Delay Estimation for Transform Domain Acoustical Echo Cancellation

Rabih Abouchakra

Department of Electrical Engineering
McGill University
Montreal, Canada

September 1997

A report submitted to the Faculty of Graduate Studies and Research in partial fulfillment of the requirements for the degree of Master of Engineering.

Abstract

Acoustic echo cancellation can be used to remove the annoying talker feedback in hands-free (teleconferencing) systems. The echo canceller identifies the response between the loudspeaker and the microphone, and produces an echo replica which is then subtracted from the signal. Adaptive filtering techniques are employed to determine the echo path response. The speech signal (or the reference signal) is used to train the algorithm. Fast convergence and good tracking capabilities can not be achieved by classical transform domain adaptive filtering algorithms when the reference signal has variable rank autocorrelation matrix. In this thesis, we examine the DCT-LMS algorithm and we emphasize on the role played by the Discrete Cosine Transform. This fixed transformation reduces the eigenvalue spread of the input autocorrelation matrix by partially decorrelating the inputs.

The autocorrelation matrix of speech signals is often rank-deficient. During the low rank phases, some of the transform-domain tap coefficients become irrelevant to the adaptation process and stop adapting. When the autocorrelation matrix gains full rank, there will be no longer any “frozen” weights. However, the weights that have been frozen are “far” from the optimal point; they require additional convergence time to track again the changes in the room impulse response. In this dissertation, we present a new method that uses the information contained in the other coefficients to move the frozen weights closer to the optimal point and, consequently, reduce the overall convergence time.

By modeling the changes in the impulse response that result from an alteration in the spacing between the microphone and the loudspeaker by a single delay, we were able to develop the “Spectrum Delay Update” method. It consists of replacing, during low-rank phase, each frozen coefficient by a delayed version of the previous full-rank solution. To estimate the corresponding delay, a novel DCT-domain delay estimation algorithm was derived.

Simulation results demonstrate the efficiency of SDU for acoustic echo cancellation, the gain in Echo Return Loss is substantial. The experimental performance analysis confirms the expected reduction in the Euclidean Distance between the filter weights and the actual room impulse response DCT. Furthermore, it shows that spectrally updating the filter weights reduces the MSE jump when the autocorrelation matrix gains full rank.

Sommaire

L'annulation des échos acoustiques est utilisée pour éliminer le retour non désirable du signal du locuteur dans les systèmes de communications à main-libre (téléconférence). L'annuleur d'écho identifie la réponse entre le haut-parleur et le microphone, et synthétise une copie de l'écho; cette dernière sera ultérieurement soustraite du signal. Des techniques adaptatives de filtrage sont employées pour déterminer la réponse de la trajectoire d'écho. Le signal de la parole (le signal de référence) est utilisé pour entrainer l'algorithme. Quand ce signal a une matrice d'autocorrelation à rang variable, les algorithmes classiques de filtrage adaptatif opérant dans le domaine transformé sont incapables de réaliser une convergence rapide avec une bonne poursuite de l'évolution de la réponse impulsionnelle. Dans cette thèse, on examine l'algorithme DCT-LMS et on s'attarde sur le rôle joué par la transformée en cosinus discrète (DCT).

La matrice d'autocorrelation de la parole a souvent un rang défectueux. Pendant ces périodes de rang faible, quelques coefficients du filtre deviennent non-pertinents au processus d'adaptation et arrêtent de s'ajuster. Quand la matrice d'autocorrelation atteint le rang complet, aucun coefficient ne restera "figé". Mais en cessant de s'ajuster pour quelques instants les coefficients figés s'éloignent du point optimal; ils requièrent un temps de convergence additionel pour suivre de nouveau les changements dans la réponse impulsionnelle de la pièce. Dans ce mémoire, on présente une nouvelle méthode qui utilise l'information contenue dans les coefficients non-figés pour rapprocher les coefficients figés du point optimal, et par consequent, réduire le temps total de convergence.

En modélisant par un délai unique les changements dans la réponse impulsionnelle qui résultent d'une modification de la distance entre le microphone et le haut-parleur, on a pu développer la méthode dite "Spectrum Delay Update" (ou SDU). Cette méthode consiste à remplacer, pendant la période de rang faible, chaque coefficient figé par une version retardée de la précédente solution à rang complet. Pour estimer le delai correspondant, on a conçu un nouvel algorithme d'estimation du délai qui opère dans le domaine de la DCT.

Les résultats de simulation démontrent l'efficacité de la SDU dans l'annulation des échos acoustiques; l'amélioration en "Perte de Retour d'Echo" est substantielle. L'analyse expérimentale de la performance confirme la réduction attendue de la distance Euclidienne entre les coefficients du filtre et la DCT de la réponse impulsionnelle de la pièce. De plus, cette analyse montre que l'usage de la SDU réduit le saut de l'erreur carrée moyenne quand

la matrice d'autocorrelation change de rang.

Acknowledgments

I wish to express my deepest gratitude to my supervisor, Prof. Peter Kabal, for his guidance throughout my graduate studies at McGill and for suggesting the research subject. His continuous support and his advises were invaluable to the preparation of the thesis.

I would like to thank the Telecommunication and Signal Processing Lab whose computer facilities provided great assistance to my research. The financial support provided by the National Science and Engineering Research Council (NSERC) was infinitely appreciated.

This thesis could not have been completed without the constant support and love of my parents and my brother. I would like finally to thank all my friends for their encouragements and companionship. Special thanks go to Rami Mehio for his enlightening suggestions.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

In teleconferencing, two geographically separate groups of people meet in a “virtual” way. By doing so, they avoid the waste of time and the expenses associated with traveling. The word “teleconferencing” means different things to different people; it can range from a conference call with speaker phones to a full scale audio for video conferencing system with satellite feeds. Regardless of the type of teleconferencing, how the audio portion of the conference is handled will ultimately determine the success of the event.

Among many phenomena which affect the quality of the communication in teleconferencing, the “acoustic echo” is the most important one. Acoustic echo appears when the conference room is operating with open microphones and loudspeakers in full duplex mode (anything either party says is heard by the other, whether or not the other party is speaking). The decoupling of the handset from the head introduces the teleconference room as another component in the audio circuit and creates an acoustic path between the loudspeaker and the microphone as shown in Fig. 1.1. Therefore, the transmitted signal is picked up by the open microphone and retransmitted to the near-end room. The result is that a person will hear his own voice coming back to him from the far room with a delay, thus the undesired echo or the “acoustic echo”. Echo feedback may also produce howling, which is a major annoyance for the users in both rooms.

One should distinguish this unwanted echo from the signal of the far-end talker and the room reflections of that talker’s signal. These reflections convey an overview of the acoustic properties of the far-end room and make the transmitted signal sound “natural”; therefore they should not be suppressed. On the other hand, acoustic echo is not only annoying, but

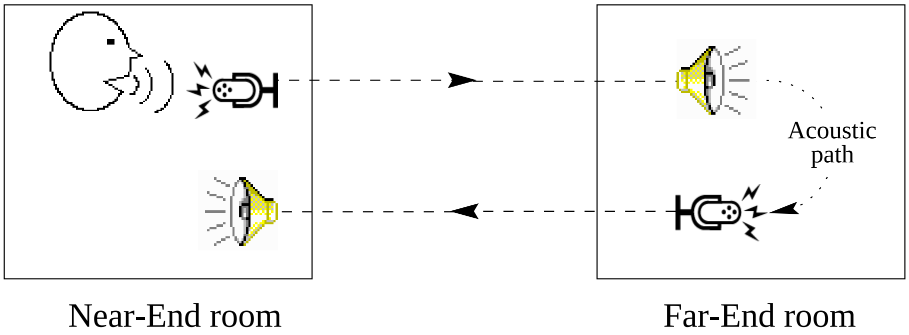


Fig. 1.1 Standard teleconferencing model.

can also be of such a level that makes it difficult to carry on a meaningful conversation. The alleviation of these deleterious effects is an active research area and will be the subject of this thesis.

The standard approach to eliminate the acoustical echo is to use an adaptive filter at the receiver where the echo is predominant. The so-called adaptive filters differ from fixed filters in that their impulse responses are adjusted as data flow through the filter. In this application, the adaptive filter is used to characterize the changing acoustical path between the speaker and the microphone, thus generating a synthesized replica of the acoustical echo. This replica is then subtracted from the microphone signal, leaving the transmitted speech undistorted.

The motions of people or objects during the conference cause fast and non-easily predictable variations of the echo path impulse response [1]. Consequently, acoustic echo cancellers must respond quickly to these variations. Numerous filter structures and *adaptation algorithms* (the algorithms used to adjust the parameters of the adaptive filter) have been proposed in the literature to perform this task. While a transversal Finite Impulse Response (or FIR) filter has been the classical structure, many alternative implementations have been proposed to overcome its limits. In [2], it was shown that a recursive structure (or an Infinite Impulse Response structure) reduces the number of filter taps required. Furthermore, some neural network structures have been suggested to surmount the inability of transversal filters to effectively cancel the nonlinearities which are generated in the loudspeaker during large signal peaks [3]. On the other hand, the transversal FIR structure is still appealing due to its ease of implementation and versatility[4].

The most widely used *adaptation algorithm* is the *least mean squares* or LMS algorithm.

The principle underlying LMS is extremely simple; it consists of defining an error function as the average square difference between the filter output and its target, and iteratively minimizing this error function over the filter coefficient space using a gradient based method. Its main disadvantage is its slow convergence under certain input conditions (input signals which have an autocorrelation matrix with high eigenvalue spread). Many modifications can be brought to the LMS to ameliorate its convergence properties. In [5], it was proposed to process the signal in subbands, i.e., divide the signal into smaller frequency bands and independently cancel echoes in each subband. Since narrower frequency bands have a smaller eigenvalue spread compared to the fullband case, the convergence speed can be increased. The aliasing that might result from the down sampling process can be avoided by the use of some special techniques [6]. A drawback of this structure is the delay introduced by the filter banks.

Another adaptation algorithm that has been suggested for acoustic echo cancellation [7] is the *recursive least squares* or RLS algorithm (see e.g.[8]). In RLS, the filter weights are made equal at each iteration to the best approximation of the Wiener solution that can be calculated based on all the data the system has seen so far. The preprocessing of the inputs by an estimate of the inverse input autocorrelation matrix, in the fashion of RLS, makes the adaptation algorithm computationally intensive and prone to numerical instabilities.

One solution that tries to combine the advantages of both LMS and RLS consists of preprocessing the inputs to the LMS filter with a fixed transformation that does not depend on the actual data and that decorrelate partially the inputs. This will reduce the eigenvalue spread of the autocorrelation matrix (and consequently improve the convergence speed of the filter) without increasing drastically the computational cost. In addition, the robustness and tracking capability of LMS will be preserved. These algorithms are called *transform-domain LMS algorithms*. A block diagram of these algorithms is shown in Fig 1.2.

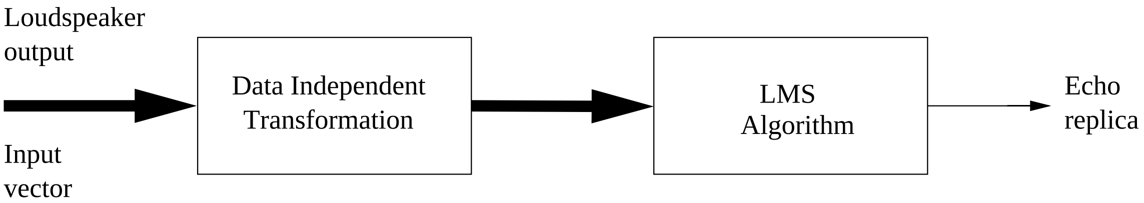


Fig. 1.2 Block diagram of the *transform-domain LMS algorithm.*

The performance of these algorithms clearly depends on the orthogonalizing capabilities of the data-independent transform used to preprocess the inputs. No general proof exists that demonstrates the superiority of one such transform over the others. A classic benchmark is the *Karhunen-Lòeve Transform* (KLT), which performs an exact decorrelation of the input data by projecting them onto the eigenvectors of their autocorrelation matrix [9]. However, KLT is impractical in real time applications because it is a data dependent transformation. Discrete Fourier Transform LMS (or DFT-LMS), which was first introduced by Narayan [10], has a strong appeal because of the familiar frequency domain interpretation of the transformed data. On the other hand, for a wide range of signals, the Discrete Cosine Transform (or DCT) offers a better approximation to KLT. Indeed, for a stationary zero-mean, first-order Markov process, the DCT is asymptotically equivalent to the KLT, with this asymptotic equivalence being demonstrated both as the sequence length increases and also as the correlation coefficient tends to one [11]. In addition, the DCT-LMS has the advantage of having real transformed coefficients. In this thesis, we will focus our analysis on the DCT-LMS algorithm.

Speech signals possess properties one has to account for in the adaptive filtering algorithms. The dynamic character of speech, including the variations in the rank of the autocorrelation matrix, slows down the convergence of the DCT-LMS algorithm. During low rank periods, some of the transform domain tap-coefficients stop adapting and effectively "freeze"; they are irrelevant to the adaptation process. These frozen taps can retain any value without affecting the Mean Square Error (MSE); on the other hand, the remaining taps track the evolution of the system and keep the MSE at a minimum.

It is important to note that in most cases rank deficiency is accompanied with the presence of spectrum gaps. When the autocorrelation matrix becomes nonsingular (or when the spectrum gap vanishes), all the filter weights become relevant and start adapting. However, the weights that have been frozen to values which are probably obsolete are "far" from the optimal point (the actual room impulse response DCT). Consequently, a large jump in the MSE is expected and additional convergence time is required for the frozen coefficients to track again.

The purpose of this thesis is to reduce this convergence time by moving the frozen weights closer to the optimal point, anticipating a change in the rank of the autocorrelation matrix. Updating the weights in that form make them better prepared to begin adapting when the gap vanishes. This process uses the information contained in the "correct" coefficients to produce an estimate of the room impulse response DCT inside the gap. In [12], several Linear Predictive (LP) methods, applied to the DCT spectrum, were suggested. The estimates in the gap were obtained by linearly combining the tracking coefficients—with the parameters calculated by LP analysis.

The key contribution of this work is to model the changes in the echo path impulse response (which we will refer to throughout the thesis as the room impulse response) that result from a change in the spacing between the microphone and loudspeaker by a single delay. This model allows us to synthesize any missing parts of the room impulse response (or its DCT) by simply delaying the original (or before the movement) impulse response. Accordingly, any filter coefficient that has frozen during the low rank phase, can be updated and brought closer to the actual room impulse response DCT by the same mechanism, which is dubbed the "Spectrum Delay Update" method. Since the adaptive filter tracks the DCT of the response, the delay estimation is done in the DCT domain.

In addition to the reduction of the Euclidean Distance between the filter weights and the actual room impulse response DCT, we found that Spectrum Delay Update improves substantially the echo canceller Echo Return Loss and reduces the corresponding MSE.

Thesis Overview

Chapter 2 provides the acoustic echo cancellation background. After a detailed introduction to room acoustics, the loudspeaker-microphone environment is presented along with the echo canceller structure. We follow with an overview of adaptive linear filters, emphasizing on the LMS algorithm and on the Transform-Domain LMS.

After presenting the relevant properties of the DCT in Chapter 3, we develop the DCT-LMS algorithm and justify intuitively its use. We then study the impact of variable rank autocorrelation matrix (of the reference signal) on the convergence speed. Also in Chapter 3, we propose the idea of weight updating to reduce the MSE jump. After describing how to model the changes in the room impulse response by a single delay, we present in detail the Spectrum Delay Update method.

Chapter 4 is devoted to create a delay estimation algorithm in the DCT domain. This algorithm will be based on the the general DCT shift property derived at the beginning of the chapter.

We define in Chapter 5 an experimental setup that will be used to evaluate the performance of Spectrum Delay Update. We also formulate several performance measures.

Their evaluation for different gap sizes and receiver displacements indicate clearly a major enhancement in the performance.

We conclude in Chapter 6 by summarizing the thesis, adding some comments, and listing several points that would be of interest for further study.

The major contributions to knowledge of this work can be summarized as follows:

- Generalization of the DCT one sample shift property to a k-samples shift property.
- Development and implementation of a delay estimation algorithm in the DCT domain.
- Representation of the changes in the room impulse response by a single delay.
- Introduction of Spectrum Delay Update method in which the non tracking DCT-LMS filter coefficients are brought closer to the optimal point. This will reduce the jump in the MSE when the reference signal autocorrelation matrix gains full rank (after a low rank period), and a faster convergence is achieved.

Chapter 2

Acoustic Echo Cancellation

2.1 Room Acoustics

The loss mechanisms which reduce the energy of sound waves when they are reflected from walls as well as during their free propagation in the air are of considerable importance to the acoustics of a room. They influence the strengths of the direct sound and of all reflected components and therefore all acoustical properties of the room.

2.1.1 Room impulse response

We shall regard the sound transmission between two points of a room as formally represented by the impulse response of the transmission path. Let $\mathbf{x} = (a, b, c)$ denote the position vector of a fixed omnidirectional point source and $\mathbf{x}' = (a', b', c')$ the position vector of an omnidirectional point receiver. According to the image model technique [13], the acoustic signal produced at position $\mathbf{x}'$ by an impulse excitation at position $\mathbf{x}$ at time $t = 0$ is given by

$$h(t) = \frac{1}{4\pi} \sum_{r} \frac{\beta_r}{|\mathbf{x}_r - \mathbf{x}'|} \delta(t - \frac{|\mathbf{x}_r - \mathbf{x}'|}{c}) \qquad (2.1)$$

where r is the image index ($r = 0$ usually corresponds to the source itself), $\mathbf{x}_r$ is the position vector of the image indexed by r, β_r is the corresponding composite reflection coefficient, $\delta(.)$ is the Dirac delta function, and c is the speed of sound in air. This impulse response is obtained by adding the responses produced at $\mathbf{x}'$ by multiple image sources located at positions $\mathbf{x}_r$: the term $|\mathbf{x}_r - \mathbf{x}'|/c$ inside the delta function represents the propagation delay from the r^{th} image to the receiver, while the multiplicative factor $1/|\mathbf{x}_r - \mathbf{x}'|$ represents the amplitude attenuation for spherical wavefront propagation (the impulse response will provide the signal amplitude and not the signal power). The factor β_r accounts for the attenuation produced by successive reflections on the planar boundaries.

It should be noted that in this approach the reflection coefficients are assumed to be frequency-independent. To reduce the amount of computation involved, it can be further assumed that the wall reflectivities are independent of the angle of incidence. It is then possible to pre-compute and store the composite reflection coefficients β_r prior to the application of the image model technique.

For a rectangular room, the image distribution forms a three-dimensional rectangular lattice and can be calculated easily. One such method is the well-known *s-room* algorithm[14]. This algorithm uses the fact that the enclosure is rectangular in order to create a mesh of "rooms", each containing an image of the source. When the source is excited, so is each image, thus creating spherical sound-pressure waves which simultaneously propagate away from each image point. A typical *s-room* impulse response (with the corresponding room dimensions and acoustic properties given in Table 2.1) is shown in Fig. 2.1. It should be kept in mind that the *s-room* algorithm is a crude model that does not take into account the presence of objects in the room. However, we can compensate for presence of furniture and other reflective materials in the room by changing the surroundings refection coefficients.

Table 2.1 Room dimensions and acoustic properties

Room Size (meters)	Length=6,Width=4,Height=4
Reflection coefficients	Walls: 0.4 Ceiling: 0.4 Floor: 0.4
Location of the source	$(a = 2, b = 2, c = 2)$
Location of the receiver	$(a' = 1, b' = 1, c' = 1)$

To limit the computations, it is desirable to truncate the impulse response. For environments such as the interior of an automobile, the impulse response can be neglected after 30 ms [15], which corresponds to 240 coefficients at a sampling rate of 8 kHz. For larger

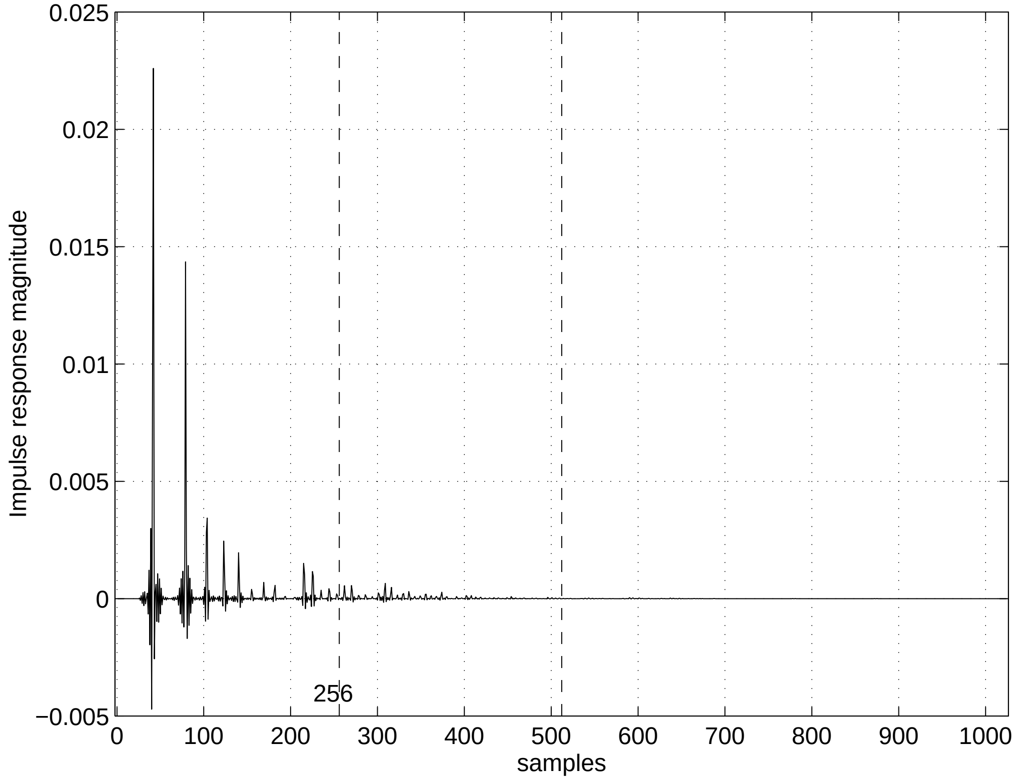


Fig. 2.1 A typical 1024 sample (8000 Hz sampling rate) *s-room* impulse response, where the room dimensions and acoustic properties are described in Table 2.1

rooms, significant delays may occur at 100 ms, which correspond to 800 taps at the same sampling rate. If we want to truncate to a shorter length, some energy will be lost. In the above example, 0.26 % of the total energy is present after the 256th sample. However, the computation reduction achieved by using 256 samples impulse responses outweighs the truncation error incurred. In the remainder of the thesis, impulse responses truncated at the 256th sample will be used.

Remark on reflections

Under certain conditions a reflection can become a distinct "echo". In that case, it is heard consciously as a repetition of the original signal. This is frequently observed outdoors with sound reflections from the walls of houses and cliffs. In closed rooms such experiences are less familiar, since the echoes are masked by the general reverberation of the room and the proximity of the reflective boundaries. Whether a reflection will become an echo or not depends on its delay with respect to the direct sound, on its relative strength, on the nature of the sound signal, and on the presence of other reflections which will eventually mask the reflection under consideration.

2.1.2 Acoustic classification of rooms

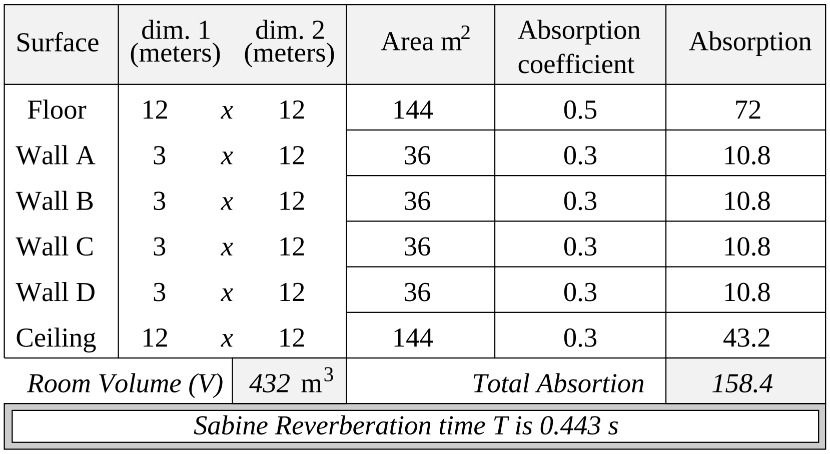
It is known that the acoustic quality of a room depends on aesthetic and psychophysiological criteria, but we will worry about more objective factors which may serve to establish the concept of a "good" room on firm grounds. The hope is to identify a single variable, or at least a finite number of them, which will allow one to characterize the acoustic quality of a room, and which provide unambiguous criteria for deciding what makes a "good" room.

Wallace Clement Sabine experimentally showed that an important criterion is the more and less persistence of a sound after the source is stopped, which he called reverberation. To measure this important effect he defined the reverberation time as the duration for the mean square pressure of a suitably chosen distribution of sound waves to diminish to one-millionth of its original intensity. Under certain statistical assumptions [16], the Sabine reverberation time T for a room of volume V is defined as

$$T = 6\ln(10)\frac{4V}{c\sum_{k}\alpha_k S_k} \tag{2.2}$$

where c is the velocity of the sound in the air, and $\alpha_k S_k$ is the equivalent absorption area for an absorber of true area S_k and absorption coefficient α_k . The absorption coefficient α is related to the reflection coefficient γ by: $\alpha = 1 - \gamma$. An example of Sabine's reverberation time computation is shown in Fig. 2.2.

Note that T does not depend on the observer's position in the room. Hence it is well suited to characterize the overall acoustic properties of a room, neglecting details which may vary from one place to another.



Surface	dim. 1 (meters)		dim. 2 (meters)	Area m^2	Absorption coefficient	Absorption
Floor	12	*x*	12	144	0.5	72
Wall A	3	*x*	12	36	0.3	10.8
Wall B	3	*x*	12	36	0.3	10.8
Wall C	3	*x*	12	36	0.3	10.8
Wall D	3	*x*	12	36	0.3	10.8
Ceiling	12	*x*	12	144	0.3	43.2
Room Volume (V)		*432* m^3			*Total Absortion*	*158.4*
Sabine Reverberation time T is 0.443 s						

Fig. 2.2 An example of Sabine's reverberation time computation

2.2 Echo cancellation

Echo is the phenomenon in which a delayed and distorted version of an original sound or signal is reflected back to the source. Echoes can be generated electrically, due to impedance mismatches at points along the transmission medium (telephone lines). Such echoes are called line echoes. When the telephone connection is between one or more hands-free telephones or between two conference rooms, a major source of echoes is the acoustic coupling between the loudspeaker and the microphone at each end. Such echoes have been called "acoustic echoes", and they will be the major topic of this thesis. Comparing typical impulse responses of these two kinds of echo paths, it becomes obvious that acoustic echo cancellation is a far more challenging task than the line echo cancellation: the duration of the impulse response of the acoustic echo path is usually several times longer. In addition, the acoustic echo path may change rapidly at any time e.g. due to a door opening or a person moving. Although we will be dealing with acoustically generated echoes, we will only consider cancellation of these echoes in the electrical portion of the circuit. We will not discuss the related, but much more difficult, problem of cancelling echoes acoustically.

