ANALYSIS AND DEMONSTRATION OF THE QUANTILE VOCODER

Thesis by

Kumar Swaminathan

In Partial Fulfillment of the Requirements

for the Degree of

Doctor of Philosophy

California Institute of Technology

Pasadena, California

1986

(Submitted July 29, 1985)

ACKNOWLEDGEMENTS

In the course of this research, I have been influenced by more people than I can ever hope to acknowledge properly. Foremost among these has been Professor E. C. Posner. It was at his suggestion that the use of quantiles for speech compression was investigated. His helpful advice, encouragement and patience throughout the course of this thesis are gratefully acknowledged.

I am also deeply indebted to Professor D. B. Pisoni and other members of the Speech Laboratory at the Psychology Department, University of Indiana, for providing me with speech data on tape as well as a lot of useful literature in speech perception.

I have also benefited greatly from discussions with Dr. S. Townes, Jet Propulsion Laboratory. In addition, I would like to thank Dr. Mark Dolson, who helped me get started in the Acoustic Signal Processing Facility at Caltech, Professor P. P. Vaidyanathan for letting me use his tape recorder and preamplifier, Dr. Li Fung Chang and Phil Merkey for helping me with the TEX in the initial stages, and to all the subjects who participated in the listening test. To all of the above people, I extend my deepest gratitude.

Finally, I would like to thank all the members of my defence committee: Professor E. C. Posner, Professor R. J. McEliece, Professor M. Konishi, Professor J. Franklin, Professor P. P. Vaidyanathan and Dr. S. Townes. This research was funded by Jet Propulsion Laboratory. Its financial support is acknowledged. The facility used is the J. R. Pierce Communications Laboratory which is funded in part by the Jet Propulsion Laboratory, operated by Caltech for the National Aeronautics and Space Administration under contract NAS7-918.

ANALYSIS AND DEMONSTRATION OF THE QUANTILE VOCODER

ABSTRACT

A new scheme for speech compression is proposed, implemented and evaluated in this thesis. In this new scheme, the spectral envelope of the power spectral density of a speech frame is encoded using quantiles or order statistics. The perceptually important features of the spectral envelope are its peaks which correspond to the formant frequencies. The shape of the spectral envelope near the formants can be encoded by a careful choice of the quantiles and quantile orders. Algorithms to choose such a set of quantiles and quantile orders are described. It turns out that this can be done using very few quantiles. Data compression is achieved chiefly this way.

The quantile decoding algorithm estimates the spectral envelope from the quantiles and quantile orders. The first step is to set up a flat spectral density approximation. In this approximation, the spectral envelope is assumed to be constant in every interquantile range. This constant value is simply the average power (i.e., ratio of the difference in quantile orders to the difference in quantiles) in that interquantile range. It is shown that the flat spectral density approximation is the maximum entropy solution to the decoding problem. The flat spectral density approximation is then smoothed by fitting an all-pole or autoregressive model. Algorithms to determine the parameters of the autoregressive model are described. These algorithms involve the solution of a system of linear equations, which has a "Toeplitz plus Hankel" structure, followed by a standard spectral factorization. The algorithms can easily be extended to pole-zero models as well.

The information about the spectral fine structure is sent through the parameters of the excitation model. A multi-pulse excitation model in cascade with a pitch

predictor model has been chosen for this purpose. The theory of the multi-pulse model is reviewed, and algorithms to estimate the parameters of the multi-pulse model as well as the pitch predictor model are presented.

Quantization and encoding schemes of various transmission parameters are described. For high and medium bit rate applications, the parameters that need to be transmitted every frame are the quantiles, quantile orders, locations and amplitudes of the excitation pulses, parameters of the pitch predictor model and a gain term. For low bit rate applications, the quantile orders are fixed and so need not be transmitted. The quantization schemes for the quantile orders and for the gain term are shown to be optimal in the sense of minimizing the maximum spectral deviation due to quantization.

The quantile vocoder has been implemented in software at 4.8, 9.6, 16 and 24 Kbits/s. In order to test the vocoder, a speech data base of ten sentences spoken by one male and one female speaker has been used. The so-called *segmental signal-to-noise ratio* has been used as an objective performance measure to evaluate the vocoder at all bit rates. A subjective method for assessing the quality of the vocoder at various bit rates is also proposed and carried out. The results of the nonreal time quantile vocoder simulations at 4.8, 9.6, 16 and 24 Kbits/s have been recorded and will be played at the end of the talk. The quantile vocoder does indeed seem equivalent to or better than other vocoders at the same bit rates, according to informal listening tests.

ANALYSIS AND DEMONSTRATION OF THE QUANTILE VOCODER

LIST OF FIGURES

To my parents

for their love, encouragement, and support

CHAPTER 1
INTRODUCTION

1.1 History

The synthesis of natural sounding speech at low bit rates has been a topic of considerable interest in speech research. The underlying objective is to transmit (or store) speech with the highest possible quality over the least possible channel capacity (or storage capacity) with minimal complexity. One seeks to accomplish this using digital signal processing techniques.

Traditional speech coding methods can be divided broadly into two categories. They are (1) Waveform coders and (2) Vocoders. The *waveform coders* attempt to duplicate the waveform. To achieve bit reduction, the waveform coders are designed to be speech-specific. This is done by observing the statistics of the speech waveform so as to obtain minimal error while encoding the signal. Thus, the design of these coders are based on a statistical characterization of the speech waveform. Typically, these waveform coders tend to be independent of speaker characteristics, robust in the presence of noise, and are of low complexity. However, they can achieve only moderate reduction in bit rate. Examples of waveform coders are pulse code modulation (PCM), differential pulse code modulation (DPCM) and delta modulation (DM). Adaptive versions of these coders also exist. For further information on this topic, see the collection of papers edited by Jayant [1].

The *vocoders* achieve bit reduction by parameterization of speech information according to some physical model of the signal. The speech model that is often used is the source-system model (Chapter 2 [2] and Chapter 3 [3]). In this model we have a linear filter to model the spectral shaping of both the vocal tract and the vocal source. In order to achieve very low bit rates, the speech signal is traditionally first

classified as voiced or unvoiced. For voiced speech, the source is assumed to be a quasi-periodic pulse train with delta functions located at pitch period intervals. For unvoiced speech, the source is assumed to be white noise . The filter parameters and the pitch period are assumed to be constant over short segments of time. The complete model is described in Fig.1.1. The performance of vocoders is typically speaker-dependent and the output speech has a synthetic quality. However, they can achieve large reduction in transmission bandwidth. Examples of vocoders are the channel vocoder ([4]-[7]), homomorphic vocoder ([8]-[12]), and the linear prediction vocoder ([13]-[19]).

So we see that a large gap lies between the performance and bandwidth compression capability of waveform coders and vocoders. Some of the attempts to bridge the gap have focussed attention on preservation of short-time amplitude spectrum in an auditorily palatable way. Such coders are called *frequency domain coders*, and they reduce the bit rate of the waveform coders by taking greater advantage of the speech production models without making the algorithm totally dependent on them as in vocoders. Examples of frequency domain vocoders are the sub-band coders ([20]-[23]) and adaptive transform coders ([23]-[25]). Other attempts to bridge the gap between waveform coders and vocoders have focussed their attention on improving models of speech production and in particular models for source excitation. One of the most significant contributions in source excitation modeling is the multi-pulse excitation model proposed by Atal and Remde ([26]). In this approach, the excitation is simply modeled as a sequence of pulses with different, possibly negative, amplitudes and at distinct locations. There is no attempt to classify speech as voiced or unvoiced. Fig.1.2 describes this model. Other vocoders which employ improved source excitation are the *voice-excited vocoders* ([27]) and the

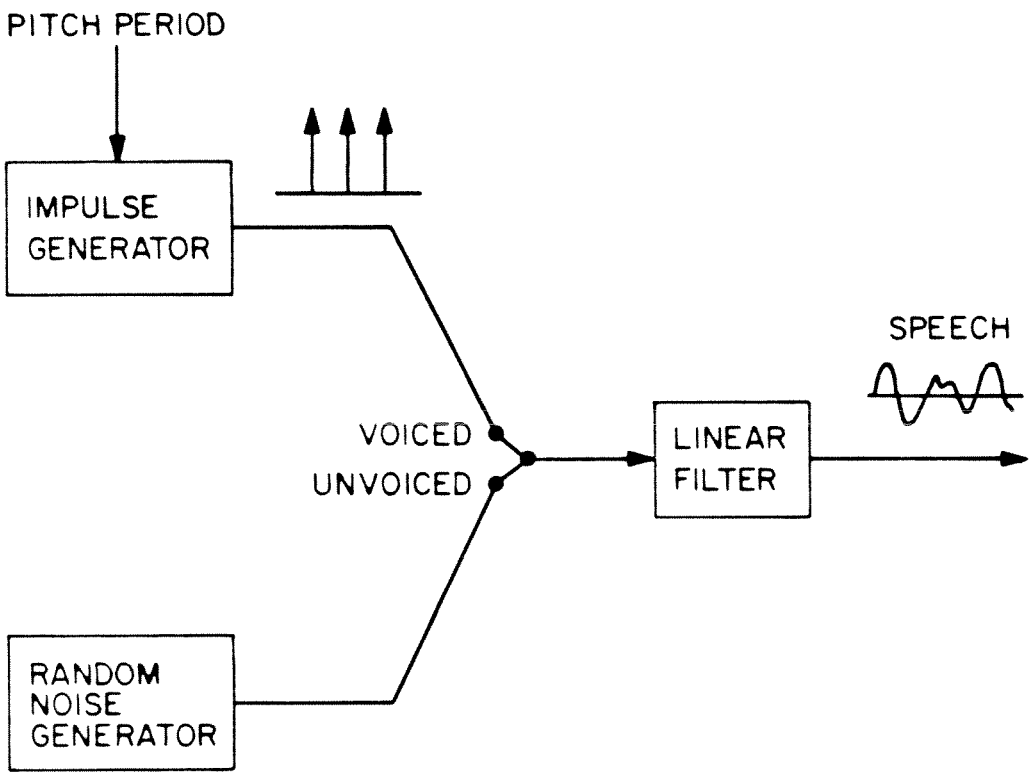


Fig 1.1 Model for speech production

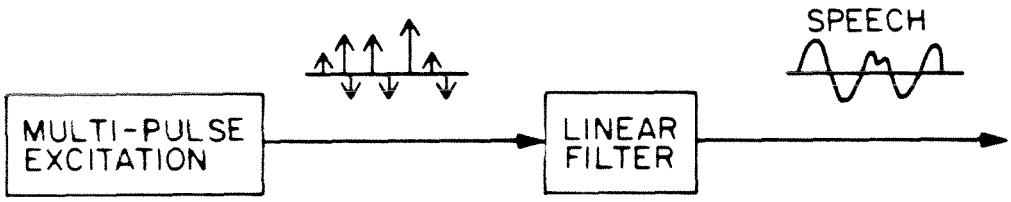


Fig 1.2 An improved model for speech production

baseband LPC residual vocoders ([28]). In addition to the above mentioned speech compression schemes, a variety of other speech coders have been proposed. These include *correlation vocoders* ([29]), *spectral envelope estimation vocoders* ([30]), etc. A brief discussion of the various speech compression schemes is given in Chapter 8 [2]. An excellent tutorial in speech coding along with a list of references is given in [31].

Despite all these schemes, the synthesis of high-quality speech at low bit rates and moderate complexity remains an elusive goal. Progress has been slow for a variety of reasons such as incomplete understanding of speech production and perception, lack of performance measures, etc. There are several approaches that can be investigated to improve the performance of speech coders at low bit rates. One approach is to investigate improved, yet tractable, models for speech production. Yet another approach is to fine-tune the performance of existing schemes. Finally, one can investigate newer ideas to carefully manipulate the speech information, with newer algorithms, so as to yield better performance at low bit rates. It is this last approach that we have taken in our work.

1.2 Overview

In our work, we propose a new scheme, the quantile vocoder, for speech compression. Quantiles or order statistics have been proposed earlier for compression of space telemetry data ([32], [33]). It had been shown in this context that quantiles are an efficient means of data compression requiring hardware of very low complexity. We will now discuss briefly the salient features of the quantile vocoder.

The short-time power spectrum of speech is characterized by a *spectral envelope* and a *spectral fine structure*. The envelope is due largely to the frequency shaping

effects of the vocal tract and, for voiced speech, to the spectrum of the glottal pulse. The fine structure is due to the excitation. The central idea in our scheme is to encode the spectral envelope using quantiles. The peaks of the spectral envelope, or the *formants*, are perceptually very important (Chapter 7 [2]). The quantiles are therefore chosen to "trap" these formants. It turns out that this can be done using very few quantiles. Data compression is achieved mostly this way.

The decoding algorithm estimates the spectral envelope as follows. It first sets up a flat spectral density approximation. Let θ_o $(= 0)$, θ_1, θ_2, ..., θ_q $(= \pi)$ be the quantiles corresponding to quantile orders E_o, E_1, E_2, ..., E_q. The quantiles are all multiples of $2\pi/N$ where N is the number of points on the unit circle at which the short-time Fourier transform of the speech segment was computed. The flat spectral density approximation is then given by

$$\begin{aligned} S_o(\omega) &= E_o & \omega &= \theta_o \\ &= \frac{2\pi(E_i - E_{i-1})}{N(\theta_i - \theta_{i-1})} & \theta_{i-1} &< \omega \leq \theta_i\,,\ 1 \leq i \leq q. \end{aligned}$$

The flat spectral density approximation is then smoothed by fitting an all-pole or autoregressive model $1/A(e^{j\omega})$. Thus, if we define $C(\omega) = |A(e^{j\omega})|^2$, then one approach to finding the parameters of the autoregressive model would be to minimize the distortion measure

$$E = \sum_{\substack{k=0 \\ \omega_k = 2\pi k/N}}^{N/2} \Big(S_o(\omega_k)C(\omega_k) - 1\Big)^2 W(\omega_k)$$

where $W(\omega)$ is a positive weighting function of ω.

Minimization of the distortion measure E leads to a system of linear equations which can be solved to obtain $C(\omega)$. This system of equations turns out to have a "Toeplitz plus Hankel" structure, and can therefore be efficiently solved using the block Levinson algorithm ([34]). One then obtains $A(e^{j\omega})$ from $C(\omega)$ using a

spectral factorization routine. However, the $C(\omega)$ so obtained is not guaranteed to be positive definite, though in practice this is the case most of the time. If $C(\omega)$ is not positive definite, the spectral factorization algorithm will fail. Spectral correction routines then have to be used to isolate the roots of $C(\omega)$ which are on the unit circle and to replace them by roots within and outside the unit circle. The autoregressive model $1/A(e^{j\omega})$ then completely defines the spectral envelope of the short-time power spectrum. One can easily extend the decoding algorithm to fit pole-zero or autoregressive moving average models as well.

The information about the spectral fine structure is sent through the parameters of the excitation model. A multi-pulse excitation model in cascade with a 1-tap pitch predictor model has been chosen for this purpose. Algorithms for obtaining the parameters of these models have been incorporated in the quantile vocoder.

Quantization and encoding schemes for various transmission parameters are described. The transmission parameters for medium and high bit rate applications are the quantiles, quantile orders, multi-pulse locations and amplitudes, pitch predictor parameters and a gain term. For low bit rate applications, quantile orders are fixed and so need not be transmitted. Quantization schemes which are optimal in the sense of minimizing the maximum spectral deviation due to quantization are developed for the quantile orders and the gain term. It turns out that for the quantile orders such an optimal quantization scheme is simply uniform quantization of the flat spectral density expressed in dB. Similarly for the gain, such an optimal quantization scheme is simply uniform quantization of gain expressed in dB. Simple combinatorial encoding schemes are used to encode the quantiles and the pulse locations. The pulse amplitudes, after proper normalization, are quantized uniformly. The tap coefficient of the 1-tap pitch predictor model is also subjected

to uniform quantization.

The quantile vocoder has been implemented at 4.8, 9.6, 16 and 24 Kbits/s. Using a data base of ten sentences spoken by one male and one female speaker, the quantile vocoder is evaluated at all the bit rates. The so-called *segmental signal-to-noise ratio* is used as an objective performance measure. The mean opinion score test is used for subjective evaluation.

1.3 Organization

The thesis is organized as follows. In Chapter 2, some basic concepts such as definition of a quantile, short-time Fourier analysis, etc. are reviewed. The chapter concludes with a brief discussion of the basic idea behind the quantile vocoder. In Chapter 3, we outline an algorithm for choosing a set of quantiles to encode the spectral envelope of a speech segment. In Chapter 4, a decoding algorithm is presented which estimates the spectral envelope from the chosen quantiles and quantile orders. Chapter 5 reviews the theory and implementation of the multi-pulse excitation model. The details of the implementation of the quantile vocoder, such as quantization and encoding of various parameters, at bit rates 24, 16, 9.6 and 4.8 Kbits/s are described in Chapter 6. Chapter 7 is mainly concerned with the evaluation of the quantile vocoder at all the bit rates. Both objective as well as subjective measures of coder performance are presented. The thesis concludes with a brief summary.

CHAPTER 2

BASIC CONCEPTS

2.1 Definitions

We begin by defining the quantile for a probability density function ([32]).

Definition 1: Consider a probability density $p(x)$ with a corresponding cumulative distribution function $F(x)$. Then a quantile x_p of order p is defined as the lower limit of all μ such that $F(\mu) > p$. (This order p is assumed to be in the range [0,1].)

We note that if the probability density $p(x)$ is non-zero over any finite interval then the quantile x_p of order p is simply that value for which $F(x_p) = p$. In Fig. 2.1, the definition is illustrated for such a case. This definition is easily extended to power spectral density since the power spectral density, if normalized, is a valid probability density function. Thus, we have

Definition 2: Consider a power spectral density $S(\Omega)$ which is normalized so that $\int_0^\infty S(\Omega)\, d\Omega = 1$. Let the corresponding cumulative power spectral density be $C_S(\Omega) = \int_0^\Omega S(\alpha)\, d\alpha$. Then a quantile Ω_p of order p is defined as the lower limit of all μ such that $C_S(\mu) > p$.

Again we note that if $S(\Omega)$ is non-zero over any finite frequency range, then the quantile Ω_p of order p is simply that value for which $C_S(\Omega_p) = p$. Since we are processing the signals digitally, the spectral density before digitization, extends only upto half the sampling frequency. In addition, one can compute the spectral density only at a finite number of frequencies, typically at a set of equally spaced frequencies. The definition of a quantile can be easily extended to such a discrete power spectral density, as follows:

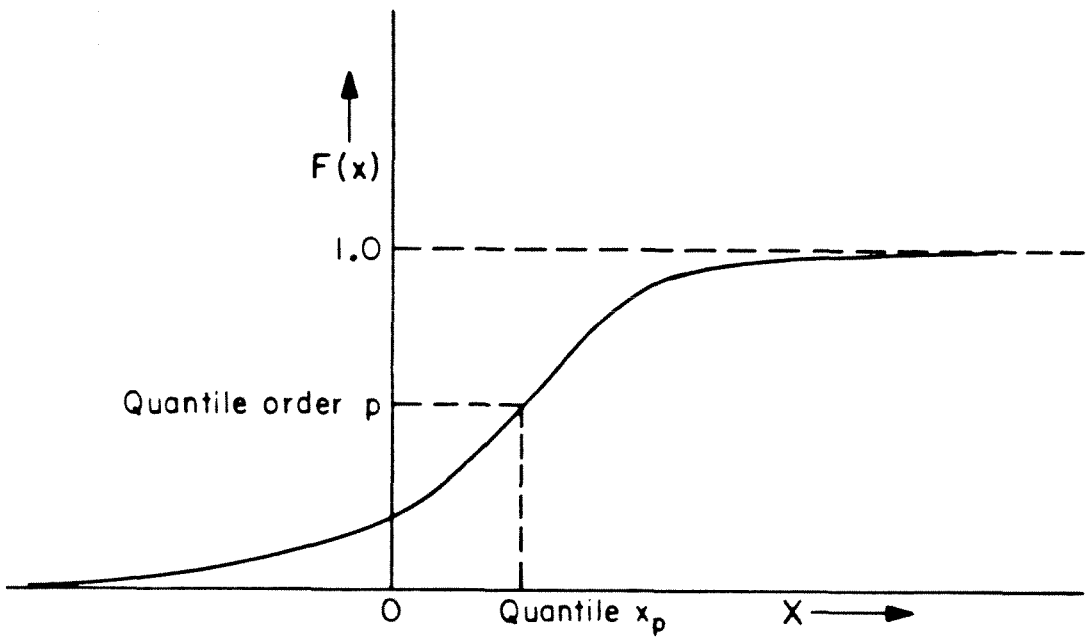


Fig. 2.1 Quantile of order p for a probability density

Definition 3: Consider a power spectral density $S(\omega_k)$ evaluated at $N/2$ equally spaced frequencies $\Omega_k = \omega_k f_S = 2\pi k f_S/N$, where f_S is the sampling frequency. We will assume that $S(\omega_k) > 0$ for all k. Further assume that the discrete power spectral density is normalized so that $\sum_{k=0}^{N/2} S(\omega_k) = 1$. If there exists a frequency $\omega_p = 2\pi k_p/N$ such that $\sum_{k=0}^{k_p} S(\omega_k) = p$, then ω_p is said to be the quantile of order p.

It must be borne in mind that in the case of a discrete power spectral density, not all orders p can be realized since the quantiles are restricted to be multiples of $2\pi/N$. However, it is possible to get as close as possible to any prescribed p by increasing N.

2.2 Short-time Fourier analysis

A primary assumption that we will make is that the speech signal is quasi-stationary, i.e., stationary over short segments of time. Such signals can be represented by a short-time Fourier transform. We will briefly discuss this concept in this section. (For further details refer to Chapter 6 of [3] and [37]-[42].)

The time-dependent short-time Fourier transform is given by

$$X_n(e^{j\omega}) = \sum_{m=-\infty}^{\infty} w(n-m)x(m)e^{-j\omega m}$$

where $x(m)$ represents samples of the speech signal and $w(n-m)$ represents a real window sequence which determines the portion of the input signal that receives emphasis at a particular time index n. We note that the short-time Fourier transform is a function of ω, the continuous frequency variable and n, the discrete time variable. Fig. 2.2 contains sketches of a typical $x(m)$ and $w(n-m)$ for several values of n.

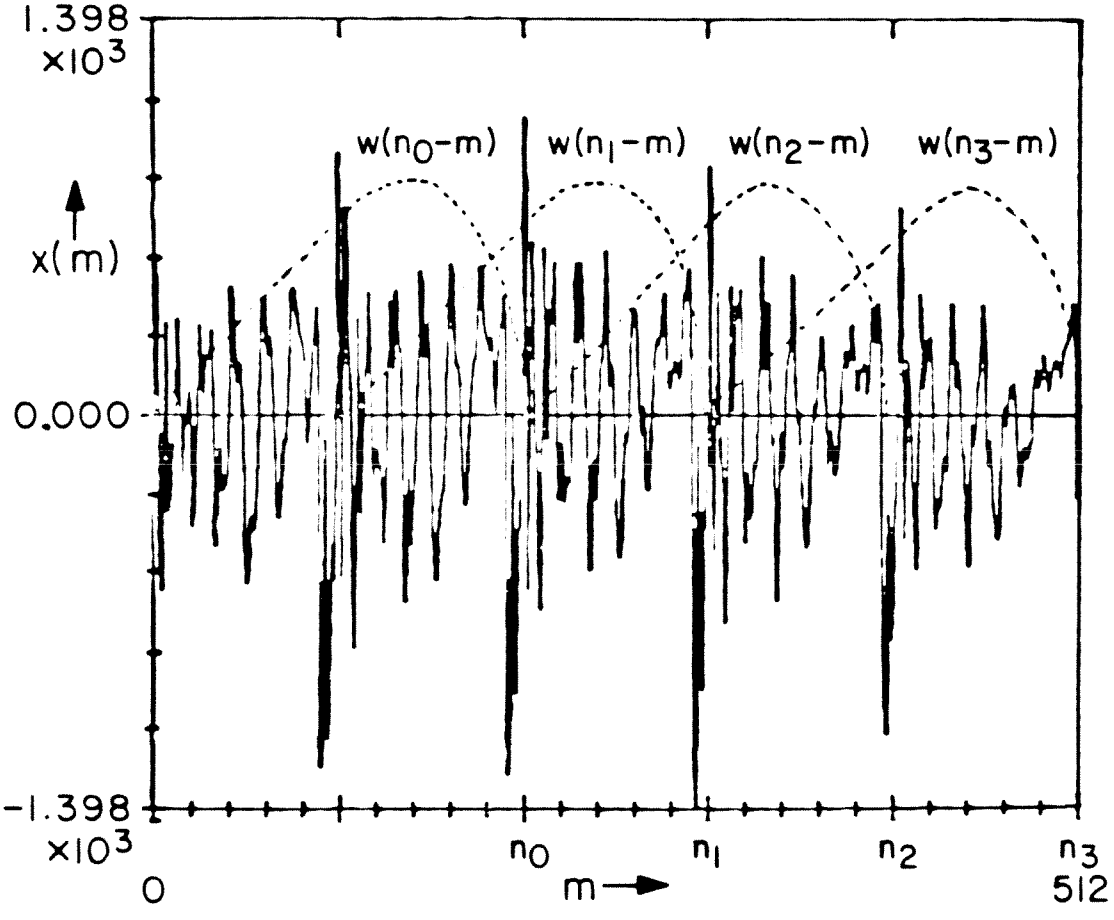


Fig. 2.2 Sketches of $x(m)$ and $w(n-m)$ for different n

The window $w(n)$ is usually a finite-length window. There are several factors which influence the choice of the window $w(n)$ as well as its length. (Refer to Section 6.1.1 of [3], Section 5.5 of [35] and Section 3.11 of [36] for detailed discussions.) We will illustrate some of these factors using the rectangular window and the Hamming window, both of equal length N_F, as examples. The rectangular window is defined as

$$w_R(n) = \begin{cases} 1 & \text{if } 0 \leq n \leq N_F - 1 \\ 0 & \text{otherwise} \end{cases}$$

and the Hamming window, which is a particular raised cosine window, is defined as

$$w_H(n) = \begin{cases} 0.54 - 0.46\cos\left(\dfrac{2\pi n}{N_F - 1}\right) & \text{if } 0 \leq n \leq N_F - 1 \\ 0 & \text{otherwise.} \end{cases}$$

In Fig. 2.3 both these windows as well as their amplitude spectra are shown.

The amplitude spectrum of a typical window, such as the rectangular or the Hamming window, is characterized by a *main lobe* and *several sidelobes.* The main lobe width is inversely related to the window length and also depends on the details of the window shape. To see how the width of the main lobe as well as the sidelobes affect the short-time Fourier transform, we recognize the short-time Fourier transform as the Fourier transform of the product of the signal and a shifted version of the window. But multiplication in the time domain is equivalent to convolution in the frequency domain. Thus, when the signal Fourier transform is convolved by the Fourier transform of shifted window $W(e^{j\omega})$, it is smeared primarily by the main lobe of $W(e^{j\omega})$, resulting in a loss of frequency resolution. The sidelobes of $W(e^{j\omega})$ cause the adjacent frequencies in the signal Fourier transform to interact by either reinforcing or cancelling. This kind of "spectral leakage" is also undesirable. To ensure that the loss of frequency resolution as well as the "spectral leakage" is kept within tolerable limits, we must choose a window whose transform has a narrow

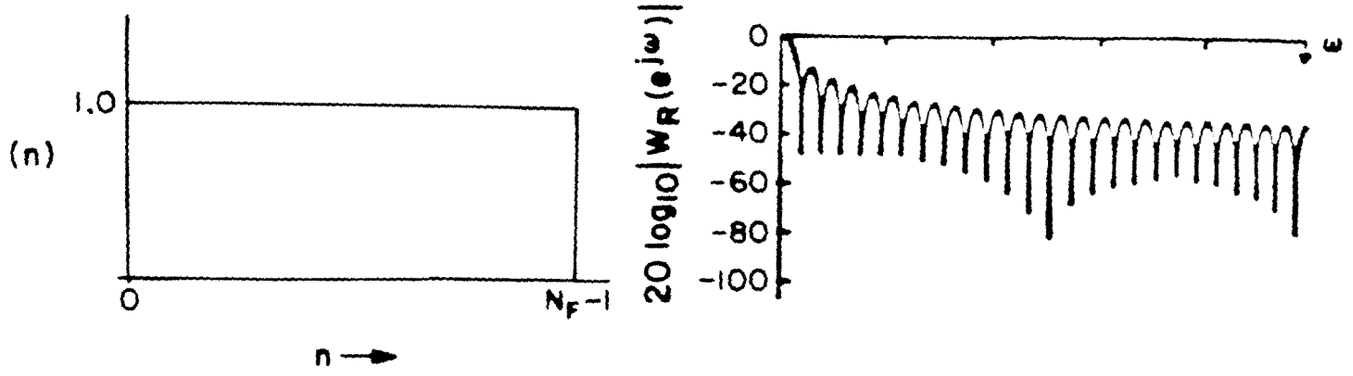


Fig. 2.3(a) Rectangular window and its amplitude spectrum

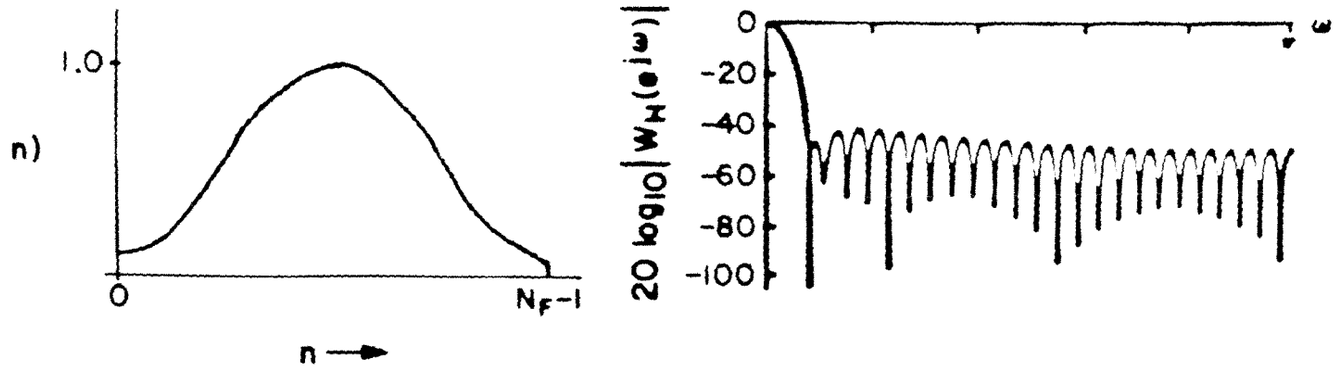


Fig. 2.3(b) Hamming window and its amplitude spectrum

main lobe and whose sidelobes are well below the main lobe level. In our examples of Fig. 2.3, we observe that the first sidelobe is 13 dB below the main lobe level for the rectangular window and 44 dB for the Hamming window. Thus the large sidelobes in the case of the rectangular window offset the benefits of the narrow main lobe. For this reason the rectangular window is seldom used. In the case of the Hamming window the low sidelobes ensure little "spectral leakage". The frequency resolution, though not as high as in the case of the rectangular window, is adequate for speech spectrum analysis. This is why the Hamming window is often used in speech.

We now consider the choice of the window size or length. As pointed out earlier, the width of the main lobe of the amplitude spectra of the window is inversely related to the window length. So a large window size implies higher frequency resolution but lower time resolution. Moreover, when the window becomes large, the windowed speech signal can no longer be considered stationary. A compromise has to be made. Typically, window sizes of 20-35 ms are chosen. This corresponds to the ranges 150-263 and 200-350 for N_F at 7.5 and 10 KHz sampling rates, respectively.

The *short-time power spectrum* is defined as

$$S_n(\omega) = |X_n(e^{j\omega})|^2.$$

The *short-time amplitude spectrum* is defined as

$$S'_n(\omega) = |X_n(e^{j\omega})|.$$

The short-time Fourier transform and hence the short-time power and amplitude spectrum can be efficiently computed at equally spaced frequencies using FFT (see Section 6.3.1 of [3]). The short-time power spectrum of a speech segment is characterized by a *spectral envelope* and a *spectral fine structure.* The spectral envelope is

determined by the frequency response of the vocal tract and also by the spectrum of the glottal pulse for voiced speech (Chapter 6 [3]). The perceptually relevant features of the spectral envelope are its peaks which correspond to the formants or vocal tract resonances. In the case of nasals, the spectral envelope is also characterized by valleys, which correspond to the antiresonances that arise due to the coupling of the oral and the nasal cavity (Section 3.1.2d [3]). The spectral fine structure, on the other hand, is largely due to the excitation signal. For voiced speech, this excitation is nearly quasi-periodic and therefore the spectrum has a "comb-like" structure.

