# Linear Predictive Spectral Shaping for Acoustical Echo Cancellation

Sanro Zlobec

Department of Electrical Engineering
McGill University, Montreal

November 1995

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

## Abstract

The purpose of this Thesis is to study adaptive acoustical echo cancellation for signals with variable-rank covariance matrices. Solutions based on the least-mean-square (LMS) algorithm are presented, with the focus being on discrete-cosine-transform- (DCT) domain finite-impulse-response (FIR) filters.

In speech-related applications, the covariance matrix of the reference signal is often nearly singular, i.e., rank-deficient, which has the effect that some of the transform-domain tap coefficients stop adapting and effectively "freeze". During this low-rank phase, the frozen taps can retain any value without effect on the mean-square error (MSE), while the remaining taps track the evolution of the system and keep the MSE at a minimum.

When the covariance matrix becomes nonsingular, however, there are no longer any frozen coefficients, and a unique tap coefficient vector yields minimum MSE. The MSE abruptly "jumps", and convergence of the taps to the unique vector will take additional time due to the (obsolete) values of the previously frozen coefficients. To remedy the situation, one applies a method dubbed "spectral shaping".

The objective of spectral shaping is to replace, during the low-rank phase, each frozen coefficient by an estimate of the corresponding coefficient of the unique full-rank solution. This is achieved in the transform domain by a combination of forward and backward linear predictors. By using spectral shaping, the frozen coefficients are thus "prepared" to be unfrozen when the covariance matrix gains full rank, resulting in a reduced jump in the MSE.

# Sommaire

Ce mémoire a pour objet d'étudier la cancellation adaptive des échos d'un signal dont la matrice de covariance a un rang variable. Les solutions sont basées sur l'algorithme LMS, et utilisent principalement des filtres a réponse impulsionnelle finie, opérant dans le domaine de la transformée discrète du cosinus (DCT).

Souvent, dans le cas de la parole humaine, le signal de référence est doté d'une matrice de covariance de bas rang, ce qui cause certains des coefficients du filtre à "geler". Durant cette période de bas rang, les coefficients gelés ne produisent aucun effet sur l'erreur carrée moyenne (ECM), tandis que les autres coefficients continuent à suivre l'évolution du système en minimisant l'ECM.

Cependant, lorsque la matrice de covariance devient invertible, aucun des coefficients ne reste gelé, et il n'y a qu'un seul vecteur qui minimise l'ECM. L'erreur augmente brusquement, et il faut un temps supplémentaire aux coefficients pour converger vers le minimum unique. Pour remédier à ce phénomène, on utilise une méthode appellée la "formation spectrale" (spectral shaping).

Durant le temps où la matrice de covariance a un bas rang, la formation spectrale a pour but de remplacer chaque coefficient gelé par un estimé du coefficient correspondant de la solution unique. Dans le domaine du DCT, ceci est accompli avec l'aide de la théorie de la prédiction linéaire. Une fois la formation spectrale appliquée, les coefficients sont mieux "préparés" à affronter une augmentation du rang de la matrice de covariance, ne menant ainsi qu'à une légère hausse de l'ECM.

# Acknowledgments

I would like to thank Dr. P. Kabal, whose suggestions and guidance at crucial stages of this Thesis were invaluable. I am also grateful for the financial support provided by Dr. Kabal throughout my graduate studies. I would also like to thank Dr. B. Champagne for his assistance with the *s-room* program.

# Contents

# List of Figures

# List of Tables

# Chapter 1

# Introduction

The presence of acoustical echoes can severely degrade the performance of a hands-free communication system [1]–[6]. The need to improve the quality of such systems is stimulated by the increasing popularity of, for example, teleconferencing and hands-free car telephones. Although the idea of cancelling acoustical echoes is very simple, its implementation is complicated, and much research is being conducted in this area in order to ultimately offer a satisfactory solution to the problem (see references in [6]).

In the literature, echo cancellation has come to signify the removal of echoes in a telephone circuit, which are due to reflections at various points (e.g., hybrid transformer boundaries) along the line [7, 8]. Acoustical echoes, on the other hand, are due to reflections off of walls or objects inside a room, as a signal makes its way from the loudspeaker to the microphone. Many ideas applied to the cancellation of acoustical echoes do stem from the standard echo cancellation literature, yet there are certain unique considerations: the signals involved are almost exclusively baseband speech signals, and the echoes are caused by a more rapidly time-varying system.

Various adaptive filter structures have been developed to solve the acoustical echo cancellation and related problems; for example, the transversal, multistage lattice, systolic array, and recursive implementations. Among these, transversal finite-impulse-response (FIR) filters are often used, due to stability considerations, and to their versatility and ease of implementation [9, 10]. Moreover, many algorithms have also been developed to adapt these filters, including the least-mean-square (LMS), recursive least-squares, sequential regression, and least-squares lattice algorithms. The LMS algorithm is computationally the least demanding, as it requires neither matrix inversion nor the calculation of correlation matrices, and therefore is often selected to perform the adaptation of the filter coefficients [5, 10]. The remainder of this Thesis focuses on transversal, transform-domain FIR filters, being adapted by the LMS algorithm.

### Organization of the Thesis

In order to introduce the problem at hand, Chapter 2 presents the loudspeaker-microphone environment along with the associated acoustical echo cancellation setup, and explains the implementation of the adaptive filter. The concept of an error surface is introduced, and the classical LMS algorithm is presented. Also in Chapter 2, the transform-domain LMS algorithm is developed, which improves the LMS algorithm's often poor convergence rate. Chapter 3 shows how certain coefficients "freeze" when the reference signal covariance matrix is of low rank. Furthermore, it is reasoned that the frozen coefficients cause an abrupt increase in MSE, occurring when the covariance matrix gains full rank. Several "spectral shaping" solutions based on linear predictive methods are proposed, which constitute the key contribution of this work. Chapter 4 defines an experimental setup and compares the different methods of spectral shaping, whereby it is shown that spectral shaping enhances performance in all cases. Finally, Chapter 5 summarizes the work in this Thesis.

# Chapter 2

# Cancellation of Acoustical Echoes

## 2.1 The Acoustical Echo Canceller

### 2.1.1 The Loudspeaker-Microphone Environment

A typical setup of a hands-free communication system is illustrated in Figure 2.1. The human talker emits a signal $s_0(t)$ and the loudspeaker emits a "reference" signal $r_0(t)$. Both signals travel through the air and reflect off of walls, people, and objects inside the room, i.e., the signals reverberate. The effects of the room are modeled by a linear filtering operation, represented mathematically as the convolution of the signal with $h_{AB}(t;\tau)$, the room impulse response from point $A$ to point $B$ at time $t$ as a function of the delay $\tau$. Generally, people or objects inside the room are in motion, and the system must be treated as time-varying.

In a discrete-time framework, where sampling above the Nyquist rate is assumed, the situation can be modeled by the block diagram in Figure 2.2. In theory, the impulse responses $h_{re}(n;k)$, $h_{tm}(n;k)$, and $h_{rm}(n;k)$ are infinite in length. Nevertheless, they can be truncated to a finite number of coefficients; an appropriate number depends on the type of enclosure. For environments such as the interior of an automobile, the impulse response can be neglected after approximately 30 ms [11],

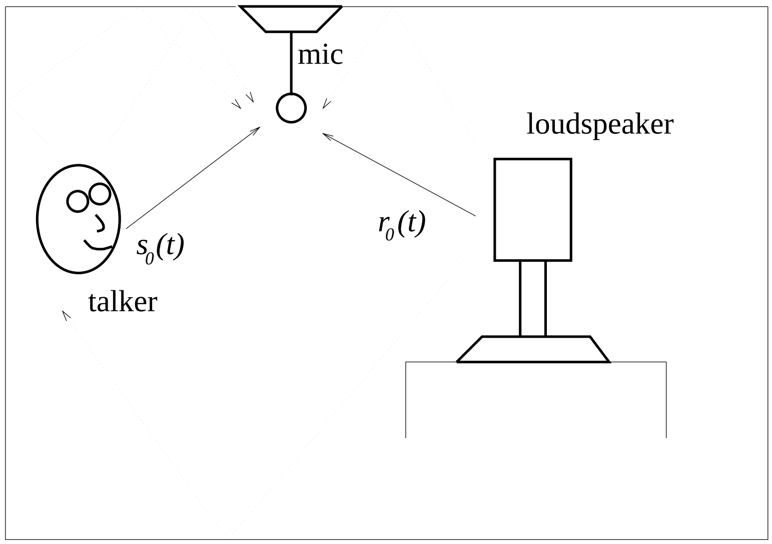


**Fig. 2.1** The loudspeaker-microphone setup: the signal received by the microphone is a mixture of the talker's signal and that originating from the loudspeaker. Each is contaminated by a number of echoes.

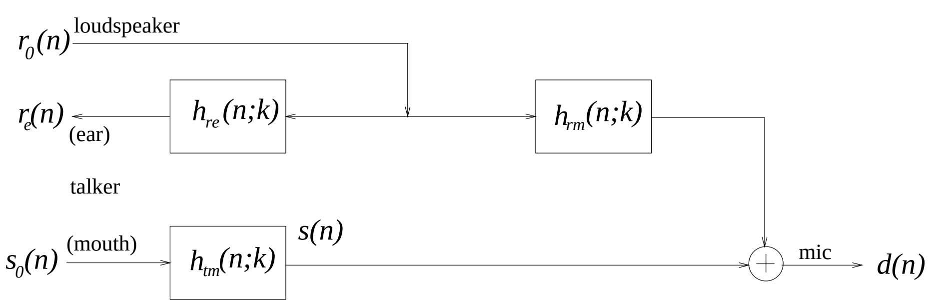


**Fig. 2.2** Block diagram of the discrete, linear, time-varying system. The impulse response $h_{rm}(n;k)$ models the distortion between loudspeaker and microphone caused by the room acoustics.

which corresponds to 240 coefficients at a sampling rate of 8 kHz. In the case of a large room, however, significant delays may occur at 200 ms or later [2], which would correspond to 1600 taps at the same sampling rate.

#### Simulating the Impulse Response

Rather than measure the actual impulse response of an enclosure, one can rely on methods that have been devised to simulate it. One such method is the well-known *s-room* algorithm [12]. The room model is a rectangular enclosure with a loudspeaker-to-microphone impulse response. The theory behind the *s-room* algorithm uses the fact that the enclosure is rectangular in order to create a mesh of “rooms”, each containing an image of the source, as illustrated in Figure 2.3. 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 room impulse response generated by *s-room* is shown in Figure 2.4 for the parameters listed in Table 2.1. At a sampling rate of 8 kHz, one sees from Figure 2.4 that the effect of delays after the 512th sample is negligible for the simulated empty, office-sized room. One would expect the presence of furniture and other sound-absorbing materials to further lower the length of the room impulse response. Nevertheless, the remainder of this Thesis assumes the empty-room scenario, in which the impulse response is truncated at the 512th sample.

### 2.1.2 The Adaptive Filter

The role of an adaptive filter is to reduce the interference due to the reverberated reference signal. One must keep in mind that although the system is time-varying, it is assumed to be varying slowly enough so as to allow for adequate tracking by the adaptive filters described in this Thesis. The distortion to $s_0(n)$ passing through $h_{tm}(n;k)$ is unavoidable and is not further considered: the “original transmitted

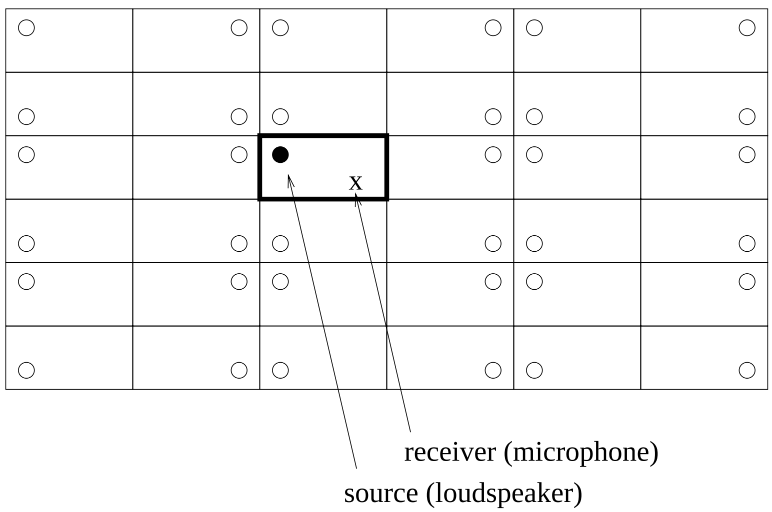


**Fig. 2.3** Illustration of the image method used for developing the *s-room* algorithm. The rectangle with the thick border represents the room along with the source ($\bullet$) and receiver ($\mathbf{x}$), while the other rectangles in the grid represent the images of the room, containing images of the source ($\circ$).

| | |
|---|---|
| Desired impulse response length (samples) | 1024 |
| Desired sampling rate (Hz) | 8000 |
| Length, Width, Height of the room (meters) | 5, 4, 3 |
| Location of the source (loudspeaker) | 1, 1, 1 |
| Location of the receiver (microphone) | 3, 2, 0.8 |
| Reflection coefficient of walls | 0.8, 0.4, 0.4, 0.4 |
| Reflection coefficient of ceiling | 0.8 |
| Reflection coefficient of floor | 0.4 |

Table 2.1 Parameters used for the generation of a typical room impulse response, shown in Figure 2.4.

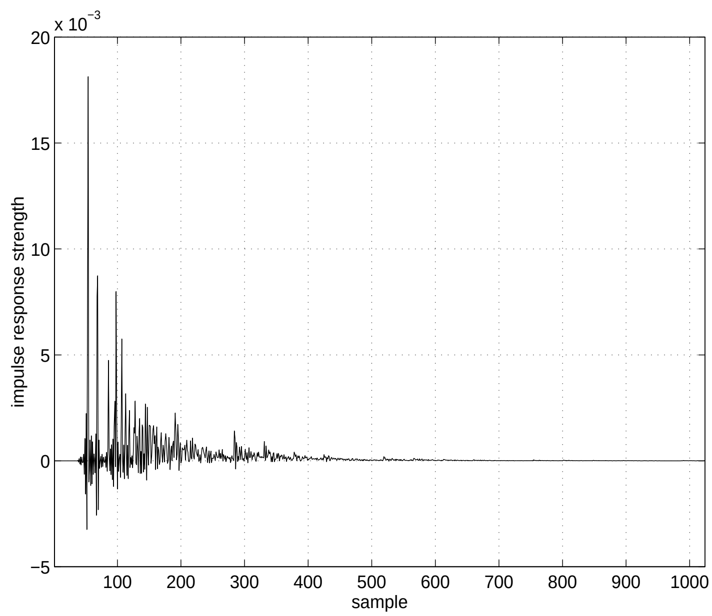


**Fig. 2.4** A typical room impulse response, as given by *s-room* using the parameters in Table 2.1. The response can be truncated at the 512th sample without consequence.

signal" is instead taken to be $s(n)$. Moreover, distortion to the reference signal $r_0(n)$ due to $h_{re}(n;k)$ does not affect the signal captured by the microphone, and is therefore ignored. By using its access to $r_0(n)$ and $d(n)$, the adaptive setup shown in Figure 2.5 recovers $e(n) = \hat{s}(n)$, and thus attempts to compensate for the echoes caused by $h_{rm}(n;k)$. For the sake of notational simplicity, $h_{rm}(n;k)$ shall from now on be referred to as $h(n;k)$.

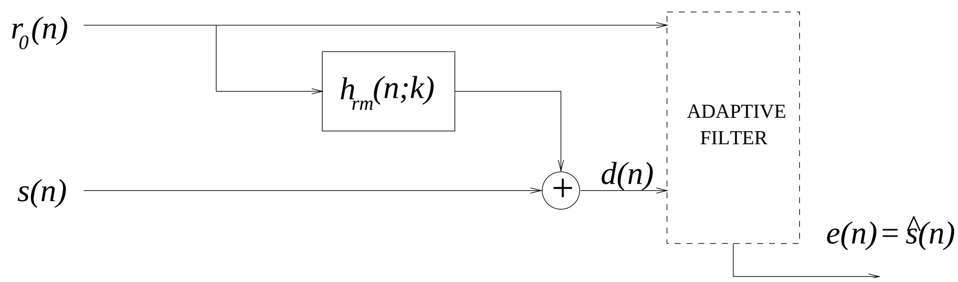


**Fig. 2.5** Setup of an adaptive filter. Both $r_0(n)$ and $d(n)$ are available, and are used to form $e(n)$, the estimate of $s(n)$. The inside of the box labelled "adaptive filter" is shown in Figure 2.6.

