ANALYTIC SIGNAL PROCESSING

BY

JOSEPH PICONE

Submitted in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy in Electrical Engineering
in the Graduate School of
Illinois Institute of Technology

Approved Joseph L. LoCicero
Adviser

Chicago, Illinois
December, 1983

ACKNOWLEDGEMENT

This thesis is a result of a joint research venture between Illinois Institute of Technology and Bell Laboratories in Naperville, Illinois. It would not have been possible without the generous financial support and the use of the resources of the Advanced Switching Services Laboratory. In particular, most of this work is a direct result of many stimulating conversations with Walt Hartwell concerning analytic signals. His many insights into analytic signals has given most of the shape to this work. I am personally indebted to Dimitrios P. Prezas, a friend and colleague, for his continuous involvement in this work. The many hours he has devoted to this project has made the past two years an invaluable experience.

To my advisor, Joseph L. LoCicero, goes a special thanks for transforming an idea about a joint research venture into a reality. His guidance and assistance is deeply appreciated. Also, I wish to thank the Department of Electrical Engineering for its financial support through my graduate studies, and in particular, Kenneth W. Haag for his personal assistance in these matters.

Finally, I wish to thank my wife for her patience, tolerance, and understanding.

J. P.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABSTRACT

High resolution spectral estimation algorithms traditionally are constrained to process a finite segment of data, assuming the data to be zero outside the analysis interval. These assumptions ultimately limit the resolution which can be achieved by these estimators. Analytic signals provide an alternate signal representation whereby long-term phase information can be incorporated into the analysis data. Analytic signal-based spectral estimators are shown to achieve higher resolution than their real signal counterparts, due to the phase-invariance property of an analytic signal. This is demonstrated in both the analog and discrete domains. The linear predictive estimates of stationary signals in additive Gaussian noise, using a complex linear predictor, are shown to be more consistent than those obtained using a comparable real linear predictor. Experimental results are included for two analysis algoithms: the Burg algorithm and the autocorrelation method.

The analytic signal representation has been applied to data compression techniques for speech encoding. Multi-pulse linear predictive coding (MPLPC) is developed as a sub-optimal two-stage system identification algorithm. Several complex parameter coding algorithms are presented as sub-optimal solutions to a two-dimensional quantization problem. Two techniques to transform complex predictor coefficients to equivalent real predictor coefficients are

included as alternate coding algorithms. An analytic multi-pulse linear predictive coding system (AMPLPC) has been designed which provides improved speech quality over its real signal counterpart at rates ranging from 9.6 kbits/s to 14.4 kbits/s.

CHAPTER I

INTRODUCTION

Since the advent of digital signal processing in the late 1950s, the goal of producing the digital equivalent of the analog spectrum analyzer has spawned literally hundreds of digital spectral estimation techniques. Some of these fall in the category of "high resolution spectral estimators" while others are simply derivatives of the analog Fourier transform. In any event, the relative shortcomings of virtually all of these approaches has been well documented in the literature [1-10]*. The common thread among all these digital techniques is that they involve the estimation of the spectrum of a sampled, bandlimited, stationary time series from a finite number of samples of that process.

In this dissertation, the conventional approach of processing a single set of samples of a one-dimensional signal is discarded in favor of operating on an analytic, or "single-sideband," representation of this signal. Throughout this work, it is assumed that a continuous stream of data is available to form the analytic signal, as is typically the case in such applications as speech encoding. This analytic signal is shown to incorporate the long-term phase information of the signal into the analysis interval. This representation of a signal is shown to be a natural

---

*Numbers in brackets refer to numbered references.

choice when dealing with finite sample sets. Surprisingly, the computational burden required by the complex, or two-dimensional, sequence is comparable to its real, or one-dimensional, equivalent.

While the analytic signal representation as applied to purely one-dimensional signals is not a traditional approach in time series analysis, the notion of an analytic signal has found its way into many applications. Single-sideband signals [11-13] utilize an analytic signal representation to minimize the bandwidth required to transmit a signal. In sonar and radar signal detection applications, spatially diverse transducers are deployed to sample a multidimensional wave front. These sensors are deployed in such a fashion that the signals generated by these transducers are normally phase delayed versions of the same signal. In a very simple case, where only a signal and its quadrature component are used, the composite signal is essentially an analytic signal. In additive noise, these sensor arrays provide improved signal detection capabilities, as the array can be focused in the direction of the strongest component of the signal. This is known as beam forming [14]. In fact, in most geophysical applications, multi-channel signal processing is quite common [1,2].

Analytic signal processing of one-dimensional signals such as speech is not an entirely new idea. Hartwell [3] proposed the analytic signal as a basis for conventional

fast Fourier transform (FFT) analysis. The work of Jackson [4], Soong [15], and Kay [16] explore some of the basic issues of frequency estimation and linear prediction, mostly from an experimental standpoint. The notions of the Hilbert transform envelope have been used in pitch detection applications by Ananthapadmanabha and Yegnanarayana [17], among others. However, the bulk of this dissertation presents a unified view of conventional time domain window theory and its relation to analytic signal processing. Also, a speech compression system which is based upon these notions of an analytic signal has been designed to provide very natural sounding speech at rates in the range of 9.6 kbits/s.

Conceptually, this dissertation is divided into three parts. First, the fundamentals of analytic signal processing are presented in Chapter II. The basic definitions of both an analog and discrete analytic signal are given, as well as the motivations behind these particular conventions. The generation of an analytic signal from a real signal is developed in terms of a Hilbert transform operation. Also, the reconstruction of a real signal from its associated analytic signal is derived. This reconstruction capability is essential to such applications as speech encoding. The technique presented here is attributed to Hartwell [3]. While it is not the only method of recovering a real signal, it is particularly efficient for most speech encoding applications, and is used

extensively in Chapter VII.

The second portion of this dissertation deals with the subject of spectral estimation and spectral resolution. This topic is dealt with in three separate chapters. Chapter III is most fundamentally important chapter to this work, as the basic characteristics of windowed signals are presented from an analog viewpoint. It is in this chapter that the phase invariance property of an analytic signal is first introduced. This property is the basic asset of the analytic representation, and its many ramifications must be fully appreciated to completely grasp the advantages of the analytic signal. Spectral resolution is discussed in terms of the capability of the Fourier transform to accurately estimate the long-term spectrum of a deterministic signal from a time limited version of that signal.

In Chapter IV, the basic results of Chapter III are extended to the discrete domain. The only major difference results from the nature of the signal representation. Discrete spectral representations normally make assumptions about the signal outside the analysis interval. Typically, the signal is assumed to be zero or periodic outside the interval of interest. Naturally, the extent to which these assumptions are valid influence the accuracy of the corresponding spectrum. The phase information embedded in the analytic signal alleviates to some degree the effects of any inaccuracies in these assumptions. Alternate solutions have traditionally involved the use of time domain window

functions, an *ad hoc* solution which doesn't really address the root of the problem. Thus, also in this chapter, discussions of the nature of the discrete spectrum and the merits of time domain windows are included.

The discussion of spectral estimation concludes with Chapter V, in which the analytic signal is shown to provide increased resolution over the real signal for linear predictive spectral estimation. As an introduction, the complex linear predictor is presented from a least squares system identification standpoint. The phase invariance property of the analytic signal is then extended to general least squares spectral estimation. The performance of the analytic signal-based estimator is then compared both theoretically and experimentally to a comparable real signal estimator by considering two particular estimation algorithms: the Burg algorithm and the autocorrelation method. These two estimators are chosen primarily because of their relevance to the speech coding applications presented in Chapter VII. The analytic signal-based estimator is shown to provide more accurate estimates of the frequencies of sinewaves in additive white Gaussian noise, thus attaining higher resolution in the frequency domain than its real signal counterpart.

The third major portion of this dissertation deals with the application of analytic signals to speech coding. Analytic signal-based linear predictive coding of speech is compared to the more common real signal coding. The key

issue examined here is whether the analytic signal can provide improved speech quality at comparable or lower bit rates than an equivalent real signal system. In order to make this comparison effectively, a speech coding algorithm which produces high quality, very intelligible synthetic speech at medium to low bit rates is required. Multi-pulse linear predictive coding (MPLPC) [18] was chosen as a suitable algorithm, primarily because of its capability to produce high quality speech at a variety of bit rates.

In Chapter VI, this algorithm is developed as a sub-optimal technique for the generation of a pulsed excitation signal for an LPC encoder, using a minimum mean-square error criterion. Chapter VII contains the details of a complete analytic multi-pulse linear predictive coding (AMPLPC) system. Several complex parameter coding techniques are are evaluated on a subjective listening basis for their quality versus bit rate and bit rate versus complexity tradeoffs.

The majority of this dissertation is intended to be a self-contained work. In particular, the material in Chapter II thru Chapter V requires very little background other than a basic signals and systems theory background. While most of the material in Chapter VII on speech coding is experimental in nature, a large part of the material dealing with algorithms and coding techniques parallels that found in the expansive collection of literature which exists on speech coding. For those unfamiliar with speech coding, a

brief survey can be obtained in [19-23].

Though most of the notation presented is fairly standard, one comment is in order. Traditionally, the notation in most discrete systems textbooks follows a convention of suppressing the sampling rate in any discrete expressions (or equivalently, assuming a sampling rate of 1 Hz). On the other hand, most analog system textbooks include the sampling rate in all expressions. In order to clearly distinguish between analog and discrete operations, the sampling rate is explicitly defined as 1/T throughout the first five chapters. In doing this, any variation of an expression with the sampling rate is clearly identified. Further, in order to accurately differentiate analog spectra from discrete spectra, analog spectra are expressed as functions of w, i.e., F(w), while discrete spectra are expressed as functions of e^{jw}, i.e., $F(e^{jw})$. In the remaining chapters, it is convenient to suppress this sampling rate, since only discrete sequences are dealt with, and the sampling rate dependence is obvious.

CHAPTER II

ANALYTIC SIGNALS

Introduction

An analytic signal, being nothing more than a signal plus its quadrature, embeds long-term phase information about a signal into the complex signal representation. In this chapter, the basic notion of an analytic signal, in both the analog and discrete domains, is presented. Further, the process of converting a discrete analytic signal to its corresponding discrete real signal is derived, a process fundamental to such applications as speech coding.

Analog Analytic Signals

The motivation underlying an analytic signal representation is best explained by examining the information content in the spectrum of a real signal. For any real-valued signal, f(t), it is easily shown that the Fourier transform of the signal, which is defined as [13]

$$F(w) = \int_{-\infty}^{\infty} f(t)e^{-jwt}\,dt\ , \tag{1}$$

has a magnitude which is an even function [13] of frequency:

$$|F(w)| = |F(-w)|\ . \tag{2}$$

This suggests that the spectral representation of this signal contains redundant information in the sense that the

signal can be reconstructed from the information in either the positive or the negative frequency domain. The frequency bandwidth occupied by this signal can be minimized by removing this redundancy, and creating what can be thought of as a "single-sideband signal." This was perhaps the first major application of an analytic signal.

The complex-valued signal, $s_a(t)$, called an analytic signal [12], can be formed as the sum of a real component, $s(t)$, and an imaginary component, $\hat{s}(t)$, such that

$$s_a(t) = s(t) + j\hat{s}(t) \ . \tag{3}$$

The components of the analytic signal need not be defined as being orthogonal in the complex plane, but are only defined that way for convenience. It is useful in certain signal processing applications to maintain a simple relationship between $s_a(t)$ and $s(t)$, and thus, $s(t)$ is constrained to be equal to the real part of $s_a(t)$. The term analytic is drawn from complex variable theory [24], in that a function is said to be analytic at a point if its derivative exists at that point and in the neighborhood of that point. In this sense, a discrete analytic signal cannot exist, since in a discrete domain, derivates do not exist. This connection with analytic function theory will be clarified presently.

An analytic signal is frequently thought of as having a one-sided spectral density [12]. By requiring that the Fourier transform of $s_a(t)$, $S_a(jw)$, be zero for negative frequencies, the relationship between $S(jw)$ and the Fourier

transform of $\hat{s}(t)$, $\hat{S}(jw)$, must be

$$\begin{aligned} \hat{S}(jw) &= jS(jw) , \qquad w < 0 , \\ &= -jS(jw) , \qquad w > 0 . \end{aligned} \tag{4}$$

The spectrum of $s_a(t)$ is now one-sided:

$$\begin{aligned} S_a(jw) &= 2S(jw) , \quad w > 0 , \\ &= 0 , \qquad w < 0 . \end{aligned} \tag{5}$$

The transformation required to produce $\hat{s}(t)$ from $s(t)$ is known as a Hilbert transform [12],

$$\begin{aligned} \hat{s}(t) &= (1/\pi) \int_{-\infty}^{\infty} f(x)/(t-x)\, dx , \\ &= f(t) * (1/\pi t) , \end{aligned} \tag{6}$$

where "*" denotes time convolution. The multiplication by j in the frequency domain in Equation 4 is equivalent to a phase shift of 90 degrees in the time domain, making $\hat{s}(t)$ in phase quadrature with $s(t)$. Thus, the Hilbert transformer is also known as a 90-degree phase shifter, and an analytic signal is really just a signal plus j times its quadrature.

The term analytic is somewhat of a misnomer. Given any real signal, the signal formed from Equation 3 is not guaranteed to be an analytic function. A simple case would be a discontinuous real signal such as a step function or an impulse function. However, given a real bandlimited signal, the complex signal formed from Equation 3 will be analytic. (Recall a bandlimited function can be represented as a sum

of sin(x)/(x) functions, each of which is analytic over its entire domain.) In communication theory and signal processing, most signals encountered are usually assumed to be bandlimited. Thus, provided s(t) is bandlimited, the signal of Equation 3 can be called an analytic signal, since it is analytic over the entire complex plane.

Discrete Analytic Signals

A bandlimited continuous signal can be uniquely represented by a set of samples taken at a sampling rate known as the Nyquist rate [25]. Even though analyticity is meaningless in a discrete domain, there exists a discrete sequence [5], $s_a(nT)$, corresponding to $s_a(t)$ above, such that

$$s_a(nT) = s_a(t)\Big|_{t=nT} , \tag{7}$$

where $T = 1/f_s$ and f_s is the Nyquist sampling frequency. The discrete signal, or sequence, $s_a(nT)$ is defined as a discrete analytic signal, since the relationship between the spectrum of $s_a(nT)$ and the spectrum of s(nT) is similar to that of Equation 5, that is,

$$\begin{aligned} S_a(e^{jw}) &= 2S(e^{jw}) , \quad 0 \leq w < \pi , \\ &= 0 , \qquad -\pi \leq w < 0 . \end{aligned} \tag{8}$$

The complex signal, $s_a(nT)$, is given as

$$s_a(nT) = s(nT) + j\hat{s}(nT) \ , \tag{9}$$

where $\hat{s}(nT)$ is formed from s(nT) via a discrete convolution process [5],

$$s(nT) = \sum_{m=-\infty}^{\infty} s((n-m)T)\ h(m) \ , \tag{10}$$

and,

$$\begin{aligned} h(m) &= [\sin^2(\pi m/2)]/(\pi m/2) \ , \quad m \neq 0 \ , \\ &= 0 \ , \qquad\qquad\qquad\quad m = 0 \ . \end{aligned} \tag{11}$$

The discrete filter, h(m), is known as a discrete Hilbert transformer. Once again, the real sequence s(nT) is equal to the real part of the complex sequence $s_a(nT)$.

Since the complex sequence $s_a(nT)$ occupies half the bandwidth of the real sequence s(nT), $s_a(nT)$ can be sampled at half the sampling rate of s(nT). To recover s(nT) from the downsampled version of $s_a(nT)$, the downsampled version of $s_a(nT)$ must first be upsampled. In discrete real signal analysis, upsampling involves interpolating the real signal, for example, using a sin(x)/x type interpolation function [25]. This process of interpolation can be viewed as a filtering operation. In analytic signal analysis, the same is true [3], but the interpolation function is somewhat different.

Let the analytic signal $s_a(t)$ be bandlimited from $0 < w < B$, such that it can be sampled at its Nyquist rate, $f_s = (B/2\pi)$. Since $s_a(t)$ is analytic, from

Equation 5, $S_a(w) = 0$ for $w < 0$. The spectrum, $S_a(w)$, can be expanded in a Fourier series [25], giving

$$S_a(w) = \sum_{n=-\infty}^{\infty} q_n e^{jn(2\pi/B)w}, \tag{12}$$

where

$$q_n = (1/B) \int_0^B S_a(w) e^{-jn(2\pi/B)w} dw, \tag{13}$$

or,

$$q_n = (2\pi/B)\, s_a(-2\pi n/B) \tag{14}$$

The Fourier coefficients, q_n, are just samples of the inverse transform of $S_a(w)$ evaluated at $t = -2\pi n/B$, so that by substituting Equation 14 into Equation 11,

$$S_a(w) = (2\pi/B) \sum_{n=-\infty}^{\infty} s_a(-2\pi n/B)\, e^{jn(2\pi/B)w}. \tag{15}$$

Taking the inverse Fourier transform of $S_a(w)$ and utilizing the fact that $S_a(w)$ is zero for negative frequencies,

$$s_a(t) = F^{-1}\{S_a(w)\}$$

$$= (1/B) \int_0^B \sum_{n=-\infty}^{\infty} s_a(-2\pi n/B) e^{j2\pi nw/B} e^{jwt} dw$$

$$= (1/B) \sum_{n=-\infty}^{\infty} s_a(-2\pi n/B) \int_0^B e^{jw[t+(2\pi n/B)]} dw$$

$$= (1/B) \sum_{n=-\infty}^{\infty} s_a(-2\pi n/B) \frac{e^{jB[t+(2\pi n/B)]} - 1}{j[t+(2\pi n/B)]}$$

$$= \sum_{n=-\infty}^{\infty} s_a(-2\pi n/B)\, e^{j[(Bt/2) + \pi n]} \times$$

$$\frac{e^{j[(Bt/2) + \pi n]} - e^{-j[(Bt/2) + \pi n]}}{2j[(Bt/2) + \pi n]}$$

$$= \sum_{n=-\infty}^{\infty} s_a(-2\pi n/B)\, e^{j[(Bt/2) + \pi n]} \times$$

$$[\sin((Bt/2) + \pi n)]/((Bt/2) + \pi n)$$

$$= \sum_{n=-\infty}^{\infty} s_a(2\pi n/B)\, e^{j[(Bt/2) - \pi n)} \times$$

$$[\sin((Bt/2) - \pi n)]/((Bt/2) - \pi n) \; . \quad (16)$$

Thus, time interpolation in the analytic domain corresponds to a convolution process between the signal and a complex sin(x)/x function. This interpolation function, while rather complicated in general, has a relatively simple form when applied to the problem of recovering a real signal sampled at its Nyquist rate from its corresponding analytic signal, also sampled at its Nyquist rate.

In order to upsample the discrete analytic sequence $s_a(nT)$, sampled at its Nyquist rate of $T = 2\pi/B$, Equation 16 can be evaluated at $t = \pi k/B$. This gives an upsampled analytic signal whose real part is the desired real signal. In this case, the interpolation function simplifies to

$$s_a(\pi k/2B) = \sum_{n=-\infty}^{\infty} s_a(2\pi n/B) e^{j\pi((k/2)-n)} \frac{\sin(\pi((k/2)-n))}{\pi((k/2)-n)} \; . \quad (17)$$

The exponential term in Equation 17 can be expanded to

$$e^{j\pi((k/2)-n)} = \cos(\pi((k/2)-n)) + j\sin(\pi((k/2)-n)) .$$

When k is even, the only non-zero term in Equation 17 contributing to the real part of $s_a(\pi k/2B)$ occurs when $k = 2n$. When k is odd, the only contributions come from the product of the imaginary part of $s_a(\pi k/2B)$ and the imaginary part of the exponential. Thus, the interpolation process gives

$$\begin{aligned} \mathrm{Re}\{s_a(\pi k/2B)\} &= \mathrm{Re}\{s_a(\pi k/2B)\} , \quad k = \text{even integers}, \\ &= \sum_{n=-\infty}^{\infty} \mathrm{Im}\{s_a(2\pi n/B)\} \times \\ &\quad \frac{\sin^2(\pi((k/2)-n))}{(\pi((k/2)-n))} , \quad k = \text{odd integers}. \end{aligned} \tag{18}$$

The latter part of Equation 18 is recognized as a convolution of the imaginary part of the analytic signal with a discrete Hilbert transformer.

The recovery process consists of using the real part of $s_a(nT)$, and placing in between these samples a sample which is computed by convolving the imaginary part of $s_a(nT)$ with a Hilbert transformer, as shown in Figure 1. The real part of the analytic signal enters the upper delay line, used to compensate for the delay in the Hilbert transformer, while the imaginary part also enters the delay line of the

discrete Hilbert transformer. Each of these sequences is "stuffed" with a zero between each sample value, and further, the imaginary part is delayed in time by one sample. The switch alternately selects the real part of $s_a(nT)$, and then the output of the Hilbert transformer. The delay lines are shifted at the upsampled rate, (2/T).

This recovery process is intuitively satisfying, since it is almost the inverse of the process used to generate the analytic signal. Further, implementation is reduced to a mere digital filtering operation. Other techniques to implement Equation 17 exist, such as a frequency domain approach using an FFT, but these are generally more computationally expensive, and certainly not as elegant. Later, it will be seen that this particular implementation is well suited to speech coding, while any implementation generally will suffice for spectrum analysis applications.

CHAPTER III

ANALYSIS OF WINDOWED ANALOG SIGNALS

Introduction

The inherent advantages in using an analytic signal representation result from the constraint of a finite length analysis interval. In signal detection and estimation problems, for practical reasons, a finite amount of data is collected, and some decision about the signal is made based on this data. Thus, only a window, or finite portion, of the actual signal is analyzed. The assumptions about the relationship of this windowed signal to the actual signal can introduce certain errors in the estimation process. In this chapter, the effects of a rectangular window upon an analog signal consisting of multiple sinewaves of arbitrary frequencies and phases are considered.

The Spectrum of a Real Sinewave

The spectrum of a windowed sinewave can be computed from Equation 1. Suppose the signal of interest is a single cosinewave of unknown frequency and phase,

$$s(t) = \cos(w_1 t + \emptyset) \ . \tag{19}$$

The finite length version of this signal can be written as the product of the signal with a rectangular window

$$s_w(t) = s(t)W(t) \ , \tag{20}$$

where W(t), a rectangular window of length T seconds, is given by

$$W(t) = 1 , \quad -T/2 \leq t \leq T/2 ,$$
$$= 0 , \quad \text{elsewhere} . \tag{21}$$

This notation will be convenient from the standpoint that both the structure of the window and the position of the window can be easily changed. The phase shift, Ø, actually represents the position of the window in time, since by varying Ø, any portion of the signal can be made to appear in the window.

The computation of the spectrum of this windowed signal is most instructively performed in the frequency domain. We note that multiplication in the time domain is equivalent to convolution in the frequency domain of the spectrum of s(t) with the spectrum of W(t), where

$$S(w) = F\{s(t)\}$$
$$= \pi[e^{j\emptyset} d(w-w_1) + e^{-j\emptyset} d(w+w_1)] ,$$

and

$$W(w) = F\{W(t)\}$$
$$= T\sin(wT/2)/(wT/2) ; \tag{22}$$

here d(w) denotes the unit impulse function. Performing the convolution in the frequency domain, we obtain

$$S_w(w) = \pi T[e^{j\emptyset} \mathrm{Sinc}(x - x_1) + e^{-j\emptyset} \mathrm{Sinc}(x + x_1)] , \tag{23}$$

where

$$x = wT/2, \quad x_1 = w_1T/2, \quad \text{and} \quad \text{Sinc}(x) = \sin(x)/x \ .$$

The $\text{Sinc}(x - x_1)$ term is the contribution from the positive frequency component, while the $\text{Sinc}(x + x_1)$ term corresponds to the negative frequency component.

To accurately estimate the frequency of this sinusoid in additive noise, given no a priori information about this frequency, the location of the peak in the magnitude spectrum of this signal can be used as a criterion. From Equation 23, it is seen that the peak in the magnitude spectrum is a function of both the phase of the signal, Ø, and the length of the window, T. The dependence on these two parameters can also be interpreted in the frequency domain as an "aliasing," or leakage, between negative and positive frequency components of the signal s(t), when rectangularly windowed by W(t). These components arise from the convolution of the Fourier transform of the signal with the Fourier transform of the rectangular window, as demonstrated above. They occur because the effective frequency response of the window is wider than the distance between these spectral components, thereby creating an overlap in the frequency domain.

In Figure 2, this aliasing is depicted by plotting the magnitude of the spectrum for both the positive and negative frequency components separately. In Figure 3, the composite magnitude spectrum from Equation 23 is plotted. In this case, a frequency of 1000 Hz, a phase of 30 degrees, and a

window length of 1.3 ms are used. The regions of overlap indicate that aliasing occurs among positive and negative components, similar to the aliasing which occurs when a signal is undersampled.

In Figure 4, the frequency estimate, as determined by the peak in the magnitude spectrum, is plotted versus the window length for Ø = 0, 30, 60, and 90 degrees. As expected, the estimate of the frequency improves as the window length increases. This is due to the fact that as the window length, T, increases, the sin(x)/x nature of the frequency response of the rectangular window approaches an ideal impulse, thereby reducing the amount of aliasing between positive and negative frequency components. The error is large when the window length is less than one-half the period of the signal, since it is difficult to determine the frequency of the signal when there is only one zero-crossing within the window. For a fixed window length, the error is frequency dependent, as the error for low frequency signals is larger than that for higher frequency signals. Stated another way, this implies the estimate is improved as more cycles of the signal are included in the window.

Figure 5 contains the frequency estimate plotted versus the phase, Ø, for window lengths of 0.75, 1.0, 2.0, and 4.0 ms. Generally, the error is sinusoidally distributed when the window is reasonably long, oscillating about the correct frequency. This variation with phase arises because

the contributions from positive and negative frequency components add vectorially in a manner determined by the phase of the signal, producing a maximum in the spectrum which is a function of phase. The random nature of the error implies that such an error cannot be artificially compensated, as if it were a fixed bias of some sort, because the amount of bias varies. In the detection of an unknown frequency, these frequency errors will severely limit the resolving power of the estimator, requiring excessively long window lengths to accurately resolve the frequency.

The Spectrum of the Sum of Two Real Sinewaves

To better understand this phenomena of aliasing between frequency components of a signal, consider a signal consisting of the sum of two sinusoids of arbitrary frequencies and phases,

$$s(t) = \cos(w_1 t + \emptyset + \emptyset_1) + \cos(w_2 t + \emptyset + \emptyset_2) \ . \qquad (24)$$

The phase angle, $\emptyset$, common to both sinewaves, will denote the general phase of the signal with respect to the window. This angle can be varied to bring any portion of the signal into the window. The individual phase angles, $\emptyset_1$ and $\emptyset_2$, denote the phase of each sinewave with respect to each other, giving some flexibility to the signal description. Obviously, a simple case is both $\emptyset_1$ and $\emptyset_2$ equal to zero.

The Fourier transform of this signal consists of impulses located at +/-w_1 and +/-w_2, that is,

$$S(w) = \pi[e^{j(\emptyset+\emptyset_1)}\,d(w-w_1) + e^{-j(\emptyset+\emptyset_1)}\,d(w+w_1)] + \pi[e^{j(\emptyset+\emptyset_2)}\,d(w-w_2) + e^{-j(\emptyset+\emptyset_2)}\,d(w+w_2)] \quad (25)$$

The spectrum of the windowed version of this signal, computed as before by convolving S(w) with Equation 22, is

$$S_w(w) = \pi T[e^{j(\emptyset+\emptyset_1)}\,Sinc(x - x_1) + e^{-j(\emptyset+\emptyset_1)}\,Sinc(x + x_1)] + \pi T[e^{j(\emptyset+\emptyset_2)}\,Sinc(x - x_2) + e^{-j(\emptyset+\emptyset_2)}\,Sinc(x + x_2)] , \quad (26)$$

where

$$x = wT/2, \quad x_1 = w_1T/2, \text{ and } x_2 = w_2T/2 .$$

The peak in the spectrum for each sinewave will be shifted due to aliasing. However, there are two contributing factors to this peak shifting. First, there is the usual aliasing between positive and negative frequency components. Second, there is additional "in-band" aliasing between the two positive components and the two negative components, more traditionally termed leakage. Both types of aliasing are a function of the phase of the signal, $\emptyset$, the phases $\emptyset_1$ and $\emptyset_2$, and the window length. This is depicted in Figure 6, where the magnitude spectrum for each frequency

component is plotted separately; the overlapping regions indicating areas where aliasing occurs. The composite spectrum is plotted in Figure 7. The spectral errors associated with real signals arise from the windowing process itself, and ultimately limit the resolution which can be achieved with a particular window length.

Analysis of a Complex Sinewave

The obvious remedy to completely eliminate the aliasing between positive and negative frequency components is to remove the negative frequency components via an analytic signal representation. For the single cosinusoid of Equation 19, the analytic counterpart can be formed from Equation 3 and Equation 4,

$$s_a(t) = \cos(w_1 t + \emptyset) + j\cos(w_1 t + \emptyset) * (1/\pi t),$$

$$\begin{aligned} S_a(w) &= \pi[e^{j\emptyset} d(w-w_1) + e^{-j\emptyset} d(w+w_1)] \\ &\quad + \pi j[-je^{j\emptyset} d(w-w_1) + je^{-j\emptyset} d(w+w_1)] \\ &= 2\pi[e^{j\emptyset} d(w-w_1)] . \end{aligned} \tag{27}$$

Performing an inverse Fourier transform, we find that

$$s_a(t) = e^{jw_1 t + \emptyset} . \tag{28}$$

As expected, the analytic signal corresponding to a single sinewave is a complex exponential, and has only positive frequency components. The spectrum of the windowed version

of $s_a(t)$, $s_{aw}(t)$, is found as follows:

$$s_{aw}(t) = s_a(t)W(t) ,$$

$$S_{aw}(w) = S_a(w) * W(w)$$

$$= 2\pi T \, \mathrm{Sinc}(x - x_1) \, e^{j\emptyset} , \qquad (29)$$

where

$$x = wT/2 \text{ and } x_1 = w_1T/2 .$$

Recall, since this is an analytic signal, there is only a contribution from the positive frequency component.

Clearly, the peak in the magnitude spectrum always occurs at $w = w_1$, since the Sinc function is a maximum when its argument is zero, and because the negative spectral components of $s_{aw}(t)$ have been removed. This is depicted in Figure 8, where the magnitude of the spectrum of $s_{aw}(t)$ is plotted versus frequency for a 1000 Hz sinewave of zero phase, with a 1.3 ms long window. The analytic signal spectrum gives higher resolution in the sense that the peak in the magnitude spectrum is only dependent on the structure of the window, and not on the position of the window in time, the phase of the signal, or the length of the window.