2.2.1 The Loudspeaker-Microphone model

In hands-free communication, the decoupling of the handset from the head introduces the room as another element in the voice system. The result is multiple reflections of the near-end talker's signal transmitted back to the near-end room and conceived as delayed replicas of the original speech. This acoustic feedback is shown in Fig. 2.3. The nature of the acoustic echo patterns can change rapidly due to variations in the environment, the system must therefore be treated as time varying. Thus, the echo path impulse response — which we will refer to throughout the thesis as the room impulse response— $h(n;k)$ should be characterized by two parameters: the time n and the delay k.

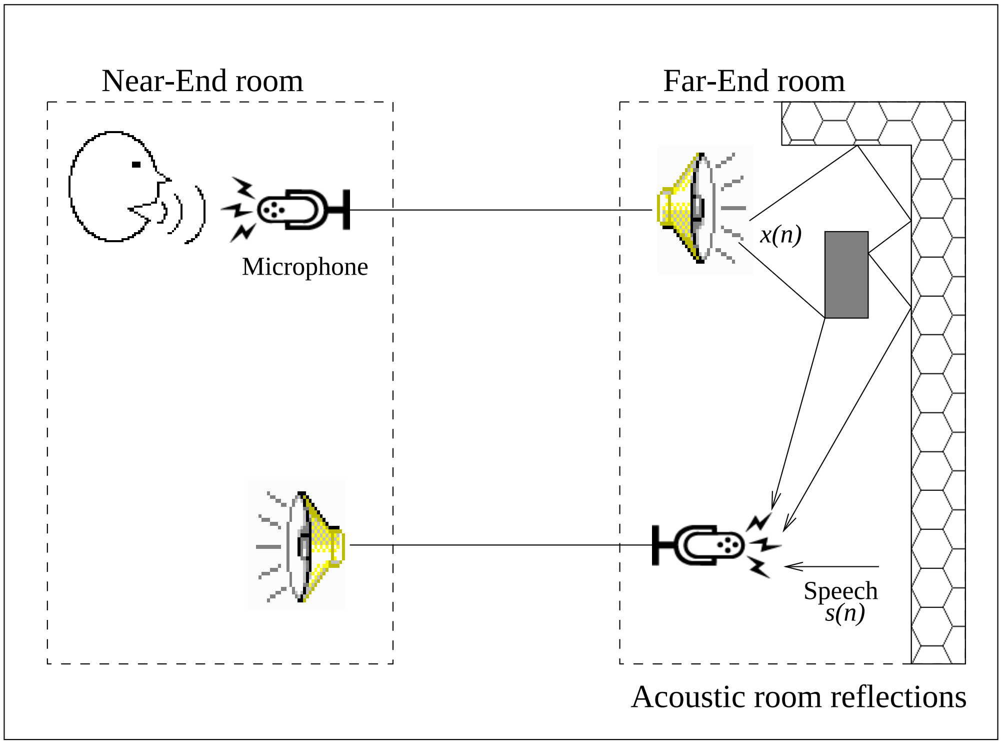


Fig. 2.3 Acoustic coupling between the loudspeaker and the microphone. The loudspeaker output or "the reference signal" is $x(n)$ and the far-end input speech is $s(n)$.

It is assumed in this thesis, that the system is varying slowly enough to allow for adequate tracking by the echo canceller.

2.2.2 Configuration of an acoustic echo canceller

The standard approach to eliminating the acoustical echo is to use a discrete time linear adaptive filter at the receiver where the echo is predominant. This filter is used to quantitatively characterize the acoustical link between the loudspeaker and the microphone, thus generating an electronically synthesized replica of the acoustical echo. This replica is then subtracted [17] from the echo received by the microphone to decouple the loudspeaker and the microphone as shown in Fig. 2.4; the adaptive filter identifies the impulse response $h(n;k)$ between the loudspeaker and the microphone. An echo replica $y(n)$ is then subtracted from the real echo $r(n)$ (which represent the convolution of $x(n)$ with $h(n;k)$) to give the residual echo $e(n)$.

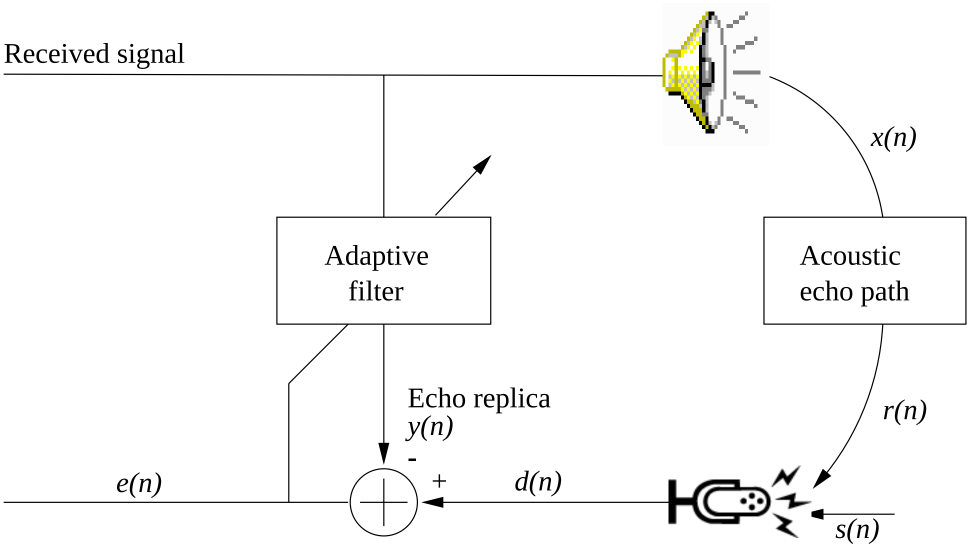


Fig. 2.4 Configuration of an acoustic echo canceller: the received signal and $d(n)$ are available to the adaptive filter and are used to form an estimate of $s(n)$.

A linear transversal (tapped delay line) filter is the chosen realization for the echo-cancelling filter. Perhaps the number of filter parameters could be reduced by modeling the acoustic echo path as a recursive, infinite impulse response (IIR) filter. However, even though the impulse response may be long, the IIR filter still has few degrees of freedom. Which will limit the ability of the adaptive filter to represent all the variations in the impulse response. Therefore, adaptive transversal finite impulse response (FIR) filters as shown in Fig. 2.5 are still the filters of choice. The output of the transversal filter is given by

$$y(n) = \sum_{i=0}^{N-1} w(n;i)x(n-i) \tag{2.3}$$

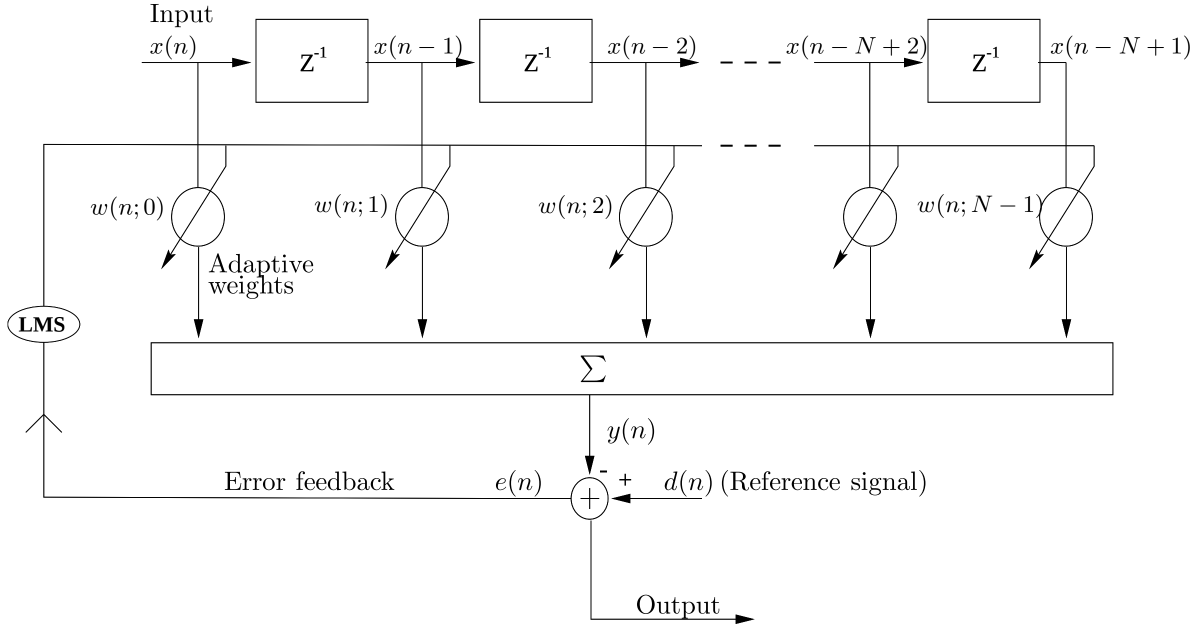


Fig. 2.5 Linear Adaptive filter of length N with tap-delayed inputs

If $w(n;k) = h(n;k)$ for all k (i.e the filter taps are matched to the impulse response coefficients) $y(n)$ will be equal to $r(n)$ and consequently perfect cancellation will be achieved

$$e(n) = s(n) + r(n) - y(n) = s(n) \tag{2.4}$$

It is important to note that the uniqueness of the filter taps that yield perfect cancellation depends (as it will be shown later in the thesis) on the structure of $x(n)$.

2.3 Adaptation Algorithms for linear filtering

The task of the adaptation algorithm is to iteratively minimize some error criterion, where by error we mean a measure of how distant the actual outputs are from the desired outputs.

Let the weight vector be $\mathbf{w}_n = [w(n;0)\ w(n;1)\ldots\ w(n;N-1)]^T$ and let the input vector be $\mathbf{x}_n = [x(n)\ x(n-1)\ldots x(N-1)]^T$ where T denotes the vector transpose. The filter output signal $y(n)$ can be then expressed as the dot product of the weight and the input vectors, $y(n) = \mathbf{w}_n^T\mathbf{x}_n$.

Typically, the error criterion ζ is chosen to be the expectation of the square of the difference $e(n)$ between the reference signal and the filter output (which is called the Mean Square Error). The MSE can be expressed as

$$\zeta(w) = E[e(n)^2] = E[(d(n) - y(n))^2] \tag{2.5}$$

where the expectation $E[.]$ is taken over the input space. Expanding, we obtain

$$\zeta(w) = E[d(n)^2] + \mathbf{w}^T\mathbf{R}\mathbf{w} - 2\mathbf{w}^T\mathbf{p} \tag{2.6}$$

where $\mathbf{R}$ is the autocorrelation matrix of the inputs $\mathbf{R} \overset{def}{=} E[\mathbf{x}_n\mathbf{x}_n^T]$[1] and $\mathbf{p}$ is the cross-correlation between the inputs and the reference signal $\mathbf{p} \overset{def}{=} E[d(n)\mathbf{x}_n]$.

The error $\zeta(w)$ is a quadratic function of the weights and assumes the shape of a hyperparaboloid as illustrated in Fig. 2.6 for a 2-weight case. The sections of the error surface $\zeta = constant$, are hyperellipsoids (ellipses in the 2-D case). The orientation and the shape of these ellipsoids depend on the eigenvalues of the input autocorrelation matrix $\mathbf{R}$. It is easy to show that the axes of the hyperellipsoids are aligned with the eigenvectors of $\mathbf{R}$ and that their lengths are inversely proportional to the square roots of the corresponding eigenvalues. In the 2-D case, if the two eigenvalues are very different the ellipses are thin and long, while, if the eigenvalues are equal the ellipses degenerate into circles.

The weight vector that minimizes the error $\zeta(\mathbf{w})$ corresponds to the "bottom of the bowl". It is obtained mathematically by taking the derivative of $\zeta(\mathbf{w})$ with respect to the weights, setting it to zero, and solving for $\mathbf{w}$.

The solution $\mathbf{w}_{opt}$, which is a special case of the Wiener solution [2], is equal to

$$\mathbf{w}_{opt} = \arg(\min_{\mathbf{w}} \zeta(\mathbf{w})) = \mathbf{R}^{-1}\mathbf{p} \tag{2.7}$$

The minimum achievable mean square error is obtained by replacing $\mathbf{w}$ with $\mathbf{w}_{opt}$ in (2.6)

$$\zeta_{min} = E[d(n)^2] - 2\mathbf{p}^T\mathbf{R}^{-1}\mathbf{p} = E[d(n)^2] - 2\mathbf{p}^T\mathbf{w}_{opt} \tag{2.8}$$

This is the function that the adaptation algorithm has to minimize.

[1] Since the inputs are tap-delayed, the matrix $\mathbf{R}$ is Toeplitz, i.e $R(l,m) = R|l-m|\ \forall(l,m)$

[2] The filter that best maps, in a least squares sense, an input signal into a given desired output is in general of infinite length. Limiting the number of filter taps to N constraints $\mathbf{w}_{opt}$ to be of finite length but it also increases the minimum achievable error.

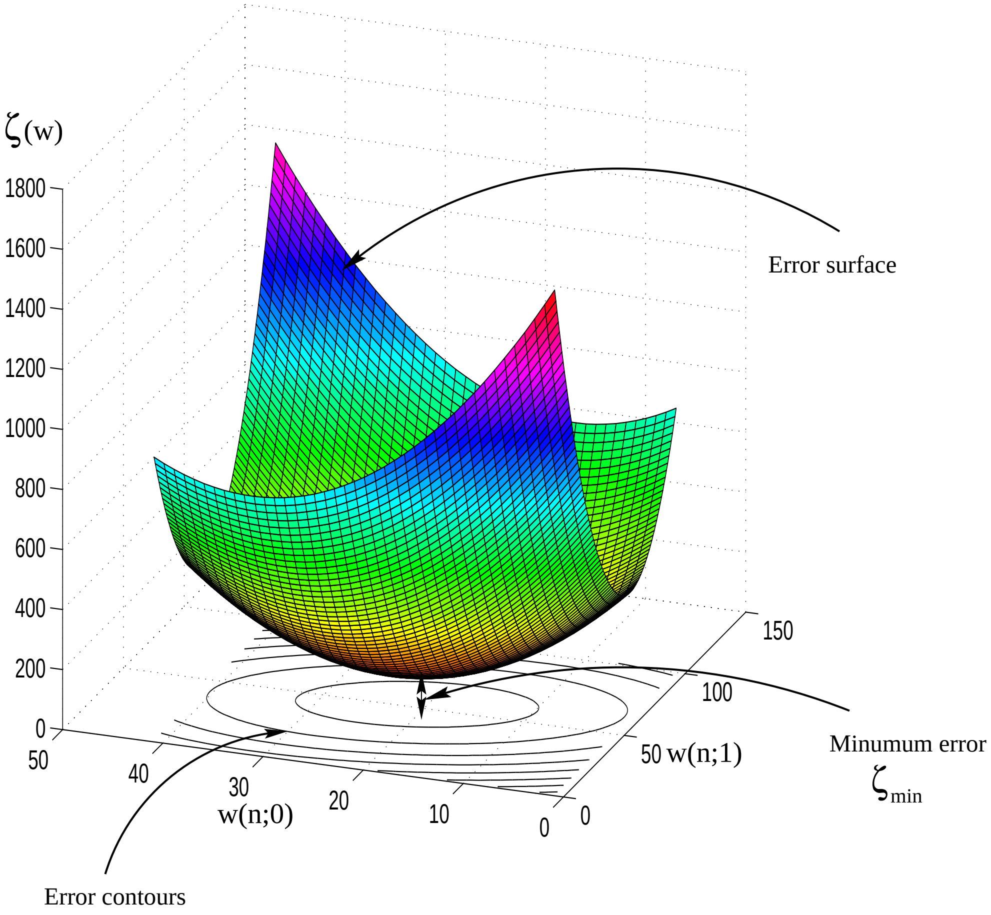


Fig. 2.6 Error surface for a 2-weight adaptive filter

Typically, adaptation algorithm works as follows. The filter weights are initially set to zero or random values. Then, at each iteration, the weights are adjusted so as to travel down the error surface and eventually reach its minimum $\mathbf{w}_{opt}$ or a vicinity of it. The speed at which this happens, the precision of the solution after convergence, the overall robustness of the algorithm, its simplicity, the number of calculations required per iteration, ... are all factors that must be taken into account when comparing different adaptation algorithms. In the following sections, we will successively discuss three families of algorithms: the least mean squares (LMS) algorithms, the recursive least squares (RLS) algorithms, and the transform domain LMS algorithms.

2.3.1 The LMS Algorithm

In this section we summarize the main features of the most widely used adaptation algorithm named *the least-mean-squares (LMS) algorithm* by its originators, Widrow and Hoff (1960)[4], insisting only on the properties that will influence the remaining of this thesis. A significant feature of the LMS algorithm is its simplicity. Moreover, it does not require measurements of the pertinent correlation functions, nor does it require matrix inversion. Indeed, it is the simplicity of the LMS algorithm that has made it the standard against which other adaptive filtering algorithms are benchmarked.

Derivation of the LMS Algorithm

The LMS algorithm minimizes the error function ζ using a stochastic steepest descent approach, that is, at each iteration the weights are updated proportionally to an estimate of the error gradient. Let ∇_n denote the true error gradient at time n, and $\hat{\nabla}_n$ its estimate. The true gradient of the MSE function is given by

$$\nabla_n = \frac{d\zeta(\mathbf{w}_n)}{d\mathbf{w}_n} \tag{2.9}$$

$$= \frac{d\,E[(d(n) - \mathbf{w}_n^T \mathbf{x}_n)^2]}{d\mathbf{w}_n} \tag{2.10}$$

The gradient estimate $\hat{\nabla}_n$ is simply obtained by omitting the expectation in (2.10) hence the name “stochastic gradient”:

$$\hat{\nabla}_n = \frac{d\,(d(n) - \mathbf{w}_n^T \mathbf{x}_n)^2}{d\mathbf{w}_n} \tag{2.11}$$

$$= -2(d(n) - \mathbf{w}_n^T \mathbf{x}_n)\mathbf{x}_n \tag{2.12}$$

$$= -2e(n)\mathbf{x}_n \tag{2.13}$$

Noting that $e(n) = d(n) - \mathbf{w}_n^T \mathbf{x}_n$. By adjusting the weights proportionally to the stochastic gradient instead of the true gradient, LMS follows on the error surface a zig-zag path whose average course is the exact steepest descent path. The whole motivation behind this stochastic approximation is to avoid the cost of computing an expectation over the whole input space at each iteration.

The LMS weight update is thus given by the simple formula:

$$\mathbf{w}_{n+1} = \mathbf{w}_n - \mu\hat{\nabla}_n \tag{2.14}$$

$$= \mathbf{w}_n + 2\mu e(n)\mathbf{x}_n \tag{2.15}$$

where the learning rate μ is a constant that governs the speed of convergence of the algorithm; large μ's allow fast convergence, but also lower the precision of the weight vector after convergence has been reached. Moreover, large μ's can create instability problems. Choosing the right value for μ is an important consideration.

In the case of complex signals, the analogous complex LMS algorithm [8] is given by:

$$\mathbf{w}_{n+1} = \mathbf{w}_n + 2\mu e(n)\mathbf{x}_n^* \tag{2.16}$$

$$e(n) = d(n) - \mathbf{w}_n^T\mathbf{x}_n \tag{2.17}$$

where "$*$" denotes complex conjugation.

Fig. 2.7 shows a signal-flow graph representation of the LMS algorithm in the form of a feedback model. This graph clearly illustrates the simplicity of the LMS algorithm: it requires $2N + 1$ complex multiplications and $2N$ complex additions per iteration yielding $O(N)$ computational cost. In the remainder of the thesis, we will consider only real signals, and the"$*$"will be dropped.

Properties of the LMS Algorithm

The exact analysis of the LMS is quite complicated. Most of the published proofs about the convergence of LMS are based on the average behavior of the algorithm rather than its stochastic behavior. In the following, we consider convergence of the mean tap weights which leads to a simple condition on μ. Other convergence criteria such as convergence of the variance of the tap weights error require further simplifying assumptions and a more complex analysis [8].

Widrow based his preliminary analysis on the exact steepest descent algorithm [4]:

$$\mathbf{w}_{n+1} = \mathbf{w}_n - \mu\nabla_n \tag{2.18}$$

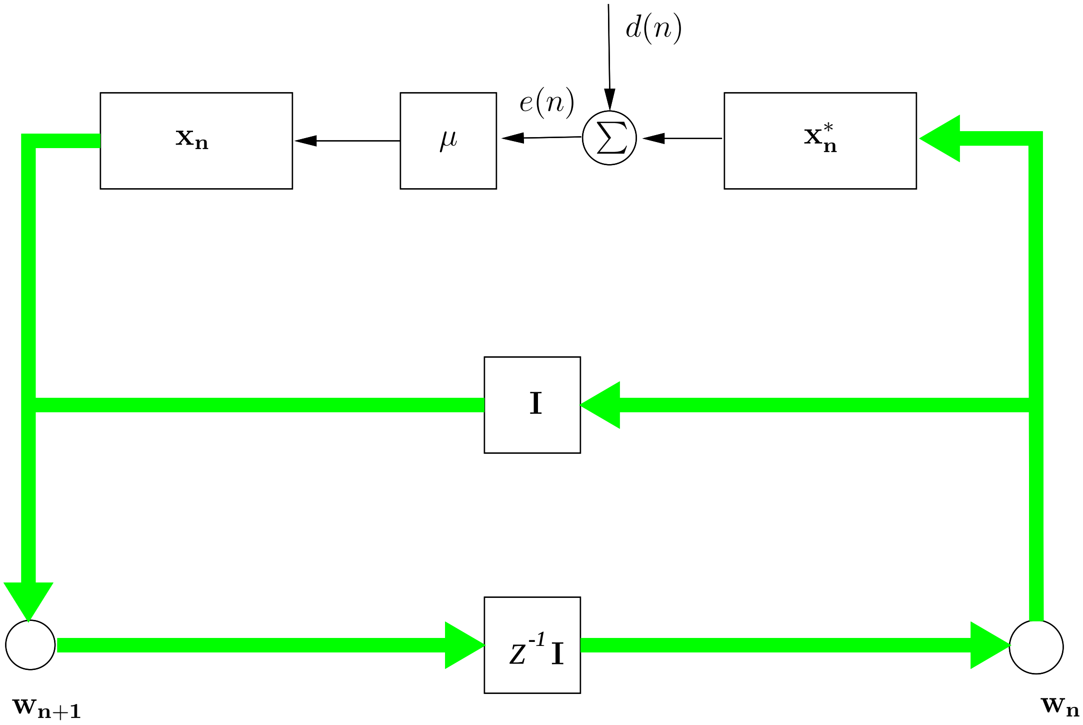


Fig. 2.7 Signal-flow graph representation of the LMS algorithm [8].

The exact error gradient at time n can be expressed as

$$\nabla_n = \frac{d\zeta(\mathbf{w}_n)}{d\mathbf{w}_n} \tag{2.19}$$

$$= 2\mathbf{R}\mathbf{w}_n - 2\mathbf{p} \tag{2.20}$$

$$= 2\mathbf{R}(\mathbf{w}_n - \mathbf{w}_{opt}) \tag{2.21}$$

Let $\mathbf{v}_n$ be the translated weight vector

$$\mathbf{v}_n \overset{def}{=} \mathbf{w}_n - \mathbf{w}_{opt} \tag{2.22}$$

$$\mathbf{v}_{n+1} = (\mathbf{I} - 2\mu\mathbf{R})\mathbf{v}_n \tag{2.23}$$

where $\mathbf{I}$ is the identity matrix. The next step consists in performing a rotation of the translated weights.

$$\mathbf{v}_n^{\dagger} \overset{def}{=} \mathbf{Q}^T\mathbf{v}_n \tag{2.24}$$

where the unitary matrix $\mathbf{Q}$ contains the eigenvectors of $\mathbf{R}$, that is $\mathbf{R} = \mathbf{Q}\Lambda\mathbf{Q}^T$, where Λ is a diagonal matrix containing the eigenvalues of $\mathbf{R}$

$$\Lambda = \begin{bmatrix} \lambda_0 & 0 & \dots & 0 \\ 0 & \lambda_1 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \lambda_{N-1} \end{bmatrix} \tag{2.25}$$

(2.23) can then be rewritten as

$$\mathbf{v}^{\dagger}_{n+1} = (\mathbf{I} - 2\mu\Lambda)\mathbf{v}^{\dagger}_{n} \tag{2.26}$$

This last formula can be iterated from time 0 to time n to give

$$\mathbf{v}^{\dagger}_{n} = (\mathbf{I} - 2\mu\Lambda)^{n}\mathbf{v}^{\dagger}_{0} \tag{2.27}$$

where $\mathbf{v}^{\dagger}_{0}$ is the initial value of the transformed weight vector. Also the error at time n can be expressed in the transform weight space [4]:

$$\zeta_n = \zeta_{min} + \sum_{i=0}^{N} v^{\dagger 2}_{0i}\lambda_i(1 - 2\mu\lambda_i)^{2n} \tag{2.28}$$

The implications of this equation are extremely important. The error decreases as a sum of a geometrical series (or exponentials if the adaptation is interpreted as a continuous process). Each exponential corresponds to one weight and evolves independently of the others (this is due to the decorrelation of the weights resulting from their transformation by the matrix $\mathbf{Q}$). The time constants of the exponentials are given by

$$\tau_i = \frac{1}{4\mu\lambda_i} \tag{2.29}$$

where λ_i is the eigenvalue associated to the i^{th} weight. Small eigenvalues (low energy modes) correspond to long time constants and slow down the overall convergence of the adaptive filter (it should be kept in mind however, that the contribution of the small eigenvalues to the total error is small). High eigenvalues, on the other hand, can cause the modulus of

$(1 - 2\mu\lambda_i)$ to be larger than one, thereby causing the algorithm to diverge. In [4], it was shown that convergence of the weight vector mean is guaranteed for

$$0 < \mu < \frac{2}{\lambda_{max}} \tag{2.30}$$

where λ_{max} is the largest eigenvalue of $\mathbf{R}$. A more detailed discussion of the convergence properties can be found in [8]. The divergence can be avoided by reducing the learning rate, but decreasing μ will have direct consequence of slowing down even further those modes that are already slow because they correspond to small eigenvalues. Input signals with high eigenvalue spread will therefore always result in poor convergence performance. Since speech signals often have near-singular autocorrelation matrices [18], one can therefore expect the LMS algorithm to converge slowly in speech-related applications.

Clearly the problem faced by LMS when its input eigenvalues are very spread apart is due to the fact that it has a single learning rate that must accommodate all the weights. The importance of the problem would be scaled down if we could associate to each decorrelated weight $v^{\dagger}(i)$ a specific learning rate μ_i such that the product $\mu_i\lambda_i$ is more or less constant over i. However this reasoning is valid only in a weight space that has been previously orthogonalized. Otherwise, each weight would be associated to a combination of modes in the error function instead of just one mode.

2.3.2 The RLS Algorithm

The Recursive Least Squares (RLS) algorithm implements recursively an exact least squares solution [8]. As shown before, the Wiener solution for an adaptive filter of finite length is given by $\mathbf{w}_{opt} = \mathbf{R}^{-1}\mathbf{p}$ where $\mathbf{R}$ is the autocorrelation matrix of the inputs and $\mathbf{p}$ is the cross-correlation between inputs and the reference signal. At each time frame, RLS estimates recursively $\mathbf{R}^{-1}$ and $\mathbf{p}$ based on all past data and computes the weight vector as $\mathbf{w}_n \overset{def}{=} \mathbf{R}_n^{-1}\mathbf{p}_n$, which is thus the best to-date approximation to the Wiener solution.

