

Adaptive Filtering Approach to Multichannel Audio Reproduction Systems

John Garas* and Piet Sommen
Eindhoven University of Technology
E-Hoog 3.34, P.O. Box 513
5600 MB Eindhoven, the Netherlands
Email: P.C.W.Sommen@tue.nl

1 Foreword

One of the research areas of the Signal Processing group at the Eindhoven University of Technology is in *adaptive array signal processing*. It is concerned with the application of adaptive signal processing techniques to the outputs of a spatially distributed array of sensors. The power of these techniques lies in the possibility of achieving improvements in directivity and measured Signal-to-Noise Ratio (SNR) as compared to that attainable with a single input sensor. The spatial dimensions introduced by processing a distributed array of sensors allow directional discrimination of signals against interference. Sensor arrays are used in many applications including radar, sonar, seismology and tomography [5]. Most of these applications use the fact that the used signals are band limited and that the sources are so far away that at the sensors they can be treated as a flat wavefront. In many (certainly 'in-the-house') audio applications these narrowband and far field assumptions certainly do not hold. Mainly due to the vast increase of available computational power, commercial applications of array processing advanced audio systems have become feasible in recent years.

Recently two PhD thesis [4, 11] have been successfully completed in the Signal Processing group in which adaptive array processing techniques were applied to audio applications. The work of [11] was focused around a part of the

transparent audio communication project. The term 'transparent communication' refers to audio communication which is free from recording and transmission artifacts, such as reverberation, noise and acoustic echoes. Also, it is desirable to separate speech signals of multiple speakers who are speaking simultaneously.

The current paper, based on the results of the PhD thesis work outlined in [4], discusses theoretical and implementation issues concerning adaptive multichannel audio reproduction systems.

2 Introduction

Control filters in loudspeaker displays must implement both binaural synthesis and cross-talk cancellation functions. The cross-talk cancellation subsystem is the inverse of an often non-minimum phase and ill-conditioned matrix of electro-acoustic transfer functions. An exact solution to the cross-talk canceller is difficult to calculate, and in some cases, it may not be possible at certain frequencies. Alternative to exact direct inversion, a statistical least mean square solution may be obtained for the matrix inverse. The advantage of such an approach is that it can be made adaptive, so that the filters can be designed *in-situ*. Such an adaptive solution also offers the possibility of tracking and correcting the filters when changes in the electro-acoustic system occur.

An adaptive loudspeaker display that creates

*John Garas is currently with TNO-TPD Delft

a single sound image at the ears of one listener requires two loudspeakers and two microphones placed inside the listener's ears. More microphones are needed as the number of listeners increases, and more filters are needed as the number of virtual sound images increases. Instead of treating any of the above special cases, a generalised model is introduced in Section 3. In addition to generalising the number of sound sources, loudspeakers and microphones used in the reproduction system, it will be shown in Section 4 that the model is capable of describing a wide class of applications including synthesis of virtual sound sources, cross-talk cancellation, and active noise control.

The optimum least mean square solution for the control filters in the generalised model is derived in Section 5. Although important from the numerical analysis point of view, real-time implementation of the system would require approaching the optimum solution using an iterative algorithm. The Multiple Error Filtered- X Least Mean Square (MEFX) algorithm, is such an algorithm and is introduced in Section 6.

Several implementation details are discussed by considering the implementation of the MEFX algorithm in active noise control, virtual sound source synthesis, and cross-talk cancellation applications in Sections 7, 8, and 9, respectively.

In reverberant acoustic environments, the impulse response between two points may last for several hundreds of milliseconds. At the standard audio compact disc sampling frequency of 44.1 kHz, thousands of FIR filter coefficients are needed to properly model and store such an impulse response. Processing many of these transfer functions requires a huge amount of computational power. Reducing the system complexity is, therefore, essential for real-time implementation. Efficient implementations using the Adjoint LMS and its Block Frequency Domain version are discussed in Section 10.

3 A Generalised Model

A generalised block diagram of a multichannel audio reproduction system is shown in

Fig. 1. A set of L reproduction loudspeakers $\{S_1, S_2, \dots, S_L\}$ are used to play K pre-recorded audio signals defined at the sample index n by

$$\underline{\mathbf{x}}(n) = \begin{bmatrix} x_1(n) & x_2(n) & \dots & x_K(n) \end{bmatrix}^T$$

The sound field generated due to these loudspeakers is required to be controlled at the proximity of a set of M microphones (receivers) $\{R_1, R_2, \dots, R_M\}$ to a set of desired values defined as

$$\underline{\mathbf{d}}(n) = \begin{bmatrix} d_1(n) & d_2(n) & \dots & d_M(n) \end{bmatrix}^T$$

Sound waves emitted from the loudspeakers are filtered through the $[M \times L]$ matrix of electro-acoustic transfer functions $\mathbf{C}(\omega)$ before reaching the microphones' positions. The matrix $\mathbf{C}(\omega)$ contains the Fourier transforms of the impulse responses $\{c_{ml} : m = 1, 2, \dots, M, l = 1, 2, \dots, L\}$ evaluated at a frequency ω and defined as

$$\mathbf{C}(\omega) = \begin{bmatrix} C_{11}(\omega) & \dots & C_{1L}(\omega) \\ C_{21}(\omega) & \dots & C_{2L}(\omega) \\ \vdots & \ddots & \vdots \\ C_{M1}(\omega) & \dots & C_{ML}(\omega) \end{bmatrix}$$

where $C_{ml}(\omega)$ is the electro-acoustic transfer function between the m^{th} microphone R_m and the l^{th} reproduction loudspeaker S_l calculated at ω . The above mentioned control task is achieved by introducing an $[L \times K]$ matrix of (adaptive) digital filters $\mathbf{W}(\omega)$ in the reproduction chain defined at a frequency ω as

$$\mathbf{W}(\omega) = \begin{bmatrix} W_{11}(\omega) & \dots & W_{1K}(\omega) \\ W_{21}(\omega) & \dots & W_{2K}(\omega) \\ \vdots & \ddots & \vdots \\ W_{L1}(\omega) & \dots & W_{LK}(\omega) \end{bmatrix}$$

where $W_{lk}(\omega)$ is the filter driving the l^{th} reproduction loudspeaker S_l and has $x_k(n)$ as its input. The filters $\mathbf{W}(\omega)$ are designed to produce the control signals $\underline{\mathbf{y}}(n)$ to drive the L reproduction loudspeakers such that the resulting sound field at the microphones $\hat{\underline{\mathbf{d}}}(n)$ is as close as possible to the desired sound field $\underline{\mathbf{d}}(n)$. The vectors $\underline{\mathbf{y}}(n)$

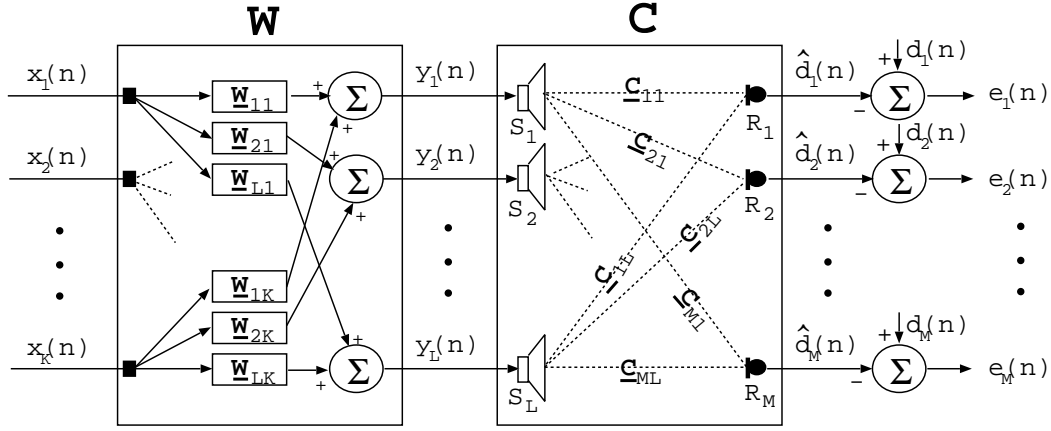


Figure 1: Generalised block diagram of multichannel audio reproduction system.

and $\hat{\underline{d}}(n)$ are defined as

$$\underline{y}(n) = \begin{bmatrix} y_1(n) & y_2(n) & \cdots & y_L(n) \end{bmatrix}^T \quad (1)$$

$$\hat{\underline{d}}(n) = \begin{bmatrix} \hat{d}_1(n) & \hat{d}_2(n) & \cdots & \hat{d}_M(n) \end{bmatrix}^T$$

