

Jacob Benesty
Arden (Yiteng) Huang

*Adaptive Signal Processing:
Application to Real-World Problems*

SPIN Springer's internal project number, if known

Springer

Berlin Heidelberg New York

Barcelona Budapest Hong Kong

London Milan Paris

Santa Clara Singapore Tokyo

Contents

4 Multichannel Frequency-Domain Adaptive Filtering with Application to Acoustic Echo Cancellation	95
<i>Herbert Buchner, Jacob Benesty, Walter Kellermann</i>	
4.1 Introduction	95
4.2 General Derivation of Multichannel Frequency-Domain Algorithms	98
4.2.1 Optimization Criterion	99
4.2.2 Normal Equation	102
4.2.3 Adaptation Algorithm	103
4.3 Convergence Analysis	106
4.3.1 Analysis Model	107
4.3.2 Convergence in the Mean	107
4.3.3 Convergence of the Mean-Squared Error	108
4.4 Generalized Frequency-Domain Adaptive MIMO Filtering	110
4.5 Approximation and Special Cases	113
4.5.1 Approximation of the Frequency-Domain Kalman Gain	113
4.5.2 Special Cases	114
4.6 A Dynamical Regularization Strategy	116
4.7 Efficient Multichannel Realization	117
4.7.1 Efficient Calculation of the Frequency-Domain Kalman Gain	117
4.7.2 Dynamical Regularization for Proposed Kalman Gain Approach	119
4.7.3 Efficient DFT calculation of overlapping data blocks	120
4.8 Simulations and Real-World Applications	122
4.8.1 Multichannel Acoustic Echo Cancellation	123
4.8.2 Adaptive MIMO Filtering for Hands-Free Speech Communication	125
4.9 Conclusions	127
References	127

4 Multichannel Frequency-Domain Adaptive Filtering with Application to Acoustic Echo Cancellation

Herbert Buchner¹, Jacob Benesty², and Walter Kellermann¹

¹ University of Erlangen-Nuremberg, Telecommunications Laboratory,
Cauerstr. 7,
D-91058 Erlangen, Germany
E-mail: {buchner, wk}@LNT.de

² Bell Laboratories, Lucent Technologies,
Murray Hill, NJ 07974, USA
E-mail: jbenesty@bell-labs.com

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 chapter, we focus on multichannel algorithms in the frequency domain that are especially well suited for input signals which are not only auto-correlated but also highly cross-correlated among the channels. These properties are particularly important for applications like multichannel acoustic echo cancellation. Most frequency-domain algorithms, as they are well known from the single-channel case, are derived from existing time-domain algorithms and are based on different heuristic strategies. Here, we present a new 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 formulate in different ways. An analysis of this multichannel algorithm shows that the mean-squared error convergence is independent of the input signal statistics. A useful approximation provides interesting links between some well-known algorithms for the single-channel case and the general framework. We also 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 to overlapping data blocks. Simulation results and real-time performance for multichannel acoustic echo cancellation show the high efficiency of the approach.

4.1 Introduction

The ability of adaptive filters to operate satisfactorily in an unknown environment and to 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.

In speech and acoustics, where all those basic types of adaptive filtering can be found, we often have to deal with very long filters (sometimes several thousand taps), unpredictably 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 chapter, we investigate adaptive MIMO (multiple input and multiple output) systems that are updated in the frequency domain. The resulting generalized multichannel frequency-domain adaptive filtering has already led to efficient real-time implementations of multichannel acoustic echo cancellers on standard personal computers [2,3].

Generally, we distinguish two classes of adaptive algorithms. One class includes filters that are updated in the time domain, usually on a sample-by-sample basis, like the classical least-mean-square (LMS) [4] and recursive least-squares (RLS) [5] algorithms. The other class may be defined as filters that are updated in the discrete Fourier transform (DFT) domain ('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 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 transversal structure of the adaptive filters. The Toeplitz matrices can be expressed by circulant matrices which are diagonalizable by the DFT. Consequently, the key for deriving the frequency-domain adaptive algorithms is to formulate the time-domain error criterion so that Toeplitz and circulant matrices are explicitly shown.

In addition to the low complexity, another advantage resulting from this diagonalization in frequency-domain adaptive filtering is that the adaptation stepsize can be normalized independently for each frequency bin, which results in a more uniform convergence over the entire frequency range.

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 [6]. Ferrara [7] was the first to present an efficient frequency-domain adaptive filter algorithm (FLMS) that converges to the optimum (Wiener) solution. Mansour and Gray [8] 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 [9]. However, in some applications, a major handicap with these structures is the delay in-

troduced between input and output. Indeed, for efficient implementations, 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 [10,11] and generalized in [12] where the new block size 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* (GMDF α) was proposed in [13,14], 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) and the delay can be (further) reduced as well. 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.

The case of multichannel adaptive filtering, as shown in Fig. 4.1, has been found to be structurally more difficult in general. In typical scenarios, the input signals $x_p(n)$, $p = 1, \dots, P$, to the adaptive filter are not only auto-correlated but also highly cross-correlated which often results in very slow convergence of the LP filter coefficients $\hat{h}_{p,\kappa}(n)$, where $\kappa = 0, \dots, L - 1$. This problem becomes particularly severe in multichannel acoustic echo cancellation [15–17], where the signals $x_p(n)$ represent loudspeaker signals that may originate from common sources. Signal $y(n)$ represents an echo received by a microphone.

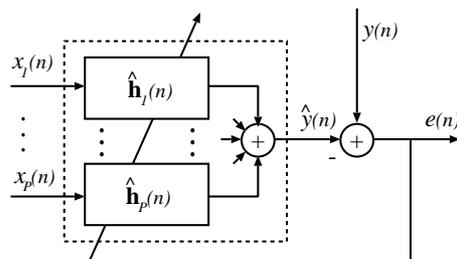


Fig. 4.1. Multichannel adaptive filtering.

Direct application of commonly used low-complexity algorithms, such as the LMS algorithm or conventional frequency-domain adaptive filtering, to the multichannel case usually leads to disappointing results as the cross-

correlations between the input channels are not taken into account [18]. 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 [18], but its complexity is prohibitively high and, e.g., will not allow real-time implementation of multichannel acoustic echo cancellation on standard hardware any time soon.

Two-channel frequency-domain adaptive filtering was first introduced in [19] in the context of stereophonic acoustic echo cancellation and derived from the extended least-mean-squares (ELMS) algorithm [20] in the time domain using similar considerations as for the single-channel case outlined above.

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.

The organization of this chapter is as follows. In Section 4.2, we introduce a frequency-domain recursive least-squares criterion from which the so-called normal equation is derived. Then, from the normal equation, we deduce a generic multichannel adaptive algorithm that we can formulate in different ways, and we introduce the so-called frequency-domain Kalman gain. In Section 4.3, we study the convergence of this multichannel algorithm. In Section 4.4, we consider the general MIMO case and, in Section 4.5, we give a very useful approximation, deduce some well-known single-channel algorithms as special cases, and explicitly show how the cross-correlations are taken into account in the multichannel case. We also 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 4.6. Section 4.7 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 4.8 presents some simulations and multichannel real-world implementations for hands-free speech communications. Finally, our results are summarized in Section 4.9.

4.2 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 multichannel algorithm follows immediately.

4.2.1 Optimization Criterion

To obtain an optimization criterion for block adaptation, we assume the (generally time-variant) adaptive filter coefficients $\hat{h}_{p,0}, \dots, \hat{h}_{p,L-1}$ for the input channels $1, \dots, P$ to be fixed within the block intervals of length N . For convenience of notation, this allows us to omit the time index of the filter coefficients during the following derivation of the block error signal.

From Fig. 4.1, it can be seen that the error signal at time n between the output of the multichannel adaptive filter $\hat{y}(n)$ and the desired output signal $y(n)$ is given by

$$e(n) = y(n) - \hat{y}(n), \quad (4.1)$$

with

$$\hat{y}(n) = \sum_{p=1}^P \sum_{\kappa=0}^{L-1} x_p(n - \kappa) \hat{h}_{p,\kappa}. \quad (4.2)$$

By partitioning the impulse responses \hat{h}_p into segments of length N as in [10,11], (4.2) can be written as

$$\hat{y}(n) = \sum_{p=1}^P \sum_{k=0}^{K-1} \sum_{\kappa=0}^{N-1} x_p(n - Nk - \kappa) \hat{h}_{p,Nk+\kappa}, \quad (4.3)$$

where we assume that the total filter length L is an integer multiple of N ($N \leq L$), so that $L = KN$.

For convenient notation of the multichannel algorithms, we rewrite (4.3) in vectorized form

$$\hat{y}(n) = \sum_{p=1}^P \sum_{k=0}^{K-1} \mathbf{x}_{p,k}^T(n) \hat{\mathbf{h}}_{p,k} = \sum_{p=1}^P \mathbf{x}_p^T(n) \hat{\mathbf{h}}_p = \mathbf{x}^T(n) \hat{\mathbf{h}}, \quad (4.4)$$

where

$$\mathbf{x}_{p,k}(n) = [x_p(n - Nk), x_p(n - Nk - 1), \dots, x_p(n - Nk - N + 1)]^T, \quad (4.5)$$

$$\hat{\mathbf{h}}_{p,k} = [\hat{h}_{p,Nk}, \hat{h}_{p,Nk+1}, \dots, \hat{h}_{p,Nk+N-1}]^T, \quad (4.6)$$

$$\mathbf{x}_p(n) = [\mathbf{x}_{p,0}^T(n), \mathbf{x}_{p,1}^T(n), \dots, \mathbf{x}_{p,K-1}^T(n)]^T, \quad (4.7)$$

$$\hat{\mathbf{h}}_p = [\hat{\mathbf{h}}_{p,0}^T, \hat{\mathbf{h}}_{p,1}^T, \dots, \hat{\mathbf{h}}_{p,K-1}^T]^T, \quad (4.8)$$

$$\mathbf{x}(n) = [\mathbf{x}_1^T(n), \mathbf{x}_2^T(n), \dots, \mathbf{x}_P^T(n)]^T, \quad (4.9)$$

$$\hat{\mathbf{h}} = [\hat{\mathbf{h}}_1^T, \hat{\mathbf{h}}_2^T, \dots, \hat{\mathbf{h}}_P^T]^T. \quad (4.10)$$

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

We now define the block error signal of length N . Based on (4.1) and (4.4) we write

$$\mathbf{e}(m) = \mathbf{y}(m) - \hat{\mathbf{y}}(m), \quad (4.11)$$

with m being the block time index, and

$$\hat{\mathbf{y}}(m) = \sum_{p=1}^P \sum_{k=0}^{K-1} \mathbf{U}_{p,k}^T(m) \hat{\mathbf{h}}_{p,k} = \sum_{p=1}^P \mathbf{U}_p^T(m) \hat{\mathbf{h}}_p = \mathbf{U}^T(m) \hat{\mathbf{h}}, \quad (4.12)$$

where

$$\mathbf{e}(m) = [e(mN), \dots, e(mN + N - 1)]^T, \quad (4.13)$$

$$\mathbf{y}(m) = [y(mN), \dots, y(mN + N - 1)]^T, \quad (4.14)$$

$$\hat{\mathbf{y}}(m) = [\hat{y}(mN), \dots, \hat{y}(mN + N - 1)]^T, \quad (4.15)$$

$$\mathbf{U}_{p,k}(m) = [\mathbf{x}_{p,k}(mN), \dots, \mathbf{x}_{p,k}(mN + N - 1)], \quad (4.16)$$

$$\mathbf{U}_p(m) = [\mathbf{U}_{p,0}^T(m), \dots, \mathbf{U}_{p,K-1}^T(m)]^T, \quad (4.17)$$

$$\mathbf{U}(m) = [\mathbf{U}_1^T(m), \dots, \mathbf{U}_P^T(m)]^T. \quad (4.18)$$

It can easily be verified that $\mathbf{U}_{p,k}$, $p = 1, \dots, P$, $k = 0, \dots, K - 1$ are Toeplitz matrices of size $(N \times N)$:

$$\mathbf{U}_{p,k}^T(m) = \begin{bmatrix} x_p(mN - Nk) & \cdots & \cdots & x_p(mN - Nk - N + 1) \\ x_p(mN - Nk + 1) & \ddots & & \vdots \\ \vdots & \ddots & \ddots & \vdots \\ x_p(mN - Nk + N - 1) & \cdots & \cdots & x_p(mN - Nk) \end{bmatrix}$$

These Toeplitz matrices are now diagonalized in two steps:

Step 1: Transformation of Toeplitz matrices into circulant matrices.