To understand how such a fine structure comes about, we consider a very simplified model of voiced speech. Assume that the speech signal $s(t)$ as seen through the window $w(t)$ can be modeled as the output of a linear filter with impulse response $h(t)$, when the input $e(t)$ is a periodic train of delta impulses with period t_p. We will refer to t_p as the *pitch period* and $f_p = 1/t_p$ as the *fundamental frequency*.

$$e(t) = \sum_{n=-\infty}^{\infty} \delta(t - nt_p)$$

$$s(t) = [h(t) * e(t)]w(t)$$

where * denotes convolution. Denoting the Fourier transforms of $e(t)$, $s(t)$, $w(t)$ and $h(t)$ by $E(f)$, $S(f)$, $W(f)$ and $H(f)$, respectively, we have

$$E(f) = \frac{1}{t_p} \sum_{n=-\infty}^{\infty} \delta(f - nf_p)$$

$$\begin{aligned} S(f) &= [H(f)E(f)] * W(f) \\ &= \frac{1}{t_p} \sum_{n=-\infty}^{\infty} H(nf_p)W(f - nf_p). \end{aligned}$$

For each n in the summation, we have the window transform $W(f)$ centred at nf_p and weighted by $H(nf_p)$, the value of the linear filter transfer function at nf_p. The

Fourier transform of the modeled speech signal $S(f)$ is the superposition of all such weighted and shifted $W(f)$'s. Each of these weighted and shifted $W(f)$'s is referred to as a *pitch harmonic.* If the window transform $W(f)$ has low sidelobes and a mainlobe whose width is comparable to f_p, which is often the case in practice, then the speech transform $S(f)$ will have a "comb-like" like structure. Thus, using a very simplifed model for speech production, we can explain how the envelope and the fine structure of the speech spectrum arise. The power spectral density of a 25.6ms Hamming-windowed speech segment (10 KHz sampling rate, $N_F = 256$) evaluated using a 512 point FFT is shown in Fig. 2.4.

2.3 Basic idea behind the quantile vocoder

Consider the power spectral density and the cumulative power spectral density of a speech segment as sketched in Fig. 2.5. Because of the integration or summation effect, the finer details of the speech spectrum are somewhat smoothed out but the features corresponding to the spectral envelope are still prominent. The peaks of the spectral envelope correspond to the steep portions of the cumulative spectral density. We can thus efficiently encode the perceptually relevant features of the spectral envelope by choosing quantile orders which are spread across these steep portions of the cumulative spectral density. The corresponding quantiles are now clustered near the formant locations. Fig. 2.5 illustrates this effect. This is the basic idea behind the quantile vocoder.

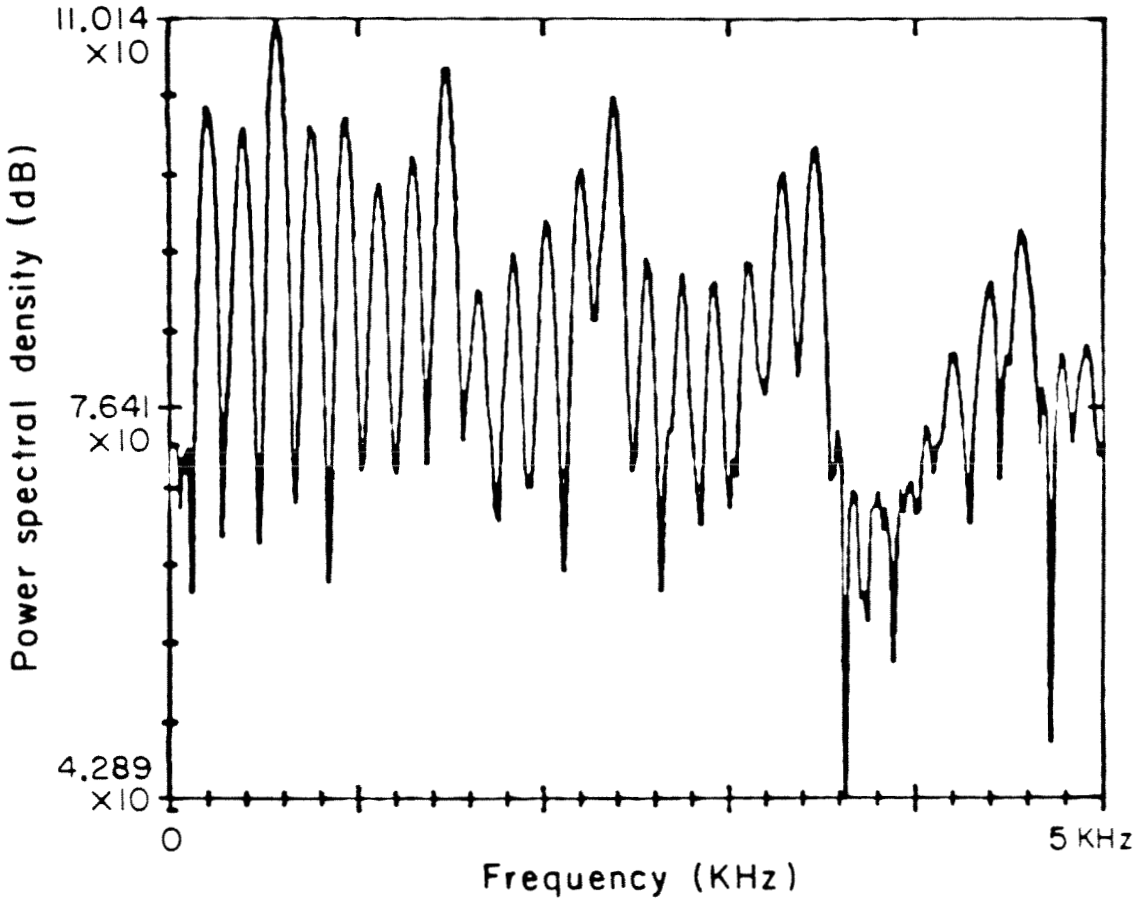


Fig. 2.4 An example of a short-time power spectral density

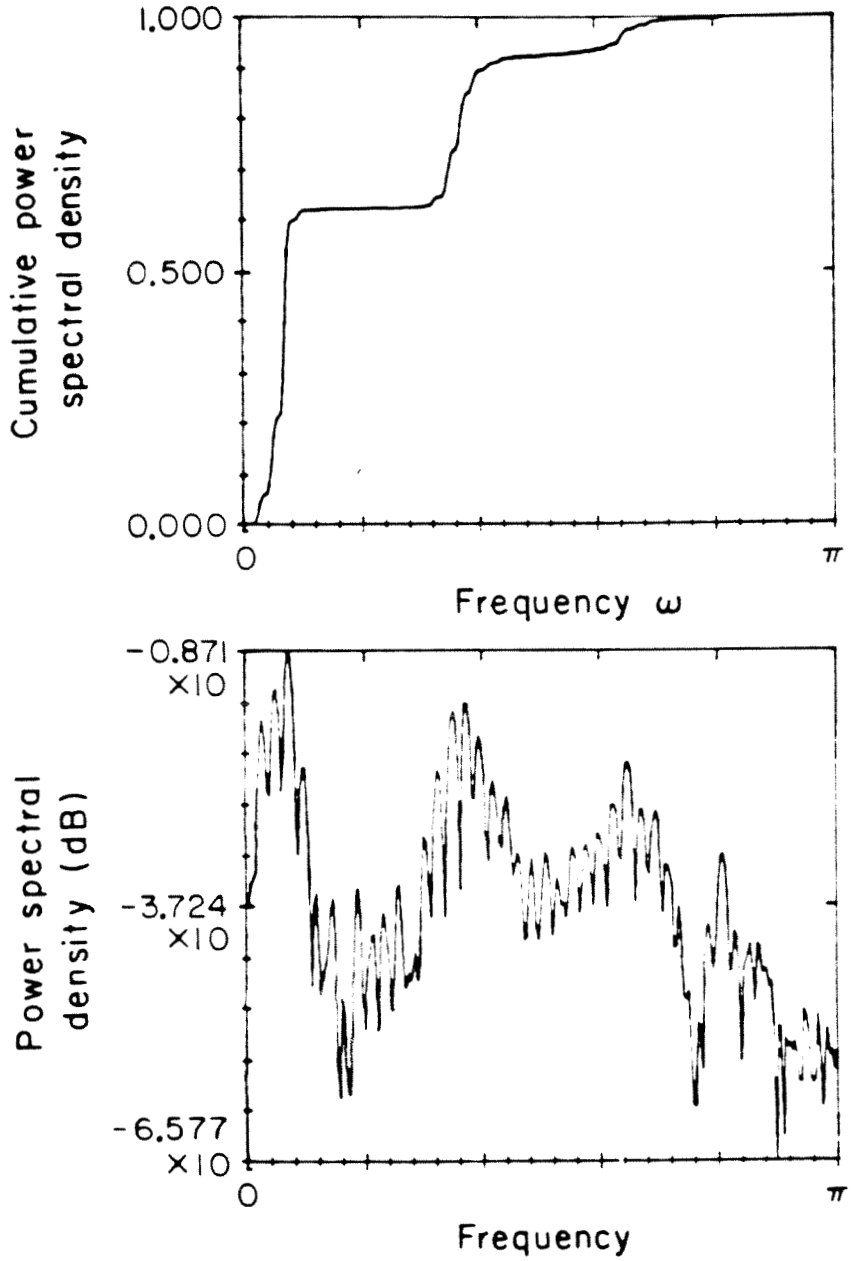


Fig. 2.5 Encoding of spectral envelope using quantiles

CHAPTER 3

CHOICE OF QUANTILES

In this chapter we will describe an algorithm for choosing a set of quantiles to encode the spectral envelope of a speech segment. The quantiles that are chosen must convey information about the shape of the spectral envelope at all frequencies. We begin by understanding the problems that arise while choosing such a set of quantiles. We then discuss methods to overcome these problems. Finally, the algorithm, which incorporates these methods, is presented.

3.1 Problems that arise while choosing quantiles

Consider Fig. 3.1. In this figure, the power spectral density and the cumulative power spectral density of a speech segment are sketched. The power spectral density has at least three distinct peaks corresponding to the first three formants. We note that the power spectral density at either of the first two formants is several dB above the power spectral density at the third formant frequency. As a consequence, only the steep slopes corresponding to the first two formants are prominent in the cumulative power spectral density. So if we were to choose q quantiles corresponding to equally spaced quantile orders n/q $(1 \leq n \leq q)$, then, unless q is very large, the quantiles would convey no information about the shape of the spectral envelope near the third or higher formants. It is clear that in order to encode the information about the spectral envelope at all frequencies, we need a more sophisticated scheme for choosing quantiles.

Next, let us take a closer look at the power spectral density near the first formant. We notice that the pitch harmonic with the maximum amplitude is several

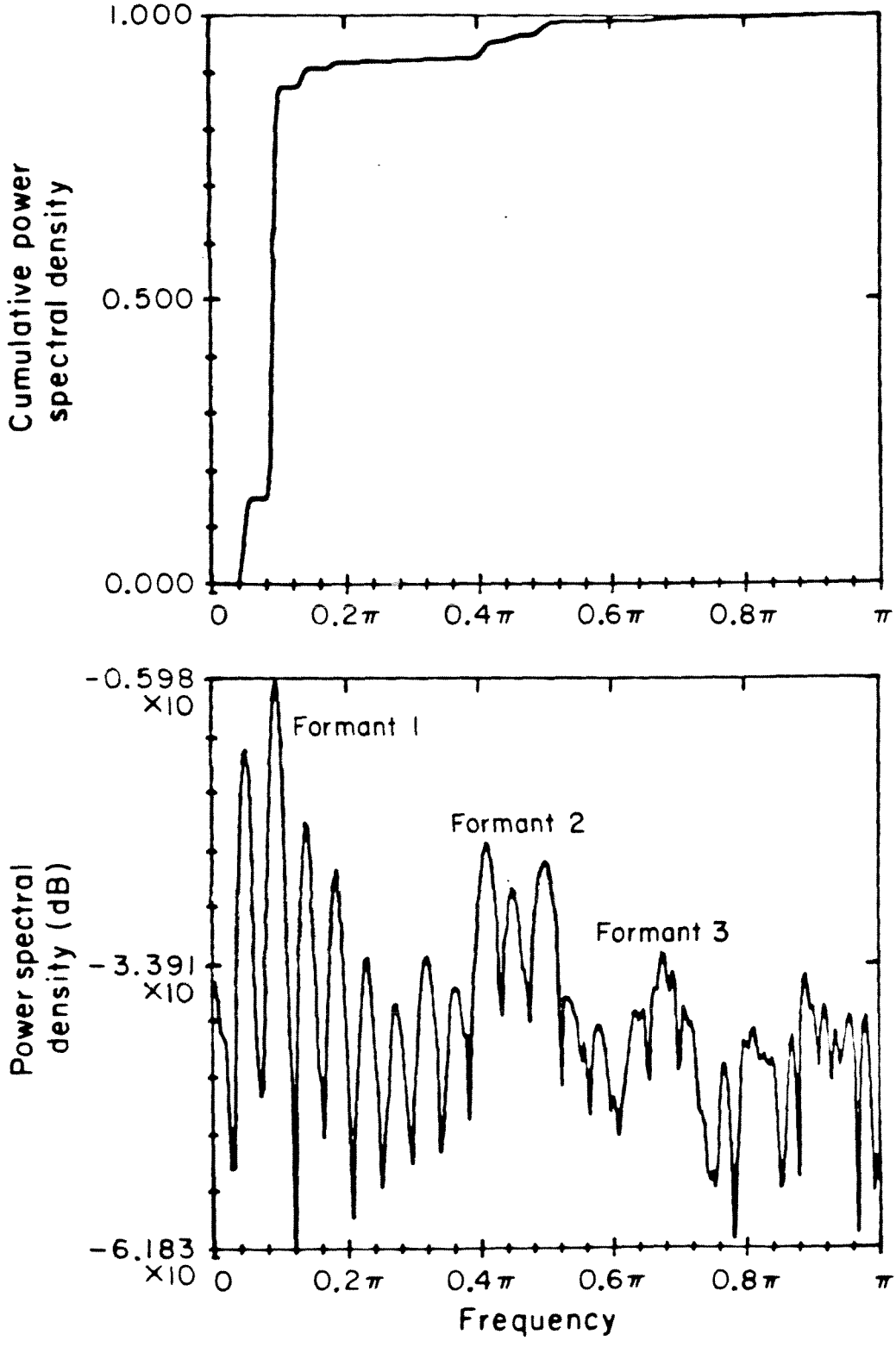


Fig. 3.1 Illustration of problems that arise in selecting quantiles

dB above the adjacent pitch harmonics. The first steep slope of the cumulative power spectral density contains information mostly about this pitch harmonic rather than the shape of the spectral envelope near the first formant. So again if we choose q quantiles corresponding to equally spaced quantile orders, then unless q is very large, the quantiles would convey little information about the shape of the spectral envelope near the first formant.

Thus there are two problems that we face. One is due to the power spectral density at the lower formants' being several dB above the power spectral density at the higher formants. We will refer to this as the overall dynamic range problem. The second is due to the pitch harmonic closest to the formant location's being several dB above the adjacent pitch harmonics. We will refer to this as the local dynamic range problem. This problem can be very severe if the formant has a narrow bandwidth and if the fundamental frequency for the speech segment is large. Finally we must bear in mind that not all quantile orders are possible, since we are dealing with a discrete spectrum.

3.2 Methods to overcome these problems

The overall dynamic range of the short-time power spectrum can be reduced by preemphasizing the input speech. The preemphasis filter that has been used is the second-order filter whose transfer function is

$$H_p(z) = 3.344 \frac{1 - 1.39z^{-1} + 0.52z^{-2}}{1 - 0.42z^{-1} - 0.13z^{-2}}.$$

This preemphasis filter was first suggested by Wong, *et al* in [43]. It approximates an ideal frequency response of unity gain from 0 to 0.1π and a 6 dB/octave slope from 0.1π to 0.8π. The frequency response of the preemphasis filter is shown in Fig. 3.2.

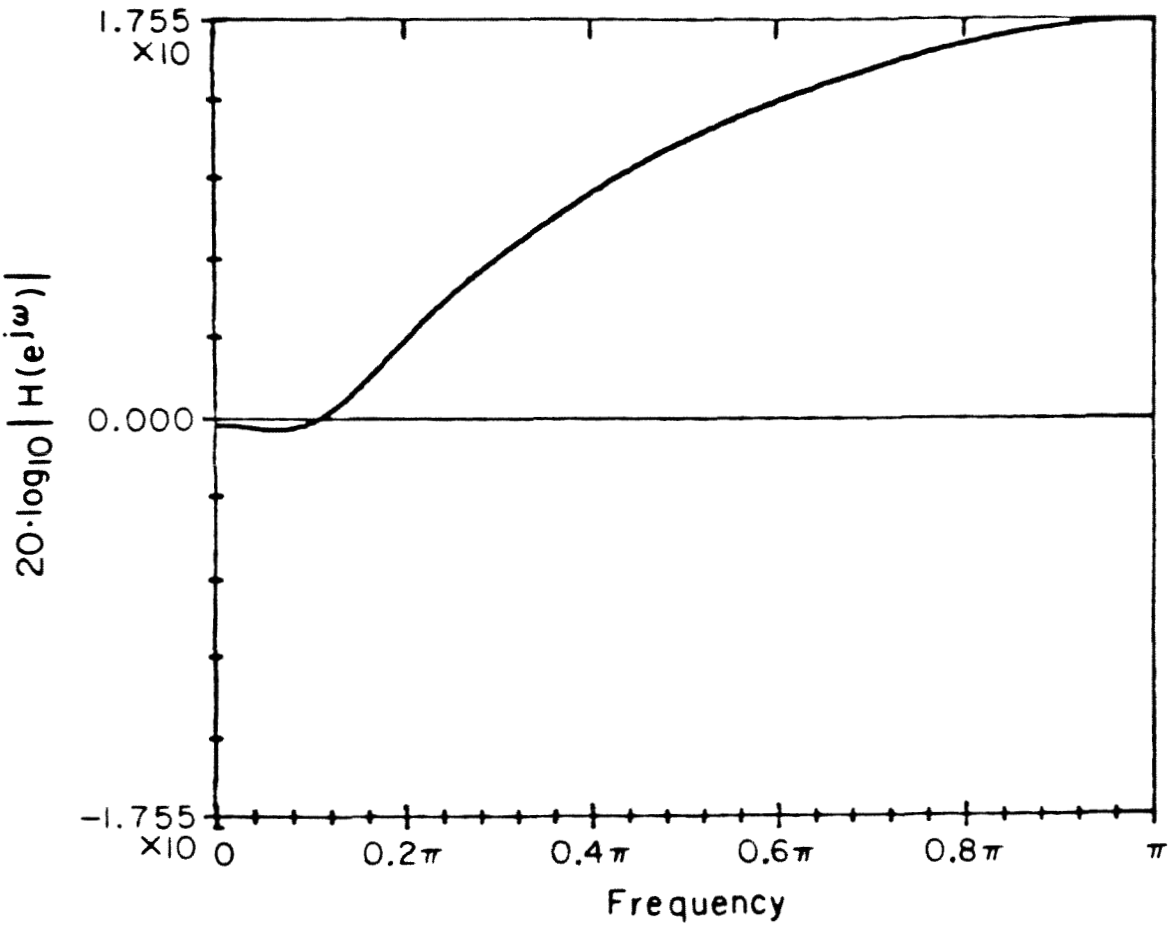


Fig. 3.2 Frequency response of preemphasis filter

For voiced speech frames, one can physically interpret the effect of preemphasis as follows. The shape of the spectral envelope is due to the frequency shaping of the vocal tract, the radiation at the lips and the shape of the glottal pulse. The combined effect of the shape of the glottal pulse and the radiation of the lips, under some simplifying conditions, is to cause a 6 db/octave drop in the spectrum of the vocal tract transfer function (Chapter 1 [16]). Thus the preemphasis of speech can be thought of as compensating for this 6 dB/octave drop so that the spectral envelope of the preemphasized spectrum is mostly due to the vocal tract alone.

The preemphasis helps overcome the overall dynamic range problem to some extent. In order to further ensure that the quantiles convey information about the spectral envelope at all frequencies we use the following approach. The entire frequency range (i.e., from zero to half the sampling frequency) is split into distinct sub-bands. In each sub-band, a fixed number of quantiles is chosen. This guarantees that the perceptually relevant features in each sub-band will be encoded.

We now address the local dynamic range problem. One way to overcome this is to choose quantiles on the basis of the amplitude spectral density rather than the power spectral density. The amplitude spectral density has only half the overall dynamic range as the power spectral density. In addition, the difference (in dB) between the levels of the adjacent pitch harmonics is halved. Thus the steep portions of the cumulative amplitude spectral density can be expected to contain information regarding the shape of the spectral envelope near the formants and not just the pitch harmonic closest to the formant.

3.3 An algorithm to choose a set of quantiles

Each speech segment is preemphasized and multiplied by a Hamming window.

In our work, each speech segment or frame is taken to be 256 samples of speech at all bit rates. For the 16 and 24 Kbits/s the sampling rate f_S is 10 KHz and so each frame is 25.6 ms of speech. For the 4.8 and 9.6 Kbits/s the sampling rate f_S is 7.5 KHz and so each frame is 34.13 ms of speech. We then compute the short-time amplitude and power spectral density as well as the cumulative amplitude and power spectrum at N equally spaced frequencies over the unit circle. Both the cumulative amplitude and cumulative power spectrum are scaled so that their values at frequency $f_o = f_S/2$ is 1.0. In our work N is taken to be 512 so that the spectrum is evaluated at multiples of $f_N = f_S/N$ Hz between 0 and f_o (257 discrete frequencies). For the 4.8 and 9.6 Kbits/s, $f_N = 14.65$ Hz and $f_o = 3.75$ KHz and for the 16 and 24 Kbits/s, $f_N = 19.53$ Hz and $f_o = 5$ KHz.

Let us first outline the algorithm for medium and high bit rate (e.g., 9.6, 16 and 24 Kbits/s) vocoders. The frequency range is split into R distinct sub-bands. The sub-bands are chosen in the following way. We first locate the frequencies F_1, F_2, ... F_R corresponding to the R most prominent peaks of the power spectrum. The R sub-bands are then chosen to be $\left(0\ ,\ \frac{F_1+F_2}{2}\right]$, $\left(\frac{F_1+F_2}{2}\ ,\ \frac{F_2+F_3}{2}\right]$, ... $\left(\frac{F_{R-1}+F_R}{2}\ ,\ f_o\right)$. Thus, each sub-band contains one of the R most prominent peaks of the power spectral density. The frequencies 0 and f_o are excluded from the sub-bands. In our implementation $R = 3$ for the 9.6 and 16 Kbits/s and $R = 4$ for 24 Kbits/s.

In order to illustrate how the quantiles are chosen in each sub-band, let us suppose that we wish to choose q_i quantiles from sub-band i. Let us also suppose that the value of the cumulative amplitude spectrum at the first discrete frequency f_{i_1} in sub-band i is E_{i_1} and at the last discrete frequency f_{i_2} in sub-band i is E_{i_2}. Then the j^{th} $(1 \leq j \leq q_i)$ quantile in the i^{th} sub-band is that frequency $f_j^{(i)}$ at which

the value of the cumulative amplitude spectrum is closest to $E_{i_1} + j(E_{i_2} - E_{i_1})/(1 + q_i)$. In addition to choosing the quantiles in each sub-band, we also include both 0 and f_o as quantiles. Having chosen the quantiles in this fashion, the corresponding quantile orders are now chosen using the cumulative power spectral density. The entire algorithm for choosing the quantiles and the corresponding quantile orders is described in Fig. 3.3, using a flowchart.

We now turn our attention to low bit rate (e.g., 4.8 Kbits/s) vocoders. For such vocoders there are very few bits per frame. One approach is to use the same algorithm as for medium and high bit rate vocoders but to have fewer quantiles and quantile orders. A second alternative is to transmit quantiles corresponding to fixed quantile orders (based on the cumulative amplitude spectrum). This way one need not transmit the quantile orders. In our experience the second alternative produces better results.

In our implementation the quantiles for the low bit rate vocoders are chosen as follows. Let us suppose that q quantiles are to be chosen. Note that the value of the cumulative amplitude spectral density due to scaling, at frequency f_o is $E_q = 1.0$. The fixed quantile orders (based on the cumulative amplitude spectrum) are taken to be $\frac{E_q}{q}$, $\frac{2E_q}{q}$, ... E_q. The j^{th} quantile is then obtained as that frequency f_j at which the value of the cumulative amplitude spectrum is closest to $\frac{jE_q}{q}$.

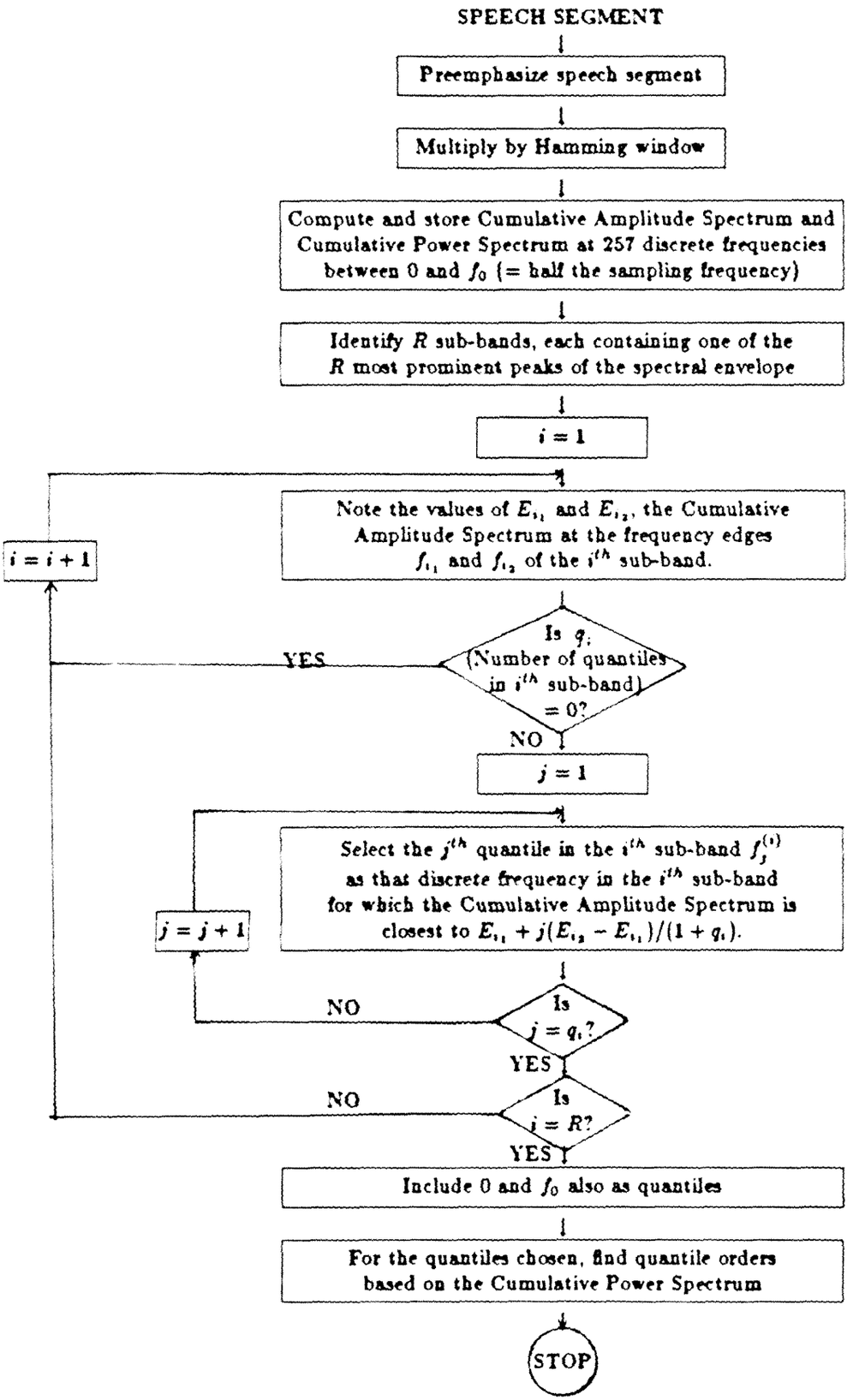


Fig. 3.3. Flowchart of Algorithm for selecting quantiles

CHAPTER 4

QUANTILE DECODING ALGORITHM

In the previous chapter, we described an algorithm for choosing a set of quantiles to represent the spectral envelope of a speech segment. In this chapter we describe how to estimate the spectral envelope of the speech segment from the quantiles and quantile orders.

4.1 Flat Spectral Density Approximation

The first step in the quantile decoding algorithm is to set up a flat spectral density approximation of the spectral envelope. Let us first consider medium and high bit rate vocoders. If the transmitted quantiles (expressed in radians) are θ_o $(= 0)$, θ_1, θ_2, ..., θ_q $(= \pi)$ and the corresponding quantile orders (based on the cumulative power spectral density) are E_o, E_1, E_2, ..., E_q $(= 1.0)$, then the flat spectral density approximation is defined as

$$\begin{aligned} S_o(\omega) &= E_o && \omega = \theta_o \\ &= \frac{2\pi(E_l - E_{l-1})}{N(\theta_l - \theta_{l-1})} && \theta_{l-1} < \omega \leq \theta_l,\ 1 \leq l \leq q. \end{aligned} \tag{1a}$$

It is interesting to note that the flat spectral density approximation of the spectral envelope is also the constrained maximum entropy solution to the decoding problem, as elaborated next.

A discrete normal stationary random process with spectral density $S(\omega)$ can be shown to have an entropy rate H_S ([44]), where H_S is given by

$$H_S = \log(\sqrt{2\pi e}) + \frac{1}{2\pi}\int_0^{\pi} \log S(\omega)\, d\omega.$$

The power spectral density of a normal stationary random process which has quantiles θ_o $(= 0)$, θ_1, ..., θ_q $(= \pi)$ corresponding to quantile orders E_o, E_1, ..., E_q and

for which the entropy rate H_S is a maximum is obtained by solving the following problem:

Maximize $\int_0^\pi \log S(\omega)\, d\omega$

subject to the constraints

$$S(\theta_o) = E_o$$

$$\int_0^\pi S(\omega)[U(\omega - \theta_{i-1}) - U(\omega - \theta_i)]\, d\omega = \frac{2\pi(E_i - E_{i-1})}{N} \qquad 1 \le i \le q$$

where $U(\omega)$ is the step function defined as

$$U(\omega) = 1 \qquad \omega > 0$$

$$= 0 \qquad \omega \le 0.$$

This is an elementary variational calculus problem (see Section 7.3 of [36] for methods to solve such problems) and its solution is given by equation (1a), i.e., the flat spectral density approximation.

For low bit rate vocoders, the quantile orders are based on the cumulative amplitude spectrum. The flat spectral density approximation $S_o(\omega)$ is then given by

$$S_o(\omega) = E_o^2 \qquad \omega = \theta_o$$

$$= \left[\frac{2\pi(E_l - E_{l-1})}{N(\theta_l - \theta_{l-1})}\right]^2 \qquad \theta_{l-1} < \omega \le \theta_l,\ 1 \le l \le q. \tag{1b}$$

Again it can easily be shown that this is the maximum entropy solution to the decoding problem with appropriate constraints. The maximum entropy solution is obtained in this case by solving the following problem:

Maximize $\int_0^\pi \log S(\omega)\, d\omega$

subject to the constraints

$$S^{0.5}(\theta_o) = E_o$$

$$\int_0^{\pi} S^{0.5}(\omega)[U(\omega - \theta_{i-1}) - U(\omega - \theta_i)]\, d\omega = \frac{2\pi(E_i - E_{i-1})}{N} \qquad 1 \leq i \leq q.$$

In our implementation, E_l $(1 \leq l \leq q)$ is fixed and is given by $\frac{l}{q}$. The value of E_o, however, is not known at the receiver since it is not transmitted. We find that E_o can be set equal to the square root of the flat spectral density approximation in the frequency range $0 < \omega \leq \theta_1$ without seriously affecting the final solution. Thus

$$E_o = \frac{2\pi(E_1 - E_o)}{N\theta_1}$$

and therefore

$$E_o = \frac{2\pi E_1}{N\theta_1 + 2\pi}.$$

The flat spectral density approximation has nearly the same overall shape as the spectral envelope of the speech segment. However, it needs to be smoothed. For the purposes of determining the parameters of some excitation models, such as multi-pulse excitation model, it is necessary to express the spectral envelope as the power spectrum of either an AR model (autoregressive or all-pole model) or an ARMA model (autoregressive moving average or pole-zero model). So we will now describe an algorithm which smoothens the flat spectral density approximation by approximating it in turn with the power spectrum of an AR or ARMA model.

