Chapter 1

ADAPTIVE ALGORITHMS FOR MIMO ACOUSTIC ECHO CANCELLATION

Jacob Benesty
Université du Québec, INRS-EMT
benesty@inrs-emt.uquebec.ca

Tomas Gänsler
Agere Systems
gaensler@agere.com

Yiteng (Arden) Huang
Bell Laboratories, Lucent Technologies
arden@research.bell-labs.com

Markus Rupp
TU Wien, Institute for Communication and RF Engineering
mrupp@nt.tuwien.ac.at

Abstract The first thing that comes in mind when we talk about acoustic echo cancellation is adaptive filtering. In this chapter, we discuss a large number of multichannel adaptive algorithms, both in time and frequency domains. This discussion will be developed in the context of multichannel acoustic echo cancellation where we have to identify a multiple-input multiple-output (MIMO) system (e.g., room acoustic impulse responses).

Keywords: Acoustic Echo Cancellation, Multichannel, Adaptive Algorithms, LMS, APA, RLS, FRLS, Exponentiated, MIMO, Frequency-Domain

1. INTRODUCTION

All today's teleconferencing systems are hands-free and single-channel (meaning that there is only one microphone and one loudspeaker). In the near future, we expect that multichannel systems (with at least two loudspeakers and at least one microphone) will be available to customers, therefore providing a realistic presence that single-channel systems cannot offer.

In hands-free systems, the coupling between loudspeakers and microphones can be very strong and this can generate important echoes that eventually make the system completely unstable (e.g., the system starts howling). Therefore, multichannel acoustic echo cancelers (MCAECs) are absolutely necessary for full-duplex communication [1]. Let P and Q be respectively the numbers of loudspeakers and microphones. For a teleconferencing system, the MCAECs consist of PQ adaptive filters aiming at identifying PQ echo paths from P loudspeakers to Q microphones. This scheme is, in fact, a multiple-input multiple-output (MIMO) system. We assume that the teleconferencing system is organized between two rooms: the "transmission" and "receiving" rooms. The transmission room is sometimes referred to as the far-end and the receiving room as the near-end. So each room needs an MCAEC for each microphone. Thus, multichannel acoustic echo cancellation consists of a direct identification of an unknown linear MIMO system.

Although conceptually very similar, multichannel acoustic echo cancellation (MCAEC) is fundamentally different from traditional mono echo cancellation in one respect: a straightforward generalization of the mono echo canceler would not only have to track changing echo paths in the receiving room, *but also in the transmission room*! For example, the canceler would have to reconverge if one talker stops talking and another starts talking at a different location in the transmission room. There is no adaptive algorithm that can track such a change sufficiently fast and this scheme therefore results in poor echo suppression. Thus, a generalization of the mono AEC in the multichannel case does not result in satisfactory performance.

The theory explaining the problem of MCAEC was described in [1] and [2]. The fundamental problem is that the multiple channels may carry linearly related signals which in turn may make the normal equations to be solved by the adaptive algorithm singular. This implies that there is no unique solution to the equations 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. As a result, intensive studies have been made of how to handle this properly. It was shown in [2] that the only solution to the nonuniqueness problem is to reduce the coherence between the different loudspeaker signals, and an efficient low complexity method for this purpose was also given.

Lately, attention has been focused on the investigation of other methods that decrease the cross-correlation between the channels in order to get well behaved estimates of the echo paths [3], [4], [5], [6], [7], [8]. The main problem is how to reduce the coherence sufficiently without affecting the stereo perception and the sound quality.

The performance of the MCAEC is more severely affected by the choice of the adaptive algorithm than the monophonic counterpart [9], [10]. This is easily recognized since the performance of most adaptive algorithms depends on the condition number of the input signal covariance matrix. In the multichannel case, the condition number is very high; as a result, algorithms such as the least-mean-square (LMS) or the normalized LMS (NLMS), which do not take into account the cross-correlation among all the input signals, converge very slowly to the true solution. It is therefore highly interesting to study multichannel adaptive filtering algorithms.

In this chapter, we develop a general framework for multichannel adaptive filters with the purpose to improve their performance in time and frequency domains. We also investigate a recently proposed class of adaptive algorithms that exploit sparsity of room acoustic impulse responses. These algorithms are very interesting both from theoretical and practical standpoints since they converge and track much better than the NLMS algorithm for example.

2. NORMAL EQUATIONS AND IDENTIFICATION OF A MIMO SYSTEM

We first derive the normal equations of a multiple-input multiple-output (MIMO) system.

2.1 NORMAL EQUATIONS

We assume that we have a MIMO system with P inputs (loudspeakers) and Q outputs (microphones). We also assume that the MIMO system (a room in our context) is linear and time-invariant. Acoustic echo cancellation consists of identifying P echo paths at each microphone so that in total, PQ echo paths need to be estimated. We have Q output

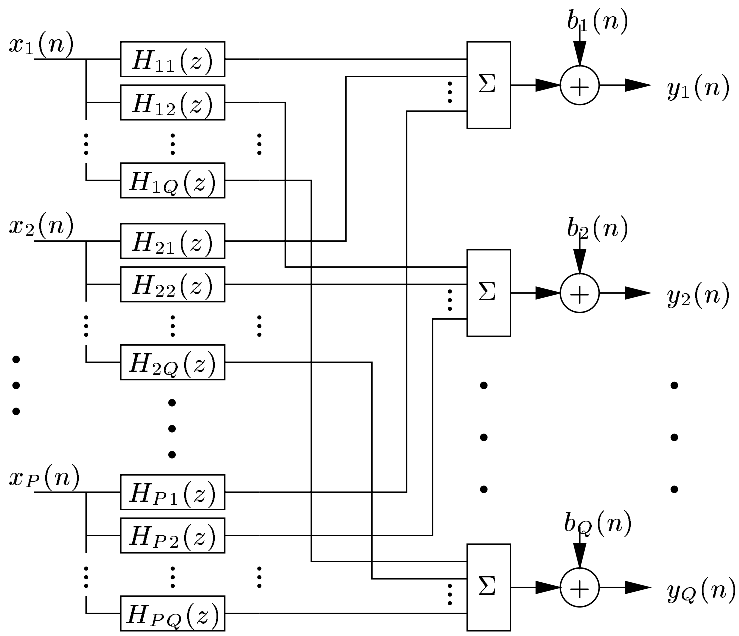

Figure 1.1 A MIMO system consisting of P inputs and Q outputs.

(microphone) signals (see Fig. 1.1):

$$\begin{aligned} y_q(n) &= \sum_{p=1}^{P} \mathbf{h}_{pq}^T \mathbf{x}_p(n) + b_q(n), \\ q &= 1, 2, ..., Q, \end{aligned} \tag{1.1}$$

where superscript ^{T} denotes transpose of a vector or a matrix,

$$\mathbf{h}_{pq} = \begin{bmatrix} h_{pq,0} & h_{pq,1} & \cdots & h_{pq,L-1} \end{bmatrix}^T$$

is the echo path – of length L – between loudspeaker p and microphone q,

$$\mathbf{x}_p(n) = \begin{bmatrix} x_p(n) & x_p(n-1) & \cdots & x_p(n-L+1) \end{bmatrix}^T, \\ p = 1, 2, ..., P,$$

is the pth reference (loudspeaker) signal (also called the far-end speech), and $b_q(n)$ is the near-end noise added at microphone q, assumed to be uncorrelated with the far-end speech. We define the error signal at time

n for microphone q as

$$\begin{aligned} e_q(n) &= y_q(n) - \hat{y}_q(n) \\ &= y_q(n) - \sum_{p=1}^{P} \hat{\mathbf{h}}_{pq}^T \mathbf{x}_p(n), \end{aligned} \tag{1.2}$$

where

$$\hat{\mathbf{h}}_{pq} = \begin{bmatrix} \hat{h}_{pq,0} & \hat{h}_{pq,1} & \cdots & \hat{h}_{pq,L-1} \end{bmatrix}^T$$

are the model filters. It is more convenient to define an error signal vector for all the microphones:

$$\begin{aligned} \mathbf{e}(n) &= \mathbf{y}(n) - \hat{\mathbf{y}}(n) \\ &= \mathbf{y}(n) - \hat{\mathbf{H}}^T \mathbf{x}(n), \end{aligned} \tag{1.3}$$

where

$$\begin{aligned} \mathbf{y}(n) &= \mathbf{H}^T \mathbf{x}(n) + \mathbf{b}(n), \\ \mathbf{H} &= \begin{bmatrix} \mathbf{h}_{11} & \mathbf{h}_{12} & \cdots & \mathbf{h}_{1Q} \\ \mathbf{h}_{21} & \mathbf{h}_{22} & \cdots & \mathbf{h}_{2Q} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{h}_{P1} & \mathbf{h}_{P2} & \cdots & \mathbf{h}_{PQ} \end{bmatrix}, \\ \mathbf{b}(n) &= \begin{bmatrix} b_1(n) & b_2(n) & \cdots & b_Q(n) \end{bmatrix}^T, \\ \mathbf{e}(n) &= \begin{bmatrix} e_1(n) & e_2(n) & \cdots & e_Q(n) \end{bmatrix}^T, \\ \mathbf{y}(n) &= \begin{bmatrix} y_1(n) & y_2(n) & \cdots & y_Q(n) \end{bmatrix}^T, \\ \hat{\mathbf{y}}(n) &= \begin{bmatrix} \hat{y}_1(n) & \hat{y}_2(n) & \cdots & \hat{y}_Q(n) \end{bmatrix}^T, \\ \hat{\mathbf{H}} &= \begin{bmatrix} \hat{\mathbf{h}}_{11} & \hat{\mathbf{h}}_{12} & \cdots & \hat{\mathbf{h}}_{1Q} \\ \hat{\mathbf{h}}_{21} & \hat{\mathbf{h}}_{22} & \cdots & \hat{\mathbf{h}}_{2Q} \\ \vdots & \vdots & \ddots & \vdots \\ \hat{\mathbf{h}}_{P1} & \hat{\mathbf{h}}_{P2} & \cdots & \hat{\mathbf{h}}_{PQ} \end{bmatrix}, \\ \mathbf{x}(n) &= \begin{bmatrix} \mathbf{x}_1^T(n) & \mathbf{x}_2^T(n) & \cdots & \mathbf{x}_P^T(n) \end{bmatrix}^T. \end{aligned}$$

Having written the error signal, we now define the recursive least-squares error criterion with respect to the modelling filters:

$$\begin{aligned} J(n) &= \sum_{i=0}^{n} \lambda^{n-i} \mathbf{e}^T(i)\mathbf{e}(i) \\ &= \sum_{q=1}^{Q} \sum_{i=0}^{n} \lambda^{n-i} e_q^2(i) \\ &= \sum_{q=1}^{Q} J_q(n), \end{aligned} \tag{1.4}$$

where λ $(0 < \lambda < 1)$ is a forgetting factor. The minimization of (1.4) leads to the multichannel normal equations:

$$\mathbf{R}_{xx}(n)\hat{\mathbf{H}}(n) = \mathbf{R}_{xy}(n), \tag{1.5}$$

where

$$\begin{aligned} \mathbf{R}_{xx}(n) &= \sum_{i=0}^{n} \lambda^{n-i} \mathbf{x}(i)\mathbf{x}^T(i) \\ &= \begin{bmatrix} \mathbf{R}_{11}(n) & \mathbf{R}_{12}(n) & \cdots & \mathbf{R}_{1P}(n) \\ \mathbf{R}_{21}(n) & \mathbf{R}_{22}(n) & \cdots & \mathbf{R}_{2P}(n) \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{R}_{P1}(n) & \mathbf{R}_{P2}(n) & \cdots & \mathbf{R}_{PP}(n) \end{bmatrix} \end{aligned} \tag{1.6}$$

is an estimate of the input signal covariance matrix – of size $(PL \times PL)$, and

$$\mathbf{R}_{xy}(n) = \sum_{i=0}^{n} \lambda^{n-i} \mathbf{x}(i)\mathbf{y}^T(i) \tag{1.7}$$

is an estimate of the cross-correlation matrix – of size $(PL \times Q)$ – between $\mathbf{x}(n)$ and $\mathbf{y}^T(n)$.