Analysis of the Sum of Two Complex Sinewaves

The analytic signal alone, however, is not the failsafe remedy for detecting windowed sinewaves. Consider the signal of Equation 24, whose analytic windowed counterpart is

$$s_{aw}(t) = [e^{j(w_1 t+\emptyset+\emptyset_1)} + e^{j(w_2+\emptyset+\emptyset_2)}]W(t) . \qquad (30)$$

The spectrum of this signal is

$$S_{aw}(w) = 2\pi T e^{j\emptyset} [e^{j\emptyset_1} \text{Sinc}(x - x_1) + e^{j\emptyset_2} \text{Sinc}(x - x_2)] . \qquad (31)$$

The peaks in the spectrum are once again shifting as a function of the phases and the window length. This shifting, however, is due only to aliasing among the positive frequency components, as depicted in Figure 9. In Figure 10, the composite spectrum is shown. This peak, however, is independent of the phase of the signal, $\emptyset$, which controls the portion of the signal within the window. Thus, the error in the frequency estimates will be fixed if the signals themselves have a fixed phase shift relative to one another ($\emptyset_1$ and $\emptyset_2$ are constant). As $\emptyset$ varies, or as the window slides along the signal, the magnitude spectrum remains fixed. This type of error is a fixed bias.

Therefore, the analytic signal, for the sinusoidal signals above, has the phase information embedded in such a manner that the spectral estimates are consistent and independent of time. This is a property of the Hilbert transform, in that it draws information about the signal outside the window, into the window, making the analytic signal phase coherent, in a sense, with the window. Real

signals, on the other hand, experience significant time variations as shown above.

Spectral Stationarity of an Analytic Signal

Just how general is this phase invariance property of the magnitude spectrum of an analytic signal? For any signal that is a sum of sinewaves, these results will hold true. This includes all periodic signals, since a periodic signal can be represented as a sum of sinusoids via a Fourier series. This property is more general, however, and can be applied to other real signals, of which sinewaves are just a special case.

Let $s_a(t)$ be an analytic signal which satisfies the Dirichlet conditions [25] (implying that its Fourier transform exists). The window, W(t), can represent any type of window function, provided it is of finite duration, and its Fourier transform exists; in general, let

$$\begin{aligned} W(t) &= f(t) , \quad -T/2 < t < T/2 , \\ &= 0 , \qquad \text{elsewhere} . \end{aligned} \tag{32}$$

Let $s_{ap}(t)$ be a phase rotated version of $s_a(t)$, where phase rotated implies some fixed phase shift is added to the Fourier phase spectrum, such that

$$s_{ap}(t) = e^{j\emptyset} s_a(t) , \tag{33}$$

and,

$$S_{ap}(w) = S_a(w) e^{j\emptyset} .$$

Then, the spectrum of the windowed version of this signal, s_{apw} is given by

$$S_{apw}(w) = S_{ap}(w) * W(w)$$
$$= e^{j\emptyset} [S_a(w) * W(w)] . \quad (34)$$

The magnitude spectrum, given by

$$|S_{apw}(w)| = |S_{aw}(w) * W(w)| , \quad (35)$$

is independent of the phase rotation. The effect of this phase rotation upon the analytic signal can be thought of in three dimensions as a rotation of the signal about its time axis, similar to the rotation of a screw .

This phase invariance property does not hold for real windowed signals. The spectrum of the real signal corresponding to Equation 33 can be found by noting that

$$s_p(t) = Re\{s_{ap}(t)\}$$
$$= (1/2) \{s_{ap}(t) + s^*_{ap}(t)\}$$
$$= (1/2) \{s_a(t)e^{j\emptyset} + s^*_a(t)e^{-j\emptyset}\} .$$

The Fourier transform of $s_p(t)$ is

$$S_p(w) = (1/2)[S_a(w)e^{j\emptyset} + S^*_a(-w)e^{-j\emptyset}] . \quad (36)$$

The spectrum of the windowed version of this signal is

$$S_{pw}(w) = S_p(w) * W(w)$$
$$= (1/2)[e^{j\emptyset} S_a(w) * W(w) + e^{-j\emptyset} S_a^*(-w) * W(w)] . \quad (37)$$

The aliasing between positive and negative frequency components appears in the presence of the exponential term $e^{j\emptyset}$ and its conjugate, $e^{-j\emptyset}$. The magnitude spectrum of this signal is a function of the angle $\emptyset$.

In the case of periodic signals, or sums of periodic signals, this phase invariance implies the magnitude spectrum of the analytic signal will be constant versus time, since this phase shift is equivalent to a time delay. This is precisely the cases presented above, in the form of the sinewaves, where the angle $\emptyset$ denoted both the phase shift and the time delay. For other signals, this phase shift can significantly change the appearance of the real signal in the time domain. An interesting example is the case of the ideal low-pass filter impulse response, the sin(t)/(t) function. The analytic signal, with a variable phase shift, which corresponds to this real signal is

$$s_{ap}(t) = e^{j\emptyset} [\sin(t)/(t) + j(1-\cos(t))/t]. \quad (38)$$

By varying the angle $\emptyset$, the real part of this signal can change dramatically, as shown in Figure 11, where this real signal is plotted for $\emptyset$ = 0, 45, 90, 135, 180 degrees. The magnitude spectrum of the windowed analytic signal will be

the same for all of these signals, while in the real case, there will be the usual variations. In time series analysis, where continuous signals are always represented by as small a segment of the signal as possible, the analytic signal becomes the natural choice for improved spectral resolution.

CHAPTER IV

ANALYSIS OF FINITE LENGTH DISCRETE SEQUENCES

Introduction

The effects of windowing on a signal are most pronounced in the discrete domain, where operations such as spectral estimation are routinely performed on a finite segment of a signal. The equivalent of the analog Fourier transform in the discrete domain is termed the discrete Fourier transform (DFT), or, when performed computationally efficient, the fast Fourier transform (FFT). This chapter presents a comparison of the performance of FFT-based spectral estimation for real and analytic signals. Actually, it is of no surprise that the basic results of the previous chapter hold, with only slight differences due to the nature of the signal representation. The results presented here were observed experimentally by Hartwell [3] using a DFT.

The Discrete Spectrum

In the discrete time domain, the spectrum of a discrete signal, or sequence, can be represented via a sum of weighted complex exponentials called the discrete Fourier transform [5]. The accuracy of this transform is related to the type of sequence being analyzed. A periodic sequence can be exactly represented by a discrete Fourier series, similar to the analog case in which a periodic signal is

expanded in a Fourier series. For an aperiodic sequence, in general, the appropriate representation is a discrete Fourier transform, which must be computed over the entire extent of the sequence. If the sequence is not time limited, this involves analyzing an infinite set of data points.

The classic question then arises: Given a segment of a sequence, how accurately can the spectrum of that sequence be computed? If the sequence is periodic, and the period is known, or if the sequence is of finite duration, the spectrum can be computed exactly [5]. Otherwise, it can only be approximated by one of several techniques to be subsequently discussed. The most popular of these is the fast Fourier transform mentioned above, or equivalently, the sampled z-transform.

A reasonable approach to estimating or detecting the frequency of a signal, with no a priori information about its frequency content, is to again examine the magnitude spectrum, as determined from a sampled z-transform. An ordinary discrete Fourier transform and the inverse discrete Fourier transform are defined as

$$F(e^{jk2\pi/NT}) = \sum_{n=0}^{N-1} f(nT)e^{-j2\pi nk/N}, \quad 0 \leq k \leq N-1,$$

$$= 0, \quad \text{elsewhere.}$$

$$f(nT) = (1/N) \sum_{k=0}^{N-1} F(e^{jk2\pi/N}) e^{j2\pi nk/N}, \quad 0 \leq n \leq N-1,$$

$$= 0, \quad \text{elsewhere,}$$

where N is the number of samples spaced T seconds apart. Observe the resolution of this transform in the frequency domain is only 1/NT Hz (the sample rate divided by the number of points used to compute the transform), as values of the frequency response are given only at the so-called orthogonal frequencies, k/NT Hz.

To obtain further resolution, that is, values of the frequency response between orthogonal frequencies, this transform can be interpolated in the frequency domain. This process of interpolation can be efficiently performed using a sampled z-transform, defined as

$$F(e^{jw}) = \sum_{n=0}^{N-1} f(nT)\, e^{-jnwT} . \qquad (39)$$

The transform is now a continuous function of frequency. However, for this interpolation to be exact, it is assumed that the sequence is a finite length sequence N points long, making the frequency response of the sampled z-transform exact. In practice, this is generally not the case, yet the latter representation of the spectrum is a reasonable one, since to some extent, the true spectrum of the signal is being approximated through a frequency domain interpolation process [5]. Thus, the problem reduces to analyzing the spectrum of a (at present) rectangularly windowed sequence.

The Discrete Spectrum of Multiple Sinewaves

Let the sequence s(nT) be composed of a sum of cosinewaves of arbitrary frequencies and phases, that is,

$$s(nT) = \sum_{i=1}^{Q} \cos(nw_iT + \emptyset + \emptyset_i) \ . \tag{40}$$

The spectrum is found from Equation 38 to be

$$S(e^{jw}) = \sum_{n=0}^{N-1} \sum_{i=1}^{Q} \cos(nw_iT + \emptyset + \emptyset_i)e^{-jnwT}$$

$$= \sum_{i-1}^{Q} \sum_{n=0}^{N-1} (1/2)(e^{j(nw_iT + \emptyset + \emptyset_i)} + e^{-j(nw_iT + \emptyset + \emptyset_i)} e^{-jnwT}$$

$$= (1/2) \sum_{i-1}^{Q} [\, e^{j(\emptyset + \emptyset_i)} \sum_{n=0}^{N-1} e^{jnT(w_i-w)} + e^{-j(\emptyset + \emptyset_i)} \sum_{n=0}^{N-1} e^{-jnT(w_i+w)} \,] \ . \tag{41}$$

Using the identity [5],

$$\sum_{n=0}^{N-1} a^{-n} = (1 - a^{-N})/(1 - a^{-1}) \ , \tag{42}$$

Equation 41 can be simplified to

$$S(e^{jw}) = (1/2) \sum_{i=1}^{Q} [\, \frac{1 - e^{-j(w - w_i)NT}}{1 - e^{-j(w - w_i)T}} e^{j\emptyset} e^{j\emptyset_i} + \frac{1 - e^{-j(w + w_i)NT}}{1 - e^{-j(w + w_i)T}} e^{-j\emptyset} e^{-j\emptyset_i} \,] \ . \tag{43}$$

Observe that the first term corresponds to the positive frequency components of the sequence, while the second term corresponds to the negative frequency components.

In the analytic case, the cosinewaves are replaced by complex exponentials, and the complex sequence has only the positive frequency component term, specifically,

$$s_a(t) = \sum_{i=1}^{Q} e^{j(nw_iT + \emptyset + \emptyset_i)}, \qquad (44)$$

and

$$S_a(e^{jw}) = e^{j\emptyset} \sum_{i=1}^{Q} \left[\frac{1 - e^{-j(w - w_i)NT}}{1 - e^{-j(w - w_i)T}} e^{j\emptyset_i} \right]. \qquad (45)$$

Recall, the phase shift $\emptyset$ actually represents the position of the window in time, while the number of data points, N, represents the length of the window (NT seconds).

It is instructive to examine Equations 43 and 45 for the case of a single cosinewave. When Q = 1, Equation 43 simplifies to

$$S(e^{jw}) = (1/2) \left[\frac{1 - e^{j(w - w_1)NT}}{1 - e^{j(w - w_1)T}} e^{j\emptyset} e^{j\emptyset_1} + \frac{1 - e^{-j(w + w_1)NT}}{1 - e^{-j(w + w_1)T}} e^{-j\emptyset} e^{-j\emptyset_1} \right], \qquad (46)$$

while Equation 45 simplifies to

$$S_a(e^{jw}) = e^{j\emptyset} \frac{1 - e^{-j(w - w_1)NT}}{1 - e^{-j(w - w_1)T}} e^{j\emptyset_1} . \qquad (47)$$

The magnitude spectrum of the real cosinewave from Equation 46 is seen to be a function of the phase angle, Ø. The presence of the $e^{j\emptyset}$ term and its conjugate indicate the usual aliasing between positive and negative frequency components of the real signal. The peak in this spectrum will shift as a function of Ø.

This is depicted in Figure 12, where this peak location is plotted versus window length, N, for Ø = 0 and Ø = 90 degrees, and in Figure 13, where this peak location is plotted versus phase, Ø, for N = 20, 40, 80, and 160 points. The frequency of the cosinewave is 313 Hz, a non-orthogonal frequency for all the window lengths used, and the sampling rate is 8 kHz. Note that the magnitude of the error in the peak location generally decreases as the frame length increases, indicating that accurate estimation requires more cycles of the signal within the window. As the window length is increased, the aliasing between positive and negative frequency components is reduced, as it was for the analog case of Chapter III.

The nature of this error produced by the sampled z-transform should be emphasized. For a given value of frequency, w_1, particular values of N will produce a correct

estimate. These values occur when the frequency w_1 falls on an orthogonal frequency. In this case, the window length is exactly an integer multiple of one period of the sinewave, and the sampled z-transform produces values of the frequency response at orthogonal frequencies which are equivalent to the discrete Fourier series [5]. Further, the discrete Fourier transform is actually equivalent to a discrete Fourier series, in that the representation of the signal as being periodic outside the window is correct. When the frequency of the sequence is not an orthogonal frequency, errors due to the windowing process exist, since the assumptions about the signal outside the window are not correct.

The analytic signal, on the other hand, produces an exact estimate of the peak location, due to the absence of frequency component aliasing. While this is more difficult to show because of the complexity of Equation 47, this equation is easily evaluated. In Figure 14, the peak location is plotted versus window length, for the same signal and sampling as in the real case. The error is zero for all values of the window length. Once again, this is not the case for the multiple sinusoids of Equation 45, as the aliasing among positive frequency components still exists. The phase invariance of the analytic signal is still preserved, however, since the magnitude spectrum of Equation 45 is not a function of Ø. As before, this phase invariance is a property of the analytic signal

representation. The estimation errors in the spectrum of the analytic signal will be less severe than in the real case, since analytic signals only suffer from positive frequency component aliasing.

Time Domain Windowing

This problem of rectangular windowing was recognized very early in the digital signal processing arena [6-8]. The classic ad hoc solution involves the use of a time domain window. The aliasing due to the rectangular window can be partially eliminated by using a non-rectangular window with a frequency response which is effectively much narrower than the frequency response of a rectangular window. Windows such as the Hamming, Hann, or Bartlett windows are all commonly used to smooth data before processing. These windows tend to reduce the effects of both types of aliasing. It is of no surprise that the time domain shape of these windows goes to zero at the edges of the window. These windows serve only to make the signal inside the window appear to be a time limited signal, thereby, allowing the sampled z-transform to be more accurate at non-orthogonal frequencies. The frequency response of these windows is a function of their length, however, the longer the window, the narrower the frequency response. A large number of points is required to obtain a window with a narrow response. Also, these windows distort any non-stationarities in the signal, as these

non-stationarities tend to be smoothed, or averaged out.

Burg [9] posed a different question: Why assume the signal to be zero outside the window, when in fact we know it not to be? Further, in his maximum entropy formulation, he hypothesizes that it is more natural to choose a signal of maximum entropy which satisfies the data constraints [9]. Intuitively, it is difficult to justify "corrupting" perfectly good data samples by applying a non-rectangular window. The analytic signal can actually be viewed as an alternate solution to the problem of estimating a sequence from only a finite number of samples. It is attractive in that the original data is retained, and phase invariance is added via the Hilbert transformer. It is elegant in that the primary aliasing effects are eliminated. To obtain further resolution, and limit the aliasing among positive frequency components, a non-rectangular window can be used, as with real signals, but this is generally not required.

CHAPTER V

LINEAR PREDICTION USING AN ANALYTIC SIGNAL

Introduction

The desire to accurately estimate the spectrum of a signal at non-orthogonal frequencies, as well as model these spectra as a smooth function of frequency (similar to the spectra normally displayed on an analog spectrum analyzer), led to the development of parametric spectral estimators. While there are many classes of these types of estimators, they are all essentially derivatives of the minimum mean-squared error estimator drawn from least-squares system identification theory [10,26,27]. The estimation of an autoregressive process in a general least-squares sense shall be considered, with applications centered around the problem of the detection of sinewaves in additive noise. While the techniques presented here are certainly not optimal estimators for this type of signal, they will serve to demonstrate the improved resolution that can be achieved using an analytic signal. Experimental results are presented for the popular autocorrelation technique [28-30], and the numerically robust Burg lattice-based algorithm [31].

The Complex Linear Predictor

While the complex linear predictor can be easily derived from the orthogonality principle [32], an alternate

derivation is presented here which highlights the least mean-square error nature of linear prediction. Let the complex sequence be

$$s(nT) = s_r(nT) + j\, s_i(nT) \ . \tag{48}$$

Suppose this signal can be approximated by a linear combination of p previous values (that is, linearly predicted). The predicted value can be written with complex prediction coefficients as

$$\tilde{s}(nT) = \sum_{i=1}^{p} a_i s((n-i)T) \ , \tag{49}$$

$$= \sum_{i=1}^{p} (u_i + jv_i)s((n-i)T) \ . \tag{50}$$

The prediction error is the difference between the actual value, $s(nT)$, and its predicted value, $\tilde{s}(nT)$. The prediction error is a complex quantity, and its energy can be computed by summing its squared magnitude over all time. In time series analysis, for practical reasons, these computations are restricted to a finite interval of N points. Thus, let the prediction error energy be defined as

$$E_n = \sum_{m=0}^{N-1} e(mT)e^*(mT) \ ,$$

$$= \sum_{m=0}^{N-1} [s(mT) - \tilde{s}(mT)][s(mT) - \tilde{s}(mT)]^* \ ,$$

$$= \sum_{m=0}^{N-1} [s(mT)s^*(mT) - s(mT)\tilde{s}^*(mT) - \tilde{s}(mT)s^*(mT) + \tilde{s}(mT)\tilde{s}^*(mT)] \quad . \qquad (51)$$

In order to find the optimal predictor parameters, this prediction error energy must be minimized.

The error energy can be minimized by finding the predictor parameters that minimize the gradient of E_n, defined as

$$\text{grad } E_n = dE_n/du_r + j dE_n/dv_r \quad , \qquad (52)$$

where dE_n/du_r denotes the partial derivative of the prediction error at time nT with respect to a prediction parameter, u_r, where $1 \leq r \leq p$. The energy error minimization implies solving

$$|\text{grad } E_n| = 0 \quad . \qquad (53)$$

While this error can be minimized over several different intervals, the approach presented here minimizes the error over the interval [0,N-1], which represents a typical segment, or frame of a sequence.

By substituting Equation 50 into Equation 51, the error energy can be written in terms of the components of the prediction parameters, u_i and v_i, i.e.,

$$E_n = \sum_{m=0}^{N-1} \{[\, s(mT)s^*(mT)$$

$$- s(mT) \sum_{i=1}^{p} (u_i - jv_i)s^*((m-i)T)$$

$$- s^*(mT) \sum_{i=1}^{p} (u_i + jv_i)s((m-i)T)$$

$$+ [\sum_{i=1}^{p} (u_i + jv_i)s((m-i)T)] \; x$$

$$[\sum_{i=1}^{p} (u_i - jv_i)s^*((m-i)T)]\} \; . \qquad (54)$$

Setting the partial derivative of Equation 54 with respect to u_r equal to zero, for $1 \leq r \leq p$, we obtain

$$\sum_{m=0}^{N-1} \{-s(mT)s^*((m-r)T) - s^*(mT)s((m-r)T)$$

$$+ s((m-r)T)[\sum_{i=1}^{p} (u_i - jv_i)s^*((m-i)T)]$$

$$+ s^*((m-r)T)[\sum_{i=1}^{p} (u_i + jv_i)s((m-i)T)] \} = 0 \; , \qquad (55)$$

or, upon simplification,

$$\sum_{m=0}^{N-1} -\{ s(mT)s^*((m-r)T) + s((m-r)T)s^*(mT) \}$$

$$+ \sum_{i=1}^{p} a_i^* \sum_{m=0}^{N-1} s((m-r)T)s^*((m-i)T)$$

$$+ \sum_{i=1}^{p} a_i \sum_{m=0}^{N-1} s^*((m-r)T)s((m-i)T) = 0 \; , \qquad (56)$$

again $1 \leq r \leq p$. This gives a system of p linear equations and p unknown coefficients.

Equation 56 can be further simplified by defining the short-term complex autocorrelation function as

$$R(r,i) = \sum_{m=0}^{N-1} s((m-r)T)s^*((m-i)T) \ . \tag{57}$$

Equation 56 then simplifies to

$$-R(0,r) - R^*(0,r) + \sum_{i=1}^{p} [a_i^* R(r,i) + a_i R^*(r,i)] = 0. \tag{58}$$

Repeating the above procedure for the partial derivative of Equation 54 with respect to v_r, we find

$$jR(0,r) - jR^*(0,r) + \sum_{i=1}^{p} [ja_i^* R(r,i) - ja_i R^*(r,i)] = 0. \tag{59}$$

Combining Equations 58 and 59 into Equation 53, we see that

$$R(0,r) = \sum_{i=1}^{p} a_i R^*(r,i) \ , \quad 1 \leq r \leq p \ . \tag{60}$$

Equation 60, of course, is known as the Yule-Walker equation [20] which we see holds true for complex predictor coefficients. This technique of determining the LPC parameters is commonly called the covariance method [19], though it really is just the least-squares solution to a system identification problem [26].

Equation 60 can be written in matrix notation as shown below for the case of p = 2,

$$\begin{bmatrix} R(0,1) \\ R(0,2) \end{bmatrix} = \begin{bmatrix} R^*(1,1) & R^*(1,2) \\ R^*(2,1) & R^*(2,2) \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \end{bmatrix} . \qquad (61)$$

Noting that $R^*(i,j) = R(j,i)$, and $R^*(i,i) = R(i,i)$, we obtain

$$\begin{bmatrix} R(0,1) \\ R(0,2) \end{bmatrix} = \begin{bmatrix} R(1,1) & R^*(1,2) \\ R(1,2) & R(2,2) \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \end{bmatrix} . \qquad (62)$$

The square matrix in Equation 62, being Hermitian [33] but not Toeplitz [34], has properties which allow its inversion to be performed efficiently [35]. While neither computational efficiency nor stability considerations will be discussed here in great detail, two popular variations to Equation 62 will be presented, the autocorrelation method and the Burg algorithm. The autocorrelation method [28-30] guarantees stability of the linear predictor. The Burg algorithm [31] guarantees stability and is also numerically robust.

The short-term autocorrelation function can be computed only over the N point interval, assuming the sequence is zero outside the interval; in this case it is given by

$$R(i) = \sum_{m=i}^{N-1} s(mT)s^*((m-i)T) , \quad 1 \leq i \leq p . \qquad (63)$$

Equation 60 then simplifies to

$$R(r) = \sum_{i=1}^{p} a_i R^*(i-r) \ , \quad 1 \leq r \leq p \ . \tag{64}$$

This is known as the autocorrelation method, and while it departs from the least squares solution, it has the property of producing a linear predictor whose poles are always stable. In certain applications like speech coding, this is of primary importance. Further, Durbin showed that this formulation can be computed in an efficient recursive manner [35], since the square matrix in Equation 62 is now both Hermitian and Toeplitz. Also, as a consequence of the Durbin recursion, Itakura [30] showed that this computation can be performed in a lattice-type structure, where predictor parameters are replaced by reflection coefficients, computed as

$$k_i = \frac{\sum_{m=0}^{N-1} e_{(i-1)}(m)\, b^*_{(i-1)}(m-1)}{\left\{ \sum_{m=0}^{N-1} |e_{(i-1)}(m)|^2 \sum_{m=0}^{N-1} |b_{(i-1)}(m-1)|^2 \right\}^{(1/2)}} \ . \tag{65}$$

The form of this lattice filter is shown in Figure 15. Makhoul has shown that Equation 65 is only a special case of a generalized lattice structure [29]. Lattice methods are very powerful tools, since they are numerically well-conditioned [20], and computationally efficient [10].

The reflection coefficients of a lattice filter can be converted to predictor coefficients using the Durbin recursion [20,35]:

$$a_{i,i} = k_i \tag{66}$$

$$a_{i,j} = a_{i-1,j} - k_i a^*_{i-1,i-j} , \quad 1 \leq j \leq i-1 . \tag{67}$$

This procedure is performed recursively for $i = 1, 2, \ldots, p$, and when $i = p$, the final predictor coefficients are obtained.

Finally, Burg theorized that by choosing a different constraint, other than the usual mean-squared error norm, the reflection coefficients could be determined directly from the signals within the lattice filter [31]. By minimizing the sum of forward and backward error signals in the lattice, $e_i(m)$ and $b_i(m)$, the reflection coefficients can be calculated as

$$k_i = \frac{2 \sum_{m=0}^{N-1} e_{(i-1)}(m)\, b^*_{(i-1)}(m-1)}{\left\{ \sum_{m=0}^{N-1} |(e_{(i-1)}(m))|^2 + \sum_{m=0}^{N-1} |(b_{(i-1)}(m-1))|^2 \right\}} . \tag{68}$$

Note that for these formulations, involving different reflection coefficients constrains, the complex reflection coefficients always have a magnitude less than unity. This is a necessary and sufficient condition for the stability of the linear predictor [9,20]. Also, these complex reflection coefficients can be computed and converted to predictor

coefficients while performing only scalar divisions. It should be emphasized that these formulations produce coefficients that are different from the least-squares solution. Once again, it is not the attempt here to advocate one particular technique, but rather to demonstrate the advantages of using a complex predictor over a real predictor.

Derivation of the Step-Down Procedure

The Durbin recursion that converts predictor parameters to reflection coefficients is also called the step-up procedure [20]. A similar procedure for converting predictor parameters to reflection coefficients, called the step-down procedure, exists in the real case [20], and here is extended to the complex case.

The z-transform of the forward prediction error at stage i of the lattice of Figure 15 can be written as

$$E_i(z) = A_i(z)S(z) \ , \tag{69}$$

where

$$A_i(z) = 1 - \sum_{j=1}^{i} a_j z^{-j} \ , \tag{70}$$

and

$$S(z) = Z\{s(mT)\} \ . \tag{71}$$

Substituting the recursion relations of Equations 66 and 67 into Equation 70, where $a_j = a_{i,j}$, we obtain

$$A_i(z) = 1 - \sum_{j=1}^{i-1} a_{i,j} z^{-j} - k_i z^{-i} \tag{72}$$

$$= 1 - \sum_{j=1}^{i-1} [a_{i-1,j} - k_i \, a^*_{i-1,i-j}] \, z^{-j} - k_i z^{-i}$$

$$= A_{i-1}(z) - [k_i z^{-i} - k_i \sum_{j=1}^{i-1} a^*_{i-1,i-j} \, z^{-j}]$$

$$= A_{i-1}(z) - k_i z^{-i} [\, 1 - \sum_{j=1}^{i-1} a^*_{i-1,i-j} \, z^{i-j}]$$

$$A_i(z) = A_{i-1}(z) - k_i z^{-i} \, \bar{A}_{i-1}(1/z) \, , \tag{73}$$

where $\bar{A}_{i-1}(1/z)$ is defined as

$$\bar{A}_{i-1}(1/z) = 1 - \sum_{j=1}^{i-1} a^*_{i-1,i-j} z^{i-j} \, . \tag{74}$$

From Equation 69, the prediction error can be written as

$$E_i(z) = A_{i-1}(z)S(z) - k_i z^{-i} \, \bar{A}_{i-1}(1/z)S(z) \, . \tag{75}$$

Note that the first term is essentially the error at stage i-1.

Let the backward prediction error be defined from the second term of Equation 75, that is,

$$B_i(z) = z^{-i} \, \bar{A}_i(1/z)S(z) \tag{76}$$

$$= z^{-i} [\, 1 - \sum_{j=1}^{i} a^*_{i,i+1-j} z^{i+1-j} \,] \, S(z) \, . \tag{77}$$

The inverse z-transform of Equation 77 yields, after an index change,

$$b_i(m) = s(m-i) - \sum_{j=1}^{i} a^*_{i,j}\, s(m+j-i) \quad . \tag{78}$$

The backward prediction error is seen as the error in predicting the sample at time m-i from all future samples.

The z-transform of the backward prediction error at stage i-1 can be written as

$$B_{i-1}(z) = z^{-i+1}\, \bar{A}_{i-1}(1/z)S(z) \tag{79}$$

$$= z^{-i+1} \left[1 - \sum_{j=1}^{i-1} a^*_{i-1,j} z^j \right] S(z) \quad . \tag{80}$$

The backward prediction error at this stage is then given as

$$b_{i-1}(m) = s(m-i+1) - \sum_{j=1}^{i-1} a^*_{i-1,j}\, s(m+j-i+1) \quad . \tag{81}$$

The z-transform of the forward error, using Equations 73 and 75, can be written as

$$E_i(z) = E_{i-1}(z) - k_i z^{-1} [\, z^{-i+1}\, \bar{A}_{i-1}(1/z)S(z)] \quad . \tag{82}$$

Using Equation 79, the prediction error is found from the inverse z-transform to be

$$e_i(m) = e_{i-1}(m) - k_i b_{i-1}(m-1) \quad . \tag{83}$$

Thus, the lattice can be viewed as the combination of a forward and backward predictor.

An expression for the backward prediction error similar to Equation 83 can be found by combining Equations 73 and 76

as follows:

$$B_i(z) = z^{-i}[\ \bar{A}_{i-1}(1/z) - k_i^* z^i A_{i-1}(z)\ ]\ S(z)$$

$$= z^{-1}z^{-i+1}\ \bar{A}_{i-1}(1/z)S(z) - k_i^* A_{i-1}(z)S(z)$$

$$= z^{-1}\ B_{i-1}(z) - k_i^* A_{i-1}(z)S(z)\ .$$

After substituting Equation 69 into Equation 84, the inverse z-transform will yield

$$b_i(m) = b_{i-1}(m-1) - k_i^* e_{i-1}(m)\ . \qquad (84)$$

Equations 83 and 84, of course, allow the reflection coefficients to be computed in the lattice filter formulation of Figure 15 via Equation 65 or 68.

The step-down procedure can now be obtained using the relationships between the forward and backward predictors, and one additional relationship for $\bar{A}_i(1/z)$ that follows from Equation 73, namely,

$$\bar{A}_i(1/z) = \bar{A}_{i-1}(1/z) - k_i^* z^i A_{i-1}(z)\ . \qquad (85)$$

Next, substituting Equation 85 into Equation 73, we obtain

$$A_i(z) = A_{i-1}(z)[1 - |k_i|^2] - k_i z^{-i}\bar{A}_i(1/z)\ . \qquad (86)$$

Solving for $A_{i-1}(z)$, we find that

$$A_{i-1}(z) = [A_i(z) - k_i z^{-i}\bar{A}_i(1/z)]/[1 - |k_i|^2]\ , \qquad (87)$$

where $k_i \neq 1$.