4.2 Autoregressive smoothing of flat spectral density approximation

Consider an autoregressive model $H(z)$ of order M. Thus,

$$H(z) = \frac{1}{a_o + a_1 z^{-1} + \ldots + a_M z^{-M}} = \frac{1}{A(z)}. \qquad (2)$$

The polynomial $A(z)$ is referred to as the inverse filter. Because of stability considerations, $A(z)$ is assumed to be minimum-phase; i.e., all its roots lie strictly within the unit circle. The power spectrum of the AR model is given by

$$\begin{aligned} |H(e^{j\omega})|^2 &= \frac{1}{|A(e^{j\omega})|^2} \\ &= \frac{1}{c_o + c_1 \cos\omega + \ldots + c_M \cos M\omega} \\ &= \frac{1}{C(\omega)}. \end{aligned} \tag{3}$$

Thus, the power spectrum of the autoregressive model can be expressed as the inverse of a positive definite trignometric polynomial $C(\omega)$, i.e., a trignometric polynomial $C(\omega)$ which is positive for all ω in the range $[0, 2\pi)$. It is clear that given a minimum-phase polynomial $A(e^{j\omega})$ one can determine uniquely the positive definite polynomial $C(\omega)$. The converse is also true. The details of this are discussed in a later section in this chapter. The important thing to bear in mind at this point is that there exists a one-to-one correspondence between the coefficients $\{a_i\}$ of $A(z)$ and the coefficients $\{c_i\}$ of $C(\omega)$.

One approach to determining the parameters of the AR model whose power spectrum fits the flat spectral density approximation $S_o(\omega)$ is to minimize the weighted mean-square error

$$E = \sum_{\substack{k=0 \\ \omega_k = 2\pi k/N}}^{N/2} \Big(S_o(\omega_k) C(\omega_k) - 1 \Big)^2 W(\omega_k) \tag{4}$$

subject to the constraint

$$C(\omega) > 0 \qquad\qquad 0 \le \omega \le \pi.$$

Here $W(\omega)$ is a positive weighting function of ω. If there were no constraint, then the problem would be a standard least-squares problem. If we incorporate the

constraint then we have a least-squares problem with linear inequality constraints. Algorithms to solve such problems have been described in [45]. These algorithms make use of Kuhn-Tucker theorem ([45], [46]) in optimization theory. They are iterative in nature though it can be proved that they converge in a finite number of steps. For speech coding applications, these algorithms are much too expensive both from a computational as well as the storage requirement point of view. Fortunately, the $C(\omega)$ that we obtain by minimizing E without any constraint turns out to be positive definite most of the time. So our approach would be to ignore the constraint, which is $C(\omega) > 0$ for all $0 \leq \omega \leq \pi$, and simply to find the $C(\omega)$ which minimizes E. If the estimated $C(\omega)$ is not positive definite, then we modify it using a spectral correction algorithm so that the modified $C(\omega)$ is positive definite. Such a modified $C(\omega)$ is only a sub-optimal solution but we are willing to accept this because:

1. If we use the spectral correction algorithm described later in this chapter, the sub-optimal solution turns out to be reasonably satisfactory.

2. The need to settle for a sub-optimal solution arises very infrequently.

3. Such a sub-optimal solution can be obtained without too many computations.

Minimization of E leads to a set of linear equations which can be described by

$$\mathbf{Ac} = \mathbf{b} \tag{5}$$

where

$$\mathbf{A} = [a_{ij}]_{(M+1)\times(M+1)} \quad ; \tag{5a}$$

$$a_{ij} = \sum_{\substack{k=0 \\ \omega_k = 2\pi k/N}}^{N/2} S_o^2(\omega_k) W(\omega_k) \cos i\omega_k \cos j\omega_k \qquad 0 \leq i \leq M,\ 0 \leq j \leq M\,;$$

$$\mathbf{c} = [c_o\, c_1 \ldots c_M]^T \quad ; \tag{5b}$$

$$\mathbf{b} = [b_o\, b_1 \ldots b_M]^T \quad ; \tag{5c}$$

$$b_i = \sum_{\substack{k=0 \\ \omega_k = 2\pi k/N}}^{N/2} S_o(\omega_k) W(\omega_k) \cos i\omega_k \qquad 0 \le i \le M \, .$$

The matrix **A** has several properties which can be exploited when solving for the vector **c**.

Property 1 : **A** is symmetric.

Proof: $a_{ij} = \sum_{\substack{k=0 \\ \omega_k = 2\pi k/N}}^{N/2} S_o^2(\omega_k) W(\omega_k) \cos i\omega_k \cos j\omega_k = a_{ji}$

Property 2 : **A** is non-negative definite.

Proof: Consider any arbitary vector **x**. Then

$$\begin{aligned} \mathbf{x}^T \mathbf{A} \mathbf{x} &= \sum_{j=0}^{M} \sum_{i=0}^{M} x_i x_j \sum_{\substack{k=0 \\ \omega_k = 2\pi k/N}}^{N/2} S_o^2(\omega_k) W(\omega_k) \cos i\omega_k \cos j\omega_k \\ &= \sum_{\substack{k=0 \\ \omega_k = 2\pi k/N}}^{N/2} S_o^2(\omega_k) W(\omega_k) \left(\sum_{i=0}^{M} x_i \cos i\omega_k \right)^2 \\ &\ge 0 \qquad \text{(since } S_o^2(\omega_k) W(\omega_k) \ge 0 \; \forall \; k), \end{aligned}$$

so **A** is non-negative definite.

Property 3: **A** can be expressed as the sum of a symmetric Toeplitz matrix **T** and a Hankel matrix **H**. (A Toeplitz matrix is one whose $(i,j)^{th}$ element depends only on $i-j$. A Hankel matrix is one whose $(i,j)^{th}$ element depends only on $i+j$. Note that a Hankel matrix is symmetric by definition.)

Proof:

$$\begin{aligned} a_{ij} &= \sum_{\substack{k=0 \\ \omega_k = 2\pi k/N}}^{N/2} S_o^2(\omega_k) W(\omega_k) \cos i\omega_k \cos j\omega_k \\ &= \frac{1}{2} \sum_{\substack{k=0 \\ \omega_k = 2\pi k/N}}^{N/2} S_o^2(\omega_k) W(\omega_k) \cos(i+j)\omega_k + \frac{1}{2} \sum_{\substack{k=0 \\ \omega_k = 2\pi k/N}}^{N/2} S_o^2(\omega_k) W(\omega_k) \cos(i-j)\omega_k \\ &= h_{i+j} + t_{i-j} \end{aligned}$$

Thus, $\mathbf{A} = \mathbf{T} + \mathbf{H}$ where

$$\mathbf{T} = [t_{ij}]_{(M+1)\times(M+1)} = [t_{i-j}]_{(M+1)\times(M+1)} = [t_{j-i}]_{(M+1)\times(M+1)}$$

$$\Rightarrow \textit{ a symmetric Toeplitz matrix}$$

$$\mathbf{H} = [h_{ij}]_{(M+1)\times(M+1)} = [h_{i+j}]_{(M+1)\times(M+1)}$$

$$\Rightarrow \textit{ a Hankel matrix.}$$

Using only the first two properties, one can solve for $\mathbf{c}$ by Choleski decomposition ([47]). This involves $O(M^3)$ arithmetic operations. But Merchant and Parks ([34]) have shown that any matrix, such as $\mathbf{A}$, which can be expressed as a sum of a Toeplitz matrix and a Hankel matrix, can be solved by a block Levinson algorithm which requires $O(M^2)$ arithmetic operations. Block Levinson-type algorithms arise in many applications of signal processing and have therefore been investigated extensively. For such algorithms, the number of computations as well as storage requirement is significantly less than other general matrix inversion algorithms such as Gaussian elimination, etc. for large matrices. They also appear to be easily implementable using VLSI. For this reason, we have chosen to solve for $\mathbf{c}$ using the Merchant-Parks approach. A brief summary of their technique is given in Appendix A.

The elements of the Toeplitz matrix $\mathbf{T}$ and the Hankel matrix $\mathbf{H}$ can be computed directly from the quantiles and quantile orders for the special case when $W(\omega)$ is of the form $S_o^{\nu}(\omega)$, where ν is some positive number.

$$h_i = \frac{1}{2} \sum_{\substack{k=0 \\ \omega_k = 2\pi k/N}}^{N/2} S_o^{2+\nu}(\omega_k) \cos i\omega_k$$

$$= \frac{1}{2}\left[E_o^{\Delta(2+\nu)} + \sum_{l=1}^{q} \left(\frac{2\pi(E_l - E_{l-1})}{N(\theta_l - \theta_{l-1})} \right)^{\Delta(2+\nu)} \sum_{k=\frac{N}{2\pi}\theta_{l-1}+1}^{\frac{N}{2\pi}\theta_l} \cos 2\pi ik/N \right]$$

$$= \frac{1}{2}\Big[E_o^{\Delta(2+\nu)} + \sum_{l=1}^{q}\Big(\frac{2\pi(E_l - E_{l-1})}{N(\theta_l - \theta_{l-1})}\Big)^{\Delta(2+\nu)} \frac{\sin\frac{(\theta_l-\theta_{l-1})i}{2}}{\sin\frac{\pi i}{N}} \cdot \cos\Big(\frac{(\theta_l+\theta_{l-1})i}{2} + \frac{\pi i}{N}\Big)\Big] \qquad (6)$$

for all $0 \leq i \leq 2M$. The value of Δ in the above expression is 1 for medium and high bit rate vocoders and 2 for low bit rate vocoders. We also note that

$$t_i = t_{-i} = h_i \qquad 0 \leq i \leq M. \qquad (7)$$

The coefficients b_i of the vector $\mathbf{b}$ can also be evaluated as

$$\begin{aligned} b_i &= \sum_{\substack{k=0 \\ \omega_k = 2\pi k/N}}^{N/2} S_o^{1+\nu}(\omega_k)\cos i\omega_k \\ &= E_o^{\Delta(1+\nu)} + \sum_{l=1}^{q}\Big(\frac{2\pi(E_l - E_{l-1})}{N(\theta_l - \theta_{l-1})}\Big)^{\Delta(1+\nu)} \sum_{k=\frac{N}{2\pi}\theta_{l-1}+1}^{\frac{N}{2\pi}\theta_l} \cos 2\pi ik/N \\ &= E_o^{\Delta(1+\nu)} + \sum_{l=1}^{q}\Big(\frac{2\pi(E_l - E_{l-1})}{N(\theta_l - \theta_{l-1})}\Big)^{\Delta(1+\nu)} \frac{\sin\frac{(\theta_l-\theta_{l-1})i}{2}}{\sin\frac{\pi i}{N}} \cos\Big(\frac{(\theta_l+\theta_{l-1})i}{2} + \frac{\pi i}{N}\Big) \qquad (8) \end{aligned}$$

for all $0 \leq i \leq M$. Note that in order to set up the system of equations $\mathbf{Ac} = \mathbf{b}$, we need only compute h_i $(0 \leq i \leq 2M)$ and b_i $(0 \leq i \leq M)$.

4.3 Spectral Correction Algorithm

As explained in the previous section, the estimated $C(\omega)$ is not guaranteed to be positive definite, though in practice this is the case most of the time. So if the estimated $C(\omega)$ is not positive definite, then it must be modified. In this section we will describe such a modification.

The analytic continuation of $C(\omega)$ is given by

$$C(z) = c_o + \frac{c_1}{2}(z + z^{-1}) + \ldots + \frac{c_M}{2}(z^M + z^{-M}).$$

The simplest modification that ensures the positive definiteness of $C(z)$ is to add a small positive constant, enough to ensure that $C(\omega) > 0$ for all ω. Unfortunately, this alters the locations of all the roots of $C(z)$, which correspond to the formant locations and formant bandwidths. We would like to avoid this as far as possible.

Let us examine a situation when a symmetric sequence $C(z)$ becomes negative for some portions of the unit circle (i.e., $z = e^{j\omega}$). If any symmetric sequence $C(z)$ with real coefficients has a root at $z = re^{j\alpha}$ $(r < 1)$, then it must have roots at $re^{\pm j\alpha}$, $\frac{1}{r}e^{\pm j\alpha}$ as well. Thus if a symmetric sequence $C(z)$ has no roots on the unit circle, then it can be expressed as

$$
\begin{aligned}
C(z) &= \prod_i (1 - 2r_i z^{-1} \cos\,\alpha_i + r_i^2 z^{-2})(1 - 2r_i \cos\,\alpha_i z + r_i^2 z^2) \\
C(\omega) &= C(z = e^{j\omega}) \\
&= \prod_i |(1 - 2r_i \cos\,\alpha_i e^{-j\omega} + r_i^2 e^{-j2\omega})|^2 \\
&> 0 \qquad (\text{ since } r_i < 1 \;\forall\, i).
\end{aligned}
$$

Thus any symmetric sequence which has no roots on the unit circle is automatically positive definite. However, if $C(z)$ has any roots at all on the unit circle, then it cannot be positive definite. If all the roots on the unit circle are of even multiplicity, then $C(z)$ will be non-negative definite. If there are any zeros on the unit circle of odd multiplicity, then $C(z)$ will become negative for some portions of the unit circle.

The modification that we propose is the following. If the estimated symmetric sequence $C(z)$ is not positive definite, then we locate the roots of $C(z)$ on the unit circle and replace them by roots within and outside the unit circle so that the modified sequence $C^*(z)$ is positive definite. Consider the case (see Fig. 4.1(a)) when $C(\omega)$ becomes negative in the frequency range $(-\alpha, \alpha)$ where $0 \leq \alpha \leq \pi$.

Clearly, $C(z)$ has roots at $e^{\pm j\alpha}$ and so

$$C(z) = \overline{W}(z)\left(\frac{2\cos\alpha - z - z^{-1}}{2}\right) \tag{9a}$$

$$C(\omega) = \overline{W}(\omega)\Big(\cos\alpha - \cos\omega\Big). \tag{9b}$$

We want to replace the roots at $e^{\pm j\alpha}$ by roots at r and $\frac{1}{r}$ $(r < 1)$. So the modified $C^*(z)$ is then given by

$$C^*(z) = \overline{W}(z)\frac{(1 - rz^{-1})(1 - rz)}{2r} \tag{9c}$$

$$C^*(\omega) = \overline{W}(\omega)\Big(\frac{1}{2}(r + \frac{1}{r}) - \cos\omega\Big). \tag{9d}$$

We will refer to this case where the negative sign region includes $\omega = 0$ as case A.

If, as shown in Fig. 4.1(b), $C(\omega)$ is negative in the frequency range (α_1, α_2) where $0 \leq \alpha_1 \leq \alpha_2 \leq \pi$ (since $C(z)$ has real coefficients, this implies that $C(\omega)$ will be negative in the frequency range $(-\alpha_2, -\alpha_1)$ as well), then $C(z)$ can be expressed as

$$C(z) = \frac{1}{4}\,\overline{W}(z)(1 - 2\cos\alpha_1\, z^{-1} + z^{-2})(1 - 2\cos\alpha_2\, z + z^2) \tag{10a}$$

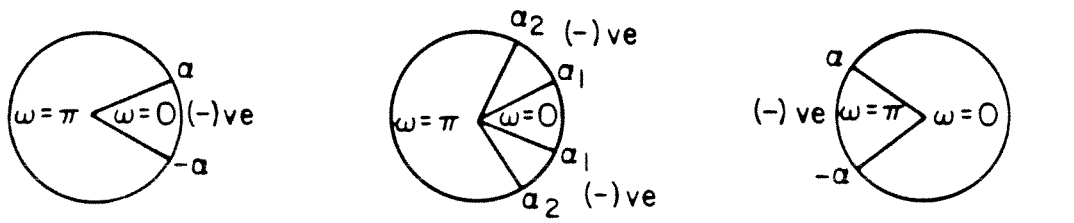
$$C(\omega) = \overline{W}(\omega)\Big(\cos^2\omega - (\cos\alpha_1 + \cos\alpha_2)\cos\omega + \cos\alpha_1\cos\alpha_2\Big). \tag{10b}$$

We replace the roots at $e^{\pm j\alpha_1}$, $e^{\pm j\alpha_2}$ by roots at $re^{\pm j\alpha}$, $\frac{1}{r}e^{\pm j\alpha}$ $(r < 1)$. The modified $C^*(z)$ would thus be

$$C^*(z) = \frac{1}{4r^2}\overline{W}(z)(1 - 2r\cos\alpha\, z^{-1} + r^2z^{-2})(1 - 2r\cos\alpha\, z + r^2z^2) \tag{10c}$$

$$C^*(\omega) = \overline{W}(\omega)\Big(\cos^2\omega - (r + \frac{1}{r})\cos\alpha\cos\omega + (1 + r^4 + 2r^2\cos 2\alpha)/4r^2\Big). \tag{10d}$$

We will refer to this case where the negative sign region neither includes $\omega = 0$ or $\omega = \pi$ as case B.



(a) CASE A

NEGATIVE SIGN REGION INCLUDES $\omega = 0$

(b) CASE B

NEGATIVE SIGN REGION INCLUDES NEITHER $\omega = 0$ NOR $\omega = \pi$

(c) CASE C

NEGATIVE SIGN REGION INCLUDES $\omega = \pi$

Fig. 4.1 Three cases of negative sign regions

If $C(\omega)$ is negative in the frequency range $(\alpha, -\alpha)$ as shown in Fig. 4.1(c), then $C(z)$ has roots at $e^{\pm j\alpha}$ and so

$$C(z) = \overline{W}(z)\left(\frac{z + z^{-1} - 2\cos\alpha}{2}\right) \tag{11a}$$

$$C(\omega) = \overline{W}(\omega)\Big(\cos\omega - \cos\alpha\Big). \tag{11b}$$

We replace the roots at $e^{\pm j\alpha}$ by roots at $-r$ and $-\frac{1}{r}$ $(r < 1)$. The modified $C^*(z)$ is then given by

$$C^*(z) = \overline{W}(z)\frac{(1 + rz^{-1})(1 + rz)}{2r} \tag{11c}$$

$$C^*(\omega) = \overline{W}(\omega)\Big(\frac{1}{2}(r + \frac{1}{r}) + \cos\omega\Big). \tag{11d}$$

We will refer to this case where the negative sign region includes $\omega = \pi$ as case C.

How do we choose the values of r in all three cases and the value of α in case B? We begin by considering case B. We would like to choose values of r and α so as to minimize the effect of the modification. One way to do this is to minimize

$$\overline{E} = \frac{1}{\pi}\int_0^{\pi}\Big(\overline{W}^{-1}(\omega)C^*(\omega) - \overline{W}^{-1}(\omega)C(\omega)\Big)^2\, d\omega. \tag{12}$$

Substituting for $C(\omega)$ and $C^*(\omega)$ from (10b) and (10d) we get

$$\begin{aligned}\overline{E} &= \frac{1}{\pi}\int_0^{\pi}(-a + b\cos\omega)^2\, d\omega \\ &= a^2 + \frac{b^2}{2}\end{aligned} \tag{13}$$

where

$$b = \cos\alpha_1 + \cos\alpha_2 - (r + \frac{1}{r})\cos\alpha \tag{13a}$$

$$a = 1 + \cos\alpha_1\cos\alpha_2 - \frac{1}{4}(r + \frac{1}{r})^2 - \cos^2\alpha. \tag{13b}$$

In Appendix B it is shown that for a given r, there exists a unique optimum $\alpha = \alpha^*$ which is obtained from

$$\cos\alpha^* = \left(\frac{-p+\sqrt{p^2+4l^3/27}}{2}\right)^{\frac{1}{3}} + \left(\frac{-p-\sqrt{p^2+4l^3/27}}{2}\right)^{\frac{1}{3}} \tag{14}$$

where

$$p = -\frac{1}{4}(r+\frac{1}{r})(\cos\alpha_1 + \cos\alpha_2) \tag{14a}$$

$$l = -\frac{1}{2}\left(2 + 2\cos\alpha_1\cos\alpha_2 - (r+\frac{1}{r})^2\right). \tag{14b}$$

We note that a, b and hence $\overline{E}$ are all functions of $r+\frac{1}{r}$, which has a broad minimum at $r = 1$. So the value of α^* or $\overline{E}(\alpha^*)$ are not very much affected by the exact value of r as long as it is close to 1. We can exploit this weak dependence of $\overline{E}$ on values of r close to 1 by choosing r which is physically more meaningful. The value of r determines the formant bandwidth. The closer r is to 1, the narrower the bandwidth. Normally for most speech spectra the larger the formant frequency, the larger is its bandwidth. An emprical relation between radial pole location r and angular pole location α^* that was developed in the context of very low bit rate formant vocoders [10], is

$$r = 0.982\ \exp(-0.056\alpha^*). \tag{15}$$

In our work, rather than solve for r and α^* from equations (14) and (15), we have chosen r according to the relation

$$r = 0.982\ \exp(-0.028(\alpha_1+\alpha_2)). \tag{16}$$

For case A and case C, we again define $\overline{E}$ as in (12). For case A, substituting for $C(\omega)$ and $C^*(\omega)$ from equations (9b) and (9d), we get

$$\overline{E} = \left(\frac{1}{2}(r+\frac{1}{r}) - \cos\alpha\right)^2. \tag{17}$$

For case C, substituting for $C(\omega)$ and $C^*(\omega)$ from equations (11b) and (11d), we get

$$\overline{E} = \left(\frac{1}{2} (r + \frac{1}{r}) + \cos \alpha \right)^2. \tag{18}$$

Clearly, in both cases $\overline{E}$ is an increasing function of $(r + \frac{1}{r})$ and attains its minimum value when $r + \frac{1}{r} = 2$ or $r = 1$. But again we exploit the weak dependence of $\overline{E}$ on values of r near 1 by assigning it a physically more meaningful value. So again we use equation (15) with $\alpha^* = 0$ for case A and $\alpha^* = \pi$ for case C. This gives $r = 0.982$ for case A and $r = 0.982e^{-0.056\pi} = 0.8236$ for case C.

We now turn our attention to the details of the implementation. If $C(z)$ is detected to be not positive definite (this detection is done by the spectral factorization algorithm, which is explained in the next section), then we determine the roots on the unit circle by computing $C(\omega)$ over a dense grid of equally spaced frequencies using an FFT algorithm; usually a 512 point FFT is adequate. But in general, there could be more than one negative sign region.

We will treat each negative sign region one by one. In the case A type situation, where $C(\omega)$ is negative in the frequency range $(-\alpha, \alpha)$, the modified $C^*(z)$ is given by

$$C^*(z) = C(z) \frac{(1 - rz^{-1})(1 - rz)}{r(2\cos \alpha - z - z^{-1})}. \tag{19}$$

The value of r here is 0.982. The coefficients of $C^*(z)$ are computed using polynomial division and multiplication routines. In the case B type situation, where $C(\omega)$ is negative in the frequency range (α_1, α_2) and $(-\alpha_2, -\alpha_1)$, the modified $C^*(z)$ is given by

$$C^*(z) = C(z) \frac{(1 - 2r\cos \alpha^* z^{-1} + r^2 z^{-2})(1 - 2r\cos \alpha^* z + r^2 z^2)}{r^2(1 - 2\cos \alpha_1 z^{-1} + z^{-2})(1 - 2\cos \alpha_2 z + z^2)} \tag{20}$$

where α^* and r are given by equations (14) and (15), respectively. In the case C type situation where $C(\omega)$ is negative in the frequency range $(\alpha, -\alpha)$, the modified

$C^*(z)$ is given by

$$C^*(z) = C(z)\,\frac{(1+rz^{-1})(1+rz)}{r(z+z^{-1}-2\cos\alpha)}. \tag{21}$$

The value of r here is 0.8236.

4.4 Spectral Factorization Algorithm

The final step in the quantile decoding algorithm is to obtain the coefficients of the inverse filter $A(z)$ from the estimated $C(\omega)$. We need an algorithm which would check to see whether the estimated $C(\omega)$ is positive definite and if so would determine a minimum-phase polynomial such that

$$A(z)A(z^{-1}) = C(z).$$

(If the estimated $C(\omega)$ is found not to be positive definite then it is sent to the spectral correction routine for modification.) This problem is called the spectral factorization problem and there exist many techniques in the literature ([48], [49], [50]) for solving it. The technique that we have chosen is due to Friedlander ([50]) because it is very simple, easy to code, and can be implemented using a lattice filter. A brief description of this algorithm is given in Appendix C.

4.5 Choice of model order M

We will now discuss the issues that are involved in the choice of the model order M. We will first show that the weighted mean-square error E, defined by equation (4), is a non-increasing function of M. To prove this we first express E in matrix

notation.

$$
\begin{aligned}
E &= \sum_{\substack{k=0 \\ \omega_k = 2\pi k/N}}^{N/2} \Big(S_o(\omega_k) C(\omega_k) - 1 \Big)^2 W(\omega_k) \\
&= \sum_{i=0}^{M} \sum_{j=0}^{M} c_i c_j \sum_{\substack{k=0 \\ \omega_k = 2\pi k/N}}^{N/2} S_o^2(\omega_k) W(\omega_k) \cos i\omega_k \cos j\omega_k \\
&\quad - 2 \sum_{i=0}^{M} c_i \sum_{\substack{k=0 \\ \omega_k = 2\pi k/N}}^{N/2} S_o(\omega_k) W(\omega_k) \cos i\omega_k + \sum_{\substack{k=0 \\ \omega_k = 2\pi k/N}}^{N/2} W(\omega_k) \\
&= \mathbf{c}^T \mathbf{A} \mathbf{c} - 2\mathbf{c}^T \mathbf{b} + d_o
\end{aligned}
$$

where

$$
d_o = \sum_{\substack{k=0 \\ \omega_k = 2\pi k/N}}^{N/2} W(\omega_k)
$$

and $\mathbf{A}$, $\mathbf{c}$ and $\mathbf{b}$ are as defined in equations (5a), (5b) and (5c).

Recall that $\mathbf{A}$ is non-negative definite and therefore has only non-negative eigenvalues. So det $\mathbf{A}$, which is the product of the eigenvalues, is also non-negative. Let us assume that for some $M = M_1$, det $\mathbf{A} > 0$. Denote the $\mathbf{A}$, $\mathbf{b}$, $\mathbf{c}$ and E for $M = M_1$ by $\mathbf{A_{M_1}}$, $\mathbf{b_{M_1}}$, $\mathbf{c_{M_1}}$ and E_{M_1}. Thus

$$
E_{M_1} = \mathbf{c}_{\mathbf{M_1}}^T \mathbf{A_{M_1}} \mathbf{c_{M_1}} - 2\mathbf{c}_{\mathbf{M_1}}^T \mathbf{b_{M_1}} + d_o.
$$

The optimum $\mathbf{c_{M_1}}$, denoted by $\mathbf{c}^*_{\mathbf{M_1}}$, is obtained by solving equation (5), i.e.,

$$
\mathbf{A_{M_1}} \mathbf{c}^*_{\mathbf{M_1}} = \mathbf{b_{M_1}}
$$

$$
\mathbf{c}^*_{\mathbf{M_1}} = \mathbf{A}^{-1}_{\mathbf{M_1}} \mathbf{b_{M_1}}.
$$

The optimum E_{M_1}, denoted by $E^*_{M_1}$ is therefore

$$
E^*_{M_1} = d_o - \mathbf{b}_{\mathbf{M_1}}^T \mathbf{A}^{-1}_{\mathbf{M_1}} \mathbf{b_{M_1}}. \tag{22}
$$

Next, let $M = M_1 + 1$. Again we denote the $\mathbf{A}$, $\mathbf{b}$, $\mathbf{c}$ and E for $M = M_1 + 1$ by $\mathbf{A_{M_1+1}}$, $\mathbf{b_{M_1+1}}$, $\mathbf{c_{M_1+1}}$ and E_{M_1+1}. Now

$$
\mathbf{A_{M_1+1}} = \begin{pmatrix} \mathbf{A_{M_1}} & \mathbf{e} \\ \mathbf{e}^T & d \end{pmatrix} \tag{23a}
$$

$$\mathbf{b}_{\mathbf{M}_1+1} = \begin{pmatrix} \mathbf{b}_{\mathbf{M}_1} \\ f \end{pmatrix} \tag{23b}$$

where

$$\mathbf{e}^T = [\, e_o \; e_1 \; \ldots \; e_{M_1} \,] \; ; \tag{23c}$$

$$e_i = \sum_{\substack{k=0 \\ \omega_k = 2\pi k/N}}^{N/2} S_o^2(\omega_k) W(\omega_k) \cos i\omega_k \cos (M_1 + 1)\omega_k \; ;$$

$$d = \sum_{\substack{k=0 \\ \omega_k = 2\pi k/N}}^{N/2} S_o^2(\omega_k) W(\omega_k) \cos^2 (M_1 + 1)\omega_k \; ; \tag{23d}$$

$$f = \sum_{\substack{k=0 \\ \omega_k = 2\pi k/N}}^{N/2} S_o(\omega_k) W(\omega_k) \cos^2 (M_1 + 1)\omega_k \; . \tag{23e}$$

Let us further assume that $\mathbf{A}_{\mathbf{M}_1+1}$ is non-singular. Making use of a standard matrix identity ([54], pp. 656), we have

$$\mathbf{A}_{\mathbf{M}_1+1}^{-1} = \begin{pmatrix} \mathbf{A}_{\mathbf{M}_1}^{-1} + r\mathbf{A}_{\mathbf{M}_1}^{-1}\mathbf{e}\mathbf{e}^T\mathbf{A}_{\mathbf{M}_1}^{-1} & -r\mathbf{A}_{\mathbf{M}_1}^{-1}\mathbf{e} \\ -r\mathbf{e}^T\mathbf{A}_{\mathbf{M}_1}^{-1} & r \end{pmatrix} \tag{24}$$

where

$$r = \frac{1}{d - \mathbf{e}^T\mathbf{A}_{\mathbf{M}_1}^{-1}\mathbf{e}}.$$

Using another standard identity ([54], pp. 650), we have

$$\det \mathbf{A}_{\mathbf{M}_1+1} = (d - \mathbf{e}^T\mathbf{A}_{\mathbf{M}_1}^{-1}\mathbf{e}) \det \mathbf{A}_{\mathbf{M}_1}$$

and so

$$r = \frac{\det \mathbf{A}_{\mathbf{M}_1}}{\det \mathbf{A}_{\mathbf{M}_1+1}} > 0.$$

The optimum E_{M_1+1}, denoted by $E_{M_1+1}^*$, is given by an expression similar to equation (22).

$$E_{M_1+1}^* = d_o - \mathbf{b}_{\mathbf{M}_1+1}^T\mathbf{A}_{\mathbf{M}_1+1}^{-1}\mathbf{b}_{\mathbf{M}_1+1} \tag{25}$$

We now substitute for $\mathbf{A}_{\mathbf{M}_1+1}^{-1}$ from equation (24) and for $\mathbf{b}_{\mathbf{M}_1+1}$ from equation (23b) in the expression for E_{M_1+1}. Thus

$$E_{M_1+1}^* = d_o - \begin{pmatrix} \mathbf{b}_{\mathbf{M}_1}^T & f \end{pmatrix} \begin{pmatrix} \mathbf{A}_{\mathbf{M}_1}^{-1} + r\mathbf{A}_{\mathbf{M}_1}^{-1}\mathbf{e}\mathbf{e}^T\mathbf{A}_{\mathbf{M}_1}^{-1} & -r\mathbf{A}_{\mathbf{M}_1}^{-1}\mathbf{e} \\ -r\mathbf{e}^T\mathbf{A}_{\mathbf{M}_1}^{-1} & r \end{pmatrix} \begin{pmatrix} \mathbf{b}_{\mathbf{M}_1} \\ f \end{pmatrix}$$

$$= d_o - \mathbf{b}_{\mathbf{M}_1}^T\mathbf{A}_{\mathbf{M}_1}^{-1}\mathbf{b}_{\mathbf{M}_1} - r\left(\mathbf{b}_{\mathbf{M}_1}^T\mathbf{A}_{\mathbf{M}_1}^{-1}\mathbf{e} - f\right)^2$$

$$= E_{M_1}^* - r\left(\mathbf{b}_{\mathbf{M}_1}^T\mathbf{A}_{\mathbf{M}_1}^{-1}\mathbf{e} - f\right)^2.$$

Therefore, $E_{M_1+1}^* \leq E_{M_1}^*$ since $r > 0$.

Thus, we have proved that the mean square error E is a non-increasing function of model order M. Note that $E_{M_1+1}^* = E_{M_1}^*$ iff $\mathbf{b}_{\mathbf{M}_1}^T\mathbf{A}_{\mathbf{M}_1}^{-1}\mathbf{e} = f$. This can be satisfied, for a given M_1, only by a very specific flat spectral density approximation. In our experience, this has never happened. So, in practice, one could expect $E_{M_1+1}^* < E_{M_1}^*$. This implies that by increasing M, the power spectrum of the AR model can be made to fit the flat spectral density approximation with arbitarily low error. However, as we go on increasing M, it has been observed that the matrix $\mathbf{A}$ becomes more and more ill-conditioned. We can explain this phenomenon as follows.

A measure of ill-conditioning for symmetric positive definite matrices that is often used is the ratio of the largest eigenvalue λ_{max} to the smallest eigenvalue λ_{min} of $\mathbf{A}$. This is called the *condition number* of the matrix $\mathbf{A}$ and is denoted by κ. For

any positive definite matrix, it is well-known that that ([51])

$$\lambda_{max} \geq \max_{0 \leq l \leq M} a_{ll}$$

$$\lambda_{min} \leq \min_{0 \leq l \leq M} a_{ll}$$

$$\text{and so } \kappa = \frac{\lambda_{max}}{\lambda_{min}} \geq \frac{\max_{0 \leq l \leq M} a_{ll}}{\min_{0 \leq l \leq M} a_{ll}}.$$

Clearly, the numerator is a non-decreasing function of M and the denominator is a non-increasing function of M. So the condition number has a lower bound that is a non-decreasing function of M. It is therefore possible for κ to be large for very large M.