The set of desired sound fields given by the vector $\underline{d}(n)$ is usually well correlated with the set of K input signals $\underline{x}(n)$. Therefore, $\underline{d}(n)$ may be considered as the result of filtering $\underline{x}(n)$ through an $[M \times K]$ matrix of transfer functions $\mathbf{H}(\omega)$ (not shown in Fig. 1). This filtering operation may be expressed in the frequency domain as

$$\underline{\mathbf{D}}(\omega) = \mathbf{H}(\omega) \underline{\mathbf{X}}(\omega)$$

where $\underline{\mathbf{D}}(\omega)$ and $\underline{\mathbf{X}}(\omega)$ are the DFTs of the time history of $\underline{d}(n)$ and $\underline{x}(n)$, respectively. The matrix $\mathbf{H}(\omega)$ contains the DFT of the impulse responses $\{\underline{h}_{mk} : m = 1, 2, \dots, M, k = 1, 2, \dots, K\}$ between the microphones and the audio signals evaluated at a frequency ω and is defined as

$$\mathbf{H}(\omega) = \begin{bmatrix} H_{11}(\omega) & \cdots & H_{1K}(\omega) \\ H_{21}(\omega) & \cdots & H_{2K}(\omega) \\ \vdots & \ddots & \vdots \\ H_{M1}(\omega) & \cdots & H_{MK}(\omega) \end{bmatrix}$$

The key point in the above mentioned model is to design the matrix $\mathbf{W}(\omega)$ of $[L \times K]$ filters to achieve the control task. The design of those filters is discussed later. Examples illustrating the use of the generalised model in different audio reproduction applications are discussed in the next section.

4 Example Applications

In the generalised model introduced above, different audio reproduction applications may be described by defining different desired sets of sound fields $\underline{d}(n)$ at the microphones. Consequently, it is the matrix $\mathbf{H}(\omega)$ that defines the nature of the application. Examples of the applications that may be described by the model are *cross-talk cancellation*, *virtual source synthesis*, *concert hall simulation*, *correction of the responses of the reproduction loudspeakers*, and *active noise control*.

In *cross-talk cancellation* (see Section 9), it is desired to exactly reproduce the input signals $\underline{x}(n)$ at the microphones. In this case, $\underline{d}(n) = \underline{x}(n)$, and $\mathbf{H}(\omega) = \mathbf{I}$, the $[K \times K]$ identity matrix. Applying this for all frequencies, the transfer function between the m^{th} microphone and the k^{th} input signal has all its elements equal to unity, $\{\underline{H}_{mk}(\omega) = \underline{1} : m = k = 1, 2, \dots, K\}$, which corresponds to a unit impulse response in the time domain $\underline{h}_{mk}(n) = \delta(n)$ ¹. Alternatively, K virtual sound images may be generated at the ears of $M/2$ (M even) listeners if $\underline{d}(n)$ is the result of filtering the input signals $\underline{x}(n)$ through the appropriate matrix of the so-called Head Related Trans-

¹Note that in practise $\underline{h}_{mk}(n) = \delta(n)$ results in a non-causal set of filters $\mathbf{W}(\omega)$. Causal filters produce a delayed version of the inputs $\hat{\underline{d}}(n) = [x_1(n - \Delta_1) \ x_2(n - \Delta_2) \ \cdots \ x_K(n - \Delta_K)]$ at the microphones. In this case, the matrix \mathbf{H} contains delayed unit impulse responses $\{\underline{h}_{mk}(n) = \delta(n - \Delta_k) : m = k = 1, 2, \dots, K\}$.

fer Functions (HRTFs).² The generalised model may also be used to *simulate a concert hall* by setting $\mathbf{H}(\omega)$ to be a matrix of transfer functions measured in a concert hall. Another application that may be described by the model is *correcting the responses of the reproduction loudspeakers* to obtain better sound quality. This may be achieved by letting $\mathbf{W}(\omega)$ be an inverse model of the loudspeaker's transfer functions. The most popular application of the generalised model is in *active noise cancellation* [3]. In such an application, a set of K (undesired rather than desired) sound disturbances $\underline{\mathbf{d}}(n)$ are to be silenced at the microphones. This is achieved by filtering the input signals through $\mathbf{W}(\omega)$ to generate sound waves, $\hat{\underline{\mathbf{d}}}(n)$, that are equal in amplitude but opposite in phase to the disturbances at the microphone's positions, such that the net microphone's outputs $\underline{\mathbf{e}}(n)$ are minimised, where $\underline{\mathbf{e}}(n)$ is given by

$$\underline{\mathbf{e}}(n) = \begin{bmatrix} e_1(n) & e_2(n) & \cdots & e_M(n) \end{bmatrix}^T. \quad (2)$$

The only difference between the above mentioned applications when using the generalised model is the desired response $\underline{\mathbf{d}}(n)$ at the microphones. Therefore, a generalised solution for $\mathbf{W}(\omega)$ that is valid for the whole class of applications may be obtained by solving the system equations for an arbitrary desired response. The solution for a specific application is then obtained by substituting its specific desired response into the general solution.

5 The Optimum Least Mean Square Solution

The optimum Least Mean Square (LMS) solution for the system shown in Fig. 1 is obtained by minimising a performance index function $\xi(n)$, usually taken as the sum of the mean squared error signals

$$\xi(n) = \sum_{m=1}^M E\{e_m^2(n)\}$$

²HRTFs are referred to as the pair of acoustic transfer functions from the source to each of the listener's eardrums embedding all physical parameters associated with the localisation cues that occur in natural listening situations.

$$= E\{\underline{\mathbf{e}}^T(n) \underline{\mathbf{e}}(n)\} \quad (3)$$

where $E\{\cdot\}$ denotes the mathematical expectation. In the following discussion, all filters $\{\underline{\mathbf{w}}_{lk} : l = 1, 2, \dots, L, k = 1, 2, \dots, K\}$ are assumed to be Finite Impulse Response (FIR) digital filters, each of length N_w and defined as

$$\underline{\mathbf{w}}_{lk} = \begin{bmatrix} w_{lk,0} & \cdots & w_{lk,N_w-1} \end{bmatrix}^T$$

The signal driving the l^{th} reproduction loudspeaker $y_l(n)$ is the sum of the outputs of the filters $\{\underline{\mathbf{w}}_{lk} : k = 1, 2, \dots, K\}$, which may be expressed as

$$y_l(n) = \underline{\mathbf{x}}_1^T(n) \underline{\mathbf{w}}_{l1} + \cdots + \underline{\mathbf{x}}_K^T(n) \underline{\mathbf{w}}_{lK}, \quad (4)$$

where the time history of the k^{th} input signal $\underline{\mathbf{x}}_k(n)$ is defined as

$$\underline{\mathbf{x}}_k(n) = \begin{bmatrix} x_k(n) & \cdots & x_k(n - N_w + 1) \end{bmatrix}^T$$

Defining the $[KN_w \times 1]$ composite input signal vector $\underline{\mathbf{x}}(n)$ and the $[KN_w \times 1]$ composite weight vector $\underline{\mathbf{w}}_l$ as

$$\begin{aligned} \underline{\mathbf{x}}(n) &= \begin{bmatrix} \underline{\mathbf{x}}_1(n) & \cdots & \underline{\mathbf{x}}_K(n) \end{bmatrix}^T \\ \underline{\mathbf{w}}_l &= \begin{bmatrix} \underline{\mathbf{w}}_{l1} & \cdots & \underline{\mathbf{w}}_{lK} \end{bmatrix}^T, \end{aligned} \quad (5)$$

equation (4) can be written as

$$y_l(n) = \underline{\mathbf{x}}^T(n) \underline{\mathbf{w}}_l \quad (6)$$

From (1) and (6), the $[L \times 1]$ vector of control signals $\underline{\mathbf{y}}(n)$ driving the reproduction loudspeakers is expressed as

$$\begin{bmatrix} y_1(n) \\ \vdots \\ y_L(n) \end{bmatrix} = \begin{bmatrix} \underline{\mathbf{x}}(n) & \cdots & \mathbf{0} \\ \vdots & \ddots & \vdots \\ \mathbf{0} & \cdots & \underline{\mathbf{x}}(n) \end{bmatrix}^T \begin{bmatrix} \underline{\mathbf{w}}_1 \\ \vdots \\ \underline{\mathbf{w}}_L \end{bmatrix}$$