It can easily be seen that the multichannel normal equations (1.5) can be decomposed in Q independent normal equations, each one corresponding to a microphone signal:

$$\mathbf{R}_{xx}(n)\hat{\mathbf{h}}_q(n) = \mathbf{r}_{xy,q}(n), \; q = 1, 2, ..., Q, \tag{1.8}$$

where $\hat{\mathbf{h}}_q(n)$ [resp. $\mathbf{r}_{xy,q}(n)$] is the qth column of matrix $\hat{\mathbf{H}}(n)$ [resp. $\mathbf{R}_{xy}$(n)]. This result implies that minimizing $J(n)$ or minimizing each $J_q(n)$ independently gives the same results. This makes sense from an identification point of view, since the identification of the impulse responses for one microphone is completely independent of the others.

2.2 THE NONUNIQUENESS PROBLEM

In many situations, the signals $x_p(n)$ are generated from a unique source $s(n)$, so that:

$$x_p(n) = \mathbf{g}_p^T \mathbf{s}(n), \; p = 1, 2, ..., P, \tag{1.9}$$

where

$$\mathbf{g}_p = \left[\begin{array}{cccc} g_{p,0} & g_{p,1} & \cdots & g_{p,L-1} \end{array} \right]^T$$

is the impulse response between the source and microphone p in the transmission room in the case of a teleconferencing system [2]. Therefore the signals $x_p(n)$ are linearly related and we have the following $[P(P-1)/2]$ relations [2]:

$$\mathbf{x}_p^T(n)\mathbf{g}_i = \mathbf{x}_i^T(n)\mathbf{g}_p, \quad i, p = 1, 2, ..., P, \; i \neq p. \tag{1.10}$$

Indeed, since $x_p = s * g_p$, therefore $x_p * g_i = s * g_p * g_i = x_i * g_p$ (the symbol $*$ is the linear convolution operator). Now, consider the following vector:

$$\mathbf{u} = \left[\begin{array}{cccc} \sum_{p=2}^{P} \zeta_p \mathbf{g}_p^T & -\zeta_2 \mathbf{g}_1^T & \cdots & -\zeta_P \mathbf{g}_1^T \end{array} \right]^T ,$$

where ζ_p are arbitrary factors. We can verify using (1.10) that $\mathbf{R}_{xx}(n)\mathbf{u} = \mathbf{0}_{PL\times 1}$, so $\mathbf{R}_{xx}(n)$ is not invertible. Vector $\mathbf{u}$ represents the nullspace of matrix $\mathbf{R}_{xx}(n)$. The dimension of this nullspace depends of the number of inputs and is equal to $(P-2)L+1$ (for $P \geq 2$). So the problem becomes worse as P increases. Thus, there is no unique solution to the problem and an adaptive algorithm will drift to any one of many possible solutions, which can be very different from the "true" desired solution $\hat{\mathbf{h}}_{pq} = \mathbf{h}_{pq}$. These nonunique "solutions" are dependent on the impulse responses in the transmission room:

$$\hat{\mathbf{h}}_{1q} = \mathbf{h}_{1q} + \beta \sum_{p=2}^{P} \zeta_p \mathbf{g}_p, \tag{1.11}$$

$$\hat{\mathbf{h}}_{pq} = \mathbf{h}_{pq} - \beta \zeta_p \mathbf{g}_1, \; p = 2, ..., P, \tag{1.12}$$

where β is an arbitrary factor. This, of course, is intolerable because $\mathbf{g}_p$ can change instantaneously, for example, as one person stops talking and another starts [1], [2].

2.3 THE IMPULSE RESPONSE TAIL EFFECT

We first define an important measure that is very useful for MCAEC.
Definition: The quantity

$$\frac{\|\mathbf{h}_q - \hat{\mathbf{h}}_q\|}{\|\mathbf{h}_q\|}, \quad q = 1, 2, ..., Q, \tag{1.13}$$

where $\|\cdot\|$ denotes the two-norm vector, is called the *normalized misalignment* and measures the mismatch between the impulse responses of the receiving room and the modelling filters. In the multichannel case, it is possible to have good echo cancellation even when the misalignment is large. However, in such a case, the cancellation will degrade if the $\mathbf{g}_p$ change. A main objective of MCAEC research is to avoid this problem.

Actually, for the practical case when the length of the adaptive filters is smaller than the length of the impulse responses in the transmission room, there is a unique solution to the normal equation, although the covariance matrix is very ill-conditioned.

On the other hand, we can easily show by using the classical normal equations that if the length of the adaptive filters is smaller than the length of the impulse responses in the receiving room, we introduce an important bias in the coefficients of these filters because of the strong cross-correlation between the input signals and the large condition number of the covariance matrix [2]. So in practice, we may have poor misalignment even if there is a unique solution to the normal equations.

The only way to decrease the misalignment is to partially decorrelate two-by-two the P input (loudspeaker) signals. Next, we summarize a number of approaches that have been developed recently for reducing the cross-correlation.

2.4 SOME DIFFERENT SOLUTIONS FOR DECORRELATION

If we have P different channels, we need to decorrelate them partially and mutually. In the following, we show how to partially decorrelate two channels. The same process should be applied for all the channels. It is well-known that the coherence magnitude between two processes is equal to 1 if and only if they are linearly related. In order to weaken this relation, some non-linear or time-varying transformation of the stereo channels has to be made. Such a transformation reduces the coherence and hence the condition number of the covariance matrix, thereby improving the misalignment. However, the transformation has to be performed cautiously so that it is inaudible and has no effect on stereo perception.

A simple nonlinear method that gives good performance uses a half-wave rectifier [2], so that the nonlinearly transformed signal becomes

$$x'_p(n) = x_p(n) + \alpha \frac{x_p(n) + |x_p(n)|}{2}, \tag{1.14}$$

where α is a parameter used to control the amount of nonlinearity. For this method, there can only be a linear relation between the nonlinearly transformed channels if $\forall n,\ x_1(n) \geq 0$ and $x_2(n) \geq 0$ or if we have $ax_1(n - \tau_1) = x_2(n - \tau_2)$ with $a > 0$. In practice however, these cases never occur because we always have zero-mean signals and $\mathbf{g}_1$, $\mathbf{g}_2$ are in practice never related by just a simple delay.

An improved version of this technique is to use positive and negative half-wave rectifiers on each channel respectively,

$$x'_1(n) = x_1(n) + \alpha \frac{x_1(n) + |x_1(n)|}{2}, \tag{1.15}$$

$$x'_2(n) = x_2(n) + \alpha \frac{x_2(n) - |x_2(n)|}{2}. \tag{1.16}$$

This principle removes the linear relation even in the special signal cases given above.

Experiments show that stereo perception is not affected by the above methods even with α as large as 0.5. Also, the distortion introduced for speech is hardly audible because of the nature of the speech signal and psychoacoustic masking effects [11]. This is explained by the following three reasons. First, the distorted signal $x'_p(n)$ depends only on the instantaneous value of the original signal $x_p(n)$ so that during periods of silence, no distortion is added. Second, the periodicity remains unchanged. Third, for voiced sounds, the harmonic structure of the signal induces "self-masking" of the harmonic distortion components. This kind of distortion is also acceptable for some music signals but may be objectionable for pure tones.

Other types of nonlinearities for decorrelating speech signals have also been investigated and compared [12]. The results indicate that, of the several nonlinearities considered, ideal half-wave rectification and smoothed half-wave rectification appear to be the best choices for speech. For music, the nonlinearity parameter of the ideal rectifier must be readjusted. The smoothed rectifier does not require this readjustment but is a little more complicated to implement.

In [6] a similar approach with non-linearities is proposed. The idea is expanded so that four adaptive filters operate on different non-linearly processed signals to estimate the echo paths. These non-linearities are chosen such that the input signals of two of the adaptive filters are independent, which thus represent a "perfect" decorrelation. Tap estimates are then copied to a fixed two-channel filter which performs the echo cancellation with the unprocessed signals. The advantage of this method is that the NLMS algorithm could be used instead of more sophisticated algorithms.

Another approach that makes it possible to use the NLMS algorithm is to decorrelate the channels by means of complementary comb filtering [1], [13]. The technique is based on removing the energy in a certain frequency band of the speech signal in one channel. This means the coherence would become zero in this band and thereby results in fast alignment of the estimate even when using the NLMS algorithm. Energy is removed complementarily between the channels so that the stereo perception is not severely affected for frequencies above 1 kHz. However, this method must be combined with some other decorrelation technique for lower frequencies [14].

Two methods based on introducing time-varying filters in the transmission path were presented in [7], [8]. In [7], left and right signals are filtered through two independent time-varying first-order all-pass filters. Stochastic time-variation is introduced by making the pole position of the filter a random walk process. The actual position is limited by the constraints of stability and inaudibility of the introduced distortion. While significant reduction in correlation can be achieved for higher frequencies with the imposed constraints, the lower frequencies are still fairly unaffected by the time-variation. In [8], a periodically varying filter is applied to one channel so that the signal is either delayed by one sample or passed through without delay. A transition zone between the delayed and non-delayed state is also employed in order to reduce audible discontinuities. This method may also affect the stereo perception.