A transversal FIR adaptive filter, shown in Figure 2.6, performs a convolution:

$$y(n) = \sum_{i=0}^{N-1} b(n;i) r_0(n-i). \tag{2.1}$$

Letting $b(n;k) = h(n;k)$ for $k \in [0, N)$, one obtains:

$$\begin{aligned} e(n) &= d(n) - y(n) && (2.2)\\ &= \left( s(n) + \sum_{k=0}^{N-1} h(n;k) r_0(n-k) \right) - \sum_{i=0}^{N-1} b(n;i) r_0(n-i) && (2.3)\\ &= s(n). && (2.4) \end{aligned}$$

It has thus been shown that perfect cancellation is achieved if the taps are matched

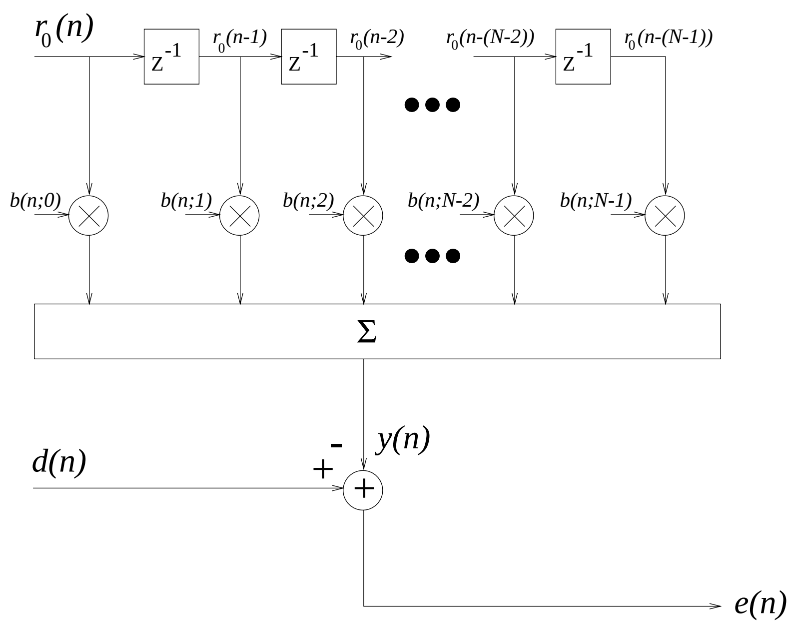


**Fig. 2.6** Transversal filtering viewed as a convolution. The condition $b(n;k) = h(n;k) \; \forall \; k \in [0, N]$ is sufficient but not necessary in order to achieve perfect cancellation.

to the impulse response coefficients:

$$\{b(n;i) = h(n;i),\ i \in [0, N)\} \ \Rightarrow\ \{e(n) = s(n)\}\,. \tag{2.5}$$

However, the reverse implication is not always true. For example, in the trivial case $r_0(n) = 0$, any coefficient vector will yield perfect cancellation. Clearly, uniqueness of the coefficient vector which yields perfect cancellation depends on the structure of $r_0(n)$. It will later be shown that an infinite number of perfect solutions exist when the rank of the $N \times N$ autocovariance matrix of $r_0(n)$ is less than $N$.

Before analyzing this phenomenon, however, a fundamental question must first be answered: how does the adaptive filter update its coefficients and actually perform the cancellation? A common approach is to use the least-mean-square (LMS) adaptation algorithm, which is introduced next.

## 2.2 The LMS and Transform-Domain LMS Algorithms

### 2.2.1 Derivation of the LMS Algortihm

The output of the transversal adaptive filter in Figure 2.6 is given by:

$$\begin{aligned}
e(n) &= d(n) - y(n) && (2.6)\\
&= d(n) - \sum_{i=0}^{N-1} b(n;i) r_0(n-i) && (2.7)\\
&= d(n) - \mathbf{b}_n^{\mathrm{T}} \mathbf{r}_n && (2.8)
\end{aligned}$$

where

$$\mathbf{r}_n = \begin{bmatrix} r_0(n) \\ r_0(n-1) \\ \cdots \\ r_0(n-N+1) \end{bmatrix} \quad \text{and} \quad \mathbf{b}_n = \begin{bmatrix} b(n;0) \\ b(n;1) \\ \cdots \\ b(n;N-1) \end{bmatrix}. \tag{2.9}$$

Squaring and taking expectations yields the mean-square error (MSE):

$$\begin{aligned}
\mathrm{MSE}(n) = \mathrm{E}[e^2(n)] &= \mathrm{E}[(d(n) - \mathbf{b}_n^{\mathrm{T}}\mathbf{r}_n)^2] && (2.10)\\
&= \mathrm{E}[d^2(n) - 2d(n)\mathbf{b}_n^{\mathrm{T}}\mathbf{r}_n + \mathbf{b}_n^{\mathrm{T}}\mathbf{r}_n\mathbf{r}_n^{\mathrm{T}}\mathbf{b}_n] && (2.11)\\
&= \mathrm{E}[d^2(n)] - 2\mathbf{b}_n^{\mathrm{T}}\mathrm{E}[d(n)\mathbf{r}_n] + \mathbf{b}_n^{\mathrm{T}}\mathrm{E}[\mathbf{r}_n\mathbf{r}_n^{\mathrm{T}}]\mathbf{b}_n. && (2.12)
\end{aligned}$$

Defining $\mathbf{p} = \mathrm{E}[d(n)\mathbf{r}_n]$ as the $N$-tuple of cross-covariances between the echo-affected signal and the reference signal, and $\mathbf{R} = \mathrm{E}[\mathbf{r}_n\mathbf{r}_n^{\mathrm{T}}]$ as the $N \times N$ covariance matrix of the reference signal[1], one obtains:

$$\mathrm{MSE}(n) = \mathrm{E}[d^2(n)] - 2\mathbf{b}_n^{\mathrm{T}}\mathbf{p} + \mathbf{b}_n^{\mathrm{T}}\mathbf{R}\mathbf{b}_n. \tag{2.13}$$

Assuming that $d(n)$ and $r_0(n)$ are wide-sense stationary random processes, both $\mathbf{p}$ and $\mathbf{R}$ are constant, and the MSE is a quadratic function of the tap coefficients. In other words, the error surface is an $N$-dimensional hyper-paraboloid, as illustrated in Figure 2.7 for $N = 2$.

### Searching the Error Surface

A consequence of the shape of the error surface is that there exists a global minimum at the bottom of the "bowl". The method of steepest descent [10, 13] seeks this minimum by adjusting the taps of the adaptive filter in the direction of the gradient of the error surface at each iteration:

$$\mathbf{b}_{n+1} = \mathbf{b}_n + \mu(-\nabla_n) \tag{2.14}$$

where $\mu$ is a constant that regulates the step size and convergence speed, and

$$\nabla_n = \frac{\partial \mathrm{MSE}(n)}{\partial \mathbf{b}_n^{\mathrm{T}}} = \begin{bmatrix} \frac{\partial \mathrm{MSE}(n)}{\partial b(n;0)} \\ \frac{\partial \mathrm{MSE}(n)}{\partial b(n;1)} \\ \cdots \\ \frac{\partial \mathrm{MSE}(n)}{\partial b(n;N-1)} \end{bmatrix}. \tag{2.15}$$

[1]Both $d(n)$ and $r_0(n)$ are assumed to be zero-mean signals.

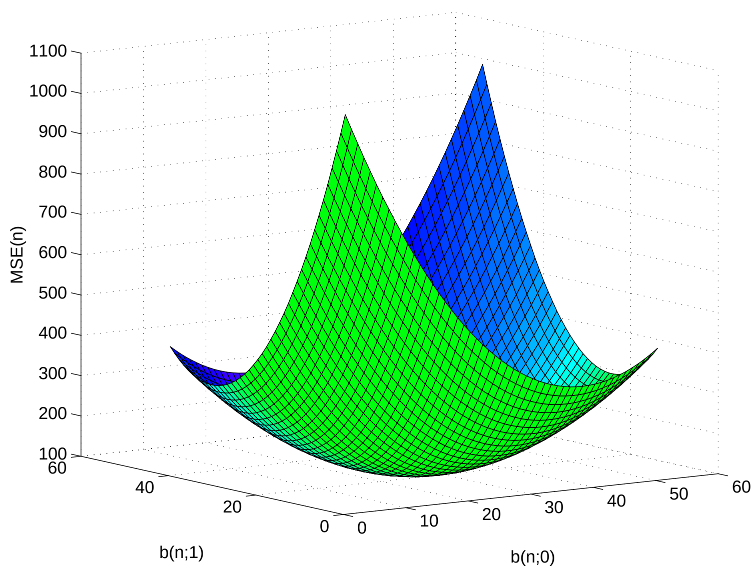


**Fig. 2.7** The error surface is an $N$-dimensional hyper-paraboloid; in this case, $N = 2$. Steepest-descent algorithms search for the bottom of the "bowl" by incrementing the tap coefficient vector in the negative direction of the gradient.

Referring to (2.13), this becomes:

$$\nabla_n = 2\mathbf{R}\mathbf{b}_n - 2\mathbf{p}. \tag{2.16}$$

To reduce the complexity of gradient estimation incurred by computation of $\mathbf{R}$ and $\mathbf{p}$, the LMS algorithm [9, 10] dispenses with expectations and considers $e^2(n)$ itself to be an estimate of the MSE. Though merely an approximation, this results in a very simple procedure:

$$\nabla_n^{\mathrm{LMS}} = \frac{\partial e^2(n)}{\partial \mathbf{b}_n^{\mathrm{T}}} \quad = \quad 2e(n)\frac{\partial e(n)}{\partial \mathbf{b}_n^{\mathrm{T}}}. \tag{2.17}$$

Now, differentiating (2.8) gives

$$\frac{\partial e(n)}{\partial \mathbf{b}_n^{\mathrm{T}}} = -\mathbf{r}_n \tag{2.18}$$

and thus

$$\nabla_n^{\mathrm{LMS}} = -2e(n)\mathbf{r}_n. \tag{2.19}$$

Using $\nabla_n^{\mathrm{LMS}}$ instead of $\nabla_n$ in (2.14) yields the LMS algorithm:

$$e(n) \quad = \quad d(n) - \mathbf{b}_n^{\mathrm{T}}\mathbf{r}_n \tag{2.20}$$

$$\mathbf{b}_{n+1} \quad = \quad \mathbf{b}_n + 2\mu e(n)\mathbf{r}_n. \tag{2.21}$$

In the case of complex signals, the analogous algorithm [9, 14] is given by

$$e(n) \quad = \quad d(n) - \mathbf{b}_n^{\mathrm{T}}\mathbf{r}_n \tag{2.22}$$

$$\mathbf{b}_{n+1} \quad = \quad \mathbf{b}_n + 2\mu e(n)\mathbf{r}_n^* \tag{2.23}$$

where "*" denotes complex conjugation.

### Convergence Characteristic

The factor $\mu$ must be large enough to permit convergence and small enough to ensure stability. In [10], it is shown that convergence of the tap coefficient vector mean is guaranteed for

$$0 < \mu < \frac{1}{\lambda_{\max}} \tag{2.24}$$

where $\lambda_{\max}$ is the largest eigenvalue of $\mathbf{R}$. A more detailed discussion of convergence properties can be found in [10]. If (2.24) is satisfied, then the convergence *rate* is governed by the disparity in the eigenvalues of $\mathbf{R}$ [9, 14, 16, 15, 17]. For example, if the reference signal consists of stationary, uncorrelated white noise with

$$\mathrm{E}[r_0(n-i)r_0(n-j)] = \begin{cases} c \in \mathrm{R}^+ & i = j \\ 0 & i \neq j \end{cases}$$

then

$$\mathbf{R} = c\mathbf{I}$$

and

$$\lambda_0 = \lambda_1 = \ldots = \lambda_{N-1},$$

and thus, with an appropriate value for $\mu$, convergence of the LMS algorithm is relatively fast. On the other hand, a reference signal whose covariance matrix is ill-conditioned will cause the LMS algorithm to converge slower. Since speech signals often have near-singular covariance matrices [6], one can therefore expect the LMS algorithm to perform poorly in speech-related applications.

### 2.2.2 The Transform-Domain LMS Algorithm

#### Derivation of the KLT LMS Algorithm

As inferred by the above, convergence can be accelerated by reducing the eigenvalue spread of $\mathbf{R}$, the reference signal covariance matrix. This would have the effect of improving how well changes in the system (i.e., $h(n;k)$) are tracked. One way to reduce the eigenvalue spread is to pass the input through a unitary transform. First, $\mathbf{R}$ is diagonalizable [13], as it is positive semi-definite[2]:

$$\mathrm{E}[\mathbf{r}_n\mathbf{r}_n^{\mathrm{T}}] = \mathbf{R} = \mathbf{P}^{\mathrm{T}}\mathbf{\Lambda}\mathbf{P} \tag{2.25}$$

[2]Since the eigenvalues and eigenvectors of a positive semi-definite matrix are purely real [18], the ordinary transpose is appropriate.

where

$$\mathbf{\Lambda} = \begin{bmatrix} \Lambda_0 & 0 & \dots & 0 \\ 0 & \Lambda_1 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \Lambda_{N-1} \end{bmatrix} \tag{2.26}$$

and $\mathbf{P}$ is the matrix of eigenvectors of $\mathbf{R}$. Continuing, one has

$$\begin{aligned} \mathrm{E}[(\mathbf{P}\mathbf{r}_n)(\mathbf{P}\mathbf{r}_n)^{\mathrm{T}}] &= \mathbf{\Lambda} && (2.27) \\ \mathrm{E}[(\mathbf{\Lambda}^{-\frac{1}{2}}\mathbf{P}\mathbf{r}_n)(\mathbf{\Lambda}^{-\frac{1}{2}}\mathbf{P}\mathbf{r}_n)^{\mathrm{T}}] &= \mathbf{I}. && (2.28) \end{aligned}$$

Hence, using $\mathbf{\Lambda}^{-\frac{1}{2}}\mathbf{P}\mathbf{r}_n$ instead of $\mathbf{r}_n$ gives a new covariance matrix $\mathbf{I}$ whose eigenvalues are unity, and there is no eigenvalue spread. The transformation $\mathbf{P}\mathbf{r}_n$ is known as the *Karhunen-Loève transform* (KLT) [13, 14] and, in combination with the normalizing factor $\mathbf{\Lambda}^{-\frac{1}{2}}$, yields a rapidly converging LMS algorithm:

$$\begin{aligned} e(n) &= d(n) - \mathbf{b}_n^{\mathrm{T}}\mathbf{\Lambda}^{-\frac{1}{2}}\mathbf{P}\mathbf{r}_n && (2.29) \\ \mathbf{b}_{n+1} &= \mathbf{b}_n + 2\mu e(n)\mathbf{\Lambda}^{-\frac{1}{2}}\mathbf{P}\mathbf{r}_n. && (2.30) \end{aligned}$$

Now, let

$$\mathbf{b}'_n = \mathbf{b}_n\mathbf{\Lambda}^{-\frac{1}{2}}. \tag{2.31}$$

The above equations thus become

$$\begin{aligned} e(n) &= d(n) - (\mathbf{b}'_n)^{\mathrm{T}}\mathbf{P}\mathbf{r}_n && (2.32) \\ \mathbf{b}'_{n+1}\mathbf{\Lambda}^{\frac{1}{2}} &= \mathbf{b}'_n\mathbf{\Lambda}^{\frac{1}{2}} + 2\mu e(n)\mathbf{\Lambda}^{-\frac{1}{2}}\mathbf{P}\mathbf{r}_n. && (2.33) \end{aligned}$$

By multiplying (2.33) by $\mathbf{\Lambda}^{-\frac{1}{2}}$ and removing the $'$ symbols, one has slightly reduced the computational requirement [14], and the KLT-domain algorithm becomes

$$\begin{aligned} e(n) &= d(n) - \mathbf{b}_n^{\mathrm{T}}\mathbf{P}\mathbf{r}_n && (2.34) \\ \mathbf{b}_{n+1} &= \mathbf{b}_n + 2\mu e(n)\mathbf{\Lambda}^{-1}\mathbf{P}\mathbf{r}_n. && (2.35) \end{aligned}$$

A more significant source of computational complexity arises from the fact that $\mathbf{P}$ is an eigendecomposition of $\mathbf{R}$. This is expensive to perform at each iteration, although several recursive methods (see, e.g., [20, 21]) have been developed. Furthermore, a practical calculation of the expectation $\mathrm{E}[\mathbf{r}_n \mathbf{r}_n^{\mathrm{T}}]$ is at best an estimate of $\mathbf{R}$, and errors in this estimation may jeopardize the accuracy of the already expensive eigendecomposition. In order to reduce the complexity of the algorithm, there exists a variety of fixed, unitary transforms that approximate the KLT for different types of signals, including speech.

### The Discrete Fourier Transform

Among the many transforms proposed for decorrelating a signal, i.e., diagonalizing its covariance matrix, one finds the discrete Fourier and discrete cosine transforms. It is shown in [25] that as $N \to \infty$, the discrete Fourier transform (DFT) asymptotically approaches an eigendecomposition when the signal in question is wide-sense stationary. In other words, for large $N$, the DFT successfully decorrelates sinusoids and other stationary signals, and is thus retained as a transform of interest. Moreover, in addition to being computable in $\mathcal{O}(N \log_2 N)$ operations using a fast Fourier transform (FFT), it offers an intuitive, frequency-domain view of the problem. The $N$-point DFT of a signal $y(n)$ is defined as

$$Y_{\mathrm{DFT},N}(k) = \sqrt{\frac{1}{N}} \sum_{i=0}^{N-1} y(i) e^{-j\frac{2\pi i k}{N}}, \quad k = 0, 1, \ldots, N-1. \tag{2.36}$$

The unitary DFT matrix operating on a column vector of length $N$ is similarly defined by

$$\mathbf{G} = \left\{ G(l,m)\ ,\ (l,m) \in [0,N) \times [0,N) \,\middle|\, G(k,n) = \sqrt{\frac{1}{N}}\, e^{-j\frac{2\pi m l}{N}} \right\}. \tag{2.37}$$

### The Discrete Cosine Transform

There are various definitions of the DCT, the one used here being based on [22]. Given a signal $y(n)$, the $N$-point DCT is computed as:

$$Y_{\mathrm{DCT},N}(k) = \sqrt{\frac{2}{N}} c(k) \sum_{i=0}^{N-1} y(i) \cos\left(\frac{\pi(2i+1)k}{2N}\right) \tag{2.38}$$

$$k = 0, 1, \ldots, N-1 \quad c(k) = \begin{cases} \frac{1}{\sqrt{2}} & k = 0 \\ 1 & k \neq 0 \end{cases}. \tag{2.39}$$

For the purpose of decorrelating a signal, it has been shown that the discrete cosine transform (DCT) comes even “closer” to emulating the KLT for speech signals than does the DFT [14]:

$$\sum_{i=0}^{N-1} |\lambda_i - Y_{\mathrm{DCT},N}(i)| \leq \sum_{i=0}^{N-1} |\lambda_i - Y_{\mathrm{DFT},N}(i)|. \tag{2.40}$$

The DCT matrix operating on a column vector of length $N$ is defined by

$$\mathbf{G} = \left\{ G(l,m)\ ,\ (l,m) \in [0,N) \times [0,N) \middle| G(l,m) = \sqrt{\frac{2}{N}} c(l) \cos\left(\frac{\pi(2m+1)l}{2N}\right) \right\} \tag{2.41}$$

where $(l,m)$ is the element in row $l$ and column $m$. In order to clarify notation to be used in the sequel, let $h(n;k)$ be a given room impulse response of length $N$ at time $n$ as a function of the delay $k$. Then the DCT of $h(n;k)$ at the same time $n$ as a function of its index $k$ will be:

$$H(n;k) = \sqrt{\frac{2}{N}} c(k) \sum_{i=0}^{N-1} h(n;i) \cos\left(\frac{\pi(2i+1)k}{2N}\right). \tag{2.42}$$

In matrix notation, this is expressed as

$$\begin{bmatrix} H(n;0) \\ H(n;1) \\ \cdots \\ H(n;N-1) \end{bmatrix} = \mathbf{H}_n = \mathbf{G}\mathbf{h}_n = \mathbf{G} \begin{bmatrix} h(n;0) \\ h(n;1) \\ \cdots \\ h(n;N-1) \end{bmatrix}. \tag{2.43}$$

It can be verified that $\mathbf{G}$ is indeed a unitary transform, i.e., that $\mathbf{G}\mathbf{G}^{\mathrm{H}} = \mathbf{I}$. In fact, a key property of the DCT is that it is real-valued, thus it is also orthogonal, i.e., $\mathbf{G}\mathbf{G}^{\mathrm{T}} = \mathbf{I}$. The real-valued property of the DCT, and its similarity to the discrete Fourier transform in terms of mathematical structure, enable its fast computation. Algorithms have been developed to compute it using a $2N$-point fast Fourier transform (FFT) [23] or, after rearranging the data, using an $N$-point FFT [22].