Further, defining the $[KLN_w \times 1]$ composite vector $\underline{\mathbf{w}} = [\underline{\mathbf{w}}_1 \cdots \underline{\mathbf{w}}_L]^T$, and the $[KLN_w \times L]$ composite matrix $\mathbf{x}(n)$ of input signals given by the first factor in the right hand side of this equation, it can be written compactly as

$$\underline{\mathbf{y}}(n) = \mathbf{x}^T(n) \underline{\mathbf{w}}$$

The system response vector $\hat{\underline{\mathbf{d}}}(n)$ results from filtering the input to the loudspeakers $\underline{\mathbf{y}}(n)$ through the matrix of electro-acoustic transfer functions $\mathbf{C}(\omega)$. The resulting component at the m^{th} microphone is, therefore, given by

$$\hat{d}_m(n) = \underline{\mathbf{c}}_{m1} * \underline{\mathbf{y}}_1(n) + \cdots + \underline{\mathbf{c}}_{mL} * \underline{\mathbf{y}}_L(n),$$

where all electro-acoustic impulse responses $\{\underline{\mathbf{c}}_{ml} : m = 1, 2, \dots, M, l = 1, 2, \dots, L\}$ are assumed to be FIR filters of length N_c , defined as

$$\underline{\mathbf{c}}_{ml} = [c_{ml,0} \quad \cdots \quad c_{ml,N_c-1}]^T$$

Defining the time history of the signal driving the l^{th} loudspeaker $y_l(n)$ as

$$\underline{\mathbf{y}}_l(n) = [y_l(n) \quad \cdots \quad y_l(n - N_c + 1)]^T$$

and the $[M \times LN_c]$ composite matrix of acoustic impulse responses \mathbf{c} as

$$\mathbf{c} = \begin{bmatrix} \underline{\mathbf{c}}_{11}^T & \underline{\mathbf{c}}_{12}^T & \cdots & \underline{\mathbf{c}}_{1L}^T \\ \underline{\mathbf{c}}_{21}^T & \underline{\mathbf{c}}_{22}^T & \cdots & \underline{\mathbf{c}}_{2L}^T \\ \vdots & \vdots & \ddots & \vdots \\ \underline{\mathbf{c}}_{M1}^T & \underline{\mathbf{c}}_{M2}^T & \cdots & \underline{\mathbf{c}}_{ML}^T \end{bmatrix},$$

the $[M \times 1]$ vector $\hat{\underline{\mathbf{d}}}(n)$ of sound waves at the microphones can be expressed as

$$\hat{\underline{\mathbf{d}}}(n) = \mathbf{c} * \underline{\mathbf{y}}(n) = \mathbf{c} * [\mathbf{x}_f^T(n) \underline{\mathbf{w}}] = \mathbf{x}_f(n) \underline{\mathbf{w}},$$

where the $[M \times KLN_w]$ matrix $\mathbf{x}_f(n)$ is given by

$$\mathbf{x}_f(n) = \begin{bmatrix} \underline{\mathbf{x}}_{f_{111}}(n) & \cdots & \underline{\mathbf{x}}_{f_{1LK}}(n) \\ \underline{\mathbf{x}}_{f_{211}}(n) & \cdots & \underline{\mathbf{x}}_{f_{2LK}}(n) \\ \vdots & \ddots & \vdots \\ \underline{\mathbf{x}}_{f_{M11}}(n) & \cdots & \underline{\mathbf{x}}_{f_{MLK}}(n) \end{bmatrix} \quad (7)$$

and the vector $\underline{\mathbf{x}}_{f_{mlk}}(n) = \underline{\mathbf{c}}_{ml} * \underline{\mathbf{x}}_k(n)$ contains the last N_w samples of the result of filtering the k^{th} input signal through the electro-acoustic impulse response between the m^{th} microphone and the l^{th} loudspeaker. The error vector $\underline{\mathbf{e}}(n)$ in Fig. 1 can then be expressed as

$$\underline{\mathbf{e}}(n) = \underline{\mathbf{d}}(n) - \mathbf{x}_f(n) \underline{\mathbf{w}} \quad (8)$$

Substituting in the performance index (3) gives

$$\xi(n) = E\{ \underline{\mathbf{d}}^T(n) \underline{\mathbf{d}}(n) - 2 \underline{\mathbf{w}}^T \mathbf{x}_f^T(n) \underline{\mathbf{d}}(n) + \underline{\mathbf{w}}^T \mathbf{x}_f^T(n) \mathbf{x}_f(n) \underline{\mathbf{w}} \} \quad (9)$$

The optimum LMS solution $\underline{\mathbf{w}}_{opt}(n)$ for the composite weight vector is the vector that minimises the performance index $\xi(n)$ given by (9). The optimum solution is, therefore, obtained by setting the first derivative (gradient) of $\xi(n)$ with respect to the composite weight vector $\underline{\mathbf{w}}$ to zero. The gradient is readily obtained from (9) as

$$\begin{aligned} \underline{\nabla}(n) &= \frac{\partial \xi(n)}{\partial \underline{\mathbf{w}}} \\ &= 2 E\{ \mathbf{x}_f^T(n) \mathbf{x}_f(n) \underline{\mathbf{w}} - \mathbf{x}_f^T(n) \underline{\mathbf{d}}(n) \} \end{aligned} \quad (10)$$

The optimum LMS solution $\underline{\mathbf{w}}_{opt}(n)$ is then obtained by setting $\underline{\nabla}(n)$ to zero resulting in:

$$E\{ (\mathbf{x}_f^T(n) \mathbf{x}_f(n))^{-1} \} E\{ \mathbf{x}_f^T(n) \underline{\mathbf{d}}(n) \}, \quad (11)$$

and the corresponding minimum value ξ_{min} is given by

$$E\{ \underline{\mathbf{d}}^T(n) \underline{\mathbf{d}}(n) \} - E\{ \underline{\mathbf{w}}_{opt}^T(n) \mathbf{x}_f^T(n) \underline{\mathbf{d}}(n) \}.$$

From the first equation it follows that the optimum weight vector $\underline{\mathbf{w}}_{opt}(n)$ exists only if the matrix $(\mathbf{x}_f^T(n) \mathbf{x}_f(n))$ is non-singular. Since the matrix $\mathbf{x}_f(n)$ is the convolution between the input signals and the electro-acoustic transfer functions as given by (7), not only $\underline{\mathbf{x}}(n)$ but also $\mathbf{C}(\omega)$ influence the solution. Depending on the dimensions of $\mathbf{C}(\omega)$, three cases may be recognised:

1. The number of loudspeakers equals the number of microphones ($L = M$):

In this case, $\mathbf{C}(\omega)$ is a square matrix. Consider $K = 1$ for simplicity and using frequency domain representations, the system of equations

$$\underline{\mathbf{E}}(\omega) = \underline{\mathbf{D}}(\omega) - \mathbf{C}(\omega) \underline{\mathbf{Y}}(\omega) \quad (12)$$

is fully determined. Provided $\mathbf{C}(\omega)$ is non-singular, a unique solution for the control vector $\underline{\mathbf{Y}}(\omega)$ exists, which drives the error vector exactly to $\underline{\mathbf{0}}$, namely $\underline{\mathbf{Y}}_{opt}(\omega) = \mathbf{C}^{-1}(\omega) \underline{\mathbf{D}}(\omega)$.

2. The number of loudspeakers is less than the number of microphones ($L < M$):

The matrix $\mathbf{C}(\omega)$ has more rows than columns and the system of equations (12) is over determined. There are more equations to solve than there are unknowns. In this case, provided that

$(\mathbf{x}_f^T(n) \mathbf{x}_f(n))$ is positive definite, a unique global solution exists and is given by (11).

3. The number of loudspeakers is greater than the number of microphones ($L > M$):

There are less equations to solve than there are unknowns. The system of equations (12) is under determined and there exists an infinite number of solutions corresponding to infinite local minima on the error surface. In this case, extra constraints must be imposed to select one of those local solutions. A practical constraint may be limiting the power of the signals $\mathbf{y}(n)$, driving the loudspeakers to avoid nonlinear distortion that may occur due to overloading of the loudspeakers. Equivalently, the same result may be achieved by imposing the constraint on the values of the coefficients of \mathbf{w} . The performance index in this case becomes

$$\xi(n) = E\{\mathbf{e}^T(n) \Gamma_e \mathbf{e}(n) + \mathbf{w}^T \Gamma_w \mathbf{w}\},$$

where Γ_e and Γ_w are (often diagonal) weighting matrices. The optimum weight vector $\mathbf{w}_{opt}(n)$ in this case is given by [6]:

$$E\{(\mathbf{x}_f^T(n) \Gamma_e \mathbf{x}_f(n) + \Gamma_w)^{-1}\} E\{\mathbf{x}_f^T(n) \Gamma_e \mathbf{d}(n)\} \quad (13)$$

and the corresponding minimum value of ξ_{min} is given by

$$E\{\mathbf{d}^T(n) \Gamma_e \mathbf{d}(n)\} - E\{\mathbf{w}_{opt}^T(n) \Gamma_e \mathbf{x}_f^T(n) \mathbf{d}(n)\}$$

The weighting Γ_w has also proven to improve the stability of the solution in the fully determined and over determined cases [6]. Alternatively, the constraint may be imposed on the number of filter taps N_w , which enables an exact solution rather than a least mean square one [7].