The weight update formula can be written [8] as

$$\mathbf{w}_n = \mathbf{w}_{n-1} + \mu_n \mathbf{R}_{n-1}^{-1} \alpha_n \mathbf{x}_n \tag{2.31}$$

where

$$\alpha_n = d(n) - \mathbf{w}_{n-1}^T \mathbf{x}_n \tag{2.32}$$

$$\mu_n = \frac{\beta^{-1}}{1 + \beta^{-1} \mathbf{x}_n^T \mathbf{R}_{n-1}^{-1} \mathbf{x}_n} \tag{2.33}$$

β is a *constant* slightly smaller than one.

This formulation places in evidence the decorrelation operation performed by RLS on the input data: the stochastic gradient $\alpha_n \mathbf{x}_n$ is pre-multiplied by an estimate of the inverse autocorrelation matrix, which has the effect of decorrelating the inputs of the adaptive filter. This reduces the sensitivity of the algorithm to its input eigenvalue spread. The pre-multiplication by $\mathbf{R}^{-1}$ can unfortunately hurt the stability of the filter if the matrix $\mathbf{R}$ is ill-conditioned; a situation that arises each time the filter contains more weights than necessary.

Compared to LMS, RLS has the advantage of fast convergence rate and low sensitivity to the input eigenvalue spread. But on the other hand, it is computationally intensive ($O(N^2)$) and prone to numerical instabilities. One solution, which we will further discuss in this thesis, consists of preprocessing the inputs to the LMS filter with a fixed transformation that does not depend on the actual input data. The decorrelation will only be approximative, but the computational cost will remain $O(N)$ and the robustness of LMS will be preserved. These algorithms are generally called transform-domain LMS or frequency-domain-LMS algorithms.

2.3.3 Transform-Domain LMS Algorithms

The structure of this family of algorithms consists of two stages [19, 10]:

1. The input vector $\mathbf{x}_n$ is transformed into a corresponding vector of uncorrelated variables.

2. The transformed vector is used as the input to an LMS algorithm.

The first objective may be realized by using the *Karhunen-Lòeve Transform* (KLT). Specifically, given an input vector $\mathbf{x}_n$ of zero mean, drawn from a wide-sense stationary environment, the KLT is defined by (for real-valued data)

$$r_{ni} = q_i^T \mathbf{x}_n \quad i = 0, 1, \ldots, N-1 \tag{2.34}$$

where q_i is the eigenvector associated with the i^{th} eigenvalue λ_i belonging to the autocorrelation matrix $\mathbf{R}$ of the input vector. The individual outputs of the KLT are zero-mean uncorrelated variables:

$$E[r_{ni} r_{nj}] = \begin{cases} \lambda_i, & j = i \\ 0, & j \neq i \end{cases} \tag{2.35}$$

Thus the autocorrelation matrix of the vector produced by the KLT will be the diagonal matrix Λ given by (2.25):

$$E[(\mathbf{Q}^T \mathbf{x}_n)(\mathbf{Q}^T \mathbf{x}_n)^T] = \Lambda \tag{2.36}$$

Multiplying both sides by $\Lambda^{\mathbf{-1}}$ [20], we obtain:

$$E[(\Lambda^{-\frac{1}{2}} \mathbf{Q}^T \mathbf{r}_n)(\Lambda^{-\frac{1}{2}} \mathbf{Q}^T \mathbf{x}_n)^T] = \mathbf{I} \tag{2.37}$$

Hence, using $\Lambda^{-\frac{1}{2}} \mathbf{Q}^T \mathbf{x}_n$ instead of $\mathbf{x}_n$ gives a new correlation matrix $\mathbf{I}$ whose eigenvalues are unity; which means that there is no eigenvalue spread. In addition, it should be noted that the vectors $\Lambda^{-\frac{1}{2}} \mathbf{Q}^T \mathbf{x}_n$ and x_n span the same space. Consequently, the same optimal solution that was achievabale with the original basis, is attainable with the new orthogonal basis.

The second step consists in feeding the transformed vector to an LMS algorithm. The combination of the two stages yields a rapidly converging LMS algorithm[10]:

$$\mathbf{w}_{n+1} = \mathbf{w}_n + 2\mu e(n) \Lambda^{-1} \mathbf{Q}^T \mathbf{x}_n \tag{2.38}$$

$$e(n) = d(n) - \mathbf{w}_n^T \mathbf{Q}^T \mathbf{x}_n \tag{2.39}$$

The KLT is a signal-dependent transformation, the implementation of which requires the estimation of the input autocorrelation matrix and the diagonalization of this matrix. These heavy computations make KLT impractical for real time applications. Fortunately, there exists a variety of fixed, unitary transforms that approximate the KLT for different types of signals.

While no general proof exists that assesses the superiority of one transform over the others, there are some results that show which transform performs better on which class of inputs. For sinusoidal signals, the Discrete Fourier Transform asymptotically approaches an eigendecomposition as $N \rightarrow \infty$ [21]. But most importantly, for first order Markov signals (which are deemed to be sufficiently general in signal processing) the Discrete Cosine Transform (which will be discussed in the next chapter) provides a predetermined set of basis vectors that forms a good approximation to the KLT. The asymptotic equivalence is presented in [11].

Other fixed, data-independent transformations have been considered in the literature to replace the KLT: the Discrete Hartley Transform (DHT) [22], the Walsh-Hadamard transform [23] etc.

In the rest of the thesis, we will focus on the structure in which the DCT is used as a preprocessing transformation and we will refer to this algorithm as DCT-LMS. Fig. 2.8 shows a block diagram of this filter. It consists of two stages, with stage I providing the implementation of a *sliding* DCT algorithm and stages II implementing a normalized version of the LMS algorithm.

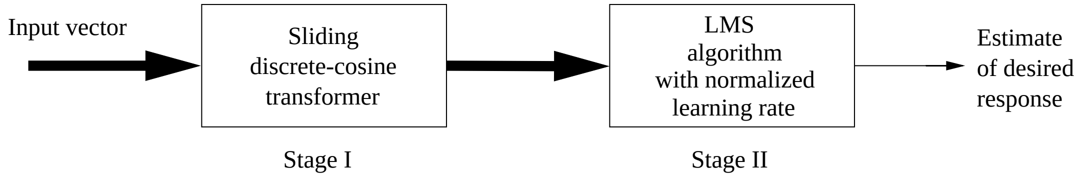

Input vector
Sliding discrete-cosine transformer
LMS algorithm with normalized learning rate
Estimate of desired response
Stage I
Stage II

Fig. 2.8 Block diagram of the DCT-LMS algorithm

Chapter 3

DCT-LMS Algorithm

3.1 The Discrete Cosine Transform

Since its introduction in 1974 by Ahmed *et al.* [24], the discrete cosine transform (DCT) has become a significant tool in many areas of digital signal processing. The original motivation for defining the DCT was that its basis set provides a good approximation to the eigenvectors of the class of Toeplitz matrices that constitutes the autocorrelation matrix of a first order Markov process (with the result that it had a better performance than the Discrete Fourier Transform or DFT). Besides being real, the attractiveness of the DCT was further accentuated by the introduction of fast algorithms.

3.1.1 Preamble

The DCT may be defined in several different ways. We shall now present the definitions for the four discrete cosine transforms as classified by Wang [25]:

In defining the DCT, only the kernel matrix (which will operate on the input vector) will be given. The notation [.] denotes a matrix, the order of which is represented by a subscript inside the pair of square brackets, while the version number is represented by a superscript. With these annotations, the four versions of the DCT matrices are:

$$[C^{I}_{N+1}] = \sqrt{\frac{2}{N}}[k_m k_n \cos(\frac{mn\pi}{N})] \qquad m,n = 0,1,\ldots,N \tag{3.1}$$

$$[C^{II}_{N}] = \sqrt{\frac{2}{N}}[k_m \cos(\frac{m(n+\frac{1}{2})\pi}{N})] \qquad m,n = 0,1,\ldots,N-1 \tag{3.2}$$

$$[C^{III}_{N}] = \sqrt{\frac{2}{N}}[k_n \cos(\frac{n(m+\frac{1}{2})\pi}{N})] \qquad m,n = 0,1,\ldots,N-1 \tag{3.3}$$

$$[C^{IV}_{N}] = \sqrt{\frac{2}{N}}[\cos(\frac{(n+\frac{1}{2})(m+\frac{1}{2})\pi}{N})] \quad m,n = 0,1,\ldots,N-1 \tag{3.4}$$

where

$$k_j = \begin{cases} 1 & \text{if } j \neq 0 \text{ and } j \neq N, \\ \frac{1}{\sqrt{2}} & \text{if } j = 0 \text{ and } j = N. \end{cases} \tag{3.5}$$

The DCT-II is the discrete cosine transform first reported by Ahmed *et al.* [24]. This version of the DCT is the one that will be used in the remainder of the thesis due to its superior decorrelating capabilities [11]. It will be referred to as the DCT. DCT-III is obviously the transpose of DCT-II, and DCT-IV is the shifted version of DCT-I.

The basis functions of the DCT-II with $N = 16$ are shown in Fig. 3.1. These basis functions are related to the zeroes of the Chebyshev polynomials [26]; where the m^{th} orthonormalized one is given by

$$T_m(u) = \sqrt{\frac{2}{N}} k_m \cos[m \cos^{-1}(u)] \tag{3.6}$$

The unitarity of the transform matrices (the rows are orthogonal to one another and have Euclidean norm one) is assured since they are the similarity transform matrices in a diagonalization problem [24]:

$$[C^{II}_{N}]^{-1} = [C^{II}_{N}]^{T} \tag{3.7}$$

From this point on $[C^{II}_{N}]$ will be referred to as $\mathbf{C}$.

Based on (3.2) and (3.7) the forward and inverse DCTs for a signal $x(n)$ can be defined

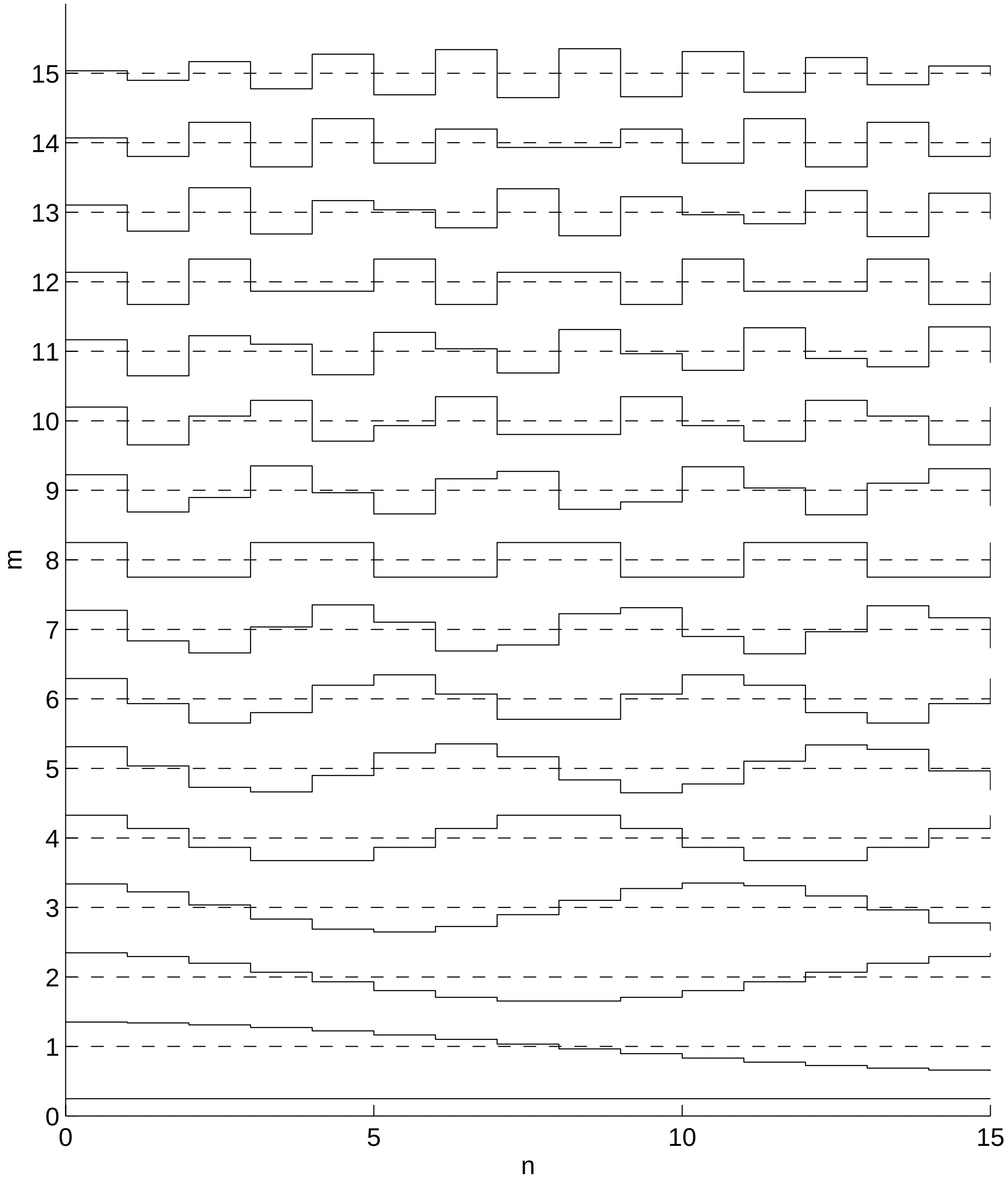


Fig. 3.1 Basis functions of the DCT-II with N=16. There are 16 basis vectors, indexed from 0 to 15 as shown on the y-axis, each of dimensionality 16. The dashed line is the reference (or the zero value) for each plot.

as follows:
Forward:

$$X_{DCT,N}(m) = \sqrt{\frac{2}{N}} k_m \sum_{n=0}^{N-1} x(n) \cos[\frac{(2n+1)m\pi}{2N}] \quad m = 0, \ldots, N-1 \tag{3.8}$$

Inverse:

$$x(n) = \sqrt{\frac{2}{N}} \sum_{m=0}^{N-1} k_m X_{DCT,N}(m) \cos[\frac{(2n+1)m\pi}{2N}] \quad n = 0, \ldots, N-1 \tag{3.9}$$

where k_m is given by (3.5). To simplify the notation, the subscript N in $X_{DCT,N}$ will be dropped whenever there is no ambiguity.

3.1.2 Relation to the DFT

It was shown in [27] that the DCT is related to the DFT of an extended sequence of length $2N$. Let $y(n)$ be defined as

$$y(n) = \begin{cases} x(n) & n = 0, 1, \ldots, N-1 \\ x(2N-1-n) & n = N, N+1, \ldots, 2N-1 \end{cases} \tag{3.10}$$

Note that $y(n)$ is obtained by extending $x(n)$ symmetrically with respect to index $N + 1/2$ as shown in Fig. 3.2.

The $2N$ point DFT of $y(n)$ is defined as:

$$Y_{DFT}(m) = \sqrt{\frac{1}{2N}} \sum_{n=0}^{2N-1} y(n) \exp \frac{-j2\pi mn}{2N} \tag{3.11}$$

$$= \exp \frac{j\pi m}{2N} \sqrt{\frac{2}{N}} \sum_{n=0}^{N-1} x(n) \cos[\frac{(2n+1)m\pi}{2N}] \tag{3.12}$$

Thus the DCT of $x(n)$ can be obtained from Y_{DFT} according to the relation

$$X_{DCT}(m) = k_m \exp \frac{-j\pi m}{2N} Y_{DFT}(m) \quad m = 0, 1, \ldots, N-1 \tag{3.13}$$

Another way of expressing a N point DCT in terms of a $2N$ point DFT can be obtained

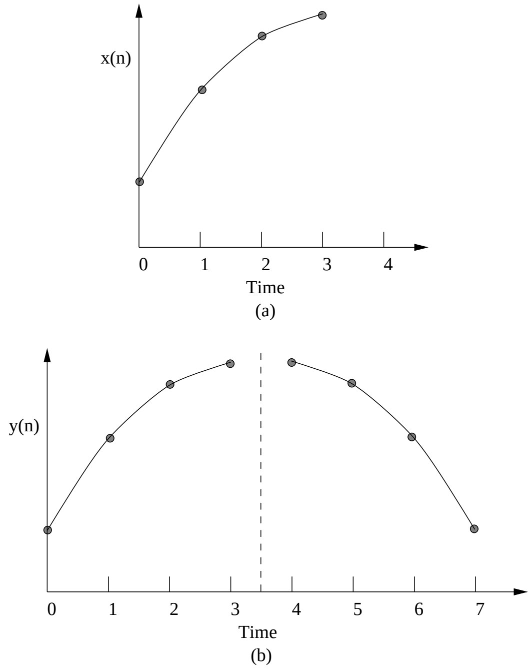


Fig. 3.2 (a) the original signal $x(n)$ of length 4, (b) the symmetrically extended signal $y(n)$ used in the DCT.

by considering the spectral interpretation of the DCT [9]. Given a $2N$ point sequence $u(n)$ such that $u(n) = x(n)$ for $0 \leq n \leq N-1$ and 0 elsewhere. Denoting $|U(m)|$ and θ_m as the magnitude and phase of the $2N$ point DFT of $u(n)$, it was shown in [27] that

$$X_{DCT}(m) = k_m |U(m)| \cos(\theta_m - \frac{\pi m}{2N}) \quad m = 0, 1, \ldots, N-1 \tag{3.14}$$

Thus, the DCT has a spectral envelope which is identical to that of the DFT and a modulating term which imposes a rapidly changing spectral component.

3.1.3 DCT filtering

In many cases, it is more convenient to perform the filtering by multiplication in the spectral domain. In this section, we present the convolution property for the DCT.

It was shown in [11] that the product of the DCT of two sequences is the DCT of the convolution of these two sequences and a third function that will be defined below. In addition, the convolutions are circular convolutions of the symmetrically extended sequences.

Consider two sequences $f(n)$ and $g(n)$, $n = 0, 1, \ldots, N-1$. Construct the $2N$ symmetrically extended sequences, as explained in (3.10), $\hat{f}(n)$ and $\hat{g}(n)$. Let $F_{DCT}(k)$ and $G_{DCT}(k)$ denote the DCT of the two original sequences. The DCT convolution theorem can be stated as follows [11]:

$$DCT[\hat{h}(n) * \hat{f}(n) * \hat{g}(n)] = F_{DCT}(m)G_{DCT}(m) \tag{3.15}$$

For $m = 0, 1, \ldots, N-1$. Where $*$ denotes circular convolution and $\hat{h}(n)$ is defined as

$$\hat{h}(n) = \frac{2}{\sqrt{2N}}\Big\{\Big(\frac{1}{2\sqrt{2}} - 1\Big) + \exp\Big[\frac{j(2n-1)(N-1)\pi}{4N}\Big]\frac{\sin\Big[\frac{(2n-1)\pi}{4}\Big]}{\sin\Big[\frac{(2n-1)\pi}{4N}\Big]}\Big\} \tag{3.16}$$

Since the DCT posses the circular convolution-multiplication property, it is advantageous to carry out the filtering process in the discrete cosine transform domain rather than in the Fourier domain, which would require further forward and inverse transformations [28].

3.2 DCT-LMS algorithm

In this section, we describe the DCT-LMS algorithm and show how to model it in order to simplify its analysis. As shown in Fig 2.8, this algorithm consists of two stages: the first stage (the sliding DCT) acts as a preprocessor that performs the “pseudo-orthogonalization” of the input vector. For that purpose the DCT uses a sliding window, with the computation being performed for each new input sample. This, in turn, enables the LMS algorithm —the second stage— to operate at the incoming data rate as in its conventional form. A general block diagram of the DCT-LMS algorithm is given in Fig. 3.3.

The vector $\mathbf{x}_n$ (consisting of delayed samples of the input signal $x(n)$) is first transformed into another vector $\mathbf{z}_n$:

$$\mathbf{z}_n = \mathbf{C}\,\mathbf{x}_n \tag{3.17}$$

$$\mathbf{z}_n = [z(n;0)\ z(n;1)\ \ldots\ z(n;N-1)]^T \tag{3.18}$$

Referring to Fig. 3.3, $\mathbf{w}_n$ represents the transform domain weight vector, and $d(n)$ the

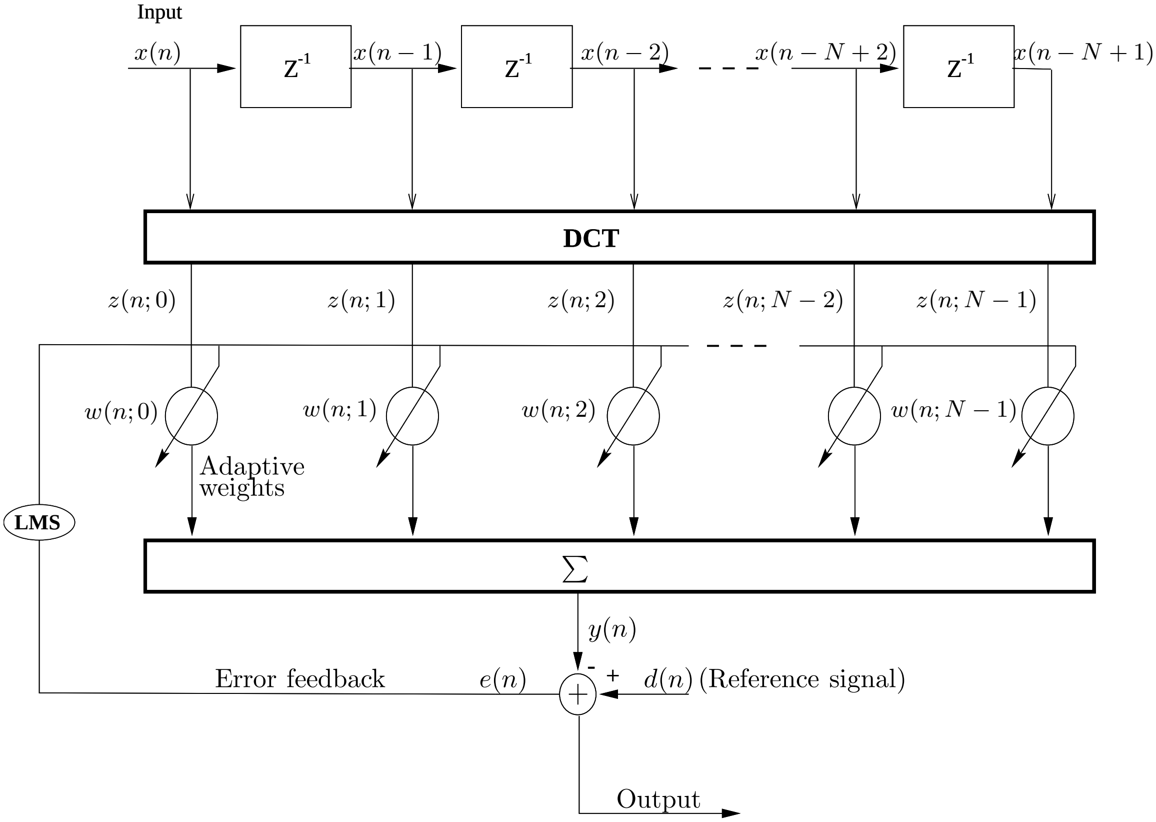


Fig. 3.3 Block diagram of the DCT-LMS adaptive filter.

reference signal. The error signal $e(n)$ is

$$e(n) = d(n) - \mathbf{w}_n^T \mathbf{z}_n \tag{3.19}$$

The weight update equation is

$$w(n+1;i) = w(n;i) + 2\mu_i e(n) z(n;i) \quad i = 0, 1, \dots, N-1 \tag{3.20}$$

where

$$\mu_i = \frac{\mu}{E(z(n;i)^2)} \quad i = 0, 1, \dots, N-1 \tag{3.21}$$

is the adaptive step size for the i^{th} DCT component and μ is a positive constant that governs the rate of convergence.

To highlight the similarity of the above algorithm to the KLT based LMS, described in (2.38), the weight update equation can be rewritten in a vector form as follows:

$$\mathbf{w}_{n+1} = \mathbf{w}_n + 2\mu e(n)(\Lambda^a)^{-1}\mathbf{C}\,\mathbf{x}_n \tag{3.22}$$

where Λ^a is an approximation to Λ, the diagonal matrix containing the eigenvalues of $\mathbf{R}$ and defined in (2.25).

$$\begin{aligned} (\mathbf{C}\,\mathbf{x}_n)(\mathbf{C}\,\mathbf{x}_n)^T &= \Lambda^a & (3.23) \\ &\simeq \Lambda & (3.24) \end{aligned}$$

The adaptive filter will track the DCT of the room impulse response

$$\mathbf{h}_n = [h(n;0)\ h(n;1)\ \dots\ h(n;N-1)]^T. \tag{3.25}$$

If the filter taps match exactly the DCT coefficients of the room impulse response, i.e. if

$\mathbf{w}_n = \mathbf{C}\,\mathbf{h}_n$, perfect cancellation will follow

$$e(n) = d(n) - \mathbf{w}_n^T \mathbf{C}\,\mathbf{x}_n \tag{3.26}$$

$$= s(n) + \mathbf{h}_n^T \mathbf{x}_n - (\mathbf{C}\,\mathbf{h}_n)^T \mathbf{C}\,\mathbf{x}_n \tag{3.27}$$

$$= s(n) + \mathbf{h}_n^T (\mathbf{I} - \mathbf{C}^\mathbf{T}\,\mathbf{C})\mathbf{x}_n \tag{3.28}$$

$$= s(n) \tag{3.29}$$

The reverse implication is not generally true. In the case where some coefficients of $\mathbf{C}\,\mathbf{x}_n$ are zero (forming a “gap” in the transform spectrum of the reference signal), the corresponding components of $\mathbf{w}_n$ will not affect the error $e(n)$, and can thus be arbitrary while still achieving perfect cancellation.

Furthermore, the presence of nulls in the DCT spectrum also affects the weights update. It can be seen from (3.20) that if $z(n;i) = 0 \quad n_o \leq n \leq n_f$ for a given i and if μ is bounded, $w(n;i)$ will “freeze” to the original value $w(n_o, i)$ and retain this value until the gap vanishes ($n = n_f$). This point will be treated in greater detail in section 3.4.

3.3 Intuitive Justification of DCT-LMS

3.3.1 Geometrical Approach

The first geometrical interpretation of the DCT-LMS algorithm was given in [29]. The DCT matrix $\mathbf{C}$ is unitary, and unitary transformations perform only rotations and symmetries, they do not modify the shape of the object they transform.

The MSE of an LMS filter is a quadratic function of its weight vector (see section 2.3). For a constant MSE, the function of the weights is a hyperellipsoid in the $N-$dimensional weight space. The DCT of the input rotates this hyperellipsoid and brings it into approximate alignment with the coordinate axes. The slight imperfection in alignment is due to the leakage in the DCT (as demonstrated in the next section). This is shown in Fig. 3.4 where the 2-D slice of a higher-dimensional ellipsoid is illustrated.

Fig. 3.4(a) shows the original MSE ellipsoid, Fig. 3.4(b) shows the same ellipsoid after transforming the input by a DCT (this will correspond to DCT-LMS MSE). The shape of the ellipsoid, and equivalently the eigenvalues of the autocorrelation matrix, are unchanged. The amplitude normalization, on the other hand, forces the MSE to cross all the coordinate axes at the same distance from the center. Since this transformation is not unitary, it does modify the eigenvalues of the autocorrelation matrix. Indeed, the new ellipsoid is more round and the eigenvalues are less spread apart. The combination of the two steps (DCT with amplitude normalization) approximates asymptotically the optimal eigendecomposition which reshapes the entire error surface so that MSE contours have a round shape, as shown in Fig. 3.4(c), and so that the eigenvalue spread becomes unity [8].