### Comparing Fixed Transforms to an Eigendecomposition

To compare these transforms, one revisits (2.27), and uses a fixed, possibly complex transform $\mathbf{G}$ instead of the KLT:

$$
\begin{aligned}
(\mathbf{G}\mathbf{r}_n)(\mathbf{G}\mathbf{r}_n)^{\mathrm{H}} &= \mathbf{\Lambda}' && (2.44) \\
\simeq \mathbf{\Lambda} &= \mathrm{E}[(\mathbf{P}\mathbf{r}_n)(\mathbf{P}\mathbf{r}_n)^{\mathrm{T}}] && (2.45)
\end{aligned}
$$

Figure 2.8 shows how, for different types of reference signal, the elements along the diagonal of $\mathbf{\Lambda}'$ compare to the actual eigenvalues, which are the elements along the diagonal of $\mathbf{\Lambda}$. For this figure, 128-sample blocks were taken from an uncorrelated noise source, a segment of 8 kHz sampled voised speech, a segment of 8 kHz sampled unvoiced speech, and a sinusoid at $\frac{1}{64}$ rad/s. From Figure 2.8, it can be seen that the DCT closely approximates an eigendecomposition for both noise and speech, whereas the DFT replicates such a decomposition for sinusoids.

### The Fixed-Transform LMS Algorithm

Unifying the above developments, the transform-domain LMS algorithm is given by a fixed-transform version of the KLT-domain algorithm. Thus, in (2.34) and (2.35), $\mathbf{P}$ is replaced by a fixed transform $\mathbf{G}$, and $\mathbf{\Lambda}$ is replaced by $\mathbf{\Lambda}'$ from (2.44). Furthermore, to safeguard against the possibility of one or more diagonal elements of

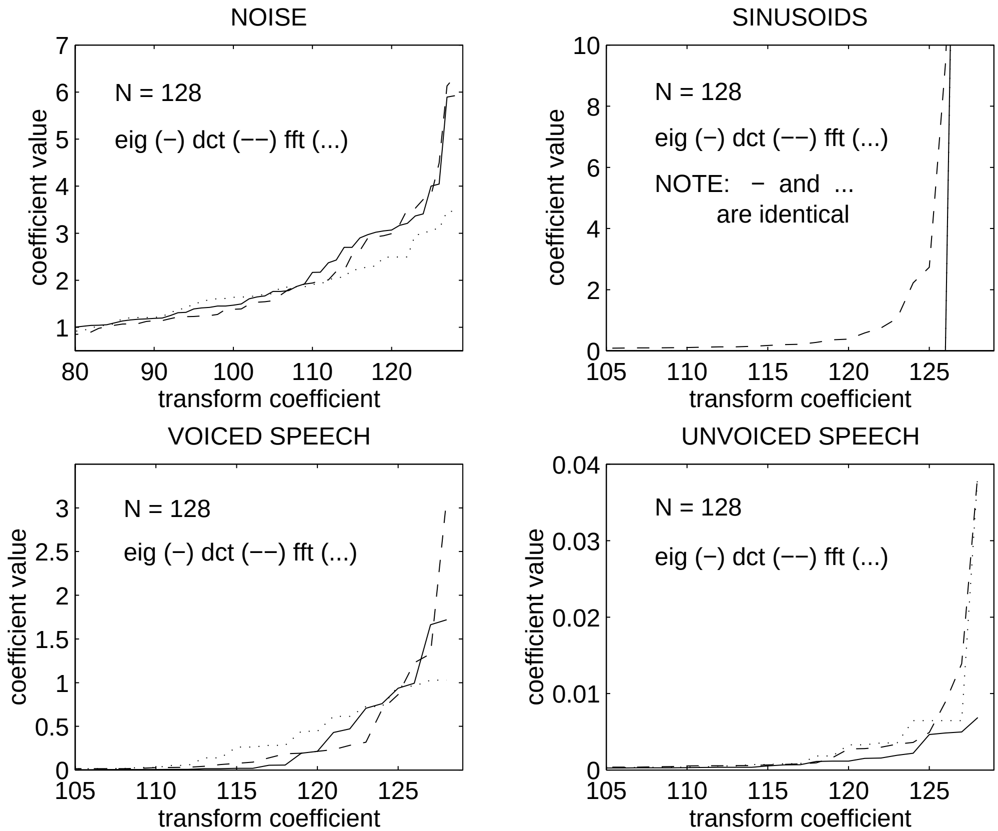


**Fig. 2.8** Comparing fixed transforms to an eigendecomposition. The values have been sorted and the axes expanded to different intervals for easier visualization. For all types of reference signal except sinusoids, the DCT is found to emulate an eigendecomposition more closely than the DFT. (Note that in the case of unvoiced speech, all but the last three DCT values are closer to the eigenvalues than are the DFT values.)

$\mathbf{\Lambda}'$ being zero, a small constant is added to its main diagonal. The transform-domain LMS algorithm is thus given by

$$\begin{aligned} e(n) &= d(n) - \mathbf{b}_n^{\mathrm{T}} \mathbf{G} \mathbf{r}_n && (2.46) \\ \mathbf{b}_{n+1} &= \mathbf{b}_n + 2\mu e(n)(\mathbf{\Lambda}' + \varepsilon \mathbf{I})^{-1} \mathbf{G} \mathbf{r}_n. && (2.47) \end{aligned}$$

The conditions under which perfect cancellation occurs can be found by rewriting (2.46):

$$\begin{aligned} e(n) &= d(n) - \mathbf{b}_n^{\mathrm{T}} \mathbf{G} \mathbf{r}_n && (2.48) \\ &= \left( s(n) + \sum_{k=0}^{N-1} h(n;k) r_0(n-k) \right) - \mathbf{b}_n^{\mathrm{T}} \mathbf{G} \mathbf{r}_n. && (2.49) \end{aligned}$$

Now, let

$$\mathbf{h}_n = \begin{bmatrix} h(n;0) \\ h(n;1) \\ \cdots \\ h(n;N-1) \end{bmatrix} \tag{2.50}$$

and replace $\mathbf{b}_n$ by $\mathbf{G}^* \mathbf{h}_n$. One then obtains:

$$\begin{aligned} e(n) &= (s(n) + \mathbf{h}^{\mathrm{T}} \mathbf{r}_n) - \mathbf{h}^{\mathrm{T}} \mathbf{G}^{\mathrm{T}^*} \mathbf{G} \mathbf{r}_n && (2.51) \\ &= s(n) + \mathbf{h}^{\mathrm{T}} (\mathbf{I} - \mathbf{G}^{\mathrm{H}} \mathbf{G}) \mathbf{r}_n && (2.52) \\ &= s(n), \text{ for } \mathbf{G} \text{ unitary.} && (2.53) \end{aligned}$$

It has thus been shown that for the transform-domain LMS algorithm,

$$\{\mathbf{b}_n = \mathbf{G}^* \mathbf{h}_n \ , \ i \in [0, N)\} \Rightarrow \{e(n) = s(n)\} \,. \tag{2.54}$$

However, the reverse implication is not always true. Consider the case where $\mathbf{G}\mathbf{r}_n$ has its $q$-th component equal to zero. Then

$$\mathbf{b}^{\mathrm{T}}\mathbf{G}\mathbf{r}_n = \left[\begin{array}{ccccc} b(n;0) & \cdots & b(n;q) & \cdots & b(n;N-1) \end{array}\right] \left[\begin{array}{l} \mathbf{g}_0\mathbf{r}_n \\ \cdots \\ \mathbf{g}_{q-1}\mathbf{r}_n \\ 0 \\ \mathbf{g}_{q+1}\mathbf{r}_n \\ \cdots \\ \mathbf{g}_{N-1}\mathbf{r}_n \end{array}\right] \tag{2.55}$$

where $\mathbf{g}_i$ is the $i$-th row of $\mathbf{G}$. One sees from (2.55) and (2.48) that $b(n;q)$ will have no effect on $e(n)$, and thus can be arbitrary. Furthermore, one sees from (2.47) that the adaptation of $b(n;q)$ depends on the value of $\mathbf{g}_q\mathbf{r}_n$, which in this case is zero. Hence, $b(n;q)$ "freezes".

In a more general sense, many coefficients of $\mathbf{G}\mathbf{r}_n$ may be null, forming a "gap" in the transform spectrum of the reference signal. In return, the corresponding components of $\mathbf{b}_n$ will be frozen and will not affect the error $e(n)$. This observation, combined with (2.54), implies that perfect cancellation and minimum MSE occur when

$$b(n;k) \longrightarrow \mathbf{g}_k^*\mathbf{h}_n, \qquad \mathbf{g}_k\mathbf{r}_n \neq 0 \tag{2.56}$$

$$b(n;k) \;\; \text{arbitrary}, \qquad \mathbf{g}_k\mathbf{r}_n = 0. \tag{2.57}$$

The effects of gaps in the reference signal transform spectrum on the mean-square error of the adaptive filter are studied more closely in the following Chapter.

# Chapter 3

# Spectral Shaping

## 3.1 Preamble: Revisiting the Error Surface

### 3.1.1 The Effect of Gaps on the Error Surface

At the end of the previous chapter, it was shown that if there is a "gap" in $\mathbf{g}_k\mathbf{r}_n$ for $k \in [k_1, k_2]$, then the transform-domain coefficients $b(n;k)$, $k \in [k_1, k_2]$ can retain arbitrary values and still satisfy the condition of minimum MSE. In terms of the error surface, this means that its minimum is not unique, and in fact is of dimensionality $M = k_2 - k_1 + 1$. For $N = 2$ and $M = 1$, the concept is illustrated in Figure 3.1. The minimum of the corresponding time-domain error surface is not unique, either, and as is now explained, the connection between the two domains is somewhat deeper.

Let there be a gap of size $M$ in the reference signal transform spectrum. The minimum of the transform-domain error surface is thus a subspace of dimension $M$. Since the transform itself is unitary (and hence nonsingular), the minimum of the time-domain error surface must also be an $M$-dimensional subspace. The effect of the transform $\mathbf{G}$ is therefore not to change the dimensionality of this minimum, but

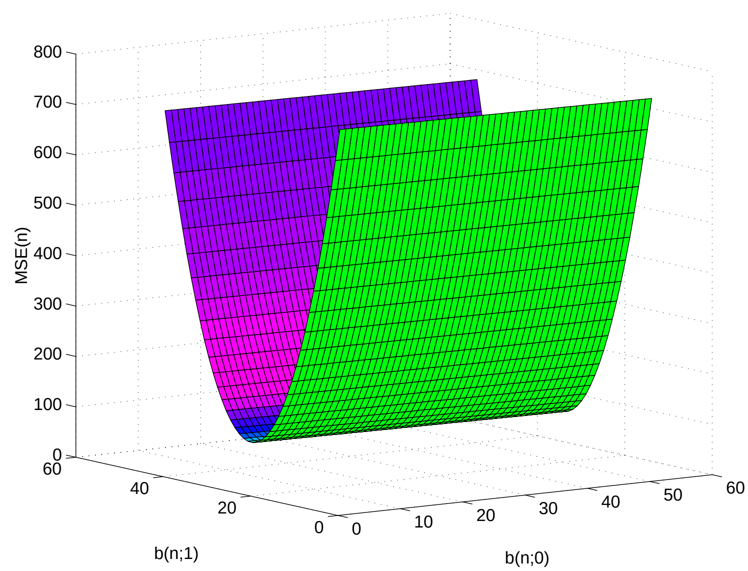


**Fig. 3.1** Non-uniqueness of minimum of the transform-domain error surface ($N = 2$ and $M = 1$, where $N$ is the number of tap coefficients and $M$ is the dimensionality of the error surface). In this simple example, it is clear that the gap is comprised of the single coefficient $b(n;0)$, since it does not affect the MSE, regardless of its value.

rather to *rotate*[1] it so that it lies orthogonal to a set of axes in the transform domain, as shown in Figure 3.2.

Recalling that the transform matrix $\mathbf{G}$ tends to approximate an eigendecomposition, one can proceed one step further and ask what happens when a true eigendecomposition is actually performed. In fact, in addition to aligning the $M$-dimensional minimum with $M$ orthogonal axes, the entire error surface is reshaped so that previously ellipsoidal contours of equal MSE become spherical (Figure 3.3). The latter property of an eigendecomposition, commonly studied alongside the LMS algorithm [9, 10], graphically distinguishes it from a fixed transform.

### 3.1.2 The Effect of Spectral Shaping

The presence of a gap in the reference signal transform spectrum will cause the tap coefficient vector $\mathbf{b}_n$ to remain fixed in the $M$-dimensional subspace described in §3.1.1, while adaptation continues in the remaining $N - M$ dimensions.

When the gap vanishes, $M$ becomes unity, and the minimum of the transform-domain error surface collapses to the single point $\mathbf{H}_n^* = \mathbf{G}^*\mathbf{h}_n$. Due to the convexity of the error surface, the speed of convergence to $\mathbf{H}_n^*$ from a given point is inversely proportional to its distance from $\mathbf{H}_n^*$. Since $\mathbf{b}_n$ has been frozen to values which may be obsolete, the distance to $\mathbf{H}_n^*$ may be large.

It is thus desirable to select the one point $\tilde{\mathbf{b}}_n$ on this $M$-dimensional hyper-surface which, based on the coefficients in the surrounding dimensions, is estimated to be closest to $\mathbf{H}_n^*$. This reasoning justifies the minimization of (3.23) and lies behind the method of spectral shaping. Figure 3.4 pictorially summarizes this concept, as seen from the point of view of the error surface.

[1] Unitary matrices preserve length and angle, and thus can be visualized as rotations [18].

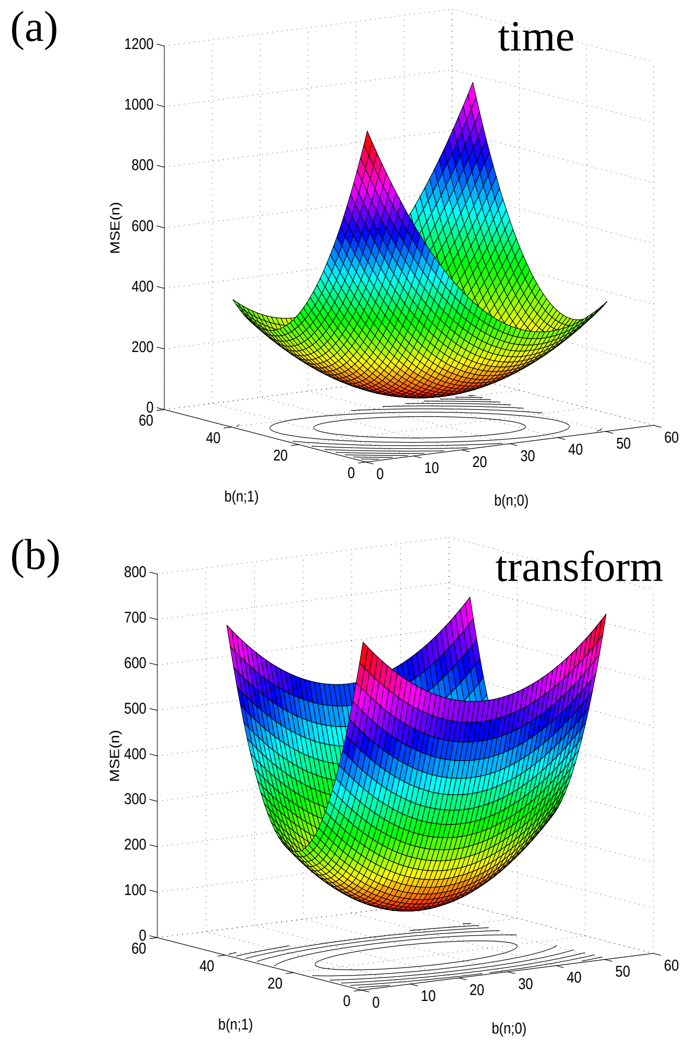


**Fig. 3.2** Rotation of a subspace by the unitary transform $\mathbf{G}$: (a) the time-domain error surface and (b) the transform-domain error surface. Note how the axes of the elliptical contours of equal MSE become aligned with the axes in the transform domain. It is understood that the coefficients referred to as $b(n;k)$ are to be interpreted in the corresponding domain (time or transform).