Any Toeplitz matrix $\mathbf{U}_{p,k}$ can be transformed, by doubling its size, to a circulant matrix

$$\mathbf{C}_{p,k}(m) = \begin{bmatrix} \mathbf{U}_{p,k}^T(m) & \mathbf{U}_{p,k}^T(m) \\ \mathbf{U}_{p,k}^T(m) & \mathbf{U}_{p,k}^T(m) \end{bmatrix}, \quad (4.19)$$

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

$$\mathbf{U}'_{p,k}(m) = \begin{bmatrix} x_p(mN - Nk - N) & \cdots & \cdots & x_p(mN - Nk + 1) \\ x_p(mN - Nk - N + 1) & \ddots & & \vdots \\ \vdots & \ddots & \ddots & \vdots \\ x_p(mN - Nk - 1) & \cdots & \cdots & x_p(mN - Nk - N) \end{bmatrix}.$$

It follows

$$\mathbf{U}_{p,k}^T(m) = \mathbf{W}_{N \times 2N}^{01} \mathbf{C}_{p,k}(m) \mathbf{W}_{2N \times N}^{10}, \quad (4.20)$$

where we introduced the windowing matrices

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

Step 2: Transformation of the circulant matrices into diagonal matrices. Using the $2N \times 2N$ DFT matrix $\mathbf{F}_{2N \times 2N}$ with elements $e^{-j2\pi\nu n/(2N)}$, where $\nu, n = 0, \dots, 2L - 1$, the circulant matrices are diagonalized as follows:

$$\mathbf{C}_{p,k}(m) = \mathbf{F}_{2N \times 2N}^{-1} \mathbf{X}_{p,k}(m) \mathbf{F}_{2N \times 2N}, \quad (4.21)$$

where the diagonal matrices $\mathbf{X}_{p,k}(m)$ can be expressed by the elements of the first columns of $\mathbf{C}_{p,k}(m)$,

$$\begin{aligned} \mathbf{X}_{p,k}(m) &= \\ &\text{diag}\{\mathbf{F}_{2N \times 2N} [x_p(mN - Nk - N), \dots, x_p(mN - Nk + N - 1)]^T\}. \end{aligned} \quad (4.22)$$

Now, (4.20) can be rewritten equivalently as

$$\mathbf{U}_{p,k}^T(m) = \mathbf{W}_{N \times 2N}^{01} \mathbf{F}_{2N \times 2N}^{-1} \mathbf{X}_{p,k}(m) \mathbf{F}_{2N \times 2N} \mathbf{W}_{2N \times N}^{10}. \quad (4.23)$$

Since

$$[\mathbf{A}\mathbf{X}_1\mathbf{B}, \dots, \mathbf{A}\mathbf{X}_P\mathbf{B}] = \mathbf{A}[\mathbf{X}_1, \dots, \mathbf{X}_P] \text{diag}\{\mathbf{B}, \dots, \mathbf{B}\}$$

for any matrices $\mathbf{A}, \mathbf{B}, \mathbf{X}_p$ with compatible dimensions, it follows for the error vector using (4.18) and (4.23):

$$\begin{aligned} \mathbf{e}(m) &= \mathbf{y}(m) - \mathbf{W}_{N \times 2N}^{01} \mathbf{F}_{2N \times 2N}^{-1} [\mathbf{X}_1(m), \dots, \mathbf{X}_P(m)] \\ &\quad \cdot \text{diag}\{\mathbf{F}_{2N \times 2N} \mathbf{W}_{2N \times N}^{10}, \dots, \mathbf{F}_{2N \times 2N} \mathbf{W}_{2N \times N}^{10}\} \hat{\mathbf{h}}, \end{aligned} \quad (4.24)$$

where

$$\mathbf{X}_p(m) = [\mathbf{X}_{p,0}(m), \mathbf{X}_{p,1}(m), \dots, \mathbf{X}_{p,K-1}(m)]. \quad (4.25)$$

If we multiply (4.24) by the $N \times N$ DFT matrix $\mathbf{F}_{N \times N}$, we obtain the error signal in the frequency domain:

$$\underline{\mathbf{e}}(m) = \underline{\mathbf{y}}(m) - \mathbf{G}_{N \times 2N}^{01} \mathbf{X}(m) \mathbf{G}_{2LP \times LP}^{10} \hat{\mathbf{h}}, \quad (4.26)$$

where

$$\underline{\mathbf{e}}(m) = \mathbf{F}_{N \times N} \mathbf{e}(m), \quad (4.27)$$

$$\underline{\mathbf{y}}(m) = \mathbf{F}_{N \times N} \mathbf{y}(m), \quad (4.28)$$

$$\mathbf{G}_{N \times 2N}^{01} = \mathbf{F}_{N \times N} \mathbf{W}_{N \times 2N}^{01} \mathbf{F}_{2N \times 2N}^{-1}, \quad (4.29)$$

$$\mathbf{G}_{2LP \times LP}^{10} = \text{diag}\{\mathbf{G}_{2N \times N}^{10}, \dots, \mathbf{G}_{2N \times N}^{10}\}, \quad (4.30)$$

$$\mathbf{G}_{2N \times 2N}^{10} = \mathbf{F}_{2N \times 2N} \mathbf{W}_{2N \times 2N}^{10} \mathbf{F}_{N \times N}^{-1}, \quad (4.31)$$

$$\mathbf{X}(m) = [\mathbf{X}_1(m), \mathbf{X}_2(m), \dots, \mathbf{X}_P(m)], \quad (4.32)$$

$$\hat{\mathbf{h}}_{p,k} = \mathbf{F}_{N \times N} \hat{\mathbf{h}}_{p,k}, \quad (4.33)$$

$$\hat{\mathbf{h}}_p = [\hat{\mathbf{h}}_{p,0}^T, \hat{\mathbf{h}}_{p,1}^T, \dots, \hat{\mathbf{h}}_{p,K-1}^T]^T, \quad (4.34)$$

$$\hat{\mathbf{h}} = [\hat{\mathbf{h}}_1^T, \hat{\mathbf{h}}_2^T, \dots, \hat{\mathbf{h}}_P^T]^T. \quad (4.35)$$

Optimization Criterion:

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

$$J_f(m) = (1 - \lambda) \sum_{i=0}^m \lambda^{m-i} \underline{\mathbf{e}}^H(i) \underline{\mathbf{e}}(i), \quad (4.36)$$

where H denotes conjugate transposition and λ ($0 < \lambda < 1$) is an exponential forgetting factor. The criterion (4.36) is very similar¹ to the one leading to the well-known RLS algorithm [5]. The main advantage of using (4.36) is to take advantage of the FFT in order to have low-complexity adaptive filters.

4.2.2 Normal Equation

Let $\nabla_{\hat{\mathbf{h}}}$ be the gradient operator with respect to $\hat{\mathbf{h}}$. Applying the operator $\nabla_{\hat{\mathbf{h}}}$ to the cost function J_f (4.36), we obtain [1,21] the complex gradient vector:

$$\begin{aligned} \nabla_{\hat{\mathbf{h}}} J_f(m) &= 2 \frac{\partial J_f(m)}{\partial \hat{\mathbf{h}}^*(m)} \quad (4.37) \\ &= -2(1 - \lambda) \sum_{i=0}^m \lambda^{m-i} (\mathbf{G}_{2LP \times LP}^{10})^H \mathbf{X}^H(i) (\mathbf{G}_{N \times 2N}^{01})^H \underline{\mathbf{y}}(i) \\ &\quad + 2(1 - \lambda) \left[\sum_{i=0}^m \lambda^{m-i} (\mathbf{G}_{2LP \times LP}^{10})^H \mathbf{X}^H(i) \right. \\ &\quad \left. \cdot \mathbf{G}_{2N \times 2N}^{01} \mathbf{X}(i) \mathbf{G}_{2LP \times LP}^{10} \right] \hat{\mathbf{h}}(m), \end{aligned}$$

where $*$ denotes complex conjugation,

$$\begin{aligned} \mathbf{G}_{2N \times 2N}^{01} &= (\mathbf{G}_{N \times 2N}^{01})^H \mathbf{G}_{N \times 2N}^{01} \\ &= \mathbf{F}_{2N \times 2N} \mathbf{W}_{2N \times 2N}^{01} \mathbf{F}_{2N \times 2N}^{-1}, \end{aligned} \quad (4.38)$$

and

$$\mathbf{W}_{2N \times 2N}^{01} = \begin{bmatrix} \mathbf{0}_{N \times N} & \mathbf{0}_{N \times N} \\ \mathbf{0}_{N \times N} & \mathbf{I}_{N \times N} \end{bmatrix}. \quad (4.39)$$

¹ Note that the time-frequency equivalence is assured by Parseval's theorem.