6 The Multiple Error Filtered- X LMS Algorithm

An alternative approach to obtaining the optimum LMS solution, other than direct calculation using (11) or (13), is to use an iterative algorithm such as the Multiple Error Least Mean Square (MELMS) algorithm [3, 8]. Such an adaptive approach is preferred over the direct calculation of

$\mathbf{w}_{opt}(n)$, since it offers *in-situ* design of the filters. It also enables a convenient method to readjust the filters whenever a change occurs in the electro-acoustic transfer functions. The MELMS algorithm employs the steepest descent approach to search for the minimum of the performance index (3). This is achieved by successively updating the filters' coefficients by an amount proportional to the negative of the gradient $\underline{\nabla}(n)$,

$$\mathbf{w}(n+1) = \mathbf{w}(n) + \mu(-\underline{\nabla}(n)), \quad (14)$$

where μ is the step size that controls the convergence speed and the final misadjustment [16]. An approximation often used in such iterative LMS algorithms is to update the vector \mathbf{w} using the instantaneous value of the gradient $\underline{\nabla}(n)$ instead of its expected value $\underline{\nabla}(n) = E\{\underline{\nabla}(n)\}$ [16], leading to the well-known LMS algorithm. Using (8) in (10), the gradient can be written as $\underline{\nabla}(n) = 2 E\{-\mathbf{x}_f^T(n) \mathbf{e}(n)\}$. The update equation for the MELMS algorithm is then given by replacing $\underline{\nabla}(n)$ in (14) by its instantaneous value,

$$\mathbf{w}(n+1) = \mathbf{w}(n) + 2 \mu \mathbf{x}_f^T(n) \mathbf{e}(n) \quad (15)$$

This update algorithm is often referred to as the Multiple Error Filtered- X (MEFX) algorithm. Implementation of (15) requires calculating the matrix $\mathbf{x}_f(n)$ given by (7), which implies measuring all the electro-acoustical transfer functions \mathbf{c}_{ml} and filtering each input signal through all ML transfer functions to construct the KLM elements of $\mathbf{x}_f(n)$. This is shown in Fig. 2, where

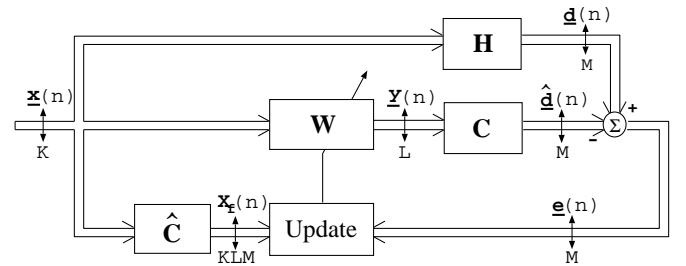


Figure 2: Multiple Error Filtered- X LMS (MEFX) algorithm.

the measured matrix of electro-acoustic transfer functions is represented by the block \hat{C} to distinguish it from the physical one represented by the

block C. The MEFX algorithm is known to be robust to estimation errors. It shows stable convergence properties as long as the phase error in any of the measured transfer functions at any frequency is less than $\pm 90^\circ$ [3, 12, 16], while amplitude errors result in less accurate solutions [10]. Similarly, the noisy steepest descent method may be used to iteratively approach the optimum solution in the case of weighted performance index. In this case

$$\underline{\nabla}(n) = 2 E\{ -\mathbf{x}_f^T(n) \Gamma_e \underline{\mathbf{e}}(n) + \Gamma_w \underline{\mathbf{w}} \}$$

which leads to the following update equation

$$\underline{\mathbf{w}}(n+1) = (\mathbf{I} - \mu \Gamma_w) \underline{\mathbf{w}}(n) + 2\mu \mathbf{x}_f^T(n) \Gamma_e \underline{\mathbf{e}}(n)$$

For $\Gamma_w = \text{diag}\{\gamma_1 \ \gamma_2 \ \cdots \ \gamma_{KLN_w}\}$, each filter weight w_i is independently weighted by the weighting factor γ_i . In this case, when $\Gamma_e = \mathbf{I}$, and letting $\Gamma = \mathbf{I} - \mu \Gamma_w$, the update equation becomes the multiple error leaky LMS algorithm

$$\underline{\mathbf{w}}(n+1) = \Gamma \underline{\mathbf{w}}(n) + 2\mu \mathbf{x}_f^T(n) \underline{\mathbf{e}}(n).$$

7 Adaptive Active Noise Cancellation

The detailed implementation of the MEFX algorithm is further explained by unpacking the composite vector $\underline{\mathbf{w}}(n)$ in (15) into its individual filters $\{\underline{\mathbf{w}}_{lk}(n) : l = 1, 2, \dots, L, k = 1, 2, \dots, K\}$ giving

$$\underline{\mathbf{w}}_{lk}(n+1) = \underline{\mathbf{w}}_{lk}(n) + 2\mu \sum_{m=1}^M e_m(n) \underline{\mathbf{x}}_{f_{lmk}}(n) \quad (16)$$

where $\underline{\mathbf{x}}_{f_{lmk}}(n)$ is the result of filtering the k^{th} input through the measured electro-acoustic impulse response $\hat{\mathbf{C}}_{ml}$ between the m^{th} microphone and the l^{th} loudspeaker. The detailed implementation of (16) is visually illustrated in Fig. 3 for a $[K \times L \times M = 2 \times 2 \times 2]$ active noise control system. In this system, it is desired to reduce the sound field due to the primary sources P_1 and P_2 at two receiver points (microphones) R_1 and R_2 using an anti-sound field generated

by the secondary loudspeakers S_1 and S_2 . Using frequency domain representations, the inputs to the secondary sources $Y_1(\omega)$ and $Y_2(\omega)$ are controlled by four adaptive filters $\{\underline{\mathbf{W}}_{lk}(\omega) : l = 1, 2, k = 1, 2\}$ to achieve the above cancellation task. The desired response is the sound field generated by the primary sources P_1 and P_2 at the microphones' positions: $\underline{\mathbf{D}}(\omega) = [D_1(\omega) \ D_2(\omega)] = \mathbf{H}(\omega) \underline{\mathbf{X}}(\omega)$, where $\mathbf{H}(\omega)$ is the matrix of electro-acoustic transfer functions between the two microphones and the primary sources and $\underline{\mathbf{X}}(\omega) = [X_1(\omega) \ X_2(\omega)]$. The microphones' outputs are, therefore, the *sum* of the sound fields due to the primary and the secondary sources $\underline{\mathbf{E}}(\omega) = [E_1(\omega) \ E_2(\omega)] = \underline{\mathbf{D}}(\omega) + \hat{\underline{\mathbf{D}}}(\omega)$, where $\hat{\underline{\mathbf{D}}}(\omega) = [\hat{D}_1(\omega) \ \hat{D}_2(\omega)] = \mathbf{C}(\omega) \underline{\mathbf{Y}}(\omega)$ is the (unmeasurable) sound field due to the secondary sources alone at the microphones.

At each time sample, the update algorithm adjusts the coefficients of the adaptive filters to minimise the error signals measured by the microphones using the MEFX algorithm (16)³. Four update blocks are needed, one for each adaptive filter. These are indicated by the MEFX boxes in Fig. 3. According to (16), each of the update blocks requires two filtered input signals and in total $KLM = 8$ filtered input signals are needed. The filtered input signals are calculated by filtering each of the two input signals through the four measured impulse responses $\{\hat{\mathbf{C}}_{ml}(\omega) : m = 1, 2, l = 1, 2\}$ that are estimates of the physical electro-acoustic transfer functions between the secondary sources and the microphones represented in Fig. 3 by the matrix \mathbf{C} .

After successful convergence, the sound waves generated by S_1 and S_2 at R_1 and R_2 equal in magnitude and opposite in phase to those generated by P_1 and P_2 , and reduction in the net sound field at R_1 and R_2 results. This may be expressed mathematically as $\mathbf{H}(\omega) \underline{\mathbf{X}}(\omega) = -\mathbf{C}(\omega) \mathbf{W}(\omega) \underline{\mathbf{X}}(\omega)$, and the solution to the matrix of control filters is given by

$$\mathbf{W}(\omega) = -\mathbf{C}^{-1}(\omega) \mathbf{H}(\omega)$$

³Since (16) was derived assuming that the error is formed by the difference between rather than the sum of $\underline{\mathbf{D}}(\omega)$ and $\hat{\underline{\mathbf{D}}}(\omega)$, the + sign in (16) must be changed to - sign.