Although the idea of adding independent perceptually shaped noise to the channels was mentioned in [1], [2], thorough investigations of the actual benefit of the technique was not presented. Results regarding variants of this idea can be found in [4], [5]. A pre-processing unit estimating the masking threshold and adding an appropriate amount of noise was proposed in [5]. It was also noted that adding a masked noise to each channel may affect the spatialization of the sound even if the noise is inaudible at each channel separately. This effect can be controlled through correction of the masking threshold when appropriate. In [4], the improvement of misalignment was studied in the SAEC when a perceptual audio coder was added in the transmission path. Reduced correlation between the channels was shown by means of coherence analysis, and improved convergence rate of the adaptive algorithm was observed. A low-complexity method for achieving additional decorrelation by modifying the decoder was also proposed. The encoder cannot quantize every single frequency band optimally due to rate constraints. This has the effect that there is a margin on the masking threshold which can be exploited. In the presented method, the masking threshold is estimated from the modified discrete cosine transform (MDCT) coefficients delivered by the encoder, and an appropriate inaudible amount of decorrelating noise is added to the signals.

In the rest of this chapter, we suppose that one of the previous decorrelation methods is used so the normal equations have a unique solution. However, the input signals can still be highly correlated, therefore requiring special treatment.

3. THE CLASSICAL AND FACTORIZED MULTICHANNEL RLS

From the normal equations (1.8), we easily derived the classical update equations for the multichannel recursive least-squares (RLS):

$$e_q(n) = y_q(n) - \hat{\mathbf{h}}_q^T(n-1)\mathbf{x}(n), \quad (1.17)$$

$$\hat{\mathbf{h}}_q(n) = \hat{\mathbf{h}}_q(n-1) + \mathbf{R}_{xx}^{-1}(n)\mathbf{x}(n)e_q(n). \quad (1.18)$$

Note that the Kalman gain $\mathbf{k}(n) = \mathbf{R}_{xx}^{-1}(n)\mathbf{x}(n)$ is the same for all the microphone signals $q = 1, 2, ..., Q$. This is important, even though we have Q update equations, the Kalman vector needs to be computed only one time per iteration. Using the matrix inversion lemma, we obtain the following recursive equation for the inverse of the covariance matrix:

$$\begin{aligned}\mathbf{R}_{xx}^{-1}(n) &= \lambda^{-1}\mathbf{R}_{xx}^{-1}(n-1) \\ &- \frac{\lambda^{-2}\mathbf{R}_{xx}^{-1}(n-1)\mathbf{x}(n)\mathbf{x}^T(n)\mathbf{R}_{xx}^{-1}(n-1)}{1+\lambda^{-1}\mathbf{x}^T(n)\mathbf{R}_{xx}^{-1}(n-1)\mathbf{x}(n)}. \quad (1.19)\end{aligned}$$

Another way to write the multichannel RLS is to first factorize the covariance matrix inverse $\mathbf{R}_{xx}^{-1}(n)$.

Consider the following variables:

$$\begin{aligned}\mathbf{z}_p(n) &= \sum_{j=1}^{P} \mathbf{C}_{pj}\mathbf{x}_j(n) \\ &= \mathbf{x}_p(n) + \sum_{j=1, j\neq p}^{P} \mathbf{C}_{pj}\mathbf{x}_j(n) \\ &= \mathbf{x}_p(n) - \hat{\mathbf{x}}_p(n),\ p = 1, ..., P, \quad (1.20)\end{aligned}$$

with $\mathbf{C}_{pp} = \mathbf{I}_{L\times L}$ and $\hat{\mathbf{x}}_p(n) = -\sum_{j=1,j\neq p}^{P} \mathbf{C}_{pj}\mathbf{x}_j(n)$. Matrices $\mathbf{C}_{pj}$ are the cross-interpolators obtained by minimizing

$$J_{z_p}(n) = \sum_{i=0}^{n} \lambda^{n-i}\mathbf{z}_p^T(i)\mathbf{z}_p(i),\ p = 1, ..., P, \tag{1.21}$$

and $\mathbf{z}_p(n)$ are the cross-interpolation error vectors.

A general factorization of $\mathbf{R}_{xx}^{-1}(n)$ can be stated as follows:

Lemma 1:

$$\begin{aligned}\mathbf{R}_{xx}^{-1}(n) &= \begin{bmatrix} \mathbf{R}_1^{-1}(n) & \mathbf{0}_{L\times L} & \cdots & \mathbf{0}_{L\times L} \\ \mathbf{0}_{L\times L} & \mathbf{R}_2^{-1}(n) & \cdots & \mathbf{0}_{L\times L} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0}_{L\times L} & \mathbf{0}_{L\times L} & \cdots & \mathbf{R}_P^{-1}(n) \end{bmatrix} \times \\ &\quad \begin{bmatrix} \mathbf{I}_{L\times L} & \mathbf{C}_{12}(n) & \cdots & \mathbf{C}_{1P}(n) \\ \mathbf{C}_{21}(n) & \mathbf{I}_{L\times L} & \cdots & \mathbf{C}_{2P}(n) \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{C}_{P1}(n) & \mathbf{C}_{P2}(n) & \cdots & \mathbf{I}_{L\times L} \end{bmatrix},\end{aligned} \tag{1.22}$$

where

$$\mathbf{R}_p(n) = \sum_{j=1}^{P} \mathbf{C}_{pj}(n)\mathbf{R}_{jp}(n),\ p = 1, 2, ..., P. \tag{1.23}$$

Proof: The proof is rather straightforward by multiplying both sides of (1.22) by $\mathbf{R}_{xx}(n)$ and showing that the result of the right-hand side is equal to the identity matrix with the help of (1.21).

Example: $P = 2$: In this case, we have:

$$\mathbf{z}_1(n) = \mathbf{x}_1(n) + \mathbf{C}_{12}\mathbf{x}_2(n), \tag{1.24}$$
$$\mathbf{z}_2(n) = \mathbf{x}_2(n) + \mathbf{C}_{21}\mathbf{x}_1(n), \tag{1.25}$$

where

$$\mathbf{C}_{12}(n) = -\mathbf{R}_{12}(n)\mathbf{R}_{22}^{-1}(n), \tag{1.26}$$
$$\mathbf{C}_{21}(n) = -\mathbf{R}_{21}(n)\mathbf{R}_{11}^{-1}(n), \tag{1.27}$$

are the cross-interpolators obtained by minimizing $\sum_{i=0}^{n} \lambda^{n-i}\mathbf{z}_1^T(i)\mathbf{z}_1(i)$ and $\sum_{i=0}^{n} \lambda^{n-i}\mathbf{z}_2^T(i)\mathbf{z}_2(i)$. Hence:

$$\begin{aligned}\mathbf{R}^{-1}(n) &= \begin{bmatrix} \mathbf{R}_1^{-1}(n) & \mathbf{0}_{L\times L} \\ \mathbf{0}_{L\times L} & \mathbf{R}_2^{-1}(n) \end{bmatrix} \times \\ &\quad \begin{bmatrix} \mathbf{I}_{L\times L} & -\mathbf{R}_{12}(n)\mathbf{R}_{22}^{-1}(n) \\ -\mathbf{R}_{21}(n)\mathbf{R}_{11}^{-1}(n) & \mathbf{I}_{L\times L} \end{bmatrix},\end{aligned} \tag{1.28}$$

where

$$\mathbf{R}_1(n) = \mathbf{R}_{11}(n) - \mathbf{R}_{12}(n)\mathbf{R}_{22}^{-1}(n)\mathbf{R}_{21}(n), \quad (1.29)$$
$$\mathbf{R}_2(n) = \mathbf{R}_{22}(n) - \mathbf{R}_{21}(n)\mathbf{R}_{11}^{-1}(n)\mathbf{R}_{12}(n), \quad (1.30)$$

are the cross-interpolation error energy matrices or the Schur complements of $\mathbf{R}_{xx}(n)$ with respect to $\mathbf{R}_{22}(n)$ and $\mathbf{R}_{11}(n)$.

From the above result (Lemma 1), we deduce the factorized multichannel RLS:

$$\hat{\mathbf{h}}_{pq}(n) = \hat{\mathbf{h}}_{pq}(n-1) + \mathbf{R}_p^{-1}(n)\mathbf{z}_p(n)e_q(n), \quad (1.31)$$
$$p = 1, 2, ..., P, \; q = 1, 2, ..., Q.$$

4. THE MULTICHANNEL FAST RLS

Because RLS has so far proven to perform better than other algorithms in the MCAEC application [15] a fast calculation scheme of a multichannel version is presented in this section. Compared to standard RLS it has a much lower complexity, $6P^2L + 2PL$ multiplications [instead of $O(P^2L^2)$] for one system output. This algorithm is a numerically stabilized version of the algorithm proposed in [16]. Some extra stability control has to be added so that the algorithm behaves well for a non-stationary speech signal. The following has to be defined:

$$\boldsymbol{\chi}(n) = [x_1(n)\; x_2(n)\; \cdots\; x_P(n)]^T, \; (P \times 1), \quad (1.32)$$
$$\tilde{\mathbf{x}}(n) = [\boldsymbol{\chi}^T(n)\; \boldsymbol{\chi}^T(n-1)\; \cdots\; \boldsymbol{\chi}^T(n-L+1)]^T, \quad (1.33)$$
$$(PL \times 1),$$
$$\tilde{\mathbf{h}}_q(n) = [\hat{h}_{1q,0}(n)\; \hat{h}_{2q,0}(n)\; \cdots\; \hat{h}_{(P-1)q,L-1}(n)\; \hat{h}_{Pq,L-1}(n)]^T,$$
$$(PL \times 1). \quad (1.34)$$

Note that the channels of the filter and state-vector [$\tilde{\mathbf{x}}(n)$] are interleaved in this algorithm. Defined also:

- $\mathbf{A}(n)$, $\mathbf{B}(n)$ = Forward and backward prediction filter matrices, $(PL \times P)$,
- $\mathbf{E}_\mathrm{A}(n)$, $\mathbf{E}_\mathrm{B}(n)$ = Forward and backward prediction error energy matrices, $(P \times P)$,
- $\mathbf{e}_\mathrm{A}(n)$, $\mathbf{e}_\mathrm{B}(n)$ = Forward and backward prediction error vectors, $(P \times 1)$,
- $\mathbf{k}'(n) = \mathbf{R}_{xx}^{-1}(n-1)\tilde{\mathbf{x}}(n)$ = *a priori* Kalman vector, $(PL \times 1)$,
- $\varphi(n)$ = Maximum likelihood related variable, (1×1),

- $\kappa \in [1.5, 2.5]$, Stabilization parameter, (1×1),
- $\lambda \in (0, 1]$, Forgetting factor, (1×1).