By setting the gradient of the cost function equal to zero and defining

$$\begin{aligned}\underline{\mathbf{y}}_{2N}(m) &= (\mathbf{G}_{N \times 2N}^{01})^H \underline{\mathbf{y}}(m) \\ &= \mathbf{F}_{2N \times 2N} \begin{bmatrix} \mathbf{0}_{N \times 1} \\ \mathbf{y}(m) \end{bmatrix},\end{aligned}\quad (4.40)$$

we obtain the so-called *normal equation*:

$$\mathbf{S}(m) \hat{\underline{\mathbf{h}}}(m) = \mathbf{s}(m), \quad (4.41)$$

where

$$\begin{aligned}\mathbf{S}(m) &= (1 - \lambda) \sum_{i=0}^m \lambda^{m-i} (\mathbf{G}_{2LP \times LP}^{10})^H \mathbf{X}^H(i) \mathbf{G}_{2N \times 2N}^{01} \mathbf{X}(i) \mathbf{G}_{2LP \times LP}^{10} \\ &= \lambda \mathbf{S}(m-1) + (1 - \lambda) (\mathbf{G}_{2LP \times LP}^{10})^H \mathbf{X}^H(m) \\ &\quad \cdot \mathbf{G}_{2N \times 2N}^{01} \mathbf{X}(m) \mathbf{G}_{2LP \times LP}^{10}\end{aligned}\quad (4.42)$$

and

$$\begin{aligned}\mathbf{s}(m) &= (1 - \lambda) \sum_{i=0}^m \lambda^{m-i} (\mathbf{G}_{2LP \times LP}^{10})^H \mathbf{X}^H(i) \underline{\mathbf{y}}_{2N}(i) \\ &= \lambda \mathbf{s}(m-1) + (1 - \lambda) (\mathbf{G}_{2LP \times LP}^{10})^H \mathbf{X}^H(m) \underline{\mathbf{y}}_{2N}(m) \\ &= \lambda \mathbf{s}(m-1) + (1 - \lambda) (\mathbf{G}_{2LP \times LP}^{10})^H \mathbf{X}^H(m) (\mathbf{G}_{N \times 2N}^{01})^H \underline{\mathbf{y}}(m).\end{aligned}\quad (4.43)$$

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

4.2.3 Adaptation Algorithm

The different formulations for filter adaptation discussed below, i.e., recursive updates of $\hat{\underline{\mathbf{h}}}(m)$, are all derived directly from the normal equation (4.41) and associated equations (4.42) and (4.43).

Here, we replace $\mathbf{s}(m)$ and $\mathbf{s}(m-1)$ in the recursive equation (4.43) by formulating (4.41) in terms of block time indices m and $m-1$, respectively. We then eliminate $\mathbf{S}(m-1)$ from the resulting equation using (4.42). Reintroducing the error signal vector (4.26), we obtain an exact recursive solution of (4.41) by the following adaptation algorithm:

$$\underline{\mathbf{e}}(m) = \underline{\mathbf{y}}(m) - \mathbf{G}_{N \times 2N}^{01} \mathbf{X}(m) \mathbf{G}_{2LP \times LP}^{10} \hat{\underline{\mathbf{h}}}(m-1) \quad (4.44)$$

$$\begin{aligned}\hat{\underline{\mathbf{h}}}(m) &= \hat{\underline{\mathbf{h}}}(m-1) + (1 - \lambda) \mathbf{S}^{-1}(m) (\mathbf{G}_{2LP \times LP}^{10})^H \\ &\quad \cdot \mathbf{X}^H(m) (\mathbf{G}_{N \times 2N}^{01})^H \underline{\mathbf{e}}(m).\end{aligned}\quad (4.45)$$

For practical purposes, it is useful to reformulate this algorithm. First, we multiply (4.44) by $(\mathbf{G}_{N \times 2N}^{01})^H$,

$$\underline{\mathbf{e}}_{2N}(m) = \underline{\mathbf{y}}_{2N}(m) - \mathbf{G}_{2N \times 2N}^{01} \mathbf{X}(m) \mathbf{G}_{2LP \times LP}^{10} \hat{\mathbf{h}}(m-1) \quad (4.46)$$

$$\begin{aligned} \hat{\mathbf{h}}(m) &= \hat{\mathbf{h}}(m-1) + (1-\lambda) \mathbf{S}^{-1}(m) (\mathbf{G}_{2LP \times LP}^{10})^H \\ &\quad \cdot \mathbf{X}^H(m) \underline{\mathbf{e}}_{2N}(m), \end{aligned} \quad (4.47)$$

where we defined analogously to (4.40)

$$\begin{aligned} \underline{\mathbf{e}}_{2N}(m) &= (\mathbf{G}_{N \times 2N}^{01})^H \underline{\mathbf{e}}(m) \\ &= \mathbf{F}_{2N \times 2N} \begin{bmatrix} \mathbf{0}_{N \times 1} \\ \mathbf{e}(m) \end{bmatrix}. \end{aligned} \quad (4.48)$$

If we multiply (4.47) by $\mathbf{G}_{2LP \times LP}^{10}$, we obtain the algorithm (4.46) and (4.47) in a slightly different form:

$$\underline{\mathbf{e}}_{2N}(m) = \underline{\mathbf{y}}_{2N}(m) - \mathbf{G}_{2N \times 2N}^{01} \mathbf{X}(m) \hat{\mathbf{h}}_{2LP}(m-1) \quad (4.49)$$

$$\begin{aligned} \hat{\mathbf{h}}_{2LP}(m) &= \hat{\mathbf{h}}_{2LP}(m-1) + (1-\lambda) \mathbf{G}_{2LP \times LP}^{10} \\ &\quad \cdot \mathbf{S}^{-1}(m) (\mathbf{G}_{2LP \times LP}^{10})^H \mathbf{X}^H(m) \underline{\mathbf{e}}_{2N}(m), \end{aligned} \quad (4.50)$$

where $\mathbf{S}(m)$ is given by (4.42), and

$$\begin{aligned} \hat{\mathbf{h}}_{2LP}(m) &= \mathbf{G}_{2LP \times LP}^{10} \hat{\mathbf{h}}(m) \\ &= \left[\hat{\mathbf{h}}_{2NP,1,0}^T(m), \dots, \hat{\mathbf{h}}_{2NP,P,K-1}^T(m) \right]^T, \\ \hat{\mathbf{h}}_{2NP,p,k}(m) &= \mathbf{F}_{2N \times 2N} \begin{bmatrix} \hat{\mathbf{h}}_{p,k}(m) \\ \mathbf{0}_{N \times 1} \end{bmatrix}. \end{aligned} \quad (4.51)$$

The rank of the matrix $\mathbf{G}_{2LP \times LP}^{10}$ is equal to LP . Since we have to adapt LP unknowns, in principle, (4.50) is equivalent to (4.47). Indeed, if we multiply (4.50) by $(\mathbf{G}_{2LP \times LP}^{10})^H$, we obtain exactly (4.47) since $(\mathbf{G}_{2LP \times LP}^{10})^H \mathbf{G}_{2LP \times LP}^{10} = \mathbf{I}_{LP \times LP}$. It is interesting to see how naturally we have ended up using blocks of length $2N$ (especially for the error signal) even though we have used an error criterion with blocks of length N . We can do even better than that and rewrite the algorithm exclusively using FFTs of size $2N$. 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

$$\begin{aligned} \mathbf{S}_d(m) &= (1-\lambda) \sum_{i=0}^m \lambda^{m-i} \mathbf{X}^H(i) \mathbf{G}_{2N \times 2N}^{01} \mathbf{X}(i) \\ &= \lambda \mathbf{S}_d(m-1) + (1-\lambda) \mathbf{X}^H(m) \mathbf{G}_{2N \times 2N}^{01} \mathbf{X}(m). \end{aligned} \quad (4.52)$$

The relation of $\mathbf{S}_d(m)$ to $\mathbf{S}(m)$ is obviously given by:

$$\mathbf{S}(m) = (\mathbf{G}_{2LP \times LP}^{10})^H \mathbf{S}_d(m) \mathbf{G}_{2LP \times LP}^{10}. \quad (4.53)$$

Next, we define

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

and

$$\mathbf{G}_{2LP \times 2LP}^{10} = \text{diag}\{\mathbf{G}_{2N \times 2N}^{10} \cdots \mathbf{G}_{2N \times 2N}^{10}\}, \quad (4.54)$$

where

$$\mathbf{W}_{2N \times 2N}^{10} = \begin{bmatrix} \mathbf{I}_{N \times N} & \mathbf{0}_{N \times N} \\ \mathbf{0}_{N \times N} & \mathbf{0}_{N \times N} \end{bmatrix}. \quad (4.55)$$

Now, we have a relation between the inverse of the two matrices \mathbf{S} (as it appears in (4.50)) and \mathbf{S}_d :

$$\mathbf{G}_{2LP \times 2LP}^{10} \mathbf{S}_d^{-1}(m) = \mathbf{G}_{2LP \times LP}^{10} \mathbf{S}^{-1}(m) (\mathbf{G}_{2LP \times LP}^{10})^H. \quad (4.56)$$

This can be verified by post-multiplying both sides of (4.56) by $\mathbf{S}_d(m) \mathbf{G}_{2LP \times LP}^{10}$ and noting that $\mathbf{G}_{2LP \times 2LP}^{10} \mathbf{G}_{2LP \times LP}^{10} = \mathbf{G}_{2LP \times LP}^{10}$. Using (4.56), the adaptive algorithm consisting of (4.42), (4.49), and (4.50) can now be formulated more conveniently:

$$\mathbf{S}_d(m) = \lambda \mathbf{S}_d(m-1) + (1-\lambda) \mathbf{X}^H(m) \mathbf{G}_{2N \times 2N}^{01} \mathbf{X}(m) \quad (4.57)$$

$$\underline{\mathbf{e}}_{2N}(m) = \underline{\mathbf{y}}_{2N}(m) - \mathbf{G}_{2N \times 2N}^{01} \mathbf{X}(m) \hat{\underline{\mathbf{h}}}_{2LP}(m-1) \quad (4.58)$$

$$\begin{aligned} \hat{\underline{\mathbf{h}}}_{2LP}(m) &= \hat{\underline{\mathbf{h}}}_{2LP}(m-1) + (1-\lambda) \mathbf{G}_{2LP \times 2LP}^{10} \mathbf{S}_d^{-1}(m) \\ &\quad \cdot \mathbf{X}^H(m) \underline{\mathbf{e}}_{2N}(m). \end{aligned} \quad (4.59)$$

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

$$\mathbf{K}(m) = (1-\lambda) \mathbf{S}_d^{-1}(m) \mathbf{X}^H(m). \quad (4.60)$$

This $2LP \times 2L$ matrix includes the inverse in (4.59) and plays a very important role in practical realizations, including a tight coupling between the multiple input channels by coherence terms, as shown in detail in subsequent sections. Figure 4.2 summarizes the general steps in multichannel frequency-domain adaptive filtering. For clarity of the figure, the case $N = L$ is depicted. The two shaded blocks represent the calculation of the Kalman gain using (4.57) and (4.60), or efficient realizations thereof.

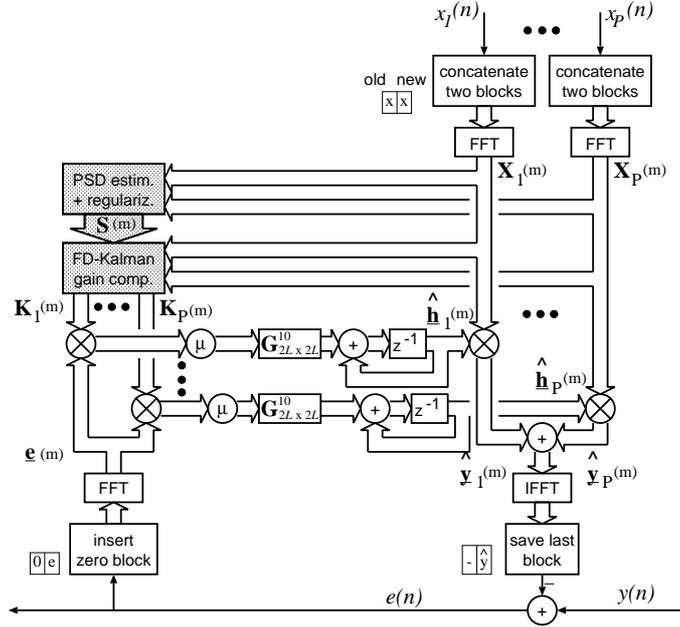


Fig. 4.2. Principle of multichannel frequency-domain adaptive filtering ($N = L$).