Remark

In the 2D case, the eigenvectors making up the KLT are the same as the DCT basis vectors. Consequently, for this particular case, the DCT is optimal (if normalized properly) and this is illustrated in Fig. 3.4(b) where no misalignment is noticed.

3.3.2 Filtering Approach

The DCT essentially decomposes the input signal into frequency-like bins. Although the DCT does not separate frequencies the way DFT does, it is a powerful signal decorrelator. After identifying the bins, the energy is redistributed almost equally between all the bins and this is done to make all the modes equally fast during adaptation. The N-point DCT can be viewed as N linear transformations from the N-dimensional input vector $\mathbf{x}_n$ to the N outputs $z(n;0), \ldots, z(n;N-1)$. Each transformation is characterized by an impulse response $g_m(n) = \sqrt{\frac{2}{N}} k_m \cos(\frac{m(n+\frac{1}{2})\pi}{N})$. The corresponding transfer function is given by [30]:

$$G_m(\omega) = \sum_{n=0}^{N-1} g_m(n) \exp(j\omega n) \tag{3.30}$$

$$= \sqrt{\frac{2}{N}} k_m \cos(\frac{m\pi}{2N}) \frac{(1-\exp(-jw))(1-(-1)^m \exp(-jwN))}{1-2\cos(\frac{m\pi}{N})\exp(-jw)-\exp(-2jw)} \tag{3.31}$$

Fig. 3.5 shows the magnitude response of one of the 64 transfer functions ($N = 64$). These transfer functions represent a bank of bandpass filters whose central frequencies span the interval $[0, \pi]$ [10]. The sinusoidal nature of the basis of the DCT explains the presence of two symmetrical peaks. Moreover, the sidelobes are the source of the leakage in the DCT bins, yielding some correlations between the outputs.

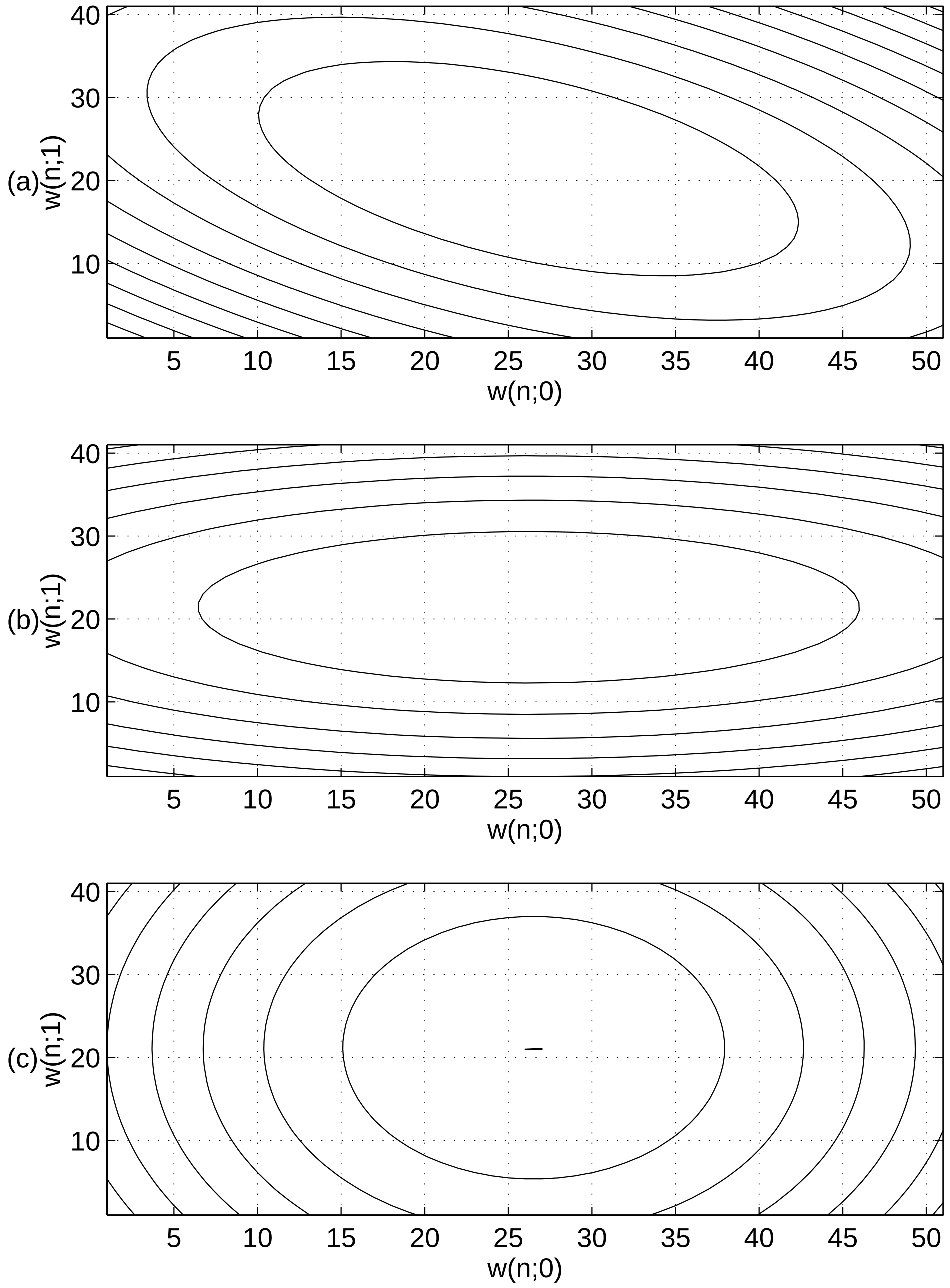


Fig. 3.4 A 2D slice of the MSE hyperellipsoid (of dimension 3) (a) before transformation (b) after DCT (c) after KLT.

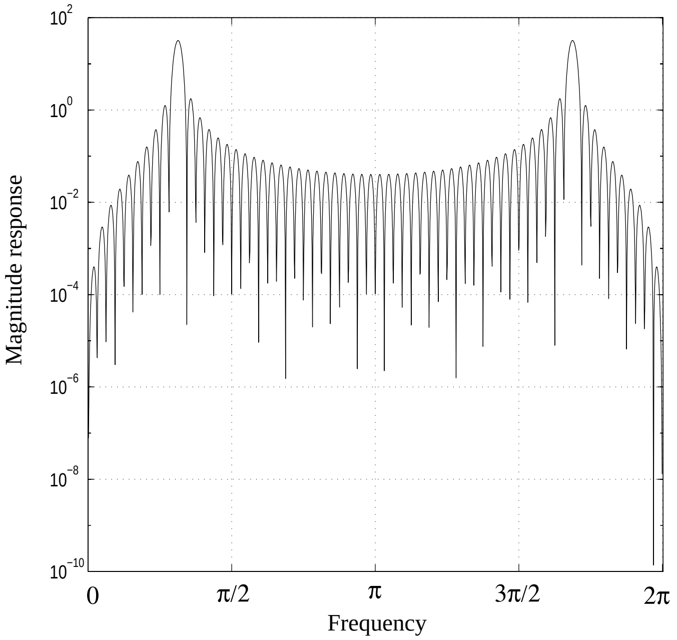


Fig. 3.5 Magnitude of the $20th$ bin transfer function for a 64-point DCT: $|G_{20}(\omega)|^2$.

3.4 Effect of spectrum gaps on the DCT-LMS algorithm

3.4.1 Spectrum gaps

If the DCT of the reference signal is null between the frequencies m_o and m_f we will say that the spectrum of the reference signal contains a gap of size M $(M = m_o - m_f)$ starting at m_o. If a gap is present between time n_o and n_f, the transform vector $\mathbf{z}_n$ will be of the form

$$\mathbf{z}_n = [z(n;0) \ldots z(n,m_o-1)\ 0 \ldots 0\ z(n;m_f+1) \ \ldots\ z(n;N-1)]^T \tag{3.32}$$

for $n_o \le n \le n_f$. An illustration of this observation is given in Fig. 3.6.

One observes that if $\mathbf{z}_n$ is given by (3.32), the weight update equation (3.20) will become

$$w(n+1;m) = w(n;m) \quad \text{for} \quad m_o \le m \le m_f \quad \text{and} \quad n_o \le n \le n_f \tag{3.33}$$

In other words, the filter weights at positions m_o through m_f will freeze and stop adapting;

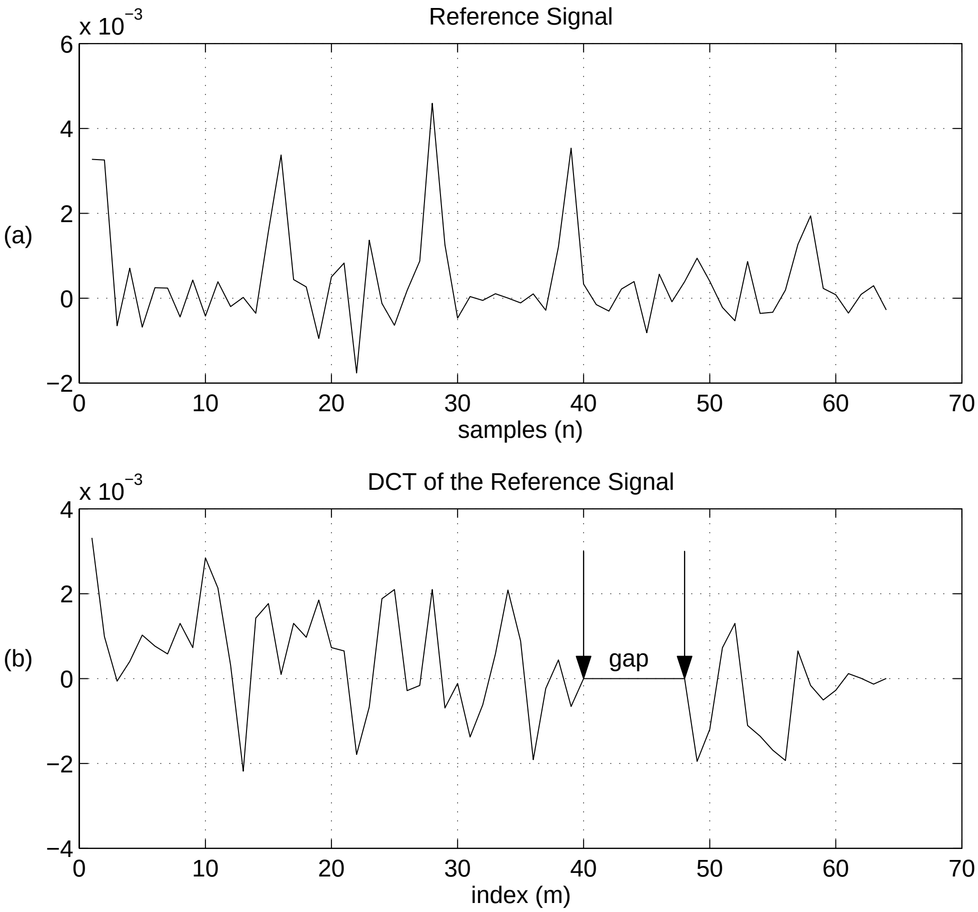


Fig. 3.6 Illustration of the concept of a spectrum gap. In (a) a 64-point frame of the reference signal $\mathbf{x}_n$ is shown. The 64-point DCT of $\mathbf{x}_n$ ($\mathbf{z}_n$) is displayed in (b). Note the presence of a gap of size 8 starting at index 40 in the spectrum of the reference signal.

they will retain the original tap value $w(n_o; m)$. On the other hand, $w(n; m)$ continues to be updated correctly for $m \in [0, m_o - 1] \cup [m_f + 1, N - 1]$.

In summary:

$$w(n; m) \longrightarrow H(n; m) \quad m \in [0, m_o - 1] \quad \text{and} \quad n_o \le n \le n_f \tag{3.34}$$

$$w(n; m) \longrightarrow w(n_o; m) \quad m \in [m_o, m_f] \quad \text{and} \quad n_o \le n \le n_f \tag{3.35}$$

$$w(n; m) \longrightarrow H(n; m) \quad m \in [m_f + 1, N - 1] \quad \text{and} \quad n_o \le n \le n_f \tag{3.36}$$

where H is the DCT of the room impulse response, i.e.,

$$\mathbf{H}_n = \mathbf{C}\,\mathbf{h}_n = \mathbf{C} \begin{bmatrix} h(n; 0) \\ h(n; 1) \\ \dots \\ h(n; N-1) \end{bmatrix} = \begin{bmatrix} H(n; 0) \\ H(n; 1) \\ \dots \\ H(n; N-1) \end{bmatrix} \tag{3.37}$$

The DCT of a typical room impulse response (defined in Table 2.1) is illustrated in Fig. 3.7.

3.4.2 Effect of gaps on the error surface

As shown previously, the presence of a spectrum gap between m_o and m_f allows the corresponding filter weights $\{w(n; m) \quad m \in [m_o, m_f]\}$ to take arbitrary values and still achieve perfect cancellation. In terms of the error surface, this means that the minimum of the MSE is not unique, it is actually independent of the coordinates corresponding to the frozen coefficients. Therefore, the minimum of the MSE will be of dimensionality M. An example of the MSE (ζ) when N=2 and $M = 1$ is shown in Fig. 3.8.

We see from this plot that the error is invariant with respect to one of the weight axis, the y axis. This means that there is an infinity of optimal solutions to the problem, all lying in the "valley" of the error function. In a typical adaptation run, the weight vector —initially set to some random value— will follow the path of steepest descent and settle at the nearest location in the valley.

The above phenomenon happens whenever the size of the LMS filter is overdetermined. Furthermore, the minimum of the corresponding time-domain LMS MSE is also of dimensionality M since the transform is unitary; the DCT only rotates the minimum so that it

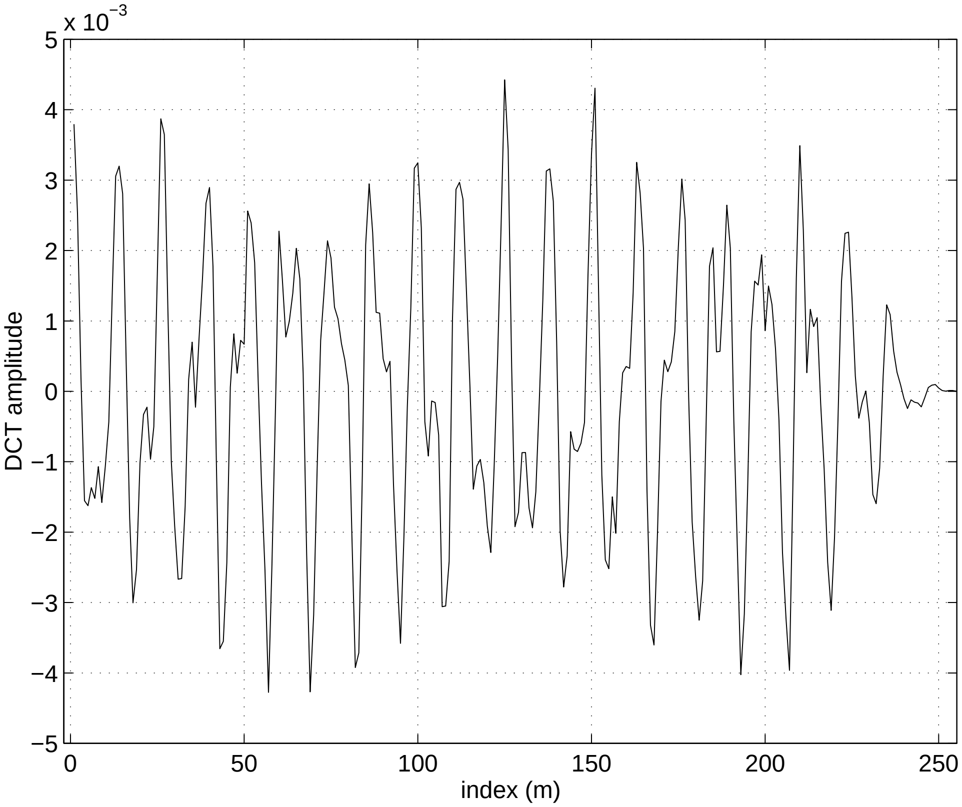


Fig. 3.7 The 256-point DCT of a typical room impulse response. The corresponding room characteristics are given in Table 2.1.

lies parallel to the irrelevant axes.

3.4.3 Meaning of a gap

To motivate later analysis, it is important to know what types of signals have gaps on their DCT spectra. Recalling that,

$$\mathbf{z}_n\,\mathbf{z}_n^T = (\mathbf{C}\,\mathbf{x}_n)(\mathbf{C}\,\mathbf{x}_n)^T = \Lambda^a \simeq \Lambda \tag{3.38}$$

one sees that nulls in $\mathbf{z}_n$ correspond to zeros along the diagonal of Λ^a which approximates the eigenvalues of $\mathbf{R}$. In [31] a signal whose autocorrelation matrix $\mathbf{R}$ has zero eigenvalues (or does not have full rank) is dubbed *rank deficient*. In most cases (certainly the practical ones) rank deficiency of the reference signal autocorrelation matrix is accompanied with

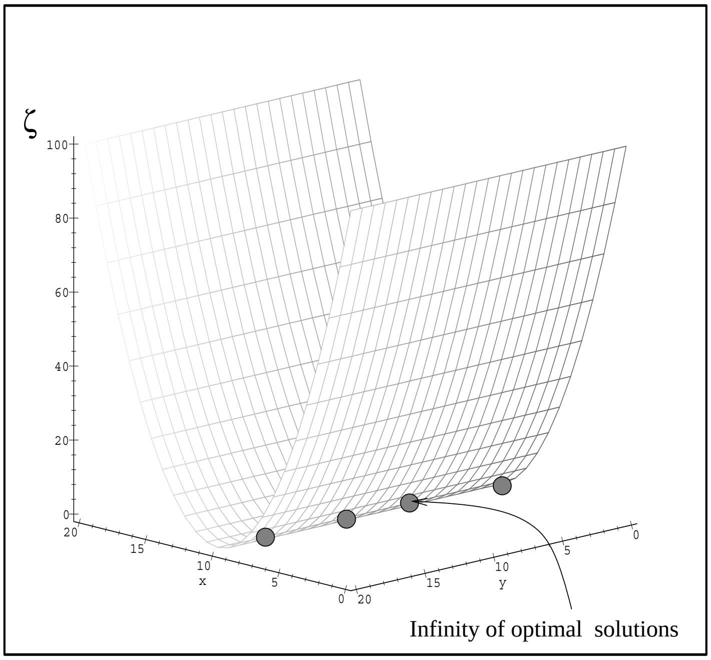


Fig. 3.8 Non Uniqueness of the transform-domain error surface: In this example the number of tap coefficients N is 2 and the dimensionality of the error surface M is 1. It is clear that the y coordinate (which represents one filter tap) does not affect the MSE, this weight can take any value.

the presence of spectrum gaps.

Recalling (3.14), it was shown that the DCT has a spectral envelope which is identical to that of the DFT of an extended sequence. Consequently, gaps in the DCT spectrum correspond to some zero-energy regions of an equivalent frequency spectrum.

3.4.4 Impact of spectral gaps on the convergence speed

For $n_o \leq n \leq n_f$

There exists a gap of size M between m_o and m_f, consequently $\{w(n;m) \quad m_o \leq m \leq m_f\}$ will freeze to the original value $w(n_o;m)$ and stop adapting. On the other hand, the remaining $N-M$ weights are successfully tracking the DCT of the room impulse response.

Thus

$$w(n;m) \longrightarrow \begin{cases} H(n;m) & m \in [0, m_o - 1] \\ w(n_o;m) & m \in [m_o, m_f] \\ H(n;m) & m \in [m_f - 1, N - 1] \end{cases} \tag{3.39}$$

Furthermore, the reference signal autocorrelation matrix has low rank and the minimum of the error surface is of dimensionality M.

For $n \geq n_f$

The reference signal autocorrelation matrix gains full rank and the minimum of transform domain MSE surface collapses to a single point. The gap vanishes and all the filter weights are adapting and converging to $\mathbf{H}_n$:

$$w(n;m) \longrightarrow H(n;m) \quad \forall\, m \tag{3.40}$$

Since M components of $\mathbf{w}_n$ have been frozen to values which are probably obsolete — knowing that the room impulse response is constantly changing— the distance between $\mathbf{w}_n$ and $\mathbf{H}_n$ may be large (many distance measures will be defined later). Noting that the speed of convergence is inversely proportional to the distance, additional convergence time is required for the frozen coefficients to converge from $w(n_o;m)$.

The purpose of the rest of the thesis is to reduce this convergence time by moving the frozen weights toward the optimal point $\mathbf{H}_n$ anticipating a change in the dimensionality of the error surface. In other words, make the weights better prepared to begin adapting once the gap vanishes. This becomes even more important when the room impulse response is changing rapidly, and a large jump in the MSE is expected at the end of a gap.

3.5 Spectral Updating

3.5.1 Objective

The aim is to estimate the changes in the room impulse response (more precisely the DCT of the room impulse response) during the low rank periods by monitoring the variation of the "tracking" coefficients. Latterly, the frozen weights will be updated to follow the changes in the room impulse response. Since DCT-LMS is the algorithm used, all the processing should be done in the DCT domain, hence the term “Spectral Updating”.

Accordingly, the updated weight vector will have the following form:

$$SU[w(n;m)] = \begin{cases} w(n;m) & m \in [0, m_o - 1] \\ w_{su}(n;m) & m \in [m_o, m_f] \\ w(n;m) & m \in [m_f - 1, N - 1] \end{cases} \tag{3.41}$$

Where SU is a particular Spectral Updating method. The DCT-LMS algorithm directly uses $SU[w(n;m)]$ as the new set of tap coefficients. The complete process is described in Fig. 3.9.

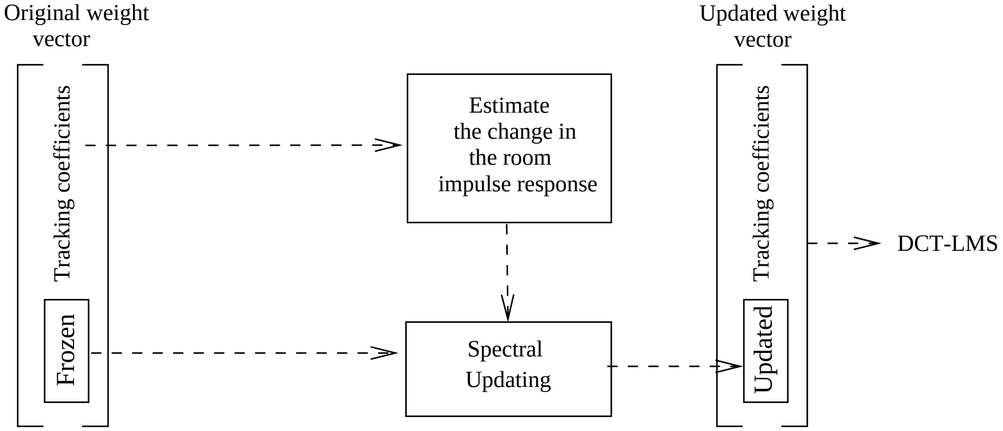


Fig. 3.9 During low rank phase, the frozen coefficients are updated to track the variation of the room impulse response. The change in the room impulse response is estimated by monitoring the adapting weights fluctuations.

3.5.2 Modeling the changes in the room impulse response by a delay

The room impulse response changes with the movement of the source (loudspeaker) or the receiver (microphone) and with the variation of the reflective materials. In teleconferencing, the movement of the talker (to whom the microphone is attached) is the major variable that will alter the room response. Our goal is to quantify the changes in the room impulse response, and come up with a set of parameters based on which the “modified” impulse response can be deduced, given the original response.

A radial movement of the source or the receiver with respect to the other, will delay the room impulse response (the term delay can mean either a time advance or a time lag: positive delay values represent time advance and negative delay values reflect a time lag). Therefore, a simple model will try to represent all the changes in the room impulse response as a time shift. Fig. 3.10 illustrates how we were able to model the effects of the movement of a far-end talker (from position $(1, 1, 1)$ to position $(1.05, 1, 1)$ in the room described in Table 2.1), by a single delay in the room impulse response. Fig. 3.11 illustrates more clearly the similarity of the delayed response to the actual room impulse response. In the remainder of the thesis, we will refer to the movement of the far-end room talker who has the microphone attached to him as the "receiver movement".

A major setback to this model is that multiple images (originating from different walls) will arrive with different delays. In the example given above, the actual delays for all the images (or lobes) are given in Table 3.1. This phenomenon will limit the accuracy of modeling the whole change in the impulse response by a single delay. Fig. 3.12 illustrates the discrepancy between the delayed room impulse response and the actual one due to this fact.

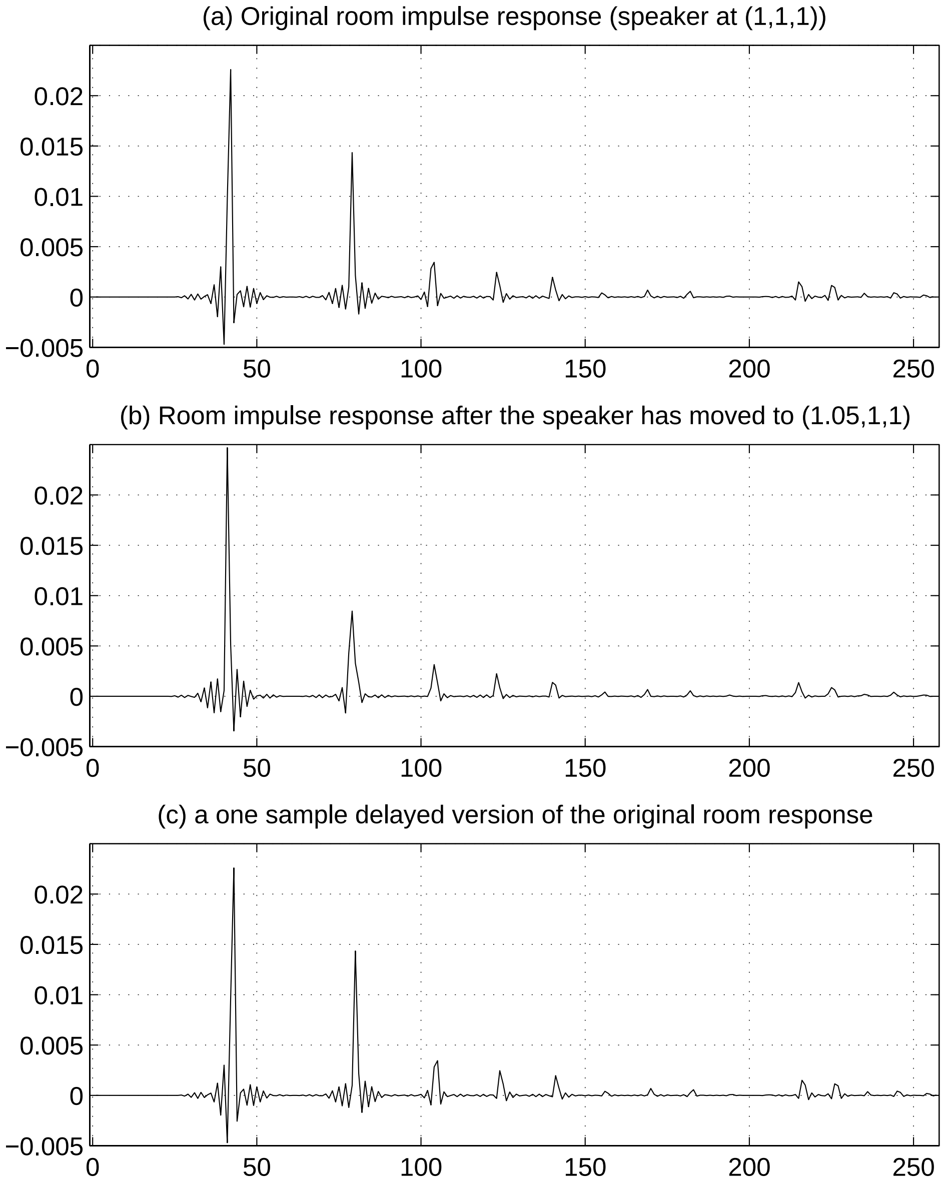
Having the dominant role in the convolution, the main lobe of the impulse response (corresponding to the direct path) will be the major element in estimating the delay.