Thus, there appears to be a conflict. If we want the power spectrum of the AR model to approximate the flat spectral density approximation closely, then we need a *large* M. On the other hand, if M is large the condition number of the matrix $\mathbf{A}$ could become large and so the vector $\mathbf{c}$ obtained by inverting $\mathbf{A}$ may be unreliable. We need a *small* M to avert this. In practice a value of $M = 10$ appears to be a good compromise. The physics of speech production also suggests that since there are typically four or five formants in the 0-5 KHz range, a model order of 10 would be appropriate on this ground alone.

4.6 Choice of the weighting function

We turn our attention to the choice of the weighting function $W(\omega)$. We would like to choose a weighting function which emphasizes the peaks of the flat spectral density approximation. This is because the peaks which correspond to the formants are the perceptually more important features. Moreover, the flat spectral density approximation fits the spectral envelope better near the peaks. A convenient choice

of the weighting function would be some non-negative power of the flat spectral density approximation itself; i.e., $W(\omega) = S_o^{\nu}(\omega)$ for all $\nu \geq 0$.

As ν is increased, there would be greater emphasis on the peaks, and so we could expect a better match between the power spectrum of the estimated AR model and the flat spectral density approximation near its peaks. However, as ν is increased, we have observed that the condition number κ, defined in the previous section, also increases. This can be explained as follows. We will make use of the bounds for the maximum eigenvalue λ_{max} and minimum eigenvalue λ_{min} of the matrix $\mathbf{A}$, mentioned in the previous section:

$$\begin{aligned}
\lambda_{max} &\geq \max_{0 \leq l \leq M} a_{ll} \\
&= \max_{0 \leq l \leq M} \sum_{\substack{k=0 \\ \omega_k = 2\pi k/N}}^{N/2} S_o^{2+\nu}(\omega_k) \cos^2 l\omega_k \\
&= \sum_{\substack{k=0 \\ \omega_k = 2\pi k/N}}^{N/2} S_o^{2+\nu}(\omega_k) \\
&= \sigma S_{max}^{2+\nu}
\end{aligned}$$

where $S_{max} = \max_{0 \leq k \leq N/2} S_o(\omega_k)$ and σ is a positive constant. Similarly,

$$\begin{aligned}
\lambda_{min} &\leq \min_{0 \leq l \leq M} a_{ll} \\
&= \min_{0 \leq l \leq M} \sum_{\substack{k=0 \\ \omega_k = 2\pi k/N}}^{N/2} S_o^{2+\nu}(\omega_k) \cos^2 l\omega_k
\end{aligned}$$

$$\text{So } \kappa = \frac{\lambda_{max}}{\lambda_{min}} \geq \frac{\sigma S_{max}^{2+\nu}}{\min_{0 \leq l \leq M} \sum_{\substack{k=0 \\ \omega_k = 2\pi k/N}}^{N/2} S_o^{2+\nu}(\omega_k) \cos^2 l\omega_k}$$

$$= \frac{\sigma}{\min_{0 \leq l \leq M} \sum_{\substack{k=0 \\ \omega_k = 2\pi k/N}}^{N/2} \left(\frac{S_o(\omega_k)}{S_{max}} \right)^{2+\nu} \cos^2 l\omega_k}.$$

Let us denote by $L(\nu)$ the lower bound for κ. Let us suppose that for $\nu = \nu_1$, l_1 causes the lower bound to be achieved; i.e.,

$$\sigma L^{-1}(\nu_1) = \min_{0 \le l \le M} \sum_{\substack{k=0 \\ \omega_k = 2\pi k/N}}^{N/2} \left(\frac{S_o(\omega_k)}{S_{max}} \right)^{2+\nu_1} \cos^2 l\omega_k$$

$$= \sum_{\substack{k=0 \\ \omega_k = 2\pi k/N}}^{N/2} \left(\frac{S_o(\omega_k)}{S_{max}} \right)^{2+\nu_1} \cos^2 l_1\omega_k .$$

Let us again suppose that for $\nu = \nu_2 > \nu_1$, l_2 causes the lower bound to be achieved; i.e.,

$$\sigma L^{-1}(\nu_2) = \min_{0 \le l \le M} \sum_{\substack{k=0 \\ \omega_k = 2\pi k/N}}^{N/2} \left(\frac{S_o(\omega_k)}{S_{max}} \right)^{2+\nu_2} \cos^2 l\omega_k$$

$$= \sum_{\substack{k=0 \\ \omega_k = 2\pi k/N}}^{N/2} \left(\frac{S_o(\omega_k)}{S_{max}} \right)^{2+\nu_2} \cos^2 l_2\omega_k .$$

Then

$$\sigma L^{-1}(\nu_2) \le \sum_{\substack{k=0 \\ \omega_k = 2\pi k/N}}^{N/2} \left(\frac{S_o(\omega_k)}{S_{max}} \right)^{2+\nu_2} \cos^2 l_1\omega_k$$

$$\le \sum_{\substack{k=0 \\ \omega_k = 2\pi k/N}}^{N/2} \left(\frac{S_o(\omega_k)}{S_{max}} \right)^{2+\nu_1} \cos^2 l_1\omega_k$$

$$\left(\text{since } \frac{S_o(\omega_k)}{S_{max}} \le 1 \ \forall \ k \right)$$

$$= \sigma L^{-1}(\nu_1).$$

We conclude that $L(\nu_2) \ge L(\nu_1)$. Note that for the equality to hold, we must have $l_1 = l_2$ and $S_o(\omega_k) = S_{max}$ for all ω_k, which is a very unlikely situation. So in practice, $L(\nu)$, the lower bound of the condition number κ, is a strictly monotonically increasing function of ν. So we can expect the condition number κ to be large for very large values of the model order M.

Here there is a conflict again. We would like a *large* value of ν so as to obtain a better fit near the peaks of the flat spectral density approximation. But if the

computations must be reliable, then the condition number κ cannot be too large, and so it would be safer to choose a *small* value of ν. In our implementation we have taken the safe route and chosen $\nu = 0$ or $W(\omega) = 1$; i.e., no weighting is used.

4.7 Smoothing of flat spectral density approximation using ARMA models

Consider an ARMA model $H(z)$ which is given by

$$H(z) = \frac{\sum_{i=0}^{L} d_i z^{-i}}{\sum_{i=0}^{M} a_i z^{-i}} = \frac{D(z)}{A(z)}$$

where $A(z)$ is assumed to be minimum-phase; i.e., all its roots lie within the unit circle. We will also assume that $D(z)$ is minimum-phase. The power spectrum of the ARMA model can be expressed as a ratio of positive definite trignometric polynomials:

$$|H(e^{j\omega})|^2 = \frac{|D(e^{j\omega})|^2}{|A(e^{j\omega})|^2}$$

$$= \frac{\sum_{i=0}^{L} e_i \cos i\omega}{\sum_{i=0}^{M} c_i \cos i\omega}$$

$$= \frac{E(\omega)}{C(\omega)}.$$

Since we can always multiply the numerator and denominator by an arbitary nonzero scale factor, we can assume without loss of generality that e_o=1.

One approach to obtaining the parameters of the ARMA model is to solve for

the $\{e_i\}$ and $\{c_i\}$ by minimizing

$$E_2 = \sum_{\substack{k=0 \\ \omega_k = 2\pi k/N}}^{N/2} \Big(S_o(\omega_k) C(\omega_k) - E(\omega_k) \Big)^2 W(\omega_k)$$

subject to the constraints

$$E(\omega) > 0 \quad , \quad C(\omega) > 0 \qquad 0 \le \omega \le \pi.$$

As in the case of the estimation of $C(\omega)$, we will just minimize E_2 and ignore the constraints. Minimization of E_2 gives rise to a system of linear equations in the $M + L + 1$ unknowns, i.e., in the $\{c_i\}$ and $\{e_i\}$. These equations can be expressed in matrix form as

$$\begin{pmatrix} \mathbf{A} & \mathbf{G} \\ \mathbf{G}^T & \mathbf{H} \end{pmatrix} \begin{pmatrix} \mathbf{c} \\ \mathbf{e} \end{pmatrix} = \begin{pmatrix} \mathbf{b} \\ \mathbf{f} \end{pmatrix}$$

where

$$\mathbf{G} = [g_{ij}]_{(M+1)\times L} \ ;$$

$$g_{ij} = - \sum_{\substack{k=0 \\ \omega_k = 2\pi k/N}}^{N/2} S_o(\omega_k) W(\omega_k) \cos i\omega_k \cos j\omega_k \qquad 0 \le i \le M,\ 1 \le j \le L \ ;$$

$$\mathbf{H} = [h_{ij}]_{L\times L} \ ;$$

$$h_{ij} = \sum_{\substack{k=0 \\ \omega_k = 2\pi k/N}}^{N/2} W(\omega_k) \cos i\omega_k \cos j\omega_k \qquad 1 \le i \le L,\ 1 \le j \le L \ ;$$

$$\mathbf{e} = [\ e_1\ e_2\ \ldots\ e_L\ ] \ ;$$

$$\mathbf{f} = [\ f_1\ f_2\ \ldots\ f_L\ ] \ ;$$

$$f_i = - \sum_{\substack{k=0 \\ \omega_k = 2\pi k/N}}^{N/2} W(\omega_k) \cos i\omega_k \ ;$$

and $\mathbf{A}$, $\mathbf{c}$ and $\mathbf{b}$ are as defined in equations (5a), (5b) and (5c). The estimated $E(\omega)$ and $C(\omega)$ are, however, not guaranteed to be positive definite and may require spectral correction as discussed earlier in Section 4.3. Finally, one obtains $D(z)$ and $A(z)$ by spectral factorization.

It has been our experience that the ARMA smoothing of the flat spectral density approximation does not offer any significant improvements in the estimation of the spectral envelope over the AR smoothing. This is perhaps due to the fact that the flat spectral density approximation fits the shape of the spectral envelope well only near the peaks and rather poorly near the valleys. Besides, the complexity of the ARMA smoothing algorithm is much higher than the AR smoothing algorithm since there are more unknowns to be solved for. It is for these reasons that we have chosen to use AR smoothing and not ARMA smoothing in our implementation.

4.8 Summary

We now summarize the various steps involved in the quantile decoding algorithm. The first step is to set up the flat spectral density approximation, which is given by equation (1a) for medium and high bit rate vocoders and (1b) for low bit rate vocoders. The flat spectral density approximation is smoothed using an AR or ARMA model. AR smoothing is preferred since it involves fewer computations, and the results are only marginally inferior to ARMA smoothing. In order to determine the coefficients of the denominator polynomial of the AR model $A(z)$, we first estimate the coefficients of $C(z) = A(z)A(z^{-1})$. This is obtained by minimizing the distortion measure E, which is defined by equation (4). This reduces to solving equation (5), which is a Toeplitz plus Hankel system of equations. Such a system of linear equations is efficiently solved using the block Levinson algorithm. The estimated $C(z)$ is not guaranteed to be positive definite and may require to be modified by a spectral correction algorithm, though in practice this happens very rarely. Finally, $A(z)$ is obtained by spectral factorization of $C(z)$. A suitable value for the model order, i.e. degree of $A(z)$, is $M = 10$. No weighting of the distortion

measure is employed.

The results of the quantile decoding algorithm when applied to four speech frames are displayed in Fig. 4.2 and Fig. 4.3. Each speech frame has 256 samples at 10 KHz sampling rate. In Fig. 4.2, for each speech frame the frequency range (0-5 KHz) is split into 4 sub-bands and 3 quantiles are chosen in each of them using the algorithm for choosing quantiles for medium and high bit rate vocoders, which was outlined in the previous chapter. Thus there are 14 quantiles, including 0 and π, that are used to represent the spectral envelope. In Fig. 4.3, the same four speech frames are used and for each speech frame the frequency range is split into 3 sub-bands and 3 quantiles are chosen in each of them. Thus there are 11 quantiles, including 0 and π, that are used to represent the spectral envelope. In both cases, the power spectral density of each Hamming-windowed preemphasized speech frame is computed using a 512 point FFT ($N = 512$), plotted and overlaid by a scaled version of the spectral envelope estimate. The scale factor is chosen such that the total power under the spectral envelope estimate is equal to the total power under the power spectral density of the Hamming-windowed preemphasized speech frame. It is clear from the figures that one can obtain a reasonably good estimate of the spectral envelope using few quantiles.

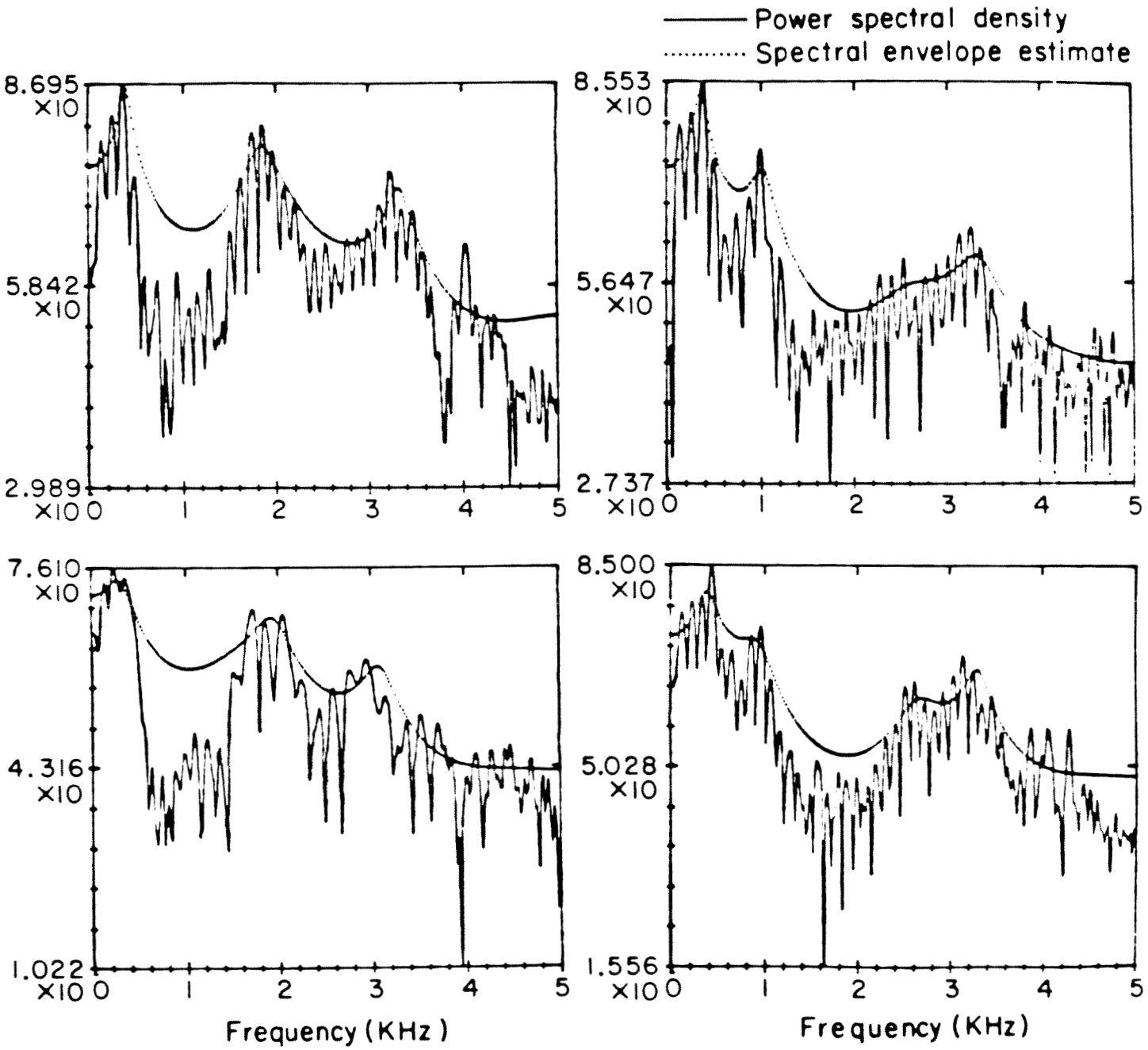


Fig. 4.2 Spectral envelope estimate using 14 quantiles

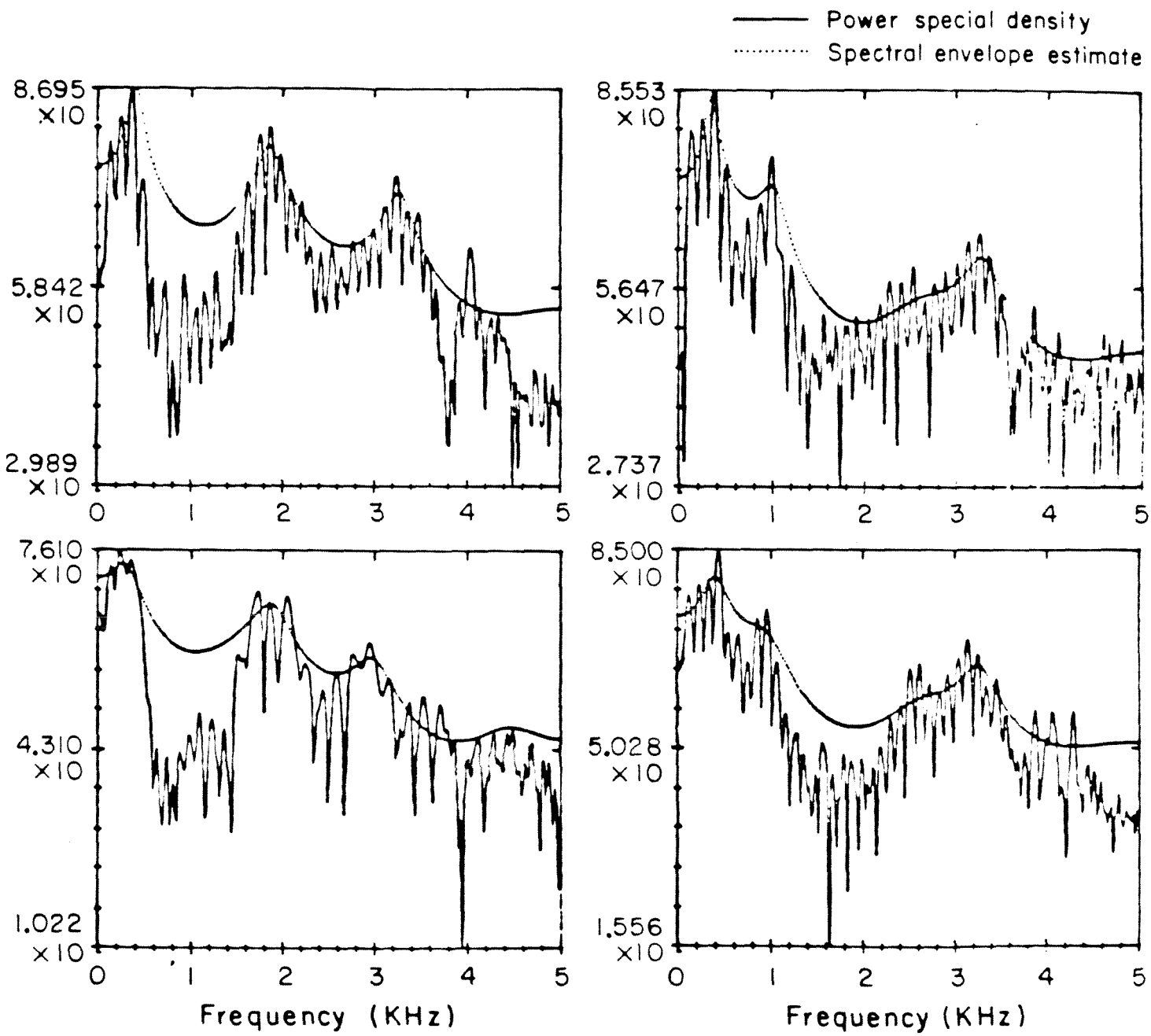


Fig. 4.3 Spectral envelope estimate using 11 quantiles

CHAPTER 5

REVIEW OF THE MULTI-PULSE EXCITATION MODEL

In this chapter we will review the theoretical and implementation aspects of the multi-pulse excitation model. Our treatment here is largely a summary of the papers due to Atal and Remde ([26]), Atal and Singhal ([55]), Kroon and Deprettere ([56]) and Berouti *et al* ([57]). Other relevant contributions in this area are [58]-[59].

5.1 Multi-pulse excitation model

In order to motivate the multi-pulse model for excitation, we first examine the earliest model for speech production and the problems associated with it. In this model for speech production (see Fig. 1.1), every speech segment is classified as *voiced* or *unvoiced* (pp. 40, [3]). For voiced speech, the excitation is a quasi-periodic pulse train with delta functions located at pitch period intervals. For unvoiced speech, the excitation is white noise. The linear filter accounts for the shape of the spectral envelope of the short-time spectrum of the speech segment. In the previous chapter, algorithms to estimate an all-pole filter from the quantiles and quantile orders which represent the spectral envelope of the preemphasized speech spectrum were described. So the linear filter in the quantile vocoder is simply a cascade of the all-pole filter $\frac{1}{A(z)}$ and the deemphasis filter $\frac{1}{H_p(z)}$, which is the inverse of the preemphasis filter $H_p(z)$ defined in Chapter 3. A gain term G is also incorporated. This gain term is chosen such that the energy of the impulse response of the linear filter is equal to the energy of the speech frame. Thus, the transfer function of the linear filter used to represent the spectral envelope is $Q(z) = \frac{G}{A(z)H_p(z)}$.

This model for source excitation was widely used because it was considered the only way to synthesize speech at bit rates around or below 4 Kbits/s. However, it is extremely difficult to produce high quality speech using it, even at high bit

rates. The problem lies in the rigid classification of the speech segment as voiced or unvoiced. Often there are more than two modes in which vocal tract is excited and often these modes are mixed. For such speech segments it is a difficult task to classify them as voiced or unvoiced. Moreover, accurate pitch estimation for voiced segments can also be very difficult. Furthermore, for voiced speech segments, there is evidence to suggest that there is more than one point of excitation during a pitch period ([60]). In addition to the main excitation that occurs at glottal closure, there is evidence of secondary excitation even after glottal closure ([60]). This suggests that the excitation of voiced speech should consist of several pulses in a pitch period and not just one at the beginning of the period.

The multi-pulse model (see Fig. 1.2) was proposed by Atal and Remde ([26]) in order to overcome the above-mentioned problems. In this model for speech production, the excitation is simply taken to be a sequence of pulses for all speech segments. No attempt is made to classify a segment as voiced or unvoiced. Thus the difficulties associated with accurate pitch estimation and voiced-unvoiced classification are avoided. Using this model, we can synthesize speech sounds with little audible distortion by employing more excitation pulses. If the number of pulses is increased to an arbitarily large value so that there is a pulse at every sampling instant, then it would be possible to duplicate the original speech waveform. At the same time, many speech sounds can be synthesized with fairly good quality using very few pulses. Thus, the model is flexible enough to be used even in low bit rate vocoders. The linear filter again accounts for the shape of the spectral envelope of the short-time spectrum of the speech segment.

An improved multi-pulse model (see Fig. 6.1) was proposed by Atal and Singhal ([55]) as well as Kroon and Deprettere ([56]). In the improved multi-pulse model,

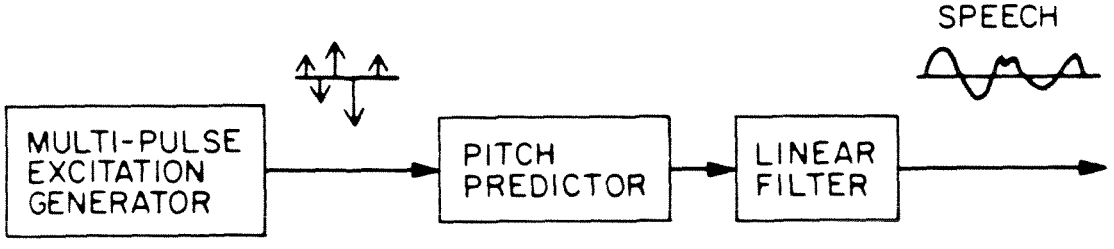


Fig. 5.1 Improved multi-pulse model for speech synthesis

the periodic nature of voiced speech segments is exploited by incorporating a pitch predictor. Thus, in this model the speech is synthesized by passing multi-pulse excitation through a cascade of the pitch predictor and the linear filter. The most general form of a pitch predictor model $P(z)$ used is

$$P(z) = \frac{1}{1 + \mu_1 z^{-(M_p-1)} + \mu_2 z^{-M_p} + \mu_3 z^{-(M_p+1)}} \tag{1}$$

where M_p represents the distance between adjacent pitch samples, and μ_1, μ_2 and μ_3 are scale factors ([61]). This predictor is called a *3-tap predictor.* When μ_1 and μ_3 are set to zero, $P(z)$ reduces to a 1-tap predictor,

$$P(z) = \frac{1}{1 + \mu z^{-M_p}}. \tag{2}$$

In our work we have chosen the 1-tap predictor since it requires fewer bits than the 3-tap predictor to encode its parameters, and the stability of the model can easily be ensured by restricting $|\mu| < 1$.

Though the pitch predictor helps improve the speech synthesis for voiced segments, it is not of much value for obviously unvoiced segments. However, in order to avoid a voiced-unvoiced classification, the pitch predictor is used for all speech segments. For unvoiced segments, the estimated value of M_p will be some random number and the estimated μ will be very small so that the pitch predictor has little effect on the synthesized speech. It must also be emphasized that unlike the earliest model for speech production, accurate estimation of pitch is not necessary. This is because the pitch predictor is used here mainly to exploit the periodic nature of voiced speech. Errors in the estimation of pitch could reduce the effectiveness of the pitch predictor in bit reduction but will not impair the intelligibility or even the

quality of the synthesized speech significantly.

5.2 Estimation of parameters of multi-pulse model

Let us assume that the speech segment which is being modeled has N_F samples. Let the number of excitation pulses be N_p ($N_p \leq N_F$) which are located at n_1, n_2, ..., n_{N_p} with amplitudes a_1, a_2, ..., a_{N_p}. The locations and amplitudes of these pulses are the parameters of this model. We will assume that the pitch predictor is completely determined. Algorithms to estimate the parameters of the pitch predictor, i.e., μ and M_p, are described in Section 5.3.

5.2.1 Basic idea behind the algorithm

The basic idea underlying the algorithm is illustrated in the block diagram of Fig. 6.2. The cascade of the pitch predictor and linear filter produces the synthetic speech samples $\hat{x}(n)$ in response to multi-pulse excitation. The synthetic speech sample $\hat{x}(n)$ is compared to the original speech sample $x(n)$ to produce an error signal $\underline{e}(n)$. This error is not perceptually meaningful and is therefore passed through a perceptual weighting filter to produce a subjectively meaningful measure of the difference between the synthesized and original speech waveforms. The weighted error is squared and averaged to produce a mean-squared error $\underline{\epsilon}$. The locations and amplitudes of the pulses are chosen to minimize the error $\underline{\epsilon}$.

5.2.2 Perceptual weighting of error

The perceptual weighting filter needs further explanation. The error between the speech signal and the synthetic speech signal can be thought of as *noise* introduced by the speech coder. The primary goal must therefore be to choose the locations and amplitudes of the pulses so as to make this noise inaudible or to

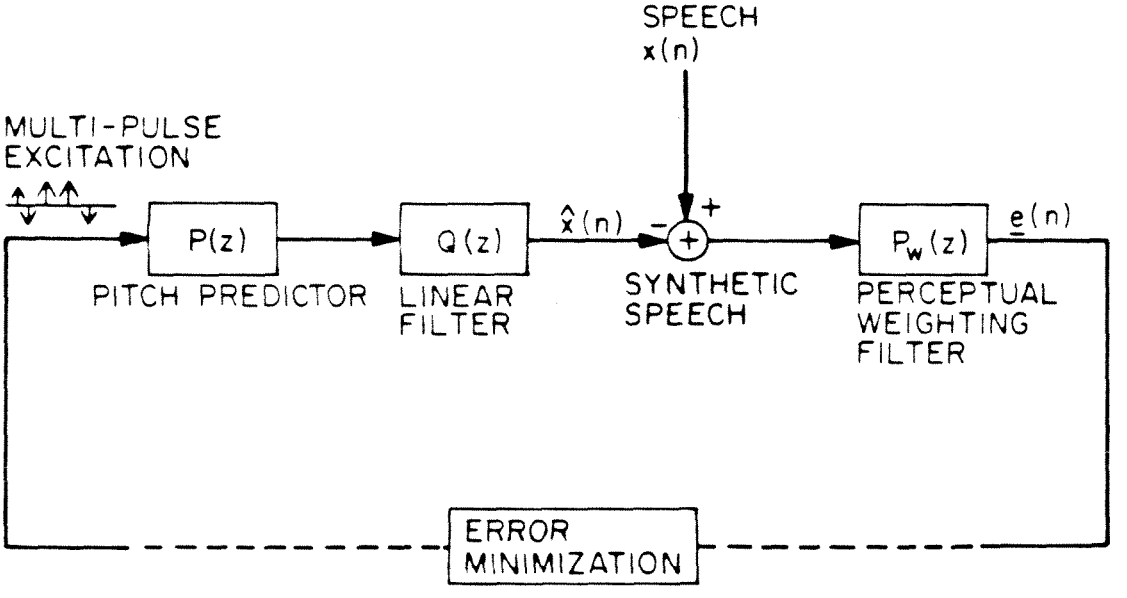


Fig. 5.2 Block diagram of procedure for estimating pulse locations and amplitudes

minimize its loudness. The loudness of the noise, as perceived by the human ear, is determined not just by its total power but by the shape of the power spectral density of the noise and the speech signal. If the noise spectrum lies under or near the peaks of the spectrum of the speech signal, then the noise is reduced in perceived loudness and can even become completely inaudible. In other words, the human ear can tolerate larger errors in the formant regions in comparison to that tolerated in the frequency regions between formants. This phenomenon is called *auditory masking* ([62]).

The masking properties of the human ear suggest that the noise should be frequency-weighted prior to minimization. This is accomplished easily by passing the noise through a filter which deemphasizes it near the formants. Thus, if the spectral envelope is represented by the linear filter $Q(z)$, then a suitable perceptual weighting filter would be

$$P_W(z) = \frac{Q(\Gamma z)}{Q(z)} \tag{3}$$

where Γ is a fraction between 0 and 1. This is because $Q(\Gamma z)$ has broader resonances than $Q(z)$ and so the magnitude of the ratio has a minimum at every formant frequency. The value of Γ is determined by the degree to which one wishes to deemphasize the formant regions in the error spectrum. Typical values lie in the range 0.8-0.9. In our work, we have chosen $\Gamma = 0.9$. Fig. 6.3 shows an example of the power spectra of the linear filter $Q(z)$ and the corresponding perceptual weighting filter $P_W(z)$ for $\Gamma = 0.9$.