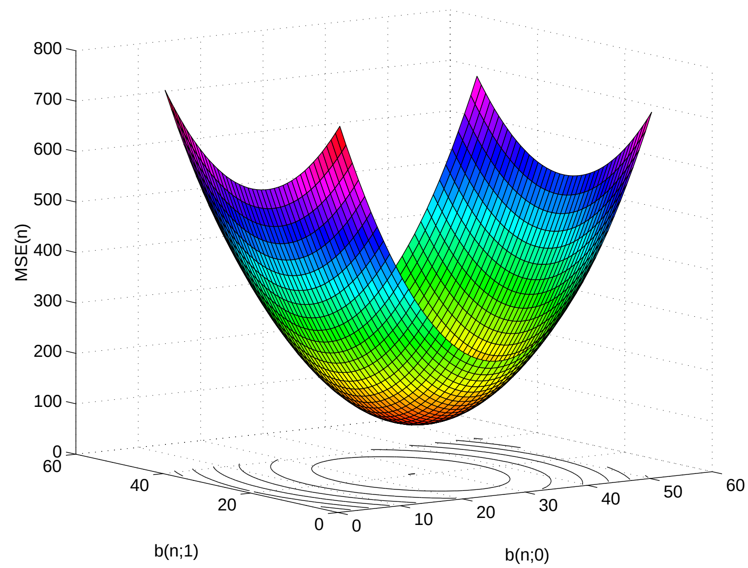


**Fig. 3.3** Effect of eigendecomposition on contours of equal MSE. The key observation is that the contours become circular instead of elliptical.

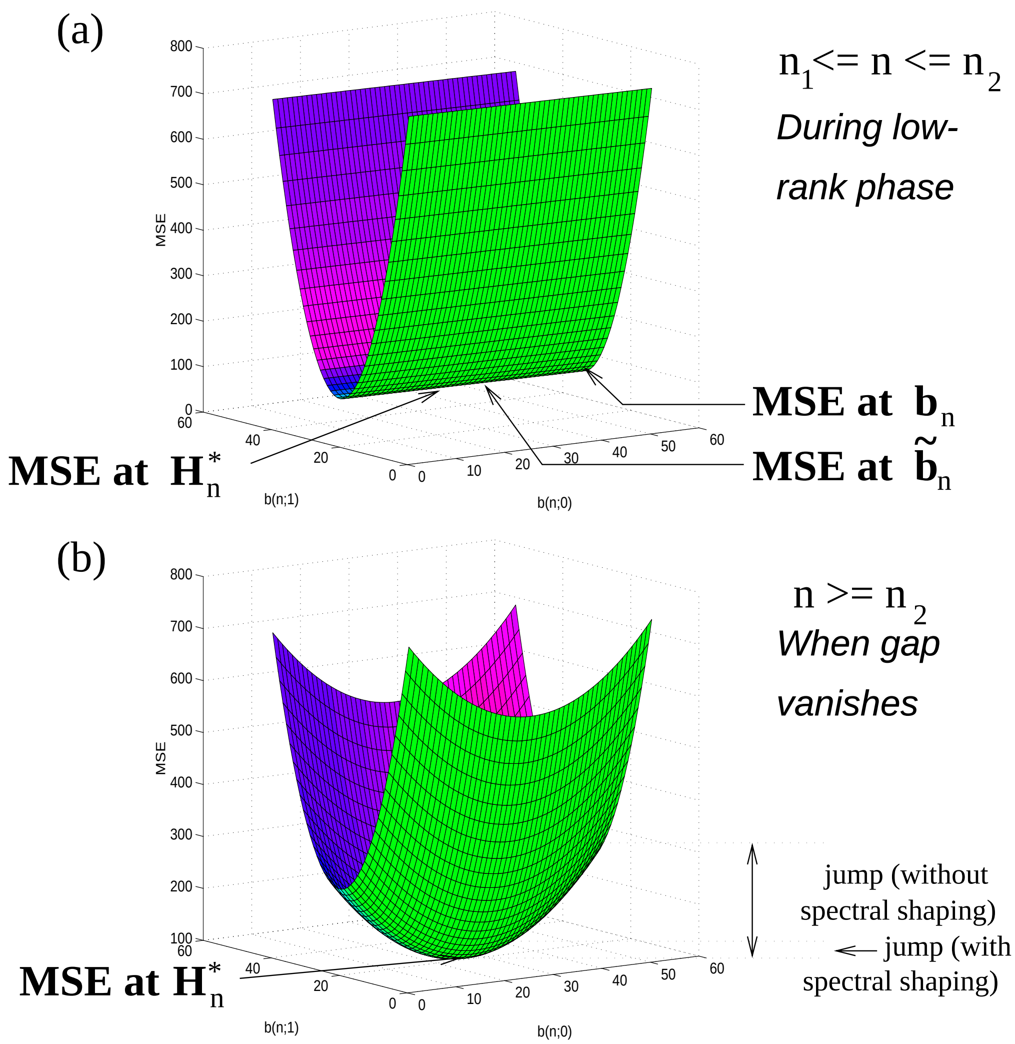


**Fig. 3.4** **(a)** Spectral shaping is applied when the reference signal covariance matrix has low rank, i.e., when the minimum of the error surface is not unique. Its purpose is to choose $\tilde{\mathbf{b}}_n$ such that it lies closer to $\mathbf{H}_n^*$ than does $\mathbf{b}_n$. **(b)** Consequently, when the gap vanishes and the minimum becomes unique, the associated jump in the MSE is greatly reduced, and the steepest-descent algorithm converges after a shorter search, i.e., after fewer iterations.

## 3.2 Motivation

To illustrate the need for spectral shaping, let $z_n(k)$ be the transform of the reference signal at time $n$, i.e.,

$$\begin{bmatrix} z(n;0) \\ z(n;1) \\ \cdots \\ z(n;N-1) \end{bmatrix} = \mathbf{z}_n = \mathbf{G}\mathbf{r}_n = \begin{bmatrix} \mathbf{g}_0\mathbf{r}_n \\ \mathbf{g}_1\mathbf{r}_n \\ \cdots \\ \mathbf{g}_{N-1}\mathbf{r}_n \end{bmatrix} \tag{3.1}$$

where $\mathbf{G}$ is unitary. Suppose that $|z_n(k)| = 0$ for $k \in [k_1, k_2]$ and $n_1 \leq n < n_2$. Then there is a gap of size $M = k_2 - k_1 + 1$ in the transform spectrum of the reference signal between coefficients $k_1$ and $k_2$ inclusively, as shown in Figure 3.5. One therefore has, for $n_1 \leq n < n_2$:

$$\mathbf{z}_n = \begin{bmatrix} z(n;0) \\ \cdots \\ z(n;k_1-1) \\ 0 \\ \cdots \\ 0 \\ z(n;k_2+1) \\ \cdots \\ z(n;N-1) \end{bmatrix}. \tag{3.2}$$

Recalling (2.47), one observes that $b(n;k)$ continues to be updated correctly for $k \in [0, k_1) \cup (k_2, N)$, but the zeros in positions $k_1$ through $k_2$ leave the corresponding coefficients "frozen"; adaptation of the coefficients outside the gap keeps the MSE at a minimum. As a generalization of (2.56) and (2.57),

$$b(n;k) \longrightarrow H^*(n;k) \qquad k \in [0, k_1)\ ,\ n_1 \leq n < n_2 \tag{3.3}$$

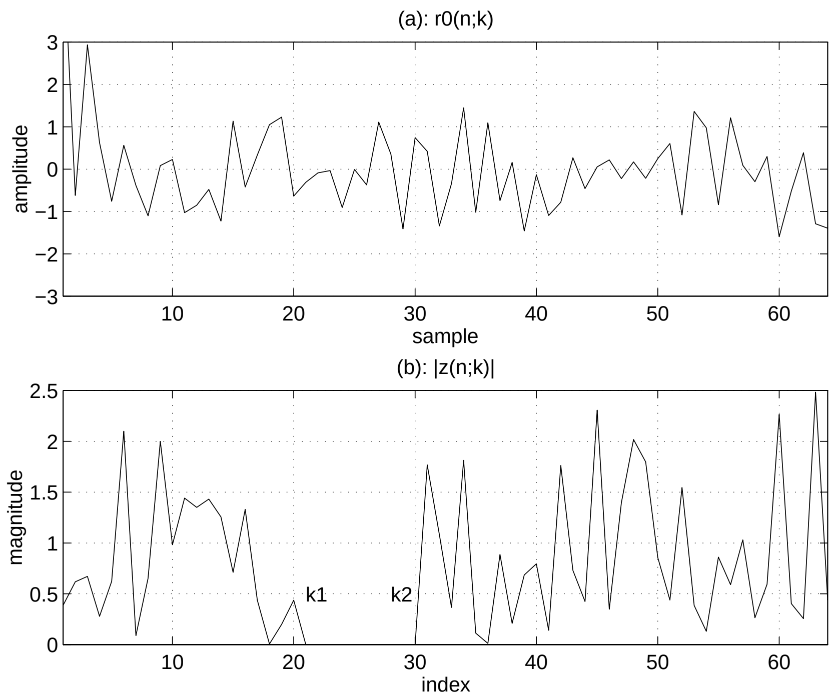


**Fig. 3.5** Illustration of the concept of a gap in the reference signal transform spectrum. The $N$-point frame of the reference signal $r_0(n)$ [in (a)] is transformed, which gives $z(n;k)$ [in (b)].

$$b(n;k) = b(n_1;k) \qquad k \in [k_1, k_2]\ ,\ n_1 \leq n < n_2 \tag{3.4}$$

$$b(n;k) \longrightarrow H^*(n;k) \qquad k \in (k_2, N)\ ,\ n_1 \leq n < n_2 \tag{3.5}$$

where $H^*(n;k) = \mathbf{g}_k^* \mathbf{h}_n$ and $\mathbf{g}_k$ is the $k$-th row of $\mathbf{G}$. The situation is illustrated in Figure 3.6.

### Vanishing Gap and the Purpose of Spectral Shaping

Now, let $z_n(k)$ be the transform of the reference signal for $n \geq n_2$ and suppose that $|z_n(k)| \neq 0\ \forall\ k$. The gap in the reference signal spectrum has vanished, and $b(n;k)$ will converge to $H^*(n;k)\ \forall\ k$:

$$b(n;k) \longrightarrow H^*(n;k) \qquad k \in [0, k_1)\ ,\ n \geq n_2 \tag{3.6}$$

$$b(n;k) \longrightarrow H^*(n;k) \qquad k \in [k_1, k_2]\ ,\ n \geq n_2 \tag{3.7}$$

$$b(n;k) \longrightarrow H^*(n;k) \qquad k \in (k_2, N)\ ,\ n \geq n_2. \tag{3.8}$$

From (3.3) through (3.8), it is clear that although the coefficients outside the gap continue to successfully track $H^*(n;k)$, additional convergence time is required for the frozen coefficients to converge to $H^*(n;k)$ from $b(n_1;k)$. The purpose of *spectral shaping* is to reduce this convergence time by accomplishing the following (while $n_1 \leq n < n_2$):

- Estimate $H^*(n;k)$ inside the gap, based on $b(n;k) = H^*(n;k)$ outside the gap, and denote these coefficients by $b_{\text{ss}}(n;k)\ ,\ k \in [k_1, k_2]$;

- Since the coefficients inside the gap do not affect the MSE, replace $b(n_1;k)$ with $b_{\text{ss}}(n;k)$ for $k \in [k_1, k_2]$.

In this way, the coefficients are better "prepared" to begin adapting once the gap in $\mathbf{z}_n$ vanishes.

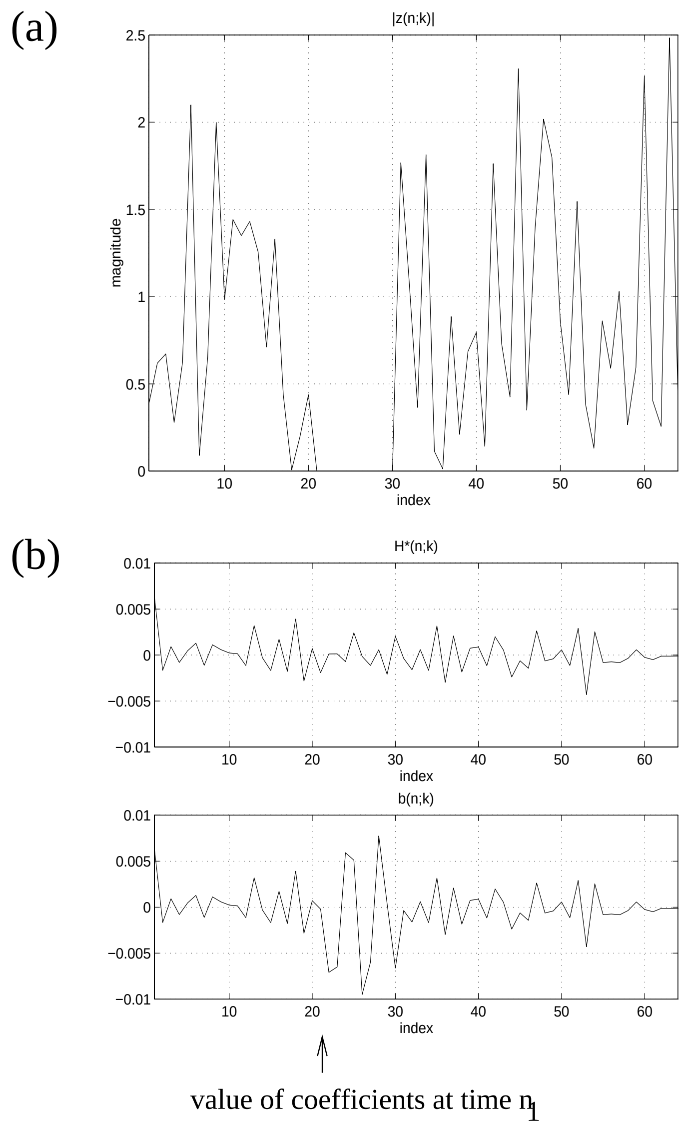


**Fig. 3.6** **(a)** A gap in $\mathbf{z}_n$ exists between $k_1 = 21$ and $k_2 = 30$. **(b)** After convergence to minimum MSE, $b(n;k) = H^*(n;k)$ for $k \notin [21, 30]$, while the value of $b(n;k)$ within the gap will be frozen to its previous value, $b(n_1;k)$, without affecting the MSE.

### Intuitive Meaning of a Gap

It is interesting to know what types of reference signals have gaps in their transform spectra. Recalling the definition of $\mathbf{z}_n$ from (3.1), one notes that gaps occur when $|z(n;k)| = 0$ for some $k$. Furthermore, recalling from (2.44) and (2.45) that each transform attempts to perform an eigendecomposition, i.e.,

$$\begin{aligned} \mathbf{z}_n\mathbf{z}_n^{\mathrm{H}} = (\mathbf{G}\mathbf{r}_n)(\mathbf{G}\mathbf{r}_n)^{\mathrm{H}} &= \mathbf{\Lambda}' && (3.9)\\ &\simeq \mathbf{\Lambda} = \mathrm{E}[(\mathbf{P}\mathbf{r}_n)(\mathbf{P}\mathbf{r}_n)^{\mathrm{T}}], && (3.10)\end{aligned}$$

one sees that the zero components of $\mathbf{z}_n$ are the zeros along the diagonal of $\mathbf{\Lambda}'$, and are closely related to the null eigenvalues[2] of $\mathbf{R}$. Insofar as $\mathbf{G}$ approximates an eigendecomposition, rank-deficiency of the reference signal covariance matrix is accompanied by the presence of gaps in the reference signal transform spectrum.

If $\mathbf{G}$ is chosen to be the discrete Fourier transform matrix, then gaps are perceived as nulls in the frequency spectrum of $r_0(n)$. Assuming that $\mathbf{G}$ approximates an eigendecomposition, one can say that changes in the rank of $\mathbf{R}$ relate to changes in the bandwidth of $r_0(n)$. Convergence problems due to vanishing gaps are therefore linked to the concept of $r_0(n)$ increasing in bandwidth. As was discussed in §2.2.2, the DCT often outperforms the DFT in terms of decorrelating the reference speech signal. Since the remainder of this Thesis deals exclusively with the DCT, it would be desirable to obtain an intuitive description of the gaps in a given signal's DCT spectrum.

In [24], it is shown that the DCT has a spectral envelope identical to that of the DFT and a modulating term, which adds a rapidly varying component to its spectrum. More precisely,

$$Y_{\mathrm{DCT},N}(k) = c(k)\,|Y_{\mathrm{DFT},2N}|\cos(\theta_{\mathrm{DFT},2N}(k) - \frac{\pi k}{2N}) \qquad (3.11)$$

[2]Signals whose covariance matrices have zero eigenvalues (and for which the LMS algorithm is known to perform poorly) are termed *rank-deficient* [26].

where $Y_{\mathrm{DCT},N}(k)$ is the $N$-point DCT of $y(n)$, $Y_{\mathrm{DFT},2N}$ is the $2N$-point DFT of $y(n)$ padded with $N$ zeroes, and $\theta_{\mathrm{DFT},2N}(k)$ is the phase of the complex quantity $Y_{\mathrm{DFT},2N}$. The vertical bars denote the operation of taking the magnitude of the given complex quantity.

Hence if $Y_{\mathrm{DCT},N}(k)$ vanishes, so does $Y_{\mathrm{DFT},2N}$, andone can immediately see the close link between the two spectra. In conclusion, therefore, gaps in the DCT spectrum of a signal can be thought of as corresponding to certain zero-energy regions of the frequency spectrum.