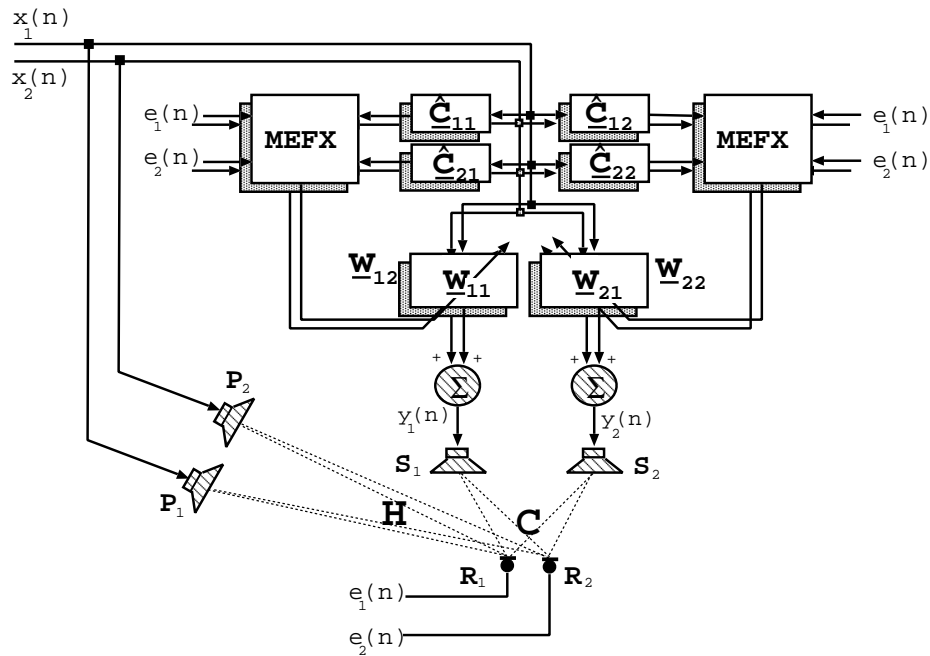


Figure 3: MEFX algorithm for a $[K \times L \times M = 2 \times 2 \times 2]$ system in an active noise cancellation setting.

8 Adaptive Virtual Sound Sources

The system in Fig. 3 is exactly (except for a – sign) what is needed to generate two virtual sound images at the positions of the primary sources P_1 and P_2 at the ears of one listener when the two microphones are positioned inside the listener’s ear canals. By controlling the sound at $M = 2B$ microphones, the same sound images are perceived by B listeners, and by increasing the number of inputs, more images are created. This suggests a procedure for implementing the loudspeaker display system. With probe microphones inserted in the listener’s ear canals, the set of adaptive filters are adjusted to cancel uncorrelated white noise signals from physical loudspeakers placed where virtual sound images are required⁴. After successful conversion, the physical primary sources are disconnected and the coefficients of the filters

⁴The adaptation process requires first measuring all electro-acoustic transfer functions \hat{C}_{ml} . On-line estimation of these transfer functions is discussed in detail in [4]

are multiplied by -1 to compensate for the sign difference mentioned above. Monophonic signals filtered through the previously obtained filters will move the auditory event to the positions where the primary sources have been. This procedure has been successfully used in [1, 14] and proved to have the following advantages compared to the other methods:

- The listener’s own HRTFs are used to design the filters and therefore correct spectral cues are maintained.
- The solution also includes the room impulse response and therefore it maintains the distance cues and the environmental context. This solves the In-Head-Localisation (IHL) problem and eliminates the need for artificial reverberation.
- Direct inversion of the electro-acoustic transfer function matrix $C(\omega)$ that is required for the cross-talk cancellation is avoided by iteratively searching the optimum solution in the least mean square sense.
- The binaural synthesis (convolution with the HRTF matrix $H(\omega)$) and the cross-talk cancellation (inversion of $C(\omega)$) are combined. This is both more efficient and numerically more stable.
- The problem is mapped from the difficult do-

main of virtual sound image synthesis to the well-developed one of multichannel active noise control (ANC). This not only allows employing the techniques used in ANC systems, but also facilitates describing the system performance in terms of the sound attenuation achieved at the listener's eardrums. The larger this attenuation, the more realistic the virtual source is perceived.

- Active noise control systems are real-time systems. Therefore, the above procedure allows real-time design and implementation of the system's filters. This differs from the commonly used methods of measuring and storing a set of HRTFs, calculating the cross-talk cancellation filters, and real-time filtering of the audio signals through those previously calculated fixed filters.

In spite of the above mentioned advantages, adaptive filters approach to virtual source synthesis suffers the following drawbacks:

- Including the room impulse response in the filters limits the virtual space to the same measurement environment.
- The listener is asked to insert a pair of microphones inside his/her ears, which may be objectionable. However, this is the only approach to obtaining individualised HRTFs.
- White noise or chirp signals that are spectrally rich must be used in the identification and adaptation stages.
- The electro-acoustic transfer functions $\mathbf{C}(\omega)$ and $\mathbf{H}(\omega)$ are very complex functions of frequency and space coordinates. The impulse response of a room may also last for several hundreds of milliseconds. At a sampling frequency of 44.1 kHz, thousands of FIR coefficients are required to properly model these transfer functions. The adaptive filters are, therefore, of high order, which makes system implementation in real-time a great challenge. Efficient implementation of the filtering and adaptation operations are, therefore, essential (see Section 10).
- At high frequencies, the acoustic wavelength is very small. Therefore, the solution obtained by the adaptive process is valid only in a very small area in space. Therefore, the listener's head must be fixed during the adaptation and filtering operations.

9 Adaptive Cross-Talk Cancellation

The cross-talk cancellation may also be realised using adaptive filters. The system shown in Fig. 3 can be used for this purpose after a few modifications. In cross-talk cancellation, it is desired to reproduce $x_1(n)$ at R_1 and $x_2(n)$ at R_2 . The desired response is, therefore, given by $\underline{\mathbf{d}}(n) = \underline{\mathbf{x}}(n)$, and the adaptive filters are required to model the inverse of the matrix $\mathbf{C}(\omega)$. For the inverse to be realisable using FIR filters, delayed input signals are used to calculate the desired responses,

$$\begin{bmatrix} d_1(n) \\ d_2(n) \end{bmatrix} = \begin{bmatrix} x_1(n - \Delta_1) \\ x_2(n - \Delta_2) \end{bmatrix},$$

where Δ_1 and Δ_2 are delays that are assumed to be longer than that introduced by the electro-acoustic transfer functions comprising $\mathbf{C}(\omega)$. Since the microphones' outputs in this application are due to S_1 and S_2 only, they correspond to the vector $\hat{\underline{\mathbf{d}}}(n)$ in Fig. 1. The error signals, to be minimised by the adaptive algorithm, are then constructed by *electrically* subtracting $\hat{\underline{\mathbf{d}}}(n)$ from $\underline{\mathbf{d}}(n)$

$$\begin{bmatrix} e_1(n) \\ e_2(n) \end{bmatrix} = \begin{bmatrix} x_1(n - \Delta_1) - \hat{d}_1(n) \\ x_2(n - \Delta_2) - \hat{d}_2(n) \end{bmatrix}.$$

Provided the FIR filters \mathbf{W} are of sufficiently high order to accommodate the inverse solution, the mean square error is minimised by the adaptive algorithm. After conversion, the sound pressure at the microphones R_1 and R_2 are the best least squares estimations of $x_1(n - \Delta_1)$ and $x_2(n - \Delta_2)$, respectively.

10 Efficient Implementations

In reverberant acoustic environments, the impulse response between two points may last for several hundreds of milliseconds. At the standard audio compact disc sampling frequency of 44.1 kHz, thousands of FIR coefficients are needed to properly model and store such an impulse response. Adaptive multichannel audio reproduction systems require estimating, filtering through, and up-

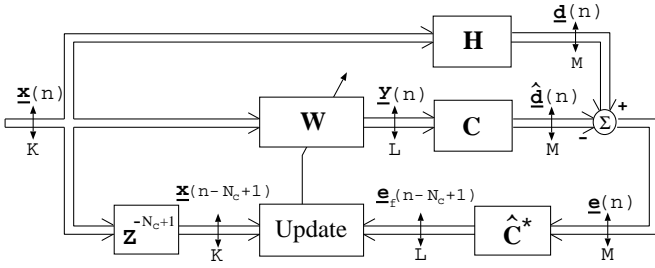


Figure 4: Adjoint Least Mean Square Algorithm.

dating the coefficients of many of these large filters. This implies a huge amount of computational power which makes real-time implementation on reasonable hardware resources a difficult task. Reducing the system complexity is, therefore, essential for real-time implementation. In this section, two approaches are introduced to decrease the number of required calculations. The adjoint LMS algorithm discussed in Section 10.1 uses filtered error signals rather than filtered input signals in the update process. This leads to a huge computational saving as the system dimensions increase. Significant saving may further be achieved by implementing the convolution and correlation operations in the frequency domain using block processing techniques. This leads to the Block Frequency Domain Adaptive Filters (BFDAF) discussed in Section 10.2.