Substituting Equations 70 and 74 into Equation 87 gives the derived result,

$$1 - \sum_{j=1}^{i-1} a_{i-1,j} z^{-j} = \{1 - \sum_{j=1}^{i} a_{i,j} z^{-i} + k_i z^{-i} \quad x$$

$$[1 - \sum_{j=1}^{i} a^{*}_{i,i+1-j} z^{i+1-j}]\}/[1 - |k_i|^2] \ . \tag{88}$$

Equation 88 is solved recursively by equating polynomial coefficients for $i = p, p-1, \ldots, 1$ to obtain a set of reflection coefficients from a set of predictor coefficients. The computational procedure can be summed up [20] as follows:

$$\begin{aligned} &a_{i-1,j} = a_{i,j} + k_j a^{*}_{i,i-j} \ , \\ &\text{with } k_i = a_{i,i} \ , \\ &\text{for } i = p,p-1,\ldots,1, \text{ and } j = 0,1,\ldots,i-1, \end{aligned} \tag{89}$$

where $a_{i,0} = 1$ for $1 \leq i \leq p$ and we shall be guaranteed that $|k_i| < 1$. The lattice filter realization is equivalent to the FIR realization, from a digital filtering standpoint.

Phase Invariance of the Analytic Signal-Based Linear Predictor

The phase invariance property of analytic signals which has been developed in the previous chapters can be extended to the case of linear prediction. Let the phase rotated

analytic sequence, $s_{ap}(nT)$, be defined as the sampled data sequence corresponding to the analog signal of Equation 33. The short-term autocorrelation function, computed from Equation 57, is found to be

$$R_{ap}(r,i) = \sum_{m=0}^{N-1} e^{j\emptyset} s_a((m-r)T) e^{-j\emptyset} s_a^*((m-i)T)$$

$$= \sum_{m=0}^{N-1} s_a((m-r)T) s_a^*((m-i)T)$$

$$= R_a(r,i) \ . \qquad (90)$$

The short-term autocorrelation function for analytic signals is thus independent of the phase angle, $\emptyset$.

This result is not altogether unexpected, as linear predictive coding (LPC), by design, is blind to the phase of the signal. The LPC parameters determine an FIR filter which, when stable, is constrained to be of minimum phase [9,10], regardless of whether or not the sequence itself is of minimum phase. The autocorrelation function and the power spectral density are a Fourier transform pair, and the power spectral density is related to the squared-magnitude of the Fourier transform. It is expected, then, that the autocorrelation should be independent of the phase of the signal, since the magnitude spectrum is also independent of this phase.

Since the short-term autocorrelation function is computed over a finite interval, this function for a real signal can easily be shown not to be phase invariant.

Starting with Equation 36, the short-term autocorrelation function for the real signal can be derived as follows:

$$s_p(nT) = (1/2) \left[e^{j\emptyset} s_a(nT) + e^{-j\emptyset} s_a^*(nT) \right] , \qquad (91)$$

$$R(r,i) = \sum_{m=0}^{N-1} s_p((m-r)T) s_p^*((m-i)T)$$

$$= (1/4) \sum_{m=0}^{N-1} \left[e^{j\emptyset} s_a((m-r)T) + e^{-j\emptyset} s_a^*((m-r)T) \right] x$$
$$\left[e^{-j\emptyset} s_a^*((m-i)T) + e^{j\emptyset} s_a((m-i)T) \right]$$

$$= (1/4) \sum_{m=0}^{N-1} \left[s_a((m-r)T) s_a^*((m-i)T) \right.$$
$$+ s_a^*((m-r)T) s_a((m-i)T)$$
$$+ e^{2j\emptyset} s_a((m-r)T) s_a((m-i)T)$$
$$\left. + e^{-2j\emptyset} s_a^*((m-r)T) s_a^*(m-i)T) \right] . \qquad (92)$$

Due to the windowing process, a dependence on the phase angle, $\emptyset$, exists. In the limit, as N (effectively the window width) goes to infinity, these phase effects average out. However, with periodic signals, and finite N, a very definite phase dependence can develop. This is most easily viewed by considering sinewaves embedded in noise, and will be included at the end of this chapter.

Downsampling the Analytic Signal

While on one hand, linear predictive coding can be viewed as a system identification algorithm, it is also essential to view it as a spectral flattening algorithm. The spectrum of the error signal that is produced at each stage of the lattice filter becomes successively more flat (or white) until the prediction error eventually becomes a white noise signal [19,36]. As such, the linear predictor tends to model any artifacts in the spectrum, attempting to produce a residual with a white spectrum.

The power spectrum of the discrete analytic signal, like its analog counterpart, is one-sided over the range - to + , and hence half the bandwidth of the original real signal. This implies that the analytic signal can be downsampled to half the original sampling rate of the real signal without any loss of information. In fact, this is vital to prevent the linear predictor from modeling any artifacts of the signal in the dead band, and to assure that any additive white noise appearing in the real signal will also be white noise in the complex signal [26]. Further, any inaccuracies in the generation of the analytic signal via the Hilbert transform process which may appear in the dead band will not be modeled. In general, linear prediction performs best when a signal is sampled at or very close to its Nyquist rate. As the sampling rate is increased, the numerical accuracy required in the computations increases, and the process tends to become more ill-conditioned [20].

A frame of N points of a real signal is then processed as (N/2) complex points, hence, the total computational load for the analytic signal is comparable to the computation required for the real signal.

Frequency Estimation of a Single Sinewave

The frequency and phase of a sinewave can be estimated by performing a linear prediction on the signal, and then computing the zeroes of the inverse LPC filter polynomial. In this section, we shall only consider what is termed "exact order" identification. A sinewave is basically a one-pole signal, that is, its spectrum contains a pole of zero bandwidth at the frequency of the sinewave; the conjugate of this pole is also present with a real signal. As such, it can be exactly modeled by using a real second order linear predictor. The zeroes of the predictor polynomial ((A(z)) can be computed using any standard polynomial rooting routine.

In the analytic case, a single complex parameter, a_1 or k_1, can be used for prediction, since the complex signal has only one pole and no conjugate. In this case, the predictor polynomial would be

$$A(z) = 1 - k_1 z^{-1} . \qquad (93)$$

The zero of this filter is k_1, and from Equation 60, is given by,

$$k_1 = R(0,1)/R^*(1,1) \quad . \tag{94}$$

This expression for k_1 can be simplified from Equation 57, noting that for $s(mT) = e^{j(w_1 mT + \emptyset)}$,

$$R(0,1) = \sum_{m=0}^{N-1} e^{j(w_1 mT + \emptyset)} \; e^{-j(w_1 (m-1)T + \emptyset)}$$

$$= e^{jw_1 T} \sum_{m=0}^{N-1} (1)$$

$$= N \, e^{jw_1 T} \quad . \tag{95}$$

Also,

$$R(1,1) = \sum_{m=0}^{N-1} e^{j(w_1 (m-1)T + \emptyset)} \; e^{-j(w_1 (m-1)T + \emptyset)}$$

$$= N \quad . \tag{96}$$

Thus, combining Equations 94, 95, and 96, we find that

$$k_1 = e^{jw_1 T} \quad . \tag{97}$$

The sinewave, or pole, is estimated exactly, independent of the data length, N, or sequence phase, $\emptyset$.

This rather remarkable result is perhaps the most attractive reason for using analytic signals in spectral estimation. It is an extreme example of what the phase invariance property can do for estimation problems. This is not the case with real signals, as has been well documented in the literature [10,37,38]. The combined effects of the

errors occurring with real signals are exhibited as line splitting and frequency bias. The root of the problem lies in the finite length of the data, and the position of the window with respect to the signal. While these errors may not seem significant for an isolated frame of data, the implications of this phenomena manifest themselves in an application such as speech coding, where continuous spectral estimates of the incoming signal are computed.

In Figures 16 and 17, the maximum entropy spectra of successive six point frames of a single sinewave, whose frequency is 1797 Hz, are computed using the Burg algorithm [31]. The sampling rate is 8 kHz, and zero mean Gaussian white noise is added such that the signal-to-noise ratio is 60 dB. Figure 16 contains a plot of the spectra computed from the LPC parameters of the real sinewave with a second order real lattice, while Figure 17 contains the analytic spectra computed using a comparable single order complex lattice, computed using three (complex) point frames. Even though the signal is periodic, the maximum entropy spectra tend to fluctuate in the real case, due to the phase dependence of the estimator. Voiced regions of speech exhibit short term periodicities that are similar in nature to this sinewave example [19,39-41]. Generally, over such a stationary section of speech, the spectrum of the LPC filter fluctuates in a similar manner, having the effect of unnaturally modulating a pole frequency and bandwidth which in actuality is fixed.

One typical remedy for this problem is to use a model order much higher than the order of the signal, and use a large number of data points, allowing the spectral estimator to offset these effects by "overfitting" the signal [26]. Another remedy is to employ the use of a time domain window over the signal before computing the autocorrelation values. Historically, there was significant motivation to use a time domain window based on previous experience with fast Fourier transforms [6,7]. A third remedy is to average the frequency estimate over many frames of data [8].

Frequency Estimates of Multiple Sinewaves

While the single sinewave is a special case for the analytic signal, the resolution achieved by the analytic signal exceeds that of the real signal even for the case of multiple sinewaves. In this section, the performance for a signal consisting of two sinewaves embedded in white Gaussian noise will be investigated. There are several measures which can be used to demonstrate this improved resolution. By treating the frequency estimates as a random variable, the performance can be compared by examining the dependence of the variances of these random variables on the frame length and signal-to-noise ratio [42].

Let the real signal to be estimated, s(nT), be defined as the sum of two cosinewaves plus additive noise,

$$s(nT) = \cos(w_1 nT + \emptyset_1) + \cos(w_2 nT + \emptyset_2) + Gv(nT) \quad . \quad (98)$$

The sequence v(nT) represents additive, zero mean, white Gaussian noise whose variance is unity. The signal-to-noise ratio is given by

$$SNR = 1/(G^2) \ , \tag{99}$$

assuming independent sinewaves. Since this is actually a two-pole signal, a fourth order real estimator and a second order complex estimator will be used. Once again, the zeroes of the LPC polynomial will be computed and the frequency estimates will be taken as the values of the zero frequencies that lie in the range $0 \leq f \leq (1/2T)$. In Figure 18, a histogram of these estimates obtained using the Burg algorithm is plotted for 100 successive frames of the real signal, while in Figure 19, a histogram is plotted for the analytic signal. The frequencies of the sinewaves are 367 Hz and 859 Hz, while the phases are all set to zero. The frame length is 20 points for the real signal, 10 complex points for the analytic signal. The signal-to-noise ratio is 40 dB. The real estimator produces estimates which have a wider distribution, indicating a larger variance. Observe the rather large probability of a "miss" with the real estimator, that is, a frequency estimate of 0 Hz.

Since these estimators are generally unbiased [26], the variance of the frequency estimate of the lower frequency sinewave can be approximated by

$$var(\tilde{f}) = \Sigma \ (\tilde{f} - f_1)^2 \ , \tag{100}$$

where f_1 = 367 Hz. In Figures 20-22, the performance of the real LPC analyzer is compared to the performance of the analytic LPC analyzer. The logarithm of the variance around f_1 is plotted versus the frame length for signal-to-noise ratios of 20 dB, 40 dB, and 60 dB, respectively. This variance is computed by finding the frequency of the zero of the LPC estimator which is closest to f_1. In low signal-to-noise environments, the LPC estimator tends to miss this frequency entirely, i.e., detect only one of the two sinewaves. In this case, the estimator usually produces a zero at DC for the lower frequency. Thus, in practice, there is an upper limit on the variance, in which case the LPC estimator misses the frequency most of the time. The numerical value of this upper limit for the examples presented here is approximately 5.0 dB. When the variance is at or above this value, the LPC estimator is missing the frequency most of the time.

In Figure 20, for a low signal-to-noise ratio, it can be observed that the real estimator does not significantly improve with an increased frame length, while the analytic estimator does indeed improve. The variance of a robust estimator should decrease asymptotically with an increasing frame length [43], as a longer frame length allows the additive noise that appears in the frame to be more uncorrelated with the signal. At the higher signal-to-noise ratio of 60 dB in Figure 22, there is a decrease in the variance in both estimators as the frame length increases.

In Figures 23-25, the variance is plotted versus signal-to-noise ratio for frame lengths of 20, 40, and 80 points, respectively. Observe the threshold effect that this algorithm experiences. In Figure 23, for instance, this threshold was found to be 45 dB for the real signal, and 20 dB for the analytic signal. Here, the effect of the phase invariance is seen to drastically improve the performance of the estimator at low signal-to-noise ratios.

In Figures 26-28, the variance of the estimate of f_1 is again plotted versus the phase $\emptyset_1$, while $\emptyset_2$ is zero. A frame length of 20 points is used, and the signal-to-noise ratio varies from 20 to 60 dB. The performance is relatively uniform, as the estimators are relatively insensitive to phase differences between the two sinewaves.

To ensure that these results are not just a function of the estimator, this sequence of plots is repeated for the autocorrelation method of Equation 64. In this case, the sinewaves are more closely spaced, at frequencies of 414 Hz and 671 Hz. Figures 29-31 contain comparative plots of the variance versus frame length for both real and analytic estimators, parallel to Figures 20-22. It should be emphasized that the improvement in performance gained by using an analytic signal is really independent of the particular algorithm; it is more a byproduct of the finite data length constraints.

Another reason for the superior performance of the analytic signal, as observed by Jackson [44], is related to

the Hilbert transform process. A sinewave goes through various zero crossings, which in the presence of additive noise, provide the correlation function with little information, since the "localized" signal-to-noise ratio at a zero crossing is very small. The analytic signal, however, being a complex exponential, always has a magnitude of one. Thus, by virtue of the imaginary-part of the analytic signal being the quadrature of its real part, the amplitude of the signal always manages to stay well above the noise. This same concept has been used in applications ranging from antenna arrays to directional sensor arrays, where spatially diverse receivers are arranged to always detect a strong component of the signal.

CHAPTER VI

LEAST SQUARES MULTI-PULSE LINEAR PREDICTIVE CODING

Introduction

A major source of error in frame-based spectral estimators as applied to speech signals typically arises when the signal does not quite correspond to the model that is assumed for the signal. An accurate model for the production of voiced sounds [19,39-41] involves driving an all-pole filter with a pulse-like glottal waveform whose spectrum can be represented as an all-zero function. Typical LPC analysis algorithms have the capability of modeling not only the all-pole nature of the vocal tract, but also some of the glottal waveform [26], leaving a prediction residual which is pulse-like in nature. By driving this LPC filter with a train of ideal impulses spaced at a pitch period that is approximately the same as the pitch period found in the residual, natural sounding voiced speech can be generated. Alternately, by driving this filter with a white noise source, unvoiced speech can be produced.

The LPC analysis normally performed, however, assumes that the driving function of the vocal tract model was actually a white noise signal, uncorrelated with the speech signal. This inaccuracy in the model leads to errors in the estimated parameters. One common complaint of LPC estimation is that the pole bandwidths produced by the LPC all-pole

model are consistently too narrow, or overspecified, during voiced segments of speech [45,46]. Unvoiced utterances are usually modeled very well by the noise-excited LPC filter. One measure of this is the fact that the higher order reflection coefficients rapidly approach zero during unvoiced passages, allowing a reduction of the model order without a perceptible degradation in the quality of the synthetic speech [19,21]. Another measure is the structure of the residual itself. Voiced utterances, and other sounds that cannot be classified as either voiced or unvoiced, generally result in a much more complicated residual containing a very detailed harmonic structure. The residual associated with an unvoiced utterance, on the other hand, resembles a white noise sequence. In this chapter, a joint process estimation algorithm is presented which attempts to simultaneously estimate the smooth spectral structure of the speech signal and its pulse-like driving function. Atal [47] and Lee [48] have considered variations of this problem, basically incorporating *a priori* knowledge of the driving function obtained from alternate estimation procedures.

Least Squares Identification of Speech Using a Non-Ideal Driving Function

Given the very basic speech production model of voiced sounds as a linear filter with some quasi-periodic input sequence, the system can be represented by the following difference equation:

$$\tilde{s}(n) = \sum_{k=1}^{p} a_k s(n-k) + \sum_{j=1}^{q} b_j u_j(n), \qquad (101)$$

where $-p \leq n \leq N-1$. Here, the sampling rate, T, will be taken as unity, and the frame length is N+p points. Also, to guarantee that a unique solution for the parameters exists, $(p+q) < (N+p)$. The speech signal, $\tilde{s}(n)$, is thus the output of an all-pole system (or linear predictor) whose input is a sum of q arbitrary sequences. The sequences $u_j(n)$ can represent any type of function from an ideal impulse to a white noise sequence. It is assumed here that the input speech signal, s(n), and the driving functions, $u_j(n)$, are zero mean. If in actuality they are not, then the formulations presented below can be easily modified to include a constant term to account for any fixed bias that may appear due to the presence of a non-zero mean input.

Let the parameter vectors ($\bar{a}$ and $\bar{b}$), the signal matrix ($\underline{U}$), and the signal vector ($\bar{Z}$) take on the following form:

$$\bar{a} = [\, a_1 \; a_2 \; \ldots \; a_p \,]^T, \quad \bar{b} = [\, b_1 \; b_2 \; \ldots \; b_q \,]^T, \qquad (102)$$

$$\underline{U} = [\, \underline{U}_s \mid \underline{U}_u \,], \qquad (103)$$

where

$$\underline{U}_s = \begin{bmatrix} s(-1) & s(-2) & \ldots & s(-p) \\ s(0) & s(-1) & \ldots & s(-p+1) \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ s(N-2) & s(N-3) & \ldots & s(N-p-1) \end{bmatrix}, \qquad (104)$$

$$\underline{U}_u = \begin{bmatrix} u_1(0) & u_2(0) & \dots & u_q(0) \\ u_1(1) & u_2(1) & \dots & u_q(1) \\ \vdots & \vdots & \vdots & \vdots \\ u_1(N-1) & u_2(N-1) & \dots & u_q(N-1) \end{bmatrix}, \tag{105}$$

and

$$\bar{Z} = [\, s(0)\ s(1)\ \dots\ s(N-1)\,]^T. \tag{106}$$

Note that $\underline{U}$ is a N x (p+q) matrix, while $\bar{Z}$ is a N-dimensional vector. Also, in the following discussion, a frame of N+p samples of the speech signal is being considered as the measurements, or frame, of the signal.

The least squares estimates of the parameter vectors $\bar{a}$ and $\bar{b}$ can be found [26] by noting that

$$\hat{\bar{Z}} = \underline{U}\,[\,\bar{a} \mid \bar{b}\,]^T . \tag{107}$$

Let the error in prediction be defined as

$$E = \sum_{n=0}^{N-1} [\, s(n) - \tilde{s}(n)\,]^2 , \tag{108}$$

$$= [\bar{Z} - \hat{\bar{Z}}\,]^T [\bar{Z} - \hat{\bar{Z}}] . \tag{109}$$

Upon taking the derivative of E with respect to each parameter, or, performing the entire operation in vector form, by differentiating E with respect to the vector $[\,\bar{a} \mid \bar{b}\,]$, (and setting the derivatives equal to zero) the least squares estimates [26] are found to be

$$\begin{bmatrix} \bar{a} \\ \hline \bar{b} \end{bmatrix} = (\underline{U}^T\underline{U})^{-1}\underline{U}^T\,\bar{Z} . \tag{110}$$

The product $\underline{U}^T\underline{U}$, a real symmetric matrix, can be partitioned to give

$$\underline{U}^T\underline{U} = \left[\begin{array}{c|c} \underline{R}_s & \underline{D} \\ \hline \underline{D}^T & \underline{R}_u \end{array}\right] . \qquad (111)$$

Here $\underline{R}_s$ is a pxp correlation matrix of the signal, whose elements are given by

$$\begin{aligned} \underline{R}_s &= \underline{U}_s^T \underline{U}_s \\ &= [s_{ij}] , \end{aligned} \qquad (112)$$

where

$$s_{ij} = \sum_{n=0}^{N-1} s(n-i)s(n-j) ,$$

and $1 \leq i \leq p$ and $1 \leq j \leq p$, while $\underline{R}_u$ is a qxq correlation matrix of the input, that is,

$$\begin{aligned} \underline{R}_u &= \underline{U}_u^T \underline{U}_u \\ &= [r_{ij}] , \end{aligned} \qquad (113)$$

where

$$r_{ij} = \sum_{n=0}^{N-1} u_i(n)u_j(n) ,$$

and $1 \leq i \leq q$ and $1 \leq j \leq q$.

The matrix $\underline{D}$, most easily viewed as a matrix containing information about the crosscorrelation of the input with the delayed output, is given by

$$\begin{aligned} \underline{D} &= \underline{U}_s^T \underline{U}_u \\ &= [d_{ij}] , \end{aligned} \qquad (114)$$

where

$$d_{ij} = \sum_{n=0}^{N-1} u_j(n)s(n-i) \ , \tag{115}$$

and $1 \leq i \leq p$ and $1 \leq j \leq q$. The vector $\underline{U}^T\bar{Z}$ can be decomposed into two parts, that is,

$$\underline{U}^T\bar{Z} = [\ \bar{r} \ | \ \bar{d}\ ]^T, \tag{116}$$

where

$$\bar{r} = [\ \sum_{n=0}^{N-1} s(n)s(n-1) \ \ldots \ \sum_{n=0}^{N-1} s(n)s(n-p)\ ] \ , \tag{117}$$

and

$$\bar{d} = [\ \sum_{n=0}^{N-1} s(n)u_1(n) \ \ldots \ \sum_{n=0}^{N-1} s(n)u_q(n)\ ] \ . \tag{118}$$

The vector $\bar{d}$ contains information about the crosscorrelation of the input with the output, while the vector $\bar{r}$ contains the p correlation values of the signal s(n).

Thus, by making the appropriate substitutions from above, Equation 110 can be rewritten as:

$$\left[\begin{array}{c|c} \underline{R}_s & \underline{D} \\ \hline \underline{D}^T & \underline{R}_u \end{array}\right] \left[\begin{array}{c} \bar{a} \\ \hline \bar{b} \end{array}\right] = \left[\begin{array}{c} \bar{r} \\ \hline \bar{d} \end{array}\right] \ , \tag{119}$$

or

$$\underline{R}_s\bar{a} + \underline{D}\bar{b} = \bar{r} \ , \tag{120}$$

and

$$\underline{D}^T\bar{a} + \underline{R}_u\bar{b} = \bar{d} \ . \tag{121}$$

Assuming $\underline{R}_u$ is not singular, this partitioned system can be solved for the least squares estimate of the parameter vector $\bar{a}$. Rewriting Equation 121 and solving for $\bar{b}$, we see that

$$\underline{R}_u \bar{b} = \bar{d} - \underline{D}^T \bar{a}$$

and

$$\bar{b} = \underline{R}_u^{-1} [\bar{d} - \underline{D}^T \bar{a}] . \tag{122}$$

Substituting Equation 122 into Equation 120, we find that

$$\underline{R}_s \bar{a} + \underline{D} \underline{R}_u^{-1} [\bar{d} - \underline{D}^T \bar{a}] = \bar{r} ,$$

or,

$$[\underline{R}_s - \underline{D} \underline{R}_u^{-1} \underline{D}^T] \bar{a} = \bar{r} - \underline{D} \underline{R}_u^{-1} \bar{d} ;$$

this gives

$$\bar{a} = [\underline{R}_s - \underline{D} \underline{R}_u^{-1} \underline{D}^T]^{-1} [\bar{r} - \underline{D} \underline{R}_u^{-1} \bar{d}] . \tag{123}$$

Observe that both inverses which appear in Equation 123 can be singular. Also, more importantly, the linear predictor portion of Equation 101 is not guaranteed to be stable.

From $\bar{a}$, the vector $\bar{b}$ can be obtained by substituting Equation 123 into Equation 122, that is,

$$\bar{b} = \underline{R}_u^{-1} \{ \bar{d} - \underline{D}^T [\underline{R}_s - \underline{D} \underline{R}_u^{-1} \underline{D}^T]^{-1} [\bar{r} - \underline{D} \underline{R}_u^{-1} \bar{d}] \} . \tag{124}$$

The vector $\bar{b}$ is a weighting function in that it assigns weights to the various components of the driving function in Equation 101.

If both the $\underline{D} \underline{R}_u^{-1} \underline{D}^T$ and $\underline{D} \underline{R}_u^{-1} \bar{d}$ terms were zero, the predictor parameters, $\bar{a}$, obtained from Equation 123 would be

equivalent to the "normal" least squares predictor coefficients, as Equation 123 reduces to the equivalent of Equation 60. The normal least squares solution assumes an input to the system which is white noise, or similarly, a prediction error which is orthogonal to the signal. If $q = 1$, $\underline{R}_u^{-1}$ becomes a scalar equal to the reciprocal of the signal power in the input u(n). The $\underline{D}\ \underline{R}_u^{-1}\underline{D}^T$ and $\underline{D}\ \underline{R}_u^{-1}\underline{\bar{d}}$ terms will be zero if the input signal, u(n), is orthogonal to the speech signal, s(n). A simple case of this is if the input signal is white noise.

Typically this is not the case with speech signals, and the parameters determined under this assumption are not the minimum mean-square error parameters. Only in unvoiced regions of speech is this assumption valid. On the other hand, given access to just the output speech signal, and no information about the input excitation signal, these optimal predictor coefficients cannot be determined in a straightforward manner. Over a finite length of data, it is difficult to satisfy the requirement that the input signal is orthogonal to the output signal. For a speech signal, this results from the fact that the actual input signal to the vocal tract is not a white noise signal. For other estimation problems, such as signals embedded in additive noise, the correlation of the noise with the signal cannot be exactly zero over a finite length of data. As this correlation fluctuates from frame to frame, so will the parameters. Hence, it is not surprising that in practice

the estimates obtained show unusual behavior for even simple signals in additive noise.

The parameters computed from Equation 123 differ from the true least mean-square error parameters in that any information about the driving function has been removed from the correlation computation. The normal least squares approach produces parameters which will fluctuate as the signal within the analysis frame changes. In speech analysis, for instance, this translates to a certain dependence of the parameters upon pitch. These parameters will change as the number or shape of the glottal pulses within the frame changes, even though the vocal tract itself may be stationary. This produces an error which is pitch dependent, since for very low pitches, the magnitude of the $\underline{D}\ \underline{R}_u^{-1}\underline{D}^T$ term will be small compared to higher pitched cases. For applications such as speech coding and speech recognition, it is useful to have a set of predictor parameters which are decoupled from the input driving function.

A Sub-Optimal Algorithm for Joint Process Estimation

A good approximation to the speech production model of voiced sounds is a time-varying linear filter driven by a pulse-train, as stated above. In low bit rate coding of speech, this pulse train is assumed to consist of ideal impulses spaced a fixed distance apart in time by the pitch period. A very general representation of this type of

structure can be defined as

$$\tilde{s}(n) = \sum_{k=1}^{p} a_k s(n-k) + \sum_{j=1}^{q} b_j \mathrm{Sinc}(\pi(n-t_j)) \ . \qquad (125)$$

The merit in this representation is that the bandlimited excitation pulses are not constrained to occur at sample instants. If a bandlimited pulse does occur at a sample instant, its response, being of the form $\mathrm{Sinc}(\pi(n-t))$, has only one non-zero value at $n = t$. If this pulse occurs in between two sample points, it has an infinite number of non-zero values, and has the shape of the Sinc function.

Ideally, Equation 125 could be solved to find the optimal predictor parameters, pulse weights, and pulse positions. However, this is a nonlinear optimization problem which is computationally intensive, with no guarantee of a unique solution. If these pulses are constrained to exist only at sample points, Equation 125 can be rewritten as

$$\tilde{s}(n) = \sum_{k=1}^{p} a_k s(n-k) + \sum_{j=1}^{q} b_j d(n-t_j) \ . \qquad (126)$$

Observe also that the excitation pulses, being impulses, are orthogonal to one another. These pulses now represent the driving function of Equation 101, where $u_j(n) = d(n-t_j)$.

In this case, the correlation matrix, $\underline{R}_u$, in Equation 113, simplifies to the identity matrix, $\underline{I}$, since

the only non-zero terms are unity valued diagonal elements. Equation 123 can now be simplified to

$$\bar{a} = [\underline{R}_s - \underline{D}\,\underline{D}^T]^{-1}\,[\bar{r} - \underline{D}\bar{d}] \; . \qquad (127)$$

Observe that the optimal predictor parameters are computed by removing the contributions to the autocorrelation function due to samples located at the pulse positions. It is equivalent to computing the correlation function only during instances when the glottis is closed, and there is no input to the vocal tract. Still, however, the least squares optimization problem remains a nonlinear one, because the pulse locations are not known *a priori*. In fact, for this reason, in speech coding, the problem has been conveniently dichotomized into two separate problems: pitch detection and coefficient computation.

The sound pressure wave which is recorded through a microphone as a speech signal in reality is an airflow which undergoes turbulence as it travels through the vocal tract region [49]. While the model of voiced speech production as an all-pole system with a periodic or quasi-periodic input approximates the vocal tract as a concatenation of lossless tubes [19], the vocal tract is more accurately modeled as a concatenation of lossy tubes, due to frictional forces acting along the vocal tract walls. The airflow through the vocal tract undergoes turbulence due in part to the velocity gradient created by these frictional forces in the vocal tract walls. It is common for turbulent systems to exhibit

a phenomena called strange attraction, where the system will exist in one metastable state for one period of time, then suddenly jump to another metastable state for another time interval, and continue doing so in an almost random fashion [50]. In some systems, like speech production, this exhibits itself as a process known as period doubling. The system begins with a certain periodic disturbance, and then slips into a state where the time scale of these fluctuations increases, giving rise to subharmonic disturbances.

In the speech signal, this phenomena tends to occur at the beginning or end of a voiced segment. At the onset of a voiced segment, where the speech waveform is changing from either silence or unvoiced to voiced, the pulses are irregularly spaced but converging towards periodicity. In Figure 32, a typical speech signal is shown with its residual below. Note that the residual "excitation" pulses can mark for us the pitch periods. It is not unusual for the perceived pitch period to change dramatically throughout a voiced segment. Extreme cases include words at the end of a sentence, where certain articulation patterns tend to produce a pitch period which can drastically increase, as shown in Figure 33 (Again the residual is given below the speech waveform.).

Another common phenomena peculiar to the speech signal is termed secondary excitation. The residual associated with a voiced segment exhibits very sharp pulses that correspond

to points in the original waveform where the signal is not linearly predictable. During such voiced segments, it is common for a secondary pulse to occur between two pitch pulses. After the main excitation that occurs at the glottal closure, there is a secondary excitation that can occur not only at glottal opening and during the open phase, but also after closure [18]. In Figure 34, a typical speech waveform and residual (given below) in which secondary excitation occurs is shown. The pitch detection process is particularly sensitive to these phenomena, and a large class of pitch detection schemes fail to properly track pitch through these kinds of waveforms [22]. In all of these cases, it becomes evident that it is desirable to discard any notion of pitch or periodicity in favor of a more flexible excitation function consisting of multiple pulses.