The multichannel fast RLS (FRLS) is then:

$$
\begin{aligned}
& && \textit{Prediction}: \\
\mathbf{e}_{\mathrm{A}}(n) &=&& \boldsymbol{\chi}(n) - \mathbf{A}^T(n-1)\tilde{\mathbf{x}}(n-1), \ \ (P \times 1), \\
\varphi_1(n) &=&& \varphi(n-1) + \mathbf{e}_{\mathrm{A}}^T(n)\mathbf{E}_{\mathrm{A}}^{-1}(n-1)\mathbf{e}_{\mathrm{A}}(n), \ \ (1 \times 1), \\
\begin{bmatrix} \mathbf{t}(n) \\ \mathbf{m}(n) \end{bmatrix} &=&& \begin{bmatrix} \mathbf{0}_{P\times 1} \\ \mathbf{k}'(n-1) \end{bmatrix} + \begin{bmatrix} \mathbf{I}_{P\times P} \\ -\mathbf{A}(n-1) \end{bmatrix} \mathbf{E}_{\mathrm{A}}^{-1}(n-1)\mathbf{e}_{\mathrm{A}}(n), \\
& && ((PL+P) \times P), \\
\mathbf{E}_{\mathrm{A}}(n) &=&& \lambda[\mathbf{E}_{\mathrm{A}}(n-1) + \mathbf{e}_{\mathrm{A}}(n)\mathbf{e}_{\mathrm{A}}^T(n)/\varphi(n-1)], \ \ (P \times P), \\
\mathbf{A}(n) &=&& \mathbf{A}(n-1) + \mathbf{k}'(n-1)\mathbf{e}_{\mathrm{A}}^T(n)/\varphi(n-1), \ \ (PL \times P), \\
\mathbf{e}_{\mathrm{B}_1}(n) &=&& \mathbf{E}_{\mathrm{B}}(n-1)\mathbf{m}(n), \ \ (P \times 1), \\
\mathbf{e}_{\mathrm{B}_2}(n) &=&& \boldsymbol{\chi}(n-L) - \mathbf{B}^T(n-1)\tilde{\mathbf{x}}(n), \ \ (P \times 1), \\
\mathbf{e}_{\mathrm{B}}(n) &=&& \kappa\mathbf{e}_{\mathrm{B}_2}(n) + (1-\kappa)\mathbf{e}_{\mathrm{B}_1}(n), \ \ (P \times 1), \\
\mathbf{k}'(n) &=&& \mathbf{t}(n) + \mathbf{B}(n-1)\mathbf{m}(n), \ \ (PL \times 1), \\
\varphi(n) &=&& \varphi_1(n) - \mathbf{e}_{\mathrm{B}_2}^T(n)\mathbf{m}(n), \ \ (1 \times 1), \\
\mathbf{E}_{\mathrm{B}}(n) &=&& \lambda[\mathbf{E}_{\mathrm{B}}(n-1) + \mathbf{e}_{\mathrm{B}_2}(n)\mathbf{e}_{\mathrm{B}_2}^T(n)/\varphi(n)], \ \ (P \times P), \\
\mathbf{B}(n) &=&& \mathbf{B}(n-1) + \mathbf{k}'(n)\mathbf{e}_{\mathrm{B}}^T(n)/\varphi(n), \ \ (PL \times P). \\
& && \textit{Filtering}: \\
e_q(n) &=&& y_q(n) - \tilde{\mathbf{h}}_q^T(n-1)\tilde{\mathbf{x}}(n), \ \ (1 \times 1), \\
\tilde{\mathbf{h}}_q(n) &=&& \tilde{\mathbf{h}}_q(n-1) + \mathbf{k}'(n)e_q(n)/\varphi(n), \ \ (PL \times 1).
\end{aligned}
$$

5. THE MULTICHANNEL LMS ALGORITHM

We derive two different versions of the multichannel LMS algorithm. The first one is straightforward and is a simple generalization of the single-channel LMS. The second one is more sophisticated and takes into account the cross-correlation among all the channels.

5.1 CLASSICAL DERIVATION

The mean-square error criterion is defined as

$$
J_{\mathrm{MS},q} = E\left\{\left[y_q(n) - \mathbf{x}^T(n)\hat{\mathbf{h}}_q\right]^2\right\}, \tag{1.35}
$$

where $E\{\cdot\}$ denotes mathematical expectation. Let $\mathbf{f}(\hat{\mathbf{h}}_q)$ denote the value of the gradient vector with respect to $\hat{\mathbf{h}}_q$. According to the steepest-descent method, the updated value of $\hat{\mathbf{h}}_q$ at time n is computed by using the simple recursive relation [17]:

$$\hat{\mathbf{h}}_q(n) \quad = \quad \hat{\mathbf{h}}_q(n-1) + \frac{\mu}{2}\left\{-\mathbf{f}\left[\hat{\mathbf{h}}_q(n-1)\right]\right\}, \tag{1.36}$$

where μ is positive step-size constant. Differentiating (1.35) with respect to the filter, we get the following value for the gradient vector:

$$\begin{aligned} \mathbf{f}(\hat{\mathbf{h}}_q) &= \begin{bmatrix} \mathbf{f}_1^T(\hat{\mathbf{h}}_q) & \mathbf{f}_2^T(\hat{\mathbf{h}}_q) & \cdots & \mathbf{f}_P^T(\hat{\mathbf{h}}_q) \end{bmatrix}^T \\ &= \partial J_{\mathrm{MS},q}/\partial\hat{\mathbf{h}}_q = -2\mathbf{r}_{xy,q} + 2\mathbf{R}_{xx}\hat{\mathbf{h}}_q, \end{aligned} \tag{1.37}$$

with $\mathbf{r}_{xy,q} = E\{y_q(n)\mathbf{x}(n)\}$ and $\mathbf{R}_{xx} = E\{\mathbf{x}(n)\mathbf{x}^T(n)\}$. By taking $\mathbf{f}(\hat{\mathbf{h}}_q) = \mathbf{0}_{LP\times 1}$, we obtain the Wiener-Hopf equations

$$\mathbf{R}_{xx}\hat{\mathbf{h}}_q = \mathbf{r}_{xy,q}, \tag{1.38}$$

which are similar to the normal equations (1.8) that were derived from a weighted least-squares criterion (1.4). Note that we use the same notation for similar variables that are derived either from the Wiener-Hopf equations or the normal equations.

The steepest-descent algorithm is now:

$$\hat{\mathbf{h}}_q(n) = \hat{\mathbf{h}}_q(n-1) + \mu E\{\mathbf{x}(n)e_q(n)\}, \tag{1.39}$$

and the classical stochastic approximation (consisting of approximating the gradient with its instantaneous value) [17] provides the multichannel LMS algorithm:

$$\hat{\mathbf{h}}_q(n) = \hat{\mathbf{h}}_q(n-1) + \mu\mathbf{x}(n)e_q(n), \tag{1.40}$$

of which the classical mean weight convergence condition under appropriate independence assumption is:

$$0 < \mu < \frac{2}{L\sum_{p=1}^{P}\sigma_{x_p}^2}, \tag{1.41}$$

where the $\sigma_{x_p}^2$ $(p = 1, 2, ..., P)$ are the powers of the input signals. When this condition is satisfied, the weight vector converges in the mean to the optimal Wiener-Hopf solution.

However, the gradient vector corresponding to the filter pq is:

$$\mathbf{f}_p(\hat{\mathbf{h}}_q) = -2\left(\mathbf{r}_{pq} - \sum_{j=1}^{P}\mathbf{R}_{pj}\hat{\mathbf{h}}_{jq}\right),\ p = 1, 2, ..., P, \tag{1.42}$$

which clearly shows some dependency of $\mathbf{f}_p$ on the full vector $\hat{\mathbf{h}}_q$. In other words, the filters $\hat{\mathbf{h}}_{jq}$ with $j \neq p$ influence, in a bad direction, the gradient vector $\mathbf{f}_p$ when seeking the minimum, because the algorithm does not take the cross-correlation among all the inputs into account.

5.2 IMPROVED VERSION

We have seen that during the convergence of the multichannel LMS algorithm, each adaptive filter depends of the others. This dependency must be taken into account. By using this information and Lemma 1, we now differentiate criterion (1.35) with respect to the tap-weight in a different way. The new gradient is obtained by writing that $\hat{\mathbf{h}}_{pq}$ depends of the full vector $\hat{\mathbf{h}}_q$. We get:

$$\begin{aligned} \mathbf{f}_p(\hat{\mathbf{h}}_{pq}) &= \frac{\partial J_{\mathrm{MS},q}}{\partial \hat{\mathbf{h}}_{pq}}(\hat{\mathbf{h}}_q) \\ &= -2E\left\{\mathbf{z}_p(n)\left[y_q(n) - \mathbf{x}^T(n)\hat{\mathbf{h}}_q\right]\right\},\ p = 1, 2, ..., P, \end{aligned} \tag{1.43}$$

with

$$\begin{aligned} \mathbf{z}_p(n) &= \sum_{j=1}^{P} [\partial \hat{\mathbf{h}}_{jq} / \partial \hat{\mathbf{h}}_{pq}]^T \mathbf{x}_j(n) \\ &= \sum_{j=1}^{P} \mathbf{C}_{pj}\mathbf{x}_j(n),\ p = 1, 2, ..., P. \end{aligned} \tag{1.44}$$

We have some interesting orthogonality and decorrelation properties.

Lemma 2:

$$E\{\mathbf{x}_p^T(n)\mathbf{z}_j(n)\} = 0, \tag{1.45}$$

$$E\{\mathbf{z}_p(n)\mathbf{x}_j^T(n)\} = \mathbf{0}_{L\times L},\ \forall p, j = 1, 2, ..., P,\ p \neq j. \tag{1.46}$$

Proof: The proof is straightforward from Lemma 1 (using mathematical expectation instead of weighted least-squares).

We can verify by using Lemma 2 that each gradient vector $\mathbf{f}_p$ ($p = 1, 2, ..., P$) now depends only of the corresponding filter $\hat{\mathbf{h}}_{pq}$. In other words, we make the convergence of each $\hat{\mathbf{h}}_{pq}$ independent of the others, which is not the case in the classical gradient algorithm.

Based on the above gradient vector, the improved steepest-descent algorithm is easily obtained, out of which a stochastic approximation leads to the improved multichannel LMS algorithm:

$$\hat{\mathbf{h}}_q(n) = \hat{\mathbf{h}}_q(n-1) + \mu \mathbf{z}(n) e_q(n) \tag{1.47}$$

with

$$\mathbf{z}(n) = \left[\begin{array}{cccc} \mathbf{z}_1^T(n) & \mathbf{z}_2^T(n) & \cdots & \mathbf{z}_P^T(n) \end{array} \right]^T$$

and

$$0 < \mu < \frac{2}{L \sum_{p=1}^{P} \sigma_{z_p}^2} \tag{1.48}$$

to guaranty the convergence of the algorithm. Note that the improved multichannel LMS algorithm can be seen as an approximation of the factorized multichannel RLS algorithm by taking $\mathbf{R}_p^{-1}(n) \approx \mu \mathbf{I}_{L\times L}$.

5.3 THE MULTICHANNEL APA

The affine projection algorithm (APA) [18] has become popular because of its lower complexity compared to RLS while it converges almost as fast in the single-channel case. Therefore it is interesting to derive and study the multichannel version of this algorithm. Like the multichannel LMS, two versions are derived.