Table 3.1 Change in the delay for various impulse response lobes when the talker moves from position $(1, 1, 1)$ to position $(1.05, 1, 1)$

Room impulse response Lobe	Change in the delay (samples)
Main	$+1.04$
Second	$+0.80$
Third	$+0.67$
Fourth	-1.13
Fifth	$+0.40$

3.5.3 Spectrum Delay Update

Assume at first, that the reference signal autocorrelation matrix has full rank for $n \leq n_o$. Subsequently, a gap appears in $\mathbf{z}_n$ between n_o and n_f and vanishes afterward for $n \geq n_f$.


(a) Original room impulse response (speaker at (1,1,1))

(b) Room impulse response after the speaker has moved to (1.05,1,1)

(c) a one sample delayed version of the original room response

Fig. 3.10 Change in the room impulse response due to movement of the talker from position $(1, 1, 1)$ to position $(1.05, 1, 1)$. The room dimensions and characteristics are described in Table 2.1. The change in the original room impulse response (a) is modeled by a one sample delay, and the resulting delayed response is shown in (c). This will be used to approximate the actual impulse response given in (b).

.

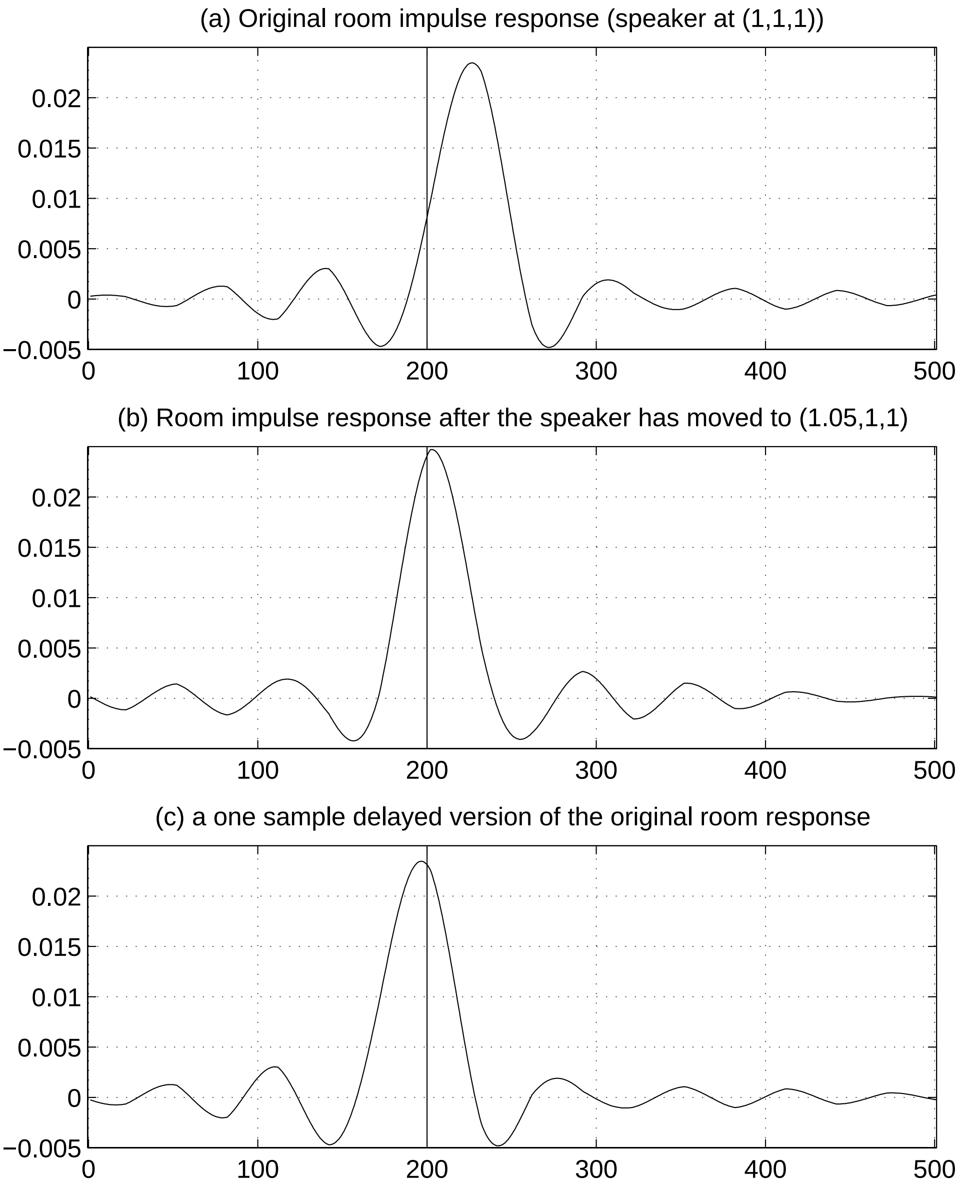


Fig. 3.11 Same as Fig. 3.10 with all the impulse responses upsampled by 30 and only displaying the first lobe of each impulse response. The correlation between the delayed response and the actual response is very high.

.

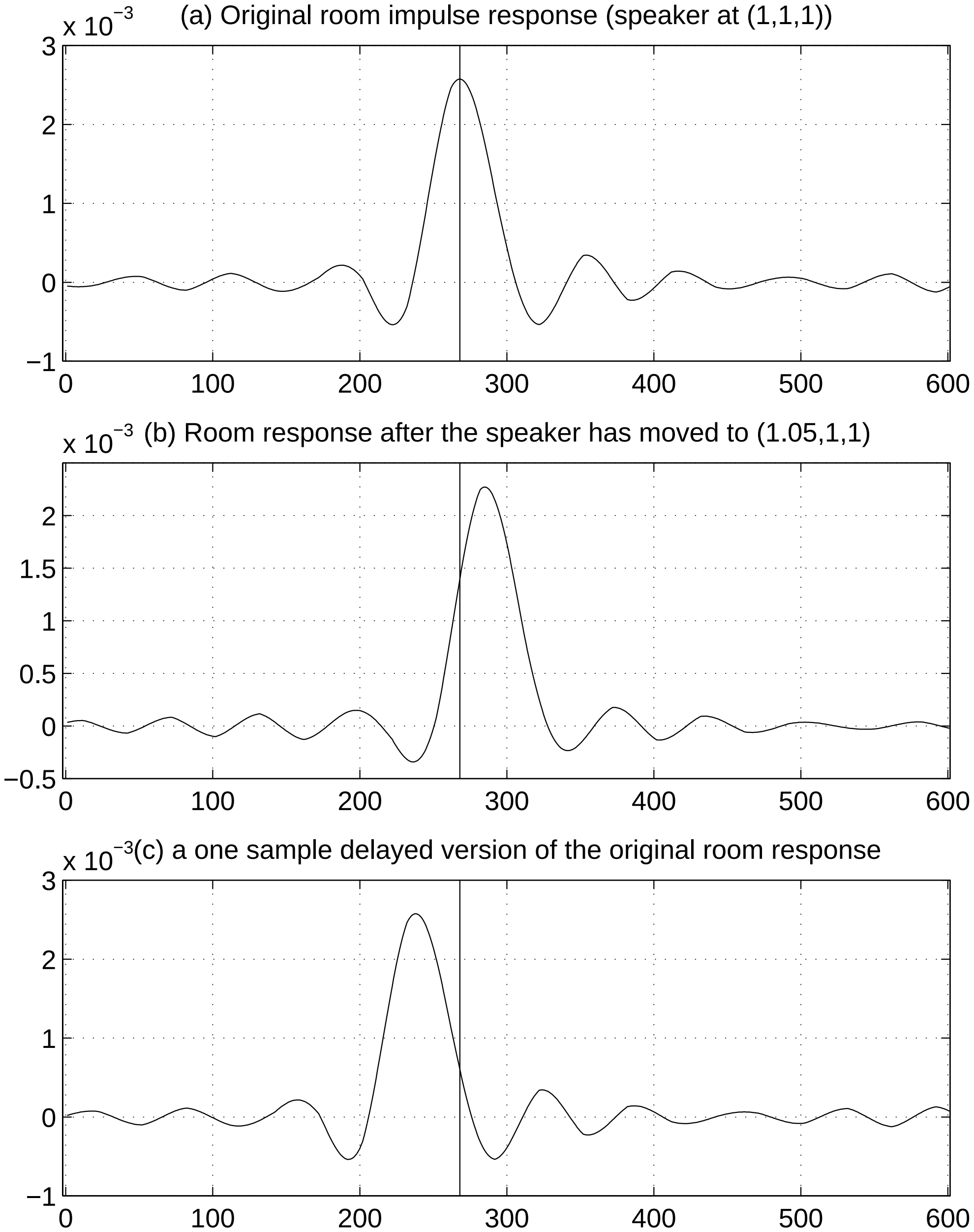


Fig. 3.12 Illustration of the multiple delay phenomenon: the fourth impulse response lobe, which is the image coming from the wall the talker is moving away from, will arrive with a longer delay and will lag the original one as shown in (b). In this case the delayed impulse response shown in (c) will be incorrect.

.

The delay based spectral updating algorithm—which we will refer to as the "Spectrum Delay Update" or SDU algorithm—models the change in the room impulse response as a pure delay, and modifies accordingly all the frozen coefficients to track the delayed version of the original room response. It can be summarized by the following steps:

1. Store $w(n_o - 1; m)$. This vector will serve as a reference for the delay estimation (delay with respect to it). It will be treated as the original room impulse response.
2. Estimate the delay between $w(n; m)$ and $w(n_o - 1; m)$ using only the tracking coefficients ($m \notin [m_o, m_f]$) for $n_o \leq n \leq n_f$. The DCT delay estimation algorithm is the topic of the next chapter.
3. Use the delay estimate to modify the frozen weights. Knowing the original weight vector $\mathbf{w}_{n_o-1}$ and the delay $\delta(n)$, the updated weights $w_{su}(n; m) \quad m \in [m_o, m_f]$ can be calculated.

The block diagram for the Spectrum Delay Update is given in Fig. 3.13.

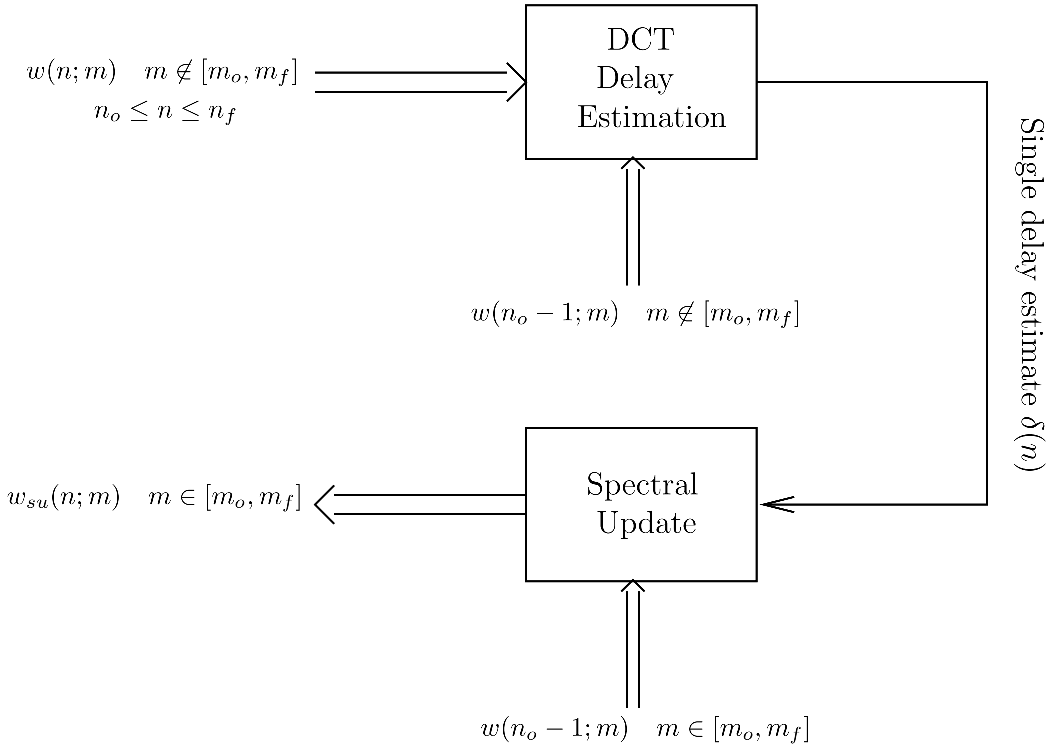

DCT
Delay
Estimation
$w(n;m) \quad m \notin [m_o, m_f]$
$n_o \leq n \leq n_f$
$w(n_o - 1; m) \quad m \notin [m_o, m_f]$
Single delay estimate $\delta(n)$
Spectral
Update
$w_{su}(n;m) \quad m \in [m_o, m_f]$
$w(n_o - 1; m) \quad m \in [m_o, m_f]$

Fig. 3.13 Spectrum Delay Update: the original weight vector $\mathbf{w}_{n_o-1}$ and the delay estimate (obtained from the tracking coefficients) are used to update the frozen coefficients.

Chapter 4

Delay estimation in the DCT domain

4.1 Shift property of the DCT

Given an original data vector $\mathbf{x} = [x(0)\ x(1)\ \dots\ x(N-1)]^T$, the one sample *left*-shifted vector $\mathbf{x}^+$ will be $[x(1)\ x(2)\ \dots\ x(N)]^T$, where $x(N)$ is the new sample pushed into the vector.

The first objective is to express the DCT of the one sample shifted vector, X^+_{DCT}, in terms of the DCT of the original vector, X_{DCT}. Before presenting the shift property of the DCT, we recall the definition of the DCT

$$X_{DCT}(m) = \sqrt{\frac{2}{N}} k_m \sum_{n=0}^{N-1} x(n) \cos\left[\frac{(2n+1)m\pi}{2N}\right] \quad m = 0, \dots, N-1$$
$$k_m = \begin{cases} 1 & \text{if } m \neq 0 \text{ and } m \neq N, \\ \frac{1}{\sqrt{2}} & \text{if } m = 0 \text{ and } m = N. \end{cases} \tag{4.1}$$

and we introduce the N-point Discrete Sine Transform X_{DST} (more precisely the DST-II as labeled by [25]) of the vector $\mathbf{x}$

$$X_{DST}(m) = \sqrt{\frac{2}{N}} k_{m+1} \sum_{n=0}^{N-1} x(n) \sin\left[\frac{(2n+1)(m+1)\pi}{2N}\right] \quad m = 0, \dots, N-1 \tag{4.2}$$

where k_{m+1} is defined in (4.1). Then the shift property of the DCT will be given by [32]

$$\begin{aligned} X^{+}_{DCT}(m) = &\cos\big(\frac{m\pi}{N}\big)X_{DCT}(m) + \sin\big(\frac{m\pi}{N}\big)X_{DST}(m) \\ &+ \sqrt{\frac{2}{N}}k_m \cos\big(\frac{m\pi}{2N}\big)\{(-1)^m x(N) - x(0)\} \\ &m = 0, 1, \ldots, N-1 \end{aligned} \tag{4.3}$$

The proof of this property —not shown in [25]— is presented in Appendix A. The equivalent DST shift property is [32]

$$\begin{aligned} X^{+}_{DST}(m) = &\cos\big(\frac{m\pi}{N}\big)X_{DST}(m) - \sin\big(\frac{m\pi}{N}\big)X_{DCT}(m) \\ &+ \sqrt{\frac{2}{N}}k_m \sin\big(\frac{m\pi}{2N}\big)\{x(0) - (-1)^m x(N)\} \\ &m = 0, 1, \ldots, N-1 \end{aligned} \tag{4.4}$$

Note that if $x(0) = x(N) = 0$, the shift property of the DCT and DST can be simplified:

$$\begin{aligned} X^{+}_{DCT}(m) = &\cos\big(\frac{m\pi}{N}\big)X_{DCT}(m) + \sin\big(\frac{m\pi}{N}\big)X_{DST}(m) \\ X^{+}_{DST}(m) = &\cos\big(\frac{m\pi}{N}\big)X_{DST}(m) - \sin\big(\frac{m\pi}{N}\big)X_{DCT}(m) \\ &m = 0, 1, \ldots, N-1 \end{aligned} \tag{4.5}$$

The next objective is to derive a more general shift formula, where the shift is not restricted to be one sample only (this case is not discussed in the literature).

Consider the k-samples left shifted vector $\mathbf{x}^{k+}$

$$\mathbf{x}^{k+} = [x(k)\ x(k+1)\ \ldots\ x(N+k-1)]^T \tag{4.6}$$

We wish to derive the relation between its DCT X^{k+}_{DCT} and the transforms of the original vector. We will only consider the case where the k pushed samples $(x(N)\ldots x(N+k-1))$ and the first k samples $(x(0)\ldots x(k))$ are zero. This condition is not restrictive when $x(n)$ represents a room impulse response, where the leading samples are always zero (since the speed of sound is finite). On the other hand, this condition will greatly simplify the general shift formula. With the above assumption, the general shift property of the DCT and DST is given by

$$\begin{aligned} X_{DCT}^{k+}(m) &= \cos(\frac{m\pi k}{N}) X_{DCT}(m) + \sin(\frac{m\pi k}{N}) X_{DST}(m) \\ X_{DST}^{k+}(m) &= \cos(\frac{m\pi k}{N}) X_{DST}(m) - \sin(\frac{m\pi k}{N}) X_{DCT}(m) \\ & m = 0, 1, \ldots, N-1 \end{aligned} \tag{4.7}$$

The proof of (4.7) is done by mathematical induction and is given in Appendix B.

It is desirable to obtain the same properties for a *right* shift. Thus, for signals with no energy in the last k samples ($x(N-k) = \ldots = x(N-1) = 0$) and shifting in zero values, the k samples right shift property is the same as (4.7) with $-k$ replacing k.

To complete the analysis, the reciprocal shift property should be derived. From (4.7) it is clear that

$$\begin{aligned} X_{DCT}(m) &= \cos(\frac{m\pi k}{N}) X_{DCT}^{k+}(m) - \sin(\frac{m\pi k}{N}) X_{DST}^{k+}(m) \\ X_{DST}(m) &= \sin(\frac{m\pi k}{N}) X_{DCT}^{k+}(m) + \cos(\frac{m\pi k}{N}) X_{DST}^{k+}(m) \\ & m = 0, 1, \ldots, N-1 \end{aligned} \tag{4.8}$$

4.2 Relation between the DST and the DCT

The DST and the DCT of a sequence are interrelated, we will give in this section a method of composing the DST from the DCT. The procedure for obtaining the sine transform of a sequence $x(n)$ is composed of three steps [33]

1. Invert the signs of all odd numbered data to form a new sequence $\overline{x}(n)$ defined as $\overline{x}(n) = (-1)^n x(n) \quad n = 0, 1, \ldots, N-1$
2. Compute the DCT of $\overline{x}(n)$
3. Reverse the sequence order of the data produced by step 2 to obtain the DST of the sequence $x(n)$.

This procedure may be represented in the form of matrix multiplication. Letting $\mathbf{S}_N$ and $\mathbf{C}_N$ be the DST and DCT matrices of order N respectively, then

$$\mathbf{S}_N = \overline{\mathbf{I}}_N \mathbf{C}_N \mathbf{D}_N \tag{4.9}$$

where $\overline{\mathbf{I}}_N$ is the opposite diagonal identity matrix, $\mathbf{D}_N$ is the odd sign-changing matrix defined as

$$\mathbf{D}_N = \begin{bmatrix} 1 & & & & & 0 \\ & -1 & & & & \\ & & 1 & & & \\ & & & \ddots & & \\ & & & & 1 & \\ 0 & & & & & -1 \end{bmatrix} \tag{4.10}$$

4.3 Estimation of the delay k

Given the shifted vector DCT and the original vector DCT and DST, we want to estimate from (4.7) by how much the shifted vector is delayed or advanced with respect to the original one. Since the DCT and the DST are interrelated, it is enough to provide the DCT, and the corresponding DST will be calculated locally using the above algorithm.

4.3.1 One frequency solution candidates

We start by considering one fixed frequency $m = m_o$, and try to solve for k from this single equation

$$X^{k+}_{DCT}(m_o) = \cos(\frac{m_o \pi k}{N}) X_{DCT}(m_o) - \sin(\frac{m_o \pi k}{N}) X_{DST}(m_o) \tag{4.11}$$

Letting $\alpha = \dfrac{\pi m_o}{N}$, $a = X^{k+}_{DCT}(m_o)$, $b = X_{DCT}(m_o)$ and $c = X_{DST}(m_o)$, (4.11) becomes

$$a = b\cos(\alpha k) + c\sin(\alpha k) \tag{4.12}$$

Using the change of variable $x = \cos(\alpha k)$ and $y = \sin(\alpha k)$, we will have to solve

$$\begin{cases} a = bx + cy \\ x^2 + y^2 = 1 \end{cases} \tag{4.13}$$

Geometrically, this correspond to solving for the intersection of a line with the unit circle as shown in Fig.4.1.

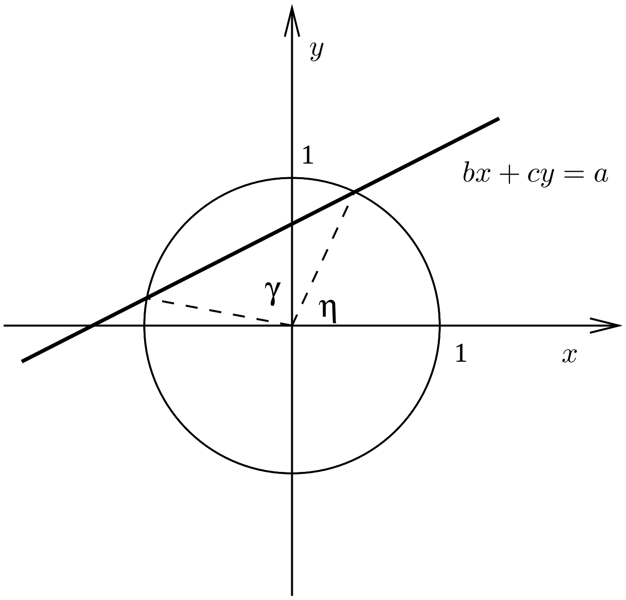


Fig. 4.1 Geometrical interpretation of the delay estimation

For $c \neq 0$, the above system of equations is equivalent to

$$(c^2 + b^2)x^2 + (-2ab)x + (a^2 - c^2) = 0 \tag{4.14}$$

The discriminant of this second order equation is given by

$$\begin{aligned} \Delta^2 &= 4a^2b^2 - 4(c^2 + b^2)(a^2 - c^2) \\ &= 4c^2(b^2 + c^2 - a^2) \end{aligned} \tag{4.15}$$

Thus, if $a^2 < b^2 + c^2$, the two solutions will be given by

$$x = \frac{ab \pm c\sqrt{b^2 + c^2 - a^2}}{c^2 + b^2} \tag{4.16}$$

and the corresponding y will be $y = \dfrac{a - bx}{c}$.

Since the objective is to solve for k, the next step would be to take the arccosine of x. For each x, $\cos^{-1}(x)$ gives two principle angles (between 0 and 2π), but taking into consideration the sign of y, only one angle will be retained. (An alternative solution is to use the arctan2 function). Thus, two uniquely defined principle angles η and γ (modulo 2π) will form the solution of the above problem (as shown in Fig. 4.1). If $c = 0$, on the other hand, x becomes $\dfrac{a}{b}$ which will yield two symmetrical angles η and $\gamma = -\eta$ (the line is parallel to the y-axis).

Consequently, the possible values for the delay k will be

$$\frac{\eta}{\alpha}, \frac{\gamma}{\alpha}, \frac{\eta + 2\pi}{\alpha}, \frac{\gamma + 2\pi}{\alpha}, \frac{\eta + 4\pi}{\alpha}, \frac{\gamma + 4\pi}{\alpha}, \ldots \tag{4.17}$$

or (replacing α by its value)

$$\frac{\eta N}{\pi m_o}, \frac{\gamma N}{\pi m_o}, \frac{\eta N}{\pi m_o} + \frac{2N}{m_o}, \frac{\gamma N}{\pi m_o} + \frac{2N}{m_o}, \frac{\eta N}{\pi m_o} + \frac{4N}{m_o}, \frac{\eta N}{\pi m_o} + \frac{4N}{m_o}, \frac{\gamma N}{\pi m_o} + \frac{4N}{m_o}, \ldots \tag{4.18}$$

In fact, k is only modulo N the above sequence, because of the wrap around. Therefore, not all the above values are distinct. The next objective is to identify and enumerate the distinct solutions.

In the two sequences

$$\begin{aligned} k_n^a &= \frac{\eta N}{\pi m_o} + \frac{2nN}{m_o} \\ k_n^a &= \frac{\gamma N}{\pi m_o} + \frac{2nN}{m_o} \\ n &= 0, 1, 2, \ldots \end{aligned} \tag{4.19}$$

the solutions start repeating when $\dfrac{2nN}{m_o}$ becomes a multiple of N. The smallest n (or n_s) satisfying the above condition determines the number of distinct solutions.

If m_o is even, n_s will be $\frac{m_o}{2}$, yielding $\frac{m_o}{2}$ distinct solutions per sequence. The overall solution (sequences a and b combined) might have some overlap between the two sequences, thus, we can only give the maximum number of possible distinct solutions which is $2\frac{m_o}{2} = 2m_o$. Similarly, if m_o is odd, n_s will be m_o and the maximum number of possible distinct solutions will be again $2m_o$.

Principle angle solution

If we want to obtain the delay k from the principle angles (which will greatly simplify the estimation procedure), k should be such that $k < 2N/m_o$. In other words, if we know the maximum possible delay k_{max}, choosing m_o such that $m_o < 2N/k_{max}$ will guarantee that k is obtained from the principle angles. It is important to note that for $k_{max} \ll N$, it is always possible to solve for k using only the principle angles (assuming there is energy at the first frequencies).

4.3.2 No solution case

If the line does not intersect with the circle, i.e if $a^2 > b^2 + c^2$ in (4.14), the two vectors are not exact delayed versions of each others.

In this case, the closest point to the line on the unit circle (point z in Fig. 4.2) should be found, the corresponding angle η will provide an approximative delay estimate. The procedure to find z is as follow:

1. Obtain the equation of the line orthogonal to $bx + cy = a$ passing through the origin: its leading vector is (b, c), thus, its equation would be $-cx + by = 0$.