5.2.3 Error minimization procedure

Let us denote the first N_F output samples of the perceptual weighting filter, when excited by the N_F-point speech segment, as $u(0), u(1), \ldots, u(N_F - 1)$. Let us also denote the first N_F samples of the impulse response of the cascade of the

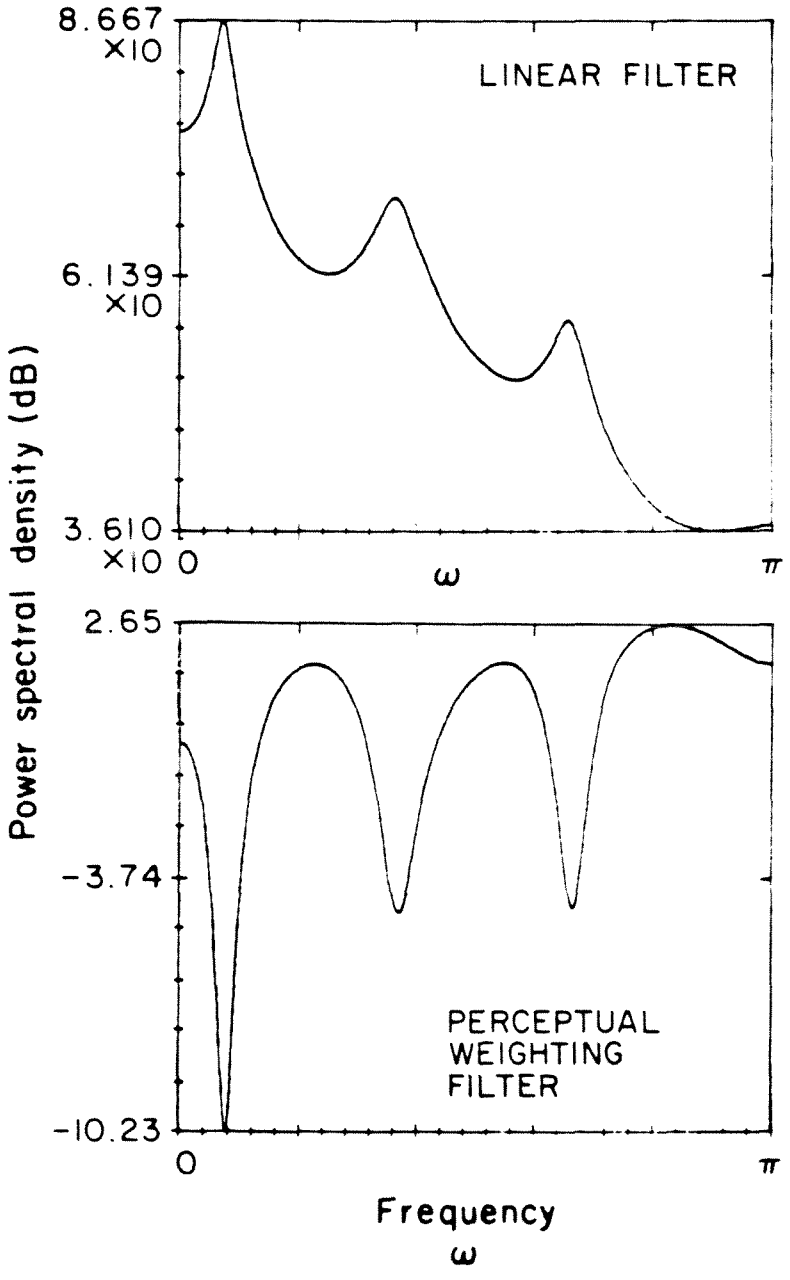


Fig. 5.3 Power spectra of linear filter and corresponding perceptual weighting filter ($\Gamma = 0.9$)

pitch predictor, linear filter and the perceptual weighting filter as $v(0)$, $v(1)$, ..., $v(N_F - 1)$.

Ideally, what we would like to do is to determine the pulse locations n_1, n_2, ..., n_{N_p} and the corresponding amplitudes a_1, a_2, ..., a_{N_p} which minimize the weighted mean-square error

$$\underline{\epsilon} = \sum_{n=0}^{N_F-1} \left[u(n) - \sum_{j=1}^{N_p} a_j v(n - n_j) \right]^2. \tag{4}$$

This can be done as follows. For a given set of pulse locations, minimization of $\underline{\epsilon}$ with respect to a_1, a_2, ..., a_{N_p} simply gives rise to a set of linear equations which can easily be solved to obtain the optimum pulse amplitudes a_1^*, a_2^*, ..., $a_{N_p}^*$. We then evaluate the error $\underline{\epsilon}(a_1^*, a_2^*, \ldots, a_{N_p}^*, n_1, n_2, \ldots, n_{N_p})$ for every set of locations. The optimum set of locations is of course the one which results in the smallest error.

It is clear, however, that such a procedure would be prohibitively expensive computationally. We therefore settle for a suboptimal solution which is less expensive to compute. One such solution, suggested by Atal and Remde ([26]), has been found to be very promising. We will first outline this solution before discussing the details of its implementation.

In this approach, we obtain the pulse locations one by one. Let us assume that j pulse locations n_1, n_2, ..., n_j and the corresponding amplitudes $a_1^{(j)}$, $a_2^{(j)}$, ..., $a_j^{(j)}$ are known. The $(j+1)^{th}$ pulse location is obtained as follows. We first set up the error measure

$$\underline{\epsilon}_{j+1} = \sum_n \left[u(n) - \sum_{i=1}^{j} a_i^{(j)} v(n - n_i) - a_{j+1} v(n - n_{j+1}) \right]^2. \tag{5}$$

We will leave the range of summation unspecified for the moment. The range will be described later in this section. For a given n_{j+1}, we see that $\underline{\epsilon}_{j+1}$ is minimized

when

$$\frac{\partial \epsilon_{j+1}}{\partial a_{j+1}} = -2\sum_n \Big[u(n) - \sum_{i=1}^{j} a_i^{(j)} v(n-n_i)\Big] v(n-n_{j+1}) + 2a_{j+1}\sum_n v^2(n-n_{j+1})$$
$$= 0. \tag{6}$$

Substituting for a_{j+1} from the above equation in equation (5), we get the minimum error $\underline{\epsilon}^*_{j+1}(n_{j+1})$ as

$$\underline{\epsilon}^*_{j+1}(n_{j+1}) = \sum_n \Big[u(n) - \sum_{i=1}^{j} a_i^{(j)} v(n-n_i)\Big]^2 - \sum_n \Big[u(n) - \sum_{i=1}^{j} a_i^{(j)} v(n-n_i)\Big] v(n-n_{j+1}). \tag{7}$$

Here only the second term depends on n_{j+1}. Thus, the $(j+1)^{th}$ pulse location that results in the smallest $\underline{\epsilon}^*_{j+1}(n_{j+1})$ is that value of n_{j+1} for which $\rho(n_{j+1}) - \sum_{i=1}^{j} a_i^{(j)} \phi(n_i, n_{j+1})$ is a maximum, where

$$\phi(i,k) = \sum_n v(n-i)v(n-k) \tag{8}$$

$$\rho(i) = \sum_n u(n)v(n-i). \tag{9}$$

This can be done very rapidly if we compute and store $\rho(i)$ and $\phi(i,k)$ for all i and k before estimating the pulse locations and amplitudes.

Having determined the $(j+1)^{th}$ pulse location n_{j+1}, we reoptimize all pulse amplitudes by minimizing

$$\underline{\epsilon}_{j+1} = \sum_n \Big[u(n) - \sum_{i=1}^{j+1} a_i v(n-n_i)\Big]^2 \tag{10}$$

with respect to $a_1, a_2, \ldots, a_{j+1}$. Thus, the new pulse amplitudes $a_1^{(j+1)}, a_2^{(j+1)}, \ldots, a_{j+1}^{(j+1)}$ can simply be obtained by solving the following linear equations:

$$\sum_{k=1}^{j+1} a_k^{(j+1)} \phi(n_i, n_k) = \rho(n_i) \qquad 1 \le i \le j+1. \tag{11}$$

Note that while the pulse locations are obtained one by one, the pulse amplitudes are optimized simultaneously after estimating each pulse location.

We now turn our attention to some practical details. Let us first consider the range of summation for n in the equations (5)-(11). One choice is to set the causal sequences $u(n)$ and $v(n)$ to zero for all $n \geq N_F$. Then the range of summation can be extended from $-\infty$ to $+\infty$. We now make the observation that

$$\begin{aligned}\phi(i,k) &= \sum_{-\infty}^{\infty} v(n-i)v(n-k) \\ &= \sum_{-\infty}^{\infty} v(n)v(n+i-k) \\ &= \phi(0,|i-k|).\end{aligned}$$

We also note that $\phi(0,m) = \phi(0,-m) = 0$ for all $m \geq N_F$, since $v(n)$ is zero for all $n \geq N_F$. So one only needs to compute and store $\phi(0,0)$, $\phi(0,1)$, ..., $\phi(0,N_F-1)$. We also note that $\phi(0,m)$ can be interpreted as the autocorrelation of the sequence $v(n)$. Similarly, $\rho(m)$ can be interpreted as the cross-correlation between $u(n)$ and $v(n)$. For this reason, the algorithm employing this choice of the range of summation is called an autocorrelation-type algorithm. Autocorrelation and cross-correlation of sequences with only a finite number of nonzero terms are efficiently computed using the FFT (Chapter 11, [35]). Thus, for the autocorrelation-type algorithm, we require $2N_F$ storage elements and, as usual for FFT-based techniques, $\Upsilon N_F \log_2 N_F$ arithmetic operations. The value for Υ depends on the particular implementation of the FFT and usually lies in the range 2-4. For typical frame sizes such as $N_F = 256$ samples, this computational and storage requirement can easily be met.

A second choice is to sum from $n = 0$ to $n = N_F - 1$. No assumption is made here about $u(n)$ or $v(n)$ for $n \geq N_F$. The algorithm employing this choice of the range of summation is called a covariance-type algorithm. The $\rho(i)$'s are directly

obtained using (11), i.e.,

$$\rho(i) = \sum_{n=i}^{N_F-1} u(n)v(n-i).$$

Since $\phi(i,k) = \phi(k,i)$, only the value of $\phi(i,k)$ for $0 \leq i \leq k \leq N_F - 1$ need be computed and stored. An efficient method for computing all the values of $\phi(i,k)$ is to first compute

$$\phi(N_F - 1 - i, N_F - 1) = v(i)v(0) \qquad 0 \leq i \leq N - 1 \tag{12}$$

and then, for k in the range $[1, N_F - 1]$, to use the recursion

$$\phi(N_F - 1 - i, N_F - 1 - k) = \phi(N_F - i, N_F - k) + v(i)v(k) \qquad k \leq i \leq N_F - 1. \tag{13}$$

Equations (12) and (13) follow directly from the definition of $\phi(i,k)$ in equation (8). For this choice of range of summation, we require $0.5N_F(N_F + 1)$ storage elements for the $\phi(i,k)$'s and another N_F for the $\rho(i)$'s. We would require $0.5N_F(N_F + 1)$ multiplications for the $\phi(i,k)$'s and another $0.5N_F(N_F+1)$ for the $\rho(i)$'s. For typical frame sizes such as $N_F = 256$, this amounts to 33152 storage elements and 65792 multiplications, which is a very heavy computational burden.

One way of getting around this problem is to divide the frame into L subframes of size $\overline{N}_F = \frac{N_F}{L}$ and estimate $\overline{N}_p = \frac{N_p}{L}$ pulse locations and amplitudes in each subframe. (For this, we assume that L divides both N_F and N_p.) Note that the $v(n)$'s and hence the $\phi(i,k)$'s are the same for all subframes. So we require $0.5\overline{N}_F(\overline{N}_F+1)$ storage elements and $0.5\overline{N}_F(\overline{N}_F + 1)$ multiplications for the $\phi(i,k)$'s.

Here, however, $u(n)$ needs to be evaluated every subframe. For the first subframe $u(n)$ is simply the first $\overline{N}_F$ output samples of the perceptual weighting filter when excited by the first $\overline{N}_F$-point speech subframe. But for the subsequent subframes, $u(n)$ is the first $\overline{N}_F$ output samples of the perceptual weighting filter when

excited by a difference signal. This is the difference between the corresponding $\overline{N}_F$-point speech subframe and the synthetic speech output generated from the memory of the cascade of the pitch predictor and the linear filter from previous subframes. The $\rho(i)$'s thus have to be computed for each subframe.

For each subframe we require $\overline{N}_F$ storage elements and $0.5\overline{N}_F(\overline{N}_F + 1)$ multiplications for the $\rho(i)$'s. Since we are estimating the pulse locations and amplitudes in each subframe one after another, the storage requirement for the $\rho(i)$'s for the entire frame is still $\overline{N}_F$, but it would require $0.5L\overline{N}_F(\overline{N}_F + 1)$ multiplications. For typical values such as $N_F = 256$ samples and $L = 4$ subframes, the total storage requirement for the entire frame is thus $\overline{N}_F + 0.5\overline{N}_F(\overline{N}_F + 1) = 2144$ storage elements. The total number of multiplications would be $0.5(L+1)\overline{N}_F(\overline{N}_F+1) = 10400$. Thus, both the storage requirements as well as the computational complexity have been considerably reduced.

We have considered two distinct choices for the range of summation. In the first choice, which led to the autocorrelation-type algorithm, the range of summation of N_F extended from $-\infty$ to $+\infty$ after setting $u(n)$ and $v(n)$ to zero for all $n \geq N_F$. In the second choice, which led to the covariance-type algorithm, the range of summation extended from $n = 0$ to $n = N_F - 1$, and no assumption was made about $u(n)$ or $v(n)$ for $n \geq N_F$. Since no assumption was made about $u(n)$ or $v(n)$ for $n \geq N_F$, the covariance-type algorithm gives better results than the autocorrelation type algorithm, as expected. However, for typical frame sizes, both the storage requirements as well as the computational load are very high for the covariance-type algorithm (unlike the autocorrelation-type algorithm). To ensure that storage requirements and computational load are within reasonable limits for typical frame sizes in the covariance-type algorithm, we divided each frame into L subframes and

estimated $\overline{N}_p = \frac{N_p}{L}$ pulse locations and amplitudes in each subframe.

For medium or high bit rate vocoders, the number of pulses per frame N_p is large and so there is no difficulty in choosing N_p to be a multiple of L. For low bit rate vocoders, where N_p is very small, this can be very inconvenient. When N_p is small, it has been our experience that the autocorrelation-type algorithm is only marginally inferior to the covariance-type algorithm. Thus for low bit rate vocoders, the autocorrelation-type algorithm seems appropriate, while for medium and high bit rate vocoders, the covariance-type algorithm seems appropriate. In our implementation, we have chosen the autocorrelation-type algorithm for the 4.8 Kbits/s and the covariance-type algorithm for the 9.6, 16 and 24 Kbits/s vocoders.

Another aspect of the multi-pulse algorithm of practical relevance is the technique used to solve for the new set of pulse amplitudes after estimating every pulse location. The new set of pulse amplitudes after estimating the $(j+1)^{th}$ pulse location is obtained by solving equation (11). Thus

$$\mathbf{a_{j+1}} = \phi_{\mathbf{j+1}}^{-1} \rho_{\mathbf{j+1}} \tag{14}$$

where

$$\mathbf{a_{j+1}} = [a_1^{(j+1)}\ a_2^{(j+1)}\ \ldots\ a_{j+1}^{(j+1)}]^T \tag{14a}$$

$$\phi_{\mathbf{j+1}} = [\phi(n_i, n_k)]_{(j+1)\times(j+1)} \tag{14b}$$

$$\rho_{\mathbf{j+1}} = [\rho(n_1)\ \rho(n_2)\ \ldots\ \rho(n_{j+1})]^T. \tag{14c}$$

It is easy to see that $\phi_{\mathbf{j+1}}$ is a symmetric matrix, since

$$\begin{aligned}(i,k)^{th} \text{ element of } \phi_{\mathbf{j+1}} &= \phi(n_i, n_k) \\ &= \phi(n_k, n_i) \\ &= (k,i)^{th} \text{ element of } \phi_{\mathbf{j+1}}.\end{aligned}$$

It can also be easily seen that ϕ_{j+1} is non-negative definite since for an arbitary $j+1$-dimensional vector $\mathbf{x}$, we have

$$\begin{aligned}\mathbf{x}^T\phi_{j+1}\mathbf{x} &= \sum_{i=1}^{j+1}\sum_{k=1}^{j+1} x_i x_k \sum_n v(n-n_i)v(n-n_k) \\ &= \sum_n \Big[\sum_{i=1}^{j+1} x_i v(n-n_i)\Big]^2 \\ &\geq 0.\end{aligned}$$

Hence, we can invert ϕ_{j+1} using the so-called *Choleski decomposition* ([47]). This would involve $O(j^3)$ operations. However, by relating ϕ_{j+1} to ϕ_j, and hence ϕ_{j+1}^{-1} to ϕ_j^{-1}, we can reduce the computational load to $O(j^2)$. To see how this can be done, we first note that

$$\phi_{j+1} = \begin{pmatrix} \phi_j & \mathbf{q} \\ \mathbf{q}^T & \phi(n_{j+1}, n_{j+1}) \end{pmatrix} \tag{15}$$

where

$$\mathbf{q} = [\phi(n_1, n_{j+1})\ \phi(n_2, n_{j+1})\ \ldots\ \phi(n_j, n_{j+1})]^T. \tag{16}$$

Using a standard matrix identity ([54], pp. 656), we have

$$\phi_{j+1}^{-1} = \begin{pmatrix} \phi_j^{-1} + t\phi_j^{-1}\mathbf{q}\mathbf{q}^T\phi_j^{-1} & -t\phi_j^{-1}\mathbf{q} \\ -t\mathbf{q}^T\phi_j^{-1} & t \end{pmatrix} \tag{17}$$

where

$$t = \frac{1}{\phi(n_{j+1}, n_{j+1}) - \mathbf{q}^T\phi_j^{-1}\mathbf{q}}. \tag{18}$$

Using equations (17) and (18), one can compute ϕ_{j+1}^{-1} in just $O(j^2)$ operations.

This concludes our discussion of the multi-pulse algorithms. We briefly summarize the various steps in the two algorithms below.

Autocorrelation-type algorithm:

Step 1: Compute $v(n)$, the impulse response of the cascade of the perceptual weighting filter, the linear filter, and the pitch predictor, for $0 \leq n \leq N_F - 1$. Then, defining $v(n) = 0$ for all $n \geq N_F$, compute and store $\phi(0, m) = \sum_{-\infty}^{\infty} v(n)v(n-m)$ for all

$0 \leq m \leq N_F - 1$. The computation can be done rapidly using FFT-based techniques (Chapter 11, [35]). Note that $\phi(i,k) = \phi(0,|i-k|)$ and that $\phi(0,m) = \phi(0,-m) = 0$ for all $m \geq N_F$.

Step 2: Compute $u(n)$, the output of the perceptual weighting filter when excited by the N_F-point speech segment, for $0 \leq n \leq N_F - 1$. Then, defining $v(n) = u(n) = 0$ for all $n \geq N_F$, compute and store $\rho(m) = \sum_{-\infty}^{\infty} u(n)v(n-m)$ for $0 \leq m \leq N_F - 1$. The computation can again be done rapidly using FFT-based techniques as above. Note that $\rho(m) = 0$ for all $m \geq N_F$.

Step 3: The first pulse location n_1 is simply that value of m $(0 \leq m \leq N_F - 1)$ for which $\rho(m)$ is a maximum. The corresponding pulse amplitude estimate is given by

$$a_1^{(1)} = \frac{\rho(n_1)}{\phi(n_1,n_1)}.$$

The $(j+1)^{th}$ pulse location n_{j+1} $(1 \leq j \leq N_p - 1)$ is obtained as that value of m $(0 \leq m \leq N_F - 1)$ for which $\rho(m) - \sum_{i=1}^{j} a_i^{(j)}\phi(n_i,m)$ is a maximum. The new set of pulse amplitudes is obtained using equation (14); i.e.,

$$\mathbf{a_{j+1}} = \phi_{j+1}^{-1}\rho_{j+1}$$

where $\mathbf{a_{j+1}}$, ϕ_{j+1} and ρ_{j+1} are as defined in equations (14a)-(14c). The inversion of the matrix ϕ_{j+1} can be done rapidly using equations (17) and (18).

Covariance-type algorithm:

Step 1: Compute $v(n)$, the impulse response of the cascade of the perceptual weighting filter, the linear filter, and the pitch predictor, for $0 \leq n \leq \overline{N}_F - 1$. Compute and store $\phi(i,k) = \sum_{n=0}^{\overline{N}_F - 1} v(n-i)v(n-k)$ for all $0 \leq i \leq k \leq \overline{N}_F - 1$. This is done efficiently using equations (12) and (13) (with N_F replaced by $\overline{N}_F$).

Steps 2 and 3 below are performed for every subframe in the frame.

Step 2: For the first subframe, $u(n)$ is defined as the output of the perceptual weighting filter when excited by the first $\overline{N}_F$-point speech subframe. For the subsequent subframes, $u(n)$ is defined as the output of the perceptual weighting filter when excited by the difference between the corresponding $\overline{N}_F$-point speech subframe and the synthetic speech output generated from the memory of the cascade of the pitch predictor and the linear filter from previous subframes. Compute $u(n)$ for $0 \leq n \leq \overline{N}_F - 1$ and also compute and store $\rho(m) = \sum_{n=m}^{\overline{N}_F-1} u(n)v(n-m)$ for $0 \leq m \leq \overline{N}_F - 1$.

Step 3: Same as in the autocorrelation-type algorithm (with N_F replaced by $\overline{N}_F$ and N_p by $\overline{N}_p$).

5.3 Estimation of parameters of pitch predictor

The parameters of the 1-tap pitch predictor are M_p and μ. One approach to estimating the pitch predictor parameters is illustrated in Fig. 6.4. The perceptually weighted error is passed through the inverse of the pitch predictor to produce a new error $\bar{e}(n)$. The new error $\bar{e}(n)$ is squared and averaged to produce a new mean square error $\bar{\epsilon}$. The parameters of the pitch predictor are chosen to minimize $\bar{\epsilon}$.

However, the locations and amplitudes of the excitation pulses are unknown at this point in the process. In fact, the algorithm for estimating the pulse locations and amplitudes, described in the previous section, assumes that the pitch predictor parameters are known and have already been determined. So, for the estimation of M_p and μ, we use some *a priori* estimate of the excitation. A simple but adequate one is to assume that the excitation is an impulse of unit strength located at $n = 0$.

Let us denote the first N_F samples of the impulse response of the cascade of the

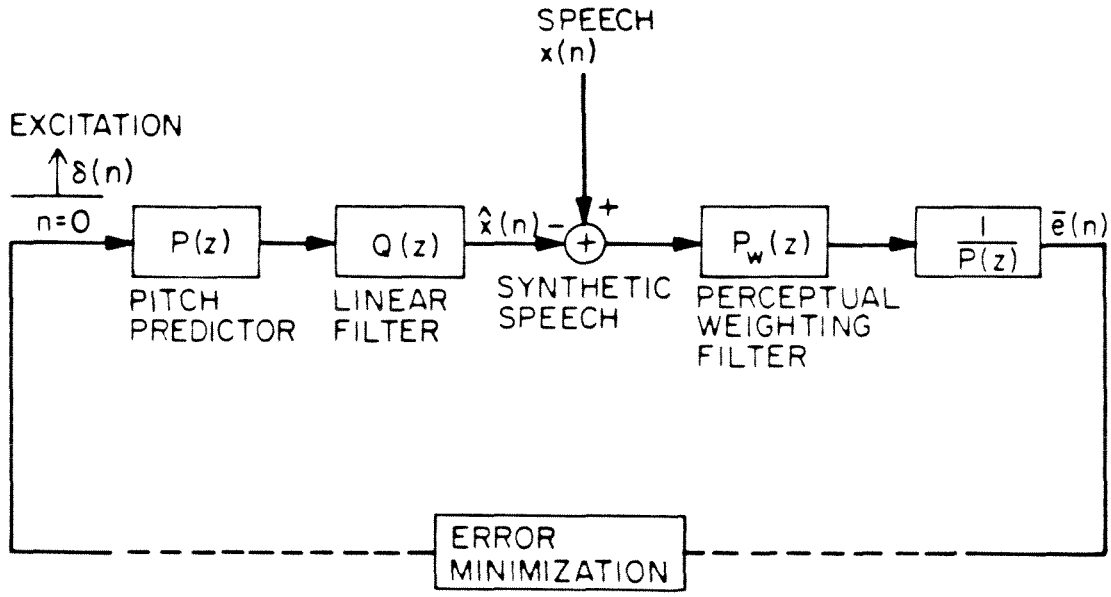


Fig. 5.4 Block diagram of procedure for estimating pitch predictor parameters

linear filter and perceptual weighting filter as $\bar{v}(0), \bar{v}(1), \ldots, \bar{v}(N_F - 1)$. As in the previous section, we denote the first N_F output samples of the perceptual weighting filter, when excited by the N_F point speech segment, as $u(0), u(1), \ldots, u(N_F - 1)$. Then $\bar{\epsilon}$ for a given μ and M_p is given by

$$\bar{\epsilon} = \sum_{n=0}^{N_F-1} \Big[u(n) + \mu u(n - M_p) - \bar{v}(n)\Big]^2. \tag{19}$$

For minimum $\bar{\epsilon}$, we must have

$$\begin{aligned}\frac{\partial \bar{\epsilon}}{\partial \mu} &= \sum_{n=0}^{N_F-1} \Big[u(n) - \bar{v}(n)\Big] u(n - M_p) + \mu \sum_{n=0}^{N_F-1} u^2(n - M_p) \\ &= 0.\end{aligned}$$

Thus, for a given M_p, the optimum $\mu = \mu^*(M_p)$ is given by

$$\mu^*(M_p) = \frac{-\sum_{n=M_p}^{N_F-1} \Big[u(n) - \bar{v}(n)\Big] u(n - M_p)}{\sum_{n=M_p}^{N_F-1} u^2(n - M_p)}. \tag{20}$$

Substituting in (19), we get $\bar{\epsilon}^*(M_p)$, the minimum $\bar{\epsilon}$ for a given M_p, as

$$\bar{\epsilon}^*(M_p) = \sum_{n=0}^{N_F-1} \Big[u(n) - \bar{v}(n)\Big]^2 - \bar{\epsilon}_1(M_p) \tag{21}$$

where

$$\bar{\epsilon}_1(M_p) = \frac{\Big[\sum_{n=M_p}^{N_F-1} [u(n) - \bar{v}(n)] u(n - M_p)\Big]^2}{\sum_{n=M_p}^{N_F-1} u^2(n - M_p)}. \tag{22}$$

The first term in equation (21) for $\bar{\epsilon}^*(M_p)$ does not depend on M_p. The optimum M_p is thus simply that value for which $\bar{\epsilon}_1(M_p)$ is a maximum. Note that in evaluating the denominator for consecutive values of M_p, one can exploit the simple recursion relation

$$\sum_{n=M_p}^{N_F-1} u^2(n - M_p) = \sum_{n=M_p+1}^{N_F-1} u^2(n - M_p - 1) + u^2(N_F - M_p - 1). \tag{23}$$

The search for the optimum distance M_p between adjacent pitch pulses is restricted only to the range in which pitch periods usually lie. In our work, the range chosen was 22 to 149 at all bit rates. At a 7.5 KHz sampling rate, which is the sampling frequency for the 4.8 and 9.6 Kbits/s vocoders in our implementation, this corresponds to the range of 50.3 Hz to 340.9 Hz for the fundamental frequency. At a 10 KHz sampling rate, which is the sampling frequency for the 16 and 24 Kbits/s vocoders in our implementation, this corresponds to the range of 67.1 Hz to 454.5 Hz.

We now summarize the procedure for estimating μ and M_p. We first compute $\bar{v}(n)$ and $u(n)$ for $0 \leq n \leq N_F - 1$. The optimum M_p is that value of M_p in the relevant range for which $\bar{\epsilon}_1(M_p)$, given by equation (22), is a maximum. The corresponding optimum μ is computed using equation (20).

CHAPTER 6

QUANTIZATION AND BIT ALLOCATION

In this chapter we consider the quantization and encoding of the various parameters that must be transmitted every frame in the quantile vocoder. For high and medium bit rate vocoders, the parameters that must be transmitted every frame are the quantile orders, quantiles, locations and amplitudes of the excitation pulses, pitch predictor parameters and the gain G of the linear filter $Q(z)$, all defined in the previous chapter. For low bit rate vocoders, the quantile orders are fixed and need not be transmitted. So the parameters that must be transmitted every frame are the quantiles, locations and amplitudes of the excitation pulses, pitch predictor parameters and the gain. A block diagram of the implementation of the transmitter and receiver of the quantile vocoder is shown in Fig. 6.1. We begin by considering the quantization and encoding of quantile orders.

6.1 Quantization and encoding of quantile orders

An accurate estimation of the spectral envelope is certainly necessary for the synthesis of good quality speech. But errors that arise due to the quantization of quantile orders result in some error in the estimation of the spectral envelope. Thus, one suitable criterion with respect to which we can develop an optimal quantization scheme is to minimize the maximum spectral deviation due to quantization.

We will now define the term *spectral deviation* more precisely. The spectral deviation $\Delta\hat{S}(\xi)$ due to a perturbation $\Delta\xi$ in some transmitted parameter ξ is defined as

$$\Delta\hat{S}(\xi) = \left[\frac{1}{1+\frac{N}{2}} \sum_{\substack{k=0 \\ \omega_k = 2\pi k/N}}^{N/2} \left|\log \frac{\hat{S}(\xi,\omega_k)}{\hat{S}(\xi+\Delta\xi,\omega_k)}\right|^p\right]^{\frac{1}{p}} \qquad (1)$$

where p is a positive integer and $\hat{S}(\xi,\omega)$ is the estimate of the spectral envelope.

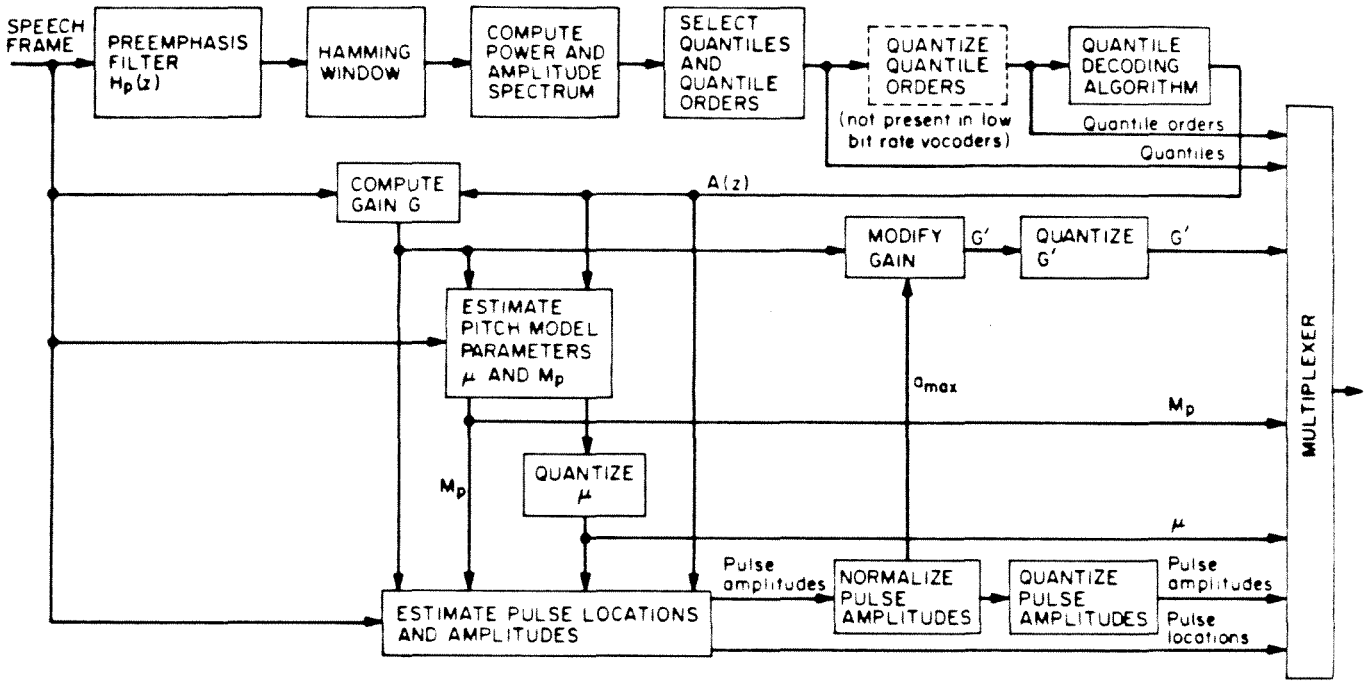


Fig. 6.1(a) Block diagram of transmitter

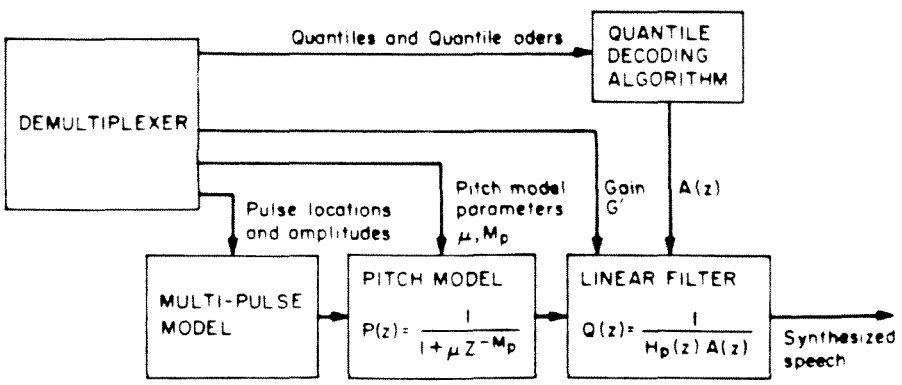


Fig. 6.1(b) Block diagram of receiver

The spectral deviation can thus be interpreted as the logarithmic difference in the estimates of the spectral envelope due to quantization, averaged over all frequencies with equal weights. Another related term is the *spectral sensitivity* $\eta(\xi)$ with respect to the transmitted parameter ξ. This is simply defined as the absolute value of the ratio of the spectral deviation $\Delta\hat{S}(\xi)$ to the perturbation $\Delta\xi$ in the limit as $\Delta\xi$ tends to zero; i.e. ,

$$\eta(\xi) = \lim_{\Delta\xi \to 0} \left| \frac{\Delta\hat{S}(\xi)}{\Delta\xi} \right|. \tag{2}$$

A similar criterion and definition of spectral deviation and spectral sensitivity (with summations replaced by integrals and p set to 1) was used by Viswanathan and Makhoul ([63]) to study the quantization properties of the transmission parameters of linear prediction coders. A similar study was also made by Gray and Markel ([64]) using $p = 2$.