4.3 Convergence Analysis

In this section, we analyze the convergence behaviour of the algorithm for stationary signals $x_p(n)$ and $y(n)$ based on (4.44) and (4.45).

Due to the assumed stationarity of the filter input signals, we obtain, after taking the expected value of (4.42):

$$E\{\mathbf{S}(m)\} = (1 - \lambda) \sum_{i=0}^m \lambda^{m-i} \mathbf{S}_e, \quad (4.61)$$

where

$$\mathbf{S}_e = E\{(\mathbf{G}_{2LP \times LP}^{10})^H \mathbf{X}^H(m) \mathbf{G}_{2N \times 2N}^{01} \mathbf{X}(m) \mathbf{G}_{2LP \times LP}^{10}\} \quad (4.62)$$

denotes the time-independent ensemble average. Noting that in (4.61) we have a sum of a finite geometric series, it can be simplified to

$$E\{\mathbf{S}(m)\} = (1 - \lambda^{m+1}) \mathbf{S}_e. \quad (4.63)$$

For a single realization of the stochastic process $\mathbf{S}(m)$, we assume that

$$\mathbf{S}(m) \approx (1 - \lambda^{m+1}) \mathbf{S}_e, \quad (4.64)$$

and for the steady state we see with $0 < \lambda < 1$ that

$$\mathbf{S}(m) \approx \mathbf{S}_e \text{ for large } m. \quad (4.65)$$

4.3.1 Analysis Model

For the following, we assume that the desired response $y(n)$ and the tap-input vector $x(n)$ are related by the multiple linear regression model [1]

$$y(n) = \mathbf{x}^T(n)\mathbf{h} + n_O(n), \quad (4.66)$$

where the $LP \times 1$ vector \mathbf{h} denotes the fixed regression parameter vector of the model and the measurement error $n_O(n)$ is assumed to be a zero-mean white noise that is independent of $x_p(n) \forall p \in \{1, \dots, P\}$. The equivalent expression in the frequency domain reads

$$\underline{\mathbf{y}}(m) = \mathbf{G}_{N \times 2N}^{01} \mathbf{X}(m) \mathbf{G}_{2LP \times LP}^{10} \underline{\mathbf{h}} + \underline{\mathbf{n}}(m), \quad (4.67)$$

where $\underline{\mathbf{h}}$ and $\underline{\mathbf{n}}(m)$ are defined in the same way as $\hat{\underline{\mathbf{h}}}$ in (4.35) and $\underline{\mathbf{y}}(m)$ in (4.28), respectively.

4.3.2 Convergence in the Mean

By noting that

$$(\mathbf{G}_{N \times 2N}^{01})^H \mathbf{G}_{N \times 2N}^{01} = \mathbf{G}_{2N \times 2N}^{01} \quad (4.68)$$

from (4.38), the coefficient update (4.45) can be written in terms of the misalignment vector $\underline{\boldsymbol{\epsilon}}(m)$ as

$$\begin{aligned} \underline{\boldsymbol{\epsilon}}(m) &= \underline{\mathbf{h}} - \hat{\underline{\mathbf{h}}}(m) \\ &= \underline{\mathbf{h}} - \hat{\underline{\mathbf{h}}}(m-1) \\ &\quad - (1-\lambda)\mathbf{S}^{-1}(m)(\mathbf{G}_{2LP \times LP}^{10})^H \mathbf{X}^H(m) \mathbf{G}_{2N \times 2N}^{01} \mathbf{X}(m) \\ &\quad \cdot \mathbf{G}_{2LP \times LP}^{10} [\underline{\mathbf{h}} - \hat{\underline{\mathbf{h}}}(m-1)] \\ &\quad - (1-\lambda)\mathbf{S}^{-1}(m)(\mathbf{G}_{2LP \times LP}^{10})^H \mathbf{X}^H(m) \underline{\mathbf{n}}(m). \end{aligned} \quad (4.69)$$

Taking the mathematical expectation of expression (4.69), using the independence theory [1], and (4.62) together with (4.65), we deduce for large m that

$$E\{\underline{\boldsymbol{\epsilon}}(m)\} = \lambda E\{\underline{\boldsymbol{\epsilon}}(m-1)\} = \lambda^m E\{\underline{\boldsymbol{\epsilon}}(0)\}. \quad (4.70)$$

Equation (4.70) expresses that the convergence rate of the algorithm is governed by λ . Most importantly, the rate of convergence is completely independent of the input statistics (even in the multichannel case). Finally, we have

$$\lim_{m \rightarrow \infty} E\{\underline{\boldsymbol{\epsilon}}(m)\} = \mathbf{0}_{LP \times 1} \Rightarrow \lim_{m \rightarrow \infty} E\{\hat{\underline{\mathbf{h}}}(m)\} = \underline{\mathbf{h}}. \quad (4.71)$$

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 $\lambda = \lambda_t^N$. For example, a typical choice for the RLS algorithm [1] is $\lambda_t = 1 - 1/(3L)$. In this case, a good choice for the frequency-domain algorithm is $\lambda = [1 - 1/(3L)]^N$.

4.3.3 Convergence of the Mean-Squared Error

The convergence of the algorithm in the mean is not sufficient for convergence to the minimum mean-squared error (MMSE) estimate [1] as it only assures a bias-free estimate $\hat{\mathbf{h}}(m)$. The algorithm converges in the mean square if

$$\lim_{m \rightarrow \infty} J'_f(m) = J'_{f,\min} < \infty, \quad (4.72)$$

where

$$J'_f(m) = \frac{1}{N} E \{ \mathbf{e}^H(m) \mathbf{e}(m) \}. \quad (4.73)$$

From (4.44), the error signal $\mathbf{e}(m)$ can be written in terms of $\underline{\mathbf{e}}(m)$ as

$$\mathbf{e}(m) = \mathbf{G}_{N \times 2N}^{01} \mathbf{X}(m) \mathbf{G}_{2LP \times LP}^{10} \underline{\mathbf{e}}(m-1) + \mathbf{n}(m). \quad (4.74)$$

Expression (4.73) becomes

$$J'_f(m) = \frac{1}{N} J_{\text{ex}}(m) + \sigma_n^2, \quad (4.75)$$

where the excess mean-square error is given by

$$J_{\text{ex}}(m) = E \{ \underline{\mathbf{e}}^H(m-1) (\mathbf{G}_{2LP \times LP}^{10})^H \mathbf{X}^H(m) \mathbf{G}_{2N \times 2N}^{01} \cdot \mathbf{X}(m) \mathbf{G}_{2LP \times LP}^{10} \underline{\mathbf{e}}(m-1) \} \quad (4.76)$$

and σ_n^2 is the variance of the noise signal $n_O(n)$. Furthermore, (4.76) can be written as

$$\begin{aligned} J_{\text{ex}}(m) &= E \{ \text{tr} [\underline{\mathbf{e}}^H(m-1) (\mathbf{G}_{2LP \times LP}^{10})^H \mathbf{X}^H(m) \\ &\quad \cdot \mathbf{G}_{2N \times 2N}^{01} \mathbf{X}(m) \mathbf{G}_{2LP \times LP}^{10} \underline{\mathbf{e}}(m-1)] \} \\ &= E \{ \text{tr} [(\mathbf{G}_{2LP \times LP}^{10})^H \mathbf{X}^H(m) \mathbf{G}_{2N \times 2N}^{01} \mathbf{X}(m) \\ &\quad \cdot \mathbf{G}_{2LP \times LP}^{10} \underline{\mathbf{e}}(m-1) \underline{\mathbf{e}}^H(m-1)] \} \\ &= \text{tr} [E \{ (\mathbf{G}_{2LP \times LP}^{10})^H \mathbf{X}^H(m) \mathbf{G}_{2N \times 2N}^{01} \mathbf{X}(m) \\ &\quad \cdot \mathbf{G}_{2LP \times LP}^{10} \underline{\mathbf{e}}(m-1) \underline{\mathbf{e}}^H(m-1) \}]. \end{aligned}$$

Invoking the independence assumption and using (4.62), we may reduce this expectation to

$$J_{\text{ex}}(m) = \text{tr}[\mathbf{S}_e \mathbf{M}(m-1)], \quad (4.77)$$

where

$$\mathbf{M}(m) = E \{ \underline{\mathbf{e}}(m) \underline{\mathbf{e}}^H(m) \} \quad (4.78)$$

is the misalignment correlation matrix.

We derive an expression for the misalignment vector $\underline{\epsilon}(m)$ using the normal equation (4.41), and (4.43):

$$\begin{aligned}\underline{\epsilon}(m) &= \underline{\mathbf{h}} - \hat{\underline{\mathbf{h}}}(m) \\ &= \underline{\mathbf{h}} - \mathbf{S}^{-1}(m)\mathbf{s}(m) \\ &= \underline{\mathbf{h}} - (1 - \lambda)\mathbf{S}^{-1}(m) \sum_{i=0}^m \lambda^{m-i} (\mathbf{G}_{2LP \times LP}^{10})^H \\ &\quad \cdot \mathbf{X}^H(i) (\mathbf{G}_{N \times 2N}^{01})^H \underline{\mathbf{y}}(i).\end{aligned}\quad (4.79)$$

Using $\underline{\mathbf{y}}(m)$ from the model (4.67), we obtain with (4.68) and (4.42):

$$\begin{aligned}\underline{\epsilon}(m) &= -(1 - \lambda)\mathbf{S}^{-1}(m) \sum_{i=0}^m \lambda^{m-i} (\mathbf{G}_{2LP \times LP}^{10})^H \mathbf{X}^H(i) \\ &\quad \cdot (\mathbf{G}_{2N \times 2N}^{01})^H \underline{\mathbf{u}}(i).\end{aligned}\quad (4.80)$$

If we plug this equation into (4.78), we obtain, after taking the expectations, and noting that for a given input sequence, the only random variable is the white measurement noise $\underline{\mathbf{u}}(m)$:

$$\begin{aligned}\mathbf{M}(m) &= \sigma_n^2 (1 - \lambda)^2 \mathbf{S}^{-1}(m) \left[\sum_{i=0}^m \lambda^{2(m-i)} (\mathbf{G}_{2LP \times LP}^{10})^H \right. \\ &\quad \left. \cdot \mathbf{X}^H(i) \mathbf{G}_{2N \times 2N}^{01} \mathbf{X}(i) \mathbf{G}_{2LP \times LP}^{10} \right] \mathbf{S}^{-1}(m),\end{aligned}\quad (4.81)$$

where $E\{\underline{\mathbf{u}}(m)\underline{\mathbf{u}}^H(m)\} = \sigma_n^2 \mathbf{I}$ was used. Analogously to (4.64), we find for the term in brackets in (4.81):

$$\begin{aligned}\sum_{i=0}^m \lambda^{2(m-i)} (\mathbf{G}_{2LP \times LP}^{10})^H \mathbf{X}^H(i) \mathbf{G}_{2N \times 2N}^{01} \mathbf{X}(i) \mathbf{G}_{2LP \times LP}^{10} \\ \approx (1 - \lambda^{2(m+1)}) \mathbf{S}_e.\end{aligned}\quad (4.82)$$