10.1 The Adjoint LMS Algorithm

Setting the electro-acoustic transfer function matrix $\mathbf{C}(\omega)$ in Fig. 2 at each frequency to be the identity matrix \mathbf{I} results in a regular adaptive filter system [16]. In this case, $\hat{\mathbf{d}}(n) = \mathbf{y}(n)$ and the adaptive filters' outputs are directly observable from the error vector $\mathbf{e}(n) = \mathbf{d}(n) - \mathbf{y}(n)$. The steepest descent update given by (15) becomes

$$\mathbf{w}(n+1) = \mathbf{w}(n) + 2\mu \mathbf{x}^T(n) \mathbf{e}(n), \quad (17)$$

where $\mathbf{x}(n)$ is defined in (5). The adaptive process in (17) basically aims at minimising the cross correlation between the input signals $\mathbf{x}(n)$ and the error signals $\mathbf{e}(n)$. When this cross correlation reaches its minimum value, the filters are as close as possible to their optimal solutions. For correct

adaptation, it is essential that the cross correlation is performed between correctly aligned history samples of $\mathbf{e}(n)$ and $\mathbf{x}(n)$.

Introducing the matrix $\mathbf{C}(\omega)$ into the system results in $\hat{\mathbf{d}}(n)$ being a filtered version of $\mathbf{y}(n)$. This has two severe consequences on the adaptation process. The first is that the cross correlation is performed between two misaligned time sequences, since $\mathbf{y}(n)$ is delayed while propagating through the matrix of acoustical transfer function $\mathbf{C}(\omega)$, which leads to an unstable adaptive algorithm. The second is the spectral deformation caused by $\mathbf{C}(\omega)$. The effect of the latter may be explained by the extreme case of each of $\{\mathbf{C}_{ml}(\omega_0) : l = 1, 2, \dots, L\}$ has a zero response at the same frequency ω_0 . In this case, $e_m(n)$ contains no information about the filters' outputs at ω_0 , resulting in an unobservable system at that frequency.

A stable adaptive algorithm with $\mathbf{C}(\omega) \neq \mathbf{I}$ may only be obtained if the above mentioned filtering effects are compensated. The MEFX algorithm discussed in Section 6 solves this problem by filtering $\mathbf{x}(n)$ through estimates of $\mathbf{C}(\omega)$ prior to using them as inputs to the adaptation process as shown in Fig. 2. This effectively delays the inputs to be correctly aligned in time with the error signals, and at the same time introduces the required spectral correction. Since the dimensions of matrices may not match and matrices do not commute, the MEFX effectively filters every input signal $\mathbf{x}_k(n)$ with every electro-acoustic response \mathbf{c}_{ml} to construct the KLM elements of the matrix $\mathbf{x}_f(n)$. This requires in total KLM convolution operations only for constructing $\mathbf{x}_f(n)$. Assuming time domain implementation, the total number of multiplications required per iteration for calculating $\mathbf{x}_f(n)$ is $KLMN_c$. Updating the weights requires $KLN_w(M+1)$ multiplications, and calculating the filters' outputs requires KLN_w multiplications. The total number of multiplications required to implement the MEFX algorithm at each iteration is, therefore,

$$\Psi_{MEFX} = KL[(N_w + N_c)M + 2N_w] \quad (18)$$

which increases rapidly with increasing system dimensions. A much more efficient algorithm may be obtained by advancing the error signals

$\underline{\mathbf{e}}(n)$ in time to be aligned with $\mathbf{x}(n)$. This leads to the Adjoint Least Mean Square (ALMS) algorithm [15], which has its update equation given by (with $n' = n - N_c + 1$):

$$\underline{\mathbf{w}}(n+1) = \underline{\mathbf{w}}(n) + 2\mu \mathbf{x}^T(n') \underline{\mathbf{e}}_f(n') \quad (19)$$

where $\underline{\mathbf{e}}_f = [\underline{\mathbf{e}}_{f_1} \quad \underline{\mathbf{e}}_{f_2} \quad \cdots \quad \underline{\mathbf{e}}_{f_L}]^T$ is the result of filtering the error signals through the *adjoint* (time mirror) of the transfer functions comprising $\mathbf{C}(\omega)$. This non-causal operation can only be performed in real-time after a delay of $N_c - 1$ samples. The reason for using delayed input and filtered error signals is given in (19). The time mirror operation is equivalent to calculating the complex conjugate of the frequency response as shown in Fig. 4. Expressing the adjoint of $\underline{\mathbf{c}}_{ml}$ as

$$\check{\underline{\mathbf{c}}}_{ml} = [\underline{\mathbf{c}}_{ml, N_c-1} \quad \cdots \quad \underline{\mathbf{c}}_{ml, 0}]^T$$

the filtered error signal vector $\underline{\mathbf{e}}_{f_i}$ may be written as (with $n' = n - N_c + 1$)

$$\underline{\mathbf{e}}_{f_i}(n') = \check{\underline{\mathbf{c}}}_{1l} \underline{\mathbf{e}}_1^T(n) + \cdots + \check{\underline{\mathbf{c}}}_{Ml} \underline{\mathbf{e}}_M^T(n)$$

The ALMS algorithm reduces the number of multiplications required for updating the adaptive weights to $2KLN_w$. Adding LMN_c multiplications for calculating $\underline{\mathbf{e}}_f(n)$, and KLN_w multiplications for calculating the filters' outputs sums to

$$\Psi_{ALMS} = KL \left[\frac{N_c}{K} M + 3N_w \right] \quad (20)$$

Comparing (20) with (18) shows that for a $[K \times L \times M = 1 \times 1 \times 1]$ system, the ALMS and MEFX have the same complexity of $(3N_w + N_c)$ multiplications. For multichannel systems, however, the ALMS algorithm is much more efficient than the MEFX.

A detailed implementation of the ALMS algorithm may be seen from the update equation of an individual filter $\underline{\mathbf{w}}_{lk}(n)$, which is given by (with $n' = n - N_c + 1$)

$$\underline{\mathbf{w}}_{lk}(n+1) = \underline{\mathbf{w}}_{lk}(n) + 2\mu e_{f_i}(n') \underline{\mathbf{x}}_k(n')$$

This update is shown in Fig. 5 for a $[K \times L \times M = 2 \times 2 \times 2]$ system in an active noise control setting similar to that in Fig. 3. A comparison of Fig. 5

and Fig. 3 reveals the structural simplicity offered by the ALMS over the MEFX, in addition to the computational saving mentioned above. Two sources of computational saving may be recognised in this example. The first is the reduction of the number of convolutions required to calculate the filtered signals (4 in case of ALMS against 8 in MEFX). The second is the simplified ALMS update that always uses a single error signal and a single input signal compared to 2 error and 2 input signals in the MEFX case.

10.2 Frequency Domain Implementations

Implementation of the MEFX (Fig. 2) or ALMS (Fig. 4) requires performing three main tasks:

1. Calculating the adaptive filters' outputs $\underline{\mathbf{y}}(n)$.
2. Calculating the filtered inputs $\mathbf{x}_f(n)$ (or errors $\underline{\mathbf{e}}_f(n)$).
3. Updating the coefficients of the adaptive filters $\underline{\mathbf{W}}$.

The first two tasks are convolution operations between $\mathbf{x}(n)$ and FIR filters comprising $\underline{\mathbf{W}}(\omega)$ and $\mathbf{C}(\omega)$, respectively. The third task implies calculating the instantaneous gradient that is essentially a crosscorrelation between the input and error signals. Since all FIR filters are of high order, implementing the convolution and correlation in the frequency domain results in considerable computational saving [9]. Since the input signals are infinitely long, real-time implementation of the convolution or correlation in the frequency domain must be performed on successive overlapping blocks of data. Two known and frequently used methods to construct and process such overlapping blocks of data are the overlap-add and overlap-save methods [9]. Applying any of those block processing techniques to the LMS adaptive process leads to the Block Frequency Domain Adaptive Filter (BFDAF) [2, 13].