5.4 THE STRAIGHTFORWARD MULTICHANNEL APA

A simple trick for obtaining the single-channel APA is to search for an algorithm of the stochastic gradient type cancelling N *a posteriori* errors [19]. This requirement results in an underdetermined set of linear equations of which the mininum-norm solution is chosen. In the following, this technique is extended to the multichannel case [20].

By definition, the set of N *a priori* errors and N *a posteriori* errors are:

$$\mathbf{e}_q(n) = \mathbf{y}_q(n) - \mathbf{X}^T(n)\hat{\mathbf{h}}_q(n-1), \tag{1.49}$$

$$\mathbf{e}_{\mathrm{a},q}(n) = \mathbf{y}_q(n) - \mathbf{X}^T(n)\hat{\mathbf{h}}_q(n), \tag{1.50}$$

where

$$\mathbf{X}(n) = \left[\begin{array}{cccc} \mathbf{X}_1^T(n) & \mathbf{X}_2^T(n) & \cdots & \mathbf{X}_P^T(n) \end{array} \right]^T$$

is a matrix of size $PL \times N$; the $L \times N$ matrix

$$\mathbf{X}_p(n) = \left[\begin{array}{cccc} \mathbf{x}_p(n) & \mathbf{x}_p(n-1) & \cdots & \mathbf{x}_p(n-N+1) \end{array} \right]$$

is made from the N last input vectors $\mathbf{x}_p(n)$; finally, $\mathbf{y}_q(n)$ and $\mathbf{e}_q(n)$ are respectively vectors of the N last samples of the reference signal $y_q(n)$ and error signal $e_q(n)$.

Using (1.49) and (1.50) plus the requirement that $\mathbf{e}_{\mathrm{a},q}(n) = \mathbf{0}_{N\times 1}$, we obtain:

$$\mathbf{X}^T(n)\Delta\hat{\mathbf{h}}_q(n) = \mathbf{e}_q(n), \tag{1.51}$$

where $\Delta\hat{\mathbf{h}}_q(n) = \hat{\mathbf{h}}_q(n) - \hat{\mathbf{h}}_q(n-1)$.

Equation (1.51) (N equations in PL unknowns, $N \leq PL$) is an underdetermined set of linear equations. Hence, it has an infinite number of solutions, out of which the minimum-norm solution is chosen. This results in [20], [21]:

$$\hat{\mathbf{h}}_q(n) = \hat{\mathbf{h}}_q(n-1) + \mathbf{X}(n)\left[\mathbf{X}^T(n)\mathbf{X}(n)\right]^{-1}\mathbf{e}_q(n). \tag{1.52}$$

However, in this straightforward APA, the normalization matrix $\mathbf{X}^T(n)\mathbf{X}(n) = \sum_{p=1}^{P}\mathbf{X}_p^T(n)\mathbf{X}_p(n)$ does not involve the cross-correlation elements of the P input signals [namely $\mathbf{X}_i^T(n)\mathbf{X}_p(n),\ i,p = 1,2,...,P,\ i \neq p$] and this algorithm may converge slowly.

5.5 THE IMPROVED TWO-CHANNEL APA

A simple way to improve the previous adaptive algorithm is to use the othogonality and decorrelation properties, which will be shown later to appear in this context. Let us derive the improved algorithm by requiring a condition similar to the one used in the improved multichannel LMS. Just use the constraint that $\Delta\hat{\mathbf{h}}_{pq}$ be orthogonal to $\mathbf{X}_j,\ j \neq p$. As a result, we take into account separately the contributions of each input signal. These constraints read:

$$\mathbf{X}_2^T(n)\Delta\hat{\mathbf{h}}_{1q}(n) = \mathbf{0}_{N\times 1}, \tag{1.53}$$

$$\mathbf{X}_1^T(n)\Delta\hat{\mathbf{h}}_{2q}(n) = \mathbf{0}_{N\times 1}, \tag{1.54}$$

and the new set of linear equations characterizing the improved two-channel APA is:

$$\begin{bmatrix} \mathbf{X}_1^T(n) & \mathbf{X}_2^T(n) \\ \mathbf{X}_2^T(n) & \mathbf{0}_{N\times L} \\ \mathbf{0}_{N\times L} & \mathbf{X}_1^T(n) \end{bmatrix} \begin{bmatrix} \Delta\hat{\mathbf{h}}_{1q}(n) \\ \Delta\hat{\mathbf{h}}_{2q}(n) \end{bmatrix} = \begin{bmatrix} \mathbf{e}_q(n) \\ \mathbf{0}_{N\times 1} \\ \mathbf{0}_{N\times 1} \end{bmatrix}. \tag{1.55}$$

The improved two-channel APA algorithm is given by the minimum-norm solution of (1.55) which is found as [20],

$$\Delta\hat{\mathbf{h}}_{1q}(n) = \mathbf{Z}_1(n)\left[\mathbf{Z}_1^T(n)\mathbf{Z}_1(n) + \mathbf{Z}_2^T(n)\mathbf{Z}_2(n)\right]^{-1}\mathbf{e}_q(n), \tag{1.56}$$

$$\Delta\hat{\mathbf{h}}_{2q}(n) = \mathbf{Z}_2(n)\left[\mathbf{Z}_1^T(n)\mathbf{Z}_1(n) + \mathbf{Z}_2^T(n)\mathbf{Z}_2(n)\right]^{-1}\mathbf{e}_q(n), \tag{1.57}$$

where $\mathbf{Z}_p(n)$ is the projection of $\mathbf{X}_p(n)$ onto a subspace orthogonal to $\mathbf{X}_j(n),\ p \neq j$, i.e.,

$$\begin{aligned}\mathbf{Z}_p(n) &= \left\{\mathbf{I}_{L\times L} - \mathbf{X}_j(n)\left[\mathbf{X}_j^T(n)\mathbf{X}_j(n)\right]^{-1}\mathbf{X}_j(n)\right\}\mathbf{X}_p(n), \\ & \quad p,j = 1,2,\ p \neq j.\end{aligned} \tag{1.58}$$

This results in the following orthogonality conditions,

$$\mathbf{X}_p^T(n)\mathbf{Z}_j(n) = \mathbf{0}_{N\times N},\ p \neq j \tag{1.59}$$

which are similar to what appears in the improved multichannel LMS (Lemma 2).

5.6 THE IMPROVED MULTICHANNEL APA

The algorithm explained for two channels is easily generalized to an arbitrary number of channels P. Define the following matrix of size $L \times (P-1)N$:

$$\begin{aligned}\underline{\mathbf{X}}_p(n) &= \left[\begin{array}{cccccc}\mathbf{X}_1(n) & \cdots & \mathbf{X}_{p-1}(n) & \mathbf{X}_{p+1}(n) & \cdots & \mathbf{X}_P(n)\end{array}\right], \\ & \quad p = 1,2,...,P.\end{aligned}$$

The P orthogonality constraints are:

$$\underline{\mathbf{X}}_p^T(n)\Delta\hat{\mathbf{h}}_{pq}(n) = \mathbf{0}_{(P-1)N\times 1},\ p = 1,2,...,P, \tag{1.60}$$

and by using the same steps as for $P = 2$, a solution similar to (1.56), (1.57) is obtained [20]:

$$\Delta\hat{\mathbf{h}}_{pq}(n) = \mathbf{Z}_p(n)\left[\sum_{j=1}^{P}\mathbf{Z}_j^T(n)\mathbf{Z}_j(n)\right]^{-1}\mathbf{e}_q(n),\ p = 1,2,...,P, \tag{1.61}$$

where $\mathbf{Z}_p(n)$ is the projection of $\mathbf{X}_p(n)$ onto a subspace orthogonal to $\underline{\mathbf{X}}_p(n)$, i.e.,

$$\begin{aligned}\mathbf{Z}_p(n) &= \left\{\mathbf{I}_{L\times L} - \underline{\mathbf{X}}_p(n)\left[\underline{\mathbf{X}}_p^T(n)\underline{\mathbf{X}}_p(n)\right]^{-1}\underline{\mathbf{X}}_p(n)\right\}\mathbf{X}_p(n), \\ & \quad p = 1,2,...,P.\end{aligned} \tag{1.62}$$

Note that this equation holds only under the condition $L \geq (P-1)N$, so that the matrix that appears in (1.62) be invertible.

We can easily see that:

$$\underline{\mathbf{X}}_p^T(n)\mathbf{Z}_p(n) = \mathbf{0}_{(P-1)N\times N},\ p = 1,2,...,P. \tag{1.63}$$

Fast versions of the single- and multi-channel APA can be derived [22], [23], [24].

6. THE MULTICHANNEL EXPONENTIATED GRADIENT ALGORITHM

Room acoustic impulse responses are often sparse. Our interest in exponentiated adaptive algorithms is that they converge and track much faster than the LMS algorithm for this family of impulse responses.

One easy way to find adaptive algorithms that adjust the new weight vector $\hat{\mathbf{h}}_q(n+1)$ from the old one $\hat{\mathbf{h}}_q(n)$ is to minimize the following function [25]:

$$J[\hat{\mathbf{h}}_q(n+1)] = d[\hat{\mathbf{h}}_q(n+1), \hat{\mathbf{h}}_q(n)] + \eta e^2_{\mathrm{a},q}(n+1), \tag{1.64}$$

where $d[\hat{\mathbf{h}}_q(n+1), \hat{\mathbf{h}}_q(n)]$ is some measure of distance from the old to the new weight vector,

$$e_{\mathrm{a},q}(n+1) = y_q(n+1) - \hat{\mathbf{h}}_q^T(n+1)\mathbf{x}(n+1) \tag{1.65}$$

is the *a posteriori* error signal, and η is a positive constant. (This formulation is a generalization of the case of Euclidean distance.) The magnitude of η represents the importance of correctness compared to the importance of conservativeness [25]. If η is very small, minimizing $J[\hat{\mathbf{h}}_q(n+1)]$ is close to minimizing $d[\hat{\mathbf{h}}_q(n+1), \hat{\mathbf{h}}_q(n)]$, so that the algorithm makes very small updates. On the other hand, if η is very large, the minimization of $J[\hat{\mathbf{h}}_q(n+1)]$ is almost equivalent to minimizing $d[\hat{\mathbf{h}}_q(n+1), \hat{\mathbf{h}}_q(n)]$ subject to the constraint $e_{\mathrm{a},q}(n+1) = 0$.