2. Find the intersection of this line with the unit circle (two possible points).

3. The point that is closest to the line $bx + cy = a$ will be z.

4.3.3 Combining various solutions to form a single delay estimate

In practice, only frequencies between some m_{min} and m_{max} are used in the estimation process. Our task is to obtain the best delay estimate based on the knowledge of all candidate solutions in the available frequency range.

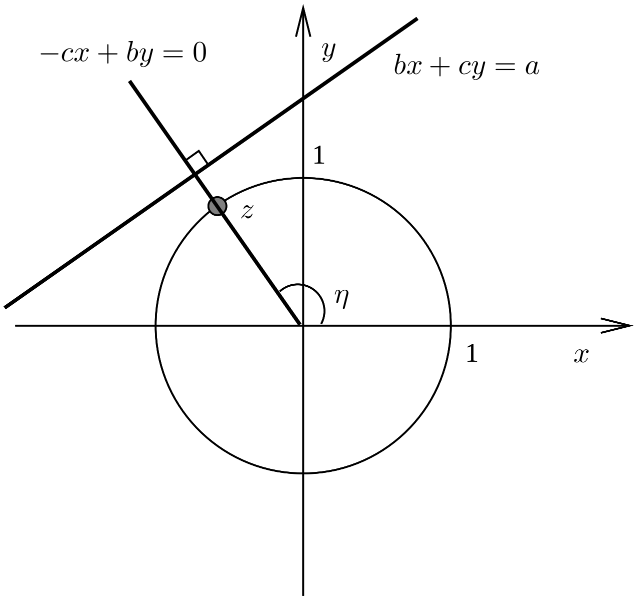


Fig. 4.2 Obtaining an approximative delay estimate when there is no solution.

Assuming that the shift is frequency-independent, the actual delay should appear as one of the solution candidates at each frequency. Therefore, our objective is to find the path through the solution candidates with the minimum variance. This path will identify the actual delay.

Correspondingly, the algorithm for obtaining a single delay estimate could be summarized as follow:

Step 1 From each solution candidate at m_{min}, initiate a path (line).

Step 2 Connect the closest points (in terms of the Euclidean Distance), at consecutive frequencies.

Step 3 Associate with each path a metric defined as the sum of the deviations (square of the distance between the point and the line).

Step 4 Compare all the metrics and save the path with the smallest metric (called the *survivor* path).

Step 5 Deduce the delay from the *survivor* path.

A simple illustration of the above algorithm is shown in Fig. 4.3.

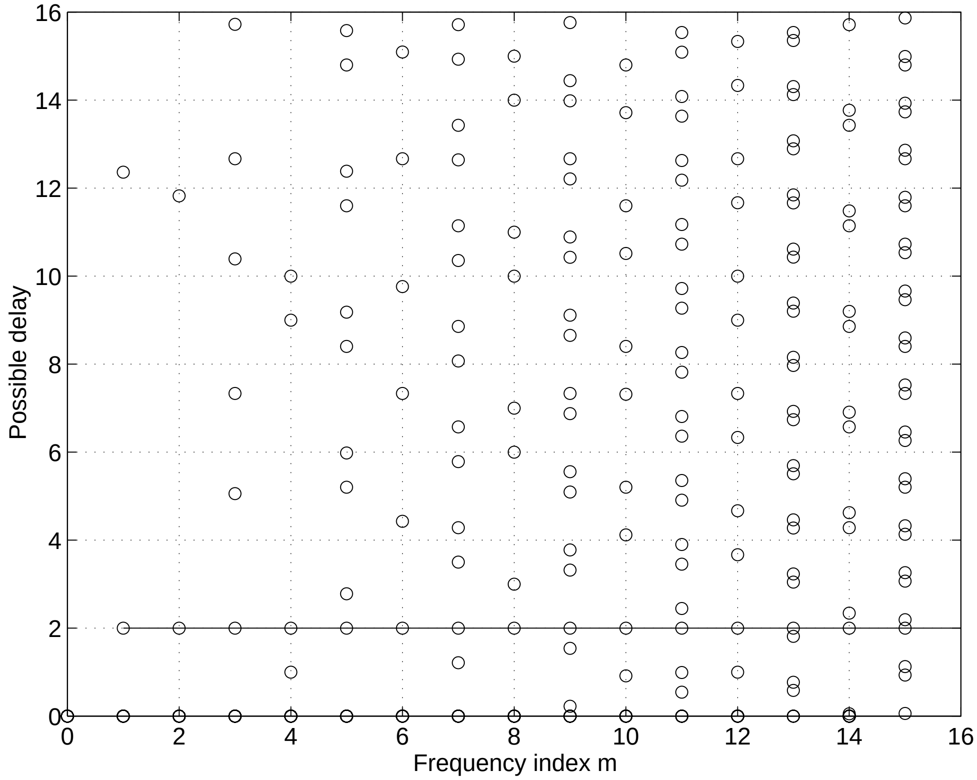


Fig. 4.3 Example of how to obtain the best delay path. The vectors used in the estimation process are of dimensionality $N = 16$ (which corresponds also to the maximum possible delay). With $m_{min} = 1$ and $m_{max} = 15$, the *survivor* path (the solid line) yield a delay estimate of 2 samples.

In determining the *survivor* path, it was proved experimentally that disregarding some outliers (with a threshold to be determined) improves the accuracy of the delay estimation process.

4.4 Delay estimation example

We want to apply the delay estimation algorithm developed in the previous section to model a specific receiver movement (in various room environment) as a delay in the impulse response.

4.4.1 Settings

In a room of dimensions 5 × 4 × 3 m, let the receiver be located at position $(1, 1, 1)$ and the source at position $(3, 2, 1)$. The receiver is moving in the x-direction with an unknown speed as shown in Fig. 4.4.

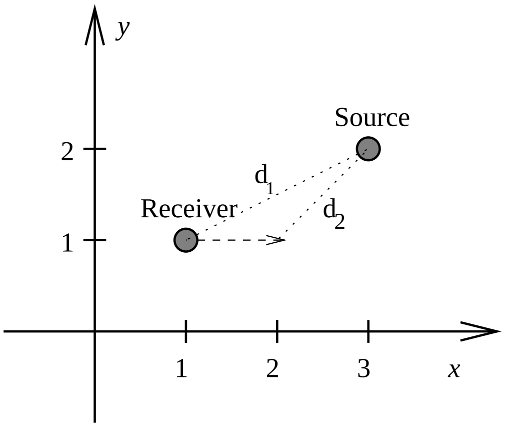


Fig. 4.4 Top view of the receiver movement

The time lag with which an excitation arrives at the receiver depend on the source-receiver distance d and the speed of sound c. Therefore, the difference in the arrival times or equivalently the relative delay between the impulse responses, when the receiver is moving as shown in Fig. 4.4, will depend on $d_1 - d_2$. More precisely, the actual delay (which we will try to estimate) for an x-direction movement of Δx with a sampling frequency f_s will be

$$\frac{(d_1 - d_2) f_s}{c} \tag{4.20}$$

samples. Where $d_1 = \sqrt{5}$ and $d_2 = \sqrt{(3 - 1 - \Delta x)^2 + 1}$.

Fig. 4.5 shows the evolution of the impulse response when Δx increases from 0 to 0.1 meter, with $f_s = 8000 Hz$.

Before proceeding, it is important to obtain some limits on the displacement of a person. The fastest walking speed recorded in the Olympics was 3.8 m/s, it would be fair to claim that a talker in teleconferencing would have a slower motion. Assuming that the DCT coefficients are updated each 256 samples or 32 ms (which corresponds to the zero overlap

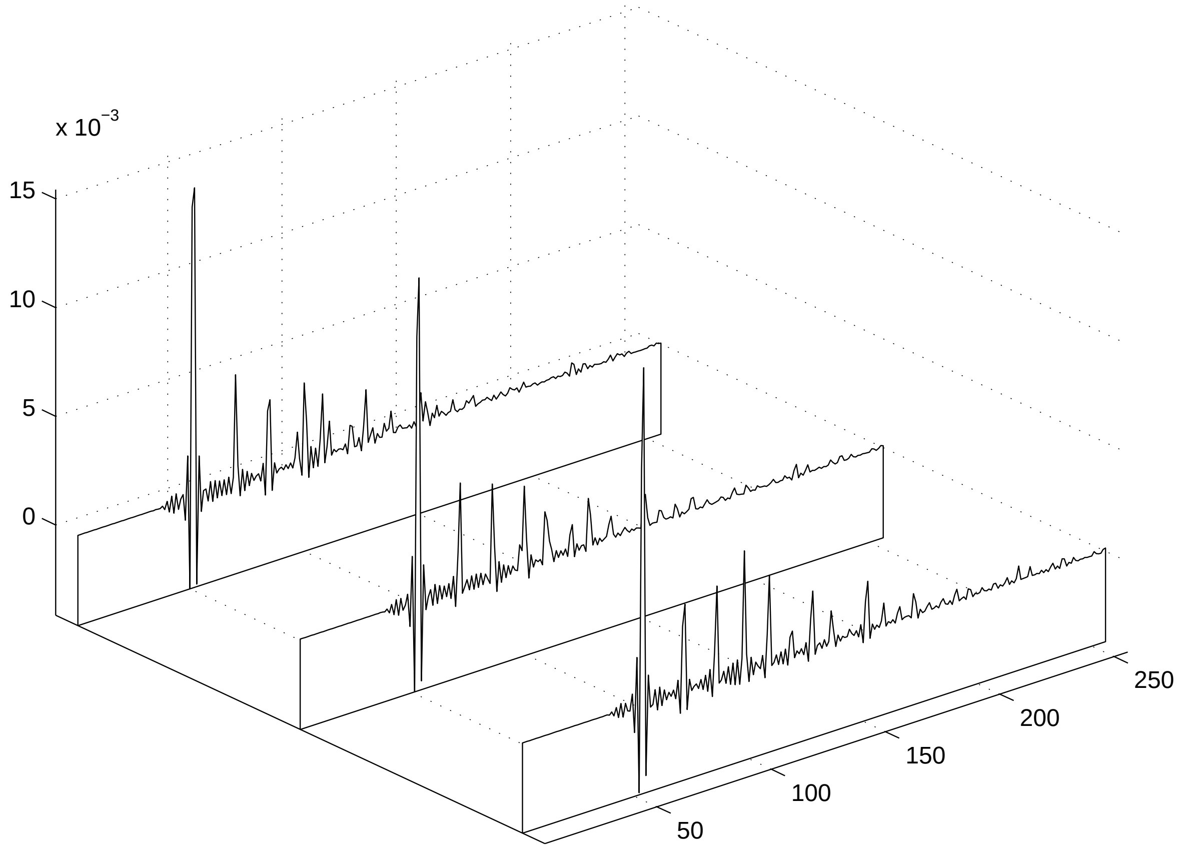


Fig. 4.5 Waterfall display illustrating the effect of the displacement of the talker on the room impulse response. The original impulse response (receiver located at position (1,1,1)) is the first plot (toward the time axis). The second plot shows the impulse response when $\Delta x = 0.05$ m and the last one displays the impulse response when $\Delta x = 0.1$ m. It is clear that as Δx increases the relative delay increases.

case), the maximum displacement of the talker will be 0.12 m. If we update the DCT coefficients more frequently (approaching the continuous update considered before), the consecutive DCT spectra will have some overlap and the maximum displacement of the talker will be less than 0.12 m, but the computation load will be bigger.

For this example, m_{min} and m_{max} are chosen to be 36 and 210 respectively, and the outliers threshold is taken to be 0.85. For each talker movement, two delay estimates are computed depending on whether we approximate the “no solution” cases or not.

4.4.2 Results

Four different acoustic environments, characterized by Sabine reverberation time T_s, are considered to test the delay estimation algorithm (with the room impulse responses generated by the *s-room* algorithm described in Chapter 2 at a sampling frequency of 8000 Hz):

- “Perfect” room environment: all reflection coefficients are zero and $T_s = 0.103$ s. The resulting estimated delays are given in Table 4.1.
- “Good” room environment: walls reflection coefficients are 0.2 and those of the floor and ceiling are 0.1, yielding a reverberation time of 0.123 s. The results are shown in Table 4.2.
- “Medium” room environment: all reflection coefficients are set to 0.4, T_s is consequently 0.173 s. Table 4.3 gives the estimated delays.
- “Bad” room environment: walls reflection coefficients are 0.8 and those of the floor and ceiling are 0.4 resulting in a reverberation time of 0.280 s. The corresponding estimated delays are shown in Table. 4.4.

For each acoustic environment, the average difference between the estimated delay and the actual delay value is computed. The results are given in Table 4.5, they clearly demonstrate that the delay estimation algorithm is performing well. In the worst case (“bad” room), the estimates and the actual delays differ by only 0.14 sample on average.

Remark

There exist simple time-domain delay estimation algorithms which yield accurate results. However, in the DCT-LMS algorithm, the time waveform are not available, as the filter

Table 4.1 Delay estimation results for the "Perfect" room example. All reflection coefficients are zero and $T_s = 0.103$ s.

Movement (x-direction)	Actual delay	Estimated delay (Approx. for $\Delta < 0$)	Estimated delay (Ignoring $\Delta < 0$ cases)
0.05 m	1.05	1.05	1.05
0.06 m	1.26	1.26	1.26
0.075 m	1.58	1.58	1.57
0.09 m	1.88	1.89	1.87
0.10 m	2.09	2.10	2.08
0.12 m	2.51	2.51	2.50

Table 4.2 Delay estimation results for the "Good" room example. Walls reflection coefficients are 0.2 and those of the floor and ceiling are 0.1, T_s =0.123 s.

Movement (x-direction)	Actual delay	Estimated delay (Approx. for $\Delta < 0$)	Estimated delay (Ignoring $\Delta < 0$ cases)
0.05 m	1.05	1.05	1.05
0.06 m	1.26	1.30	1.31
0.075 m	1.58	1.60	1.59
0.09 m	1.88	1.91	2.24
0.10 m	2.09	2.11	2.18
0.12 m	2.51	2.50	2.50

weights track the DCT of the room impulse response. Therefore, if time-domain delay estimation algorithms were to be used, the original and delayed filter tap vectors would have to be transformed to the time domain (by an inverse DCT) prior to the estimation process. This will slow down the convergence of the adaptive filter and defeat the purpose of the introduction of Spectrum Delay Update.

The maximum correlation method is one of the most effective-time domain delay estimation algorithms: to obtain the relative delay between an original signal and a delayed version of it, the original signal is delayed progressively (with one sample increment) and then multiplied by the second signal. The resulting cross-correlation is then normalized by dividing by the square root of the product of the energy of the two signals. The maxi-

Table 4.3 Delay estimation results for the "Medium" room example. All reflection coefficients are set to 0.4, T_s=0.173 s.

Movement (x-direction)	Actual delay	Estimated delay (Approx. for $\Delta < 0$)	Estimated delay (Ignoring $\Delta < 0$ cases)
0.05 m	1.05	1.12	1.14
0.06 m	1.26	1.25	1.29
0.075 m	1.58	1.62	1.68
0.09 m	1.88	1.88	1.85
0.10 m	2.09	2.08	2.12
0.12 m	2.51	2.45	2.56

Table 4.4 Delay estimation results for the "Bad" room example. Walls reflection coefficients are 0.8 and those of the floor and ceiling are 0.4, $T_s = 0.280$ s.

Movement (x-direction)	Actual delay	Estimated delay (Approximating $\Delta < 0$ cases)	Estimated delay (Ignoring $\Delta < 0$ cases)
0.05 m	1.05	1.33	1.40
0.06 m	1.26	1.25	1.37
0.075 m	1.58	1.66	1.78
0.09 m	1.88	1.94	3.15
0.10 m	2.09	2.09	2.42
0.12 m	2.51	2.95	2.55

mum of the normalized cross-correlation yields the relative delay between the two signals as shown in Fig. 4.6.

To obtain a more precise delay estimate—to the fraction of a sample— both signals should be upsampled. In the above example, the delays estimated with this method (upsampling the weights 20 times) are on average 0.01 sample different from the actual delays in all acoustic environments.

Table 4.5 Average differences between the actual and estimated delays (Approximating $\Delta < 0$ cases)

Room environment	Average difference
Perfect	0.01 sample
Good	0.02 sample
Medium	0.03 sample
Bad	0.14 sample

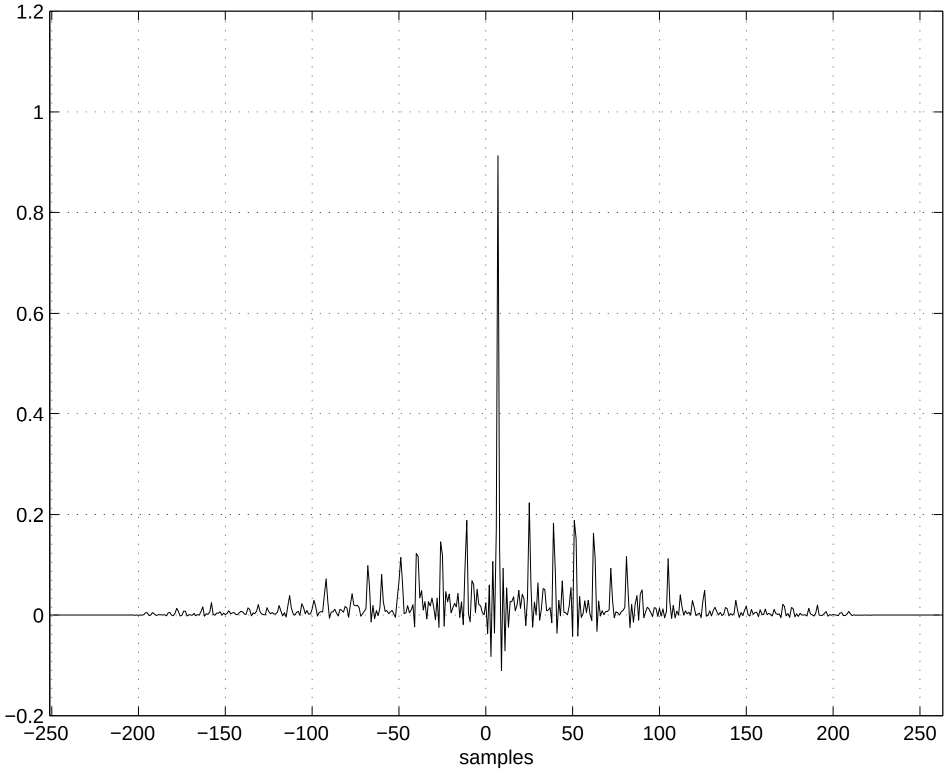


Fig. 4.6 Normalized cross correlation between the original and the delayed signals. The delay is the index for which the the cross correlation is maximum, in this case it is estimated to be 2 samples.

Chapter 5

Performance Analysis

5.1 Preamble

5.1.1 Implementing the DCT-LMS algorithm

The DCT-LMS adaptive filter, the dashed box of Fig. 5.1, is implemented in Matlab. The inputs to this filter are the reference signal $d(n)$ and the loudspeaker output $x(n)$. In addition, a constant step size and an initial weight vector should be provided to start the adaptation process. The algorithm returns a “clean” (or echo cancelled) signal that will be transmitted back through the telephone network and a matrix that keeps track of the progression of the tap coefficients.

In continuing operation, the filter weights track the DCT of the room impulse response. To test the implementation, the sentence “Cats and Dogs each hates the other” is taken as the loudspeaker output; the resulting filter weights and “clean” signal are shown in Fig. 5.2. Since the far-end talker signal is taken to be zero, the “clean” signal should be as close as possible to zero. Comparing Fig. 5.2 (a) and Fig. 5.2 (b), we note that the amplitude scale of $e(n)$ is two order of magnitude less than the amplitude scale of $x(n)$ and we observe that $e(n)$ is converging to zero. Furthermore, it is clear from the comparison of Fig. 5.2 (c) and Fig. 3.7 that the final values of the filter weights approximate very closely the DCT of the room impulse response.

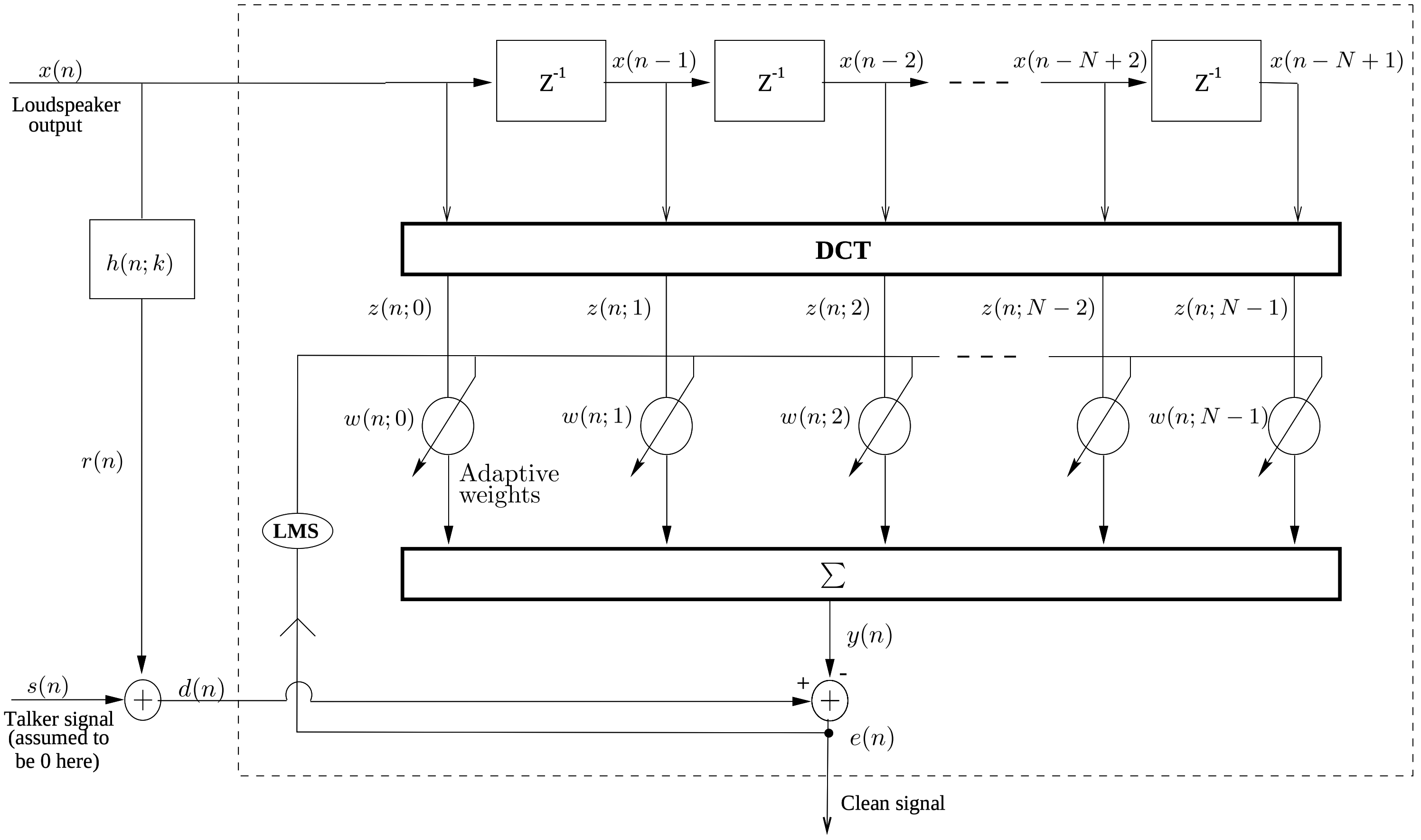


Fig. 5.1 The simulation bench used to evaluate the performance of the DCT-LMS echo canceller with Spectrum Delay Update. $s(n)$ is taken to be zero to reduce the computation and to simplify the comparison of the clean signal (or error) $e(n)$ with the ideal case ($e(n) = 0$).

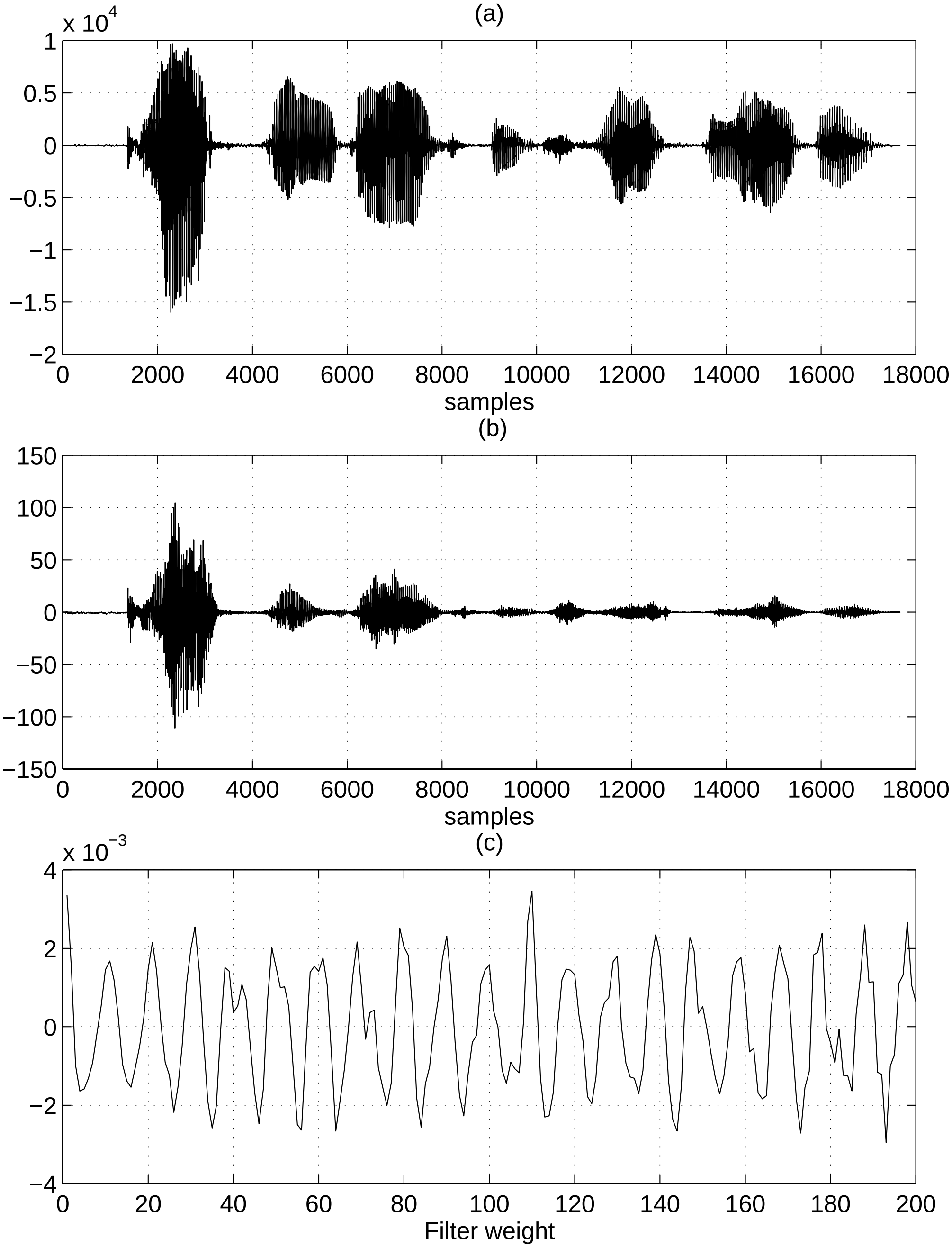


Fig. 5.2 An echo cancellation example: $x(n)$ shown in (a) is the speech sentence "Cats and Dogs each hates the other" and $s(n)$ is set to zero to simplify the calculations. The step size μ is chosen to be 0.0001 and the initial filter weights are set to zero. The resulting "clean" signal is shown in (b) and the final values of the filter weights are displayed in (c). The room used in this experiment is given by Table 2.1. **Note** the scale of the vertical axis in the three plots.