Besides the speed gained by carrying out the convolution and correlation operations in the frequency domain, BFDAF offers the possibility of approximate decorrelation of the adaptive process from the input signals' statistics. This is done by normalising each frequency bin by the input power in that bin [13]. Unlike the systems dis-

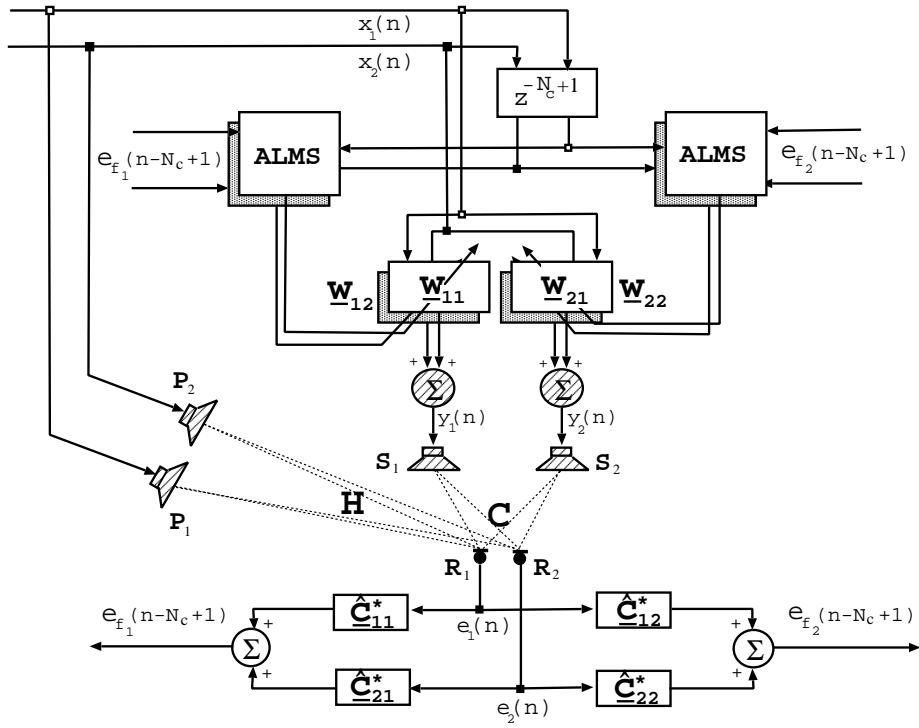


Figure 5: ALMS algorithm for a $[K \times L \times M = 2 \times 2 \times 2]$ system in an active noise cancelling setting (refer with Fig. 3)

cussed previously which update the coefficients of the adaptive filters every sample, BFDAF performs such update every $L \geq 1$ samples, which leads to lower complexity. Due to block processing, reduction in the step size upper bound and processing delay of L samples are also introduced. In this section, implementations of 3D sound systems using BFDAF are considered. Since the adjoint LMS algorithm is more efficient for multichannel systems as shown in Section 10.1, only BFDAF implementation of that algorithm is considered below. For clarity, the BFDAF implementation of the ALMS algorithm is illustrated for a single channel active noise control system, since the extension to multichannel systems is straight-forward as given by (19). The block diagram of the ALMS algorithm implemented using overlap-save BFDAF for $[K \times L \times M = 1 \times 1 \times 1]$ ANC system is shown in Fig. 6. In this block diagram, vectors are represented by double parallel lines while scalars are represented by single lines. Since we are dealing with a single channel case, vectors are used here to represent frequency samples of the signals rather than

multiple signals at the same frequency.

The overlap-save method requires sectioning the stream of input time samples $x(n)$ into overlapping blocks, each of length N_B . Each block contains L new samples and $N_w + N_c - 2$ samples from the previous block. The sectioning and overlapping operations are represented in Fig. 6 by the *serial-to-parallel* and *overlap* blocks. For each L samples, a complete block of time samples $\underline{x}_B(nL)$ is constructed and processing of this block of data may be commenced. For correct real-time operation, the processing of each block must be completed before the next block is constructed, i.e. in a maximum of L samples. To perform convolution and correlation in the frequency domain, each block $\underline{x}_B(nL)$ is transformed to the frequency domain using a length $N_B = N_w + N_c + L - 2$ Fast Fourier Transform (FFT) algorithm, $\underline{X}(nL) = \mathbb{F}_{N_B} \underline{x}_B(nL)$. The length of the block N_B is chosen such that the circular correlation performed by element-wise multiplication in the frequency domain results in N_w correct time domain correlation coefficients as will be shown shortly. The processing of each block can be di-

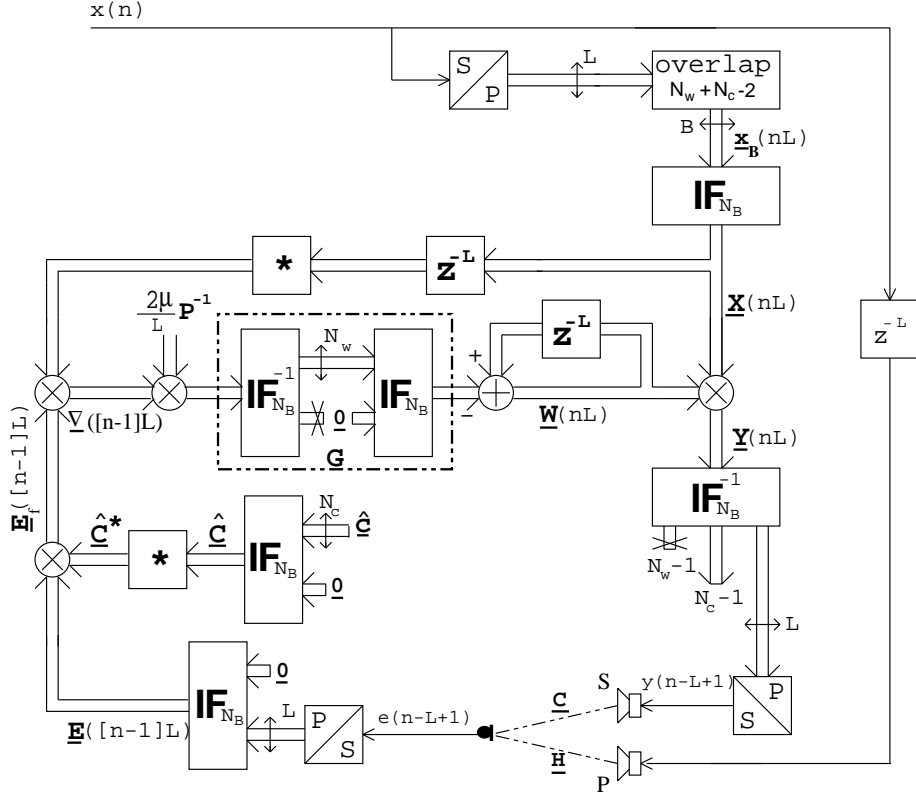


Figure 6: Block Frequency Domain implementation of the ALMS algorithm for a single channel $[K \times L \times M = 1 \times 1 \times 1]$ system.

vided into three main tasks, as mentioned above:

1. Calculation of the filter's output $y(n-L+1)$:

The convolution between the transformed input block $\underline{\mathbf{X}}(nL)$ and the frequency domain adaptive filter weights $\underline{\mathbf{W}}(nL)$ is carried out by element-wise multiplication. Defining the matrix $\mathbf{X}(nL) = \text{diag}\{\underline{\mathbf{X}}(nL)\}$, this convolution may be represented by

$$\underline{\mathbf{Y}}(nL) = \mathbf{X}(nL) \underline{\mathbf{W}}(nL)$$

The vector $\underline{\mathbf{Y}}(nL)$ is then transformed back into the time domain using an Inverse Fast Fourier Transform (IFFT) algorithm of length N_B . The last L samples of this time domain vector represent the required linear convolution. During processing the next block, these L samples are sent, one sample at each clock cycle, to drive the loudspeaker S . This is represented in Fig. 6 by the *parallel-to-serial* block. The first $N_w - 1$ samples represent cyclic convolution results and must be discarded. An extra $N_c - 1$ samples represent

correct linear convolution results but are not used since only L samples are needed to be sent out at every block. This extra $N_c - 1$ samples are due to the choice of a large block length N_B to accommodate successive convolution and correlation as described below.

2. Calculation of the filtered error $\underline{\mathbf{E}}_f([n-1]L)$:

The microphone signal $e(n-L+1)$ is buffered into a length L vector. This vector is augmented with $N_B - L$ leading zeros and transformed into the frequency domain using an FFT of length N_B , resulting in the block error vector $\underline{\mathbf{E}}([n-1]L)$. A previously measured impulse response $\hat{\mathbf{c}}$ of length N_c between loudspeaker S and the microphone is padded with $N_B - N_c$ trailing zeros and transformed to the frequency domain using an FFT of length N_B , producing the vector $\hat{\mathbf{C}}$. The adjoint (time mirror) operation required for the ALMS is performed by calculating the complex conjugate of $\hat{\mathbf{C}}$. The filtered error vector $\underline{\mathbf{E}}_f([n-1]L)$

is then calculated by element-wise multiplication of $\hat{\underline{\mathbf{C}}}^*$ and $\underline{\mathbf{E}}([n-1]L)$. The length of the result of this convolution is $L + N_c - 1$, and since N_B is chosen larger than this length, all samples of $\underline{\mathbf{E}}_f([n-1]L)$ represent correct linear convolution results. Defining the matrix $\hat{\underline{\mathbf{C}}} = \text{diag}\{\hat{\underline{\mathbf{C}}}\}$, this convolution may be expressed mathematically as

$$\underline{\mathbf{E}}_f([n-1]L) = \hat{\underline{\mathbf{C}}}^* \underline{\mathbf{E}}([n-1]L)$$

Since the microphone signal $e(n-L+1)$ is due to the output samples $y(n-L+1)$ from the previous block, the error signal is already delayed by L samples. Provided $L \geq N_c$, this delay will suffice to implement the delay required by the ALMS as mentioned in Section 10.1. To further ensure that the optimum filter solution is causal, the delay in the path through the loudspeaker P must be longer than that through the loudspeaker S . Since the algorithm introduces a delay of L samples, this delay must be compensated. Assuming that the acoustic transfer function $\underline{\mathbf{H}}$ possesses longer delay than $\underline{\mathbf{C}}$, this compensation may be done by delaying the signal to loudspeaker P by L samples. This is represented in Fig. 6 by the box z^{-L} . An elegant alternative is to move loudspeaker P an equivalent distance away from the microphone to save DSP memory.