## 3.3 DCT Spectral Shaping

### 3.3.1 Introduction

Let $H(n;k)$ be the DCT of $h(n;k)$, i.e.,

$$\begin{bmatrix} H(n;0) \\ H(n;1) \\ \cdots \\ H(n;N-1) \end{bmatrix} = \mathbf{H}_n = \mathbf{G}\mathbf{h}_n = \mathbf{G} \begin{bmatrix} h(n;0) \\ h(n;1) \\ \cdots \\ h(n;N-1) \end{bmatrix} \tag{3.12}$$

where $\mathbf{G}$ is defined as in (2.41). The DCT of a typical room impulse response is illustrated in Figure 3.7. Now, let

$$|z(n;k)| = 0, \ \ k \in [k_1, k_2]. \tag{3.13}$$

Moreover, suppose that $n \geq n_1$ and

$$b(n;k) = \begin{cases} H(n;k) & k \in [0, k_1) \\ b(n_1;k) & k \in [k_1, k_2] \\ H(n;k) & k \in (k_2, N) \end{cases} . \tag{3.14}$$

In other words, there is a gap in the reference signal DCT spectrum, and the tap coefficients have converged to $H(n;k) = H^*(n;k)$ outside the gap, but have remained frozen within the gap[3].

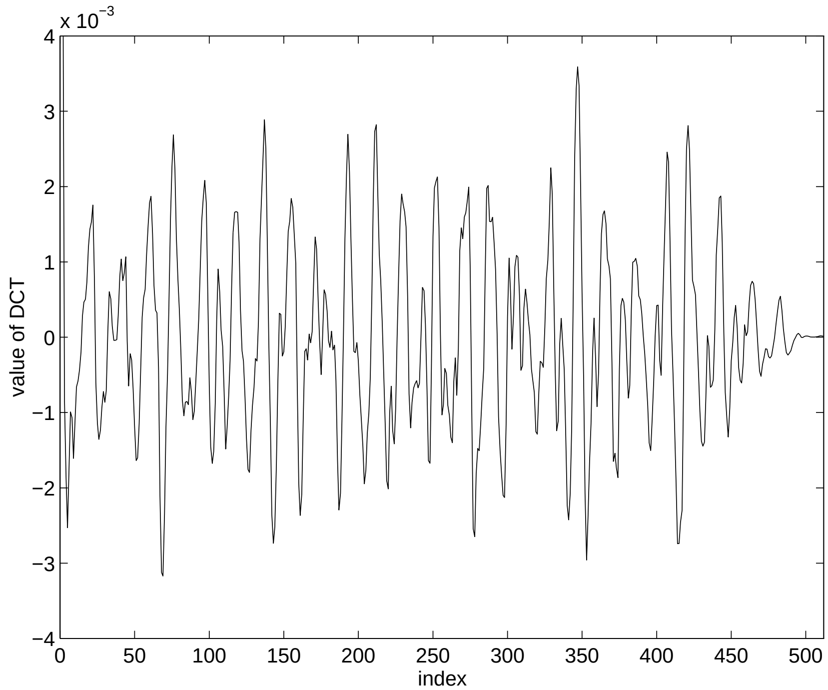


**Fig. 3.7** The DCT of a typical room impulse response; $N = 512$. From (2.42), the magnitude and frequency of the dominant cosinusoid are due to the magnitude of the first nonzero value of the impulse response, and the delay at which it occurs.

The effect of spectral shaping on $b(n;k), k \in [k_1, k_2]$ is illustrated in Figure 3.8, where $\tilde{b}(n;k)$ is defined as

$$\tilde{b}(n;k) = \mathcal{S}[b(n;k)] = \begin{cases} b(n;k) & k \in [0, k_1) \\ b_{\mathrm{ss}}(n;k) & k \in [k_1, k_2] \\ b(n;k) & k \in (k_2, N) \end{cases} \quad (3.15)$$

[3]Since the DCT is a real-valued transform, $H^*(n;k) = H(n;k)$, and thus $H(n;k)$ is used throughout the remainder of this section.

where $\mathcal{S}$ is a particular method of spectral shaping, e.g., linear predictive spectral shaping. The transform-domain LMS algorithm directly uses $\tilde{b}(n;k)$ as the new set of tap coefficients:

$$\begin{aligned} e(n) &= d(n) - \mathbf{b}_n^{\mathrm{T}}\mathbf{G}\mathbf{r}_n && (3.16)\\ \mathbf{b}_{n+1} &= \mathbf{b}_n + 2\mu e(n)(\mathbf{\Lambda}' + \varepsilon\mathbf{I})^{-1}\mathbf{G}\mathbf{r}_n && (3.17)\\ \mathbf{b}_{n+1} &= \tilde{\mathbf{b}}_{n+1} = \mathcal{S}[\mathbf{b}_{n+1}] && (3.18) \end{aligned}$$

### 3.3.2 Linear Predictive Spectral Shaping

Consider applying linear predictive theory [13] in order to find $\tilde{b}(n;k)$. To introduce the required notation, let $P_f$ and $P_b$ represent the forward and backward orders of the predictor, respectively:

$$b_{\mathrm{ss}}(n;k) = \sum_{i=1}^{P_f} c_{ik} b(n;k_1 - i) + \sum_{i=1}^{P_b} d_{ik} b(n;k_2 + i) \tag{3.19}$$

Since $b(n;k) = H(n;k)$ for $k$ outside the gap, this is also equal to

$$b_{\mathrm{ss}}(n;k) = \sum_{i=1}^{P_f} c_{ik} H(n;k_1 - i) + \sum_{i=1}^{P_b} d_{ik} H(n;k_2 + i). \tag{3.20}$$

Defining

$$\mathbf{c}_k = \begin{bmatrix} c_{1k} \\ c_{2k} \\ \cdots \\ c_{P_f k} \end{bmatrix} \quad \mathbf{d}_k = \begin{bmatrix} d_{1k} \\ d_{2k} \\ \cdots \\ d_{P_b k} \end{bmatrix} \quad \mathbf{H}_f = \begin{bmatrix} H(n;k_1-1) \\ H(n;k_1-2) \\ \cdots \\ H(n;k_1-P_f) \end{bmatrix} \quad \mathbf{H}_b = \begin{bmatrix} H(n;k_2+1) \\ H(n;k_2+2) \\ \cdots \\ H(n;k_2+P_b) \end{bmatrix} \tag{3.21}$$

one obtains

$$b_{\mathrm{ss}}(n;k) = \mathbf{c}_k^{\mathrm{T}}\mathbf{H}_f + \mathbf{d}_k^{\mathrm{T}}\mathbf{H}_b. \tag{3.22}$$

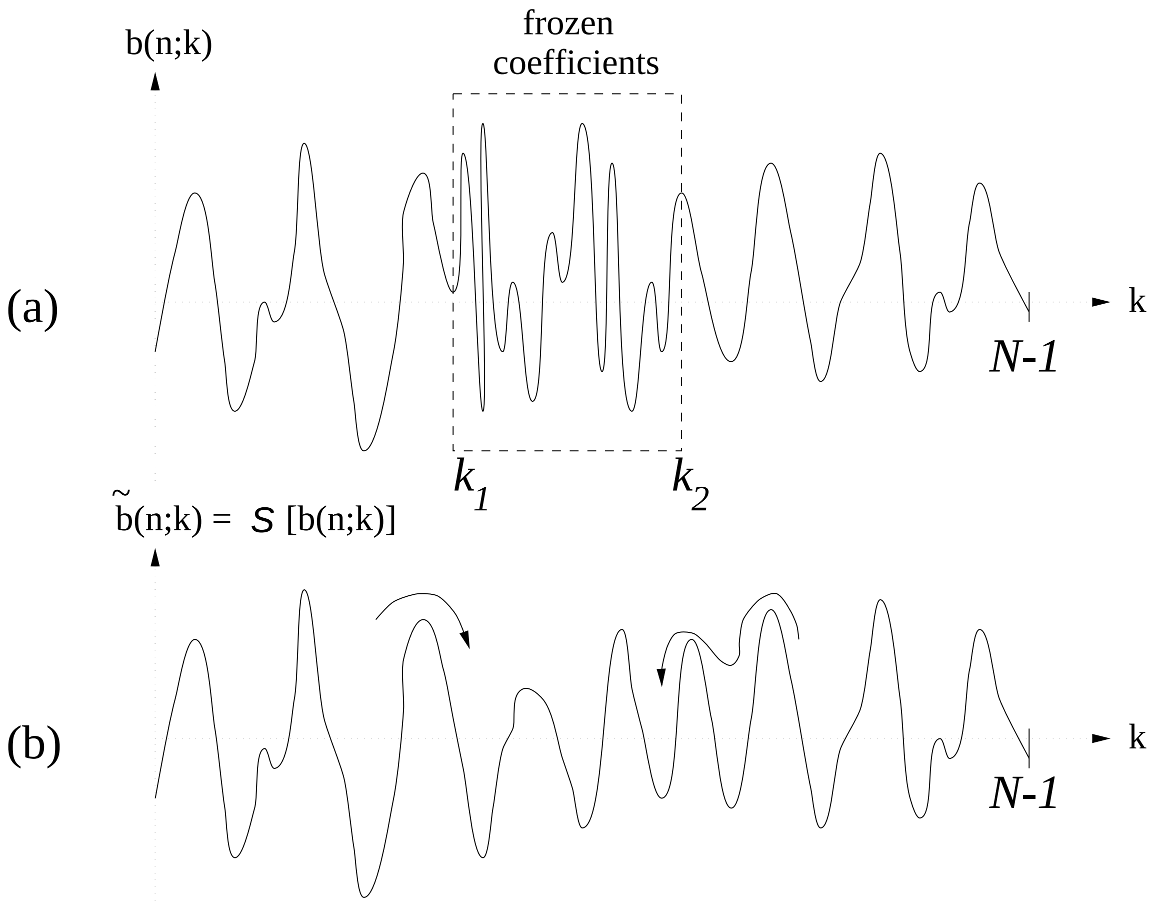


**Fig. 3.8** The effect of spectral shaping. **(a)** As explained thus far in this Chapter, the coefficients outside the gap continuously track $H(n;k)$ and those inside the gap will be frozen to values which become obsolete with time. The MSE will not be affected by the actual values of the frozen coefficients until $\mathbf{R}$ increases in rank, in which case the coefficients inside the gap must converge to $H(n;k)$. **(b)** Spectral shaping uses information contained in the neighboring "correct" coefficients to produce an estimate of $H(n;k) = H^*(n;k)$ inside the gap. Hence, large increases in the MSE due to an increase in the rank of $\mathbf{R}$ are averted.

The mean-square estimation error, defined as the mean-square difference between $H(n;k)$ and $b_{ss}(n;k)$ for $k \in [k_1, k_2]$, will thus equal[4]

$$\mathrm{MSE}_{\mathrm{est}}(k) = \mathrm{E}[e^2_{\mathrm{est}}(n;k)] = \mathrm{E}[(H(n;k) - b_{ss}(n;k))^2]. \tag{3.23}$$

As mentioned in §3.1.2, the idea behind linear predictive spectral shaping is to select $\mathbf{c}_k$ and $\mathbf{d}_k$ for each $k \in [k_1, k_2]$ so as to minimize $\mathrm{MSE}_{\mathrm{est}}(k)$.

**Forward LP**

In *forward* linear prediction, $P_b = 0$ and thus from (3.22) and (3.23):

$$\begin{aligned} b_{ss}(n;k) &= \mathbf{c}_k^{\mathrm{T}} \mathbf{H}_f & (3.24) \\ \mathrm{MSE}_{\mathrm{est}}(k) &= \mathrm{E}[(H(n;k) - \mathbf{c}_k^{\mathrm{T}} \mathbf{H}_f)^2] & (3.25) \\ &= \mathrm{E}[H^2(n;k)] - 2\mathbf{c}_k^{\mathrm{T}} \mathrm{E}[H(n;k)\mathbf{H}_f] + \mathbf{c}_k^{\mathrm{T}} \mathrm{E}[\mathbf{H}_f \mathbf{H}_f^{\mathrm{T}}]\mathbf{c}_k \,. & (3.26) \end{aligned}$$

Differentiating with respect to $\mathbf{c}_k^{\mathrm{T}}$, one obtains

$$\frac{\partial \mathrm{MSE}_{\mathrm{est}}(k)}{\partial \mathbf{c}_k^{\mathrm{T}}} = -2\mathrm{E}[H(n;k)\mathbf{H}_f] + 2\mathrm{E}[\mathbf{H}_f \mathbf{H}_f^{\mathrm{T}}]\mathbf{c}_k. \tag{3.27}$$

Setting this equal to zero yields

$$\mathrm{E}[\mathbf{H}_f \mathbf{H}_f^{\mathrm{T}}]\mathbf{c}_k = \mathrm{E}[H(n;k)\mathbf{H}_f]. \tag{3.28}$$

The matrices involved may be expanded to give

$$\mathrm{E}[H(n;k)\mathbf{H}_f] = \begin{bmatrix} \mathrm{E}[H(n;k)H(n;k_1-1)] \\ \mathrm{E}[H(n;k)H(n;k_1-2)] \\ \cdots \\ \mathrm{E}[H(n;k)H(n;k_1-P_f)] \end{bmatrix} \tag{3.29}$$

[4]Note that $\mathrm{MSE}_{\mathrm{est}}(k)$ is defined only for $k \in [k_1, k_2]$.

$$\mathrm{E}[\mathbf{H}_f\mathbf{H}_f^{\mathrm{T}}] = \begin{bmatrix} \mathrm{E}[H(n;k_1-1)H(n;k_1-1)] & \cdots & \mathrm{E}[H(n;k_1-1)H(n;k_1-P_f)] \\ \mathrm{E}[H(n;k_1-2)H(n;k_1-1)] & \cdots & \mathrm{E}[H(n;k_1-2)H(n;k_1-P_f)] \\ \cdots & & \cdots \\ \mathrm{E}[H(n;k_1-P_f)H(n;k_1-1)] & \cdots & \mathrm{E}[H(n;k_1-P_f)H(n;k_1-P_f)] \end{bmatrix}. \tag{3.30}$$

Consider using an unbiased estimator with window size $W$ for the practical calculation of the autocovariance function[5] of $H(n;k)$:

$$\mathrm{E}[H(n;k)H^*(n;k_B)] \simeq \frac{1}{W-k_B}\sum_{i=k_B}^{W-1} H(n;i)H(n;i-k_B) \tag{3.31}$$

$$= q_n(k_B)\ ,\ k_B > 0. \tag{3.32}$$

Note that the autocovariance function is only calculated on the points available inside the window. For $k_B < 0$, it is defined as

$$\mathrm{E}[H(n;k)H^*(n;k_B)] \simeq \frac{1}{W-k_B}\sum_{i=0}^{W-k_B+1} H(n;i)H(n;i-k_B) \tag{3.33}$$

$$= q_n(k_B)\ ,\ k_B > 0. \tag{3.34}$$

When $H(n;k)$ is a sinusoid in $k$ (and more generally, a *sum* of sinusoids), one has $q_n(k_B) = q_n(-k_B)$ as is now shown. First, let

$$H(n;k) = \sin\left(\frac{2\pi\omega_n k}{N}\right). \tag{3.35}$$

Then

$$q_n(k;k_B) = \frac{1}{W-k_B}\sum_{i=k_B}^{W-1} \sin\left(\frac{2\pi\omega_n i}{N}\right)\sin\left(\frac{2\pi\omega_n(i-k_B)}{N}\right) \tag{3.36}$$

$$= \frac{1}{W-k_B}\sum_{i=k_B}^{W-1} -\frac{1}{2}\left(\cos\left(\frac{4\pi\omega_n i}{N} - \frac{2\pi\omega_n k_B}{N}\right) - \right.$$

[5] Although $H(n;k)$ is real, the conjugate symbol "*" is preserved in order to be consistent, and to encourage the use of this method with other, possibly complex transforms. The unbiased covariance estimator used is that on p. 237 of [13].

$$
\begin{aligned}
& \cos\left(\frac{2\pi\omega_n k_B}{N}\right)\Bigg) && (3.37) \\
&= \frac{-1}{2(W-k_B)} \sum_{i=k_B}^{W-1} \cos\left(\frac{4\pi\omega_n i}{N} - \frac{2\pi\omega_n k_B}{N}\right) + \\
& \frac{1}{2(W-k_B)} \cos\left(\frac{2\pi\omega_n k_B}{N}\right) && (3.38) \\
&= q_n(k_B). && (3.39)
\end{aligned}
$$

On the other hand,

$$
\begin{aligned}
q_n(k;k_B) &= \frac{1}{W-k_B} \sum_{i=0}^{W-k_B+1} \sin\left(\frac{2\pi\omega_n i}{N}\right) \sin\left(\frac{2\pi\omega_n (i+k_B)}{N}\right) && (3.40) \\
&= \frac{-1}{2(W-k_B)} \sum_{i=0}^{W-k_B+1} \cos\left(\frac{4\pi\omega_n i}{N} + \frac{2\pi\omega_n k_B}{N}\right) + \\
& \frac{1}{2(W-k_B)} \cos\left(\frac{2\pi\omega_n (-k_B)}{N}\right) && (3.41) \\
&= q_n(-k_B). && (3.42)
\end{aligned}
$$

Performing the change of variables $x = i - k_B$, one obtains:

$$
\begin{aligned}
q_n(-k_B) &= \frac{-1}{2(W-k_B)} \sum_{x=i+k_B}^{W-1} \cos\left(\frac{4\pi\omega_n (x-k_B)}{N} + \frac{2\pi\omega_n k_B}{N}\right) + \\
& \frac{1}{2(W-k_B)} \cos\left(\frac{2\pi\omega_n (-k_B)}{N}\right) && (3.43) \\
&= \frac{-1}{2(W-k_B)} \sum_{x=i+k_B}^{W-1} \cos\left(\frac{4\pi\omega_n (x)}{N} - \frac{2\pi\omega_n k_B}{N}\right) + \\
& \frac{1}{2(W-k_B)} \cos\left(\frac{2\pi\omega_n (-k_B)}{N}\right) && (3.44) \\
&= q_n(k_B) && (3.45)
\end{aligned}
$$

Hence, using the fact that $H(n;k)$ is the DCT of an impulse response, and is therefore a sum of sinusoids, one has $q_n(k_B) = q_n(-k_B)$, and the expectations in (3.29) and (3.30) now become

$$\mathrm{E}[H(n;k)\mathbf{H}_f] \simeq \begin{bmatrix} q_n(k-k_1+1) \\ q_n(k-k_1+2) \\ \cdots \\ q_n(k-k_1+P_f) \end{bmatrix} \tag{3.46}$$

$$\mathrm{E}[\mathbf{H}_f\mathbf{H}_f^{\mathrm{T}}] \simeq \begin{bmatrix} q_n(0) & q_n(1) & \cdots & q_n(P_f-1) \\ q_n(1) & q_n(0) & \cdots & q_n(P_f-2) \\ \vdots & \vdots & \ddots & \vdots \\ q_n(P_f-1) & q_n(P_f-2) & \cdots & q_n(0) \end{bmatrix} . \tag{3.47}$$

To solve the forward linear prediction problem, one uses (3.46) and (3.47) to solve for $\mathbf{c}_k$ in (3.28). The result is then inserted into (3.24) to obtain $b_{\mathrm{ss}}(n;k)$ for $k \in [k_1, k_2]$. With regard to computation, (3.47) needs to be computed only once, whereas (3.46) needs to be obtained for each $k$ in the gap. Further analysis of the computational complexity is offered in §4.3.