Assuming strict equality in (4.82), using (4.64), and $1 - \lambda^{2(m+1)} = (1 - \lambda^{m+1})(1 + \lambda^{m+1})$, this leads to

$$\mathbf{M}(m) = \sigma_n^2 (1 - \lambda)^2 \frac{1 + \lambda^{m+1}}{1 - \lambda^{m+1}} \mathbf{S}_e^{-1}.\quad (4.83)$$

Finally, we obtain for (4.75) with (4.77)

$$J'_f(m) = \left[\frac{LP}{N} (1 - \lambda)^2 \frac{1 + \lambda^m}{1 - \lambda^m} + 1 \right] \sigma_n^2.\quad (4.84)$$

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 (4.72):

$$J'_f(m \rightarrow \infty) = J'_{f,\min} = \left[\frac{LP}{N} (1 - \lambda)^2 + 1 \right] \sigma_n^2.\quad (4.85)$$

Moreover, we see from (4.84) that the convergence behaviour of the mean-squared error is independent of the eigenvalues of the ensemble-averaged matrix \mathbf{S}_e . The scalar

$$J_{\text{mis}}(m) = E \{ \underline{\boldsymbol{\epsilon}}^H(m) \underline{\boldsymbol{\epsilon}}(m) \} \quad (4.86)$$

describes the convergence of the misalignment, i.e. the coefficient convergence. Using (4.83), we deduce that

$$\begin{aligned} J_{\text{mis}}(m) &= \text{tr}[\mathbf{M}(m)] \\ &= \sigma_n^2 (1 - \lambda)^2 \frac{1 + \lambda^{m+1}}{1 - \lambda^{m+1}} \text{tr}[\mathbf{S}_e^{-1}] \\ &= \sigma_n^2 (1 - \lambda)^2 \frac{1 + \lambda^{m+1}}{1 - \lambda^{m+1}} \sum_{i=0}^{LP-1} \frac{1}{\lambda_{s,i}}, \end{aligned} \quad (4.87)$$

where the $\lambda_{s,i}$ denote the eigenvalues of the ensemble-averaged matrix \mathbf{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 \mathbf{S}_e (i.e., it is also independent of the channel coherence), the misalignment is magnified by the inverse of the smallest eigenvalue $\lambda_{s,\min}$ of \mathbf{S}_e (and thus of $\mathbf{S}(m)$). 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 $\mathbf{S}(m)$. In Section 4.6, this important topic is discussed in more detail.

4.4 Generalized Frequency-Domain Adaptive MIMO Filtering

In this section, we consider the extension of the algorithm proposed in Section 4.2 to the general MIMO case, i.e., we have P input signals $x_p(n)$, $p = 1, \dots, P$, and Q desired signals $y_q(n)$, output signals $\hat{y}_q(n)$, and error signals $e_q(n)$, $q = 1, \dots, Q$, respectively (Fig. 4.3). In the sequel, the follow-

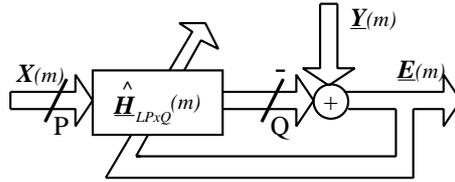


Fig. 4.3. Adaptive MIMO filtering in the frequency domain.

ing questions are discussed: What is the optimum solution? Can correlation

between the error signals $e_q(n)$ be exploited and how do the resulting update equations look like?

Let us define signal block vectors $\mathbf{y}_q(m)$, $\mathbf{e}_q(m)$, $\underline{\mathbf{y}}_q(m)$, $\underline{\mathbf{e}}_q(m)$ for each output channel in the same way as shown in (4.14), (4.13), (4.28), and (4.27), respectively. These quantities can be combined in the $(N \times Q)$ matrices

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

We consider three conceivable generalizations of the recursive least-squares error criterion proposed in (4.36):

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:

$$J_{f1,q}(m) = (1 - \lambda) \sum_{i=0}^m \lambda^{m-i} \underline{\mathbf{e}}_q^H(i) \underline{\mathbf{e}}_q(i) \quad (4.88)$$

for $q = 1, \dots, Q$. This criterion has been traditionally used in all approaches for multichannel echo cancellation which is a system identification problem.

Error criterion 2: Joint optimization

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

$$\begin{aligned}J_{f2}(m) &= \sum_{q=1}^Q J_{f1,q}(m) \\ &= (1 - \lambda) \sum_{i=0}^m \lambda^{m-i} \sum_{q=1}^Q \underline{\mathbf{e}}_q^H(i) \underline{\mathbf{e}}_q(i) \\ &= (1 - \lambda) \sum_{i=0}^m \lambda^{m-i} \text{tr}[\underline{\mathbf{E}}^H(i) \underline{\mathbf{E}}(i)] \\ &= (1 - \lambda) \sum_{i=0}^m \lambda^{m-i} \|\text{diag}\{\underline{\mathbf{E}}^H(i) \underline{\mathbf{E}}(i)\}\|_1,\end{aligned} \quad (4.89)$$

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

based on the subfilter coefficient vectors $\hat{\mathbf{h}}_{p,k,q}$ (p, k, q denote input channel, partition, and output channel, respectively),

$$\hat{\mathbf{H}}_{LP \times Q} = \begin{bmatrix} \hat{\mathbf{h}}_{1,0,1} & \cdots & \hat{\mathbf{h}}_{1,0,Q} \\ \hat{\mathbf{h}}_{1,1,1} & \cdots & \hat{\mathbf{h}}_{1,1,Q} \\ \vdots & \ddots & \vdots \\ \hat{\mathbf{h}}_{P,K-1,1} & \cdots & \hat{\mathbf{h}}_{P,K-1,Q} \end{bmatrix}, \quad (4.90)$$

and using the same approach as in Section 4.2, we obtain the following normal equation:

$$\mathbf{S}(m) \hat{\mathbf{H}}_{LP \times Q} = \mathbf{s}_{LP \times Q}(m). \quad (4.91)$$

Fortunately, this matrix equation can be easily decomposed into Q equations (4.41). Therefore, criteria 1 and 2 are strictly equivalent for the behaviour of the adaptation. We note, however, that the compact formulation (4.91) of the normal equation can be used, e.g., to obtain a generalized control of the adaptation for the echo cancellation application [22].

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

The last formulation of Criterion 2 (4.89) reveals an interesting possibility to take the cross-correlations between the error signals into account by optimizing

$$J_{f3}(m) = (1 - \lambda) \sum_{i=0}^m \lambda^{m-i} \|\underline{\mathbf{E}}^H(i) \underline{\mathbf{E}}(i)\|_1. \quad (4.92)$$

Let us consider the optimization of the additional off-diagonal elements $\mathbf{e}_q^H(i) \mathbf{e}_r(i)$ ($q \neq r$) of $\underline{\mathbf{E}}^H(i) \underline{\mathbf{E}}(i)$. According to [1,21], we obtain

$$\frac{\partial}{\partial \hat{\mathbf{h}}_q(i)} \mathbf{e}_q^H(i) \mathbf{e}_r(i) = 0, \quad (4.93)$$

and from

$$\frac{\partial}{\partial \hat{\mathbf{h}}_r(i)} \mathbf{e}_q^H(i) \mathbf{e}_r(i), \quad (4.94)$$

we obtain the well-known normal equations (4.41) for $\hat{\mathbf{h}}_q$.

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

$$\mathbf{S}_d(m) = \lambda \mathbf{S}_d(m-1) + (1 - \lambda) \mathbf{X}^H(m) \mathbf{G}_{2N \times 2N}^{01} \mathbf{X}(m) \quad (4.95)$$

$$\mathbf{K}(m) = (1 - \lambda) \mathbf{S}_d^{-1}(m) \mathbf{X}^H(m) \quad (4.96)$$

$$\underline{\mathbf{E}}_{2N \times Q}(m) = \underline{\mathbf{Y}}_{2N \times Q}(m) - \mathbf{G}_{2N \times 2N}^{01} \mathbf{X}(m) \hat{\mathbf{H}}_{2LP \times Q}(m-1) \quad (4.97)$$

$$\hat{\mathbf{H}}_{2LP \times Q}(m) = \hat{\mathbf{H}}_{2LP \times Q}(m-1) + \mathbf{G}_{2LP \times 2LP}^{10} \mathbf{K}(m) \underline{\mathbf{E}}_{2N \times Q}(m) \quad (4.98)$$

in analogy to equations (4.57) to (4.60).

4.5 Approximation and Special Cases

We start this section by giving a very useful approximation of the algorithm proposed in the preceding Section. This allows us both, to show explicitly the links to the classical single-channel algorithms, and also to derive new and very efficient multichannel algorithms. The list of special cases of the framework is not exhaustive and several other algorithms may also be derived.

4.5.1 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 [7]. Unfortunately, the matrix \mathbf{S}_d is generally not diagonal, so its inversion in (4.96) has a high complexity and the algorithm may not be very useful in practice. Since \mathbf{S}_d is composed of $(K \cdot P)^2$ sub-matrices

$$\mathbf{S}_{i,j,k} = \lambda \mathbf{S}_{i,j,k}(m-1) + (1-\lambda) \mathbf{X}_{i,k}^*(m) \mathbf{G}_{2N \times 2N}^{01} \mathbf{X}_{j,k}(m), \quad (4.99)$$

it is desirable that each of those sub-matrices be a diagonal matrix. In the next paragraph, we will argue that $\mathbf{G}_{2N \times 2N}^{01}$ can be well approximated by the identity matrix with weight 1/2; accordingly, after introducing the positive factor $\mu \leq 2$ in (4.98) and the matrix $\mathbf{S}'(m)$ approximating $2\mathbf{S}_d(m)$, we then obtain the following approximate algorithm:

$$\mathbf{S}'(m) = \lambda \mathbf{S}'(m-1) + (1-\lambda) \mathbf{X}^H(m) \mathbf{X}(m) \quad (4.100)$$

$$\mathbf{K}(m) = (1-\lambda) \mathbf{S}'^{-1}(m) \mathbf{X}^H(m) \quad (4.101)$$

$$\underline{\mathbf{E}}_{2N \times Q}(m) = \underline{\mathbf{Y}}_{2N \times Q}(m) - \mathbf{G}_{2N \times 2N}^{01} \mathbf{X}(m) \hat{\underline{\mathbf{H}}}_{2LP \times Q}(m-1) \quad (4.102)$$

$$\hat{\underline{\mathbf{H}}}_{2LP \times Q}(m) = \hat{\underline{\mathbf{H}}}_{2LP \times Q}(m-1) + \mu \mathbf{G}_{2LP \times 2LP}^{10} \mathbf{K}(m) \underline{\mathbf{E}}_{2N \times Q}(m), \quad (4.103)$$

where each sub-matrix of \mathbf{S}' and \mathbf{K} is now a diagonal matrix and $\mu \leq 2$ is a positive number. Note that the imprecision introduced by the approximation in (4.100) and thus in the Kalman gain (4.101) will only affect the convergence rate. Obviously, we cannot permit the same kind of approximation in (4.102), because that would result in approximating a linear convolution by a circular one, which of course can have a disastrous impact in the adaptive filter behaviour.

