence of audio coding on stereophonic acoustic echo cancellation," in *Proc. ICASSP,* pp. 3649-3652, May 1998.

[30] A. Sugiyama, Y. Joncour, and A. Hirano, "A stereo echo canceler with correct echo-path identification on an input-sliding technique," in *IEEE Trans Signal Processing,* vol. 49, no. 11, pp. 2577-2587, Nov. 2001.

[31] W. Kellermann, "Acoustic echo cancellation for beamforming microphone arrays," in M.S. Brandstein and D.B. Ward (eds.), *Microphone Arrays: Signal Processing Techniques and Applications,* Springer Verlag, 2001.

[32] L.J. Griffith and C.W. Jim, "An alternative approach to linearly constrained adaptive beamforming," *IEEE Trans Antennas Propagation,* vol. 30, no. 1, pp. 27-34, Jan. 1982.

[33] O. Hoshuyama, A. Sugiyama, and A. Hirano, "A robust adaptive beamformer for microphone arrays with a blocking matrix using constrained adaptive filters," *IEEE Trans on Signal Processing,* vol. 47, no. 10, pp. 2677-2684, 1999.