### Backward LP

Along the same lines as forward linear prediction, *backward* LP sets $P_f = 0$ and uses $H(n;k), k \in [k_2+1, k_2+P_b]$ to predict $H(n;k),\ k \in [k_1, k_2]$:

$$b_{\mathrm{ss}}(n;k) = \mathbf{d}_k^{\mathrm{T}}\mathbf{H}_b\ . \tag{3.48}$$

Following a similar reasoning as for forward LP, one obtains

$$\mathrm{E}[\mathbf{H}_b\mathbf{H}_b^{\mathrm{T}}]\mathbf{d}_k = \mathrm{E}[H(n;k)\mathbf{H}_b] \tag{3.49}$$

where

$$\mathrm{E}[H(n;k)\mathbf{H}_b] = \begin{bmatrix} q_n(k-k_2-1) \\ q_n(k-k_2-2) \\ \vdots \\ q_n(k-k_2-P_b) \end{bmatrix} \tag{3.50}$$

$$\mathrm{E}[\mathbf{H}_b\mathbf{H}_b^{\mathrm{T}}] = \begin{bmatrix} q_n(0) & q_n(1) & \cdots & q_n(P_b-1) \\ q_n(1) & q_n(0) & \cdots & q_n(P_b-2) \\ \vdots & \vdots & \ddots & \vdots \\ q_n(P_b-1) & q_n(P_b-2) & \cdots & q_n(0) \end{bmatrix}. \tag{3.51}$$

In summary, one computes $\mathbf{d}_k$ from (3.49), then finds $b_{\mathrm{ss}}(n;k)$ from (3.48), thus obtaining an estimate of $H(n;k)$ inside the gap. The computational complexity is identical to that of forward LP for the same predictor order.

### Simultaneous LP

When both a forward and a backward linear predictor are desired, both $P_f$ and $P_b$ are nonzero, and

$$b_{\mathrm{ss}}(n;k) = \mathbf{c}_k^{\mathrm{T}}\mathbf{H}_f + \mathbf{d}_k^{\mathrm{T}}\mathbf{H}_b. \tag{3.52}$$

The mean-square estimation error is calculated as follows:

$$\begin{aligned} \mathrm{MSE}_{\mathrm{est}}(k) &= \mathrm{E}[(H(n;k) - (\mathbf{c}_k^{\mathrm{T}}\mathbf{H}_f + \mathbf{d}_k^{\mathrm{T}}\mathbf{H}_b))^2] & (3.53) \\ &= \mathrm{E}[H^2(n;k)] - \\ &\quad 2\mathrm{E}[H(n;k)(\mathbf{c}_k^{\mathrm{T}}\mathbf{H}_f + \mathbf{d}_k^{\mathrm{T}}\mathbf{H}_b)] + \\ &\quad \mathrm{E}[(\mathbf{c}_k^{\mathrm{T}}\mathbf{H}_f + \mathbf{d}_k^{\mathrm{T}}\mathbf{H}_b)(\mathbf{c}_k^{\mathrm{T}}\mathbf{H}_f\mathbf{d}_k^{\mathrm{T}}\mathbf{H}_b)^{\mathrm{T}}] & (3.54) \\ &= \mathrm{E}[H^2(n;k)] \\ &\quad -2\mathbf{c}_k^{\mathrm{T}}\mathrm{E}[H(n;k)\mathbf{H}_f] - 2\mathbf{d}_k^{\mathrm{T}}\mathrm{E}[H(n;k)\mathbf{H}_b] \\ &\quad +\mathbf{c}_k^{\mathrm{T}}\mathrm{E}[\mathbf{H}_f\mathbf{H}_f^{\mathrm{T}}]\mathbf{c}_k + \mathbf{c}_k^{\mathrm{T}}\mathrm{E}[\mathbf{H}_f\mathbf{H}_b^{\mathrm{T}}]\mathbf{d}_k \\ &\quad +\mathbf{d}_k^{\mathrm{T}}\mathrm{E}[\mathbf{H}_b\mathbf{H}_f^{\mathrm{T}}]\mathbf{c}_k + \mathbf{d}_k^{\mathrm{T}}\mathrm{E}[\mathbf{H}_b\mathbf{H}_b^{\mathrm{T}}]\mathbf{d}_k. & (3.55) \end{aligned}$$

Differentiating with respect to $\mathbf{c}_k^{\mathrm{T}}$ and $\mathbf{d}_k^{\mathrm{T}}$, and noting that

$$\mathbf{c}_k^{\mathrm{T}}\mathrm{E}[\mathbf{H}_f\mathbf{H}_b^{\mathrm{T}}]\mathbf{d}_k = \left(\mathbf{c}_k^{\mathrm{T}}\mathrm{E}[\mathbf{H}_f\mathbf{H}_b^{\mathrm{T}}]\mathbf{d}_k\right)^{\mathrm{T}} = \mathbf{d}_k^{\mathrm{T}}\mathrm{E}[\mathbf{H}_b\mathbf{H}_f^{\mathrm{T}}]\mathbf{c}_k \tag{3.56}$$

and

$$\mathbf{d}_k^{\mathrm{T}}\mathrm{E}[\mathbf{H}_b\mathbf{H}_f^{\mathrm{T}}]\mathbf{c}_k = \left(\mathbf{d}_k^{\mathrm{T}}\mathrm{E}[\mathbf{H}_b\mathbf{H}_f^{\mathrm{T}}]\mathbf{c}_k\right)^{\mathrm{T}} = \mathbf{c}_k^{\mathrm{T}}\mathrm{E}[\mathbf{H}_f\mathbf{H}_b^{\mathrm{T}}]\mathbf{d}_k, \tag{3.57}$$

one obtains:

$$\frac{\partial \mathrm{MSE}_{\mathrm{est}}(k)}{\partial \mathbf{c}_k^{\mathrm{T}}} = -2\mathrm{E}[H(n;k)\mathbf{H}_f] + 2\mathrm{E}[\mathbf{H}_f\mathbf{H}_f^{\mathrm{T}}]\mathbf{c}_k + 2\mathrm{E}[\mathbf{H}_f\mathbf{H}_b^{\mathrm{T}}]\mathbf{d}_k \tag{3.58}$$

$$\frac{\partial \mathrm{MSE}_{\mathrm{est}}(k)}{\partial \mathbf{d}_k^{\mathrm{T}}} = -2\mathrm{E}[H(n;k)\mathbf{H}_b] + 2\mathrm{E}[\mathbf{H}_b\mathbf{H}_f^{\mathrm{T}}]\mathbf{c}_k + 2\mathrm{E}[\mathbf{H}_b\mathbf{H}_b^{\mathrm{T}}]\mathbf{d}_k. \tag{3.59}$$

To solve for $\mathbf{c}_k$ and $\mathbf{d}_k$, one sets both equations to zero, thus obtaining

$$\begin{bmatrix} \mathrm{E}[\mathbf{H}_f\mathbf{H}_f^{\mathrm{T}}] & \mathrm{E}[\mathbf{H}_f\mathbf{H}_b^{\mathrm{T}}] \\ \mathrm{E}[\mathbf{H}_b\mathbf{H}_f^{\mathrm{T}}] & \mathrm{E}[\mathbf{H}_b\mathbf{H}_b^{\mathrm{T}}] \end{bmatrix} \begin{bmatrix} \mathbf{c}_k \\ \mathbf{d}_k \end{bmatrix} = \begin{bmatrix} \mathrm{E}[H(n;k)\mathbf{H}_f] \\ \mathrm{E}[H(n;k)\mathbf{H}_b] \end{bmatrix} \tag{3.60}$$

where $\mathrm{E}[\mathbf{H}_f\mathbf{H}_f^{\mathrm{T}}]$, $\mathrm{E}[H(n;k)\mathbf{H}_f]$, $\mathrm{E}[\mathbf{H}_b\mathbf{H}_b^{\mathrm{T}}]$ and $\mathrm{E}[H(n;k)\mathbf{H}_b]$ are defined as before, and

$$\mathrm{E}[\mathbf{H}_f\mathbf{H}_b^{\mathrm{T}}] = \left(\mathrm{E}[\mathbf{H}_b\mathbf{H}_f^{\mathrm{T}}]\right)^{\mathrm{T}} =$$
$$\begin{bmatrix} q_n(k_1-k_2-2) & q_n(k_1-k_2-3) & \cdots & q_n(k_1-k_2-(P_b+1)) \\ q_n(k_1-k_2-3) & q_n(k_1-k_2-4) & \cdots & q_n(k_1-k_2-(P_b+2)) \\ q_n(k_1-k_2-4) & q_n(k_1-k_2-5) & \cdots & q_n(k_1-k_2-(P_b+3)) \\ \cdots & \cdots & \ddots & \cdots \\ q_n(k_1-k_2-(1+P_f)) & q_n(k_1-k_2-(2+P_f)) & \cdots & q_n(k_1-k_2-(P_b+P_f)) \end{bmatrix}. \tag{3.61}$$

The large, left-hand side matrix in (3.60) is symmetric, but unlike the matrices in simple forward or simple backward LP, it is not Toeplitz. The computational complexity of the solution is analyzed in §4.3.

### Convex-Combination LP

To save computation time but still maintain some form of forward and backward prediction, one may opt for *convex-combination* LP. The idea is to calculate $\mathbf{c}_k$ as for forward LP and $\mathbf{d}_k$ as for backward LP, and then to form

$$b_{\mathrm{ss}}(n;k) = \alpha(k)\ \mathbf{c}_k^{\mathrm{T}}\mathbf{H}_f + (1-\alpha(k))\mathbf{d}_k^{\mathrm{T}}\mathbf{H}_b \tag{3.62}$$

where

$$\alpha(k) = \frac{k_2 - k + 1}{k_2 - k_1 + 2} \;,\; k \in [k_1, k_2] \tag{3.63}$$

weights the prediction towards either forward or backward LP, as shown in Figure 3.9. It is implicit in the definition of convex-combination LP that the variation of $\alpha(k)$ with $k$ is linear. This particular shape was chosen to illustrate the concept of a transition region between forward and backward LP, rather than to stress the merits of a linear dependency on $k$.

Figures 3.10 to 3.13 show typical results obtained using the linear predictive method of spectral shaping described in this Section. Preliminary observations suggest that the values of $H(n;k)$ in the gap can be quite successfully estimated from values outside the gap. A more detailed and measurable assessment of the performance of LP spectral shaping is offered in Chapter 4.

### 3.3.3 Limitations of Linear Predictive Spectral Shaping

The single most important limitation of the LP spectral shaping method is the requirement that the gap be a *contiguous* block with $P_f$ and $P_b$ points available, *contiguously*, on either side. In other words, there must be enough "room" around a gap in order to apply linear predictive methods. This condition can be relaxed, however, by allowing a more general, albeit more complex formulation of the problem. Though not treated here, such a reformulation would employ much of the groundwork set forth in this Thesis.

The fact that exclusive treatment has been give to the DCT throughout this work can also be considered as a limitation. The preliminary phase has involved the DCT, and future work may involve the application of spectral shaping to a wider variety of transforms. Nevertheless, spectral shaping in the DCT domain yields some useful results, as is shown in the following chapter.

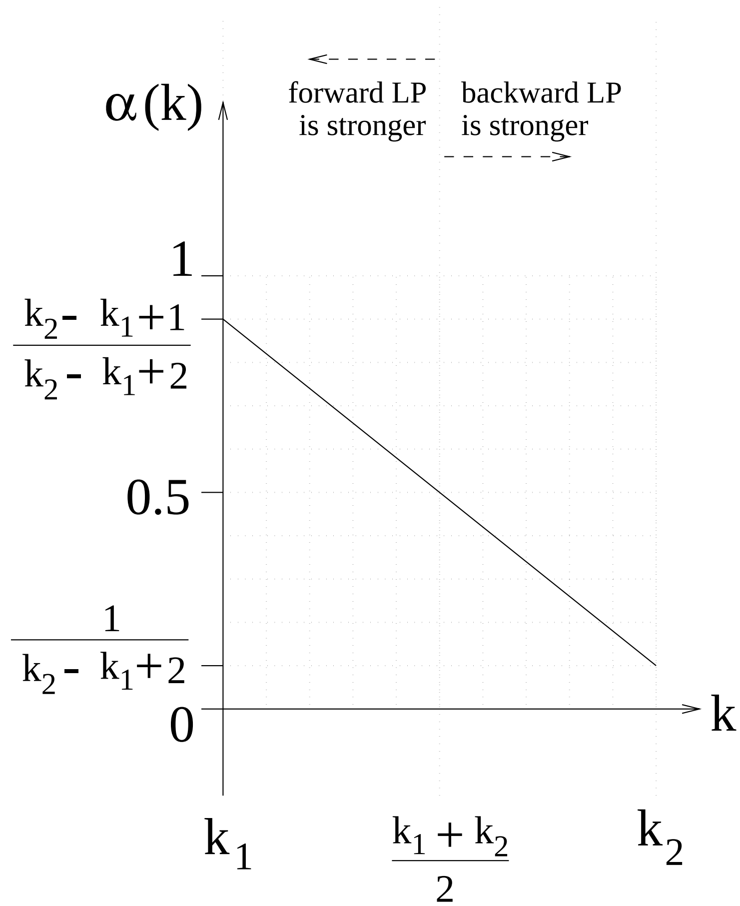


**Fig. 3.9** The shape of $\alpha(k)$. The relative influence of forward or backward LP in the estimation of a particular coefficient inside the gap depends on its relative position. Forward LP is used to shape coefficients at the "left" end, which gradually mixes with backward LP as the coefficients move "right". Finally, backward LP assumes control at the "right" end.

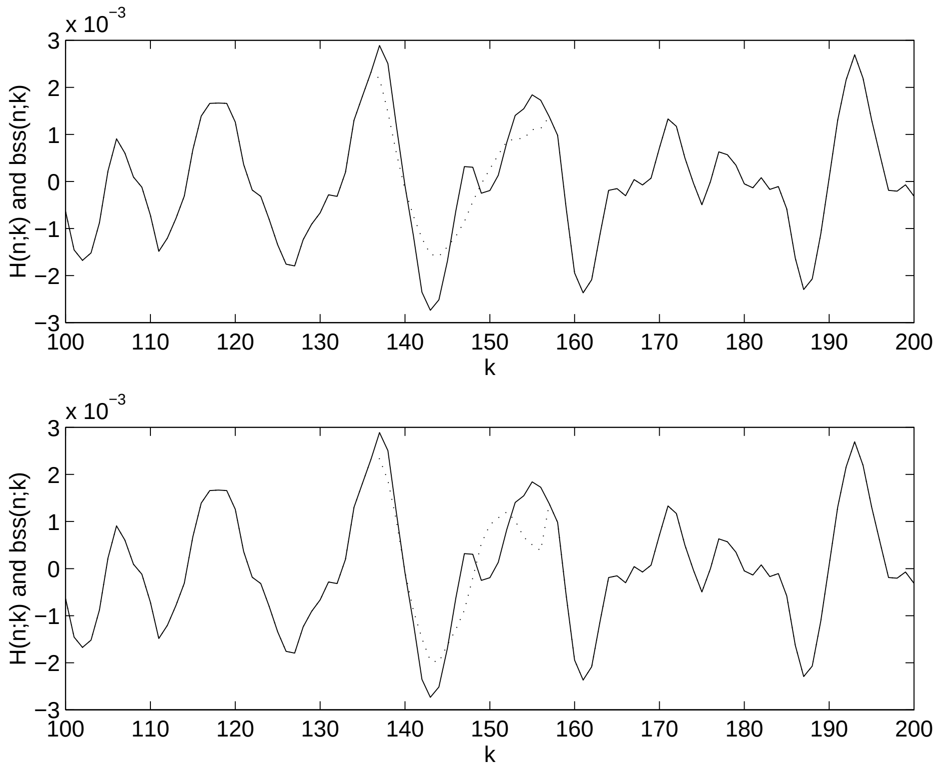


**Fig. 3.10** An example of forward LP with $k_1 = 137$, $k_2 = 156$, $P_f = 8$ (top), $P_f = 32$ (bottom). The solid line represents $H(n;k)$ and the dotted line represents the estimate obtained using LP spectral shaping, i.e., $b_{ss}(n;k)$.