In the quantile vocoder, we claim that the spectral deviation $\Delta\hat{S}(\xi)$ in the final estimate of the spectral envelope can be approximated by the spectral deviation $\Delta S_o(\xi)$ in the flat spectral density approximation for the same perturbation $\Delta\xi$ in the transmitted parameter ξ. This can be justified as follows. We first observe

$$\begin{aligned}
\log \frac{\hat{S}(\xi,\omega_k)}{\hat{S}(\xi+\Delta\xi,\omega_k)} &= \log \frac{\hat{S}(\xi,\omega_k)}{S_o(\xi,\omega_k)} + \log \frac{S_o(\xi,\omega_k)}{S_o(\xi+\Delta\xi,\omega_k)} + \log \frac{S_o(\xi+\Delta\xi,\omega_k)}{\hat{S}(\xi+\Delta\xi,\omega_k)} \\
&= -\log T(\xi,\omega_k) + \log \frac{S_o(\xi,\omega_k)}{S_o(\xi+\Delta\xi,\omega_k)} + \log T(\xi+\Delta\xi,\omega_k) \\
&= \log \frac{S_o(\xi,\omega_k)}{S_o(\xi+\Delta\xi,\omega_k)} + \frac{\Delta\xi}{T(\xi,\omega_k)} \frac{\partial T(\xi,\omega_k)}{\partial\xi} + o(\Delta\xi)
\end{aligned} \tag{3}$$

where $T(\xi,\omega_k)$ is the ratio of the flat spectral density approximation to the estimated spectral envelope; i.e. ,

$$T(\xi,\omega_k) = \frac{S_o(\xi,\omega_k)}{\hat{S}(\xi,\omega_k)} \tag{4}$$

Recall that the estimated spectral envelope is a smoothed version of the flat spectral density approximation. The smoothing is accomplished by determining an autoregressive model whose power spectrum fits the flat spectral density approximation. For very large model orders, this fit is very good and so $T(\xi, \omega_k) \approx 1$ for all possible values of ξ and ω_k. As a result, $T(\xi, \omega_k)$ is not sensitive to small changes in the value of the transmission parameter ξ for all ω_k; i.e. ,

$$\frac{\partial T(\xi, \omega_k)}{\partial \xi} \approx 0 \qquad \forall \omega_k. \tag{5}$$

However, as explained in Section 4.5, we cannot have large values for the model order in practice due to computational limits. The approximation in equation (5) is therefore less accurate in practice, but is still a reasonable one. Hence, for small $\Delta\xi$, we have

$$\log \frac{\hat{S}(\xi, \omega_k)}{\hat{S}(\xi + \Delta\xi, \omega_k)} \approx \log \frac{S_o(\xi, \omega_k)}{S_o(\xi + \Delta\xi, \omega_k)} \qquad \forall \omega_k. \tag{6}$$

Substituting in equation (1), we have

$$\Delta\hat{S}(\xi) \approx \Delta S_o(\xi) \tag{7}$$

as claimed. As a result of this, the spectral sensitivity, defined in equation (2), can also be written as

$$\eta(\xi) \approx \lim_{\Delta\xi \to 0} \left| \frac{\Delta S_o(\xi)}{\Delta\xi} \right|. \tag{8}$$

We now turn our attention to developing a quantization scheme which would minimize the maximum spectral deviation (i.e., $\max_{\xi} \Delta\hat{S}(\xi)$). One approach is to quantize the quantile orders using some non-uniform quantization scheme. In order to reduce the spectral deviation due to the quantization of the i^{th} quantile order E_i, we assign more steps near those values of E_i where the spectral sensitivity with

respect to E_i is higher and fewer steps near those values of E_i where the spectral sensitivity is lower.

Equivalently, we can form a new set of parameters $\{\xi_i\}$ from the quantile orders $\{E_i\}$ using some nonlinear transformation and then subject that to uniform quantization. Uniform quantization of the transformed parameters would be optimal only if the spectral sensitivity with respect to the transformed parameters were a constant. Otherwise, we can assign more quantization steps near those values of the transformed parameter where the spectral sensitivity is higher and fewer steps in the lower spectral sensitivity region. This would then lead to a smaller spectral deviation. So the problem of developing an optimal quantization scheme is equivalent to finding the transformation such that the spectral sensitivity with respect to the transformed parameters is a constant. Note that since this constant is arbitary, such a transformation can be found only to within a multiplicative constant.

Consider the following transformation:

$$\xi_o = 10 \log_{10} E_o \tag{9}$$

$$\xi_i = 10 \log_{10} \frac{2\pi(E_i - E_{i-1})}{N(\theta_i - \theta_{i-1})} \qquad 1 \leq i \leq q. \tag{10}$$

Note that the transformation merely gives the flat spectral density approximation expressed in dB. For this,

$$\Delta \hat{S}(\xi_o) \approx \Delta S_o(\xi_o)$$

$$= \Delta \xi_o \left[\frac{\log^p 10}{(1 + \frac{N}{2}) 10^p} \right]^{\frac{1}{p}} \tag{11}$$

and

$$\Delta \hat{S}(\xi_i) \approx \Delta S_o(\xi_i)$$

$$= \Delta\xi_i \left[\frac{N(\theta_i - \theta_{i-1}) \log^p 10}{2\pi(1 + \frac{N}{2})10^p} \right]^{\frac{1}{p}} \qquad 1 \leq i \leq q. \tag{12}$$

Thus, the spectral sensitivity with respect to the transformed parameters is given by

$$\eta(\xi_o) \approx \lim_{\Delta\xi_o \to 0} \left| \frac{\Delta S_o(\xi_o)}{\Delta \xi_o} \right|$$

$$= \left[\frac{\log^p 10}{(1 + \frac{N}{2})10^p} \right]^{\frac{1}{p}}$$

$$= \text{a constant}$$

$$\eta(\xi_i) \approx \lim_{\Delta\xi_i \to 0} \left| \frac{\Delta S_o(\xi_i)}{\Delta \xi_i} \right|$$

$$= \left[\frac{N(\theta_i - \theta_{i-1}) \log^p 10}{2\pi(1 + \frac{N}{2})10^p} \right]^{\frac{1}{p}}$$

$$= \text{a constant} \qquad 1 \leq i \leq q.$$

Hence, the transformation defined by (9) and (10) is the required one.

We note that while the quantile orders lie between 0 and 1, the transformed parameters are not restricted in range. But since we know that $E_q = 1.0$, we can fix one end of this range arbitarily. In our implementation, the maximum of all the $\{\xi_i\}$ was set to zero. Usually the flat spectral density approximation has a dynamic range of not more than 30 dB. So the other end of the range is fixed at $-D_R$ dB where $D_R \geq 30$. The complete quantization scheme can be summarized as follows:

Step 1: Compute ξ_i for $0 \leq i \leq q$ from the quantiles and quantile orders using equations (9) and (10).

Step 2: Let $\xi_{max} = \max_i \xi_i$. Then compute $\xi_i' = \max(\xi_i - \xi_{max}, -D_R)$.

Step 3: The parameters ξ_i' which lie in the range $[-D_R, 0]$ are quantized uniformly and transmitted.

The quantile orders can be computed from the transmitted parameters ξ_i' as follows:

Step 1: Compute

$$\overline{E}_o = 10^{\frac{\xi_o'}{10}}$$

$$\overline{E}_i = \overline{E}_{i-1} + \frac{N(\theta_i - \theta_{i-1})}{2\pi} 10^{\frac{\xi_i'}{10}} \qquad 1 \leq i \leq q.$$

Step 2: The quantile orders $\{E_i\}$ are then obtained by normalizing $\{\overline{E}_i\}$. That is, $E_i = \overline{E}_i / \overline{E}_q$ for all $0 \leq i \leq q$.

In our implementation of the 9.6 Kbits/s and 16 Kbits/s vocoders, we have a total of 11 quantile orders ($q = 10$). The value of D_R is taken to be 30 and the quantization step size used in the uniform quantizer is 2 dB. Thus, each ξ_i' is encoded using 4 bits. We see that a total of 44 bits are required to encode the quantile orders every frame. In 24 Kbits/s vocoder, we have a total of 14 quantile orders ($q = 13$). The value of D_R is taken to be 63 and the quantization step size is 1 dB. Here each ξ_i' is encoded using 6 bits. Here a total of 84 bits are required to encode the quantile orders every frame.

6.2 Encoding of quantiles

The quantiles $\{\theta_i\}$ themselves are all multiples of $\frac{2\pi}{N}$. The first quantile is $\theta_o = 0$

and the last quantile is $\theta_q = \pi$. The intermediate quantiles satisfy the inequality

$$\frac{2\pi}{N} \leq \theta_1 < \theta_2 \ \ldots \ < \theta_{q-1} \leq \frac{2\pi}{N}(\frac{N}{2} - 1).$$

The set of intermediate quantiles can be considered as an output symbol of a *source*, whose $(\frac{N}{2} - 1)$-bit symbols have exactly $q - 1$ ones and the remaining zeros. Clearly, there are $\binom{\frac{N}{2}-1}{q-1}$ symbols, so it would require $B_q = \left\lceil \log_2 \binom{\frac{N}{2}-1}{q-1} \right\rceil$ bits to encode each symbol ($\lceil x \rceil$ denotes the smallest integer greater than or equal to x). Thus, the total number of bits required to encode the quantiles is B_q bits.

In our implementation, $N = 512$ at all the bit rates. The total number of intermediate quantiles $(= q - 1)$ that were used to encode the spectral envelope at 4.8, 9.6, 16 and 24 Kbits/s are 8, 9, 9 and 12, respectively. Thus, the total number of bits B_q required to encode the quantiles for every frame is 49, 54, 54 and 67 bits at 4.8, 9.6, 16 and 24 Kbits/s, respectively.

6.3 Encoding of pulse locations

It is the autocorrelation-type multi-pulse algorithm that is used to estimate the pulse locations in our implementation of the 4.8 Kbits/s vocoder. In the autocorrelation-type multi-pulse algorithm, the estimated pulse locations n_1, n_2, ..., n_{N_q} are all distinct integers that lie between 1 and N_F (= the total number of samples in the frame). These pulse locations can always be arranged in increasing order and can therefore be encoded using $B_l = \left\lceil \log_2 \binom{N_F}{N_q} \right\rceil$ bits (as explained in the previous section). The total number of samples in each frame is $N_F = 256$ and the number of pulses estimated in each frame is $N_q = 9$ in the 4.8 Kbits/s vocoder. Thus, the total number of bits per frame required to encode the pulse locations is $B_l = 54$ bits.

It is the covariance-type multi-pulse algorithm that is used to estimate the pulse locations in our implementation of the 9.6, 16 and 24 Kbits/s vocoders. In the covariance-type multi-pulse algorithm, each frame is divided into L sub-frames, each containing $\overline{N}_F = \frac{N_F}{L}$ samples. In each sub-frame, $\overline{N}_q = \frac{N_q}{L}$ pulse locations are estimated. Thus we require $\left\lceil \log_2 \left(\frac{\overline{N}_F}{\overline{N}_q} \right) \right\rceil$ bits to encode the pulse locations in each sub-frame and $B_l = L \left\lceil \log_2 \left(\frac{\overline{N}_F}{\overline{N}_q} \right) \right\rceil$ bits for the entire frame. At all three bit rates used with the covariance-type algorithm, there are $N_F = 256$ samples per frame and $L = 4$ sub-frames; hence, we require $\overline{N}_F = 64$ samples per sub-frame. The value of $\overline{N}_q$ is 6 at 9.6 Kbits/s, 9 at 16 Kbits/s and 13 at 24 Kbits/s. Thus, the total number of bits B_l required to encode the pulse locations for the entire frame is 108, 140 and 176 bits at 9.6, 16 and 24 Kbits/s, respectively.

6.4 Quantization and encoding of pulse amplitudes

Let the largest magnitude of all the pulse amplitudes be a_{max} and let the number of bits used to encode each pulse amplitude be B_p bits. Then the pulse amplitudes are all multiplied by $a_{max}^{-1}(1 - 2^{-(B_p-1)})$. This ensures that the pulse amplitudes are in the range $[-(1 - 2^{-(B_p-1)}), (1 - 2^{-(B_p-1)})]$. The scaled pulse amplitudes are then quantized uniformly using a quantizer step size of $2^{-(B_p-1)}$. The scaling is compensated by modifying the gain G of the linear filter $Q(z)$. The modified gain G' is thus

$$G' = G \cdot \frac{a_{max}}{1 - 2^{-(B_p-1)}}. \tag{13}$$

In our implementation, 4 bits are used to encode each pulse amplitude at 4.8, 9.6 and 16 Kbits/s, while 5 bits are used at 24 Kbits/s. There are 9, 24, 36 and 52 pulses per frame at 4.8, 9.6, 16 and 24 Kbits/s. Thus, a total of 36, 96, 144 and 260 bits are used per frame to encode all the pulse amplitudes at 4.8, 9.6, 16 and

24 Kbits/s, respectively.

6.5 Quantization and encoding of gain

The modified gain G' can be considered as a parameter of the linear filter $Q(z)$ which accounts for the spectral envelope of the speech segment:

$$Q(z) = \frac{G}{A(z)H_p(z)} = \frac{G'(1-2^{-(B_p-1)})}{a_{max}A(z)H_p(z)}.$$

One can define, as in equation (1), the spectral deviation $\Delta Q(\xi)$, in the power spectrum of the linear filter $Q(z)$, with respect to a perturbation $\Delta\xi$ in some parameter ξ of the linear filter. Thus,

$$\Delta Q(\xi) = \left[\frac{1}{1+\frac{N}{2}} \sum_{\substack{k=0 \\ \omega_k=2\pi k/N}}^{N/2} \left|\log \frac{|Q(e^{j\omega},\xi)|^2}{|Q(e^{j\omega},\xi+\Delta\xi)|^2}\right|^p\right]^{\frac{1}{p}}. \tag{14}$$

The parameter ξ could be one of the quantile orders or the gain or any function of them.

As an aside, we note that any quantization scheme for the quantile orders which minimizes the spectral deviation in the spectral envelope estimate also minimizes the spectral deviation in the power spectrum of $Q(z)$. This is because

$$|Q(e^{j\omega},\xi)|^2 = \frac{\left(G'(1-2^{-(B_p-1)})\right)^2}{a_{max}^2|H_p(e^{j\omega})|^2|A(e^{j\omega},\xi)|^2} = \frac{\left(G'(1-2^{-(B_p-1)})\right)^2}{a_{max}^2|H_p(e^{j\omega})|^2}\hat{S}(\omega,\xi)$$

where ξ is one of the quantile orders or some function of them. Hence,

$$\Delta Q(\xi) = \Delta\hat{S}(\xi).$$

Thus the quantization scheme proposed in Section 6.1 for the quantile orders is optimal not only in the sense of minimizing the maximum spectral deviation in the

spectral envelope estimate but also in minimizing the maximum spectral deviation in the power spectrum of the linear filter $Q(z)$.

Referring to our prior considerations, if the parameter ξ is the modified gain G', then

$$\Delta Q(G') = \left[\frac{1}{1+\frac{N}{2}} \sum_{\substack{k=0 \\ \omega_k = 2\pi k/N}}^{N/2} \left|\log \frac{|G'|^2}{|G'+\Delta G'|^2}\right|^p\right]^{\frac{1}{p}}$$
$$= \left|\log\left(1+\frac{\Delta G'}{G'}\right)^2\right|$$

and the spectral sensitivity with respect to G' is therefore

$$\eta(G') = \lim_{\Delta G' \to 0}\left|\frac{\Delta Q(G')}{\Delta G'}\right| = \left|\frac{2}{G'}\right|.$$

Now as before, the optimal quantization scheme for G' is first to transform the gain and uniformly quantize the transformed parameter. The transformation, as explained in Section 6.1, must be such that the spectral sensitivity with respect to the transformed parameter is a constant. Consider the transformation $\xi = 20\log_{10} G'$, so that G' is expressed in dB. Then

$$\eta(\xi) = \eta(G')\left|\frac{dG'}{d\xi}\right|$$
$$= \left|\frac{2}{G'} \cdot \frac{G' \log 10}{20}\right|$$
$$= \frac{\log 10}{10}$$

which is a constant. So the transformation is the required one.

In practice, the modified gain G', expressed in dB, is seldom below 0 dB or above 200 dB and is therefore limited to the range $[0,\ 200(1-2^{-B_G})]$ dB, where B_G is the number of bits used to encode the gain. If G' is below 0 dB, then it is assumed to be 0 dB. If it is above $200(1-2^{-B_G})$ dB, then it is taken to be $200(1-2^{-B_G})$. It

is then quantized uniformly using a step size of $200 \cdot 2^{-B_G}$ dB. The number of bits used to encode the gain in our implementation is 11 at 4.8 and 9.6 Kbits/s and 13 at 16 and 24 Kbits/s.

6.6 Quantization and encoding of pitch predictor parameters

The pitch predictor parameters are M_p and μ. As explained in Section 5.3, the value of M_p is restricted to the range [22, 149] at all the bit rates. So it takes exactly 7 bits to encode M_p. But the magnitude of μ must be less than one to ensure stability of the pitch predictor. Thus, if B_μ bits are used to encode μ, then μ is actually restricted to lie in the range $[-(1-2^{-(B_\mu-1)}), (1-2^{-(B_\mu-1)})]$. If μ has a value greater than $(1-2^{-(B_\mu-1)})$, then it is assumed to be $(1-2^{-(B_\mu-1)})$. If μ has a value less than $-(1-2^{-(B_\mu-1)})$, then it is assumed to be $-(1-2^{-(B_\mu-1)})$. The parameter μ is quantized uniformly using a step size of $2^{-(B_\mu-1)}$. In our implementation, 6 bits are used to encode μ at 4.8 Kbits/s and 7 bits at 9.6, 16 and 24 Kbits/s.

This concludes our discussion of the quantization and encoding of various parameters in the quantile vocoder. A table of bit allocations at bit rates 4.8, 9.6, 16 and 24 Kbits/s is displayed in Fig. 6.2.

6.7 Results

The spectral envelope estimate and the synthesized speech waveform of the quantile vocoder for a typical speech frame at bit rates 4.8, 9.6, 16 and 24 Kbits/s are displayed in Fig. 6.3-6.6. Fig. 6.3 has two parts. One part contains the power spectral density of a Hamming-windowed (see Section 2.2) preemphasized 256-point speech frame overlain by a scaled version of the spectral envelope estimate at 24

Kbits/s. The scaling factor is chosen so that the total power under the spectral envelope estimate is equal to the total power under the power spectral density of the Hamming-windowed preemphasized speech frame. The second part contains the speech waveform of the same 256-point speech frame and is overlain by the synthesized speech waveform at 24 Kbits/s. Figs. 6.4-6.6 are similar except that the corresponding bit rates are 16, 9.6 and 4.8 Kbits/s, respectively.

Bit rate in Kbits/s	4.8	9.6	16	24
Sampling rate in KHz	7.5	7.5	10	10
Frame size in samples	256	256	256	256
Frame size in ms	34.13	34.13	25.6	25.6
Number of quantiles	9	11	11	14
Number of excitation pulses	9	24	36	52
Bits used for quantile orders	0	44	44	84
Bits used for quantiles	49	54	54	67
Bits used for pulse locations	54	108	140	176
Bits used for pulse amplitudes	36	96	144	260
Bits used for gain	11	11	13	13
Bits used for pitch model parameters	13	14	14	14
Total number of bits used per frame	163	327	409	614

Fig. 6.2 Bit allocation in each frame at various bit rates

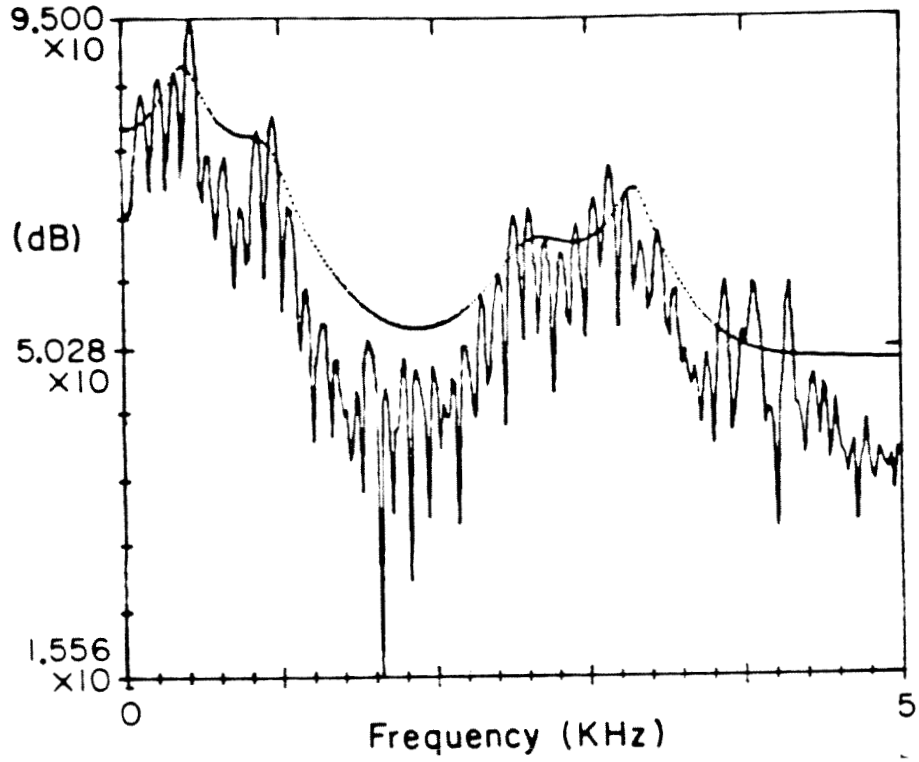


Fig. 6.3(a) Spectral envelope estimate at 24 Kbits/s

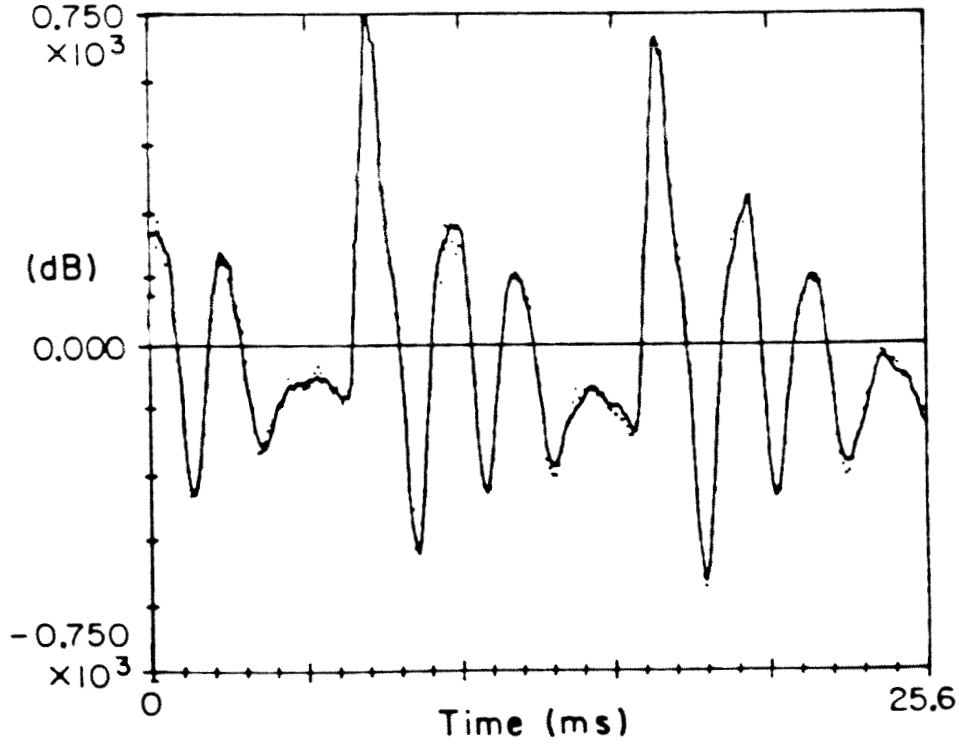


Fig. 6.3(b) Speech waveform overlaid by synthesized speech waveform at 24 Kbits/s

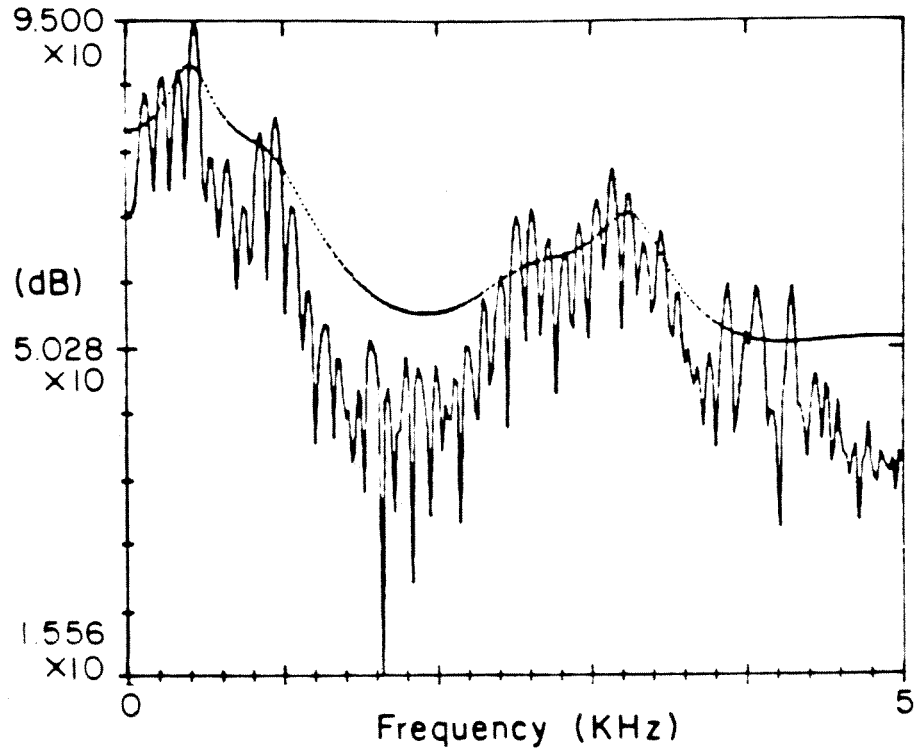


Fig. 6.4(a) Spectral envelope estimate at 16 Kbits/s

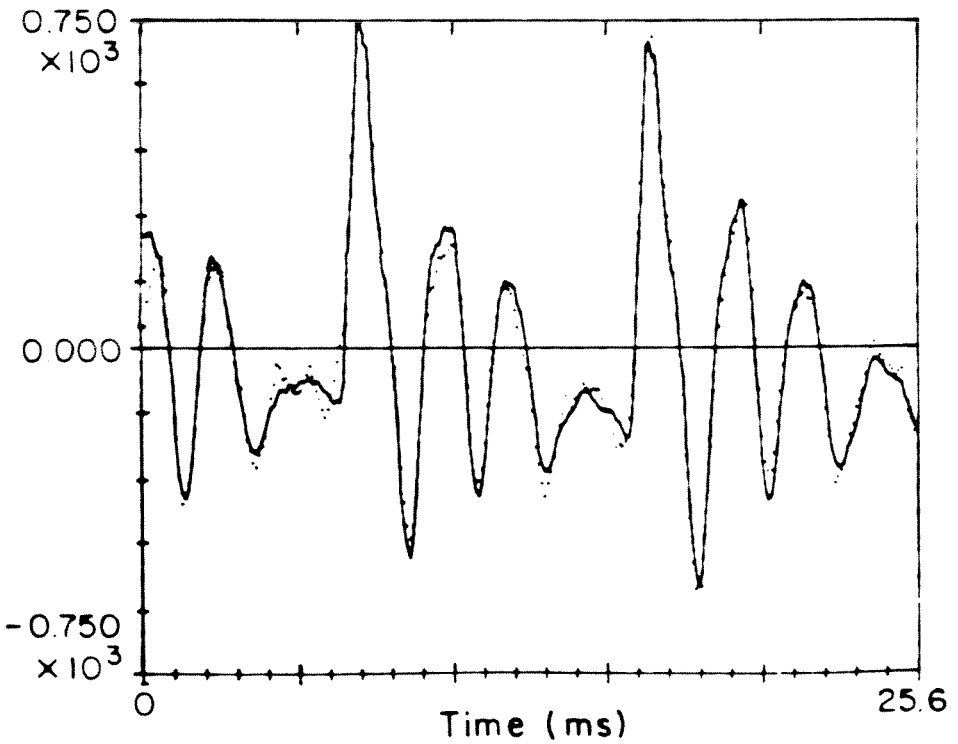


Fig. 6.4(b) Speech waveform overlaid by synthesized speech waveform at 16 Kbits/s

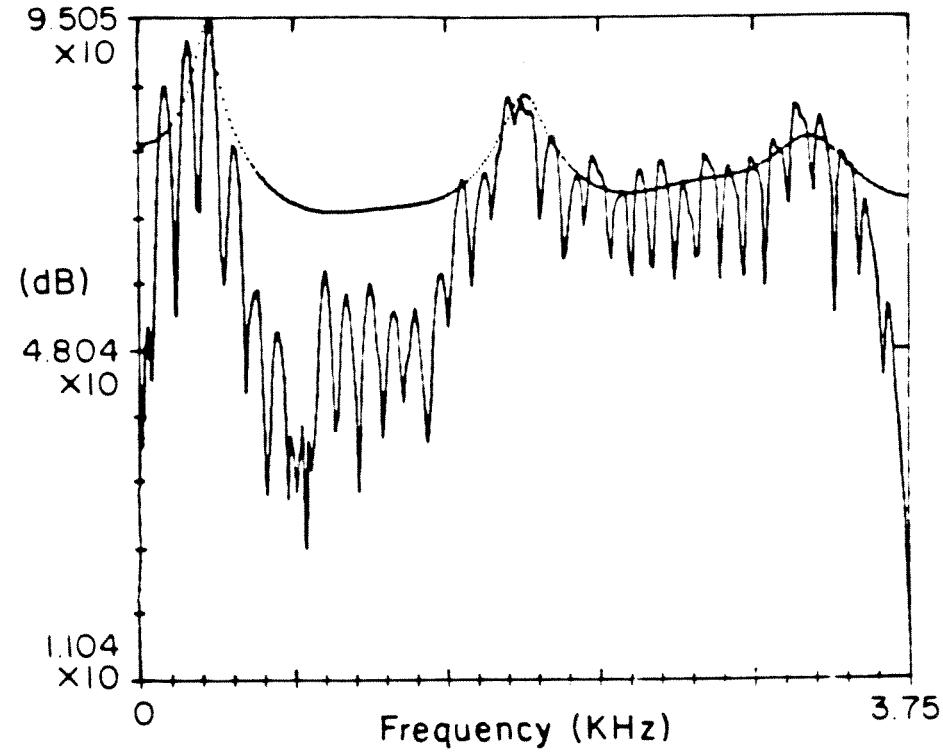


Fig. 6.5(a) Spectral envelope estimate at 9.6 Kbits/s

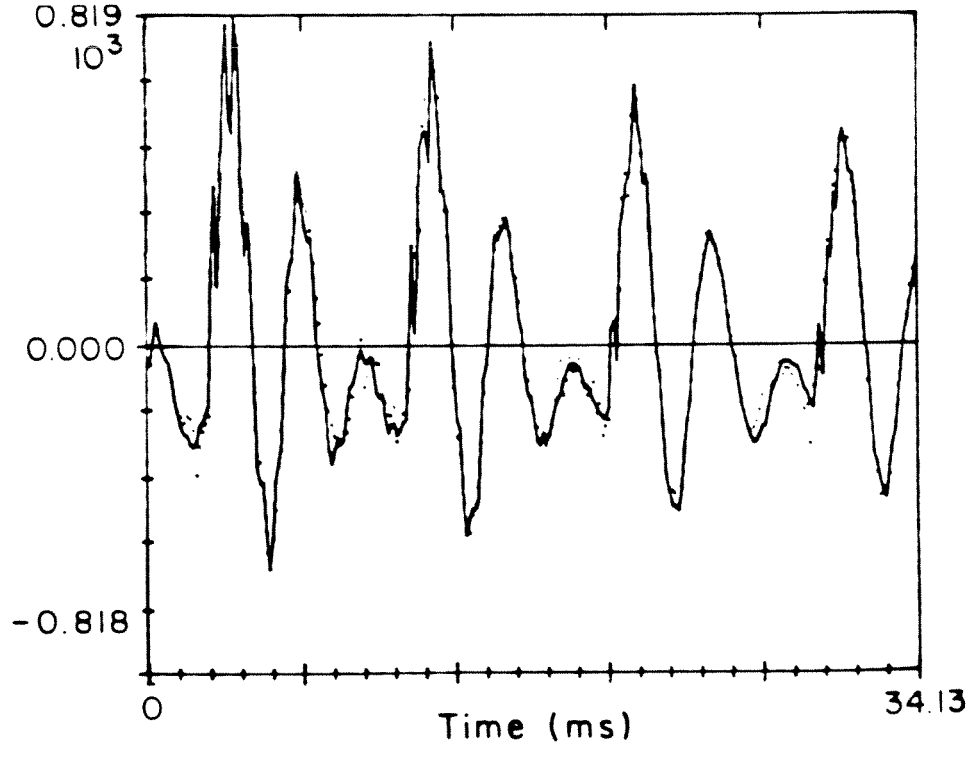


Fig. 6.5(b) Speech waveform overlaid by synthesized speech waveform at 9.6 Kbits/s