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

$$(\mathbf{G}_{2N \times 2N}^{01})^* = \mathbf{F}_{2N \times 2N}^{-1} \mathbf{W}_{2N \times 2N}^{01} \mathbf{F}_{2N \times 2N}. \quad (4.104)$$

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

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

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

$$\begin{aligned} g_k &= \frac{1}{2N} \sum_{l=N}^{2N-1} \exp(-j2\pi kl/2N) \\ &= \frac{(-1)^k}{2N} \sum_{l=0}^{N-1} \exp(-j\pi kl/N), \end{aligned} \quad (4.105)$$

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

$$\begin{aligned} g_k &= \begin{cases} 0.5 & k = 0 \\ \frac{(-1)^k}{2N} \frac{1 - \exp(-j\pi k)}{1 - \exp(-j\pi k/N)} & k \neq 0 \end{cases} \\ &= \begin{cases} 0.5 & k = 0 \\ 0 & k \text{ even} \\ -\frac{1}{2N} \left[1 - j \cot\left(\frac{\pi k}{2N}\right)\right] & k \text{ odd,} \end{cases} \end{aligned} \quad (4.106)$$

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

$$\mathbf{g}^H \mathbf{g} - g_0^2 = \sum_{l=1}^{2N-1} |g_l|^2 = 2 \sum_{l=1}^{N-1} |g_l|^2 = \frac{1}{4}. \quad (4.107)$$

We can see from (4.107) that the first element of vector \mathbf{g} , i.e., g_0 , is dominant in a mean-square sense, and from (4.106) that the absolute values of the N first elements of \mathbf{g} decrease rapidly to zero as k increases. Because of the conjugate symmetry, i.e. $|g_k| = |g_{2N-k}|$ for $k = 1, \dots, N - 1$, the last few elements of \mathbf{g} are not negligible, but this affects only the first and last columns of $\mathbf{G}_{2N \times 2N}^{01}$ since this matrix is circulant with \mathbf{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 N large, only the very first (few) off-diagonals of $\mathbf{G}_{2N \times 2N}^{01}$ will be non-negligible while the others can be completely neglected. We also neglect the influence of the two isolated peaks $|g_{2N-1}| = |g_1| < g_0$ on the lower left corner and the upper right corner, respectively. Thus, approximating $\mathbf{G}_{2N \times 2N}^{01}$ by a diagonal matrix, i.e., $\mathbf{G}_{2N \times 2N}^{01} \approx g_0 \mathbf{I} = \mathbf{I}/2$, 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 chapter, we suppose that $0 < \mu \leq 2$.

4.5.2 Special Cases

In the single-channel case $P = Q = 1$, \mathbf{S}' and \mathbf{K} are diagonal matrices for $N = L$ and the classical constrained FLMS [7] follows immediately from (4.100)-(4.103). This algorithm requires the computation of 5 FFTs of length $2L$ per block. By approximating $\mathbf{G}_{2LP \times 2LP}^{10}$ in (4.103) to the identity matrix, we obtain the unconstrained FLMS (UFLMS) algorithm [8] which requires

only 3 FFTs per block. Many simulations show that the two algorithms have virtually the same performance.

For $N < L$, $\mathbf{S}_d(m)$ in (4.95) 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 a new algorithm that we call *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 $\mathbf{S}'(m)$ in (4.100) by dropping the off-diagonal components in $\mathbf{S}'(m)$:

$$\mathbf{S}''(m) = \text{diag}\{\mathbf{S}_{1,1,0}(m), \dots, \mathbf{S}_{1,1,K-1}(m)\}, \quad (4.108)$$

where

$$\mathbf{S}_{1,1,k}(m) = \lambda \mathbf{S}_{1,1,k}(m-1) + (1-\lambda) \mathbf{X}_{1,k}^*(m) \mathbf{X}_{1,k}(m)$$

are $(2N \times 2N)$ diagonal matrices.

In the MIMO case, (4.101) is the solution of a $P \times P$ system of linear equations of block matrices (which consist of K^2 diagonal block matrices each):

$$\mathbf{K}(m) = [\mathbf{K}_1^T(m), \dots, \mathbf{K}_P^T(m)]^T. \quad (4.109)$$

This allows us to *formally* write the update equation (4.103) as PQ tightly coupled 'single-channel' update equations

$$\hat{\mathbf{h}}_{p,q}(m) = \hat{\mathbf{h}}_{p,q}(m-1) + \mu \mathbf{G}_{2N \times 2N}^{10} \mathbf{K}_p \mathbf{e}_q(m) \quad (4.110)$$

($p = 1, \dots, P$, $q = 1, \dots, Q$) with the sub-matrices $\mathbf{K}_p(m)$ taking the cross-correlations between the input channels into account. These update equations (4.110) can then be calculated element-wise and the (cross) power spectra are estimated recursively:

$$\mathbf{S}_{i,j}(m) = \lambda \mathbf{S}_{i,j}(m-1) + (1-\lambda) \mathbf{X}_i^*(m) \mathbf{X}_j(m), \quad (4.111)$$

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

It is important to note that the calculation of the Kalman gain (Eqs. (4.95) and (4.96)), 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 (4.110) formally correspond to single-channel algorithms (e.g., (U)FLMS for $N = L$).

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

$$\mathbf{K}_1 = \mathbf{D}(m) \mathbf{S}_{1,1}^{-1}(m) [\mathbf{X}_1^*(m) - \mathbf{S}_{1,2}(m) \mathbf{S}_{2,2}^{-1}(m) \mathbf{X}_2^*(m)] \quad (4.112)$$

$$\mathbf{K}_2 = \mathbf{D}(m) \mathbf{S}_{2,2}^{-1}(m) [\mathbf{X}_2^*(m) - \mathbf{S}_{2,1}(m) \mathbf{S}_{1,1}^{-1}(m) \mathbf{X}_1^*(m)], \quad (4.113)$$

with the abbreviation

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

The solutions of (4.101) for more than two input channels may be formulated similarly to the corresponding part of the stereo update equations (4.112) and (4.113) (e.g. using Cramer's rule). These representations allow an intuitive interpretation as a correction of the interchannel-correlations in \mathbf{K}_i between \mathbf{X}_i^* and the other input signals $\mathbf{X}_j^*, j \neq i$.

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

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

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

Unfortunately, for a higher number of channels (and/or a higher number of sub-filters in case of the extended multidelay filter), the number of update terms increases rapidly, and the equations become too complicated for practical use. Therefore, a more efficient scheme for these cases will be proposed in section 4.7.

4.6 A Dynamical Regularization Strategy

In most practical scenarios, the desired signal $y(n)$ is disturbed, e.g., by some acoustic background noise. As shown above (c.f. (4.87)), the parameter estimation (i.e., misalignment) is very sensitive in poorly excited frequency bins. For robust adaptation the power spectral densities $\mathbf{S}_{i,i}$ are replaced by regularized versions according to $\tilde{\mathbf{S}}_{i,i} = \mathbf{S}_{i,i} + \text{diag}\{\delta_i\}$ prior to inversion in (4.96). The basic feature of the regularization is a compromise between fidelity to data and fidelity to some prior information about the solution [23]. The latter increases the robustness, but leads to biased solutions. Therefore, we propose here a *bin-selective dynamical regularization vector*

$$\delta_i(m) = \delta_{\max} \cdot [e^{-S_{i,i}^{(0)}(m)/S_0}, \dots, e^{-S_{i,i}^{(2L-1)}(m)/S_0}]^T \quad (4.114)$$

with two scalar parameters δ_{\max} and S_0 . $S_{i,i}^{(\nu)}$ denotes the ν -th frequency component ($\nu = 0, \dots, 2N - 1$) on the main diagonal of $S_{i,i}$. Note that for efficient implementation, e in (4.114) may be replaced by a basis 2 and modified S_0 . δ_{\max} should be chosen according to the (estimated) disturbing noise level in the desired signal $y(n)$.

As shown in Fig. 4.4, this exponential method provides a smooth transition between regularization for low input power and data fidelity whenever

the input power is large enough, and yields improved results compared to fixed regularization and to the popular approach of choosing the maximum out of the respective component $\mathbf{S}_{i,i}^{(\nu)}$ and a fixed threshold δ_{th} . Results of numerical simulations can be found in Section 4.8. The method also 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.

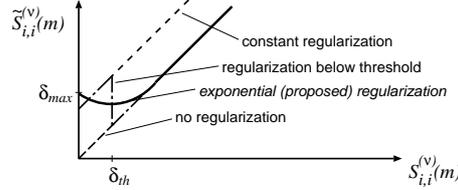


Fig. 4.4. Different regularization methods (channel i , bin ν).

4.7 Efficient Multichannel Realization

As will be demonstrated by simulation results and real-world applications in Section 4.8, the presented algorithm copes well with multichannel input. The cases of a larger number of filter input channels (P larger than 2 or 3) and/or a larger number of sub-filters ($N < L$) when using the EMDF algorithm (c.f. Section 4.5.2) call for further improvement of the computational efficiency. In this section, we propose efficient and stable recursive calculation schemes for the frequency-domain Kalman gain 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 [13]. Incorporating this extension in the proposed algorithm is very simple. Essentially, only the way the input data matrices (4.22) are calculated, is modified to

$$\begin{aligned} \mathbf{X}_{p,k}(m) = \text{diag}\{ & \mathbf{F}_{2N \times 2N} [x_p(m \frac{N}{\alpha} - Nk - N), \dots \\ & \dots, x_p(m \frac{N}{\alpha} - Nk + N - 1)]^T \}. \end{aligned} \quad (4.115)$$

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

4.7.1 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 (4.101) as follows.

Due to the block diagonal structure of (4.101), it can be simply decomposed w.r.t. the DFT components $\nu = 0, \dots, 2N - 1$ into $2N$ equations

$$\mathbf{K}^{(\nu)}(m) = (1 - \lambda)(\mathbf{S}^{(\nu)}(m))^{-1}(\mathbf{X}^{(\nu)}(m))^H \quad (4.116)$$

with (usually small) $KP \times KP$ unitary and positive definite matrices $\mathbf{S}^{(\nu)}$ containing the ν -th components on the block diagonals of \mathbf{S}'^{-1} in (4.101). Both $\mathbf{K}^{(\nu)}$ and $\mathbf{X}^{(\nu)}$ are *vectors* of length KP . Note that for real input signals x_i we need to solve (4.116) only for $N + 1$ bins.