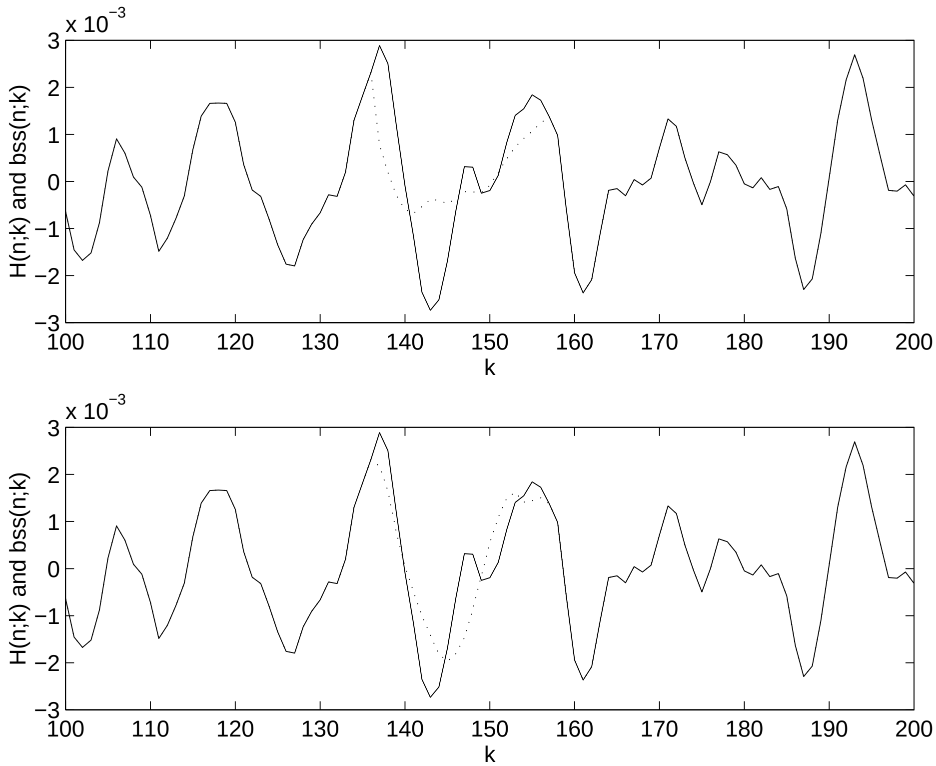


**Fig. 3.11** An example of backward LP with $k_1 = 137$, $k_2 = 156$, $P_b = 8$ (top), $P_b = 32$ (bottom). The solid line represents $H(n;k)$ and the dotted line represents the estimate obtained using LP spectral shaping, i.e., $b_{ss}(n;k)$.

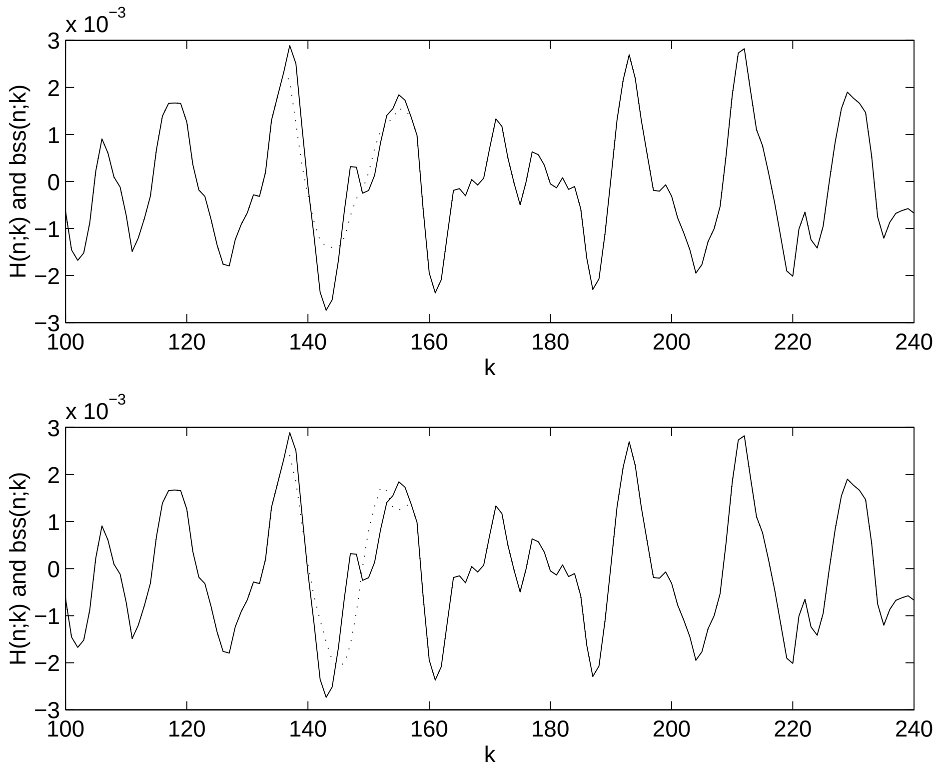


**Fig. 3.12** An example of simultaneous LP with $k_1 = 137$, $k_2 = 156$, $P'_f = P'_b = 4$ (top), $P'_f = P'_b = 16$ (bottom). The solid line represents $H(n;k)$ and the dotted line represents the estimate obtained using LP spectral shaping, i.e., $b_{ss}(n;k)$.

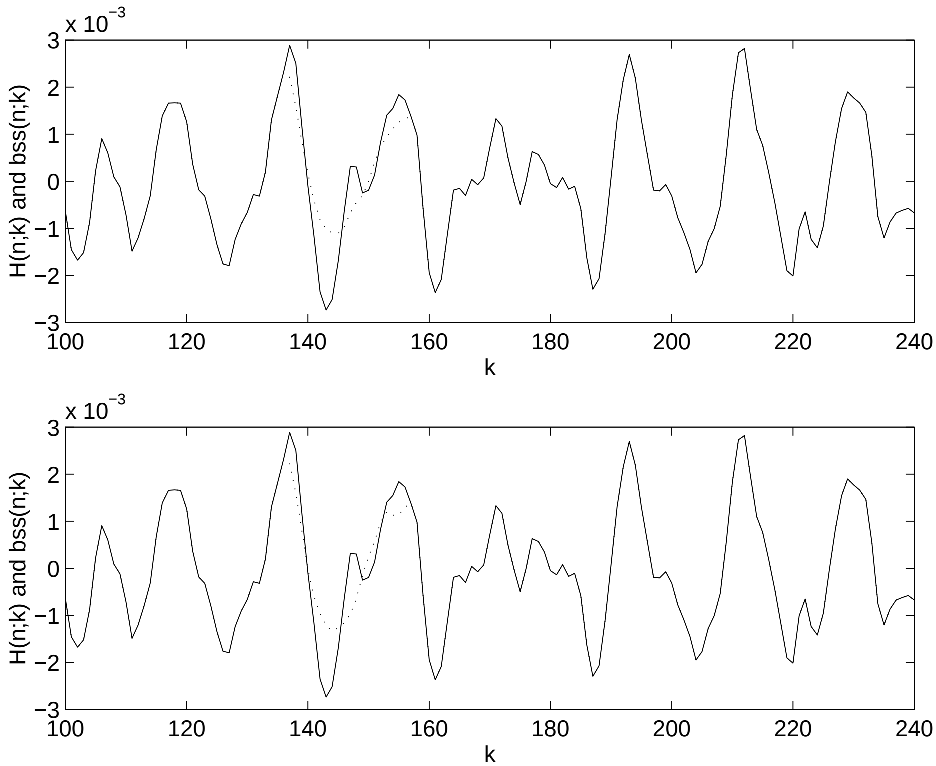


**Fig. 3.13** An example of convex-combination LP with $k_1 = 137$, $k_2 = 156$, $P'_f = P'_b = 4$ (top), $P'_f = P'_b = 16$ (bottom). The solid line represents $H(n;k)$ and the dotted line represents the estimate obtained using LP spectral shaping, i.e., $b_{ss}(n;k)$.

# Chapter 4

# Results

## 4.1 Experimental Protocol

Consider the transform-domain acoustical echo cancellation scenario shown in Figure 4.1, where $h(n;k)$ is taken as in Figure 2.4 and simulates propagation through a room. In addition, consider the following assumptions:

- All coefficients of the 512-tap DCT-domain adaptive filter have been initialized to zero at time $n = -\infty$;
- The talker's signal is null, i.e., the microphone signal contains pure echoes;
- There is a gap in the DCT of the reference signal in positions 137 through 156, between time $n \in (-\infty, 0)$;
- After an initial convergence period, for time $n \in (-\infty, 0)$, the filter coefficients $b(n;1)$ through $b(n;136)$ and $b(n;157)$ through $b(n;512)$ will properly track changes in $\mathbf{H}_n$, while coefficients $b(n;137)$ to $b(n;156)$ "freeze" and retain their initial value of zero.

At time $n = 0$, the reference signal covariance matrix gains full rank, i.e., its transform no longer contains any gaps. (Since the DCT is computed at every sample,

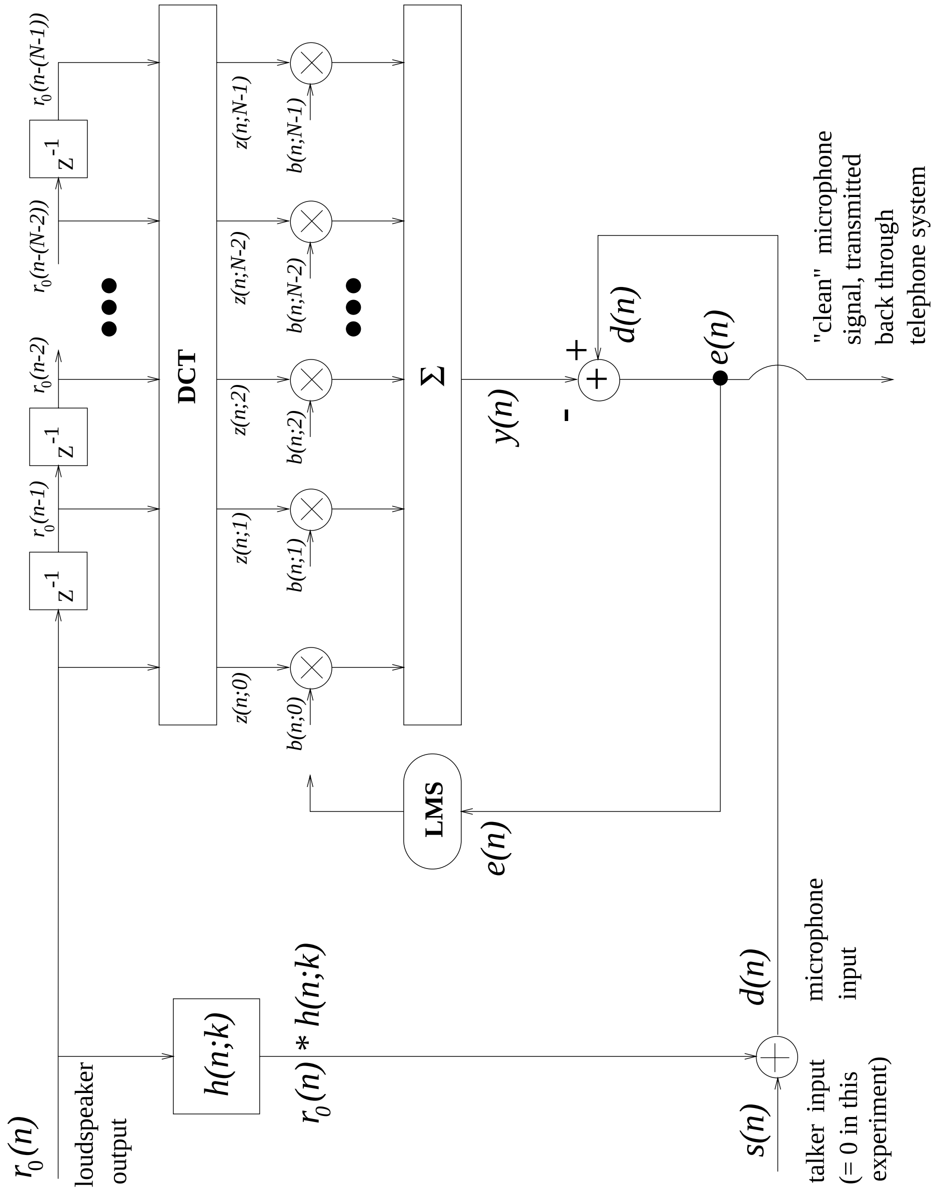


**Fig. 4.1** The simulation setup used to test the merits of linear predictive spectral shaping. The resulting signal, $e(n)$, is compared to the talker's signal $s(n)$. The latter has been conveniently set to zero, allowing easy calculation and comparison of the MSE. The LMS algorithm used is taken from (2.46) and (2.47).

it will take 512 samples before the gap vanishes entirely from the spectrum; for shorter filters used in enclosures with faster-decaying impulse responses, this delay is correspondingly reduced.) As a result, the previously frozen coefficients begin adapting from their value at time $n = 0$. If the coefficients have remained frozen to their initial value of zero, a large jump will result in the MSE.

However, if each coefficient in the gap is spectrally shaped, its value at time $n = 0$ is closer (in a mean-square sense) to the optimal value of the DCT of the room impulse response $H(0; k)$ for $k \in [137, 156]$. Consequently, the jump in MSE is significantly reduced, as shown by the dotted curve in Figure 4.2. The term "performance improvement" is defined as the mean separation (in dB) between the two curves after time $n = 0$.

## 4.2 Performance vs. Predictor Order

Using this experimental test bench, the performance improvement was calculated for various predictor orders, and is shown in Figure 4.3. The total number of predictor coefficients is defined as the sum of the forward and backward predictor orders. At first glance, the improvement gained by using spectral shaping is evident. In effect, the performance improvement leads to a quicker decay of the MSE and to a more effective cancellation of the acoustical echoes. Generalizing the results of this simulation to the real-world situation, the interference due to the reference signal is reduced. A more complete description of Figure 4.3 and subsequent curves is provided in §4.5.

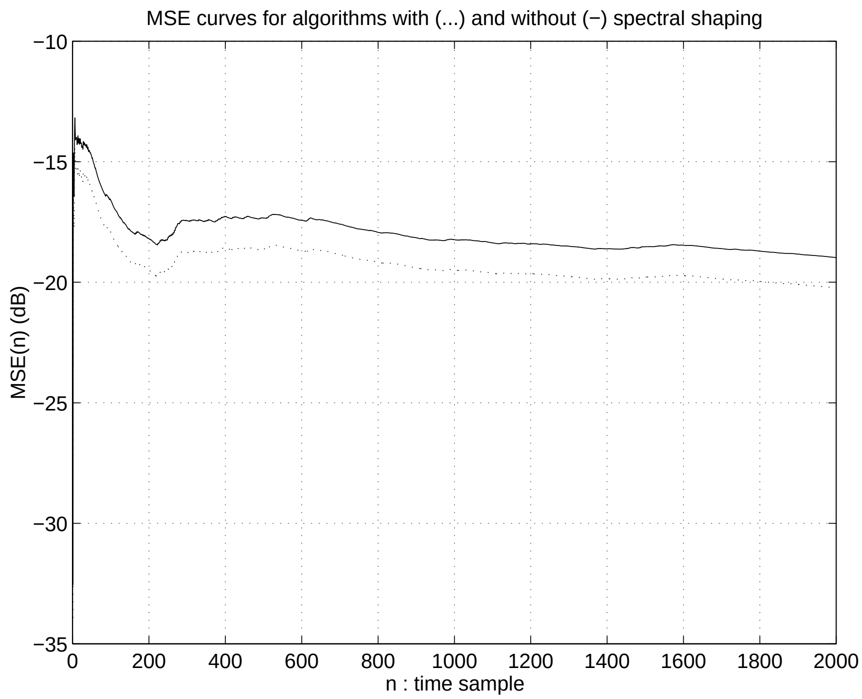


**Fig. 4.2** The curves illustrate the variation of the MSE as a function of time. For $n < 0$, rank($\mathbf{R}$) is low, and a gap is present in the DCT of the reference signal; at $n = n_1 = 0$, rank($\mathbf{R}$) increases to 512, its maximal value, though its effect is felt gradually by taking the DCT at every sample. The solid curve represents MSE($n$) that results from the coefficients in the gap being frozen to zero, i.e., their value at initialization. The dotted curve, on the other hand, shows MSE($n$) that results from having spectrally shaped the coefficients in the gap in anticipation of the increase in rank($\mathbf{R}$). The improvement in performance is calculated as the mean value of the difference between the two curves for $n > 0$.

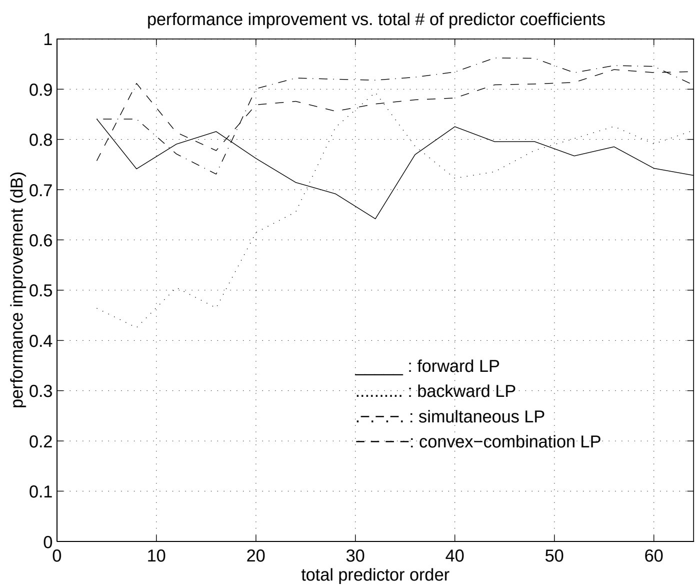


**Fig. 4.3** Performance improvement vs. total number of predictor coefficients. As well as demonstrating the clear performance advantage of using spectral shaping, this graph compares the relative merits of forward, backward, simultaneous, and convex-combination LP.