The model in Equation 126 can provide this multiple pulse excitation as long as the pulse locations can be determined. To avoid the nonlinear optimization problem, a suboptimal approach can be employed in which the pulses are located in a recursive fashion. This is most straightforwardly performed by searching all possible pulse locations for the pulse position and amplitude that minimize the error between the model and the actual speech signal. After this pulse is located, its position remains fixed, and all other positions are searched for the next pulse, which is again determined on a squared error basis. This process is repeated until all pulses in the frame are located.

This pulse location process can be constrained in several different manners. A little thought dictates that some notion of pitch should be maintained, that is, the pulses should be located somewhat synchronous to a pitch period. If the pulses are constrained to minimize the error over an entire segment of speech, they all might be allocated in one particular area, and not distributed throughout the segment. For this reason, a pragmatic tradeoff is to allocate a fixed number of pulses per frame, the same frame used to compute the LPC coefficients.

The number of pulses located in this frame should ideally depend upon the pitch period. Since the highest pitch normally found in human speech is approximately 500 Hz, it is possible to have pitch pulses spaced at a minimum of 2 ms intervals. Using a sampling rate of 8 kHz, this is equivalent to a pitch distance of 16 samples. The spectrum of the speech signal is slowly varying with time, such that over a frame of 160 samples (20 ms) the signal appears pseudo-stationary. Thus, by locating 16 pulses every 160 samples, the multiple pulse excitation should produce at least two pulses for every pitch period of the highest possible pitch frequency. In this manner, any type of pitched excitation should be reasonably modeled, yet no explicit pitch estimate is required.

The least squares multi-pulse linear predictive coding (LSMPLPC) algorithm, where Q pulses are located in every N point frame of a speech signal, can be summarized as follows

(after computing $\underline{R}_s$ and $\bar{r}$ from the speech data):

1. Set q = 0, and compute $\bar{a}$ from Equation 127.
2. Let q = q + 1, t_q = 0, and compute $\underline{D}$, $\underline{R}_u$, and $\bar{d}$.
3. Compute $\bar{a}$ from Equation 123 and
   $\bar{b}$ from Equation 124.
4. Synthesize $\tilde{s}(n)$ and compute the mean-squared
   error over the frame, that is,

$$E(t_q,q) = \sum_{n=0}^{N-1} (s(n) - \tilde{s}(n))^2 .$$

5. Let $t_q = t_q + 1$, if$(t_q < N)$,
   compute $\underline{D}$, $\underline{R}_u$, and $\bar{d}$, and go to step 3.
6. Choose the position, k, for which $E(t_q,q)$ is a
   minimum, and let t_q = k.
7. Retaining the q excitation positions,
   if (q < Q), go to step 2.
8. Using the final parameter vectors, $\bar{a}$ and $\bar{b}$,
   along with the Q excitation positions, $\{t_j\}$,
   synthesize the speech estimates, $\tilde{s}(n)$,
   for n = 0, ..., N-1 from Equation 126, and
   repeat for the next N samples of the signal.

The pulse locations are determined by selecting the position that minimizes the error between the model and the signal. A pulse is then installed at that position, and the search

procedure for the next pulse is initiated. Throughout the process, the predictor parameters and pulse amplitudes are continually optimized. While this algorithm is only suboptimal, it presents a systematic technique for generating the driving function for this particular model. In this case, we have used ideal impulse functions as our excitation format.

This algorithm is quite computationally intensive, generally running approximately 10,000 times real-time on a general purpose machine. Experimental results indicate that the interaction between the ideal impulse functions and the predictor parameters is relatively small. The parameter optimization technique does not significantly alter the values of the LPC coefficients, thereby producing a pulse train very close in structure to that which would have been generated assuming no interaction between the driving function and the LPC estimation.

In Figure 35, a sample speech segment and its prediction residual (below) are shown for a female speaker whose pitch period is 3.6 ms. A relatively high-pitched speaker is chosen in order to maximize the effect of the non-ideal driving function. In Figure 36, the impulse responses of both the LPC and LSMPLPC filters are shown. In Figure 37, the spectrum implied by the predictor parameters computed from the above algorithm is compared to the spectrum produced by the parameters computed at step 1 (the conventional LPC covariance method parameters) of the

algorithm. In Figures 38 thru 40, the case of a male speaker whose pitch period is 5.1 ms is presented. In both the female and male examples, we have used N = 160, (1/T) = 8 kHz, p = 16, and Q = 16. The LPC spectra and the LSMPLPC sptectra are very close. In fact, the actual differences between the LPC parameters and the LSMPLPC impulse response are extremely small. This implies that the pulse locations will not be greatly affected by the differences in parameters. The slight differences that exist in these spectra, as well as the slight differences in the corresponding LPC parameters, certainly are not significant when normal parameter coding techniques are applied. This is not to say that there is no interaction between the speech excitation signal and its LPC spectrum, but rather that the ideal impulse model is not accurate enough to remove these effects.

Of course, as more pulses are added, the effects of the pulsed excitation on the parameter calculation will increase. However, the discussion presented above is actually an introduction to the speech coding application discussed in detail in Chapter VII. In order to achieve a reasonable data compression of the speech signal, the number of pulses must be kept to the minimum required for an acceptable synthetic speech quality. After discovering that the interaction between coefficients and pulses is small when only a few excitation pulses are used, this algorithm can be extensively simplified.

Multi-Pulse Linear Predictive Coding

Atal introduced multi-pulse linear predictive coding (MPLPC) [18] as an analysis-by-synthesis technique to construct the multiple pulse excitation function. As such, it is a suboptimal approach to a least squares optimization problem. Since it requires no voiced/unvoiced decision, or pitch detection, it is an attractive medium rate speech coding algorithm. However, the algorithm as detailed above is not quite complete. A very essential part of this algorithm involves the use of perceptual weighting [18].

All data compression algorithms for speech will generate some errors. In order to produce the most natural-sounding synthetic speech, it is desirable to force these errors to occur in regions of the spectrum that are tolerable to the human ear, specifically, in the vicinity of the formant frequencies. The mean-square error criterion typically employed in speech coding systems is not the best metric for human perception, but a convenient one for computational reasons. The performance of MPLPC is disappointing without some perceptual weighting, because a large number of pulses are required to produce good quality speech. By de-emphasizing the error spectrum in the vicinity of a formant frequency, a subjectively meaningful measure of the difference between the synthetic speech and the original speech can be produced [51-55]. Since the pulse location process is essentially an analysis-by-synthesis technique, if it is performed on the perceptually weighted speech, then

the errors produced by the pulses will be masked to the extent that our model of masking is correct. The errors produced by the inaccuracies of the pulsed excitation tend to appear as noise that is flat in the frequency domain. By de-emphasizing the speech energy in the regions around the formants during the pulse location process, the effective signal-to-noise ratio around a formant will be lower than that in a region between two formants. When this noise weighting process is inverted at the synthesizer, the majority of the spectral errors generated by this algorithm will fall around a formant, and be subjectively less perceptible than if they were distributed equally throughout the spectrum.

Atal [18] has shown that this can be accomplished by processing the speech through a noise-weighting network whose transfer function is defined as:

$$H(z) = \left[1 - \sum_{k=1}^{p} a_k z^{-k} \right] / \left[1 - \sum_{k=1}^{p} a_k (rz)^{-k} \right] . \quad (128)$$

The noise-weighting constant, r, ranges from zero to one. In the extreme, where r = 0, the output of the network is the prediction residual. When r = 1, H(z) = 1, and the output is the original speech. This network produces broadened-bandwidth speech, that is, speech where the bandwidth around formants have been widened [20] by approximately (1/ T)ln(1/r) Hz. This broadened-bandwidth speech, then, is somewhere between the residual and the

original speech in its character. Speech with a high Q spectrum (i.e., narrow bandwidth about a formant), when passed through this network, losses its resonant structure in favor of the almost white nature of the prediction residual. Speech with a low Q spectrum passes through the network virtually unchanged in appearance, as the change in the formant bandwidths produced by this network is not noticeable.

The advantage of this formulation is that it can be easily inverted at the synthesizer, and requires no extra computation. This is not the case in the LSMPLPC algorithm presented above, because the parameters are computed jointly with the pulses. The computational savings is perhaps the most important reason why it is desirable to decouple the parameter calculation from the pulse location process. Atal [18] found from subjective listening tests that $r = 0.85$, a bandwidth broadening of 413 Hz at $1/T = 8$ kHz, was optimal for MPLPC. Use of this noise-weighting factor allows the location of 8 pulses every 10 ms to accurately model the speech signal, as only minimal improvements are obtained by adding additional pulses [18].

It is important also to recognize the computational implications this perceptual weighting has upon MPLPC. Because the parameters are now decoupled from the predictor coefficients, and the formant bandwidths widened to a minimum of 413 Hz, the computations required to find the pulses can be vastly simplified through the use of a

crosscorrelation function (see Chapter VII). Perceptual weighting has a very definite impact upon the effectiveness of this procedure, as seen in the next chapter, where MPLPC using analytic signals is considered.

CHAPTER VII

ANALYTIC MULTI-PULSE LINEAR PREDICTIVE CODING

Introduction

Analytic multi-pulse linear predictive coding (AMPLPC) is conceptually identical to its real signal equivalent, the only major difference being the nature of the parameters themselves. The analytic signal based system uses complex parameters, that require somewhat more sophisticated two-dimensional coding algorithms, while MPLPC uses real parameters that employ well understood coding techniques. In this chapter, the details of an AMPLPC system are presented. This encoder compresses speech sampled at 8 kHz (conventional 64 kbits/s companded PCM) to a rate of 9.6 kbits/s.

Pulse Location for a Complex Signal

The pulse location process can be greatly simplified through the use of a crosscorrelation function. Let the analytic speech sequence, or any complex sequence, be denoted by $s_a(n)$. Since the LPC analyzer is equivalent to filtering the original speech with an all-zero FIR filter, the original speech can be regenerated by passing the prediction residual through an all-pole IIR filter that is the inverse of the all-zero FIR filter. Similarly, the synthetic speech is generated by applying the multi-pulse driving function to the input of the same IIR filter. When

the LPC synthesizer with zero initial conditions is excited with an ideal impulse of unit amplitude, it will produce an output signal defined as the impulse response, h(n). From Equation 69, if we drop the stage i notation, cast E(z) as the input and S(z) as the output of the synthesizer, then we see that

$$h(n) = Z^{-1}\{1/A(z)\} . \tag{129}$$

Since h(n) is the output of a stable all-pole IIR filter, h(n) is infinite in duration, and asymptotically decays to zero when all the poles of this filter are strictly inside the unit circle in the z-plane [19].

The optimal place to locate a pulse would be the position that produces the minimum error between the actual speech, $s_a(n)$, and the synthetic speech produced by exciting the synthesizer with this pulse. The pulse amplitude, defined as G, must also be determined on a minimum mean-square error basis. The mean-square error is the difference between the speech and this scaled impulse response, that is,

$$E = \sum_n e_n e_n^* ,$$

$$= \sum_n [s_a(n) - G\, h(n-m)][s_a^*(n) - G^* h^*(n-m)] , \tag{130}$$

$$= \sum_n [\, s_a(n)s_a^*(n) - G^* s_a(n)h^*(n-m)$$

$$- G\, s_a^*(n)h(n-m) + GG^* h(n-m)h^*(n-m)\,] . \tag{131}$$

Here, m denotes the position of the pulse in time. The limits on the summation are not explicitly stated for the following reason. In theory, these limits should extend for all time, while in practice, they usually extend over a single frame of speech. In fact, if assumptions about the length of the impulse response are made, the range of the summation can be even smaller, as any errors due to truncation effects will be minimal.

The optimal amplitude of the pulse can be easily determined by minimizing Equation 131 with respect to both the real and imaginary parts of G. Let G be defined as

$$G = A + jB \ . \tag{132}$$

Setting the derivative of Equation 131 with respect to A equal to zero gives

$$\sum_n [\ -s_a(n)h^*(n-m) - s_a^*(n)h(n-m) + 2A\ h(n-m)h^*(n-m)\ ] = 0 \ ,$$

or alternately,

$$\sum_n [\ -2\ \mathrm{Re}\{s_a(n)h^*(n-m)\} + 2A\ h(n-m)h^*(n-m)\ ] = 0 \ . \tag{133}$$

Solving for A, we obtain

$$A = \mathrm{Re}\{\ \sum_n [s_a(n)h^*(n-m)]\}/\{\ \sum_n [h(n-m)h^*(n-m)]\} \ . \tag{134}$$

Setting the derivative of Equation 131 with respect to B equal to zero gives

$$\sum_n [j\ s_a(n)h^*(n-m) - j\ s_a^*(n)h(n-m) + 2B\ h(n-m)h^*(n-m)\ ] = 0 \ ,$$

or,

$$\sum_n [\ -2\ \mathrm{Im}\{s_a(n)h^*(n-m)\} + 2B\ h(n-m)h^*(n-m)\ ] = 0\ .$$

Solving this equation for B gives

$$B = \mathrm{Im}\{\sum_n [s_a(n)h^*(n-m)]\}/\{\sum_n [h(n-m)h^*(n-m)]\}\ . \quad (135)$$

Combining Equations 134 and 135, we observe that

$$G = \{\sum_n [s_a(n)h^*(n-m)]\}/\{\sum_n [h(n-m)h^*(n-m)]\}\ . \quad (136)$$

The optimal pulse amplitude is identified as just the crosscorrelation of the impulse response with the signal at time instant m, divided by the power in the impulse response.

The mean-square error expression of Equation 131 can be simplified by substituting G from Equation 136, that is,

$$\begin{aligned} E = \sum_n [\ s_a(n)s_a^*(n)\ ] \\ - \{[\sum_n [s_a^*(n)h(n-m)]]/[\sum_n [h(n-m)h^*(n-m)]]\}\ x \\ \sum_n [\ s_a(n)h^*(n-m)\ ] \\ - \{[\sum_n [s_a(n)h^*(n-m)]]/[\sum_n [h(n-m)h^*(n-m)]]\}\ x \\ \sum_n [\ s_a^*(n)h(n-m)\ ] \\ + \{[\sum_n [s_a(n)h^*(n-m)]]/[\sum_n [h(n-m)h^*(n-m)]]\}\ x \\ \{\sum_n [s_a^*(n)h(n-m)]/[\sum_n [h(n-m)h^*(n-m)]]\}\ x \\ \sum_n [\ h(n-m)h^*(n-m)\ ]\ , \end{aligned}$$

$$= \sum_n s_a(n)s_a^*(n) - |\sum_n [s_a(n)h^*(n-m)]|^2/\{\sum_n [h(n-m)h^*(n-m)]\}. \quad (137)$$

Since this error is always a positive quantity, it will be minimum when the second term is maximum. The critical part of the second term is just the magnitude of the crosscorrelation function. Thus, the optimal pulse location is the point at which the crosscorrelation function is a maximum, and the optimal amplitude of this pulse is the value of this crosscorrelation function divided by the power in the impulse response, as seen from Equation 136.

Once the first pulse is located, its scaled impulse response can be subtracted from the original speech, and the procedure repeated in an iterative fashion for all the subsequent excitation pulses. A good approximation to this procedure is to merely subtract the autocorrelation function of the impulse response, h(n), from the crosscorrelation function. Thus, the crosscorrelation function need only be computed once, thereby reducing the computational requirements even further.

Perceptual weighting has a very important impact upon the pulse location procedure. If this algorithm is performed on the original speech, a large number of pulses will be required, and the required bit rate will once again be disappointing. Speech waveforms for both males and females during voiced intervals can exhibit poles whose bandwidths are very small. The impulse response, h(n), will

be very long in these cases, requiring the crosscorrelation function to be computed over an excessively long interval. Without a reliable crosscorrelation function, the pulse locations cannot be accurately determined.

It is important to find the pulse locations with resolution down to a sample time, as slight errors in the pulse locations can produce significant errors in the harmonics of the excitation signal. For instance, suppose the optimal pulse locations were equally spaced at a distance of 40 samples with an 8 kHz sampling rate. If this distance was computed to be 39 samples, the first harmonic of the excitation signal would be in error by 5 Hz, the second harmonic in error by 10 Hz, and so forth. Errors of this order are certainly perceptible [55], as the synthetic speech will become hoarse and scratchy if the errors occur in a random fashion. By increasing the bandwidths of the poles of the synthesis filter using perceptual weighting, the impulse response length is guaranteed to be very short, typically 2 ms long, allowing an accurate computation of the crosscorrelation function.

The broadened-bandwidth speech then is desirable not only for its auditory masking properties, but more importantly, for the fact that the pulse location process can be performed accurately with a minimal number of computations. This computational consideration makes the system very sensitive to noise-weighting. In the noise-weighted domain, the crosscorrelation function can be

accurately computed using a window on the order of 5 ms. Because of the restriction of a finite analysis interval, the analytic signal is again an attractive alternative. However, the difference here is that the time resolution that can be attained with the analytic signal is half that of the real signal, due to the downsampling process.

Complex Parameter Coding

While the LPC parameters can be computed in a number of ways, subjective listening tests performed during this study indicate that there is no perceptible difference in the synthetic speech when any of these techniques is used with MPLPC or AMPLPC. The lattice-based Burg algorithm of Chapter V is a natural choice to compute LPC parameters because it is robust and because it always generates a stable synthesis filter. Since the complex reflection coefficients are guaranteed to have a magnitude less than or equal to one, these coefficients can be treated as a two-dimensional random variable distributed within or on a unit circle. Reflection coefficients, or some transformation of reflection coefficients (such as log area ratios), generally are less sensitive to quantization errors than predictor parameters [36]. Gallagher [56], among others, has studied the properties of optimal multidimensional quantizers, specifically for problems dealing with the transmission of Fourier transform data. In this section, the problem of coding reflection coefficients

shall be considered.

In theory, the optimal way to code a complex reflection coefficient would be to design an optimal two-dimensional quantizer. Generally, this involves collecting statistics on the probability density function of the coefficient, and designing a quantizer which produces a minimum quantization error. Let the density function for a reflection coefficient, k, be defined as $p_k(u,v)$, where $k = u + jv$. If D bits are assigned to this coefficient, $L = 2^D$ possible values are allowed for k, each value denoted as k_j. The quantization error can be computed as

$$E = \sum_{j=1}^{L} \int_{S_j} p_k(u,v) |(k - k_j)|^2 \, du \, dv \, ,$$

where S_j denotes the quantization region for which all values of k in S_j are quantized to k_j. Optimization techniques [57] can yield the exact regions S_j for a given distribution, $p_k(u,v)$, and the values k_j. Alternately, the S_j can be approximately determined by designing the quantizer such that areas of highest probability are most accurately quantized while areas of lower probability are allowed greater quantization error. This procedure is similar to those implemented in [58-60] for a one-dimensional random variable. However, the drawback with these approaches is that they are not that simple to implement, requiring an exhaustive search procedure similar to that found in vector quantization encoders [60].

Further, they do not exploit any relationship between reflection coefficients and the speech signal.

One of the desirable properties of the lattice is that a measure of both the error in the estimation process and the order of the signal is obtained at each stage of the lattice [19]. The energy in the error at stage i in the lattice decreases by a factor of $1 - |k_i|^2$. If, for instance, the signal being analyzed is a harmonic process, it will at some point in the lattice produce a reflection coefficient equal (or in practice very close) to unity in magnitude. At this point, the analysis should be terminated since the error energy becomes very small. A single sinewave analyzed using the analytic signal based lattice will produce a first coefficient, k_1, whose magnitude is unity.

Further, in speech analysis, typically anywhere from five to six poles are sufficient to model the smooth spectral structure of the signal, and any coefficients beyond this order are usually very small. When the signal is unvoiced, usually only two or three poles are required. Since a reflection coefficient in the lattice is a measure of the correlation between the backward and forward errors, when a reflection coefficient is small, it is not as sensitive to quantization errors as when it is large. Thus, since the magnitudes of the reflection coefficients decrease at each successive stage in the lattice, fewer bits can be allotted to the higher order coefficients. The quantizer

can also be structured to weight larger reflection coefficients more than smaller ones.

The first reflection coefficient in the lattice acts as an adaptive pre-emphasizer in that it attempts to flatten the spectrum of the signal. Typically, the distribution of this coefficient is skewed, as will be seen presently. During voiced segments, the value of k_1 from Equation 68 tends to have a value around 0.6 + 0.6j, implying a zero at 500 Hz in the spectrum. During unvoiced segments, this same coefficient tends to take a value around -0.3 - 0.3j. In order to distribute this coefficient more uniformly about the unit circle, and decrease the sensitivities of all the coefficients to quantization error, a fixed pre-emphasis characteristic is employed with the filter defined by

$$H_p(z) = 1 - (0.3 + 0.3j)z^{-1} . \qquad (138)$$

This pre-emphasis transformation is generally called "soft" pre-emphasis. During voiced segments, a "hard" pre-emphasis transformation of $1 - (0.6 + 0.6j)z^{-1}$ should be used. However, unvoiced signals require a different transformation, since their spectrum typically has a high-pass frequency response from 0 to 4 kHz. This is opposite to the low-pass frequency response of the voiced speech spectrum [20]. The pre-emphasis transformation of Equation 138 is a good compromise for both cases. Adaptive pre-emphasis can also be used, but transmission of an adaptive pre-emphasis constant is not efficient.

Pre-emphasis does not impair the performance of the system, and most importantly, allows the reflection coefficients to be efficiently encoded on a magnitude/phase basis.

The long-term statistics of these reflection coefficients have been compiled using a data base of speech which consists of Harvard phonetically balanced sentences [61] spoken by six male and six female speakers plus some conversational speech. The total data base is ten minutes in duration, and includes both speech low-pass filtered to a 3.3 kHz bandwidth, and telephone bandwidth speech which has a bandpass characteristic extending from 300 Hz to 3.3 kHz [62]. The statistics depend somewhat upon the input filtering conditions, so these two common filtering characteristics have been included in the data base.

While log area ratios have traditionally been shown to be the transformation of reflection coefficients least sensitive to quantization error [19], these are not desirable for complex coefficients. The transformation of a complex reflection coefficient to a log area ratio requires one complex division per coefficient, making it a computationally expensive transformation. More importantly, the resulting distributions of parameters are not amenable to the magnitude/phase coding algorithms described below, since the distributions of log area ratios are considerably skewed. Instead, comparable performance can be achieved by directly coding the magnitude and phase of the coefficient.

In Figures 41 and 42, two-dimensional scatter plots of the distributions of the eight reflection coefficients bounded by the unit circle are shown. The density, or intensity, of the dots is proportional to the amplitude of the probability density function of the coefficient. The corresponding magnitude and phase distributions are also shown in Figures 43 and 44. Note the considerable differences between the distributions of the four lower order coefficients and the four higher order coefficients. The phase of the higher order coefficients tends to be a uniformly distributed random variable, mainly due to the pre-emphasis filter, while the magnitude tends to have a Rayleigh-type distribution. The magnitude of a higher order coefficient typically is bounded by a constant less than one. The lower order coefficents, however, tend to have skewed distributions, especially with respect to the phase.

One simple approach to coding a complex parameter would be to encode both the magnitude and phase using a linear quantizer. As an example, suppose a total of four bits is allotted to this coefficient, two bits for the magnitude and two bits for the phase. One possible quantizer design to satisfy these constraints is shown at the upper left of Figure 45. Each cell depicts a region for which all values of the coefficient within that region are set to the value at its center. There are 4x4 or 16 possible values. What is objectionable about this quantizer (for lower order coefficients) is that the reflection coefficients of smaller

magnitude are quantized more accurately than the more probable reflection coefficients of larger magnitude.

The structure of this quantizer can be altered to better suit the coefficient distribution by doing one of two things. First, more bits could be allotted to the phase than to the magnitude. Second, the magnitude could be transformed such that the larger magnitudes are favored. One such transformation, commonly used, though not optimal, is the inverse sine function. The inverse sine function allows the distribution of the magnitude to appear more uniform. The quantizer at the upper right of Figure 45 is designed such that the inverse sine of the magnitude is encoded linearly, while the phase is still encoded uniformly. With suitable scaling of the magnitude, this coding scheme represents a good tradeoff between complexity and optimality. This two-dimensional quantizer design, called fixed magnitude/fixed phase coding, can be summarized as follows (Note that u_i is a scale factor or maximum magnitude.):

1. Convert k_i to magnitude/phase, that is,
   $$k_i = |k_i| \, e^{j\emptyset_i} .$$
2. If $|k_i| > u_i$, set $|k_i| = u_i$.
3. Compute $R_i = (2/\pi) \sin^{-1}[\,|k_i|/u_i]$.
4. Encode R_i and $\emptyset_i$ on a linear scale,
   using m_i bits for the magnitude and
   n_i bits for the phase.

The values of the scale factors, u_i, and bit assignments are given below in Table 1.

Table 1. Complex Reflection Coefficient Bit Allocations

Coefficient Number	Maximum Magnitude	Magnitude Bits	Phase Bits
1	1.0	7	7
2	1.0	7	7
3	0.9	6	6
4	0.8	6	6
5	0.7	5	5
6	0.6	4	4
7	0.5	4	4
8	0.5	3	3

The values in Table 1 were determined by subjective listening tests in which the quantization of a particular parameter was isolated, using the AMPLPC algorithm.

Still, however, it seems that a better job of quantization can be done, at a reasonable level of complexity. What is required is a coding scheme where coefficients with large magnitudes are favored over coefficients with smaller magnitudes. A technique to do this is called fixed magnitude/variable phase coding (see Figure 45 for an example). Here, the number of bits allotted to the phase depends upon the value of the magnitude. The phase bit allotment table can even be made nonlinear; it is easy to see that there are quite a number of possibilities. All of this quantization can be

conveniently performed using a table lookup procedure. In general, this coefficient coding can reduce the bit rate of the coefficient information by as much as 25%, or allow the improved quantization of the coefficients at an equivalent rate.

Let us examine the quantizer shown at the bottom left of Figure 45. A maximum of three bits is allotted to the phase, while a minimum of one bit is allotted when the magnitude is very small. By utilizing the long term statistics of a coefficient, the average bit rate for this coefficient will be the same four bits as in the fixed magnitude/fixed phase coding scheme. The structure is however better suited to the distributions in Figures 41 thru 44, provided we use a scale factor for the magnitude. Subjective listening tests indicate that this variable rate quantization of the coefficients does not degrade the speech noticeably, because the smaller reflection coefficients are not as sensitive to quantization as the larger ones. This is, in fact, merely a more exact approach to an optimal quantizer design, where source characteristics are exploited using a variable length code. At the bottom right of Figure 45 , the inverse sine transformation of the magnitude is combined with the variable phase coding. Here, even more emphasis is placed upon quantization of the large reflection coefficients.

In MPLPC, however, a majority of the rate is devoted to the driving function. The savings in rate by utilizing

these variable rate schemes are balanced by the increased system complexity, and the fact that they are variable rate schemes. Variable rate coding algorithms are notorious for being very sensitive to channel errors. For this reason, the fixed magnitude/fixed phase coding scheme was chosen. Though sub-optimal, its performance is very close to that achieved by the variable rate schemes.

Complex to Real Parameter Transformations

The capability to transform the complex reflection coefficients to the corresponding real coefficients is useful from both a coding standpoint and a compatibility standpoint. The AMPLPC system could be made compatible with MPLPC by transforming both the LPC polynomial and multi-pulse excitation into its real signal equivalent. While the transformation of the driving function will not be discussed here, there are several simple possibilities that might be considered. In fact, in a basic pitch-excited LPC system, the only transformation required is that of the LPC polynomial. It should be emphasized that the technique presented here is a transformation based on an impulse invariance design strategy. It does not produce the identical parameters that would have been computed using a real signal-based system, but rather produces a real LPC polynomial which approximates the frequency response (actually the impulse response) of the complex LPC polynomial.

The basic difference between the complex LPC polynomial and the real LPC polynomial is that the complex polynomial does not have conjugate poles. A simple technique to transform the complex polynomial is to find the roots of the polynomial, and add the conjugates of these roots to the polynomial. This is also the most exact way of performing the transformation. However, this is a computationally expensive process, generally involving extreme numerical accuracy for an accurate estimation of the roots. Alternately, the LPC polynomial can be transformed directly.

There exists a one-to-one mapping between the predictor coefficients and the autocorrelation function that produces these coefficients [40], which is

$$R(i) = \sum_{j=0}^{i} a_{i,j} R(i-j) , \quad 1 \leq i \leq p , \tag{139}$$

where the $a_{i,j}$ are defined in Equations 66 and 67. There also exists a simple relationship between the autocorrelation function of the analytic signal and the autocorrelation function of the original real signal [32]. This can be derived by computing the long-term autocorrelation function for the signal of Equation 3, and simplifies to

$$R_a(i) = 2R_{ss}(i) + 2jR_{ss}\hat{}(i) , \tag{140}$$

where $R_a(i)$ denotes the autocorrelation function of the analytic signal, $R_{ss}(i)$ denotes the autocorrelation function

of the real signal, and $R_{ss}\hat{}(i)$ denotes the crosscorrelation between the signal and its Hilbert transform.

A simple transformation strategy would be to compute the autocorrelation function from the predictor parameters, upsample the autocorrelation function using Equation 16, take the real part of this upsampled autocorrelation function, and perform the Durbin recursion [35] to obtain the corresponding real predictor coefficients, a_i. This is an autocorrelation-invariant type design, as the autocorrelation structure of the real LPC polynomial approximates that of its analytic counterpart. The above strategy can be circumvented because this type of transformation can also be performed directly on the LPC polynomial.

Suppose a set of p complex reflection coefficients are given. It is assumed that these coefficients imply the spectrum of an analytic signal which has been downsampled, otherwise the problem is trivial. The reflection coefficients can be transformed to predictor coefficients using the step-up procedure of Equation 66 and Equation 67 in Chapter V. The predictor polynomial of order p now represents the impulse response of a filter, and can be upsampled to yield 2p complex points using the complex interpolation function of Equation 16 in Chapter II, giving an upsampled polynomial, $H_u(z)$. This interpolation process is not exact in the sense that the impulse response in the upsampled domain is not a finite length impulse response. In

fact, to be more exact, the predictor polynomial can be upsampled to L complex points, where L >> 2p. However, since it is restricted to be a stable filter, the impulse response amplitude decays with time, and the errors introduced by truncation are minimized.

This complex polynomial still has no conjugate poles, but these can be introduced by defining a new polynomial, $H_c(z)$, such that

$$H_c(z) = 1 - \sum_{i=1}^{2p} a_i^* z^{-i} .$$

$H_c(z)$ is easily shown to have zeroes that are the conjugates of the original polynomial, $H_u(z)$.