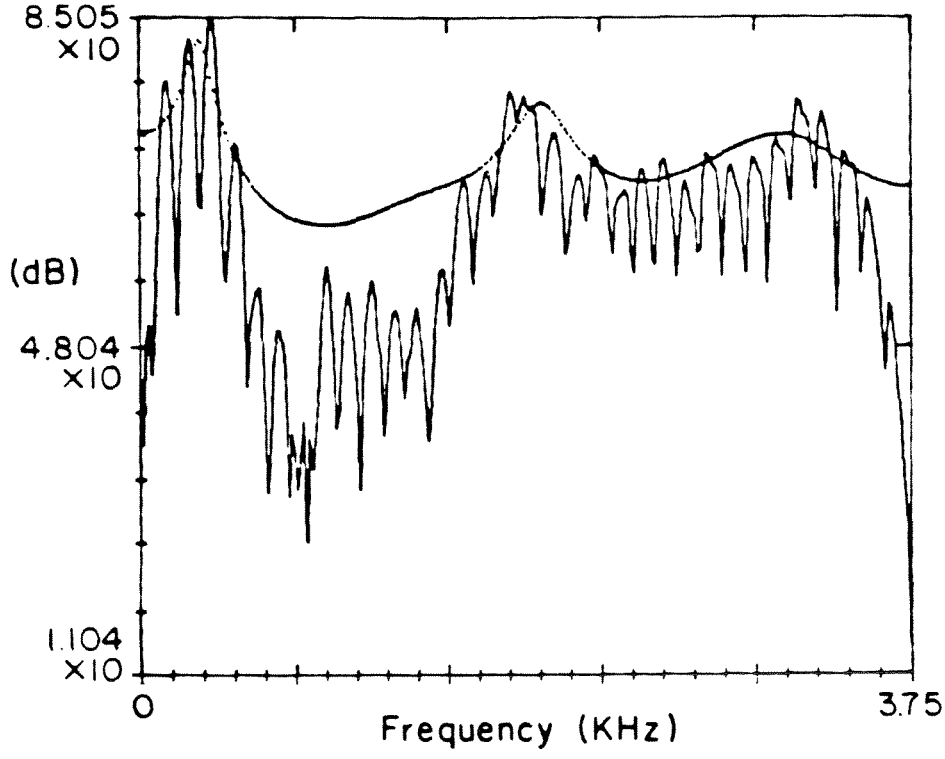


Fig. 6.6(a) Spectral envelope estimate at 4.8 Kbits/s

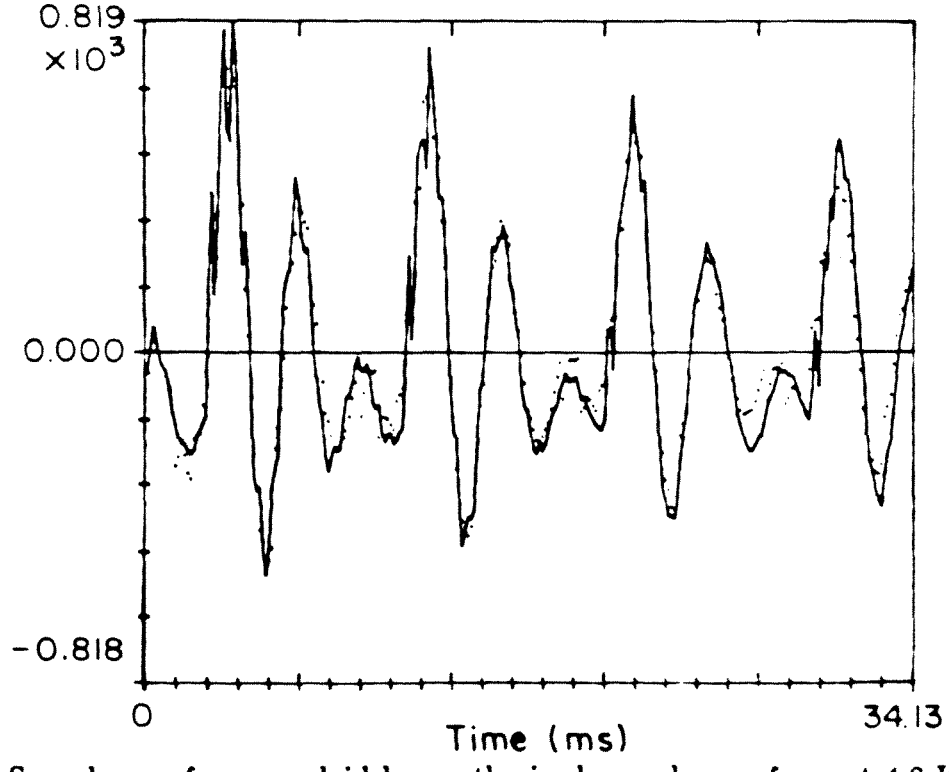


Fig. 6.6(b) Speech waveform overlaid by synthesized speech waveform at 4.8 Kbits/s

CHAPTER 7

EVALUATION OF THE QUANTILE VOCODER

In this chapter, we consider the evaluation of the quantile vocoder at various bit rates. Ideally, we would like a performance measure that is not only subjectively meaningful but also repeatable; i.e., the same performance measure must be obtained on repetitions of the same experiment. Unfortunately, such a performance measure does not exist for evaluating the quality of the speech produced by a vocoder.

Objective measures of vocoder performance (see Appendix E of [65]) are repeatable. They usually are refinements of the conventional signal-to-noise ratio. These refinements have been proposed to ensure that the objective measures are more representative of the quality of the synthesized speech. Despite these refinements, objective measures can never be completely descriptive of perceived speech quality and so can only partially describe the performance of the vocoder.

Subjective measures of vocoder performance (see Appendix F of [65]) are directly related to the quality of the synthesized speech and are therefore truly meaningful. However, they are not repeatable and therefore not very reliable. The reliability can be improved only by using a large speech data base, a large number of subjects and more complicated test procedures to assess the quality of the vocoder. Even then complete reliability cannot be guaranteed.

Thus both objective and subjective performance measures have their shortcomings. So in practice, both are required in order to assess vocoder performance properly. We begin in Section 7.1 by first describing the experimental details that are involved in setting up the speech data base as well as in recording the compressed speech. In Section 7.2, an objective performance measure, the so-called *segmental*

signal-to-noise ratio, is defined and is used to assess the quantile vocoder at all bit rates. In Section 7.3, a formal subjective evaluation is proposed and carried out.

7.1 Experimental details

In order to evaluate the quantile vocoder, ten sentences spoken by one male and one female speaker were used. The sentences are a subset of the grammatical sentences listed in [66]. The sentences, each about 4 seconds long, were spoken through a microphone, lowpass filtered up to 4.8 KHz, and digitized at a sampling rate of 10 KHz using a 12 bit A/D converter. The speech samples are all thus integers that lie between -2048 and 2047. The sentences were digitized at the Speech Laboratory at Indiana University. The digitized speech data were then transferred by tape to an 80 Mbyte disk in the Acoustic Signal Processing Facility at Caltech.

For both 4.8 and 9.6 Kbits/s, we require speech sampled at 7.5 KHz. A method for conversion of sampling rate by a rational fraction $\overline{L}/\overline{M}$ (both $\overline{L}$ and $\overline{M}$ are integers) is described in Fig. 7.1 (see Chapter 2 of [67]). In this method, the speech samples are first passed through an $\overline{L}$-fold sampling rate expander; i.e., each input speech sample is padded through a lowpass filter. This filter approximates an ideal response of $\overline{L}$ in the frequency range $[0, \min(\pi/\overline{L}, \pi/\overline{M})]$, and is zero elsewhere. Finally, the output of the lowpass filter is passed through a $\overline{M}$-fold sampling rate compressor which picks every $\overline{M}^{th}$ sample and discards the rest. In our application, the sampling rate conversion from 10 KHz to 7.5 KHz was accomplished by choosing $\overline{L} = 3$ and $\overline{M} = 4$. The lowpass filter used was a linear phase FIR filter which was designed by the Parks-McClellan algorithm ([68]). The designed filter had a deviation of 0.08 dB in the passband $[0, 0.23\pi]$ and an attenuation of -40.65 dB in the stopband $[0.25\pi, \pi]$. The filter length was 201. After data compression, the

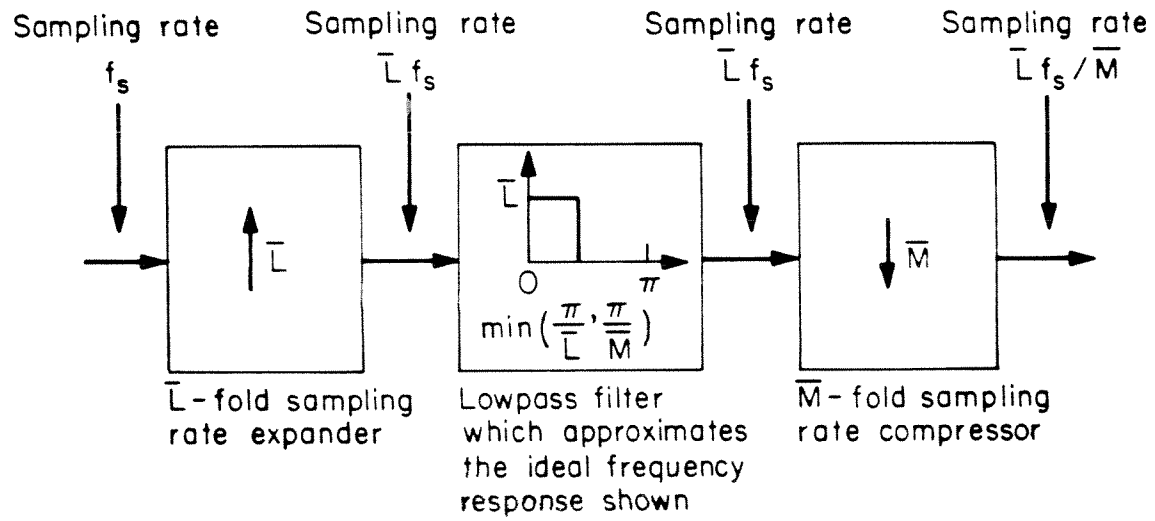


Fig. 7.1 Sampling rate conversion by a rational fraction $\overline{L}/\overline{M}$

sampling frequency has to be reconverted to 10 KHz before conversion to analog. This was accomplished by choosing $\overline{L} = 4$ and $\overline{M} = 3$ and the same lowpass filter.

The complete speech recording sequence is shown in Fig. 7.2. The synthesized speech sentences, which were stored in an 80 Mbyte disk, were converted to analog by a 16 bit D/A converter using an external 10 KHz clock. The output of the D/A is a staircase waveform and is smoothed using a 8-pole Butterworth filter with cutoff at 5 KHz. It is then preamplified and recorded on a high quality low-noise tape. The recording can be monitored using headphones or speakers connected to the preamplifier.

The recording was done using the Dolby noise reduction scheme so as to reduce the effects of tape noise. The synthesized speech samples, as do the original speech samples, lie between -2048 and 2047. However, the D/A used has 16 bits and can therefore accept speech samples that lie in the range -16384 to 16383. So before recording, the speech samples can be scaled up by a factor less than or equal to 16. The advantage of scaling is that while it does not affect the degradation introduced by the data compression, it improves the signal-to-(output device) noise ratio. In our experience, a scale factor of 8 is adequate to ensure clean recording.

7.2 Objective evaluation of quantile vocoder

An important and widely used objective measure of vocoder performance is the *segmental signal-to-noise ratio* (see Appendix E.2 in [65] and also [69]-[71]), denoted by $SNRSEG$. The segmental signal-to-noise ratio is expressed in dB and is defined as

$$SNRSEG = E[SNR(m)]$$

where $SNR(m)$ is the conventional signal-to-noise ratio in dB for segment (or

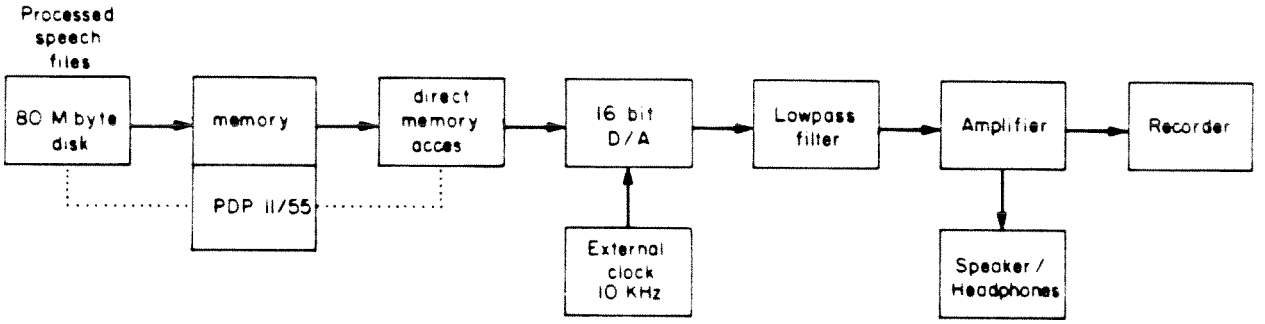


Fig. 7.2 Speech recording sequence

frame) m, and the expectation is in practice a time average over all segments of interest. In our work, all the segments in the ten sentences have been included in computing the $SNRSEG$ for each speaker at every bit rate. This performance measure is preferred for vocoders to the conventional signal-to-noise ratio because it takes into account the fact that the same amount of noise has different perceptual effects depending on the signal level.

Fig. 7.3 contains a table of values of $SNRSEG$ at 24, 16, 9.6 and 4.8 Kbits/s. At each bit rate, the segmental signal-to-noise ratio for the male speaker, the female speaker and the overall average is presented. The segment size used in computing the $SNRSEG$ is just the frame size. Thus for 16 and 24 Kbits/s, the segment size is 25.6ms. For 4.8 and 9.6 Kbits/s, the segment size is 34.13ms.

A plot of the SNR and the speech power (in dB) for each segment versus the segment number is given in Fig. 7.4 at all the bit rates for a sentence spoken by a female speaker. This gives us a rough idea of the fluctuation of the signal-to-noise ratio from segment to segment in a sentence at various bit rates.

7.3 Subjective evaluation of quantile vocoder

In this section, we will describe a formal subjective test in order to assess the quality of the speech synthesized by the quantile vocoder at various bit rates. This test is referred to in the literature as the *mean opinion score test.* Our treatment closely follows the one given in Appendix F.1 in [65]. Another excellent reference is the paper by Daumer ([72]).

In the mean opinion score test, several subjects are recruited and each of them classifies the synthesized speech on a 5 point scale for speech quality or speech

Bit rate (Kbits/s)	*SNRSEG* (dB) Male speaker	*SNRSEG* (dB) Female speaker	*SNRSEG* (dB) Overall
4.8	8.64	8.31	8.48
9.6	12.96	12.00	12.48
16	14.52	13.65	14.09
24	17.55	16.38	16.96

Fig. 7.3 Segmental signal-to-noise ratio at various bit rates

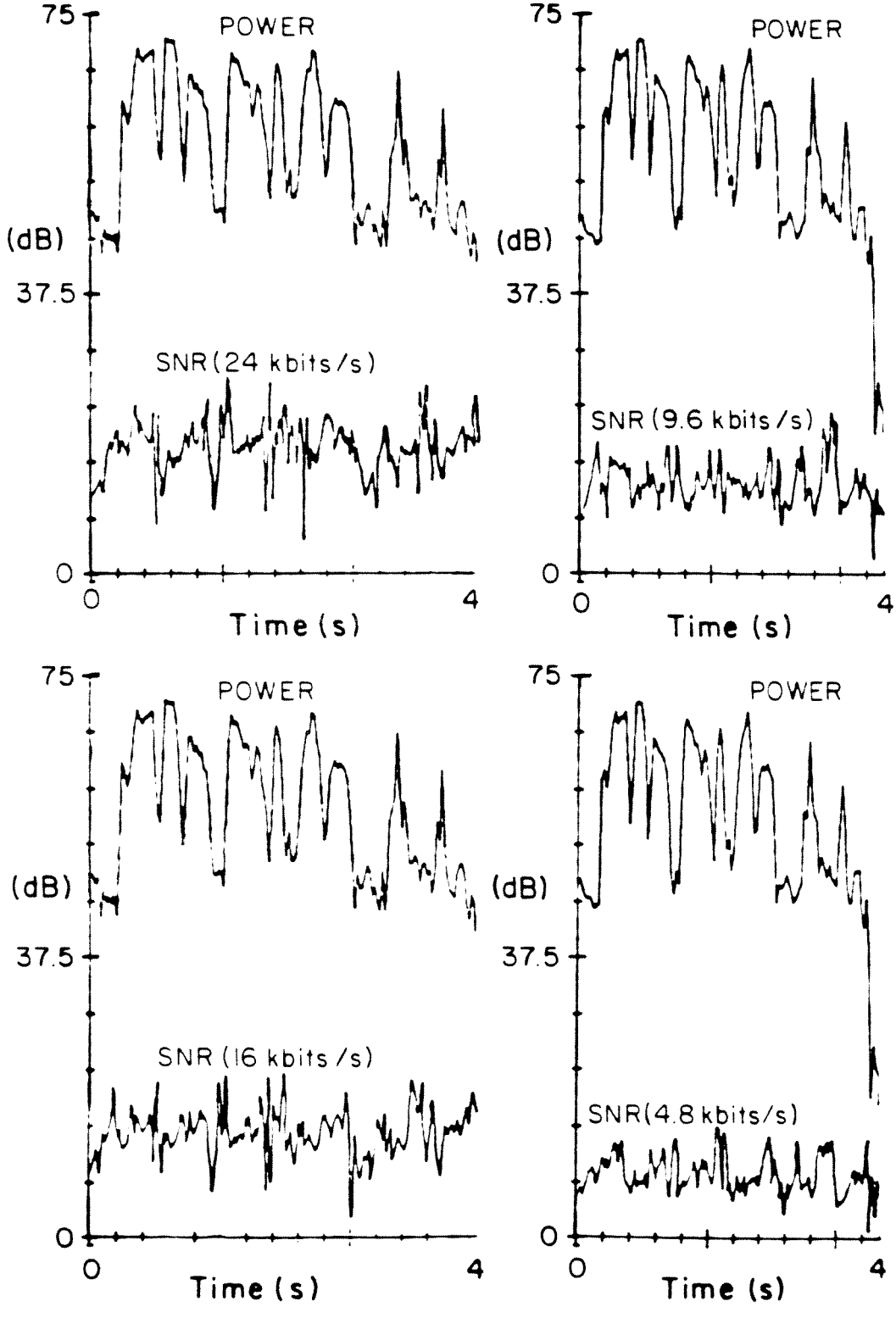


Fig. 7.4 Plot of SNR and speech power for successive time frames for the sentence

"The Holy Bible inspired a deep reverence"

spoken by a female speaker

impairment. Thus, the speech quality is judged as excellent, good, fair, poor or bad. These categories correspond to speech impairment which is imperceptible, just perceptible but not annoying, annoying but not objectionable, or very annoying and objectionable. These categories are also associated with numbers, so that judgements can be on a scale of , say, 1 to 5. The five scale steps for speech quality and impairment and the associated number scores are presented in the form of a table in Fig. 7.5.

The scores from these tests are averaged over all the subjects, speakers and sentences spoken by each speaker. This pooled average judgement is called the *mean opinion score (MOS)* for the ensemble of listeners, speakers and sentences. Since MOS values are very difficult to duplicate in repetitions of an experiment, the standard deviation σ_{MOS} of the MOS value across the population of subjects, talkers and sentences is very useful in assessing the repeatability of any MOS rating.

In our work, we have used the ten sentences spoken by one male speaker and one female speaker and six subjects to obtain an MOS rating at all the bit rates for the quantile vocoder as well as for 7, 6, 5, 4 and 3 bits/sample (10 KHz sampling rate) μ-255 law (Section 5.3.2, [3]) PCM coders, as an initial comparison of our vocoder. The output speech samples of the μ-255 law PCM coders were obtained as follows. The input speech data, which lie between -2048 and 2047, are first passed through a μ-255 law compander. Thus, if x_n is the input speech sample, then the companded speech sample $\bar{x}_n$ is given by

$$\bar{x}_n = \frac{\text{sign}(x_n) \cdot 2048}{\log 256} \cdot \log\left(1 + \frac{255\,|x_n|}{2048}\right).$$

We note that $\bar{x}_n$ also lies between -2048 and 2047. The companded speech sample is then quantized uniformly using 7, 6, 5, 4 and 3 bits. After quantization, the companded speech sample $\hat{\bar{x}}_n$ is expanded according to

Number scores	Speech quality scale	Speech impairment scale
5	Excellent	Imperceptible
4	Good	Just perceptible but not annoying
3	Fair	Perceptible and slightly annoying
2	Poor	Annoying but not objectionable
1	Unsatisfactory (Bad)	Very annoying and objectionable

Fig. 7.5 Five scale steps for speech quality and impairment and associated number scores

$$\hat{x}_n = \frac{\text{sign}(\hat{\bar{x}}_n) \cdot 2048}{255} \cdot \{\exp(\frac{|\hat{\bar{x}}_n| \log 256}{2048}) - 1\}.$$

Finally $\hat{x}_n$ is rounded off to the nearest integer to produce the output speech sample of the μ-255 law PCM coder.

The mean opinion score test is administered as follows. Each speech sentence synthesized by the quantile vocoder at 4.8, 9.6, 16 and 24 Kbits/s as well as by the 7, 6, 5, 4 and 3 bits/sample (10 KHz sampling rate) μ-255 law PCM coders is recorded in a random order. A tape containing ten such sentences spoken by one male and one female speaker is provided to the subject who assigns a score between 1 and 5 for each sentence. The subject is allowed to listen to any sentence as often as he or she desires. The MOS rating and the associated standard deviation for both the male and the female speaker as well as the overall MOS rating are determined for the quantile vocoder at 4.8, 9.6, 16 and 24 Kbits/s and for the μ-255 law PCM coder at the five quantization levels corresponding to 70, 60, 50, 40 and 30 Kbits/s. The results are presented in the form of a table in Fig. 7.6.

The conclusions of the subjective evaluation tests can be summarized as follows. The MOS rating for the 4.8 Kbits/s quantile vocoder is higher than the MOS rating for the 30 Kbits/s (3 bits/sample) μ-255 law PCM coder for the male speaker but lower for the female speaker. The overall MOS rating for the 4.8 Kbits/s quantile vocoder is marginally higher than the overall MOS rating for the 30 Kbits/s μ-255 law PCM coder. The MOS rating for the 9.6 Kbits/s quantile vocoder lies between the MOS ratings of the 30 and 40 Kbits/s μ-255 law PCM coders for both the male and the female speaker. The MOS rating for the 16 Kbits/s quantile vocoder lies between the MOS ratings of the 40 and 50 Kbits/s μ-255 law PCM coders for both the male and the female speaker. The MOS rating for the 24 Kbits/s quantile

vocoder lies between the MOS ratings of the 50 and 60 Kbits/s μ-255 law PCM coders for both the male and the female speaker.

Bit rate (Kbits/s)	MOS (Male)	σ_{MOS} (Male)	MOS (Female)	σ_{MOS} (Female)	MOS (Overall)	σ_{MOS} (Overall)
4.8 (Quantile)	1.62	0.77	1.22	0.40	1.42	0.65
9.6 (Quantile)	2.17	0.69	2.00	0.71	2.09	0.70
16 (Quantile)	3.18	0.80	2.87	0.82	3.03	0.83
24 (Quantile)	3.82	0.60	3.68	0.66	3.75	0.64
30 (PCM)	1.40	0.66	1.38	0.59	1.39	0.63
40 (PCM)	2.58	0.73	2.70	0.53	2.64	0.64
50 (PCM)	3.55	0.69	3.52	0.83	3.54	0.77
60 (PCM)	4.50	0.67	4.45	0.74	4.48	0.71
70 (PCM)	4.70	0.64	4.73	0.54	4.72	0.59

Fig. 7.6 Results of the subjective evaluation tests

CHAPTER 8

SUMMARY AND CONCLUSIONS

In this thesis, a new speech compression scheme, the quantile vocoder, was investigated. The basic idea behind this new speech compression scheme is the encoding of the spectral envelope using quantiles. Algorithms to reestimate the spectral envelope from the quantiles and the quantile orders were developed. A multi-pulse excitation model in cascade with a 1-tap pitch predictor model was used to model the excitation. Algorithms to estimate the parameters of the excitation model were reviewed, evaluated, and implemented. Quantization schemes for the transmission parameters of the quantile vocoder were developed. The quantile vocoder was implemented at 4.8, 9.6, 16 and 24 Kbits/s. The segmental signal-to-noise ratio, an objective performance measure, and the mean opinion score, a subjective performance measure, were used to evaluate the vocoder at these bit rates.

The performance of the vocoder at 4.8, 9.6, 16 and 24 Kbits/s, based on the segmental signal-to-noise ratio, the mean opinion score and informal listening tests, has been found to be very promising, especially at 4.8 Kbits/s. But further development is necessary, especially at low bit rates, before the quality of the speech synthesized by the quantile vocoder becomes acceptable for many commercial applications.

APPENDIX A

Merchant-Parks method for solving Toeplitz plus Hankel system of equations

In this appendix we will briefly describe an efficient method for solving Toeplitz plus Hankel system of equations. This method is due to Merchant and Parks ([34]). The central idea in their method is to convert the Toeplitz plus Hankel matrix into a block Toeplitz matrix and then employ block Levinson algorithm ([52]).

We first introduce some notation. Let $\mathbf{A}$ be any $(M+1) \times (M+1)$ matrix and $\mathbf{c}$ be a $(M+1)$ dimensional vector. Define the *exchange operator* $\mathbf{J}$ as the $(M+1) \times (M+1)$ matrix

$$\mathbf{J} = \begin{pmatrix} 0 & 0 & \dots & 0 & 1 \\ 0 & 0 & \dots & 1 & 0 \\ \vdots & \vdots & & \vdots & \vdots \\ 0 & 1 & \dots & 0 & 0 \\ 1 & 0 & \dots & 0 & 0 \end{pmatrix}.$$

Define the following:

$\mathbf{A}^T$ = transpose of $\mathbf{A}$. If $\mathbf{A}^T = \mathbf{A}$, $\mathbf{A}$ is said to be *symmetric*.

$\mathbf{A}^x$ = cross transpose of $\mathbf{A}$ around main cross diagonal. If $\mathbf{A}^x = \mathbf{A}$, $\mathbf{A}$ is said to be *persymmetric*.

$\mathbf{c}_+ = \mathbf{c} = [\, c_o \; c_1 \; \dots \; c_M \,]^T$

$\mathbf{c}_- = \mathbf{J}\mathbf{c} = [\, c_M \; c_{M-1} \; \dots \; c_o \,]^T$

$\mathbf{T}$ = Toeplitz matrix, i.e. $\{\mathbf{T}\}_{ij} = t(i-j)$, a function of $i-j$ only.

$\mathbf{H}$ = Hankel matrix, i.e. $\{\mathbf{H}\}_{ij} = h(i+j)$, a function of $i+j$ only.

Note that $\mathbf{J}^2 = \mathbf{I}$ = identity matrix, and $\mathbf{JAJ} = \mathbf{A}^{Tx} = \mathbf{A}^{xT}$. The operations $(.)^T$ and $(.)^x$ commute. Further, if $\mathbf{B}$ is any other $(M+1) \times (M+1)$ matrix, then

$$(\mathbf{AB})^{Tx} = \mathbf{JABJ} = \mathbf{JAJ} \cdot \mathbf{JAJ} = \mathbf{A}^{Tx}\mathbf{B}^{Tx}.$$

Note that a Toeplitz matrix $\mathbf{T}$ is persymmetric and a Hankel matrix $\mathbf{H}$ is symmetric. We define a $(2M+2) \times (2M+2)$ *interleaving operator* $\mathbf{Q}$ such that

$$\{\mathbf{Q}\}_{ij} = \begin{cases} 1 & \text{if } i = 2r,\ j = r,\ 0 \le r \le M\ ; \\ 1 & \text{if } i = 2r+1,\ j = M+r+1,\ 0 \le r \le M\ ; \\ 0 & \text{otherwise.} \end{cases}$$

Note that $\mathbf{Q}^T\mathbf{Q} = \mathbf{Q}\mathbf{Q}^T = \mathbf{I}$. If we operate on a $(2M+2)$-dimensional vector $\mathbf{p}$ with $\mathbf{Q}$, then $\mathbf{Q}$ simply interleaves p_r and p_{M+r+1} for $0 \le r \le M$. That is,

$$\mathbf{Q}[\ p_o\ p_1\ \cdots\ p_M\ p_{M+1}\ \cdots\ p_{2M+1}\ ]^T = [\ p_o\ p_{M+1}\ p_1\ p_{M+2}\ \cdots\ p_M\ p_{2M+1}\ ]^T.$$

Now let us consider a Toeplitz plus Hankel system of equations

$$(\mathbf{T}+\mathbf{H})\mathbf{c} = \mathbf{b}. \tag{A1}$$

Write $(A1)$ as two different equations:

$$\mathbf{T}\mathbf{c} + \mathbf{H}\mathbf{J}\cdot\mathbf{J}\mathbf{c} = \mathbf{b} \tag{A2a}$$

$$\mathbf{J}\mathbf{T}\mathbf{J}\cdot\mathbf{J}\mathbf{c} + \mathbf{J}\mathbf{H}\mathbf{c} = \mathbf{J}\mathbf{b} \tag{A2b}$$

or in matrix form as

$$\begin{pmatrix} \mathbf{T} & \mathbf{H}\mathbf{J} \\ \mathbf{J}\mathbf{H} & \mathbf{J}\mathbf{T}\mathbf{J} \end{pmatrix} \begin{pmatrix} \mathbf{c}_+ \\ \mathbf{c}_- \end{pmatrix} = \begin{pmatrix} \mathbf{b}_+ \\ \mathbf{b}_- \end{pmatrix}. \tag{A3}$$

Since $\mathbf{T}$ is persymmetric $\mathbf{J}\mathbf{T}\mathbf{J} = \mathbf{T}^{Tr} = \mathbf{T}^T$. Denoting $\mathbf{H}\mathbf{J}$ by $\mathbf{T}_H$, we note that $\mathbf{T}_H^T = (\mathbf{H}\mathbf{J})^T = \mathbf{J}^T\mathbf{H}^T = \mathbf{J}\mathbf{H}$. So $(A3)$ can also be written as

$$\begin{pmatrix} \mathbf{T} & \mathbf{T}_H \\ \mathbf{T}_H^T & \mathbf{T}^T \end{pmatrix} \begin{pmatrix} \mathbf{c}_+ \\ \mathbf{c}_- \end{pmatrix} = \begin{pmatrix} \mathbf{b}_+ \\ \mathbf{b}_- \end{pmatrix}. \tag{A4}$$

We now note that the matrix $\mathbf{T}_H = \mathbf{H}\mathbf{J}$ is a Toeplitz matrix with $\{\mathbf{T}_H\}_{ij} = h(M+i-j)$. Each block matrix in $(A4)$ is thus a Toeplitz matrix. Finally, using the interleaving operator $\mathbf{Q}$ on $(A4)$, we get

$$\mathbf{Q}\begin{pmatrix} \mathbf{T} & \mathbf{T}_H \\ \mathbf{T}_H^T & \mathbf{T}^T \end{pmatrix}\mathbf{Q}^T \cdot \mathbf{Q}\begin{pmatrix} \mathbf{c}_+ \\ \mathbf{c}_- \end{pmatrix} = \mathbf{Q}\begin{pmatrix} \mathbf{b}_+ \\ \mathbf{b}_- \end{pmatrix}.$$

What we have now is

$$\mathbf{R} \cdot \bar{\mathbf{c}} = \bar{\mathbf{b}} \qquad (A5)$$

where

$$\mathbf{R} = \mathbf{Q} \begin{pmatrix} \mathbf{T} & \mathbf{T}_H \\ \mathbf{T}_H^T & \mathbf{T}^T \end{pmatrix} \mathbf{Q}^T = \begin{pmatrix} R_o & R_{-1} & R_{-2} & \dots & R_{-M} \\ R_1 & R_o & R_{-1} & \dots & R_{-M+1} \\ \vdots & \vdots & \vdots & & \vdots \\ R_M & R_{M-1} & R_{M-2} & \dots & R_o \end{pmatrix} ;$$

$$R_l = \begin{pmatrix} t(l) & h(M+l) \\ h(M-l) & t(-l) \end{pmatrix} \qquad -M \le l \le M ;$$

$$\bar{\mathbf{c}} = \mathbf{Q} \begin{pmatrix} \mathbf{c}_+ \\ \mathbf{c}_- \end{pmatrix} = \begin{pmatrix} \bar{c}_o \\ \bar{c}_1 \\ \vdots \\ \bar{c}_M \end{pmatrix} ;$$

$$\bar{c}_l = \begin{pmatrix} c_l \\ c_{M-l} \end{pmatrix} \qquad 0 \le l \le M ;$$

$$\bar{\mathbf{b}} = \mathbf{Q} \begin{pmatrix} \mathbf{b}_+ \\ \mathbf{b}_- \end{pmatrix} = \begin{pmatrix} \bar{b}_o \\ \bar{b}_1 \\ \vdots \\ \bar{b}_M \end{pmatrix} ;$$

$$\bar{b}_l = \begin{pmatrix} b_l \\ b_{M-l} \end{pmatrix} \qquad 0 \le l \le M .$$

We have thus converted the Toeplitz plus Hankel system of equations ($A1$) into a block Toeplitz system of equations ($A5$).