To minimize $J[\hat{\mathbf{h}}_q(n+1)]$, we need to set its PL partial derivatives $\partial J[\hat{\mathbf{h}}_q(n+1)]/\partial \hat{h}_{pq,l}(n+1)$ to zero. Hence, the different weight coefficients $\hat{h}_{pq,l}(n+1)$, $l = 0, 1, ..., L-1$, $p = 1, 2, ..., P$, will be found by solving the equations:

$$\frac{\partial d[\hat{\mathbf{h}}_q(n+1), \hat{\mathbf{h}}_q(n)]}{\partial \hat{h}_{pq,l}(n+1)} - 2\eta x_p(n+1-l)e_{\mathrm{a},q}(n+1) = 0. \tag{1.66}$$

Solving (1.66) is in general very difficult. However, if the new weight vector $\hat{\mathbf{h}}_q(n+1)$ is close to the old weight vector $\hat{\mathbf{h}}_q(n)$, replacing the *a posteriori* error signal $e_{\mathrm{a},q}(n+1)$ in (1.66) with the *a priori* error signal $e_q(n+1)$ is a reasonable approximation and the equation

$$\frac{\partial d[\hat{\mathbf{h}}_q(n+1), \hat{\mathbf{h}}_q(n)]}{\partial \hat{h}_{pq,l}(n+1)} - 2\eta x_p(n+1-l)e_q(n+1) = 0 \tag{1.67}$$

is much easier to solve for all distance measures d.

The LMS algorithm is easily obtained from (1.67) by using the squared Euclidean distance

$$d_{\rm E}[\hat{\mathbf{h}}_q(n+1), \hat{\mathbf{h}}_q(n)] = \|\hat{\mathbf{h}}_q(n+1) - \hat{\mathbf{h}}_q(n)\|_2^2. \quad (1.68)$$

The exponentiated gradient (EG) algorithm with positive weights results from using for d the *relative entropy*, also known as *Kullback-Leibler divergence*,

$$d_{\rm re}[\hat{\mathbf{h}}_q(n+1), \hat{\mathbf{h}}_q(n)] = \sum_{p=1}^{P}\sum_{l=0}^{L-1} \hat{h}_{pq,l}(n+1) \ln \frac{\hat{h}_{pq,l}(n+1)}{\hat{h}_{pq,l}(n)}, \quad (1.69)$$

with the constraint $\sum_p \sum_l \hat{h}_{pq,l}(n+1) = 1$, so that (1.67) becomes:

$$\frac{\partial d_{\rm re}[\hat{\mathbf{h}}_q(n+1), \hat{\mathbf{h}}_q(n)]}{\partial \hat{h}_{pq,l}(n+1)} - 2\eta x_p(n+1-l)e_q(n+1) + \gamma = 0, \quad (1.70)$$

where γ is the Lagrange multiplier. Actually, the appropriate constraint should be $\sum_p \sum_l \hat{h}_{pq,l}(n+1) = \sum_p \sum_l h_{pq,l}$ but $\sum_p \sum_l h_{pq,l}$ is not known in practice, so we use the arbitrary value 1 instead.

The algorithm derived from (1.70) is:

$$\hat{h}_{pq,l}(n+1) = \frac{\hat{h}_{pq,l}(n) r_{pq,l}(n+1)}{\sum_{i=1}^{P}\sum_{j=0}^{L-1} \hat{h}_{iq,j}(n) r_{iq,j}(n+1)}, \quad (1.71)$$

where

$$r_{pq,l}(n+1) = \exp\left[2\eta x_p(n+1-l)e_q(n+1)\right]. \quad (1.72)$$

This algorithm is valid only for positive coefficients. To deal with both positive and negative coefficients, we can always find two vectors $\hat{\mathbf{h}}_q^+(n+1)$ and $\hat{\mathbf{h}}_q^-(n+1)$ with positive coefficients, in such a way that the vector

$$\hat{\mathbf{h}}_q(n+1) = \hat{\mathbf{h}}_q^+(n+1) - \hat{\mathbf{h}}_q^-(n+1) \quad (1.73)$$

can have positive and negative components. In this case, the *a posteriori* error signal can be written as:

$$e_{{\rm a},q}(n+1) = y_q(n+1) - [\hat{\mathbf{h}}_q^+(n+1) - \hat{\mathbf{h}}_q^-(n+1)]^T \mathbf{x}(n+1) \quad (1.74)$$

and the function (1.64) will change to:

$$\begin{aligned} J[\hat{\mathbf{h}}_q^+(n+1), \hat{\mathbf{h}}_q^-(n+1)] &= d[\hat{\mathbf{h}}_q^+(n+1), \hat{\mathbf{h}}_q^+(n)] \quad (1.75) \\ &+ d[\hat{\mathbf{h}}_q^-(n+1), \hat{\mathbf{h}}_q^-(n)] + \frac{\eta}{u} e_{{\rm a},q}^2(n+1), \end{aligned}$$

where u is a positive scaling constant. Using the same approximation as before and choosing the Kullback-Leibler divergence plus the constraint $\sum_p \sum_l [\hat{h}^+_{pq,l}(n+1) + \hat{h}^-_{pq,l}(n+1)] = u$, the solutions of the equations

$$\frac{\partial d_{\mathrm{re}}[\hat{\mathbf{h}}^+_q(n+1), \hat{\mathbf{h}}^+_q(n)]}{\partial \hat{h}^+_{pq,l}(n+1)} - 2\frac{\eta}{u} x_p(n+1-l)e_q(n+1) + \gamma = 0, \tag{1.76}$$

$$\frac{\partial d_{\mathrm{re}}[\hat{\mathbf{h}}^-_q(n+1), \hat{\mathbf{h}}^-_q(n)]}{\partial \hat{h}^-_{pq,l}(n+1)} + 2\frac{\eta}{u} x_p(n+1-l)e_q(n+1) + \gamma = 0, \tag{1.77}$$

give the so-called EG± algorithm:

$$\hat{h}^+_{pq,l}(n+1) = u\frac{\hat{h}^+_{pq,l}(n) r^+_{pq,l}(n+1)}{s_q(n+1)}, \tag{1.78}$$

$$\hat{h}^-_{pq,l}(n+1) = u\frac{\hat{h}^-_{pq,l}(n) r^-_{pq,l}(n+1)}{s_q(n+1)}, \tag{1.79}$$

where

$$s_q(n+1) = \sum_{i=1}^{P}\sum_{j=0}^{L-1}\left[\hat{h}^+_{iq,j}(n) r^+_{iq,j}(n+1) + \hat{h}^-_{iq,j}(n) r^-_{iq,j}(n+1)\right], \tag{1.80}$$

$$r^+_{pq,l}(n+1) = \exp\left[\frac{2\eta}{u} x_p(n+1-l)e_q(n+1)\right], \tag{1.81}$$

$$\begin{aligned} r^-_{pq,l}(n+1) &= \exp\left[-\frac{2\eta}{u} x_p(n+1-l)e_q(n+1)\right] \\ &= \frac{1}{r^+_{pq,l}(n+1)}, \end{aligned} \tag{1.82}$$

$$e_q(n+1) = y_q(n+1) - [\hat{\mathbf{h}}^+_q(n) - \hat{\mathbf{h}}^-_q(n)]^T \mathbf{x}(n+1). \tag{1.83}$$

We can check that we always have $\|\hat{\mathbf{h}}^+_q(n+1)\|_1 + \|\hat{\mathbf{h}}^-_q(n+1)\|_1 = u$. Upon convergence:

$$\begin{aligned} \|\hat{\mathbf{h}}_q(\infty)\|_1 &= \|\mathbf{h}_q\|_1 \\ &= \|\hat{\mathbf{h}}^+_q(\infty) - \hat{\mathbf{h}}^-_q(\infty)\|_1 \\ &\leq \|\hat{\mathbf{h}}^+_q(\infty)\|_1 + \|\hat{\mathbf{h}}^-_q(\infty)\|_1 = u, \end{aligned} \tag{1.84}$$

hence, the constant u should be chosen such that $u \geq \|\mathbf{h}_q\|_1$.

A normalized version of the multichannel EG± algorithm is given below:

$$
\begin{aligned}
& & & Initialization: \\
\hat{h}^{+}_{pq,l}(0) &= & & \hat{h}^{-}_{pq,l}(0) = c > 0,\ p = 1,2,...,P,\ l = 0,1,...,L-1. \\
& & & Parameters: \\
u &\geq & & \|\mathbf{h}_q\|_1, \\
0 < \alpha & & & \leq 1, \quad \delta > 0. \\
& & & Error: \\
e_q(n+1) &= & & y_q(n+1) - [\hat{\mathbf{h}}^{+}_q(n) - \hat{\mathbf{h}}^{-}_q(n)]^T \mathbf{x}(n+1). \\
& & & Update: \\
\mu(n+1) &= & & \frac{\alpha}{\mathbf{x}^T(n+1)\mathbf{x}(n+1) + \delta}, \\
r^{+}_{pq,l}(n+1) &= & & \exp\left[PL\frac{\mu(n+1)}{u} x_p(n+1-l) e_q(n+1)\right], \\
r^{-}_{pq,l}(n+1) &= & & \frac{1}{r^{+}_{pq,l}(n+1)}, \\
s_q(n+1) &= & & \sum_{i=1}^{P}\sum_{j=0}^{L-1}\left[\hat{h}^{+}_{iq,j}(n) r^{+}_{iq,j}(n+1) + \hat{h}^{-}_{iq,j}(n) r^{-}_{iq,j}(n+1)\right], \\
\hat{h}^{+}_{pq,l}(n+1) &= & & u\frac{\hat{h}^{+}_{pq,l}(n) r^{+}_{pq,l}(n+1)}{s_q(n+1)}, \\
\hat{h}^{-}_{pq,l}(n+1) &= & & u\frac{\hat{h}^{-}_{pq,l}(n) r^{-}_{pq,l}(n+1)}{s_q(n+1)}, \\
& & & p = 1,2,...,P,\ l = 0,1,...,L-1.
\end{aligned}
$$

Intuitively, exponentiating the update has the effect of assigning larger relative updates to larger weights, thereby deemphasizing the effect of smaller weights. This is qualitatively similar to the PNLMS algorithm [26] which makes the update *proportional* to the size of the weight. This type of behavior is desirable for sparse impulse responses where small weights do not contribute significantly to the *mean* solution but introduce an undesirable noise-like *variance*.