5.1.2 Performance measures

Given a room impulse response DCT vector **H** and a filter weights vector **w** we are interested in knowing "how far apart" they are. Most of the distances used in speech applications are particular cases of the Minkowski distance defined as follows: Let $\mathbf{H}_k$ denote the k^{th} component of the N-vector **H**. Then the *Minkowski distance of order s* or l_s between vectors **H** and **w** is

$$d_s(\mathbf{H}, \mathbf{w}) \overset{def}{=} \sqrt[s]{\sum_{k=1}^{N} |\mathbf{H}_k - \mathbf{w}_k|^s} \qquad (5.1)$$

The particular case that we will be using is the l_2 or the *Euclidean Distance*

$$d_2(\mathbf{H}, \mathbf{w}) = \sqrt{\sum_{k=1}^{N} |\mathbf{H}_k - \mathbf{w}_k|^2} \qquad (5.2)$$

The objective of Spectrum Delay Update (SDU) which replaces the frozen coefficients by a delayed version of the previous full rank impulse response DCT is to decrease the jump in the MSE when the gap vanishes again. Consequently, the average reduction in the MSE will reflect the improvement gained by SDU.

Another measure used to describe the performance of the echo canceller is the echo return loss (*ERL*), which estimates the amount of echo removed by the echo canceller [34], and is defined as

$$ERL = 10 \log \frac{\mathbf{x}^T \mathbf{x}}{\mathbf{e}^T \mathbf{e}} \qquad (5.3)$$

where $\mathbf{x}^T\mathbf{x}$ is the far-end loudspeaker output power and $\mathbf{e}^T\mathbf{e}$ is the power of the uncancelled echo.

5.2 Experimental set-up

Consider the following assumptions:

- Room dimensions are $5 \times 4 \times 3$ m, the receiver is located at position $(1, 1, 1)$ and the source at position $(3, 2, 1)$.

- The reference signal is created by filtering a speech segment by the Chebyshev type II bandstop filter, whose frequency response is shown in Fig. 5.3. The resulting signal will contain a gap in the DCT spectrum; the size and position of this gap will depend on the the size and position of the reject band of the filter.
- All coefficients of the 256-tap DCT-LMS filter have been initialized to zero at time $n = -\infty$.
- The far-end talker's signal is null (the microphone signal contains only echoes).

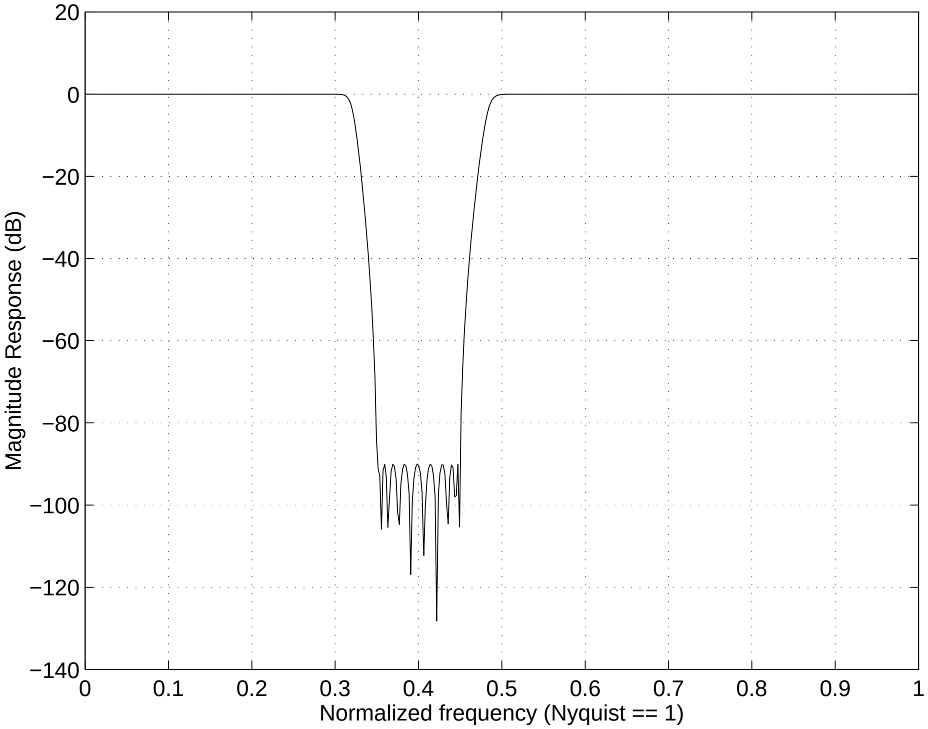


Fig. 5.3 Magnitude response of the 10^{th} order Chebyshev type II bandstop filter, with the stopband ripple 90 dB down. For this particular example, the stopband was chosen between 0.35 and 0.45 (normalized frequency).

First phase

The receiver moves to $(1, 1, 1 + \Delta x)$, where Δx ranges between 0.05 m and 0.12 m. After an initial convergence period, the tracking coefficients are used to estimate the delay in the impulse response. The frozen coefficients are then spectrally updated as shown in Fig. 5.4. The resulting tap vector is “closer” to the actual room impulse response DCT. On the other hand, if the coefficients had remained frozen to their initial value, the Euclidean Distance between the filter weights and the actual room impulse response DCT would have been bigger. Fig. 5.5 illustrates this difference for a particular reference signal. The mean difference (in dB) between the two distances will provide our first performance improvement measure, identified as *EDMD* (Euclidean Distance Mean Difference)

$$\begin{aligned}\mathit{EDMD} =& E[20\log\{d_2(\mathbf{H},\mathbf{w}_{su})\} - 20\log\{d_2(\mathbf{H},\mathbf{w})\}]\\ =& E[20\log\{\frac{d_2(\mathbf{H},\mathbf{w}_{su})}{d_2(\mathbf{H},\mathbf{w})}\}]\end{aligned} \tag{5.4}$$

Second phase

The reference signal autocorrelation matrix gains full rank, i.e, its transform no longer contains any gap (the reference signal is not filtered anymore). Since the DCT is computed every sample, it will take 256 samples before the gap vanishes entirely from the spectrum.

The previously frozen coefficients begin adapting from their value at the end of phase one, and a large jump in the MSE is expected. However, if the weight vector was spectrally updated in anticipation of the increase in the rank of **R**, the resulting MSE is smaller, as shown by Fig. 5.6 for a particular reference signal. The mean separation between the two MSE curves (in Fig. 5.6) reflects the improvement gained by SDU; we will refer to it as *MSEMS* (MSE Mean Separation).

Finally, the overall *ERL* is computed for the two cases (with and without SDU), the difference between the two *ERL* values or *ERLD*, will characterize the combined (two phases) performance improvement.

5.3 Results

5.3.1 Performance vs. gap size

Using the experimental set-up outlined in the previous section, with $\Delta x = 0.05$ m, the gap start is fixed to position 95 but the gap end is varied to yield various gap sizes (7 to 34

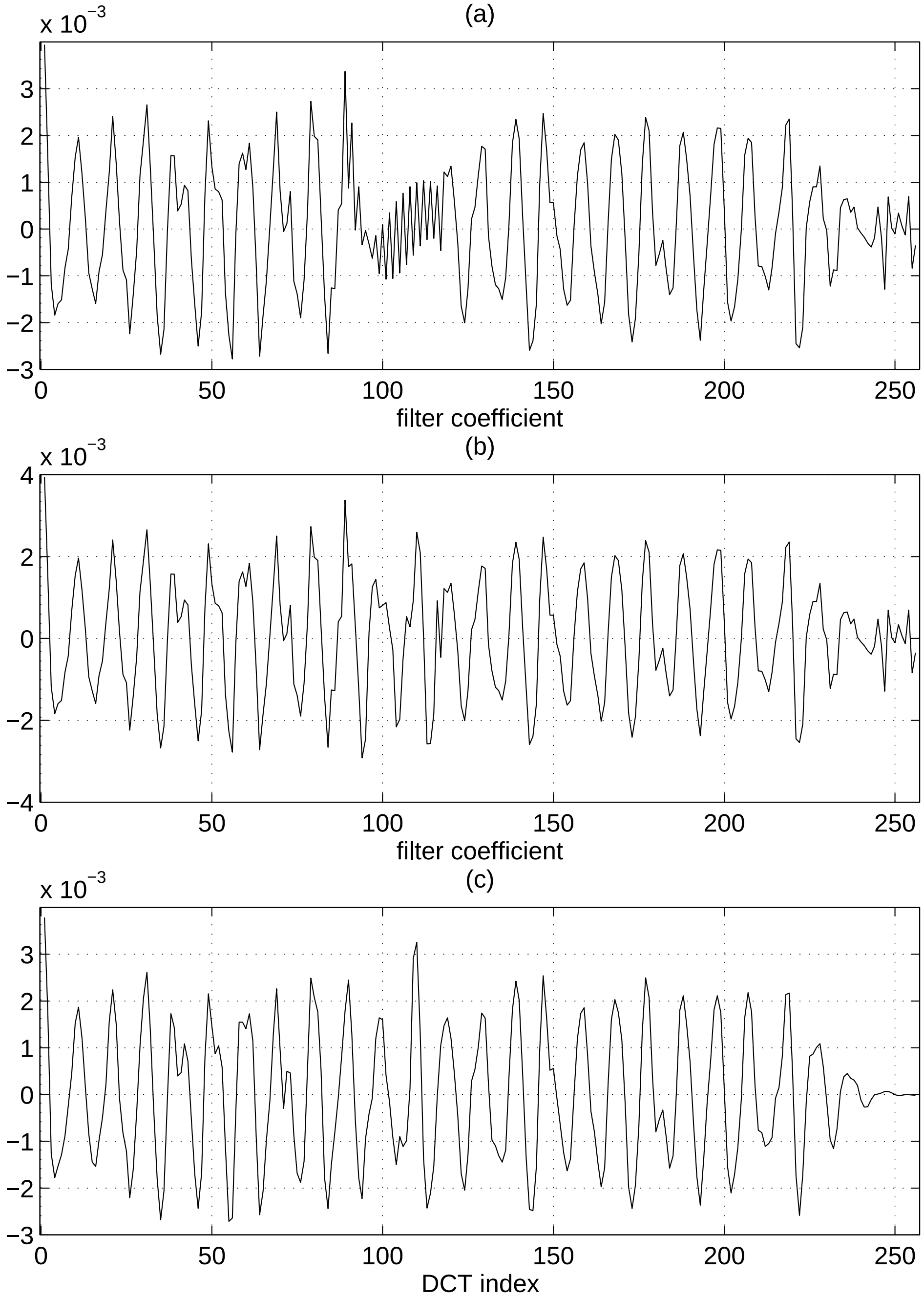


Fig. 5.4 Effect of Spectrum Delay Update on the filter coefficients. There is a gap in the DCT spectrum of the reference signal in positions 90 through 115. Equivalently, the coefficients 90 through 115 of the filter weight vector, displayed in (a), will “freeze”. The spectrally updated filter weights, shown in (b), are closer to the actual room impulse response DCT, shown in (c).

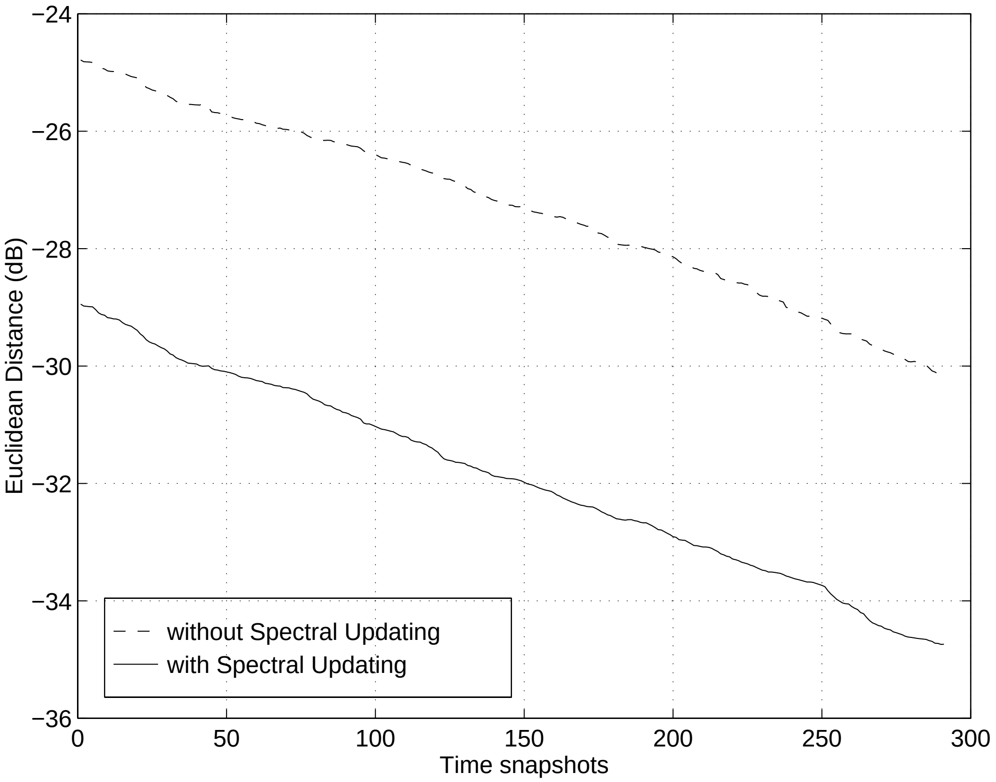


Fig. 5.5 Euclidean Distance between the filter weights and the room impulse response DCT. Every 10 samples a snapshot of the filter weights is taken. (the reference signal DCT has a gap in positions 90 through 115)

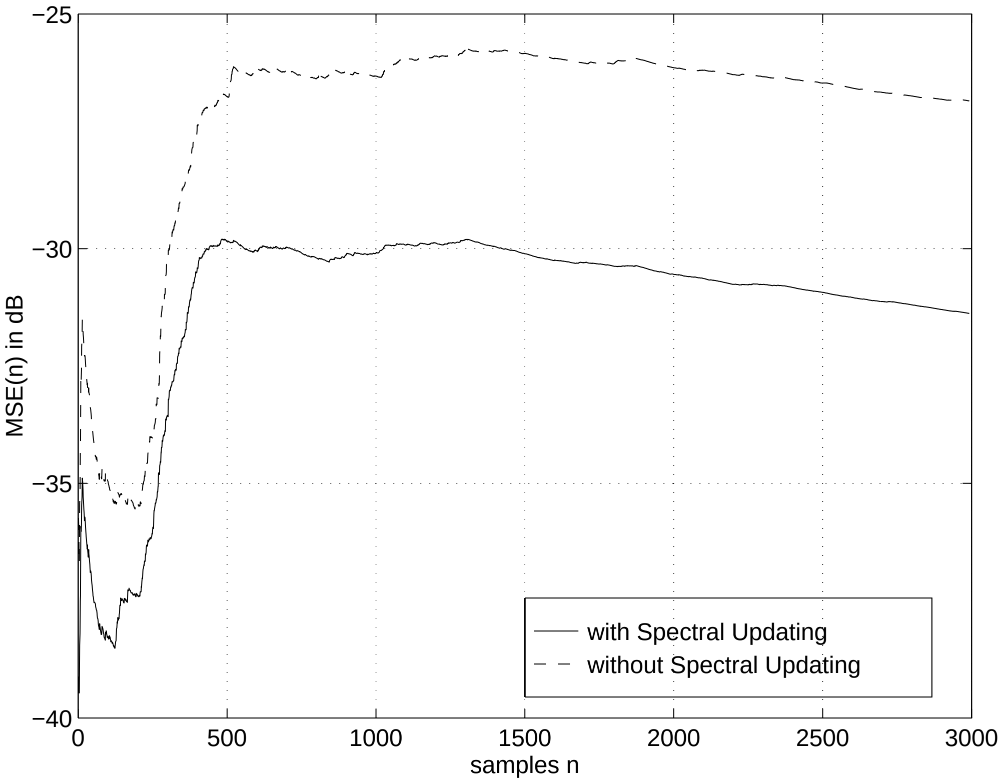


Fig. 5.6 Evolution of the MSE with time. At $n = 0$, the autocorrelation matrix, $\mathbf{R}$, of the reference signal gains full rank (but its effect is felt gradually since the DCT is taken at every sample). The dashed curve represents the MSE that results from the coefficients in the gap (positions 90 through 115) being frozen to their value at the end of phase one. The solid curve gives the MSE that results from having spectrally updated the coefficients in the gap in anticipation of the increase in rank($\mathbf{R}$).

samples). The three performance measures, *EDMD*, *MSEMS* and *ERLD* are obtained for the four acoustic environments described in section 4.4.2. The results are shown in Fig. 5.7, Fig. 5.8, and Fig. 5.9. At first glance, the improvement gained by Spectrum Delay Update is evident.

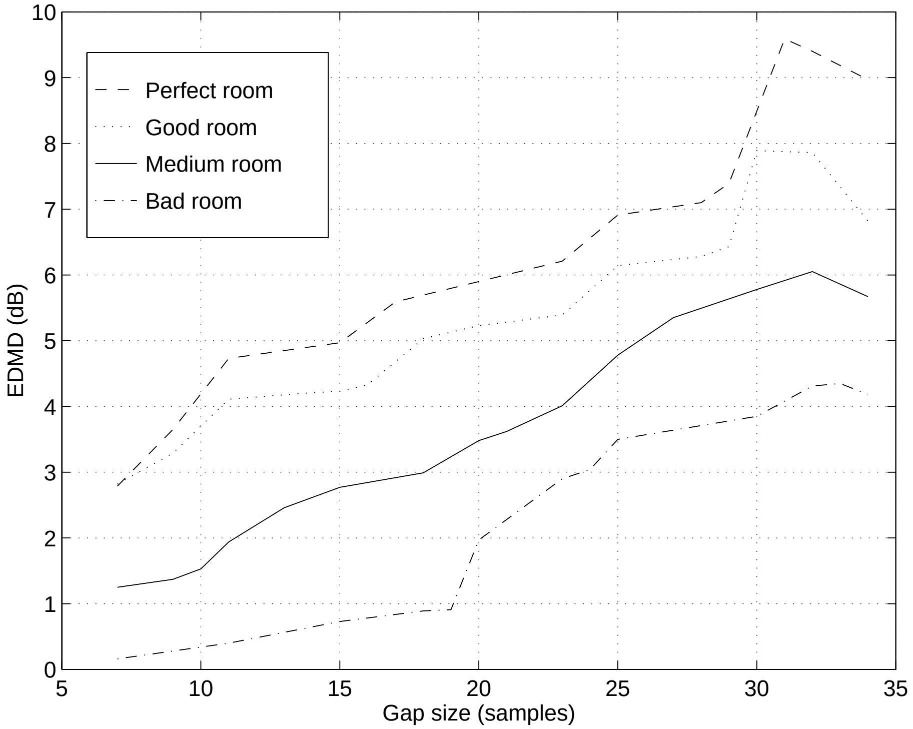


Fig. 5.7 Variations of *EDMD* with respect to the gap size in four different acoustic environments. The gap start is fixed to position 95 and $\Delta x = 0.05$ m.

It is obvious from Fig. 5.7 that SDU makes the filter weights closer to the room impulse response DCT. The acoustic environment, through its effects on the accuracy of the delay based model (representing the change in the room impulse response) and the exactness of the delay estimate, plays an important role in determining the Euclidean Distance gain. It is clear from Fig. 5.7 that the better the acoustic environment, the bigger the gain. In the perfect room, where the impulse response consists of only one lobe, the delay model reflects precisely the change in the room impulse response and the estimated delay is very accurate. Thereby, the perfect room spectrally updated weight vector is the closest (between all acoustic environments) to the actual room impulse response DCT: *EDMD*

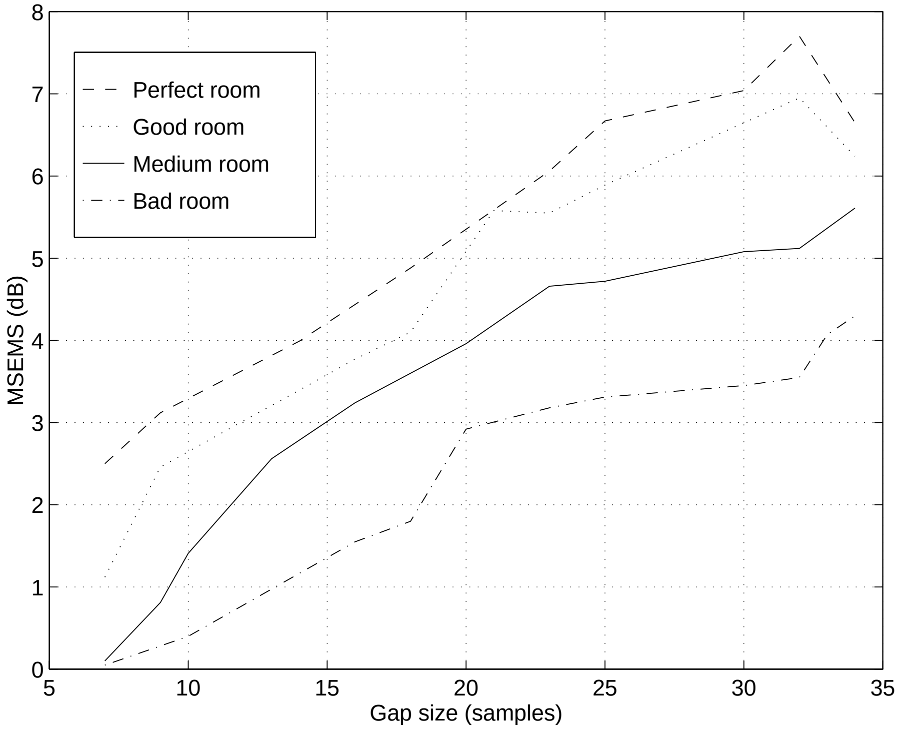


Fig. 5.8 Variations of *MSEMS* with respect to the gap size in four different acoustic environments. The gap start is fixed to position 95 and $\Delta x = 0.05$ m.

ranges between 3 dB for small gaps (7 samples) to more than 9 dB for large gaps (32 samples). On the other hand, the l_2 gain is less dramatic (nevertheless remarkable) for medium and bad rooms; the presence of multiple lobes in the impulse response reduces noticeably the *EDMD*. For all environments, *EDMD* increases with the gap size: as the gap gets bigger, more filter coefficients freeze, and more weights are spectrally updated. The medium room, which is the most realistic one, will be taken as a reference to assess the performance improvement. For this case, *EDMD* increases almost linearly from 1.25 dB for a gap of 7 samples to 6.05 dB for a 32 sample gap.

The reduction in the MSE that results from spectrally updating the filter coefficients in anticipation of the increase in the rank of $\mathbf{R}$, is shown in Fig. 5.8. It is clear that *MSEMS* increases with the quality of the acoustic environment (for the same reasons outlined in the previous paragraph) and the size of the gap. If more coefficients were frozen in phase one of the experiment (i.e. if the gap was bigger), the effects of SDU on the MSE become more

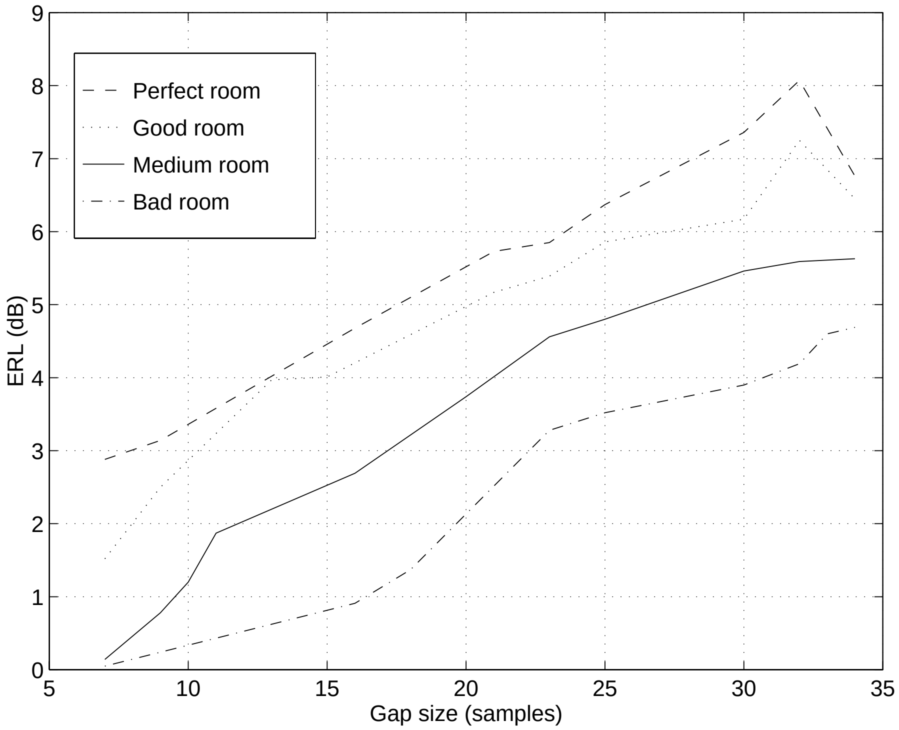


Fig. 5.9 Variations of ERL gain with respect to the gap size in four different acoustic environments. The gap start is fixed to position 95 and $\Delta x = 0.05$ m.

evident. In the medium room, the reduction in the MSE goes from a fraction of a dB in very small gaps (less than 10 samples) to more than 5 dB for gaps bigger than 30 samples.

The improvements in ERL, which measures the amount of echo removed by the echo canceller, are remarkable as shown in Fig. 5.9. Spectrum Delay Update boosts the ERL for all acoustic environments (to a different extents). The gain in ERL for a 20 samples gap for example, ranges between 2.1 dB and 5.5 dB. It is clear from Fig. 5.9 that the ERL improvement is proportional to the gap size. In the medium room, SDU increases ERL by 0.1 to 5.6 dB depending on the gap size.

5.3.2 Performance vs. receiver movement

It is interesting to determine the performance improvement for various receiver movements. For this purpose, the reference signal spectrum gap is fixed to positions 95 through 118 in the first phase of the experimental protocol outlined in section 5.2. Δx is increased gradually from 0.05 m to 0.12 m. The resulting performance measures, *EDMD*, *MSEMS* and *ERLD* are then obtained for the good, medium, and bad acoustic environments described in section 4.4.2.

It is obvious from Fig. 5.10 that for all displacements, Spectrum Delay Update makes the filter weights closer to the room impulse response DCT (*EDMD* is positive). But the gain in terms of the Euclidean Distance decreases slightly when the receiver displacement increases, especially for the bad acoustic environment.