## 4.3 Complexity vs. Predictor Order

In the forward and backward LP methods, replacing a coefficient in the gap with its spectrally shaped counterpart requires solving a system of the form

$$[\mathbf{T}_P]\,\mathbf{x} = \mathbf{b} \tag{4.1}$$

where $\mathbf{T}_P$ is a fixed symmetric Toeplitz covariance matrix of order $P \in \{P_f, P_b\}$, while the right-hand cross-covariance vector varies with $k$, i.e., with the position in the gap. The solution to such a system requires $2P^2$ floating-point operations, or flops [27]. In the case of simultaneous LP, one is faced with a positive-definite system of the following form:

$$\begin{bmatrix} \mathbf{T}_{P'_f} & \mathbf{M} \\ \mathbf{M}^{\mathrm{T}} & \mathbf{T}_{P'_b} \end{bmatrix} \mathbf{x} = \mathbf{b} \tag{4.2}$$

where $\mathbf{T}_{P'_f}$ and $\mathbf{T}_{P'_b}$ are symmetric Toeplitz matrices of the corresponding order, and $\mathbf{M}$ is a $P'_f \times P'_b$ matrix. The entire matrix is thus symmetric, but not Toeplitz. An efficient approach to solving this system computes the Cholesky decomposition [27] in $(P'_f + P'_b)^3/6$ flops, then solves two lower-triangular systems, each requiring $(P'_f + P'_b)^2/2$ flops. For convex-combination LP, the computational requirement is equivalent to that of forward LP in addition to that of backward LP, with $P_f$ and $P_b$ replaced by $P'_f$ and $P'_b$, respectively.

The reason for appending "primes" to $P_f$ and $P_b$ in the case of simultaneous and convex-combination LP stems from the observation that performance must be compared "fairly" for each type of predictor. Hence the notion of *total number of predictor coefficients*, which allows each method of LP spectral shaping to be compared on an equal basis:

$$P_{\text{total}} = P_f + P_b + P'_f + P'_b. \tag{4.3}$$

For example, the following combinations have the same $P_{\text{total}}$: forward LP with $P_f = 8$, $P_b = 0$, $P'_f = 0$, $P'_b = 0$; backward LP with $P_f = 0$, $P_b = 8$, $P'_f = 0$, $P'_b = 0$;

| | flops for $\mathbf{c}_k$ | flops for $\mathbf{d}_k$ | flops for $b_{ss}(n;k)$ given $\mathbf{c}_k, \mathbf{d}_k$ | total flops for $b_{ss}(n;k)$ |
|---|---|---|---|---|
| Forw LP | $2P_f^2$ | 0 | $P_f$ | $2P_f^2 + P_f$ |
| Back LP | 0 | $2P_b^2$ | $P_b$ | $2P_b^2 + P_b$ |
| Sim. LP | $\frac{1}{6}(P'_f + P'_b)^3 + (P'_f + P'_b)^2$ | | $P'_f + P'_b$ | $\frac{1}{6}(P'_f + P'_b)^3 + P'_f$ $+(P'_f + P'_b)^2 + P'_b$ |
| C.C. LP | $2(P'_f)^2$ | $2(P'_b)^2$ | $P'_f + P'_b + 2$ | $2(P'_f)^2 + 2(P'_b)^2$ $+P'_f + P'_b + 2$ |

Table 4.1 Computational requirements of LP spectral shaping. The two additional flops needed for convex-combination LP reflect the need for multiplication by $\alpha$ and $1 - \alpha$.

simultaneous LP with $P_f = 0$, $P_b = 0$, $P'_f = 4$, $P'_b = 4$; convex-combination LP with $P_f = 0$, $P_b = 0$, $P'_f = 4$, $P'_b = 4$. Parallel comparison of the four methods would not be possible without the additional notation.

A summary of the computational complexity comparison is presented in Table 4.1. Based on this data, the actual number of flops required to compute $b_{ss}(n;k)$ for each of the spectral predictors is shown in Figure 4.4 for different values of $P_{\text{total}}$.

## 4.4 Performance vs. Complexity

Finally, the performance improvement obtained using LP spectral shaping is compared directly with complexity, by combining the two preceding plots (Figure 4.5). The complexity again refers to the number of flops required to compute each coefficient in the gap for a given $P_{\text{total}}$. This third plot clearly depicts simultaneous and

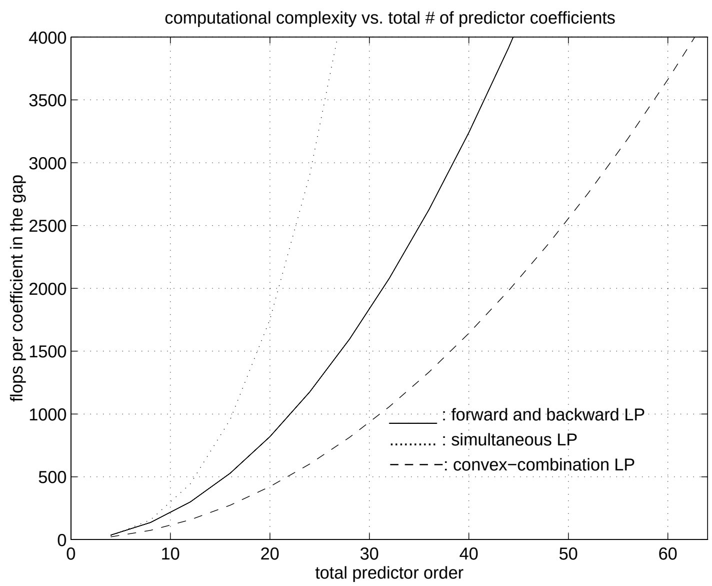


**Fig. 4.4** Computational complexity vs. total number of predictor coefficients. The complexity of simultaneous LP increases as the cube of the total predictor order, while that of forward, backward, and convex-combination LP increases as the square. Moreover, the complexity of convex-combination LP asymptotically approaches half that of forward or backward LP.

convex-combination LP as having the highest gain for almost any level of complexity.

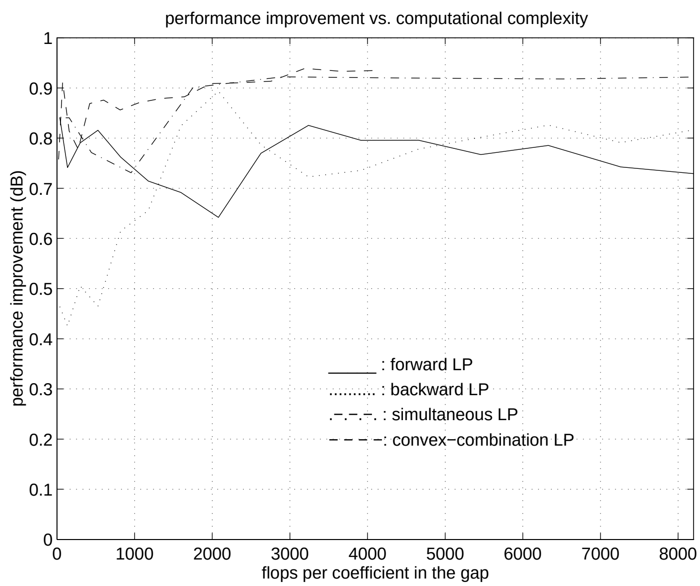


**Fig. 4.5** Performance vs. Computational Complexity. As can be seen, simultaneous and convex-combination LP almost always outperform forward and backward LP, for a given complexity.

## 4.5 Observations and Explanations

For a given complexity or $P_{\text{total}}$, the best improvement in performance is generally offered by simultaneous or convex-combination LP. This result is intuitively expected, as the two hybrid methods combine the benefits of both forward and backward linear prediction.

Forward and backward LP differ at low predictor orders, but offer similar performance at higher predictor orders. This is likely due to an "edge effect", whereby in this example $H(n;k)$ is smoother for $k < k_1$ than for $k > k_2$, hence allowing forward LP to predict the remainder of the spectrum more accurately than backward LP. The leveling occurs as the order increases because the importance of the local topography around the gap edge is gradually lost, giving way to an average smoothness on either side of the gap.

The complexity of simultaneous LP increases as the cube of $P_{\text{total}}$, while that of forward and backward LP increases as the square. The complexity of convex-combination LP is asymptotic to half that of forward or backward LP.

In all cases, LP spectral shaping reduces the jump in MSE caused by a vanishing gap in the DCT spectrum of the reference signal. Overall, one observes a reduction in MSE on the order of 1 dB. Although the improvement may not be dramatic in therms of dB, this result nevertheless that LP spectral shaping can be successfully applied to a practical situation.

# Chapter 5

# Summary and Conclusions

The discussion in Chapter 2 led to the implementation of a transversal FIR filter in order to cancel acoustical echoes in a room. The algorithm that is used must be adaptive, as the room impulse response is time-varying. Such an algorithm is the widely used LMS, which unfortunately suffers from a poor convergence rate, resulting in poor tracking of the room impulse response. To alleviate this problem, §2.2.2 introduces the transform-domain LMS algorithm, which substantially increases the convergence speed by closely replicating an eigendecomposition and effectively normalizing the transform spectrum of the reference signal. Since evidence in the literature points to the discrete cosine transform (DCT) as the transform of choice in speech-related applications, the DCT-domain LMS algorithm was adopted as the primary cancellation technique studied in this Thesis.

Chapter 3 analyzed in detail the phenomenon that occurs when the reference signal covariance matrix is rank-deficient: the presence of "gaps" in the transform spectrum causes "freezing" of the respective tap coefficients. While the remaning coefficients continue to adapt and respond to the changing room dynamics (and drive the MSE to a minimum), the frozen coefficients do as their title implies: they retain their value. In other words, after convergence, the resulting tap coefficient vector is one of many optimal vectors at the multi-dimensional minimum of the error surface. However, the full-rank solution is unique, and though it is located on that same hypersurface of minimum MSE, this unique solution may be "far" from the present solution containing frozen coefficients.

The undesirable consequence of the above is that the MSE will "jump" when the reference signal covariance matrix gains full rank. This occurs since there are no longer any gaps, and the (previously) frozen coefficients must converge to the unique solution. One can decrease the magnitude of this jump by "preparing" the frozen coefficients while the rank of the covariance matrix is still low. 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 time when the covariance matrix has low rank) is based on the values and positions of the coefficients which are outside the gap. From the theory of linear prediction thus arises the method of linear predictive (LP) spectral shaping.

Chapter 3 also describes several methods of LP spectral shaping which can be applied to a single, contiguous region of the spectrum, occupying the position of a gap in the reference signal DCT spectrum. Two of these are the conventional forward and backward methods. One also sees the introduction of simultaneous LP, which couples both forward and backward prediction into a single set of equations. The complexity of this hybrid predictor increases as the cube of the total number of predictor coefficients, in comparison to the square-law behavior of forward or backward LP. Finally, convex-combination LP is introduced. Instead of coupling both forward and backward LP into one set of equations, convex-combination LP finds both a forward and a backward solution (for each coefficient in the gap), and then weights the set of coefficients to be used towards either method, depending on the position of each coefficient in the gap. In contrast to simultaneous LP, the complexity of convex-combination LP is only half that of either forward or backward

LP.

The experiment performed in Chapter 4 consisted of shaping the coefficients in a portion of the spectrum corresponding to a gap of size 20. The gap vanished, i.e., the rank of the covariance matrix increased to its maximum, and the MSE was compared to that which would be obtained if the coefficients had been frozen to their initial value of zero. The performance improvement was substantial for all methods of LP spectral shaping, among which simultaneous and convex-combination LP consistently yielded the best performance for a given complexity.

There are, of course, limitations to the method of spectral shaping. For example, in the cases considered in this Thesis, it is necessary that there be a sufficient number of consecutive coefficients available on both sides of the gap, so as to accommodate the order of the chosen predictor, and to increase the reliability of the autocovariance function. In addition, the gap is assumed fixed in terms of its size and position. Future work could therefore focus on relaxing these constraints. Also, by defining a gap to be a set of DCT coefficients of value less than or equal to $\varepsilon$ (instead of zero), one is leaving the door open for further investigation. Finally, the approach is based solely on the DCT, and on only one transform-domain adaptation algorithm. An exhaustive study would require extending the experiment to a wider range of transforms and algorithms.

Nevertheless, the simplified cases studied here have served to promote LP spectral shaping as an effective method to counter the negative effects of a reference signal with a variable-rank covariance matrix. In addition, the methods developed in this Thesis may be directly applicable to linear predictive coding or signal reconstruction, when the estimation of two or more consecutive samples is required.

# Bibliography


[1] R. Wehrmann, "Concepts of Improving Hands-Free Speech Communication", *Proc. IEEE Int. Symp. Comp. Sys.*, 1992, pp. 1918–1921.

[2] J.R. Cásar-Corredera and Alcázar-Fernández, J., "An Acoustic Echo Canceller for Teleconference Systems", *Proc. IEEE Int. Conf. Acoust. Sp. Sig. Proc.*, 1986, pp. 1317–1320.

[3] H. Yasukawa *et al.*, "Echo Return Loss Required for Acoustic Echo Canceller Based on Subjective Assessment", *Proc. IEEE Int. Conf. Acoust. Sp. Sig. Proc.*, 1986, pp. 1309–1312.

[4] D. Bees, Blostein, M., and Kabal, P., "Reverberant Speech Enhancement Using Cepstral Processing", *Proc. IEEE Int. Conf. Acoust. Sp. Sig. Proc.*, 1991, pp. 977–980.

[5] S. Makino and Y. Kaneda, "Acoustic Echo Canceller Algorithm Based on the Variation Characteristics of a Room Impulse Response", *Proc. IEEE Int. Conf. Acoust. Sp. Sig. Proc.*, 1990, pp. 1133–1136.

[6] E. Hänsler, "The Hands-Free Telephone Problem – An Annotated Bibliography", *Signal Processing*, vol. 27 no. 3, June 1992, pp. 259–271.

[7] M.M. Sondhi and Berkley, D.A., "Silencing Echoes on the Telephone Network", *Proc. IEEE*, vol. 68, no. 8, August 1980, pp. 948–963.

[8] D.G. Messerschmitt, "Echo Cancellation in Speech and Data Transmission", *IEEE J. Selected Areas in Comm.*, vol. SAC-2 no. 2, March 1982, pp. 283–297.

[9] S. Haykin, *Adaptive Filter Theory*, 2nd ed., Prentice-Hall, Englewood Cliffs, NJ, 1991.

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

[11] T. Pétillon, Gilloire, A., and Theodoridis, S., "The Fast Newton Transversal Filter: An Efficient Scheme for Acoustic Echo Cancellation in Mobile Radio", *IEEE Trans. Sig. Proc.*, vol. 42 no. 3, March 1994, pp. 509–518.

[12] J.B. Allen and Berkley, D.A., "Image Method for Efficiently Simulating Small-Room Acoustics", *J. Acoust. Soc. Am.*, vol. 65 no. 4, April 1979, pp. 943–950.

[13] J.R. Deller, Proakis, J.G., and Hansen, J.H.L., *Discrete-Time Processing of Speech Signals*, Macmillan, New York, NY, 1993.

[14] S.S. Narayan, Peterson, A.M., and Narasimha, M.J., "Transform Domain LMS Algorithm", *IEEE Trans. Acoust. Sp. Sig. Proc.*, vol. ASSP-31 no. 3, June 1983, pp. 609–615.

[15] R. D. Gitlin and Magee, F.R., Jr., "Self-Orthogonalizing Adaptive Equalization Algorithms", *IEEE Trans. Comm.*, vol. COM-25 no. 7, July 1977, pp. 666–672.

[16] J.C. Lee and UN, C.K., "Performance of Transform-Domain LMS Adaptive Digital Filters", *IEEE Trans. Acoust. Sp. Sig. Proc.*, vol. ASSP-34 no. 3, June 1986, 499-510.

[17] R.D. Gitlin, Hayes, J.F., and Weinstein, S.B., *Data Communications Principles*, Plenum Press, New York, NY, 1992.

[18] B. Noble and Daniel, J.W., *Applied Linear Algebra*, 3rd ed., Prentice-Hall, Englewood Cliffs, NJ, 1988.

[19] B. Farhang-Boroujeny and Gazor, S., "Performance Analysis of the Transform Domain Normalized LMS Algorithm", *IEEE Proc. Int. Conf. Acoust. Sp. Sig. Proc.*, 1991, pp. 2133-2136.

[20] B. Champagne, "Adaptive Eigendecomposition of Data Covariance Matrices Based on First-Order Perturbations", *IEEE Trans. Sig. Proc.*, vol. 42 no. 10, October 1994, pp. 2758-2770.

[21] K.B. Yu, "Recursive Updating the Eigenvalue Decomposition of a Covariance Matrix", *IEEE Trans. Sig. Proc.* , vol. 39 no. 5, May 1991, pp. 1136-1145.

[22] M.J. Narasimha and Peterson, A.M., "On the Computation of the Discrete Cosine Transform", *IEEE Trans. Comm.*, vol. COM-26, no. 6, June 1978, pp. 934–936.

[23] N. Ahmed, Natarajan, T., and Rao, K.R., "Discrete Cosine Transform", *IEEE Trans. Computers*, January 1974, pp. 90-93.

[24] J.M. Tribolet and Crochiere, R.E., "Frequency Domain Coding of Speech", *IEEE Trans. Acoust. Sp. Sig. Proc.*, vol. ASSP-27, no. 5, October 1979, pp. 512–530.

[25] R.M. Gray, "On the Asymptotic Eigenvalue Distribution of Toeplitz Matrices", *IEEE Trans. Inf. Th.*, vol. IT-18 no. 6, Nov. 1972, pp. 725-730.

[26] F. Ling, "Convergence Characteristics of LMS and LS Adaptive Algorithms for Signals with Rank-Deficient Correlation Matrices", *Proc. IEEE Int. Conf. Acoust. Sp. Sig. Proc.*, 1988, pp. 1499–1502.

[27] G.H. Golub and Van Loan, C.F., *Matrix Computations*, Johns Hopkins University Press, Baltimore, MD, 1983.