Let A(x) be some polynomial with p complex zeroes, and no conjugate zeroes. Then, A(x) can be written as the product of its roots:

$$A(x) = (x-r_1)(x-r_2) \ldots (x-r_p) . \qquad (141)$$

The conjugate of A(x) is given by

$$A^*(x) = [(x-r_1)(x-r_2) \ldots (x-r_p)]^* ,$$

$$= [(x^*-r_1^*)(x^*-r_2^*) \ldots (x^*-r_p^*)] .$$

Now define a new polynomial, $A_c(z)$, such that

$$A_c(z) = A^*(z=x^*)$$

$$A_c(z) = [(z-r_1^*)(z-r_2^*)\ldots(z-r_p^*)] \ . \qquad (142)$$

The polynomial $A_c(z)$ is just a polynomial whose coefficients are the conjugates of the coefficients of A(x), and obviously it has zeroes that are the conjugates of the zeroes of A(x).

To complete the conversion we observe that the product of $H_u(z)$ and $H_c(z)$ will give a polynomial that has zeroes occuring as conjugate pairs. However, the product of these two polynomials is not in general an FIR filter due, once again, to the upsampling process. Two approaches can be taken at this point. An LPC analysis can be performed on this new series to obtain an FIR filter of order 2p, or the series can be truncated to the first 2p points. The first technique is optimal requiring further computation, while the second technique is only approximate requiring no additional work.

In this manner, a complex polynomial can be converted to its real equivalent. This allows simple one-dimensional coding schemes to be applied to this filter, as well as allowing the analytic system to be compatible with its real signal equivalent. The transformation process can be viewed as an alternate coding technique, though a very complicated and computationally expensive one. What remains as an adjunct to this thesis is a more complete study of this parameter transformation that includes performance comparisons between real and analytic LPC systems.

Complex Pulse Coding

The largest portion of the bit rate of the AMPLPC system is devoted to the multi-pulse excitation. These pulses have a complex-valued amplitude and a location within the frame that must be coded. Within a frame of speech, some of these pulses corresponding to the pitch pulses will be large, while the majority of these pulses will be small in amplitude, and randomly distributed throughout the frame. As stated in Chapter VI, typically 16 real-valued pulses located every 20 ms are sufficient. In the analytic case, this corresponds to 8 complex-valued pulses located over the same time interval. In order to accurately quantize the larger pulses, that are fewer in number, a gain term, corresponding to the magnitude of the largest pulse within a 10 ms frame, is transmitted. All pulse amplitudes are then normalized by this value, and encoded once again on a magnitude/phase basis. Since the smaller amplitude pulses are less significant, a variable number of bits can be allotted for the phase, depending upon the value of the magnitude. In practice however, this does not provide a significant savings in rate to warrant the increased complexity.

The actual magnitude/phase quantization and the location of the pulse can all be encoded into one value via a table lookup procedure. Subjective listening tests indicate that 13 levels of quantization for the magnitude and 15 levels of quantization for the phase of each pulse

are sufficient. Since the complex pulses are constrained to 20 sample point locations (similar to the real signal case where pulses are constrained to be located within 40 sample point locations), the entire information for a pulse can be efficiently encoded into 12 bits. The individual quantizers for the magnitude and phase are uniform quantizers, as these variables tend to be uniformly distributed. Two gain terms are transmitted per frame, each encoded logarithmically using six bits per gain term.

A 9.6 kbits/s AMPLPC System

The complete analytic signal-based MPLPC system is shown in Figure 46. The analytic signal is formed by passing the input speech through a Hilbert transformer. This Hilbert transformer is implemented as an 81-point FIR. The frequency response of the Hilbert transformer is shown in Figure 47 for 41 and 81-point designs. A 41-point design produces significant distortion in the ouput speech, while the differences in designs above 81-points is minimal. The bandpass filter at the input of the system eliminates energy in the speech signal around DC and 4 kHz, regions in which the Hilbert transform design has its largest error. Bandpass filtering the speech is a compromise that allows a reduction in the order of the Hilbert transform filter. The transfer function of the Hilbert transformer will always go to zero at DC and 4 kHz, so it is important that there be little speech energy in these regions.

After the analytic signal is downsampled by a factor of two, an eighth order complex Burg algorithm analysis is performed to generate the complex reflection coefficients. The residual produced by this analysis is used to synthesize the noise-weighted speech. Since the sampling rate is actually 4 kHz, the equivalent bandwidth-broadening factor (r') for AMPLPC can be found by recalling that the bandwidth increase, BW, is given by

$$BW = (1/\pi T)\ln(1/r) ,$$

or,

$$r = e^{-2\pi T(BW)} .$$

For the analytic system, since the sample period is

$$T' = 2T ,$$

then the equivalent bandwidth-broadening factor, r', is

$$r' = e^{-4\pi T(BW)} = r^2 .$$

Thus, since r = 0.85 for MPLPC, r' = 0.72 for AMPLPC.

The noise-weighted speech is used to locate the multi-pulse excitation. To allow the system to operate at 9.6 kbits/s, two complex pulses are located every 40 ms (20 complex samples). Since the frame length is 160 real points (80 complex samples), eight pulses are allocated per frame. At 12 bits per pulse, plus 12 bits for the two gain terms, 108 bits are expended for the driving function. After the driving function is found, both the

driving function and coefficients are coded. The coefficients are coded using the allocations of Table 1, bringing the total rate to 192 bits per frame, or 9.6 kbits/s. It is important to observe that the uncoded parameters are used in the pulse location process. This minimizes any effects due to quantization errors, and improves the quality of the synthetic speech considerably.

At the synthesizer, the packet of coded speech is unpacked. The driving function is constructed and fed into a complex lattice synthesizer along with the coefficients. The output of this synthesizer is fed into the reconstruction filter detailed in Chapter II. An 81-point FIR design for the Hilbert transformer is also used at the synthesizer. The use of a high-pass filter (f_c = 200 Hz) at the output of the system tends to remove some of the noise which appears due to quantization effects.

A Comparison of the Performance of AMPLPC to MPLPC

The performance of the AMPLPC system was compared to a similar MPLPC system. The MPLPC system implemented follows that developed by Atal [18]. The only fair way such a comparison can be made is to require both systems to operate at an equivalent rate. These comparisons were performed at two bit rates, 9.6 kbits/s and 14.4 kbits/s. In the complex case, the 14.4 kbits/s system differed from the 9.6 kbits/s system in three ways. First, so as to remove any severe quantization effects from the coefficients, eight bits were

allotted to the magnitude and phase of each of the eight coefficients. Next, eight bits were expended on each gain term. Finally, 12 complex pulses were used every frame, instead of eight.

Subjective listening tests indicated the AMPLPC system to be superior to MPLPC at both rates. In particular, the difference was larger at 14.4 kbits/s than at 9.6 kbits/s. The AMPLPC system generates synthetic speech that is less hoarse and more smooth than MPLPC. It is interesting to note that the LPC coefficients themselves are coded more efficiently as real coefficients than as complex coefficients. The system performance degrades considerably as the number of bits allotted to the complex coefficients is lowered from the 84 bits in Table 1, when using a fixed bit allocation. However, because of the downsampling process, the multi-pulse excitation, comprising the majority of the rate, is coded more efficiently in the complex domain. Thus, the overall effect is to provide improved quality at an equivalent rate.

The differences in quality are probably not great enough to justify the increased system complexity, however. The algorithm itself still exhibits the same problems in both real and complex domains. Typical of many waveform coders, as the bit rate goes down below a certain threshold, the synthetic speech becomes hoarse and scratchy. The system itself is very sensitive to different types of speakers, as well as coding algorithms, parameter

statistics, etc. The analytic signal offers an improvement, but in itself is not the solution. If further degradations in quality are accepted, by utilizing a fixed magnitude/variable phase coding algorithm, the rate of the AMPLPC system can be lowered to approximately 8 kbits/s. Here, the performance will be similar to that of the 9.6 kbits/s MPLPC system, but consistently of a lower quality than the 9.6 kbits/s AMPLPC system. Subjective listening tests seem to indicate a preference for the higher rate versions of this algorithm over the lower rate versions, as a definite threshold effect in performance versus bit rate exists.

CHAPTER VIII

SUMMARY

This dissertation has presented two new results concerning analytic signals. First, the analytic signal representation has been shown to be a natural choice for time series analysis. A new property of the analytic time series, called phase-invariance, allows analytic signal-based spectral estimators to achieve higher resolution than their real signal counterparts. This property implies that the magnitude spectrum of a periodic analytic signal computed from a frame-based spectral analysis algorithm will not vary as a function of time or equivalently, as a function of the position of the window. Thus, the spectral estimates obtained from parametric spectral analysis algorithms are more consistent, especially in the presence of additive noise.

Second, a new type of speech coder utilizing analytic signals has been shown to provide improved synthetic speech quality. An analytic signal-based multi-pulse LPC algorithm, the first of its kind, was implemented at 9.6 kbits/s to verify this. A fixed magnitude/variable phase coding algorithm was introduced as a sub-optimal technique of encoding a complex LPC reflection coefficient. The complex reflection coefficients, however, when viewed as a part of the entire AMPLPC system, were shown to be more sensitive to quantization than the real reflection coefficients. The multi-pulse excitation, on the other

hand, was more efficiently coded as an analytic quantity.

The design of the complex parameter coding algorithms presented in Chapter VII are by no means meant to be optimal. Instead, these algorithms were developed as a means to compare the performance of one system to another. Future work should be directed towards developing coding schemes which explicitly exploit the complex nature of the parameters. The variable rate algorithm presented was an initial attempt at doing this. There are many interesting possibilities involving vector quantization techniques which lend themselves to analytic signals, and any coding efficiencies which can be derived from these should be investigated.

The overall approach of this work has been to compare the performance of equivalent real and analytic systems. However, there is no guarantee that what is optimal for a real signal is optimal for an analytic signal. The development of a purely analytic speech coding algorithm, that truly exploits the complex nature of the signal, should be investigated in detail. There are several interesting questions to be resolved in regards to the coding efficiencies that can be achieved with a one-sided spectrum. Further, the sensitivity of the reconstruction process to these algorithms must be examined.

Finally, more basic questions exist in relation to the problem of parameter estimation. The LSMPLPC algorithm presented in Chapter VI was an attempt to decouple the

driving function of the speech signal from the vocal tract transfer function. This problem of joint estimation is common to many areas of signal processing. The iterative approach presented here is a promising technique. There are many questions to be answered concerning the validity of such an approach, as well as the convergence (or lack of convergence) of this kind of an iterative algorithm. Given some relatively accurate initial estimate of the driving function, can an estimate of both the driving function and the filter characteristic be obtained simultaneously? The answer to this question has many implications not only in spectral estimation problems, but also in the adjacent areas of speech recognition and speech coding.