Recently, the proportionate normalized least-mean-square (PNLMS) algorithm was developed for use in network echo cancelers [26]. In comparison to the NLMS algorithm, PNLMS has very fast initial convergence and tracking when the echo path is sparse. As previously mentioned, the idea behind PNLMS is to update each coefficient of the filter independently of the others by adjusting the adaptation step size in proportion to the estimated filter coefficient. More recently, an improved PNLMS (IPNLMS) [27] was proposed that performs better than NLMS and PNLMS, whatever the nature of the impulse response is. The IPNLMS is summarized below:

$$
\begin{aligned}
& && Initialization: \\
\hat{h}_{pq,l}(0) &=& & 0,\ p=1,2,...,P,\ l=0,1,...,L-1. \\
& && Parameters: \\
0 < & \alpha & & \leq 1,\ \delta_{\text{IPNLMS}} > 0, \\
-1 & \leq & & \kappa \leq 1, \\
& \varepsilon & & > 0 \text{ (small number to avoid division by zero).} \\
& && Error: \\
e_q(n+1) &=& & y_q(n+1) - \hat{\mathbf{h}}_q^T(n)\mathbf{x}(n+1). \\
& && Update: \\
g_{pq,l}(n) &=& & \frac{1-\kappa}{2PL} + (1+\kappa)\frac{|\hat{h}_{pq,l}(n)|}{2\|\hat{\mathbf{h}}_q(n)\|_1 + \varepsilon}, \\
p &=& & 1,2,...,P,\ l=0,1,...,L-1, \\
\mu(n+1) &=& & \frac{\alpha}{\sum_{i=1}^{P}\sum_{j=0}^{L-1} x_i^2(n+1-j)g_{iq,j}(n) + \delta_{\text{IPNLMS}}}, \\
\hat{h}_{pq,l}(n+1) &=& & \hat{h}_{pq,l}(n) + \mu(n+1)g_{pq,l}(n)x_p(n+1-l)e_q(n+1), \\
p &=& & 1,2,...,P,\ l=0,1,...,L-1.
\end{aligned}
$$

In general, $g_{pq,l}$ in the IPNLMS provides the "proportionate" scaling of the update. The parameter κ controls the amount of proportionality in the update. For $\kappa = -1$, it can easily be checked that the IPNLMS and NLMS algorithms are identical. For κ close to 1, the IPNLMS behaves like the PNLMS algorithm [26]. In practice, a good choice for κ is 0 or -0.5.

We can show that the IPNLMS and EG± algorithms are related [28]. The IPNLMS is in fact an approximation of the EG± if we approximate in the latter $\exp(a) \approx 1 + a$ for $|a| \ll 1$.

7. THE MULTICHANNEL FREQUENCY-DOMAIN ADAPTIVE ALGORITHM

Adaptive algorithms in the frequency domain are, in general, extremely efficient since they use the fast Fourier transform (FFT) as an intermediary step. As a result, they are now implemented in many prototypes and products for acoustic echo cancellation. In this section, we briefly explain how these algorithms can be derived rigorously from a block error signal.

From now on and for simplification, we drop the parameter q in all equations. With this simplification, the error signal at time n is now:

$$e(n) = y(n) - \sum_{p=1}^{P} \hat{\mathbf{h}}_p^T \mathbf{x}_p(n), \tag{1.85}$$

where $\hat{\mathbf{h}}_p$ is the estimated impulse response of the pth channel,

$$\hat{\mathbf{h}}_p = \left[\begin{array}{cccc} \hat{h}_{p,0} & \hat{h}_{p,1} & \cdots & \hat{h}_{p,L-1} \end{array} \right]^T.$$

We now define the block error signal (of length $N \leq L$). For that, we assume that L is an integer multiple of N, i.e., $L = KN$. We have:

$$\begin{aligned} \mathbf{e}(m) &= \mathbf{y}(m) - \hat{\mathbf{y}}(m) \\ &= \mathbf{y}(m) - \sum_{p=1}^{P} \mathbf{X}_p^T(m)\hat{\mathbf{h}}_p, \end{aligned} \tag{1.86}$$

where m is the block time index, and

$$\begin{aligned} \mathbf{e}(m) &= \left[\begin{array}{ccc} e(mN) & \cdots & e(mN+N-1) \end{array} \right]^T, \\ \mathbf{y}(m) &= \left[\begin{array}{ccc} y(mN) & \cdots & y(mN+N-1) \end{array} \right]^T, \\ \mathbf{X}_p(m) &= \left[\begin{array}{ccc} \mathbf{x}_p(mN) & \cdots & \mathbf{x}_p(mN+N-1) \end{array} \right], \\ \hat{\mathbf{y}}(m) &= \left[\begin{array}{ccc} \hat{y}(mN) & \cdots & \hat{y}(mN+N-1) \end{array} \right]^T. \end{aligned}$$

It can easily be checked that $\mathbf{X}_p$ is a Toeplitz matrix of size $(L \times N)$.

We can show that for $K = L/N$, we can write

$$\mathbf{X}_p^T(m)\hat{\mathbf{h}}_p = \sum_{k=0}^{K-1} \mathbf{T}_p(m-k)\hat{\mathbf{h}}_{p,k}, \tag{1.87}$$

where $\mathbf{T}(m-k)$ is an $(N \times N)$ Toeplitz matrix and

$$\hat{\mathbf{h}}_{p,k} = \left[\begin{array}{cccc} \hat{h}_{p,kN} & \hat{h}_{p,kN+1} & \cdots & \hat{h}_{p,kN+N-1} \end{array} \right]^T, \ k = 0, 1, ..., K-1,$$

are the sub-filters of $\hat{\mathbf{h}}_p$. In (1.87), the filter $\hat{\mathbf{h}}_p$ (of length L) is partitioned into K sub-filters $\hat{\mathbf{h}}_{p,k}$ of length N and the rectangular matrix $\mathbf{X}_p^T$ [of size $(N \times L)$] is decomposed to K square sub-matrices of size $(N \times N)$.

It is well known that a Toeplitz matrix $\mathbf{T}_p$ can be transformed, by doubling its size, to a circulant matrix $\mathbf{C}_p$. Also, a circulant matrix is easily decomposed as follows: $\mathbf{C}_p = \mathbf{F}_{2N\times 2N}^{-1}\mathbf{D}_p\mathbf{F}_{2N\times 2N}$, where $\mathbf{F}_{2N\times 2N}$ is the Fourier matrix [of size $(2N \times 2N)$] and $\mathbf{D}_p$ is a diagonal matrix whose elements are the discrete Fourier transform of the first column of $\mathbf{C}_p$. If we multiply (1.86) by $\mathbf{F}_{N\times N}$ [Fourier matrix of size $(N \times N)$], we get the error signal in the frequency domain (denoted by underbars):

$$\begin{aligned}
\underline{\mathbf{e}}(m) &= \underline{\mathbf{y}}(m) - \mathbf{G}_{N\times 2N}^{01}\sum_{p=1}^{P}\sum_{k=0}^{K-1}\mathbf{D}_p(m-k)\mathbf{G}_{2N\times N}^{10}\underline{\hat{\mathbf{h}}}_{p,k} \\
&= \underline{\mathbf{y}}(m) - \mathbf{G}_{N\times 2N}^{01}\sum_{p=1}^{P}\sum_{k=0}^{K-1}\mathbf{U}_p(m-k)\underline{\hat{\mathbf{h}}}_{p,k} \\
&= \underline{\mathbf{y}}(m) - \mathbf{G}_{N\times 2N}^{01}\sum_{p=1}^{P}\underline{\mathbf{U}}_p(m)\underline{\hat{\mathbf{h}}}_p \\
&= \underline{\mathbf{y}}(m) - \mathbf{G}_{N\times 2N}^{01}\underline{\mathbf{U}}(m)\underline{\hat{\mathbf{h}}}, \qquad (1.88)
\end{aligned}$$

where

$$\begin{aligned}
\underline{\mathbf{e}}(m) &= \mathbf{F}_{N\times N}\mathbf{e}(m), \\
\underline{\mathbf{y}}(m) &= \mathbf{F}_{N\times N}\mathbf{y}(m), \\
\mathbf{G}_{N\times 2N}^{01} &= \mathbf{F}_{N\times N}\mathbf{W}_{N\times 2N}^{01}\mathbf{F}_{2N\times 2N}^{-1}, \\
\mathbf{W}_{N\times 2N}^{01} &= \begin{bmatrix} \mathbf{0}_{N\times N} & \mathbf{I}_{N\times N} \end{bmatrix}, \\
\mathbf{G}_{2N\times N}^{10} &= \mathbf{F}_{2N\times 2N}\mathbf{W}_{2N\times N}^{10}\mathbf{F}_{N\times N}^{-1}, \\
\mathbf{W}_{2N\times N}^{10} &= \begin{bmatrix} \mathbf{I}_{N\times N} \\ \mathbf{0}_{N\times N} \end{bmatrix}, \\
\underline{\hat{\mathbf{h}}}_{p,k} &= \mathbf{F}_{N\times N}\hat{\mathbf{h}}_{p,k}, \\
\underline{\hat{\mathbf{h}}}_p &= \begin{bmatrix} \underline{\hat{\mathbf{h}}}_{p,0}^T & \underline{\hat{\mathbf{h}}}_{p,1}^T & \cdots & \underline{\hat{\mathbf{h}}}_{p,K-1}^T \end{bmatrix}^T, \\
\underline{\hat{\mathbf{h}}} &= \begin{bmatrix} \underline{\hat{\mathbf{h}}}_1^T & \underline{\hat{\mathbf{h}}}_2^T & \cdots & \underline{\hat{\mathbf{h}}}_P^T \end{bmatrix}^T, \\
\mathbf{D}_p(m-k) &= \mathbf{F}_{2N\times 2N}\mathbf{C}_p(m-k)\mathbf{F}_{2N\times 2N}^{-1}, \\
\mathbf{U}_p(m-k) &= \mathbf{D}_p(m-k)\mathbf{G}_{2N\times N}^{10}, \\
\underline{\mathbf{U}}_p(m) &= \begin{bmatrix} \mathbf{U}_p(m) & \mathbf{U}_p(m-1) & \cdots & \mathbf{U}_p(m-K+1) \end{bmatrix},
\end{aligned}$$

$$\begin{aligned}
\underline{\mathbf{U}}(m) &= \begin{bmatrix} \underline{\mathbf{U}}_1(m) & \underline{\mathbf{U}}_2(m) & \cdots & \underline{\mathbf{U}}_P(m) \end{bmatrix}, \\
\underline{\mathbf{D}}_p(m) &= \begin{bmatrix} \mathbf{D}_p(m) & \mathbf{D}_p(m-1) & \cdots & \mathbf{D}_p(m-K+1) \end{bmatrix}, \\
\underline{\mathbf{D}}(m) &= \begin{bmatrix} \underline{\mathbf{D}}_1(m) & \underline{\mathbf{D}}_2(m) & \cdots & \underline{\mathbf{D}}_P(m) \end{bmatrix}, \\
\mathbf{G}^{10}_{2PL\times PL} &= \mathrm{diag}\begin{bmatrix} \mathbf{G}^{10}_{2N\times N} & \cdots & \mathbf{G}^{10}_{2N\times N} \end{bmatrix}, \\
\underline{\mathbf{U}}(m) &= \underline{\mathbf{D}}(m)\mathbf{G}^{10}_{2PL\times PL}.
\end{aligned}$$

The size of the matrix $\underline{\mathbf{U}}$ is $(2N \times PL)$ and the length of $\hat{\underline{\mathbf{h}}}$ is PL.