A well-known and numerically stable method for this type of problems is the Cholesky decomposition of $\mathbf{S}^{(\nu)}$ followed by solution via backsubstitution, see [24]. The resulting total complexity for one output value is then

$$O(KP \cdot \log_2(2N)) + O((KP)^3), \quad (4.117)$$

where for the (U)FLMS algorithm in the two-channel (stereo) case the second term $O((KP)^3)$ 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 4.8) we introduce a recursive solution of (4.116) that jointly estimates the *inverse* power spectra $(\mathbf{S}^{(\nu)})^{-1}$ in (4.100) using the matrix-inversion lemma, e.g. [1]. This lemma relates a matrix

$$\mathbf{A} = \mathbf{B}^{-1} + \mathbf{C}\mathbf{D}^{-1}\mathbf{C}^H \quad (4.118)$$

to its inverse according to

$$\mathbf{A}^{-1} = \mathbf{B} - \mathbf{B}\mathbf{C}(\mathbf{D} + \mathbf{C}^H\mathbf{B}\mathbf{C})^{-1}\mathbf{C}^H\mathbf{B}, \quad (4.119)$$

as long as \mathbf{A} and \mathbf{B} are positive definite. Comparing (4.100) to (4.118), we immediately obtain from (4.100) an update equation for the *inverse* matrices

$$\begin{aligned} (\mathbf{S}^{(\nu)}(m))^{-1} &= \lambda^{-1} \left[(\mathbf{S}^{(\nu)}(m-1))^{-1} \right. \\ &\quad \left. - \frac{(\mathbf{S}^{(\nu)}(m-1))^{-1}(\mathbf{X}^{(\nu)}(m))^H \mathbf{X}^{(\nu)}(m)(\mathbf{S}^{(\nu)}(m-1))^{-1}}{\lambda(1-\lambda)^{-1} + \mathbf{X}^{(\nu)}(m)(\mathbf{S}^{(\nu)}(m-1))^{-1}(\mathbf{X}^{(\nu)}(m))^H} \right] \end{aligned}$$

using the bin-wise quantities introduced in (4.116) (making the denominator a scalar value). Introduction of the common vector

$$\mathbf{T}_1^{(\nu)}(m) = (\mathbf{S}^{(\nu)}(m-1))^{-1}(\mathbf{X}^{(\nu)}(m))^H \quad (4.120)$$

in the numerator and the denominator leads to

$$\begin{aligned} (\mathbf{S}^{(\nu)}(m))^{-1} &= \lambda^{-1}(\mathbf{S}^{(\nu)}(m-1))^{-1} \\ &\quad - \frac{\mathbf{T}_1^{(\nu)}(m)(\mathbf{T}_1^{(\nu)}(m))^H}{\lambda^2(1-\lambda)^{-1} + \lambda\mathbf{X}^{(\nu)}(m)\mathbf{T}_1^{(\nu)}(m)}. \end{aligned} \quad (4.121)$$

The Kalman gain (4.116) can then be efficiently calculated (using (4.121)) by

$$\mathbf{K}^{(\nu)}(m) = \frac{1-\lambda}{\lambda}\mathbf{T}_1^{(\nu)}(m) \left[1 - \frac{(\mathbf{T}_1^{(\nu)}(m))^H(\mathbf{X}^{(\nu)}(m))^H}{\lambda(1-\lambda)^{-1} + \mathbf{X}^{(\nu)}(m)\mathbf{T}_1^{(\nu)}(m)} \right]. \quad (4.122)$$

Again, there are common terms

$$T_2^{(\nu)}(m) = \mathbf{X}^{(\nu)}(m)\mathbf{T}_1^{(\nu)}(m) \quad (4.123)$$

in (4.122) and (4.121).

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 usually small $KP \times KP$ systems, where $KP \ll N$, (4.116), it is numerically much less critical than in the RLS algorithm. Note that for $N = L$, there is no analogon to a more efficient *fast* RLS [25] due to the different matrix structures (in this case, vector $\mathbf{X}^{(\nu)}(m)$ does not reflect a tapped delay line).

The complexity of the different computation methods for the Kalman gains (in MUL/ADDs for one output value $e(n)$) are compared in Fig. 4.5 for the case $N = L$ (i.e., $K = 1$). Finally, we note that further gains in computational complexity can be achieved this way when employing the extended multidelay filter for $N < L$.

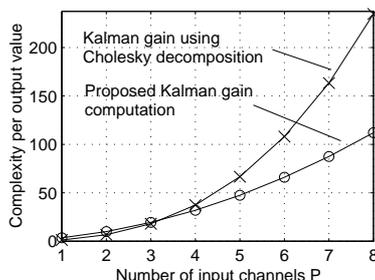


Fig. 4.5. Complexity (MUL/ADDs) of Kalman gain for $K = 1$.

4.7.2 Dynamical Regularization for Proposed Kalman Gain Approach

Due to the recursion (4.121) the regularization according to (4.114) is not immediately applicable. Therefore, an equivalent modification is applied directly to the data matrices $\mathbf{X}^{(\nu)}(m)$ by addition of mutually uncorrelated white noise sequences to each channel and frequency bin, respectively. Using the modified signal vectors, denoted by

$$\tilde{\mathbf{X}}^{(\nu)}(m) = \mathbf{X}^{(\nu)}(m) + \mathbf{N}^{(\nu)}(m), \quad (4.124)$$

where $\mathbf{N}^{(\nu)}(m)$ are the vectors of the white noise signals, we obtain the modified power spectral density matrices (c.f. Eq. (4.100))

$$\tilde{\mathbf{S}}^{(\nu)}(m) \approx (1 - \lambda) \sum_{q=0}^m \lambda^{m-q} \mathbf{X}^{(\nu)H}(q) \mathbf{X}^{(\nu)}(q)$$

$$+ (1 - \lambda) \sum_{q=0}^m \lambda^{m-q} \text{diag}\{|N_1^{(\nu)}(q)|^2, \dots, |N_{P.K}^{(\nu)}(q)|^2\}^T. \quad (4.125)$$

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

$$\begin{aligned} \delta_i^{(\nu)}(m) &= (1 - \lambda) \sum_{q=0}^m \lambda^{m-q} |N_i^{(\nu)}(q)|^2, \\ &= \lambda \delta_i^{(\nu)}(m-1) + (1 - \lambda) |N_i^{(\nu)}(m)|^2. \end{aligned} \quad (4.126)$$

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

$$|N_i^{(\nu)}(m)|^2 = \frac{\delta_i^{(\nu)}(m) - \lambda \delta_i^{(\nu)}(m-1)}{1 - \lambda}. \quad (4.127)$$

On the other hand, according to (4.114), the regularization should be chosen according to

$$\begin{aligned} \delta_i^{(\nu)}(m) &= \delta_{\max} \cdot \exp\left(-\frac{S_{i,i}^{(\nu)}(m)}{S_0}\right) \\ &= \delta_{\max} \cdot \exp\left(-\frac{\lambda S_{i,i}^{(\nu)}(m-1) + (1 - \lambda) |X_i^{(\nu)}(m)|^2}{S_0}\right). \end{aligned} \quad (4.128)$$

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 therefore need not be computed at all due to the matrix-inversion lemma (4.119)), since

$$\begin{aligned} \delta_i^{(\nu)}(m) &= \delta_{\max} \left[\exp\left(-\frac{S_{i,i}^{(\nu)}(m-1)}{S_0}\right) \right]^\lambda \cdot \exp\left(-\frac{(1 - \lambda) |X_i^{(\nu)}(m)|^2}{S_0}\right) \\ &= \delta_{\max}^{1-\lambda} (\delta_i^{(\nu)}(m-1))^\lambda \cdot \exp\left(-\frac{(1 - \lambda) |X_i^{(\nu)}(m)|^2}{S_0}\right). \end{aligned} \quad (4.129)$$

4.7.3 Efficient DFT calculation of overlapping data blocks

In this section we address the first term of the computational cost given in (4.117) which is mainly determined by the DFTs of the frequency-domain adaptive filtering scheme (Fig. 4.2). The $2N$ -point DFT calculation in (4.115) has to be carried out for each of the P loudspeaker signals and is therefore most costly. Moreover, as will be discussed in Section 4.8, 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 (4.115) recursively as well. For simplicity, we assume a block length of $N = L$ in this section.

For the following derivation,

$$x_i^{(n)}(m) = x_i \left(m \frac{L}{\alpha} - L + n \right) \quad (4.130)$$

denotes the n -th component ($n = 0, \dots, 2L - 1$) of the time domain vector (block index m) to be transformed in (4.115). Let us now consider the ν -th element on the diagonal of $\mathbf{X}_i(m)$ where $w = e^{-j2\pi/2L}$:

$$X_i^{(\nu)}(m) = \sum_{n=0}^{2L-1} x_i^{(n)}(m) w^{\nu n}. \quad (4.131)$$

Separating the summation into one for previous and one for new input values (Fig. 4.6), followed by the introduction of the previous vector elements $x_i^{(n)}(m - 1)$ leads to

$$\begin{aligned} X_i^{(\nu)}(m) &= \sum_{n=0}^{2L-L/\alpha-1} x_i^{(n)}(m) w^{\nu n} + \sum_{n=2L-L/\alpha}^{2L-1} x_i^{(n)}(m) w^{\nu n} \\ &= \sum_{n=L/\alpha}^{2L-1} x_i^{(n)}(m-1) w^{\nu(n-L/\alpha)} + \Delta X_i^{(\nu)}(m), \end{aligned} \quad (4.132)$$

where

$$\Delta X_i^{(\nu)}(m) = \sum_{n=2L-L/\alpha}^{2L-1} x_i^{(n)}(m) w^{\nu n} \quad (4.133)$$

contains the new input values and will be the update term in our recursive scheme. Next, we introduce the previous DFT output values $X_i^{(\nu)}(m - 1)$ by

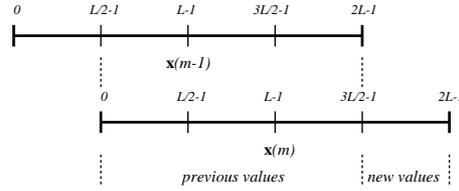


Fig. 4.6. Example: overlapping data blocks, $\alpha = 2$.

subtracting the vector elements of $\mathbf{x}_i^{(n)}(m - 1)$ of the previous data vector

shifted out of the DFT length $2L$:

$$\begin{aligned}
X_i^{(\nu)}(m) &= w^{-\nu L/\alpha} \left(\sum_{n=0}^{2L-1} x_i^{(n)}(m-1) w^{\nu n} \right. \\
&\quad \left. - \sum_{n=0}^{L/\alpha-1} x_i^{(n)}(m-1) w^{\nu n} \right) + \Delta X_i^{(\nu)}(m) \\
&= w^{-\nu L/\alpha} X_i^{(\nu)}(m-1) - w^{-\nu L/\alpha} \\
&\quad \cdot \sum_{n=2L-L/\alpha}^{2L-1} x_i^{(n-2L+L/\alpha)}(m) w^{\nu(n-2L+L/\alpha)} \\
&\quad + \Delta X_i^{(\nu)}(m). \tag{4.134}
\end{aligned}$$

Using (4.130), we can show that

$$x_i^{(n-2L+L/\alpha)}(m) = x_i^{(n)}(m-2\alpha+1). \tag{4.135}$$

Finally, we obtain

$$\begin{aligned}
X_i^{(\nu)}(m) &= w^{-\nu L/\alpha} X_i^{(\nu)}(m-1) \\
&\quad - w^{-\nu 2L} \Delta X_i^{(\nu)}(m-2\alpha+1) + \Delta X_i^{(\nu)}(m). \tag{4.136}
\end{aligned}$$

Again, this recursive update needs to be carried out only for the bins $\nu = 0, \dots, L$ if $x_i^{(n)}(m)$ 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 4.7 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. Generally, the elimination of operations on zeros in the FFT is referred to as pruning and was first described by Markel [26]. Since then, several pruning algorithms with increased efficiency have been proposed. A summary and further references of different approaches may be found, e.g., in [27].