The block Levinson algorithm can now be applied. The block Levinson recursions can be summarized as follows:

Step 1 :

$$X_o = Y_o = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} , \ \bar{c}_o = P_1 = R_o^{-1} \bar{b}_o , \ V_x = R_o .$$

Step 2 : For $1 \leq i \leq M$,

$$(a)\ E_x = \sum_{j=0}^{i-1} R_{i-j} X_j$$

$$(b)\ \bar{e}_p = \sum_{j=0}^{i-1} R_{i-j} \bar{c}_j$$

$$(c)\ B_x = (V_x^{T_x})^{-1} E_x$$

$$(d)\ \begin{pmatrix} X_1 \\ X_2 \\ \vdots \\ X_{i-1} \\ X_i \end{pmatrix} \leftarrow \begin{pmatrix} X_1 \\ X_2 \\ \vdots \\ X_{i-1} \\ 0 \end{pmatrix} - \begin{pmatrix} X_{i-1}^{T_x} \\ X_{i-2}^{T_x} \\ \vdots \\ X_1^{T_x} \\ I \end{pmatrix} \cdot B_x$$

$$(e)\ V_x \leftarrow V_x - E_x^{T_x} B_x$$

$$(f)\ \bar{g} = (V_x^{T_x})^{-1} (\bar{b}_i - \bar{e}_p)$$

$$(g)\ P_{i+1} = \begin{pmatrix} \bar{c}_o \\ \bar{c}_1 \\ \vdots \\ \bar{c}_{i-1} \\ \bar{c}_i \end{pmatrix} \leftarrow \begin{pmatrix} \bar{c}_o \\ \bar{c}_1 \\ \vdots \\ \bar{c}_{i-1} \\ \bar{0} \end{pmatrix} + \begin{pmatrix} X_{i-1}^{T_x} \\ X_{i-2}^{T_x} \\ \vdots \\ X_1^{T_x} \\ I \end{pmatrix} \cdot \bar{g}$$

Step 3: The elements of $\mathbf{c}$, i.e, c_o, c_1, ... c_M can be directly read off P_{M+1}.

This completes our brief discussion of the Merchants-Parks algorithm.

APPENDIX B

Evaluation of optimum α in spectral correction algorithm

The error measure $\overline{E}$ that we seek to minimize in the spectral correction algorithm is given by

$$\overline{E} = a^2 + \frac{b^2}{2} \tag{B1}$$

$$\text{where } a = 1 + \cos\alpha_1 \cos\alpha_2 - \frac{1}{4}(r + \frac{1}{r})^2 - \cos^2\alpha \tag{B2}$$

$$b = \cos\alpha_1 + \cos\alpha_2 - (r + \frac{1}{r})\cos\alpha. \tag{B3}$$

In this appendix we will show that there exists a unique optimum α for a given r ($r < 1$), which minimizes $\overline{E}$. We will also obtain a closed form expression for the optimum α.

This appendix is divided into three parts. In the first part, we will show that the optimum α=α^* must satisfy the cubic equation

$$c_1 \cos^3\alpha + b_1 \cos\alpha + a_1 = 0 \tag{B4}$$

where c_1, b_1 and a_1 are all functions of r, α_1 and α_2. In the second part, we will show that equation ($B4$) always has a unique real solution for α^*. In the third part, we will present a closed form expression for this real solution.

Part 1 : Derivation of equation (B4)

For minimum $\overline{E}$, we must have

$$\frac{d\overline{E}}{d\alpha} = 0$$

$$\frac{d^2\overline{E}}{d\alpha^2} > 0.$$

So let us evaluate the first two derivatives of $\overline{E}$ with respect to α.

$$\frac{d\overline{E}}{d\alpha} = b\frac{db}{d\alpha} + 2a\frac{da}{d\alpha}$$

$$= b(r + \frac{1}{r})\sin\alpha + 4a\sin\alpha\cos\alpha$$

Substituting for a and b from equations $(B2)$ and $(B3)$, we get

$$\frac{d\overline{E}}{d\alpha} = \sin\alpha(\, c_1 \cos^3\alpha + b_1 \cos\alpha + a_1\,) \qquad (B5)$$

where

$$c_1 = -4 \qquad (B6)$$

$$b_1 = 4 + 4\cos\alpha_1 \cos\alpha_2 - 2(r + \frac{1}{r})^2 \qquad (B7)$$

$$a_1 = (r + \frac{1}{r})(\cos\alpha_1 + \cos\alpha_2). \qquad (B8)$$

The second derivative of $\overline{E}$ with respect to α can now be expressed as

$$\frac{d^2\overline{E}}{d\alpha^2} = \cos\alpha(\, c_1 \cos^3\alpha + b_1 \cos\alpha + a_1\,) + \sin^2\alpha(\, -3c_1 \cos^2\alpha - b_1\,). \qquad (B9)$$

From $(B5)$ it is clear that the first derivative vanishes at $\alpha = 0$, $\alpha = \pi$ and at those values of $\alpha=\alpha^*$ which satisfy the cubic equation $(B4)$. Let us now examine whether the second derivative becomes positive at these values of α. At $\alpha = 0$,

$$\begin{aligned}
\frac{d^2\overline{E}}{d\alpha^2}|_{\alpha=0} &= c_1 + b_1 + a_1 \\
&= 4\cos\alpha_1 \cos\alpha_2 - 2(r + \frac{1}{r})^2 + (r + \frac{1}{r})(\cos\alpha_1 + \cos\alpha_2) \\
&\leq 4 - 2(r + \frac{1}{r})^2 + 2(r + \frac{1}{r}) \qquad (\text{ since } |\cos\alpha_1| \leq 1,\ |\cos\alpha_2| \leq 1\,) \\
&= 4.5 - 2(r + \frac{1}{r} - 0.5)^2 \\
&< 0 \qquad (\text{ since } r + \frac{1}{r} > 2 \text{ if } r < 1\,).
\end{aligned}$$

Thus we see that $\alpha = 0$ is not a minimum. At $\alpha = \pi$,

$$\begin{aligned}
\frac{d^2\overline{E}}{d\alpha^2}|_{\alpha=\pi} &= c_1 + b_1 - a_1 \\
&= 4\cos\alpha_1 \cos\alpha_2 - 2(r + \frac{1}{r})^2 - (r + \frac{1}{r})(\cos\alpha_1 + \cos\alpha_2) \\
&\leq 4 - 2(r + \frac{1}{r})^2 + 2(r + \frac{1}{r}) \qquad (\text{ since } |\cos\alpha_1| \leq 1,\ |\cos\alpha_2| \leq 1\,) \\
&= 4.5 - 2(r + \frac{1}{r} - 0.5)^2
\end{aligned}$$

which is again less than 0. So $\alpha = \pi$ is not a minimum. Finally, consider an $\alpha = \alpha^*$ which is a solution of the cubic equation $(B4)$:

$$\begin{aligned}
\frac{d^2\overline{E}}{d\alpha^2}|_{\alpha=\alpha^*} &= \sin^2\alpha^*(-3c_1\cos^2\alpha^* - b_1) \\
&= \sin^2\alpha^*\Big(12\cos^2\alpha^* - 4 - 4\cos\alpha_1\cos\alpha_2 + 2(r+\frac{1}{r})^2\Big) \\
&> 12\sin^2\alpha^*\cos^2\alpha^* \quad (\text{ since } |\cos\alpha_1\cos\alpha_2| \le 1,\ (r+\frac{1}{r}) > 2 \text{ if } r < 1\ ) \\
&\ge 0.
\end{aligned}$$

Thus, we have shown that any $\alpha=\alpha^*$ which satisfies equation $(B4)$ minimizes $\overline{E}$.

Part 2 : Uniqueness of α^*

Consider the cubic polynomial $f(x)$ defined by

$$\begin{aligned}
f(x) &= x^3 + \frac{b_1}{c_1}x + \frac{a_1}{c_1} \qquad (B10) \\
&= x^3 + \Big(\frac{1}{2}(r+\frac{1}{r})^2 - 1 - \cos\alpha_1\cos\alpha_2\Big)x - \frac{1}{4}(r+\frac{1}{r})(\cos\alpha_1 + \cos\alpha_2).
\end{aligned}$$

Now

$$\begin{aligned}
f(1) &= \frac{1}{2}(r+\frac{1}{r})^2 - \cos\alpha_1\cos\alpha_2 - \frac{1}{4}(r+\frac{1}{r})(\cos\alpha_1+\cos\alpha_2) \\
&\ge \frac{1}{2}(r+\frac{1}{r})^2 - 1 - \frac{1}{2}(r+\frac{1}{r}) \qquad (\text{ since } |\cos\alpha_1| \le 1,\ |\cos\alpha_2| \le 1\ ) \\
&= \frac{1}{2}(r+\frac{1}{r})(r+\frac{1}{r}-1) - 1 \\
&> \frac{1}{2}\cdot 2\cdot 1 - 1 \\
&= 0
\end{aligned}$$

$$\begin{aligned}
f(-1) &= -\frac{1}{2}(r+\frac{1}{r})^2 + \cos\alpha_1\cos\alpha_2 - \frac{1}{4}(r+\frac{1}{r})(\cos\alpha_1+\cos\alpha_2) \\
&\le -\frac{1}{2}(r+\frac{1}{r})^2 + 1 + \frac{1}{2}(r+\frac{1}{r}) \qquad (\text{ since } |\cos\alpha_1| \le 1,\ |\cos\alpha_2| \le 1\ ) \\
&= -\frac{1}{2}(r+\frac{1}{r})(r+\frac{1}{r}-1) + 1 \\
&< -\frac{1}{2}\cdot 2\cdot 1 + 1 \\
&= 0.
\end{aligned}$$

So, clearly, there exists at least one value $x = x^*$ in the interval [-1, 1] for which $f(x)=0$. We next show that the other two solutions of $f(x)=0$ are complex rather than real:

$$\begin{aligned} f(x) &= x^3 + \Big(\frac{1}{2}(r+\frac{1}{r})^2 - 1 - \cos\alpha_1 \cos\alpha_2\Big)x - \frac{1}{4}(r+\frac{1}{r})(\cos\alpha_1 + \cos\alpha_2) \\ &= (x - x^*)(x^2 + x^*x + w). \end{aligned}$$

By equating coefficients of x we get

$$\begin{aligned} w - x^{*2} &= \frac{1}{2}(r+\frac{1}{r})^2 - 1 - \cos\alpha_1 \cos\alpha_2 \\ &> \frac{1}{2}(r+\frac{1}{r})^2 - 1 - 1 \qquad (\text{ since } |\cos\alpha_1 \cos\alpha_2| \leq 1\ ) \\ &> \frac{1}{2}\cdot 2^2 - 1 - 1 \\ &= 0. \end{aligned}$$

But the discriminant of the quadratic $x^2 + x^*x + w$ is $4(x^{*2} - w)$, which is negative. So the quadratic has no real roots. We conclude that the cubic equation

$$f(x) = x^3 + \frac{b_1}{c_1}x + \frac{a_1}{c_1} = 0$$

has only one real root which lies in [-1, 1]. There is only one real solution $\alpha = \alpha^*$ which satisfies

$$c_1 \cos^3\alpha + b_1 \cos\alpha + a_1 = 0.$$

Part 3 : Closed form solution for α^*

We now proceed to give a closed form expression for x^*, the real root of $f(x) = 0$, which is the same as $\cos\alpha^*$. The closed form expression has been obtained using a standard method for solving cubic equations ([53]):

$$\begin{aligned} \cos\alpha^* &= x^* \\ &= \Big(\frac{-p + \sqrt{p^2 + 4l^3/27}}{2}\Big)^{\frac{1}{3}} + \Big(\frac{-p - \sqrt{p^2 + 4l^3/27}}{2}\Big)^{\frac{1}{3}} \end{aligned}$$

where

$$p = \frac{a_1}{c_1} = -\frac{1}{4}(r + \frac{1}{r})(\cos \alpha_1 + \cos \alpha_2)$$

$$l = \frac{b_1}{c_1} = -\frac{1}{2}\left(2 + 2\cos \alpha_1 \cos \alpha_2 - (r + \frac{1}{r})^2\right).$$

APPENDIX C

Friedlander's spectral factorisation algorithm

We are given a polynomial $C(z)$ where

$$C(z) = c_o + \frac{c_1}{2}(z + z^{-1}) + \frac{c_2}{2}(z^2 + z^{-2}) + \ldots + \frac{c_M}{2}(z^M + z^{-M})$$

$$= \sum_{i=-M}^{M} R_i z^{-i}$$

$$\text{where } R_i = R_{-i} = \frac{c_i}{2} \qquad 1 \leq i \leq M$$

$$\text{and } R_o = c_o.$$

Let us temporarily assume that $C(z)$ is positive definite. Such a positive definite polynomial can always be expressed as

$$C(z) = A(z)A(z^{-1})$$

where $A(z) = \sum_{i=0}^{M} a_i z^{-i}$ is a minimum-phase polynomial; i.e., the roots of $A(z)$ all lie within the unit circle. This kind of factorization is called the *spectral factorization*. The spectral factorization problem is simply this: Given $c_0, c_1, \ldots, c_M$, solve for $a_o, a_1, \ldots, a_M$. The approach that we will describe is due to Friedlander ([50]). In this appendix we will describe only the basic idea behind the algorithm. For further details of implementation the reader is referred to [50].

Any positive definite sequence such as $\{R_{-M}, \ldots, R_M\}$ can be thought of as an autocorrelation sequence of a moving average random process $\{y_t\}$ which is generated according to

$$y_t = \sum_{i=0}^{M} a_i e_{t-i}$$

where $\{e_i\}$ is a sequence of uncorrelated random variables with zero mean and unit variance. One can easily verify that

$$R_i = E[y_t y_{t+i}] = \begin{cases} \sum_{j=0}^{M-|i|} a_j a_{j+|i|} & \text{if } |i| \leq M \\ 0 & \text{otherwise .} \end{cases}$$

We define the covariance matrix $\mathbf{R}_N$ as the $(N+1)\times(N+1)$ matrix whose $(i,\ j)^{th}$ element is R_{j-i} $(0 \le i \le N,\ 0 \le j \le N)$. Thus, for N sufficiently larger than M,

$$\mathbf{R}_N = \begin{pmatrix} R_o & \cdots & R_M & & & & 0 \\ \vdots & \ddots & & \ddots & & & \\ R_{-M} & & & & & & \\ & \ddots & & \ddots & & \ddots & \\ & & & & & & R_M \\ & & & \ddots & & \ddots & \vdots \\ 0 & & & & R_{-M} & \cdots & R_o \end{pmatrix}.$$

Note that the covariance matrix $\mathbf{R}_N$ is positive definite because if $\mathbf{z}$ is any non-zero $(N+1)$ dimensional vector then

$$\begin{aligned} \mathbf{z}^T\mathbf{R}_N\mathbf{z} &= \sum_{i=0}^{N}\sum_{j=0}^{N} z_i z_j E[y_t y_{t+j-i}] \\ &= \sum_{i=0}^{N}\sum_{j=0}^{N} z_i z_j E[y_{t+i} y_{t+j}] \\ &= E\left[\sum_{i=0}^{N} z_i y_{t+i}\right]^2 \\ &> 0. \end{aligned}$$

We also note that $\mathbf{R}_N$ is Toeplitz, symmetric and has a banded structure. The Choleski factorization ([47]) of the matrix $\mathbf{R}_N$ can be expressed as

$$\mathbf{R}_N = \mathbf{R}_N^{\frac{1}{2}}\mathbf{R}_N^{\frac{1}{2}\,T}$$

where

$$\mathbf{R}_N^{\frac{1}{2}} = \begin{pmatrix} \psi_{M,N} & \cdots & \psi_{0,N} & & & & & & 0 \\ & \ddots & \vdots & \ddots & & & & & \\ & & \psi_{M,N-M} & \cdots & \psi_{0,N-M} & & & & \\ & & & \ddots & & \ddots & & & \\ & & & & \ddots & & \ddots & & \\ & & & & & \psi_{M,M} & \cdots & \psi_{0,M} \\ & & & & & & \ddots & \vdots \\ 0 & & & & & & & & \psi_{M,0} \end{pmatrix}.$$

Note that $\mathbf{R}_N^{\frac{1}{2}}$ has all real elements and it is upper triangular and also has a banded structure.

As the size of the matrix N increases ($N \rightarrow \infty$), the top row of $\mathbf{R}_N^{\frac{1}{2}}$ matrix will converge to the coefficients of $A(z)$:

$$\lim_{N\rightarrow\infty} \psi_{M-i,N} = a_i \qquad 0 \leq i \leq M. \tag{C1}$$

Friedlander explains this observation as follows. We can express the R_i's as

$$(R_o \quad R_1 \quad \ldots \quad R_M) = (a_o \quad a_1 \quad \ldots \quad a_M) \begin{pmatrix} a_o & \ldots & 0 \\ \vdots & \ddots & \vdots \\ a_M & \ldots & a_o \end{pmatrix}. \tag{C2}$$

Now the top row of $\mathbf{R}_N$ can also be expressed as

$$(R_o \quad R_1 \quad \ldots \quad R_M) = (\psi_{M,N} \quad \psi_{M-1,N} \quad \ldots \quad \psi_{0,N}) \begin{pmatrix} \psi_{M,N} & \ldots & 0 \\ \vdots & \ddots & \vdots \\ \psi_{0,N} & \ldots & \psi_{M,N-M} \end{pmatrix}.$$

For sufficiently large N,

$$(R_o \quad R_1 \quad \ldots \quad R_M) \approx (\psi_{M,N} \quad \psi_{M-1,N} \quad \ldots \quad \psi_{0,N}) \begin{pmatrix} \psi_{M,N} & \ldots & 0 \\ \vdots & \ddots & \vdots \\ \psi_{0,N} & \ldots & \psi_{M,N} \end{pmatrix}. \tag{C3}$$

Comparison of $(C2)$ and $(C3)$ validates $(C1)$.

The convergence rate of the algorithm depends on the location of the roots of $A(z)$. The closer the roots are to the unit circle, the longer the algorithm will take to converge. This difficulty, however, is inherent to the spectral factorization problem and is not caused by the specific technique that we have used. The convergence can be checked by finding out how much the parameter values $\{\psi_{i,N}\}$, which correspond to the top row of the $\mathbf{R}_N^{\frac{1}{2}}$ matrix, change with each iteration.

However, in our application the estimated $C(\omega)$ is not guaranteed to be positive definite. If the estimated $C(\omega)$ is not positive definite, then the matrix $\mathbf{R}_N$ will not

be positive definite, either. As a consequence, at least some of the $\psi_{M,i}$'s, which are the diagonal elements of the matrix $\mathbf{R}_N^{\frac{1}{2}}$, will become zero or imaginary. Thus we can always detect when $C(\omega)$ is not positive definite, so that we can use the spectral correction routine.

So in practice as we run the algorithm on the estimated $C(\omega)$, any of these three situations can arise:

1. $C(z)$ is positive definite and has roots not too close to the unit circle. In this situation, the elements of $\mathbf{R}_N^{\frac{1}{2}}$ are all real and the algorithm converges within a specified number of iterations N_{max}. (The value of N_{max} in our implementation is 400.)

2. $C(z)$ is positive definite but some of its roots are close to the unit circle. In this situation, the elements of $\mathbf{R}_N^{\frac{1}{2}}$ are still real but the algorithm does not converge in fewer than N_{max} iterations. To speed up the algorithm, we add a very small constant to $C(z)$ and run the algorithm again. The value of this small constant in our implementation is $0.001|c_o|$.

3. $C(z)$ is not positive definite. This situation is detected when some of the diagonal elements of the matrix $\mathbf{R}_N^{\frac{1}{2}}$ become zero or imaginary. So the sequence $C(z)$ is sent to a spectral correction routine (see Section 4.3) to be modified. The algorithm is rerun on the modified $C(z)$, which is guaranteed to be positive definite.

This concludes our discussion of Friedlander's spectral factorization algorithm.

REFERENCES

1. N. S. Jayant, *Waveform Quantization and Coding*, IEEE Press, New York, 1976

2. J. L. Flanagan, *Speech Analysis, Synthesis and Perception*, second edition, Springer-Verlag, New York, 1972

3. L. R. Rabiner and R. W. Schafer, *Digital Processing of Speech Signals*, Englewood Cliffs, NJ: Prentice-Hall, 1978

4. B. Gold and C. M. Rader, "The Channel Vocoder," *IEEE Trans. Audio Electroacoust.*, vol. AU-15, pp. 148-160, 1967

5. B. Gold, P. E. Blankenship and R. J. Mcaulay, "New applications of channel vocoder," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-29, pp. 13-22, 1981

6. M. R. Schroeder, "Vocoders: Analysis and Synthesis of Speech," *Proc. IEEE*, vol. 54, pp. 720-734, 1966

7. B. Gold and C. M. Rader, "Systems for compressing the bandwidth of speech," *IEEE Trans. Audio Electroacoust.*, vol. AU-15, pp. 131-135, 1967

8. A. V. Oppenheim and R. W. Schafer, "Homomorphic Analysis of Speech," *IEEE Trans. Audio Electroacoust.*, vol. AU-16, pp. 221-226, 1968

9. G. E. Kopec, A. V. Oppenheim and J. M. Tribolet, "Speech Analysis by Homomorphic Prediction," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-25, pp. 40-49, 1977

10. C. J. Weinstein and A. V. Oppenheim, "Predictive Coding in a Homomorphic Vocoder," *IEEE Trans. Audio Electroacoust.*, vol. AU-19, pp. 243-248, 1971

11. A. V. Oppenheim, "A speech analysis-synthesis system based on homomorphic

filtering," *J. Acoust. Soc. Amer.*, vol. 45, pp. 458-465, 1969

12. A. V. Oppenheim, R. W. Schafer and T. G. Stockham, "Nonlinear filtering of multiplied and convolved signals," *Proc. IEEE*, vol. 56, pp. 1264-1291, 1968

13. B. S. Atal and S. L. Hanauer, "Speech analysis and synthesis by linear prediction of the speech wave," *J. Acoust. Soc. Amer.*, vol. 50, pp. 637-655, 1973

14. J. Makhoul, "Spectral linear prediction," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-23, pp. 283-296, 1975

15. J. Makhoul, "Linear Prediction," *Proc. IEEE*, vol. 63, pp. 561-580, 1975

16. J. D. Markel and A. H. Gray, Jr., *Linear Prediction of Speech*, Springer-Verlag, New York, 1976

17. J. Makhoul, "Spectral analysis of speech by linear prediction," *IEEE Trans. Audio Electroacoust.*, vol. AU-21, pp. 140-148, 1973

18. J. D. Markel and A. H. Gray, Jr., "A linear prediction vocoder simulation based upon the autocorrelation method," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-22, pp. 124-134, 1974

19. M. R. Sambur, "An Efficient Linear Prediction Vocoder," *Bell. Syst. Tech. J.*, vol. 54, pp. 1693-1723, 1975

20. R. E. Crochiere, S. A. Webber and J. L. Flanagan, "Digital Coding of Speech in Sub-bands," *Bell Syst. Tech. J.*, vol. 55, pp. 1069-1085, 1976

21. R. E. Crochiere, "On the design of sub-band coders for low bit rate speech communication," *Bell. Syst. Tech. J.*, vol. 56, pp. 747-770, 1977

22. C. Grauel, "Sub-band coding with adaptive bit allocation," *Signal Processing*, vol. 2, pp. 23-30, 1980

23. J. M. Tribolet and R. E. Crochiere, "Frequency domain coding of speech," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-27, pp. 512-530, 1979

24. R. Zelinski and P. Noll, "Adaptive Transform Coding of Speech Signals," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-25, pp. 299-309, 1977

25. R. V. Cox and R. E. Crochiere, "Real-time simulation of adaptive transform coding," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-29, pp. 147-154, 1981

26. B. S. Atal and J. R. Remde, "A new model of LPC excitation for producing natural-sounding speech at low bit rates," *Proc. Int. Conf. Acoust., Speech, Signal Processing*, Paris, France, pp. 614-617, 1982

27. M. R. Schroeder, "Recent progress in speech coding at Bell Laboratories," *Proc. III Int. Congress on Acoustics*, pp. 201-210, Elsevier Publishing Co., Amsterdam, 1961

28. V. R. Viswanathan, A. Higgins and W. Rusell, "Design of a robust baseband LPC coder for speech transmission over 9.6 kbits/s noisy channels," *IEEE Trans. Comm.*, vol. 30, pp. 663-673, 1982

29. M. R. Schroeder, "Correlation techniques for speech bandwidth compression," *J. Audio Eng. Soc.*, no. 10, pp. 163-166, 1962

30. D. B. Paul, "Spectral envelope estimator vocoder," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-29, pp. 786-794, 1981

31. J. L. Flanagan, M. R. Schroeder, B. S. Atal, R. E. Crochiere, N. S. Jayant and J. M. Tribolet, "Speech coding," *IEEE Trans. Comm.*, vol. COM-27, pp. 710-737, 1979

32. I. Eisenberger and E. C. Posner, "Systematic statistics used for data compression in space telemetry," *J. Amer. Statistical Assn.* , vol. 60, pp. 97-133, 1967

33. E. C. Posner, "The use of quantiles for space telemetry data compression," *Proc. of the 1964 National Telemetering Conf.*, International Foundation for Telemetering, Los Angeles, California, pp. 1.3-1.6, 1964

34. G. A. Merchant and T. W. Parks, "Efficient solution of a Toeplitz plus Hankel coefficient matrix system of equations," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-30, pp. 40-44, 1982

35. A. V. Oppenheim and R. W. Schafer, *Digital Signal Processing*, Englewood Cliffs, NJ: Prentice-Hall, 1975

36. J. R. Pierce and E. C. Posner, *Introduction to Communication Science and Systems*, Plenum Press, New York, 1980

37. J. B. Allen, "Short-term spectral analysis and synthesis and modification by discrete fourier transform," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-25, pp. 235-238, 1977

38. J. B. Allen and L. R. Rabiner, "A unified theory of short-time spectrum analysis and synthesis," *Proc. IEEE*, vol. 65, pp. 1558-1564, 1977

39. R. W. Schafer and L. R. Rabiner, "Design and simulation of a speech analysis-synthesis system based on short-time Fourier analysis," *IEEE Trans. Audio Electroacoust.*, vol. AU-21, pp. 165-174, 1973

40. M. R. Portnoff, "Time-frequency representation of digital signals and systems based on short-time Fourier analysis," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-28, pp. 55-69, 1980

41. M. R. Portnoff, "Short-time Fourier analysis of sampled speech," *IEEE Trans., Acoust., Speech, Signal Processing*, vol. ASSP-29, pp. 364-373, 1981

42. R. W. Schafer and L. R. Rabiner, "Digital representations of speech signals," *Proc IEEE*, vol. 63, pp. 662-677, 1978

43. D. Y. Wong, C. C. Hsiao and J. D. Markel, "Spectral Mismatch Due to Preemphasis in LPC Analysis/Synthesis, " *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-28, pp. 263-264, 1980

44. A. Papoulis, "Maximum Entropy and Spectral Estimation: A review," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-29, pp. 1176-1186, 1981

45. C. L. Lawson and R. J. Hanson, *Solving Least Squares Problems*, Englewood Cliffs, NJ: Prentice-Hall, 1974

46. J. Franklin, *Methods of Mathematical Economics*, New York: Springer-Verlag, 1980

47. J. Franklin, *Matrix theory*, Englewood Cliffs, NJ: Prentice-Hall, 1968

48. G. T. Wilson, "Factorization of the covariance function of a pure moving average process," *SIAM J. Numer. Anal.*, vol. 6, pp. 1-7, 1969

49. G. T. Wilson, "The factorization of matricial spectral densities," *SIAM J. App. Math*, vol. 23, pp. 420-426, 1972

50. B. Friedlander, "A lattice algorithm for factoring the spectrum of a moving average process," *IEEE Trans. Automat. Contr.*, vol. AC-28, pp. 1051-1055, 1983

51. H. R. Schwarz, *Numerical Analysis of Symmetric Matrices*, Englewood Cliffs, NJ: Prentice-Hall, 1973

52. H. Akaike, "Block Toeplitz matrix inversion," *SIAM J. Appl. Math*, vol. 24,

pp. 234-241, 1973

53. L. W. Griffiths, *Introduction to the Theory of Equations*, second edition, New York: John Wiley & Sons, Inc. , 1946

54. T. Kailath, *Linear Systems*, Englewood Cliffs, NJ: Prentice-Hall, 1980

55. S. Singhal and B. S. Atal, "Improving performance of multi-pulse LPC coders at low bit rates, " *Proc. Int. Conf. Acoust., Speech, Signal Processing*, pp. 1.3.1-1.3.4, 1984

56. P. Kroon and E. F. Deprettere, "Experimental evaluation of different approaches to the multi-pulse coder, " *Proc. Int. Conf. Acoust., Speech, Signal Processing*, pp. 10.4.1-10.4.4, 1984

57. M. Berouti, H. Garten, P. Kabal and P. Mermelstein, "Efficient computation and encoding of the multi-pulse excitation for LPC, " *Proc. Int. Conf. Acoust., Speech, Signal Processing*, pp. 10.1.1-10.1.4, 1984

58. V. K. Jain and R. Hangartner, "Efficient algorithm for multi-pulse LPC analysis of speech, " *Proc. Int. Conf. Acoust., Speech, Signal Processing*, pp. 1.4.1-1.4.4, 1984

59. A. Parker, S. T. Alexander and H. J. Trucell, "Low bit rate speech enhancement using a new method of multiple impulse excitation, " *Proc. Int. Conf. Acoust., Speech, Signal Processing*, pp. 1.5.1-1.5.4, 1984

60. J. N. Holmes, "Formant excitation before and after glottal closure, " *Conf. Rec. 1976 IEEE Int. Conf. Acoust., Speech, Signal Processing*, pp. 39-42, Apr. 1976

61. B. S. Atal, "Predictive coding of speech at low bit rates, " *IEEE Trans. Comm.*, vol. COM-30, pp. 600-614, 1982

62. M. R. Schroeder, B. S. Atal and J. L. Hall, "Optimizing digital speech coders by exploiting masking properties of the human ear, " *J. Acoust. Soc. Amer.*, vol. 66, pp. 1647- 1652, 1979

63. V. R. Viswanathan and J. Makhoul, "Quantization properties of transmission parameters in linear predictive systems, " *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-23, pp. 309-321, 1975

64. A. H. Gray, Jr. and J. D. Markel, "Quantization and bit allocation in speech processing, " *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-24, pp. 459-473, 1976

65. N. S. Jayant and P. Noll, *Digital Coding of Waveforms*, Englewood Cliffs, NJ: Prentice-Hall, 1984

66. G. A. Miller and S. Isard, "Some perceptual consequences of linguistic rules, " *J. Verb. Learn. Verb. Behav.*, vol. 2, pp. 217-228, 1963

67. R. E. Crochiere and L. R. Rabiner, *Multirate Digital Signal Processing*, Englewood Cliffs, NJ: Prentice-Hall, 1983

68. J. H. McClellan, T. W. Parks and L. R. Rabiner, "A computer program for designing optimum linear phase digital filters, " *IEEE Trans. Audio Electroacoust.*, vol. AU-21, pp. 506-526, 1973

69. P. Noll, "Adaptive quantization in speech coding systems, " *Proc. Int. Zurich Seminar on Digital Communications*, pp. B3.1-B3.6, 1974

70. B. J. McDermott, C. Scagliola and D. J. Goodman, "Perceptual and objective evaluation of speech processed by adaptive differential PCM, " *Bell Syst. Tech. J.*, vol. 57, pp. 1597-1618, 1978

71. M. Nakatsui and P. Mermelstein, "Subjective speech-to-noise ratio as a measure of speech quality for digital waveform coders, " *J. Acoust. Soc. Amer.*, vol. 72, pp. 1136-1144, 1982

72. W. R. Daumer, "Subjective evaluation of several efficient speech coders, " *IEEE Trans. Comm.*, vol. COM-30, pp. 655-673, 1982