CHAPTER IX

FIGURES

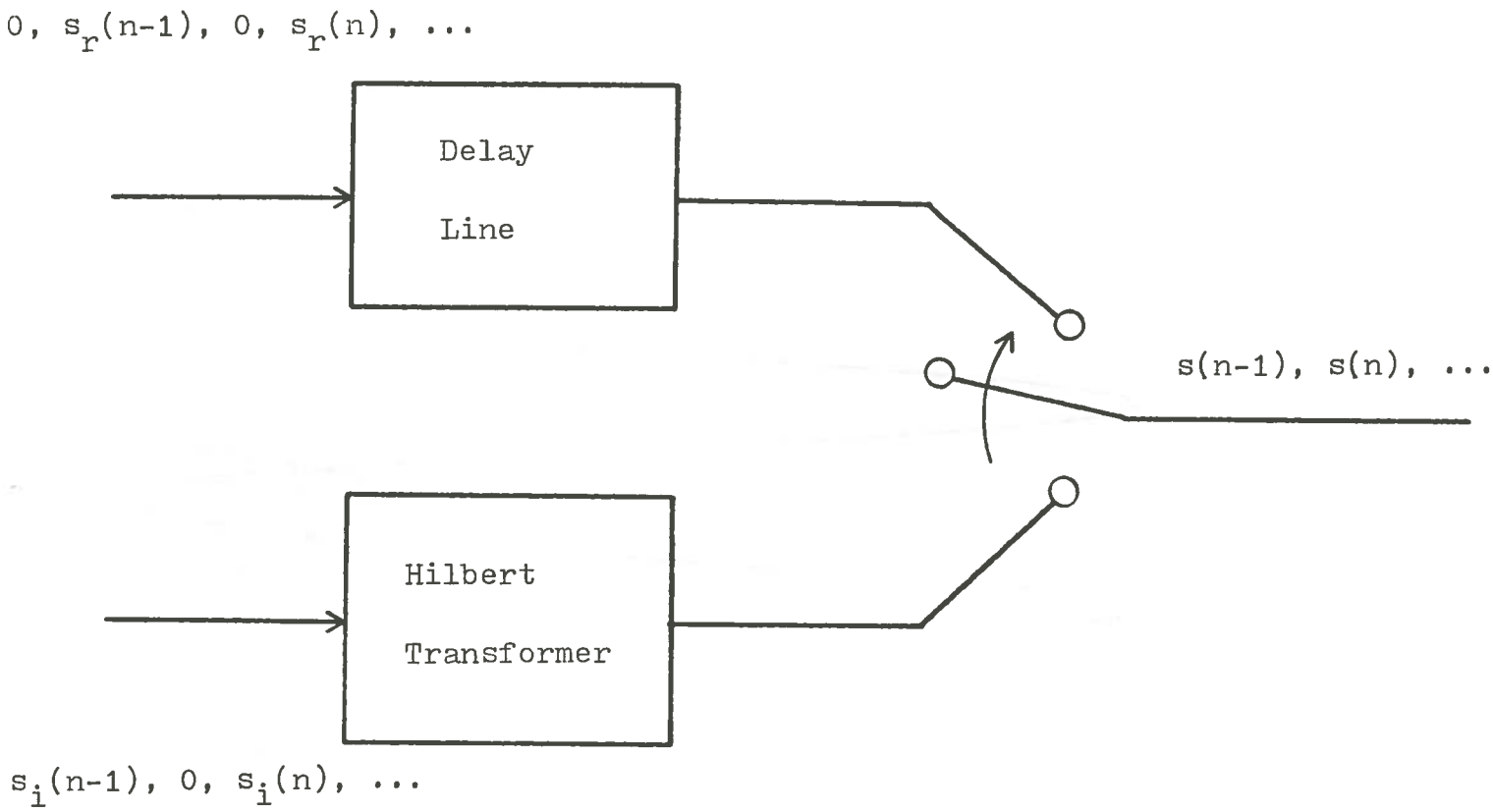


Figure 1. Reconstruction of a Real Signal from a Downsampled Analytic Signal

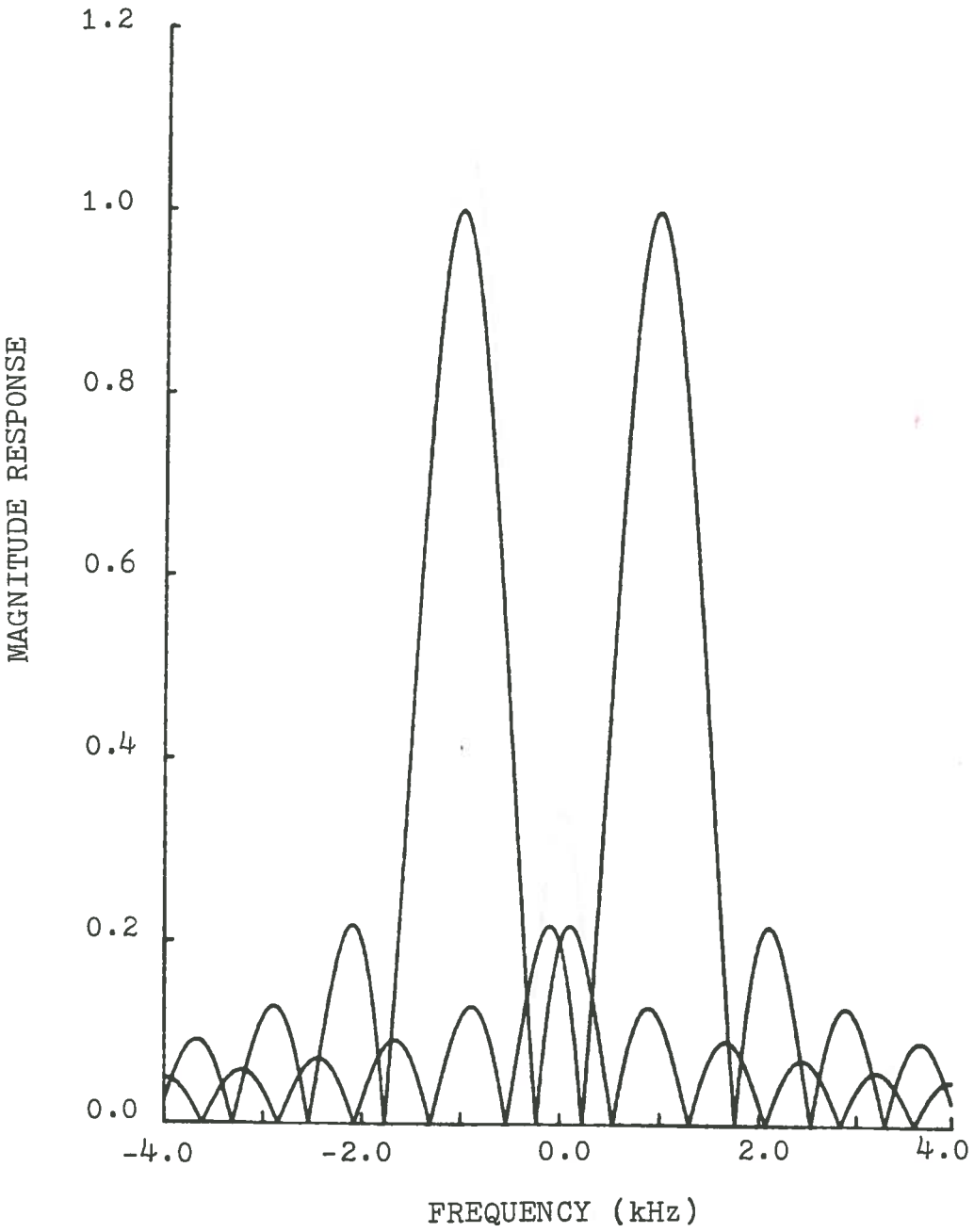


Figure 2. Aliasing in the Spectrum of a Single Real Windowed Sinewave

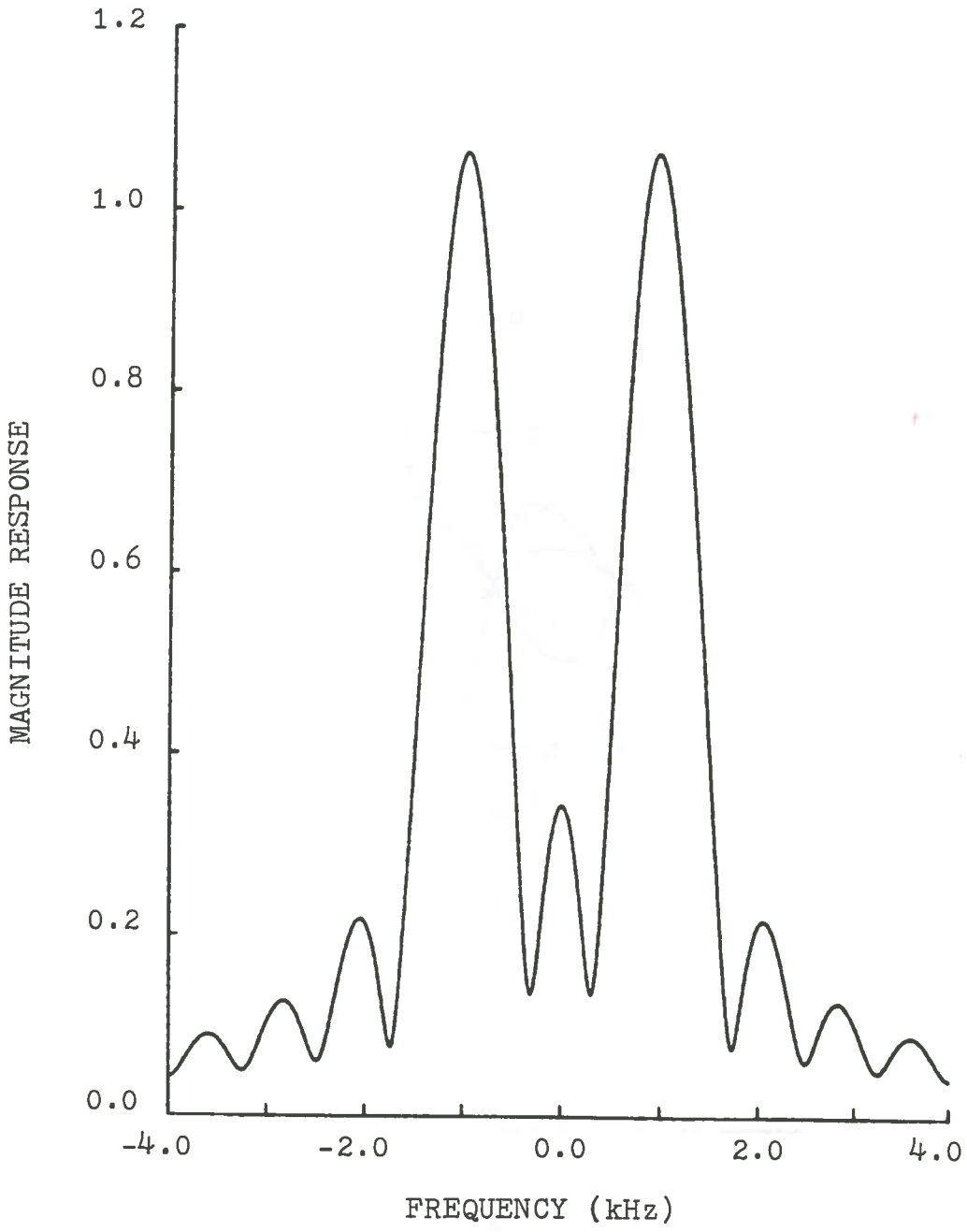


Figure 3. Composite Magnitude Spectrum of a Single Real Windowed Sinewave

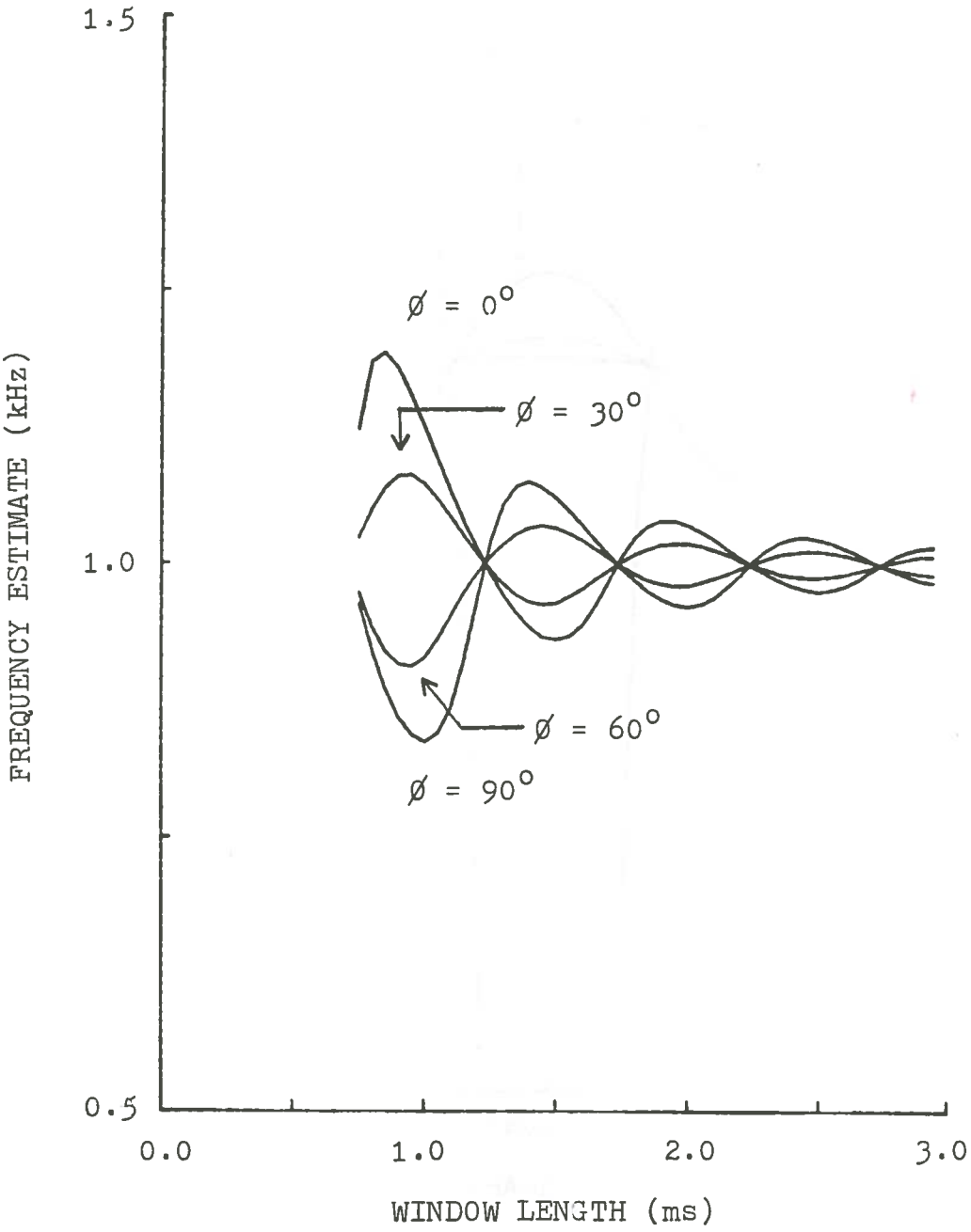


Figure 4. Dependence of the Magnitude Spectrum on Window Length

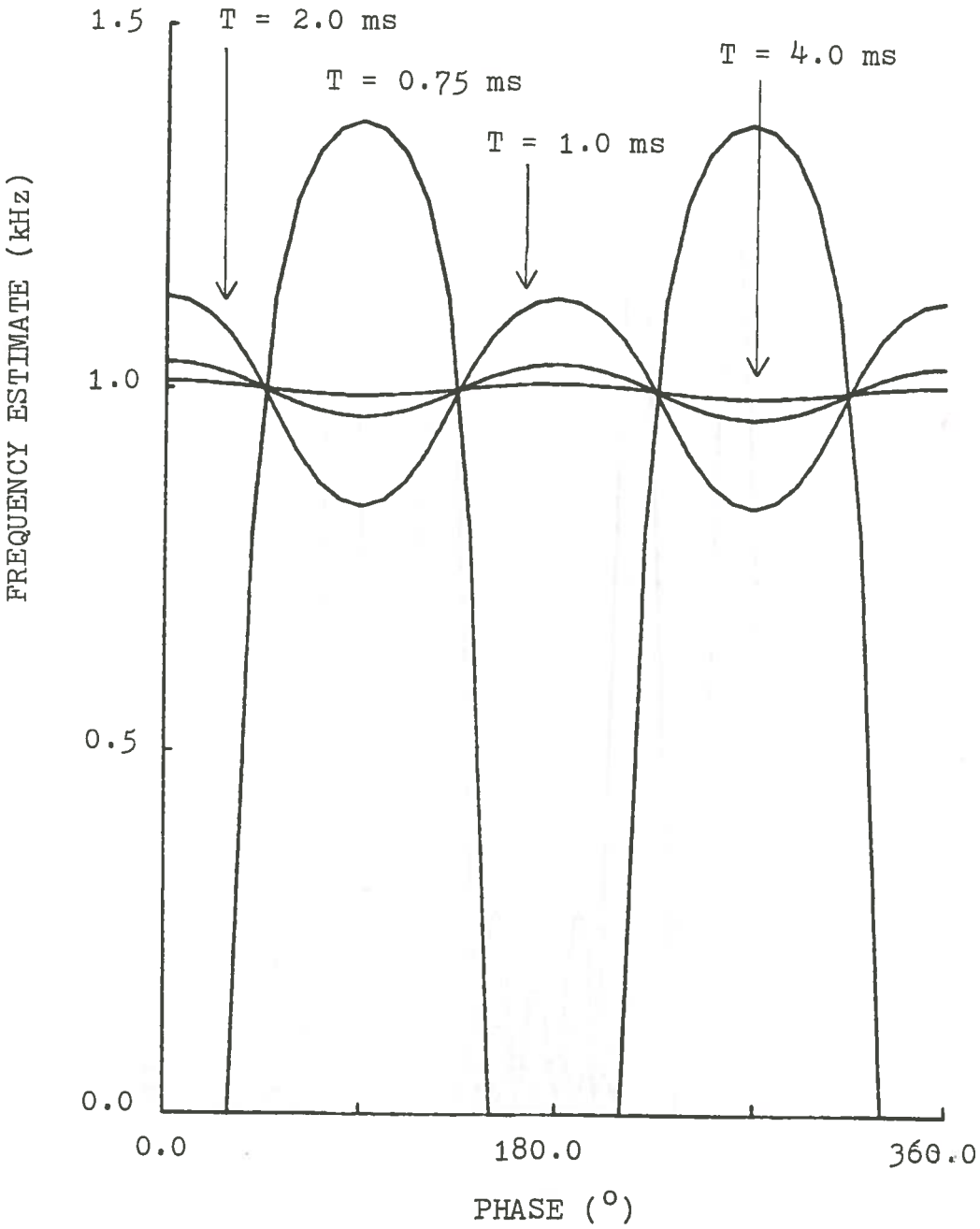


Figure 5. Dependence of the Magnitude Spectrum on Phase

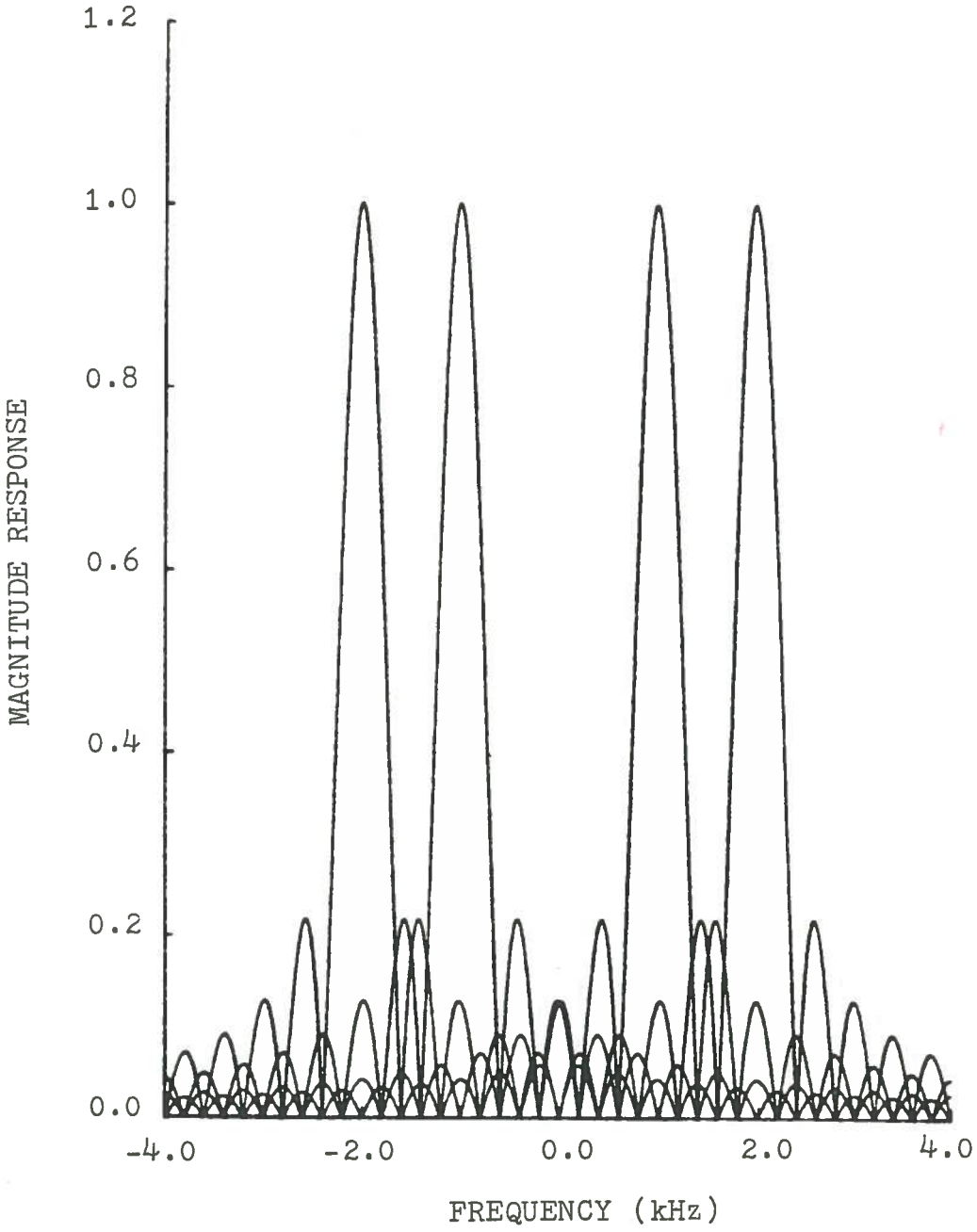


Figure 6. Aliasing in the Spectrum of Two Sinewaves

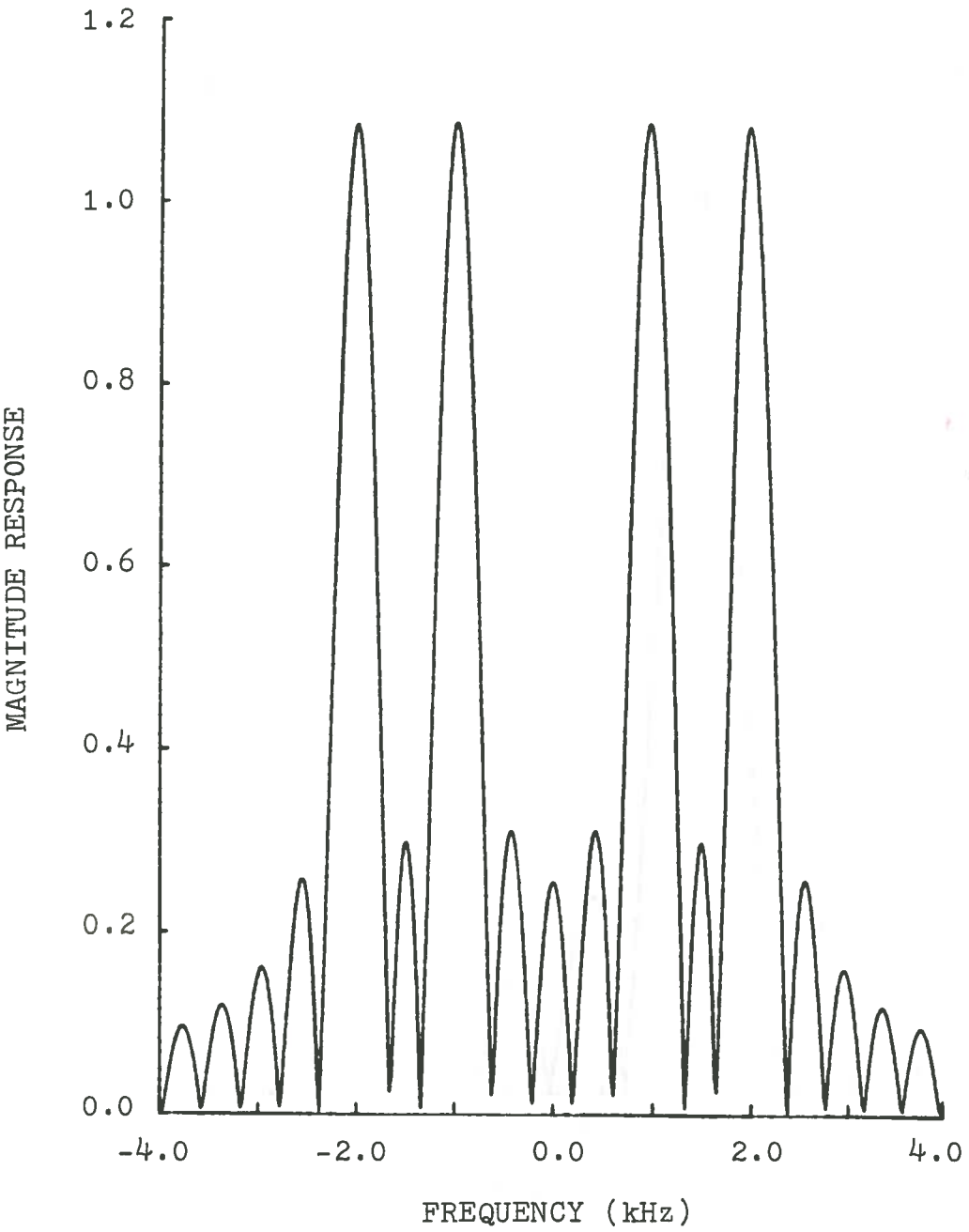


Figure 7. Composite Spectrum of Two Sinewaves

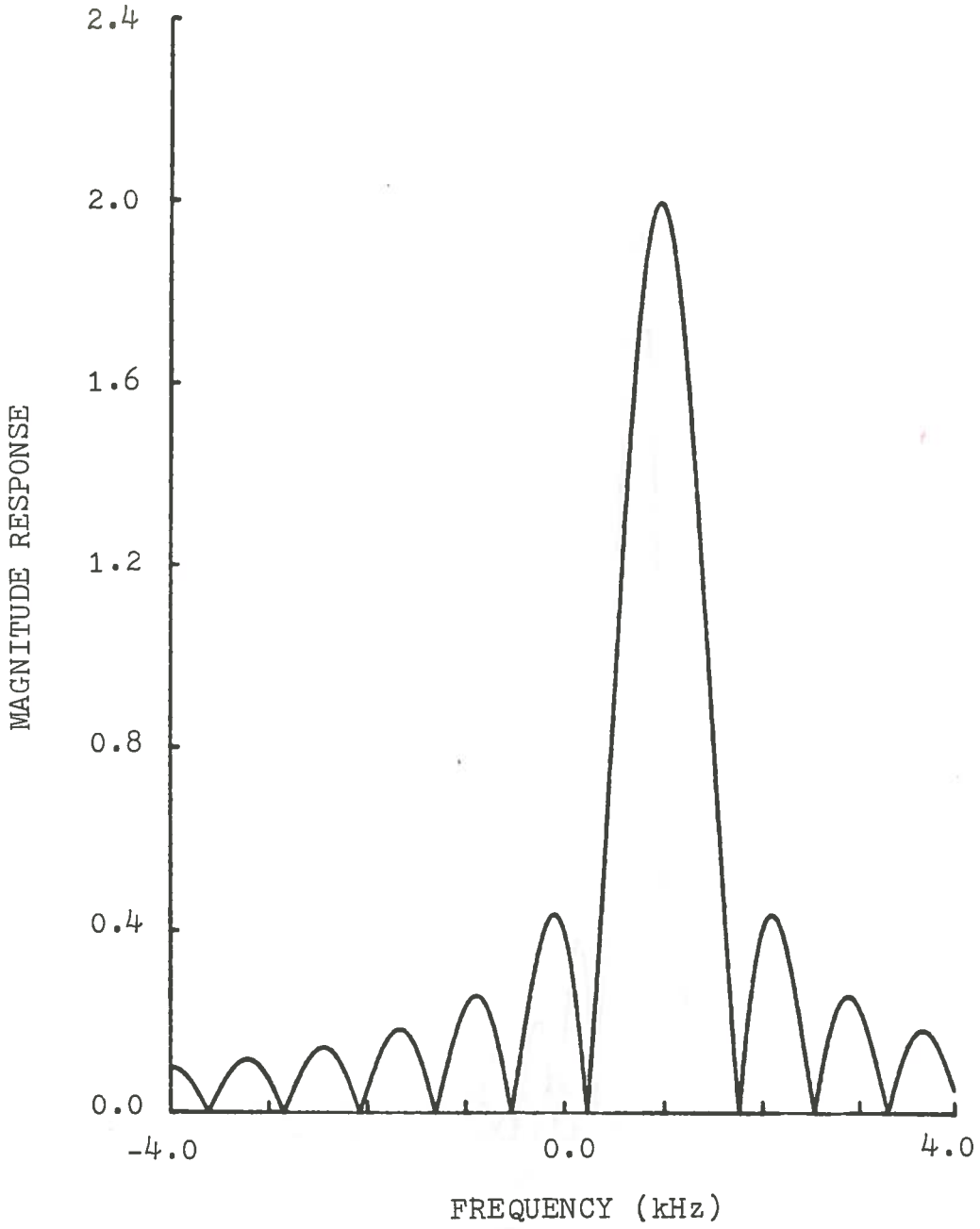


Figure 8. Composite Spectrum of a Complex Sinewave

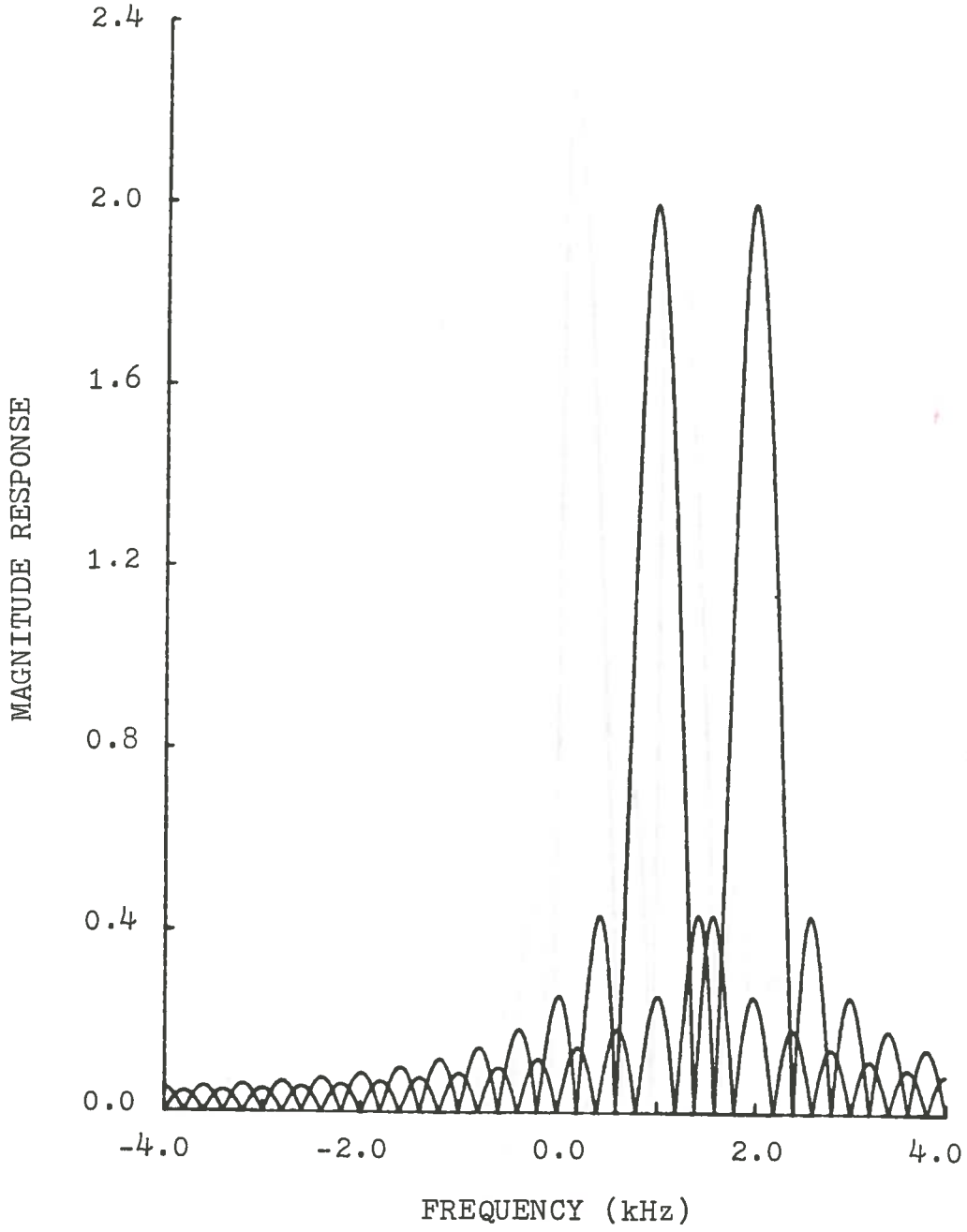


Figure 9. Aliasing in the Spectrum of Two Complex Sinewaves

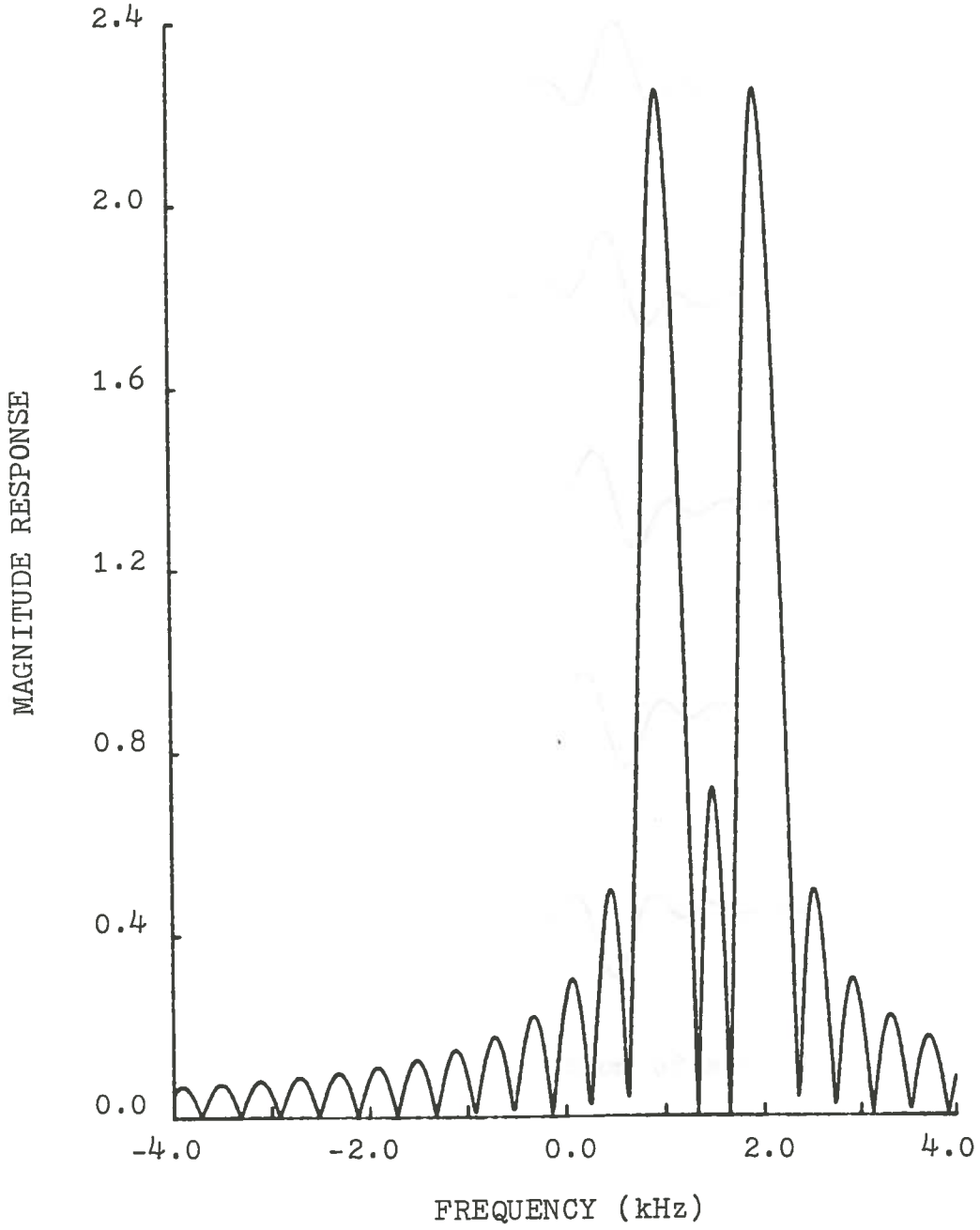


Figure 10. Composite Spectrum of Two Complex Sinewaves

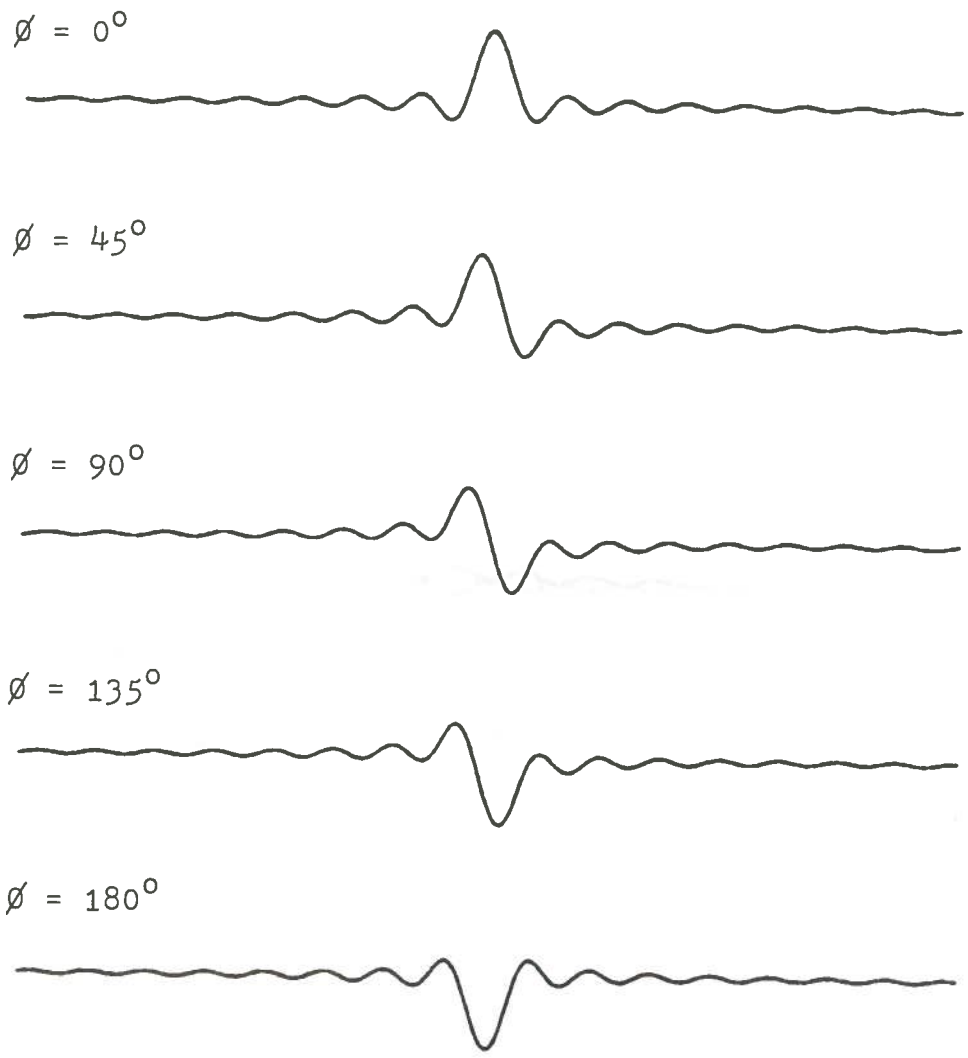


Figure 11. Phase Rotation of a Real Signal

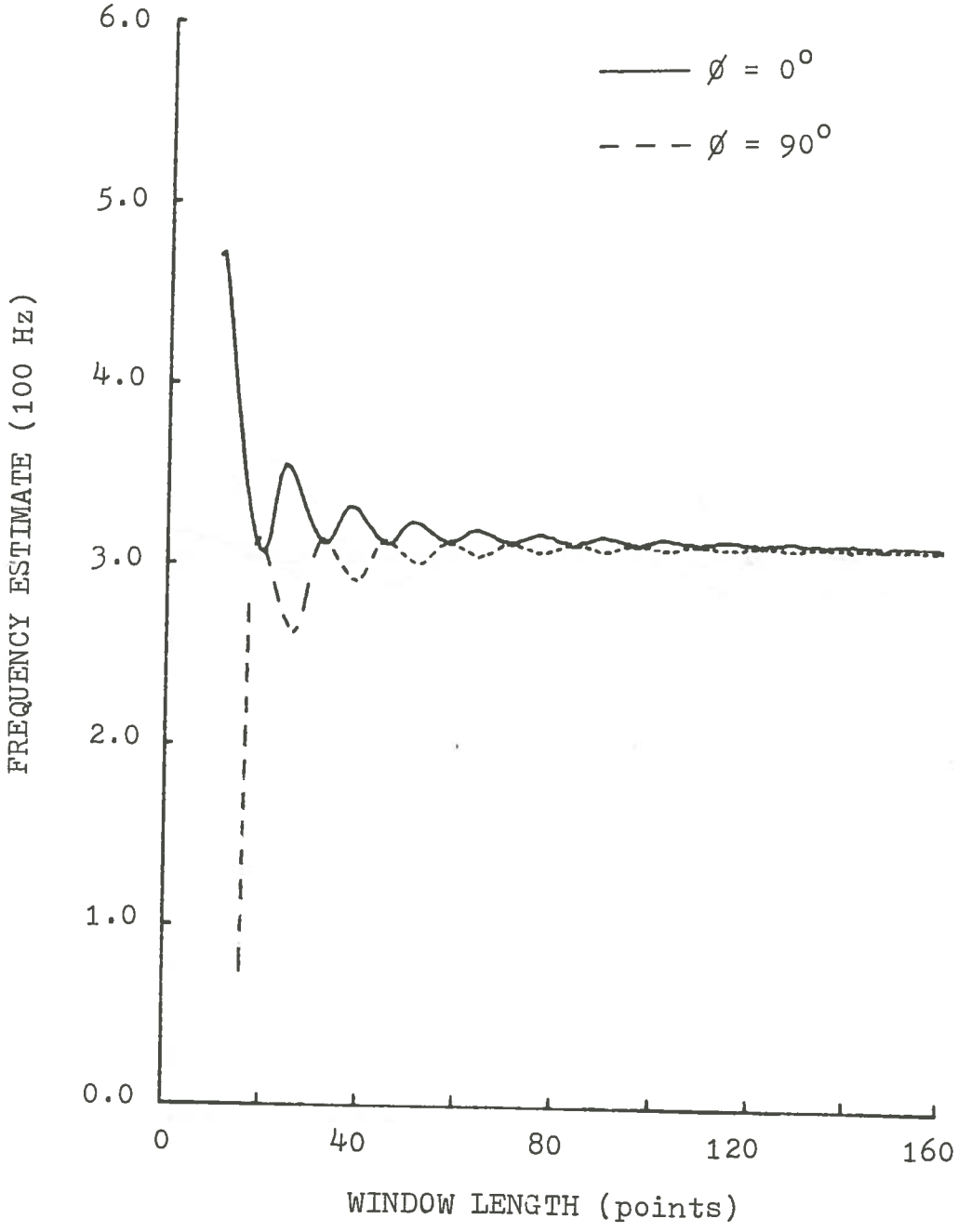


Figure 12. Dependence of the Discrete Magnitude Spectrum of a Real Sinewave on Window Length

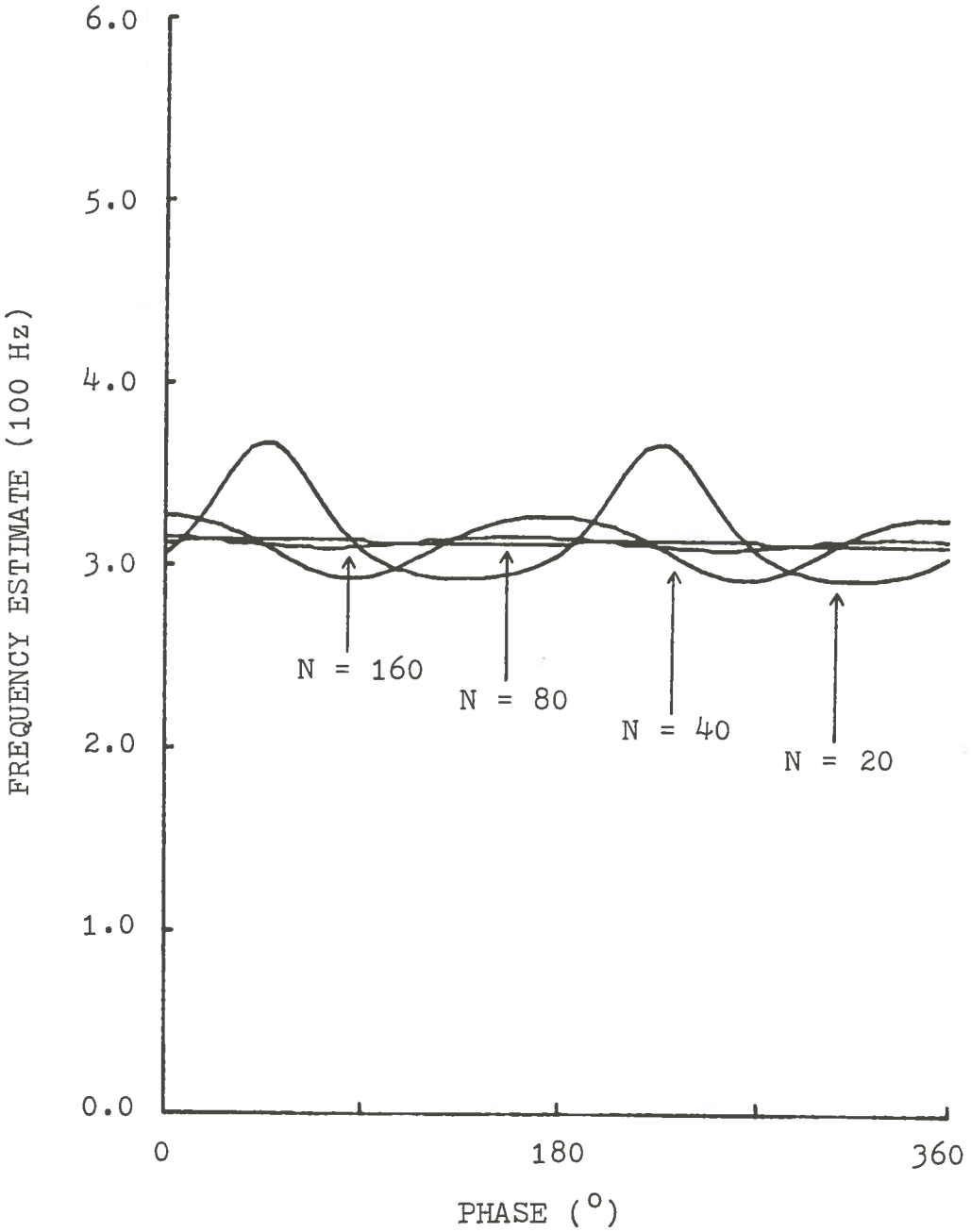


Figure 13. Dependence of the Discrete Magnitude Spectrum of a Real Sinewave on Phase

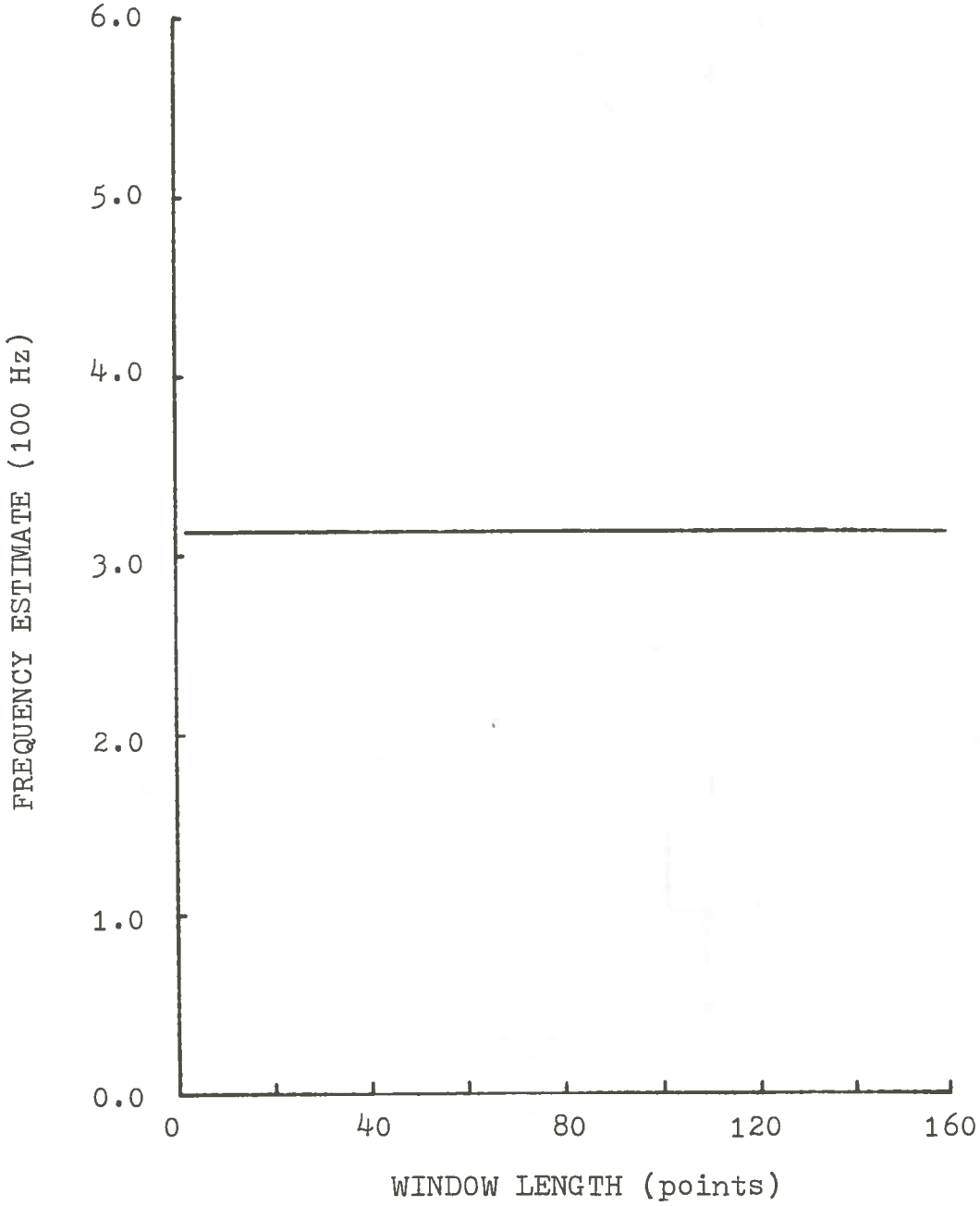


Figure 14. Invariance of the Discrete Spectrum of a Complex Sinewave to Window Length