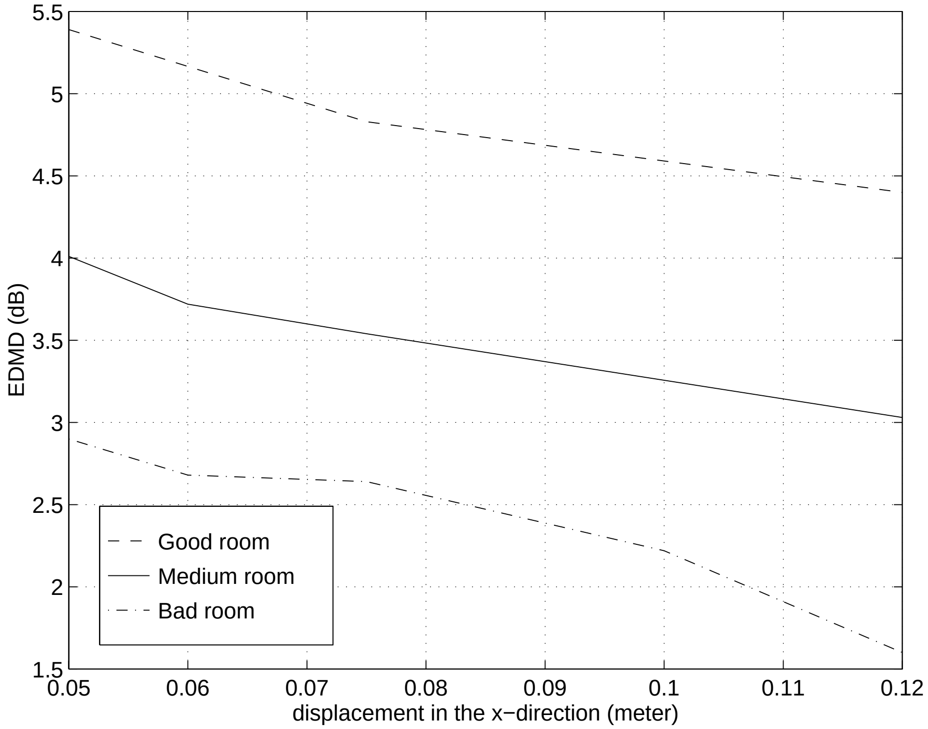


Fig. 5.10 Variations of *EDMD* with respect to the receiver movement in three different acoustic environments. The gap start is fixed to positions 95 to 118.

When the receiver movement is large, the difference between the impulse response individual sidelobe delays becomes large (some images will arrive to the receiver before others as explained in section 3.5.2) and consequently, the accuracy of modeling the change in the impulse response by a single delay is reduced. The severeness of this phenomenon depends on the quality of the acoustic environment. For the bad room, where the sidelobes are strong, *EDMD* varies from 2.9 dB for Δx=0.05 m to 1.6 dB for Δx=0.12 m. On the other hand, in the good room, *EDMD* decreases only by 0.9 dB when Δx spans the whole range.

The more prepared the filter weights are to begin adapting once the gap vanishes, i.e. the closer they are to the actual room impulse response DCT, the smaller the MSE jump is. This is demonstrated in Fig. 5.11 where *MSEMS* has the same pattern as *EDMD*. The biggest reduction in the MSE (due to SDU) is achieved when Δx is small. In the medium room, *MSEMS* decreases from 4.6 dB for Δx=0.05 m to 2.9 dB for Δx=0.12 m.

It should be also noticed that the accuracy of the delay estimate plays an important role in determining the performance gain. This is clearly illustrated by the large decrease in *MSEMS* when Δx goes from 0.1 m to 0.12 m in the bad room. The inaccurate delay estimate (as given by Table 4.4) is contributing largely to the reduction of *MSEMS*.

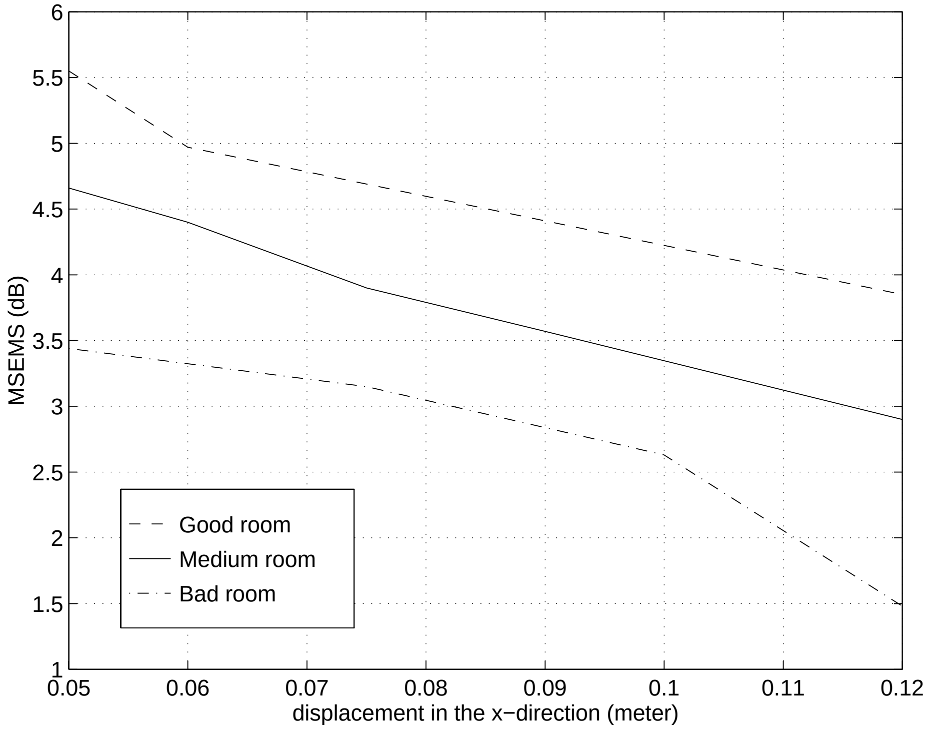


Fig. 5.11 Variations of *MSEMS* with respect to the receiver movement in three different acoustic environments. The gap start is fixed to positions 95 to 118.

Finally, the variations of the *ERLD* with respect to the receiver movement, as given by Fig. 5.12, confirm the observations already made in this section. In order to maximize the *ERL* gain, the filer weights should be updated as frequently as possible, keeping the receiver displacement small. This arises from the fact that faster update times result in shorter receiver displacement. In the medium room, *ERLD* decreases by 1 dB when Δx increases from 0.05 m to 0.12 m.

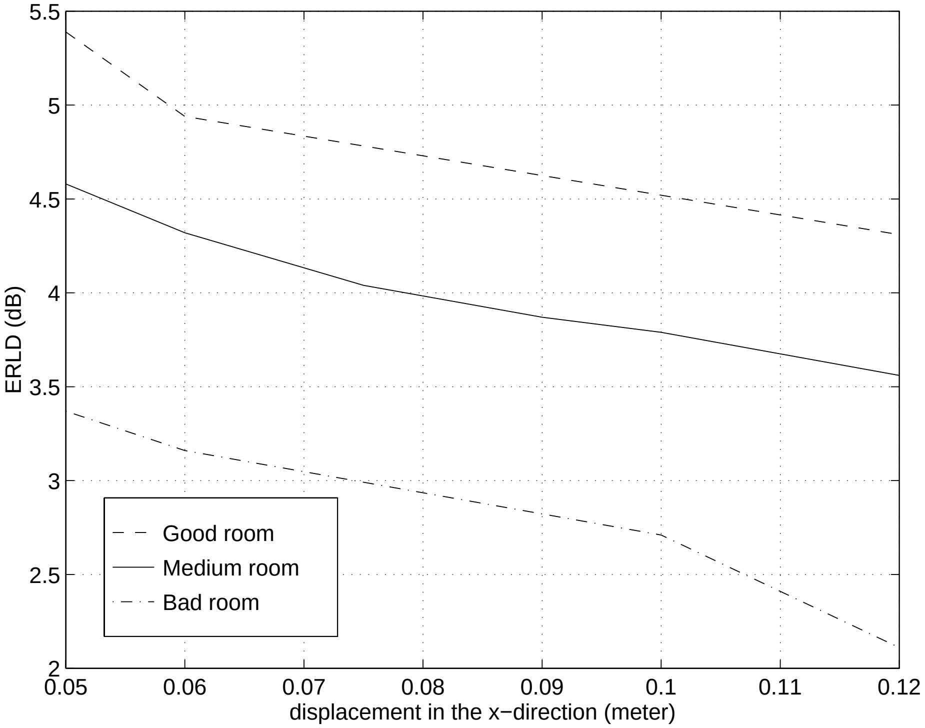


Fig. 5.12 Variations of *ERLD* with respect to the receiver movement in three different acoustic environments. The gap start is fixed to positions 95 to 118.

5.3.3 Observations

As intuitively expected, Spectrum Delay Update improves the performance of the echo canceller. The experimental results, obtained in this section, prove that the objectives of SDU were achieved. The spectrally updated weight vector is closer to the actual room impulse response DCT, and the resulting MSE is reduced. Most importantly, the Echo Return Loss is augmented.

The best improvements in performance are generally achieved when the acoustic environment is favorable, i.e. when the room impulse response does not have strong sidelobes. In this case, the delay based model, representing the change in the room impulse response, is accurate and the estimated delay is precise.

The gap (in the reference signal DCT) size plays an important role in determining the performance improvement. All performance measures introduced here indicate that the performance gain is proportional to the gap size: the bigger the gap the bigger the gain. On the other hand, the evaluation of these measures, with various receiver displacements, indicates that the faster the filter weights are updated the bigger the performance improvement is.

In all cases, Spectrum Delay Update increases the amount of echoes removed by the echo canceller, which suggests that SDU can be successfully applied to practical situations.

Chapter 6

Conclusion

More and more people are getting used to the comfort and flexibility of teleconferencing. With the rising use of satellite communications, this will become the rule rather than the exception. Good speech intelligibility is vital. However, the teleconference room is now part of the audio circuit, which creates an acoustic coupling between the loudspeaker and the microphone. An acoustic echo canceller is a key technology to overcome this problem. It identifies the impulse response between the loudspeaker and the microphone and produce an echo replica which is then subtracted from the real echo.

In this work, we chose to focus on the so-called DCT-LMS algorithm whose main adaptive component is the simple LMS filter, but whose convergence properties are closer to those of the RLS algorithm. The DCT is used to preprocess the input data in order to decorrelate them and consequently reduce the input autocorrelation matrix eigenvalue spread. The resulting structure is thus a two-layer linear adaptive filter with improved convergence properties.

The intuitive justification of the DCT-LMS algorithm brought up some issues we would like to briefly summarize here. Rank deficiency of the reference signal autocorrelation matrix, which is usually accompanied with the presence of spectrum gaps, causes the "freezing" of some filter taps (corresponding to the frequencies of the gap). The remaining coefficients, on the other hand, continue to adapt and respond to the changing room dynamics. In terms of the MSE, this means that the MSE is independent of the coordinates corresponding to the frozen coefficients and consequently, its minimum is of dimensionality bigger than one.

The full rank solution, however, is unique and may be "far" (in terms of some distance measure) from the solution obtained with frozen taps. Noting that the speed of convergence is inversely proportional to the distance, additional convergence time is required for the frozen coefficients to converge when the reference signal autocorrelation matrix gains full rank. One can decrease this convergence time (or correspondingly the MSE jump) by better preparing the frozen coefficients to begin adapting once the gap vanishes. This is done by choosing a point on the multi-dimensional error surface which is closer to the unique full rank solution. The selection of this point, which occurs during the low rank period, is based on the values of the tracking filter coefficients. More precisely, the variations of the room impulse response are first estimated by monitoring the tracking weights fluctuations; the frozen coefficients are then updated to follow these variations.

As explained in Chapter 3, the change in the room impulse due to a variation of the spacing between the microphone and the loudspeaker can be modeled by a single delay (the term delay can mean either a time advance or a time lag). This model allowed us to develop the Spectrum Delay Update method which will replace the "obsolete" frozen coefficients by a delayed version of the most recent "full rank" tap vector. The first part of SDU consists of estimating the corresponding delay. The simplest estimation procedure consists of finding the maximum of the cross-correlation between the original and the delayed room impulse responses. However, this method for estimating the delay is impractical because it is done in the time domain: knowing that the adaptive filter tracks the DCT of the room impulse response, the filter weights have to be transformed from the DCT domain to the time domain prior to every delay estimation.

In Chapter 4, a DCT-based delay estimation algorithm was developed. For every frequency, the generalized DCT shift property—derived in this chapter—gives a multitude of possible delays. The second step in the delay estimation process consists of obtaining the best "combined" delay estimate based on the knowledge of all candidate solutions in the available frequency range (some frequencies are frozen and others are not reliable). This is done by finding the "solutions' path" with minimum variance. To illustrate the effectiveness of the new delay estimation algorithm in various room environments, a specific receiver movement example was provided. The average difference between the actual and the estimated delay ranged from 0.01 sample to 0.14 sample depending on the quality of the acoustic environment.

Once the delay value is obtained, SDU uses this delay to update the frozen weights. The resulting filter weight vector is closer, in terms of Euclidean Distance, to the actual room impulse response DCT. This was demonstrated in Chapter 5, where an experimental protocol was designed to evaluate the performance of the DCT-LMS echo canceller with Spectrum Delay Update. As intuitively expected, SDU reduced the MSE jump and increased the overall Echo Return Loss. The improvement in performance was evaluated for various (reference signal spectrum) gap sizes, receiver displacements, and acoustic environments. All the performance improvement measures introduced in Chapter 5, demonstrate that, for any acoustic environment, the gain in performance is proportional to the gap size but inversely proportional to the receiver displacement. Furthermore, the performance improvement due to SDU is more evident when the acoustic environment is favorable.

Keeping in mind that the faster we update the filter weights the shorter the receiver displacement is, one way to increase the performance gain is to update the filter weights more frequently.

The new acoustic echo canceller has overcome several limitations of previous works by taking an entirely new approach. In [12], the frozen coefficients had to be contiguous and it was necessary to have a sufficient number of consecutive coefficients available on both sides of the gap. However, with Spectrum Delay Update the freezing coefficients can be sparse. In addition, the size of the gap is not constrained to a fixed number. The only concern is that large gaps will reduce the accuracy of the estimated delay (less tracking coefficients will be available in the estimation process).

Needless to say that Spectrum Delay Update method has many limitations. It was shown in Chapter 3 that a movement of the receiver delays various lobes of the impulse response differently. Hence, representing the change in the impulse response by a single delay limits the accuracy of the model. In addition, the assumption that the delay is frequency independent makes the algorithm less general. Future work could therefore focus on relaxing these constraints. More precisely, the frequency dependence of the delays can be taken into account by not restricting the *survivor path* to a straight line. Finally, the real time implementation of the DCT-LMS echo canceller with SDU will require the design of an automatic gap detector.

The simplified cases studied here have served to promote Spectrum Delay Update as an effective method to counter the negative effects of having a reference signal with variable rank autocorrelation matrix. Other growing applications of this type of echo cancellation are the cellular and mobile phones. This new method will provide means for “comfortable” hands-free telephone conversation for speaker phones.

Appendix A

DCT shift property

In this appendix the shift property for the DCT-II, presented in [32] and stating that

$$\begin{aligned} X^{+}_{DCT}(m) = {} & \cos\big(\frac{m\pi}{N}\big) X_{DCT}(m) + \sin\big(\frac{m\pi}{N}\big) X_{DST}(m) \\ & + \sqrt{\frac{2}{N}} k_m \cos\big(\frac{m\pi}{2N}\big) \{(-1)^m x(N) - x(0)\} \\ & m = 0, 1, \dots, N-1 \end{aligned} \tag{A.1}$$

is proved.

Replacing n by $n+1-1$ in the DCT of a sequence $x(n)$, we obtain

$$X_{DCT}(m) = \sqrt{\frac{2}{N}} k_m \sum_{n=0}^{N-1} x(n) \cos\left[\frac{(n+1)m\pi}{N} + \frac{m\pi}{2N} - \frac{m\pi}{N}\right] \tag{A.2}$$

Hence, the delayed DCT spectrum will be given by

$$X^{+}_{DCT}(m) = \sqrt{\frac{2}{N}} k_m \sum_{n=0}^{N-1} x(n+1) \cos\left[\frac{(n+1)m\pi}{N} + \frac{m\pi}{2N} - \frac{m\pi}{N}\right] \tag{A.3}$$

Letting $a = \frac{m\pi}{2N}$ and $i = n+1$, $X^{+}_{DCT}(m)$ becomes

$$X^{+}_{DCT}(m) = \sqrt{\frac{2}{N}} k_m \sum_{i=1}^{N} x(i) \big\{\cos\big[2a(i+\frac{1}{2})\big] \cos(2a) + \sin\big[2a(i+\frac{1}{2})\big] \sin(2a)\big\} \tag{A.4}$$

Changing the limits of the sum to 0 and $N-1$, we obtain

$$\begin{aligned} X^{+}_{DCT}(m) &= \cos(2a)X_{DCT}(m) + \sin(a)X_{DST}(m) \\ &-\sqrt{\frac{2}{N}}k_m x(0)\cos(a)\cos(2a) + \sqrt{\frac{2}{N}}k_m x(N)\cos(m\pi + a)\cos(2a) \\ &-\sqrt{\frac{2}{N}}k_m x(0)\sin(a)\sin(2a) + \sqrt{\frac{2}{N}}k_m x(N)\sin(m\pi + a)\sin(2a) \end{aligned} \tag{A.5}$$

Using the trigonometric identities

$$\begin{aligned} \cos(a)\cos(2a) =& \frac{1}{2}[\cos(a) + \cos(3a)] \\ \sin(a)\sin(2a) =& \frac{1}{2}[\cos(a) - \cos(3a)] \\ \cos(m\pi + a) = (-1)^m &\cos(a) = \sin(m\pi + a) \end{aligned} \tag{A.6}$$

The delayed DCT spectrum $X^{+}_{DCT}(m)$ will be finally given by

$$\begin{aligned} X^{+}_{DCT}(m) =& \cos\left(\frac{m\pi}{N}\right)X_{DCT}(m) + \sin\left(\frac{m\pi}{N}\right)X_{DST}(m) \\ &+ \sqrt{\frac{2}{N}}k_m \cos\left(\frac{m\pi}{2N}\right)\{(-1)^m x(N) - x(0)\} \end{aligned} \tag{A.7}$$

for all $m \in [0, N-1]$. And the proof is complete.

Appendix B

The generalized DCT delay formula

In this appendix we prove the simplified general DCT shift property given in (4.7) and revisited here

$$\begin{gathered} X_{DCT}^{k+}(m) = \cos\big(\frac{m\pi k}{N}\big) X_{DCT}(m) + \sin\big(\frac{m\pi k}{N}\big) X_{DST}(m) \\ m = 0, 1, \dots, N-1 \end{gathered} \tag{B.1}$$

Note that this property is only valid if $x(0) = x(1) = \ldots = x(k) = 0$ and
$x(N) = x(N+1) = \ldots = x(N+k-1) = 0$. The proof proceeds by induction.

Initialization: for k=1, it was shown that for $x(0) = x(N) = 0$

$$\begin{gathered} X_{DCT}^{+}(m) = \cos\big(\frac{m\pi}{N}\big) X_{DCT}(m) + \sin\big(\frac{m\pi}{N}\big) X_{DST}(m) \\ m = 0, 1, \dots, N-1 \end{gathered} \tag{B.2}$$

The next step is to show that the formula is valid for $k+1$, given it is satisfied for k. For all $m \in [0, N-1]$, we have

$$\begin{aligned} X_{DCT}^{(k+1)+}(m) &= \cos\big(\frac{m\pi}{N}\big) X_{DCT}^{k+}(m) + \sin\big(\frac{m\pi}{N}\big) X_{DST}^{k+}(m) \\ &= \cos\big(\frac{m\pi}{N}\big) \big[\cos\big(\frac{m\pi k}{N}\big) X_{DCT}(m) + \sin\big(\frac{m\pi k}{N}\big) X_{DST}(m)\big] \\ &\quad + \sin\big(\frac{m\pi}{N}\big) \big[\cos\big(\frac{m\pi k}{N}\big) X_{DST}(m) + \sin\big(\frac{m\pi k}{N}\big) X_{DCT}(m)\big] \end{aligned} \tag{B.3}$$

Using the "product to sum" trigonometric identities $X_{DCT}^{(k+1)+}(m)$ will become

$$\begin{aligned} X_{DCT}^{(k+1)+}(m) =& \frac{1}{2}X_{DCT}(m)\big[\cos(\frac{(m+1)\pi k}{N}) + \cos(\frac{(m-1)\pi k}{N})\big] \\ &+ \frac{1}{2}X_{DCT}(m)\big[\cos(\frac{(m+1)\pi k}{N}) - \cos(\frac{(m-1)\pi k}{N})\big] \\ &+ \frac{1}{2}X_{DST}(m)\big[\sin(\frac{(m+1)\pi k}{N}) - \sin(\frac{(m-1)\pi k}{N})\big] \\ &+ \frac{1}{2}X_{DST}(m)\big[\sin(\frac{(m+1)\pi k}{N}) + \sin(\frac{(m-1)\pi k}{N})\big] \end{aligned} \tag{B.4}$$

Rearranging the terms we obtain

$$X_{DCT}^{(k+1)+}(m) = \cos\big(\frac{(m+1)\pi k}{N}\big)X_{DCT}(m) + \sin\big(\frac{(m+1)\pi k}{N}\big)X_{DST}(m) \tag{B.5}$$

Which means that the formula is valid for $k+1$. And the proof is complete.

Bibliography

[1] O. Muron and J. Sikorav, “Modeling of reverberators and audioconference rooms,” in *Proc. Int. Conf. Acoust., Speech, Signal Proc.*, (Tokyo, Japan), pp. 921–924, 1986.

[2] J. Chao, S. Kawabe, and S. Tsujii, “A new IIR adaptive echo canceler,” *IEEE Journal on selc. areas in com.*, vol. 12, pp. 1530–1539, December 1994.

[3] A. Birkett and R. Goubran, “Acoustic echo cancellation using NLMS-Neural network structures,” in *Proc. Int. Conf. Acoust., Speech, Signal Proc.*, (Detroit MI), pp. 3035–038, 1995.

[4] B. Widrow and S. Stearns, *Adaptive Signal Processing.* Englewood Cliffs, NJ: Prentice-Hall, 1985.

[5] W. Kellermann, “Analysis and design of multirate systems for cancellation of acoustical echoes,” in *Proc. Int. Conf. Acoust., Speech, Signal Proc.*, (New York), pp. 2570–2573, 1988.

[6] A. Gilloire and M. Vetterli, “Adaptive filtering in sub-bands,” in *Proc. Int. Conf. Acoust., Speech, Signal Proc.*, (New York, NY), April 1988.

[7] A. Gilloire, “Experiments with sub-band acoustic echo cancellers for teleconferencing,” in *Proc. Int. Conf. Acoust., Speech, Signal Proc.*, (Dallas,Texas), pp. 2141–2144, 1987.

[8] S. Haykin, *Adaptive Filter Theory.* Englewood Cliffs, NJ: Prentice-Hall, 3rd ed., 1996.

[9] N. Ahmed and K. Rao, *Orthogonal Transforms for Digital Signal Processing.* New York: Springer-Verlag, 1975.

[10] S. Narayan, A. Peterson, and M. Narasimba, “Transform domain LMS algorithm,” *IEEE Trans. Acoust., Speech, Signal Proc.*, vol. 31, pp. 609–615, June 1983.

[11] K. Rao and P. Yip, *Discrete Cosine Transform: Algorithms, Advantages, Applications.* San Diego, Calif.: Academic Press, 1990.

[12] S. Zlobec, "Linear predictive spectral shaping for acoustucal echo cancellation," Master's thesis, McGill University, Montreal, Canada, November 1995.

[13] H. Kuttruf, *Room Acoustics.* London: Elsevier Applied Science, 3rd ed., 1991.

[14] J. Allen and D. Berkley, "Image method for efficiently simulating small-room acoustics," *J. Acoust. Soc. Am.*, vol. 65, pp. 943–950, April 1979.

[15] T. Petillon, A. Gilloire, and S. Theodoris, "The fast Newton transversal filter: An efficient scheme for acoustic echo cancellation in mobile radio," *IEEE Trans. Sig. Proc.*, vol. 42, pp. 509–518, March 1994.

[16] F. Mortessage, O. Legrand, and D. Sornette, "Transient chaos in room acoustics," *Chaos*, vol. 3, no. 4, pp. 529–541, 1993.

[17] C. Gritton and D. Lin, "Echo cancellation algorithms," *IEEE Trans. Acoust., Speech, Signal Proc.*, vol. 32, April 1984.

[18] E. Hansler, "The hands-free telephone problem — an annotated bibliography," *Signal Processing*, vol. 27, pp. 259–271, June 1992.

[19] C. Cowan and P. Grant, *Adaptive Filters.* Englewood Cliffs, NJ: Prentice-Hall, 1985.

[20] J. Deller, Jr., J. Proakis, and J. H. L. Hansen, *Discrete-Time Processing of Speech Signals.* New York: MacMillan, 1993.

[21] R. Gray, "On the asymptotic eigenvalue distribution of Toeplitz matrices," *IEEE Trans. Inf. Th.*, vol. 18, pp. 725–730, November 1972.

[22] R. Bracewell, *The Harley Transform.* New York, NY: Oxford University Press, 1986.

[23] J. Walsh, "A closed set of normal orthogonal functions," *American Journal of Mathematics*, no. 1, pp. 5–24, 1923.

[24] N. Ahmed, T. Natarajan, and K. Rao, "Discrete Cosine Transform," *IEEE Trans. Comput.*, vol. 23, pp. 90–93, January 1974.

[25] Z. Wang, "Fast algorithms for the Discrete W Transform and for the Discrete Fourier Transform," *IEEE Trans. Acoust., Speech, Signal Proc.*, vol. 32, pp. 803–816, August 1984.

[26] O. Ersoy, *Fourier-Related Transforms, Fast Algorithms and Applications.* Upper Saddle River, NJ: Prentice Hall, 1997.

[27] J. Tribolet and R. E. Crochiere, "Frequency domain coding of speech," *IEEE Trans. Acoust., Speech, Signal Proc.*, vol. 27, pp. 512–530, October 1979.

[28] K. Ngan and R. Clarke, "Lowpass filtering in the cosine transform domain," in *Intl. Conf. on Com.*, (Seattle, WA), pp. 31.7.1–31.7.5, June 1980.

[29] D. Marshall, W. Jenkins, and J. Murphy, "The use of orthogonal transforms for improving performance of adaptive filter," *IEEE Trans. on Circuits and Systems*, vol. 36, pp. 474–484, April 1989.

[30] F. Beaufays, "Transform-domain adaptive filters: An analytical approach.," *IEEE Trans. on Signal Proc.*, vol. 43, pp. 422–431, February 1995.

[31] F. Ling, "Convergence characteristics of LMS and LS adaptive algorithms for signals with rank-deficient correlation matrices," in *Proc. Int. Conf. Acoust., Speech, Signal Proc.*, (New York), pp. 1499–1502, 1988.

[32] P. Yip and K. Rao, "On the shift property of DCT's and DST's," *IEEE Trans. Acoust., Speech, Signal Proc.*, vol. 35, pp. 404–406, March 1987.

[33] Z. Wang, "Fast algorithm for the Discrete Sine Transform implemented by the Fast Cosine Transform," *IEEE Trans. Acoust., Speech, Signal Proc.*, vol. 30, pp. 814–815, October 1982.

[34] S. Park and G. Hillman, "On acoustic echo cancellation implementation with multiple cascade adaptive FIR filter chip," in *Proc. Int. Conf. Acoust., Speech, Signal Proc.*, (Glasgow, Scotland), pp. 952–955, May 1989.