3. Updating the filter weights:

Transforming the update equation of the ALMS (19) to the frequency domain results in the following block frequency domain update equation ⁵ (with $\underline{\nabla}([n-1]L) = \mathbf{X}^*([n-1]L) \underline{\mathbf{E}}_f([n-1]L)$)

$$\begin{aligned} \underline{\mathbf{W}}(nL) &= \underline{\mathbf{W}}([n-1]L) - \\ &\quad - \frac{2\mu}{L} \underline{\mathbf{G}} \underline{\mathbf{P}}^{-1} \underline{\nabla}([n-1]L), \end{aligned} \quad (21)$$

where it is assumed that $L \geq N_c$, the delay required by the ALMS algorithm. The diagonal matrix $\underline{\mathbf{P}}$ represents an estimate of the input signal power at each frequency bin and performs the decorrelation mentioned above. The inverse of this power matrix is scaled by a step size of $2\mu/L$ and the result is used as the new step

⁵Since signals are summed up at the microphones in Fig. 6 rather than subtracted, the + sign in (19) has been changed to a - sign in (21).

size. This frequency-dependent step size vector is used to scale the estimated gradient vector. The instantaneous block mean square gradient estimate vector given by the cross correlation $\underline{\nabla}([n-1]L) = \mathbf{X}^*([n-1]L) \underline{\mathbf{E}}_f([n-1]L)$ is obtained by first delaying $\underline{\mathbf{X}}(nL)$, calculating its complex conjugate, and element-wise multiplication of the result by the block filtered error as shown in Fig. 6. Since $\underline{\mathbf{X}}([n-1]L)$ is not padded with zeros, this element-wise multiplication corresponds in the time domain to N_w linear correlation coefficients, while the last $N_B - N_w$ samples are the cyclic correlation coefficients and must be discarded. However, the correct correlation coefficients must be extracted in the time domain, which is done using the $N_B \times N_B$ constraining window $\underline{\mathbf{G}}$. This window transforms the estimated gradient to the time domain, replaces the last $N_B - N_w$ samples by zeros, and transforms the result back to the frequency domain. Finally, the weighted and constrained gradient is used to update the weight vector $\underline{\mathbf{W}}([n-1]L)$ according to (21).

Finally, it is worth mentioning that the frequency domain implementation of a single channel system using the adjoint LMS may lead to more computational saving over that using the filtered- x algorithm if N_w is much larger than N_c . This second source of computational saving stems from the fact that the filtered- x algorithm calculates \mathbf{x}_f by filtering the infinitely long input signal through the finite impulse response $\underline{\mathbf{C}}$. Implementing this convolution in the frequency domain requires the use of an overlapping method, and possibly two extra FFTs to separate the correct convolution samples in the time domain. On the other hand, the frequency domain adjoint LMS algorithm calculates $\underline{\mathbf{E}}_f$ by filtering (the already in block form) error signal through $\underline{\mathbf{C}}^*$ and no overlapping is needed. BFDAF implementations of the single channel adjoint LMS and several implementations of the single channel filtered- x have already been presented in the previous works [1, 14]. A complexity comparison between these implementations shows that for a single channel BFDAF, the filtered- x is more efficient than the adjoint LMS when $N_w < 2N_c$,

while the adjoint LMS is much more efficient when $N_w > 2N_c$.

11 Conclusions

One of the research areas of the Signal Processing group at the Eindhoven University of Technology is *adaptive array signal processing*. It has been shown that these adaptive array signal processing techniques can also be applied to audio applications as well. The current paper, based on the results of the PhD work [4], discusses theoretical and implementation issues concerning adaptive multichannel audio reproduction systems.

First a general model was derived that is capable of describing a wide class of multichannel audio reproduction system applications including synthesis of virtual sound sources, cross-talk cancellation and active noise control. The optimum least squares solution was given and from this an adaptive solution was derived resulting in the Multiple Error Filtered-X Least Mean Square (MEFX) algorithm. Efficient implementations have been discussed.

References

- [1] R.M. Aarts, P.C.W. Sommen, A.W.M. Mathijssen, and J. Garas, *Efficient Block Frequency Domain Filtered-x applied to Phantom Sound Source Generation*, AES 104th convention, Amsterdam, May 1998, Preprint No. 4650.
- [2] G.A. Clark, S.R. Parker, and S.K. Mitra, *A Unified Approach to Time- and Frequency-Domain Realization of FIR Adaptive Digital Filters*, IEEE Trans. on Acoust., Speech and Sig. Proc., Vol. ASSP-31, No. 5, Oct. 1983, pp. 1073-1083.
- [3] S.J. Elliott, I.M. Stothers, and P.A. Nelson, *A Multiple Error LMS Algorithm and Its Application to the Active Control of Sound and Vibration*, IEEE Trans. on Acoust., Speech and Sig. Proc., Vol. ASSP-35, No. 10, Oct. 1987, pp. 1423-1434.
- [4] J. Garas, *Adaptive 3D Sound Systems*, PhD dissertation, Eindhoven University of Technology, The Netherlands, September 1999, ISBN 90-386-1640-6.
- [5] D. H. Johnson and D. E. Dudgeon *Array Signal Processing: Concepts and Techniques* Prentice Hall Canada, 1993, ISBN 0-13-048513-6.
- [6] O. Kirkeby, P.A. Nelson, H. Hamada, and F. Orduna-Bustamante, *Fast Convolution of Multichannel Systems Using Regularization*, IEEE Trans. on Speech and Audio Proc., Vol. 6, No. 2, March 1998, pp. 189-194.
- [7] M. Miyoshi and Y. Kaneda, *Inverse Filtering of Room Acoustics*, IEEE Trans. on Acoust., Speech and Sig. Proc., Vol. ASSP-36, No. 2, Feb. 1988, pp. 145-152.
- [8] P.A. Nelson, F. Orduna-Bustamante, and H. Hamada, *Multichannel Signal Processing Techniques in the Reproduction of Sound*, J. Audio Eng. Soc., Vol. 44, No. 11, Nov. 1996, pp. 973-989.
- [9] A.V. Oppenheim and R.W. Schaffer, *Discrete-Time Signal Processing*, Prentice-Hall, Englewood Cliffs, NJ, 1989, ISBN 0-13-216-771-9.
- [10] N. Saito and T. Sone, *Influence of Modeling Error on Noise Reduction Performance of Active Noise Control Systems using Filtered-X LMS Algorithm*, J. Acoust. Soc. Jpn. (E), Vol. 17, No. 4, 1996, pp. 195-202.
- [11] D.W.E. Schobben, *Efficient Adaptive Multichannel Concepts in Acoustics: Blind Signal Separation and Echo Cancellation*, PhD dissertation, Eindhoven University of Technology, The Netherlands, September 1999, ISBN 90-386-1630-9.
- [12] S.D. Snyder and C.H. Hansen, *The Influence of Transducer Transfer Functions and Acoustic Time Delays on the Implementation of the LMS Algorithm in Active Noise*

Control Systems, J. of Sound and Vibration,
Vol. 141, No. 3, 1990, pp. 409-424.

- [13] P.C.W. Sommen, *Adaptive Filtering Methods*, PhD dissertation, Eindhoven University of Technology, The Netherlands, June 1992, ISBN 90-9005143-0.
- [14] P.C.W. Sommen, R.M. Aarts, A.W.M. Mathijssen, J. Garas, and H. He, *Efficient Frequency Domain Filtered-x Realization of Phantom Sources*, Proc. CCSP-97, Mierlo, The Netherlands.
- [15] E.A. Wan, *Adjoint LMS: An Efficient Alternative to the Filtered-X LMS and Multiple Error LMS Algorithm*, Proc. ICASSP-96, pp. 1842-1845.
- [16] B. Widrow and S.D. Stearns, *Adaptive Signal Processing*, Prentice-Hall Inc., Englewood Cliffs, NJ, 1985, ISBN 0-13-004029-0