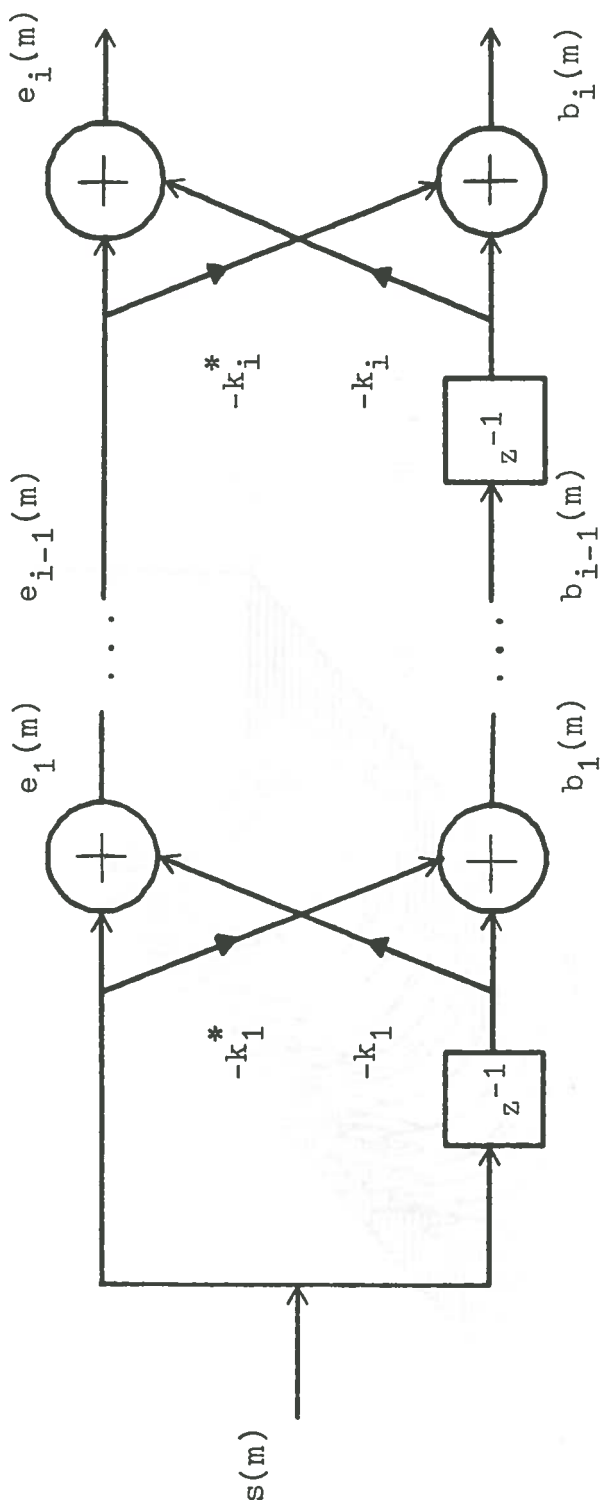


Figure 15. Complex Lattice Filter Analyzer

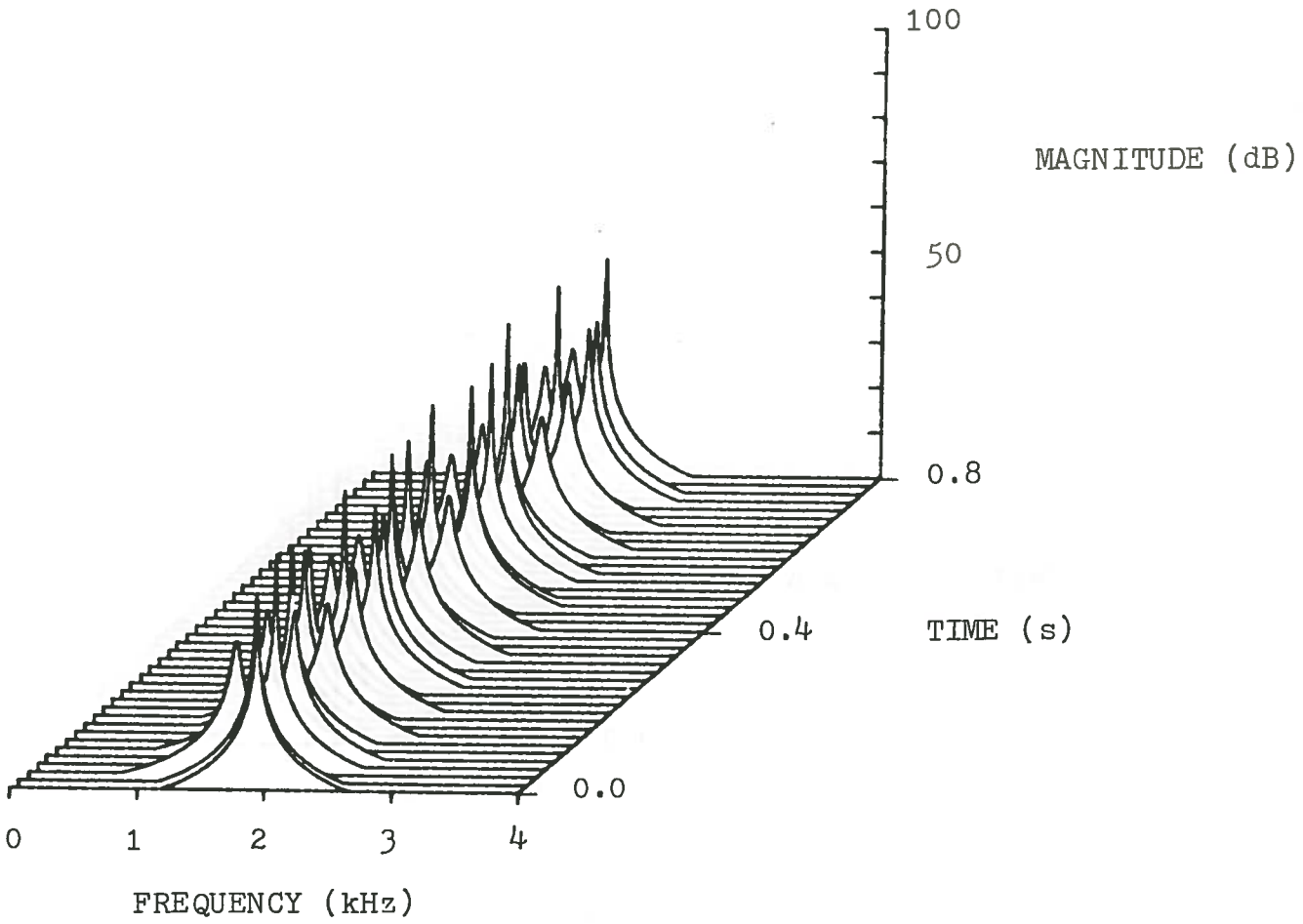


Figure 16. Maximum Entropy Spectra of a Real Sinewave

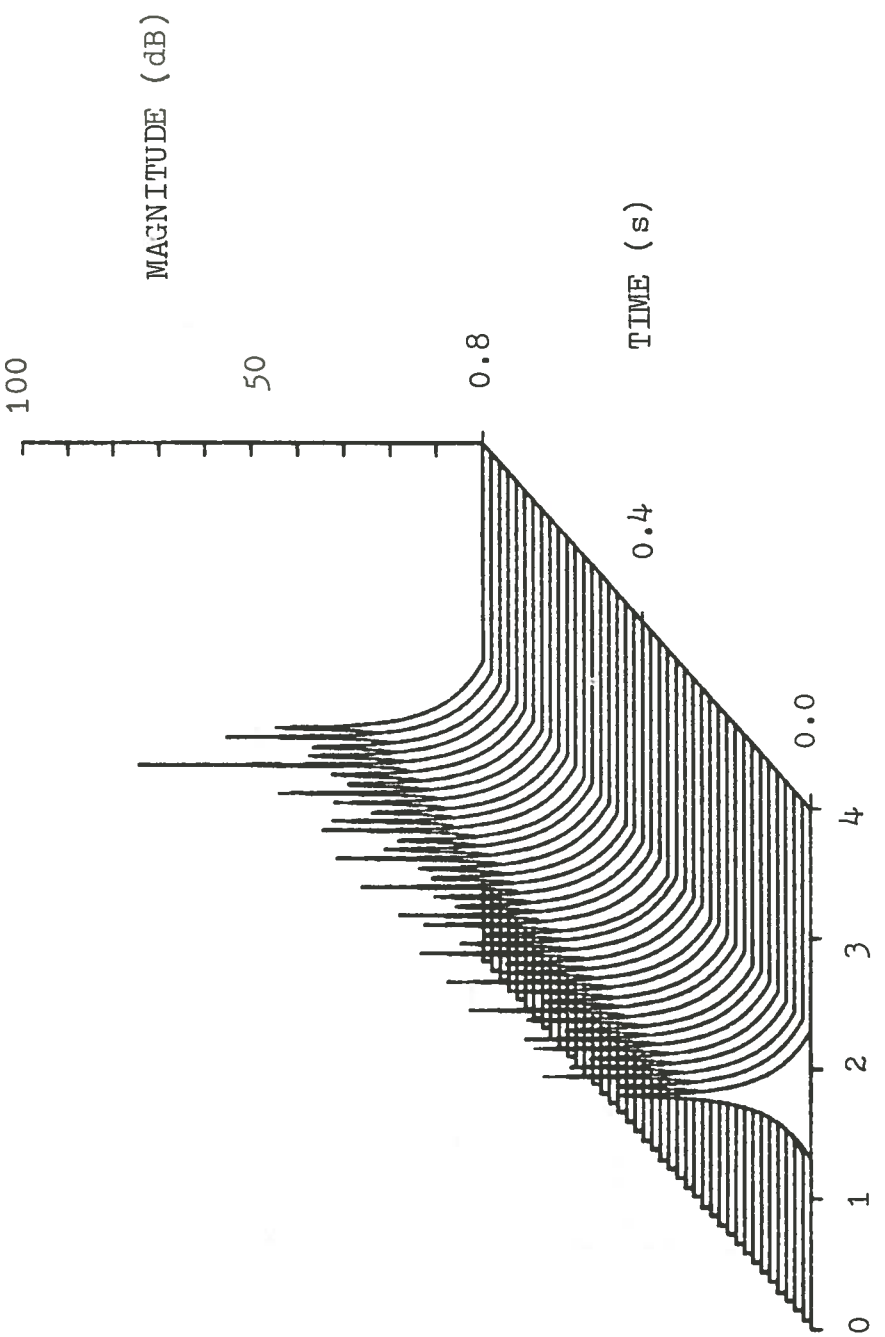


Figure 17. Maximum Entropy Spectra of a Complex Sinewave

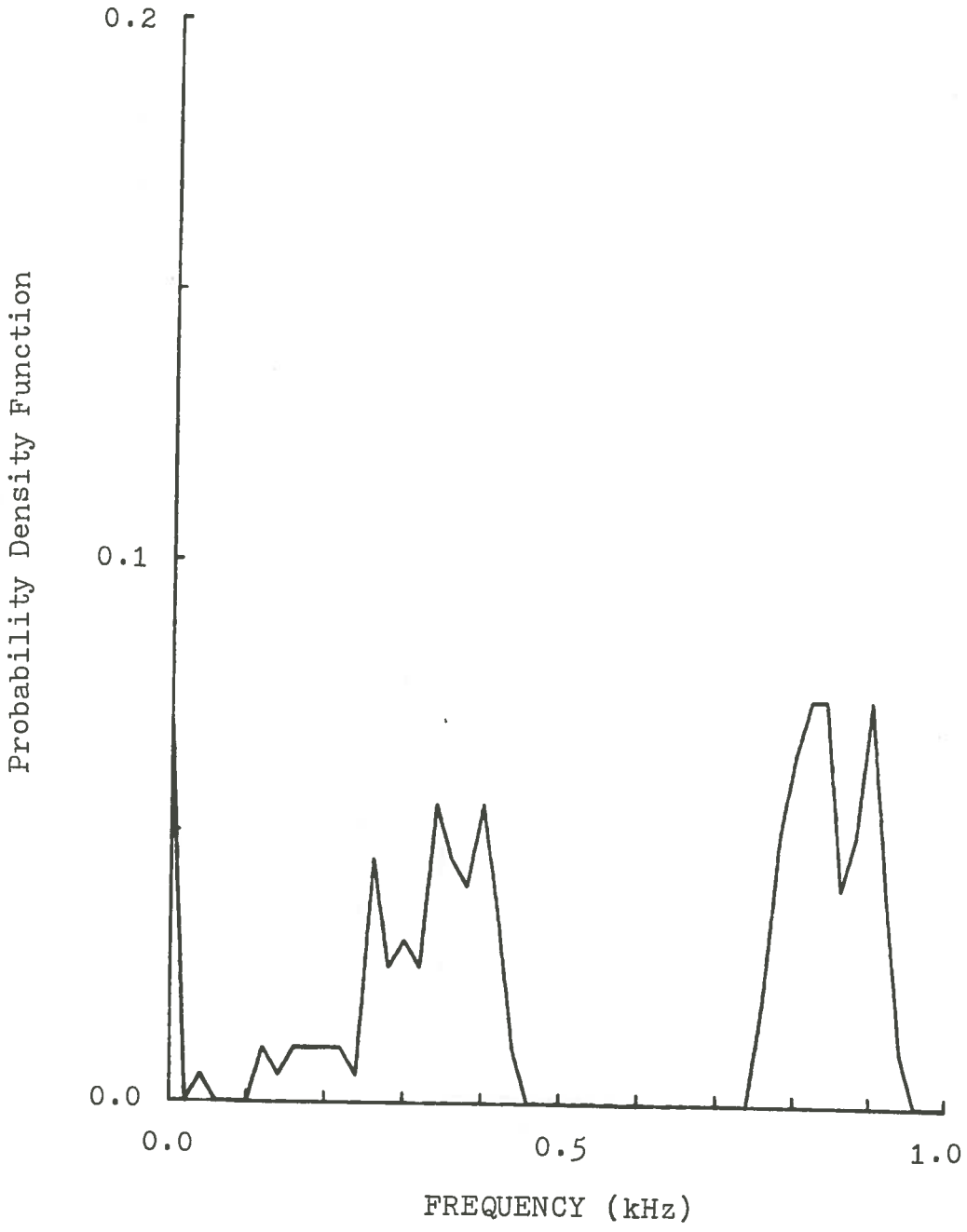


Figure 18. Distribution of Frequency Estimates for Two Real Sinewaves in Noise (f_1 = 367 Hz, f_2 = 859 Hz)

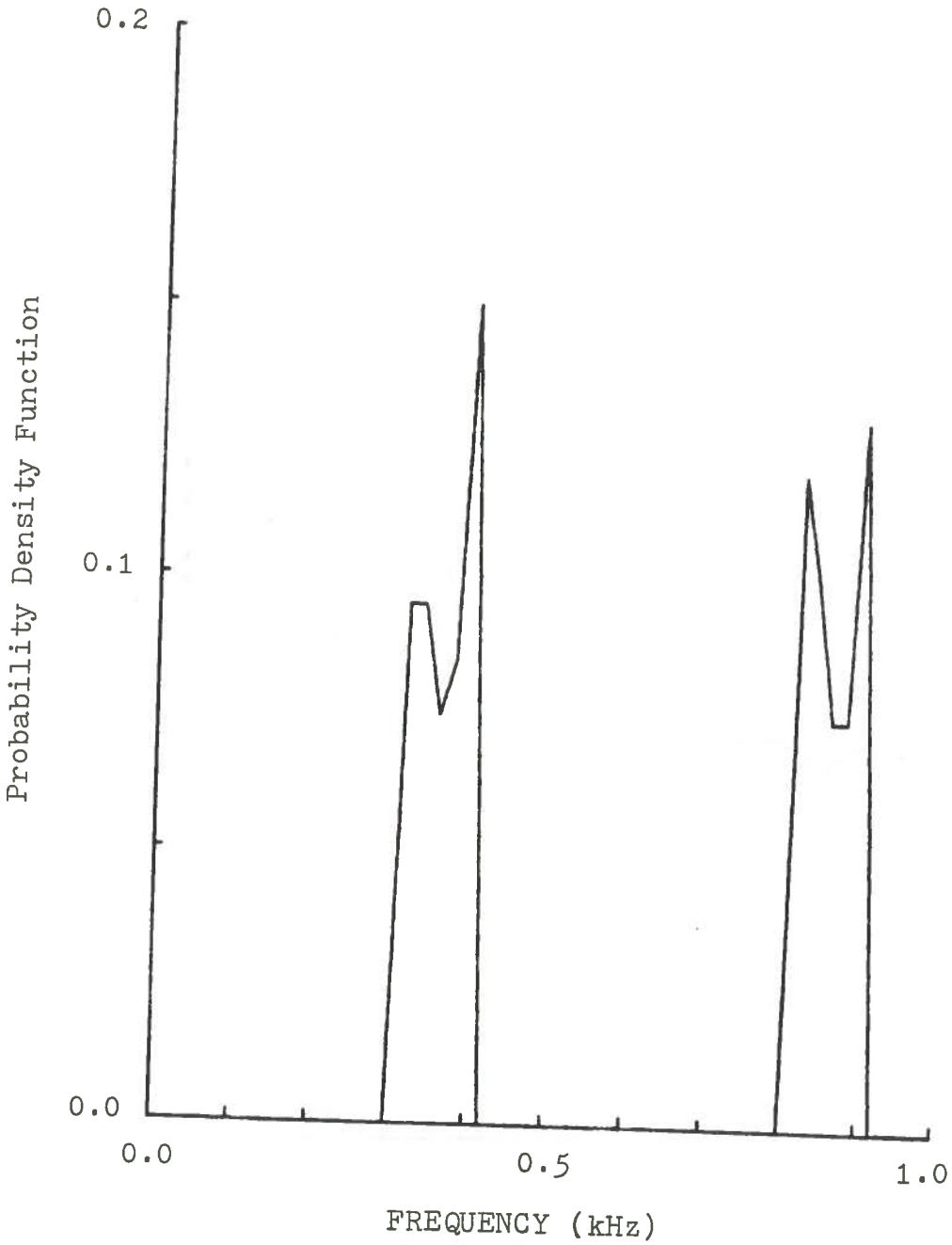


Figure 19. Distribution of Frequency Estimates for Two Complex Sinewaves in Noise (f_1 = 367 Hz, f_2 = 859 Hz)

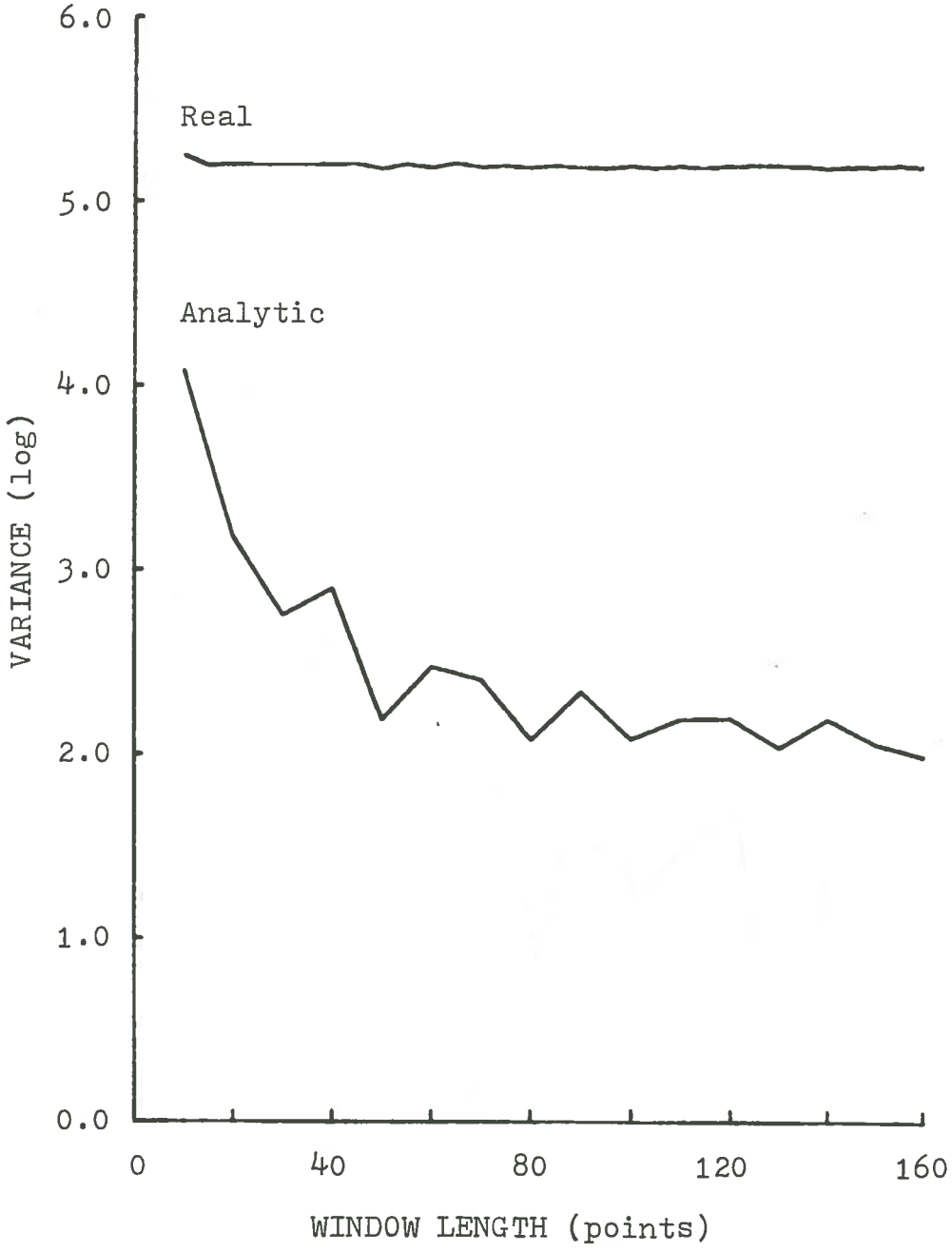


Figure 20. Dependence of the Variance of the Frequency Estimates on Window Length (SNR = 20 dB)

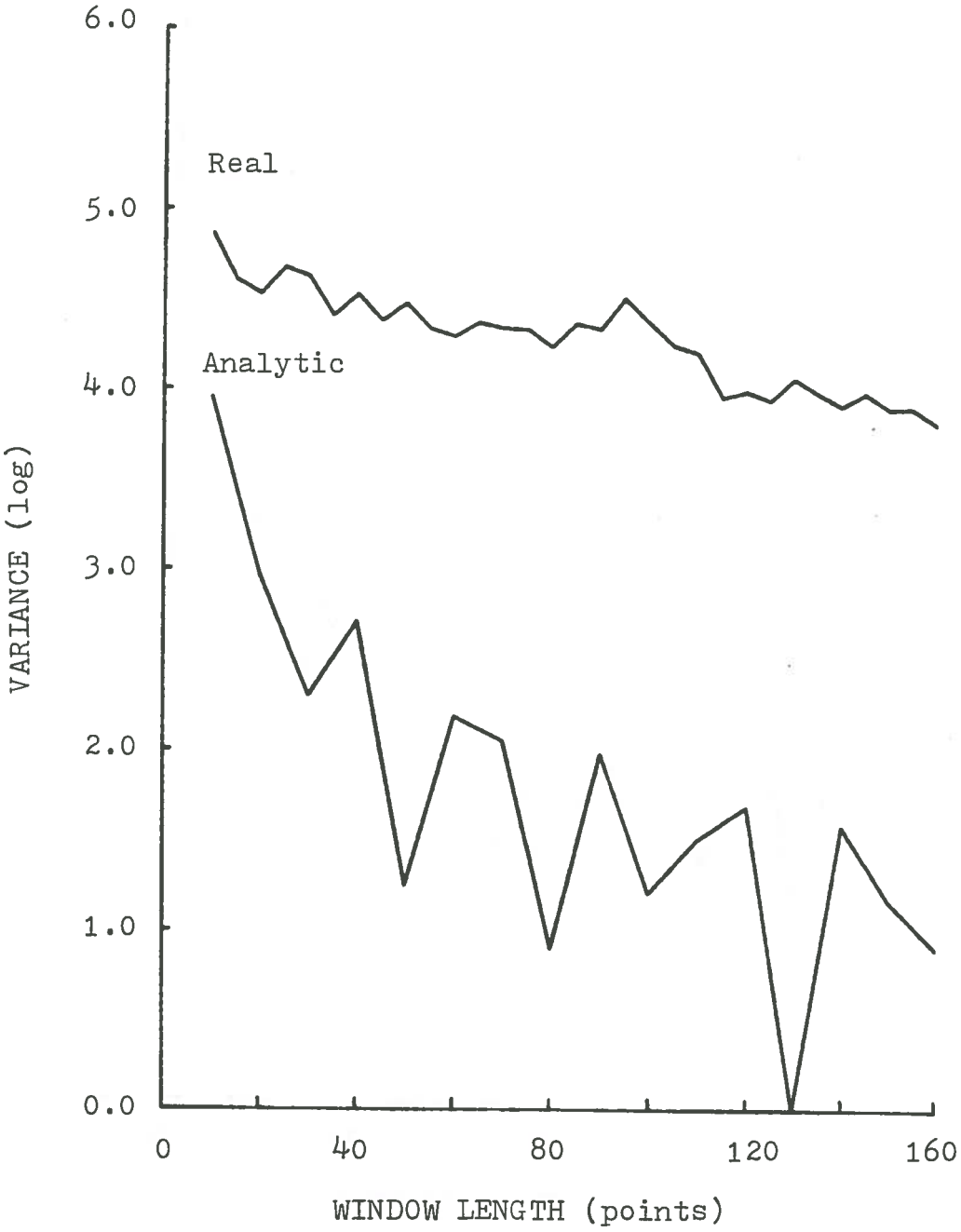


Figure 21. Dependence of the Variance of the Frequency Estimates on Window Length (SNR = 40 dB)

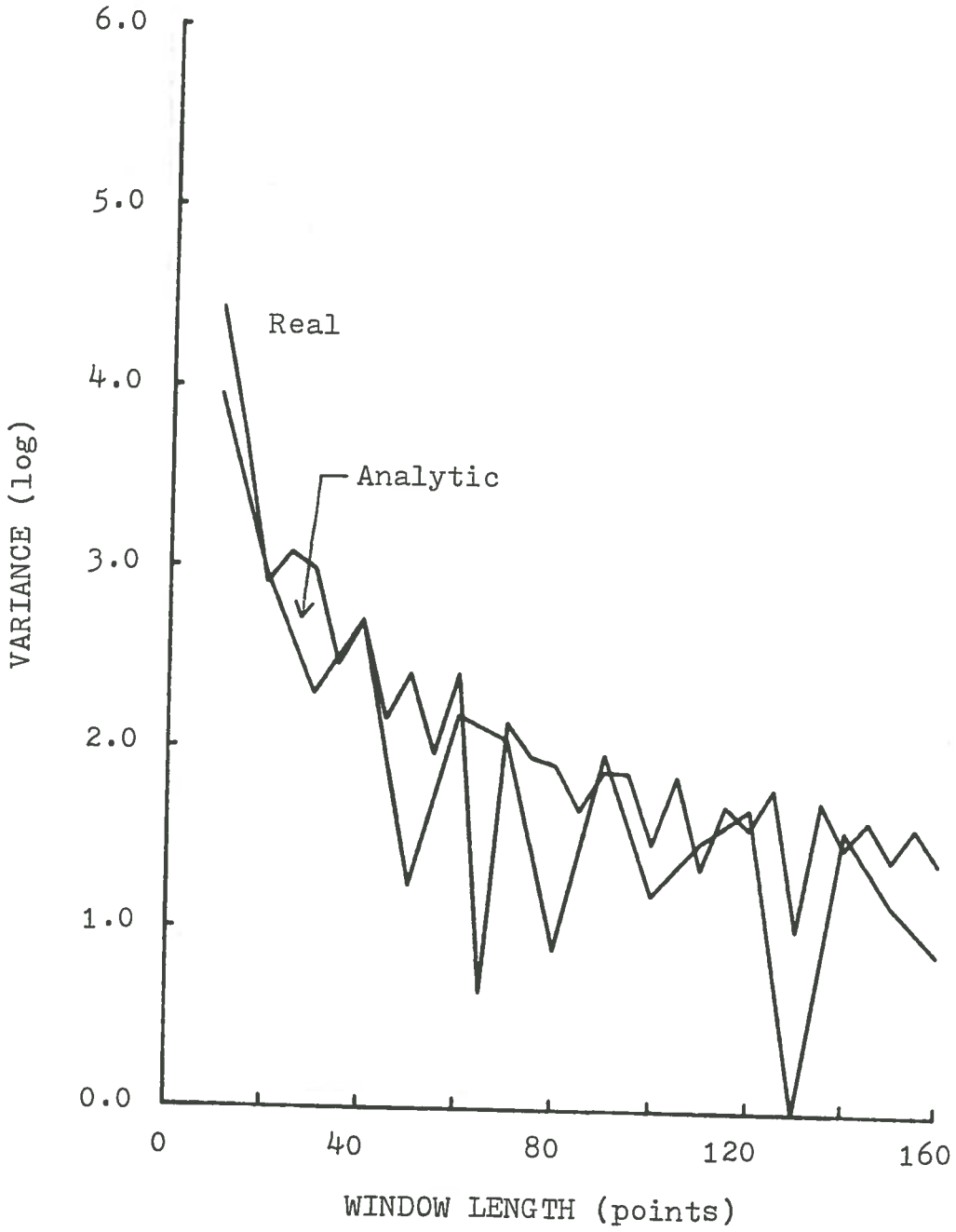


Figure 22. Dependence of the Variance of the Frequency Estimates on Window Length (SNR = 60 dB)