By minimizing the criterion

$$J_{\mathrm{f}}(m) = (1-\lambda)\sum_{i=0}^{m} \lambda^{m-i}\underline{\mathbf{e}}^H(i)\underline{\mathbf{e}}(i) \ , \tag{1.89}$$

where ^{H} denotes conjugate transpose and λ $(0 < \lambda < 1)$ is an exponential forgetting factor, we obtain the normal equations for the multichannel case:

$$\mathbf{S}(m)\hat{\underline{\mathbf{h}}}(m) = \mathbf{s}(m), \tag{1.90}$$

where

$$\begin{aligned}
\mathbf{S}(m) &= \lambda\mathbf{S}(m-1) \\
&+ (1-\lambda)(\mathbf{G}^{10}_{2PL\times PL})^H\underline{\mathbf{D}}^H(m)\mathbf{G}^{01}_{2N\times 2N}\underline{\mathbf{D}}(m)\mathbf{G}^{10}_{2PL\times PL}
\end{aligned} \tag{1.91}$$

is a $(PL \times PL)$ matrix,

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

and

$$\mathbf{s}(m) = \lambda\mathbf{s}(m-1) + (1-\lambda)(\mathbf{G}^{10}_{2PL\times PL})^H\underline{\mathbf{D}}^H(m)\underline{\mathbf{y}}_{2N}(m) \tag{1.92}$$

is a $(PL\times 1)$ vector. Assuming that the P input signals are not perfectly pairwise coherent, the normal equations have a unique solution which is the optimal Wiener solution.

Define the following variables:

$$\begin{aligned}
\underline{\mathbf{y}}_{2N}(m) &= (\mathbf{G}^{01}_{N\times 2N})^H\underline{\mathbf{y}}(m) \\
&= \mathbf{F}_{2N\times 2N}\begin{bmatrix} \mathbf{0}_{N\times 1} \\ \mathbf{y}(m) \end{bmatrix}, \\
\underline{\mathbf{e}}_{2N}(m) &= \mathbf{F}_{2N\times 2N}\begin{bmatrix} \mathbf{0}_{N\times 1} \\ \mathbf{e}(m) \end{bmatrix} \\
&= (\mathbf{G}^{01}_{N\times 2N})^H\underline{\mathbf{e}}(m), \\
\mathbf{G}^{10}_{2PL\times 2PL} &= \mathrm{diag}\begin{bmatrix} \mathbf{G}^{10}_{2N\times 2N} & \cdots & \mathbf{G}^{10}_{2N\times 2N} \end{bmatrix}, \\
\hat{\underline{\mathbf{h}}}_{2PL}(m) &= \mathbf{G}^{10}_{2PL\times PL}\hat{\underline{\mathbf{h}}}(m).
\end{aligned}$$

We can show that from the normal equations, we can exactly derive the following multichannel frequency-domain adaptive algorithm:

$$\mathbf{Q}(m) = \lambda \mathbf{Q}(m-1) + (1-\lambda)\underline{\mathbf{D}}^H(m)\mathbf{G}^{01}_{2N\times 2N}\underline{\mathbf{D}}(m), \quad (1.93)$$

$$\underline{\mathbf{e}}_{2N}(m) = \underline{\mathbf{y}}_{2N}(m) - \mathbf{G}^{01}_{2N\times 2N}\underline{\mathbf{D}}(m)\underline{\hat{\mathbf{h}}}_{2PL}(m-1), \quad (1.94)$$

$$\underline{\hat{\mathbf{h}}}_{2PL}(m) = \underline{\hat{\mathbf{h}}}_{2PL}(m-1) + (1-\lambda)\mathbf{G}^{10}_{2PL\times 2PL}\mathbf{Q}^{-1}(m)\underline{\mathbf{D}}^H(m)\underline{\mathbf{e}}_{2N}(m). \quad (1.95)$$

Depending of the approximations we make on matrix $\mathbf{Q}(m)$, we obtain different algorithms. There is a compromise between performance (convergence rate) and complexity. Several algorithms can be deduced from the previous form, some of them are well-known while others are new. See [29] for a general derivation of adaptive algorithms in the frequency domain. See also [30] and [31] for new algorithms derived directly from the above equations.

8. CONCLUSIONS

In this chapter, we have given an overview on multichannel adaptive algorithms in the context of multichannel acoustic echo cancellation. We have first derived the normal equations of a MIMO system and discussed the identification problem. We have shown that, in the multichannel case and when the input signals are linearly related, there is a nonuniqueness problem that does not exist in the single-input single-output (SISO) case. In order to have a unique solution, we have to decorrelate somehow the input signals without affecting the spatialization effect and the quality of the signals. We have then derived many useful adaptive algorithms in a context where the strong correlation between the input signals affects seriously the performance of several well-known algorithms.

References

[1] M. M. Sondhi, D. R. Morgan, and J. L. Hall, "Stereophonic acoustic echo cancellation—An overview of the fundamental problem," *IEEE Signal Processing Lett.*, vol. 2, pp. 148–151, Aug. 1995.

[2] 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. Speech Audio Processing*, vol. 6, pp. 156–165, Mar. 1998.

[3] T. Gänsler and J. Benesty, "New insights into the stereophonic acoustic echo cancellation problem and an adaptive nonlinearity solution," *IEEE Trans. Speech Audio Processing*, vol. 10, pp. 257–267, July 2002.

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

[5] A. Gilloire and V. Turbin, "Using auditory properties to improve the behavior of stereophonic acoustic echo cancellers," in *Proc. IEEE ICASSP*, 1998, pp. 3681–3684.

[6] S. Shimauchi, Y. Haneda, S. Makino, and Y. Kaneda, "New configuration for a stereo echo canceller with nonlinear pre-processing," in *Proc. IEEE ICASSP*, 1998, pp. 3685–3688.

[7] M. Ali, "Stereophonic echo cancellation system using time-varying all-pass filtering for signal decorrelation," in *Proc. IEEE ICASSP*, 1998, pp. 3689–3692.

[8] Y. Joncour and A. Sugiyama, "A stereo echo canceller with pre-processing for correct echo path identification," in *Proc. IEEE ICASSP*, 1998, pp. 3677–3680.

[9] J. Benesty, T. Gänsler, and P. Eneroth, "Multi-channel sound, acoustic echo cancellation, and multi-channel time-domain adaptive filtering," in *Acoustic Signal Processing for Telecommunication*, S. L. Gay and J. Benesty, Eds., Kluwer Academic Publishers, Boston, 2000, Chap. 6.

[10] T. Gänsler and J. Benesty, "Stereophonic acoustic echo cancellation and two-channel adaptive filtering: an overview," *Int. J. Adapt. Control Signal Process.*, vol. 14, pp. 565–586, Sept. 2000.

[11] B. C. J. Moore, *An introduction to the Psychology of Hearing*. Academic Press, London, 1989, Chap. 3.

[12] D. R. Morgan, J. L. Hall, and J. Benesty, "Investigation of several types of nonlinearities for use in stereo acoustic echo cancellation," *IEEE Trans. Speech Audio Processing*, vol. 9, pp. 686–696, Sept. 2001.

[13] J. Benesty, D. R. Morgan, J. L. Hall, and M. M. Sondhi, "Stereophonic acoustic echo cancellation using nonlinear transformations and comb filtering," in *Proc. IEEE ICASSP*, 1998, pp. 3673–3676.

[14] J. Benesty, D. R. Morgan, and M. M. Sondhi, "A hybrid mono/stereo acoustic echo canceler," *IEEE Trans. Speech Audio Processing*, vol. 6, pp. 468–475, Sept. 1998.

[15] J. Benesty, F. Amand, A. Gilloire, and Y. Grenier, "Adaptive filtering algorithms for stereophonic acoustic echo cancellation," in *Proc. IEEE ICASSP*, 1995, pp. 3099–3102.

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

[17] S. Haykin, *Adaptive Filter Theory*. Fourth Edition, Prentice Hall, Upper Saddle River, N.J., 2002.

[18] K. Ozeki and T. Umeda, "An adaptive filtering algorithm using an orthogonal projection to an affine subspace and its properties," *Elec. Comm. Japan*, vol. J67-A, pp. 126–132, Feb. 1984.

[19] M. Montazeri and P. Duhamel, "A set of algorithms linking NLMS and block RLS algorithms," *IEEE Trans. Signal Processing*, vol. 43, pp. 444–453, Feb. 1995.

[20] J. Benesty, P. Duhamel, and Y. Grenier, "A multi-channel affine projection algorithm with applications to multi-channel acoustic echo cancellation," *IEEE Signal Processing Lett.*, vol. 3, pp. 35–37, Feb. 1996.

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

[22] S. L. Gay and S. Tavathia, "The fast affine projection algorithm," in *Proc. IEEE ICASSP*, 1995, pp. 3023–3026.

[23] M. Tanaka, Y. Kaneda, S. Makino, and J. Kojima "Fast projection algorithm and its step size control," in *Proc. IEEE ICASSP*, 1995, pp. 945–948.

[24] F. Amand, J. Benesty, A. Gilloire, and Y. Grenier, "A fast two-channel affine projection algorithm for stereophonic acoustic echo cancellation," in *Proc. IEEE ICASSP*, 1996, pp. 949–952.

[25] J. Kivinen and M. K. Warmuth, "Exponentiated gradient versus gradient descent for linear predictors," *Inform. Comput.*, vol. 132, pp. 1–64, Jan. 1997.

[26] D. L. Duttweiler, "Proportionate normalized least mean square adaptation in echo cancelers," *IEEE Trans. Speech Audio Processing*, vol. 8, pp. 508–518, Sept. 2000.

[27] J. Benesty and S. L. Gay, "An improved PNLMS algorithm," in *Proc. IEEE ICASSP*, 2002, pp. 1881–1884.

[28] J. Benesty, Y. Huang, and D. R. Morgan, "On a class of exponentiated adaptive algorithms for the identification of sparse impulse responses," in *Adaptive Signal Processing—Applications to Real-World Problems*, J. Benesty and Y. Huang, Eds., Springer-Verlag, Berlin, 2003, Chap. 1.

[29] J. Benesty, T. Gänsler, D. R. Morgan, M. M. Sondhi, and S. L. Gay, *Advances in Network and Acoustic Echo cancellation*. Springer-Verlag, Berlin, 2001.

[30] H. Buchner, J. Benesty, and W. Kellermann, "Multichannel frequency-domain adaptive filtering with application to multichannel acoustic echo cancellation," in *Adaptive Signal Processing—Applications to Real-World Problems*, J. Benesty and Y. Huang, Eds., Springer-Verlag, Berlin, 2003, Chap. 4.

[31] H. Buchner, J. Benesty, and W. Kellermann, "An extended multidelay filter: fast low-delay algorithms for very high-order adaptive systems," in *Proc. IEEE ICASSP*, 2003, pp. V-385–V-388.