In summary, using FFT pruning, the recursive DFT approach reduces the first term of the complexity in (4.117) for $N = L$ to $O(P \cdot \log_2(L/\alpha))$ for each output point.

4.8 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 perfor-

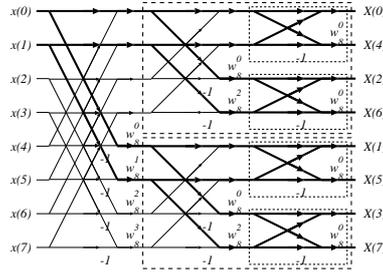


Fig. 4.7. Illustration of decimation-in-frequency FFT with windowed input.

mance of our approach in a few examples for hands-free speech communication.

4.8.1 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. 4.8).

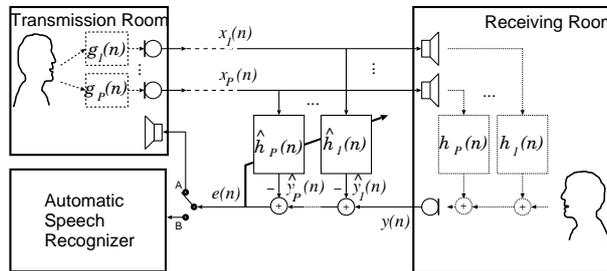


Fig. 4.8. Multichannel acoustic echo cancellation.

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 [15,16]. It is shown in [17] that the only solution to the nonuniqueness problem is to reduce the correlation between the different signals. Three methods of preprocessing can be distinguished: inaudible nonlinear processing, e.g., [17], additive noise (preferably below the masking threshold of human hearing),

e.g., [28], and time-varying filtering, e.g., [29]. For the following example, 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 [17] (P different nonlinearities with factor 0.5). In this subsection we consider only $Q = 1$ microphone in the receiving room. The convergence behaviour is shown both in terms of system misalignment (ratio of the squared norms of (4.69) 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 $y(n) - n_O(n)$ and the residual echo $e(n) - n_O(n)$. (For smoothing the $ERLE$ curves, a moving average filter of length 256 was used.)

Figure 4.9 illustrates the effect of taking the cross-correlations in (4.112) and (4.113) for $P = 2$ into account. As input $x_p(n)$, 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).

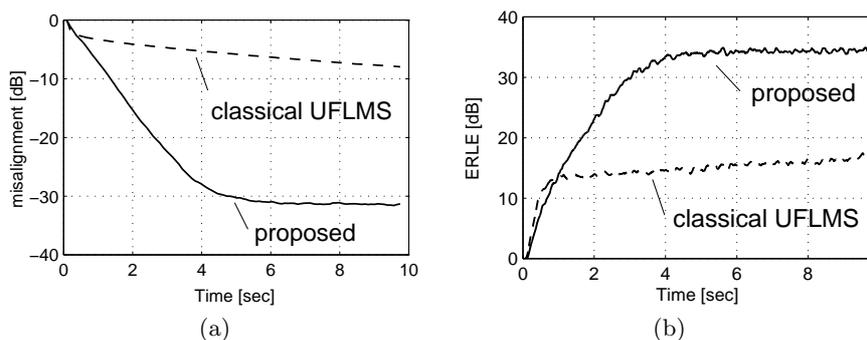


Fig. 4.9. Effect of taking cross-correlation into account ($P=2$ channels, $\alpha = 4$). (a) Misalignment, (b) $ERLE$.

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 4.10 shows the misalignment convergence of the described algorithm (solid) for the multichannel cases $P = 2$ (lowest curve), 3, 4, 5 (uppermost curve), 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 4.11 (a) shows the corresponding $ERLE$ curves.

Figure 4.11 (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 computation 4.122 (lower solid line).

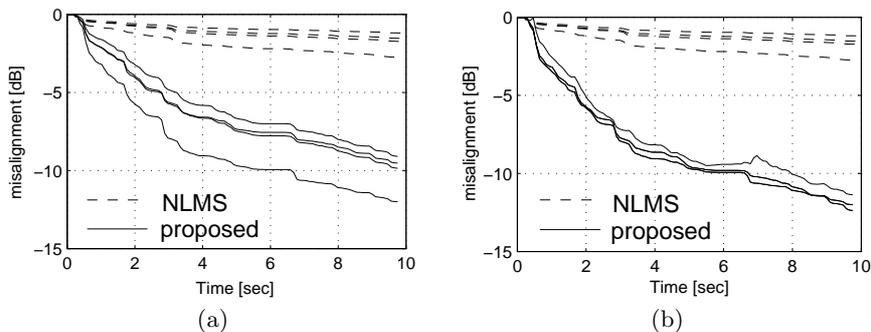


Fig. 4.10. Misalignment convergence for the multichannel cases $P=2$ (lowest), 3, 4, 5 (uppermost). (a) Overlap $\alpha = 4$, (b) overlap α adjusted.

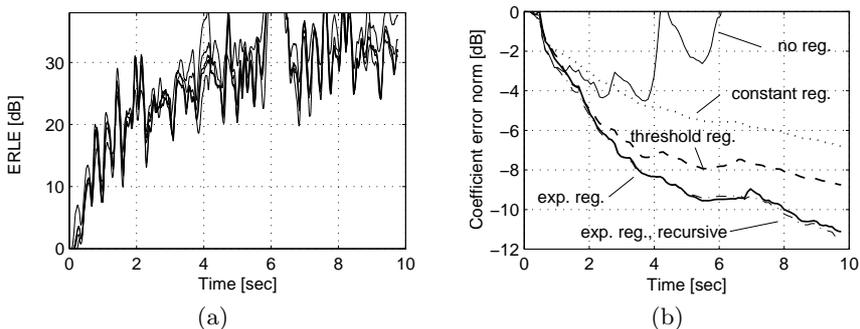


Fig. 4.11. (a) *ERLE* convergence for the multichannel cases $P=2, 3, 4, 5$ and adjusted α and (b) comparison of regularization methods, $P=5$, $\alpha=16$.

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].

4.8.2 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. 4.12).

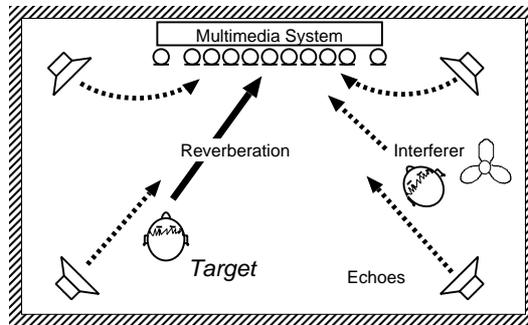


Fig. 4.12. Hands-free speech recognition in multimedia systems.

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 [30]. In any practical system, this scenario presents a MIMO system identification problem for the acoustic echo canceller [30,3]. Fortunately, as noted in Section 4.4, the costly calculation of the Kalman gain is necessary only once, i.e., it is independent of the number of microphones. Figure 4.13 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 [30]. Thanks to the efficient frequency-domain ap-

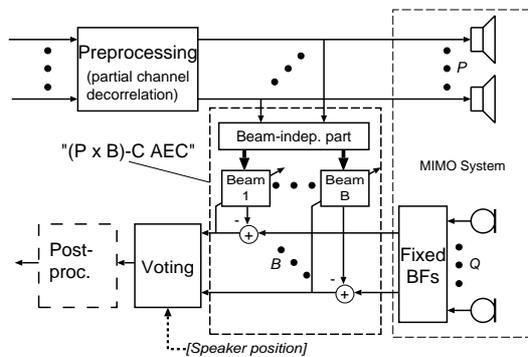


Fig. 4.13. A human/machine interface for hands-free speech recognition.

proach, it has become possible to run such a system in real-time on standard PC platforms. The implementation is fully scalable (e.g., sampling rate, number of loudspeakers, microphones, and beams) with dynamical allocation of

computational power. Example parameters for a speech recognition interface with 5.1-channel surround sound reproduction running on a 1.7 GHz dual processor board are: $PL = 5 \cdot 4096$ adaptive filter coefficients, an overlap factor $\alpha = 16$, and a sampling rate of 12kHz for the acquisition.

4.9 Conclusions

In many applications where an adaptive filter is required, frequency-domain algorithms are an attractive alternative to time-domain algorithms, especially 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.

A general framework for multichannel frequency-domain adaptive filtering was presented and its efficiency in actual applications was demonstrated. A generic multichannel algorithm with a MMSE 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 both, well-known algorithms in the single-channel case, such as the FLMS and UFLMS, and new algorithms such as the EMDF. For the multichannel case the framework is attractive as the cross-correlations between all input signals are efficiently taken into account. We have also presented strategies to improve the computational efficiency further by introducing stable schemes for recursive DFT and Kalman gain computation. Several simulations and real-time implementations illustrate the benefits of the multichannel algorithm.

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 Canceller for Microsoft Windows," in *Proc. Int. Workshop on Acoustic Echo and Noise Control*, Darmstadt, Germany, pp. 87-90, Sept. 2001.
3. H. Buchner, W. Herboldt, 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. M. Dentino, J. McCool, and B. Widrow, "Adaptive filtering in the frequency domain," *Proc. IEEE*, vol. 66, pp.1658-1659, Dec. 1978.

7. 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.
8. 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.
9. 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.
10. 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.
11. J. Benesty and P. Duhamel, "Fast constant modulus adaptive algorithm," *IEE Proc.-F*, vol. 138, pp. 379-387, Aug. 1991.
12. 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.
13. 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.
14. J. Prado and E. Moulines, "Frequency-domain adaptive filtering with applications to acoustic echo cancellation," *Ann. Télécommun.*, vol. 49, pp. 414-428, 1994.
15. 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.
16. S. Shimauchi and S. Makino, "Stereo projection echo canceller with true echo path estimation," in *Proc. ICASSP*, pp. 3059-3062, May 1995.
17. 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.
18. 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.
19. 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.
20. 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.
21. 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.
22. 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.
23. V.A. Morozov, *Regularization Methods for Ill-posed Problems*, CRC Press, Boca Raton, FL, 1993.

24. G.H. Golub and C.F. Van Loan, *Matrix Computations*, 2nd ed., Johns Hopkins, Baltimore, MD, 1989.
25. 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.
26. J.D. Markel, "FFT pruning," *IEEE Trans. Audio Electroacoust.*, vol. AU-19, pp. 305-311, Dec. 1971.
27. S. Holm, "FFT pruning applied to time domain interpolation and peak localization," *IEEE Trans. Acoustics, Speech, and Signal Processing*, vol. ASSP-35, no. 12, pp. 1776-1778, Dec. 1987.
28. T. Gänsler and P. Eneroth, "Influence of audio coding on stereophonic acoustic echo cancellation," in *Proc. ICASSP*, pp. 3649-3652, May 1998.
29. 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.
30. 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.