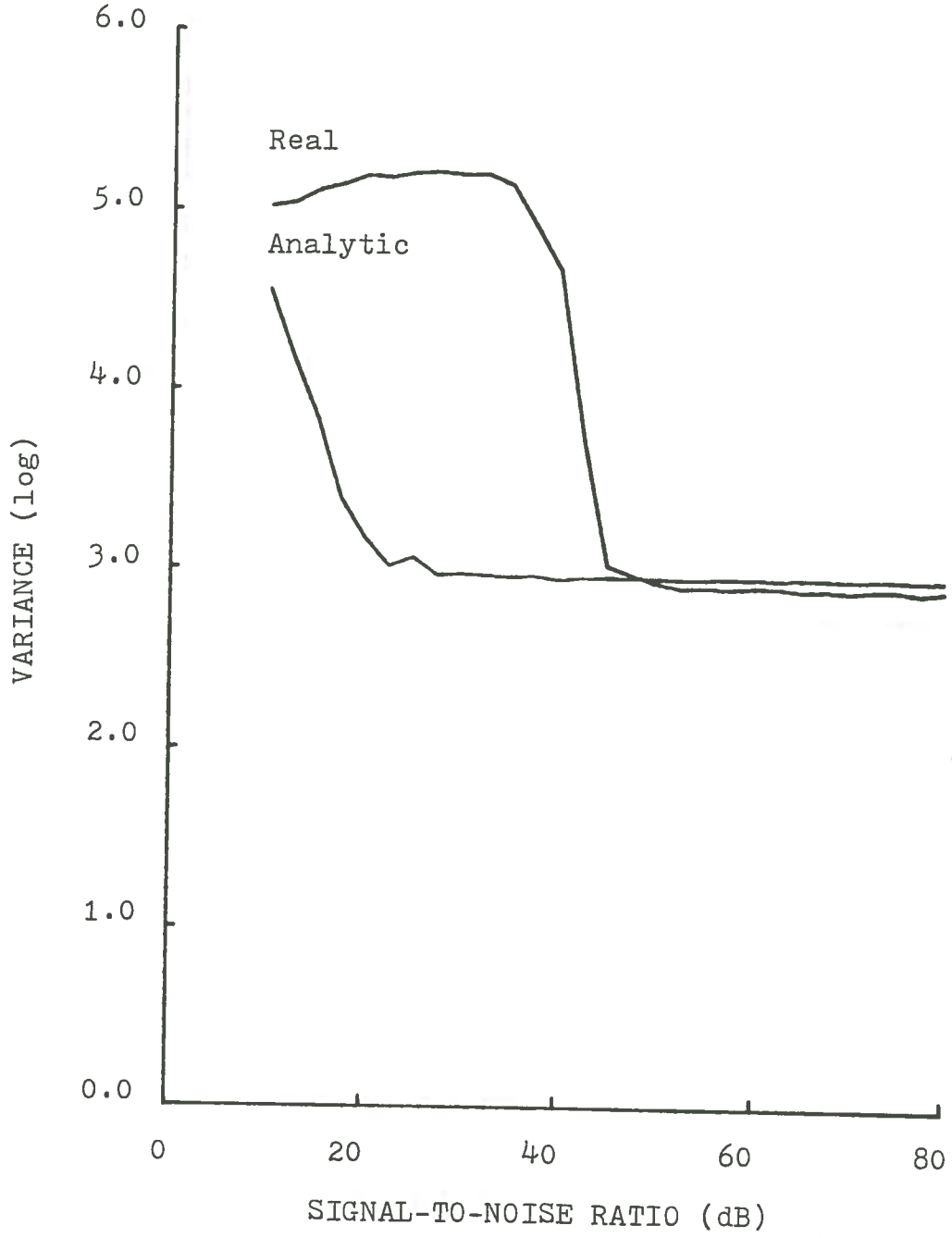


Figure 23. Dependence of the Variance of the Frequency Estimates on Signal-to-Ratio (N = 20 points)

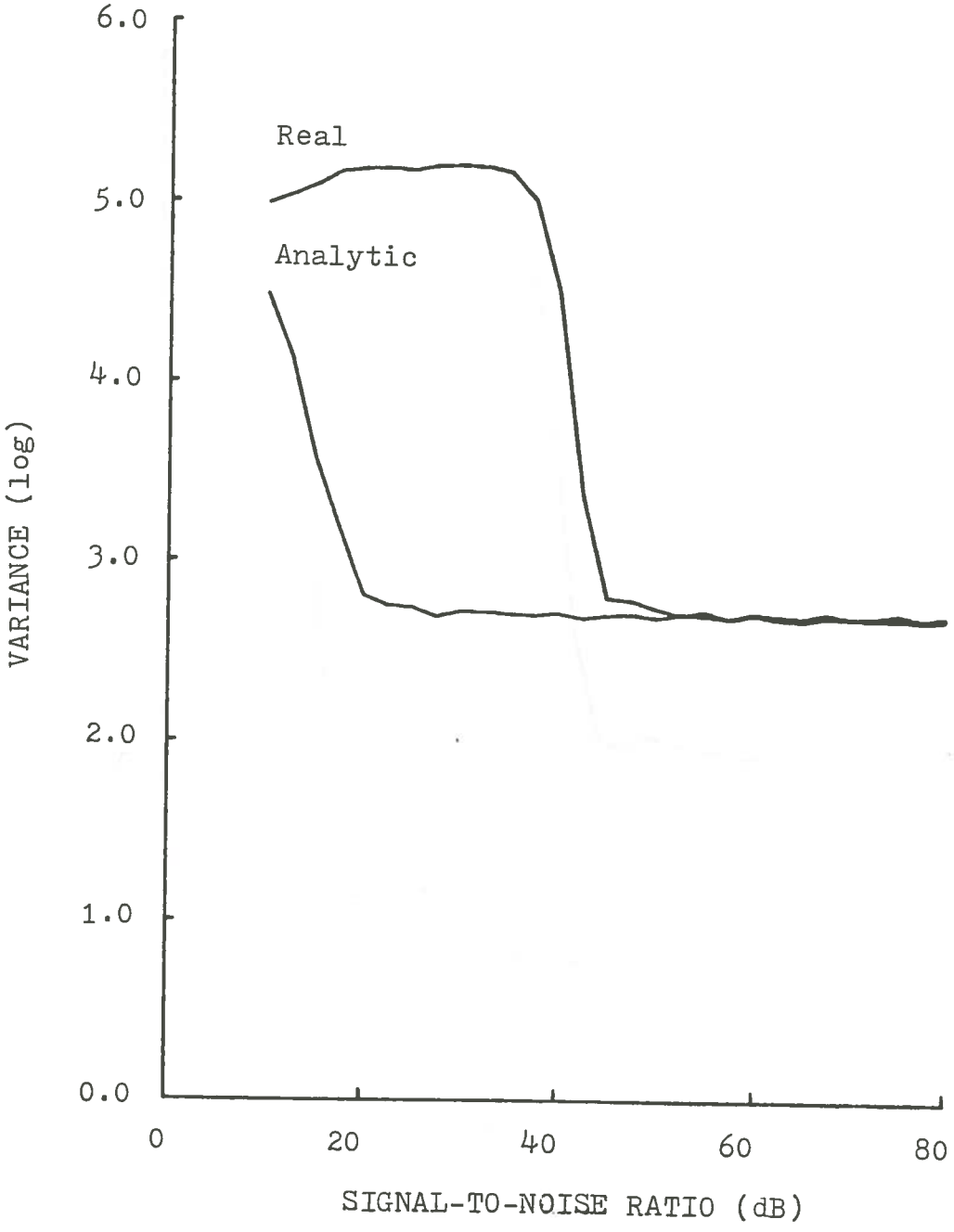


Figure 24. Dependence of the Variance of the Frequency Estimates on Signal-to-Noise Ratio (N = 40 points)

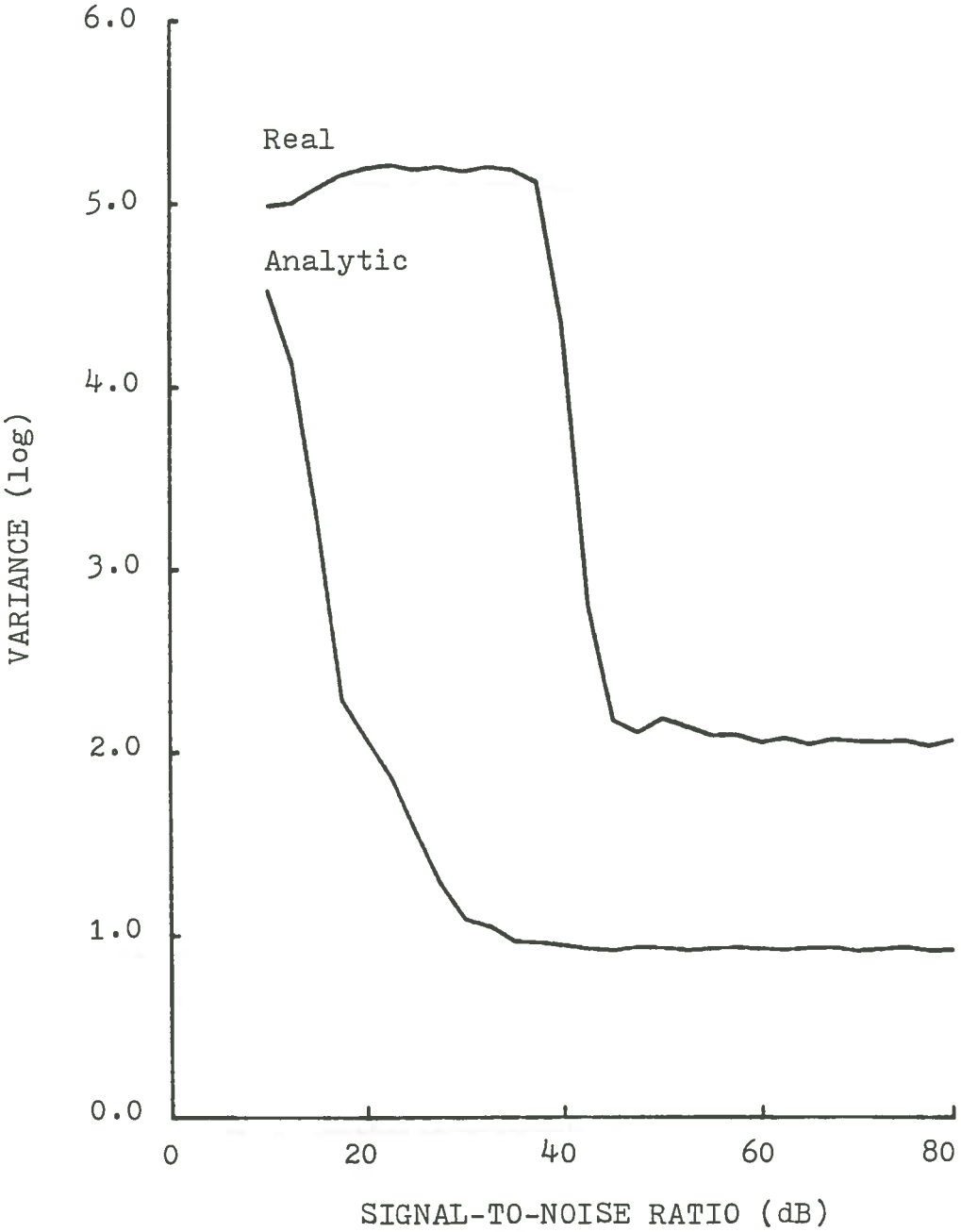


Figure 25. Dependence of the Variance of the Frequency Estimates on Signal-to-Noise Ratio (N = 80 points)

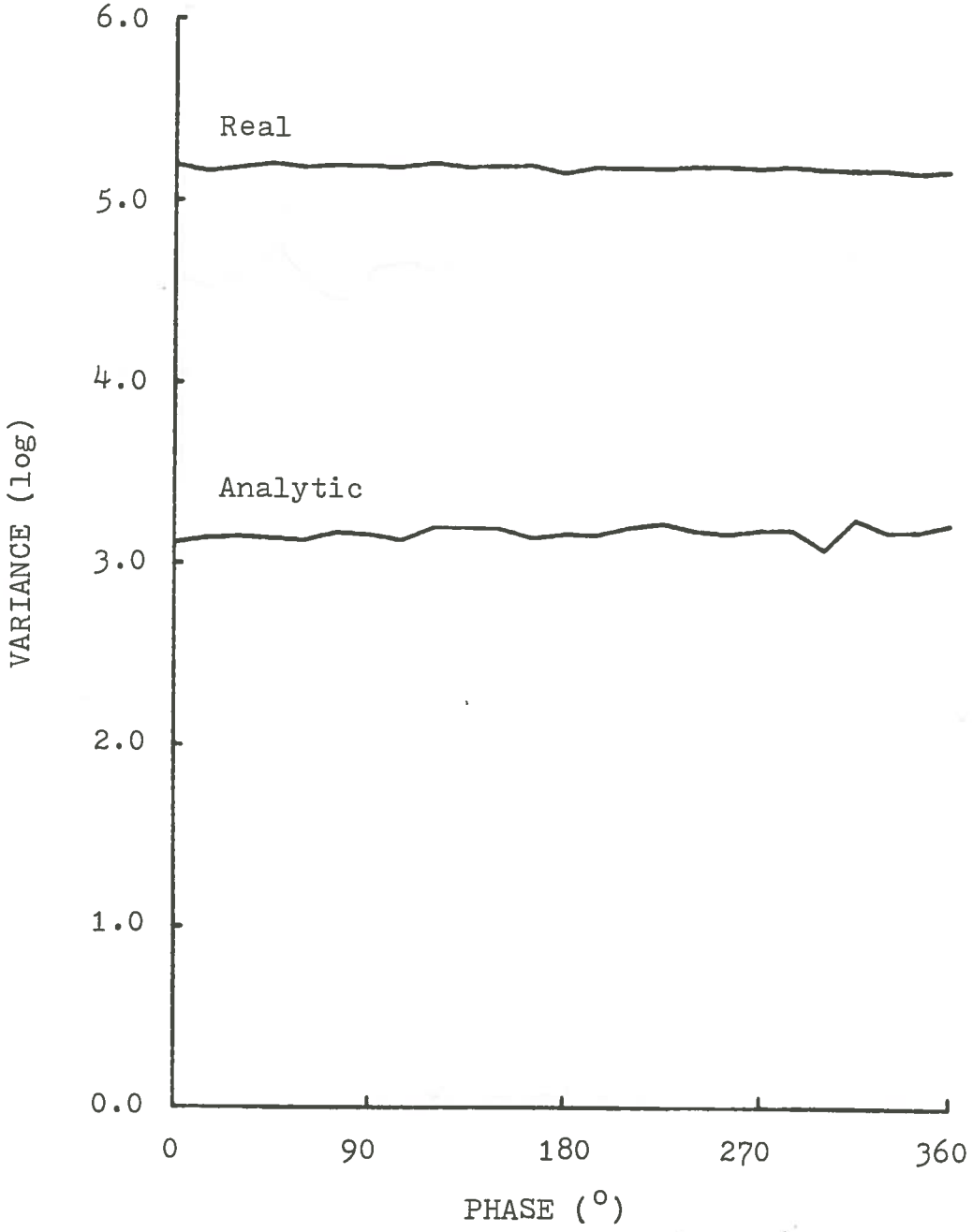


Figure 26. Dependence of the Variance of the Frequency Estimates on Phase (SNR = 20 dB)

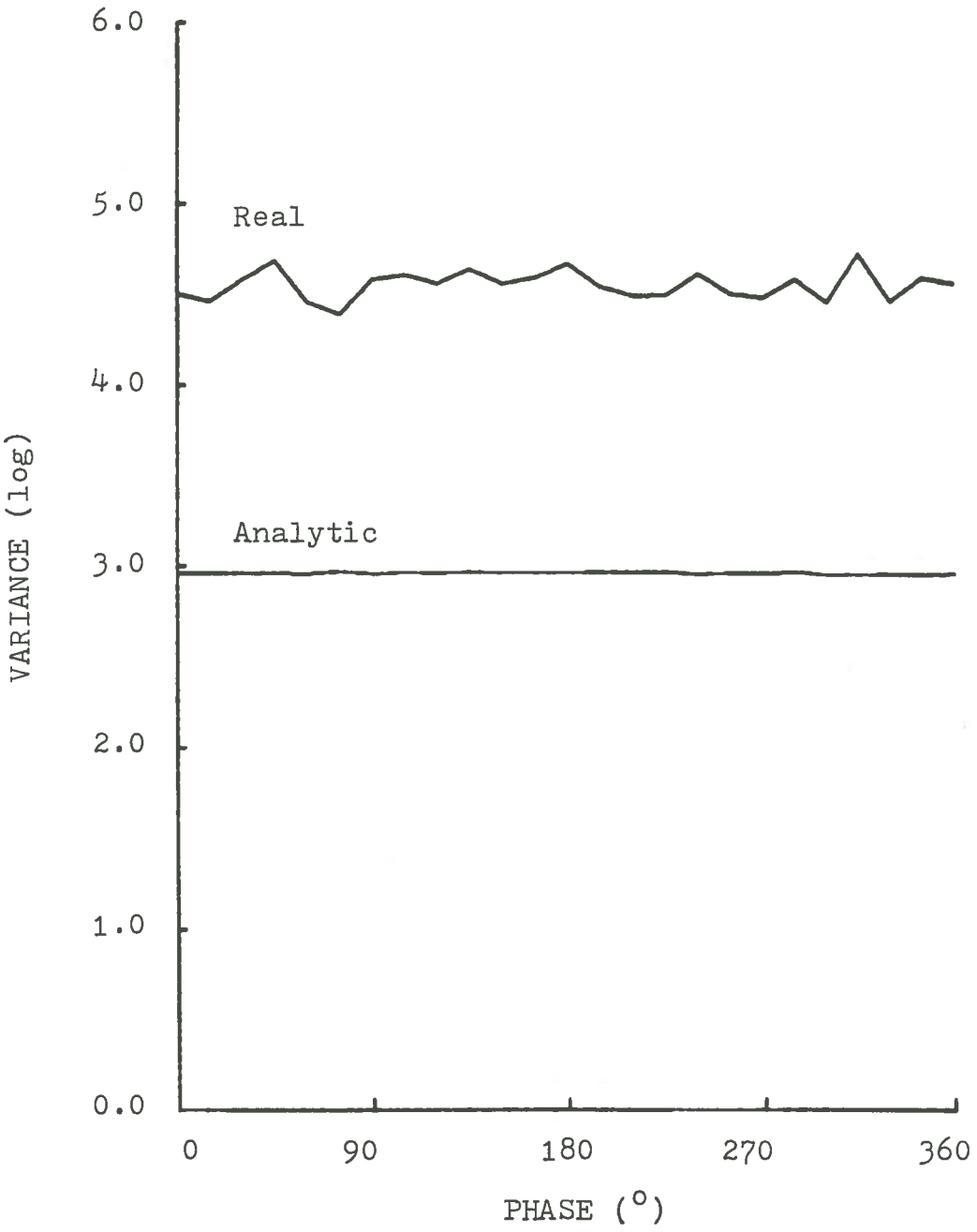


Figure 27. Dependence of the Variance of the Frequency Estimates on Phase (SNR = 40 dB)

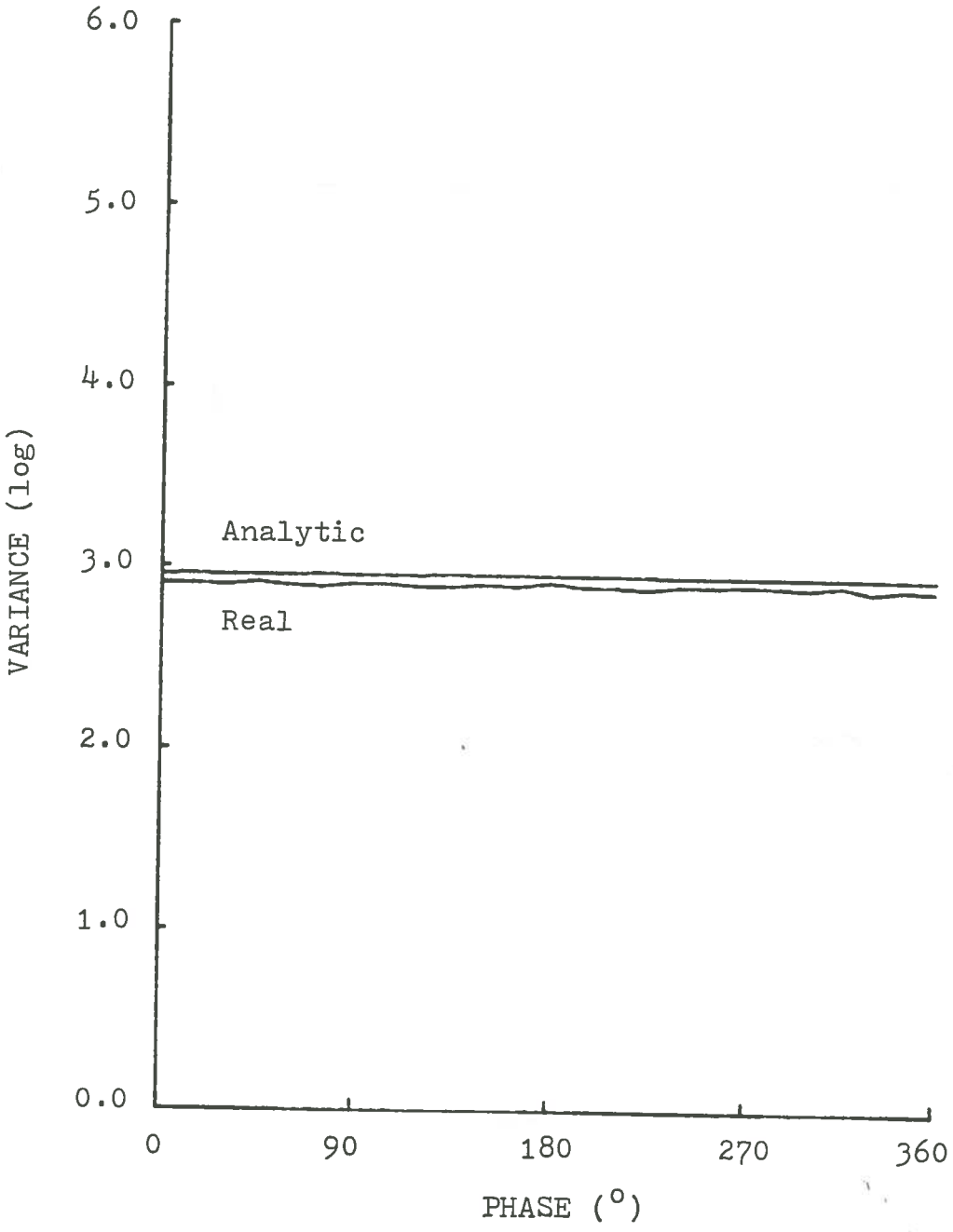


Figure 28. Dependence of the Variance of the Frequency Estimates on Phase (SNR = 60 dB)

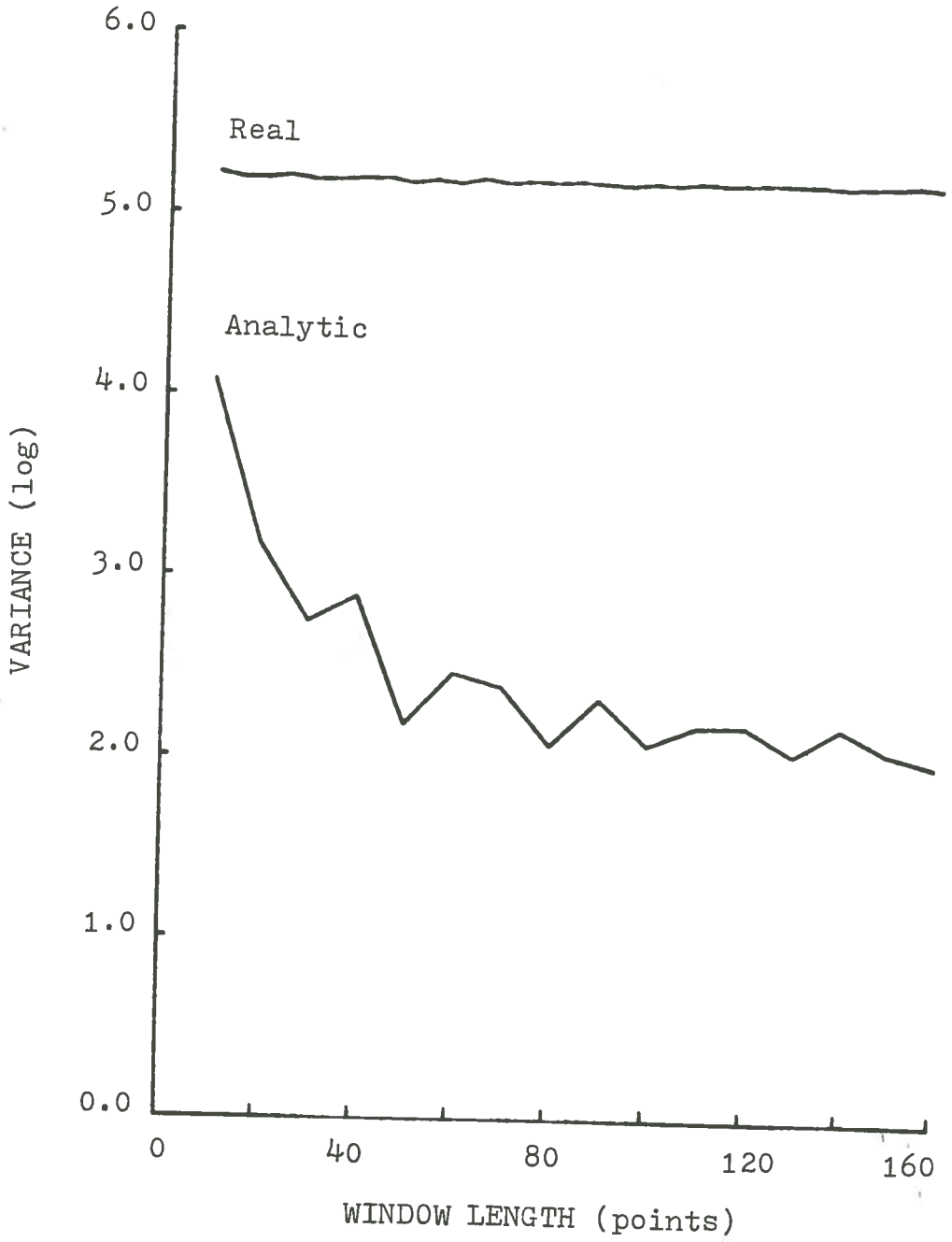


Figure 29. Dependence of the Variance on Window Length for the Autocorrelation Method (SNR = 20 dB)

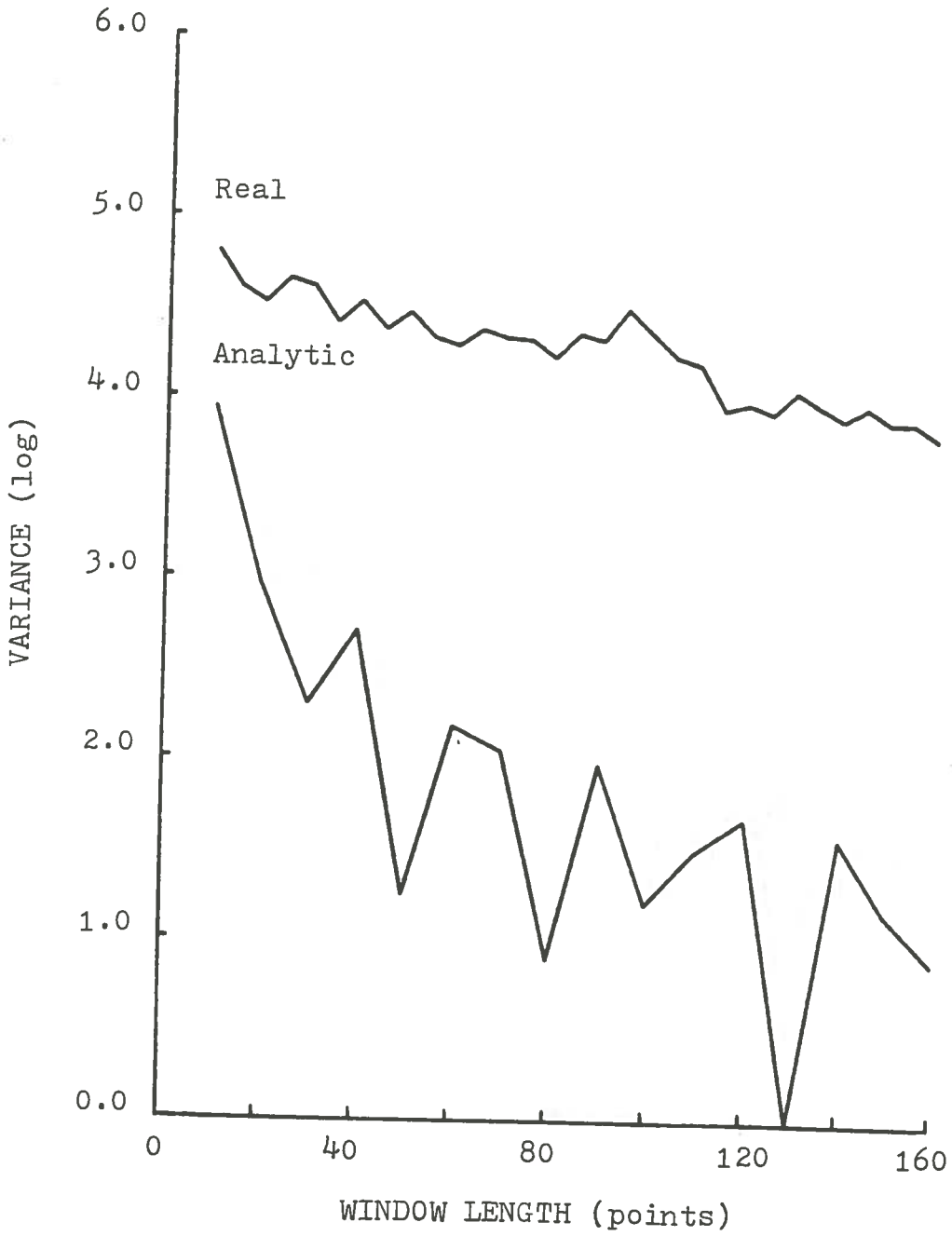


Figure 30. Dependence of the Variance on Window Length for the Autocorrelation Method (SNR = 40 dB)

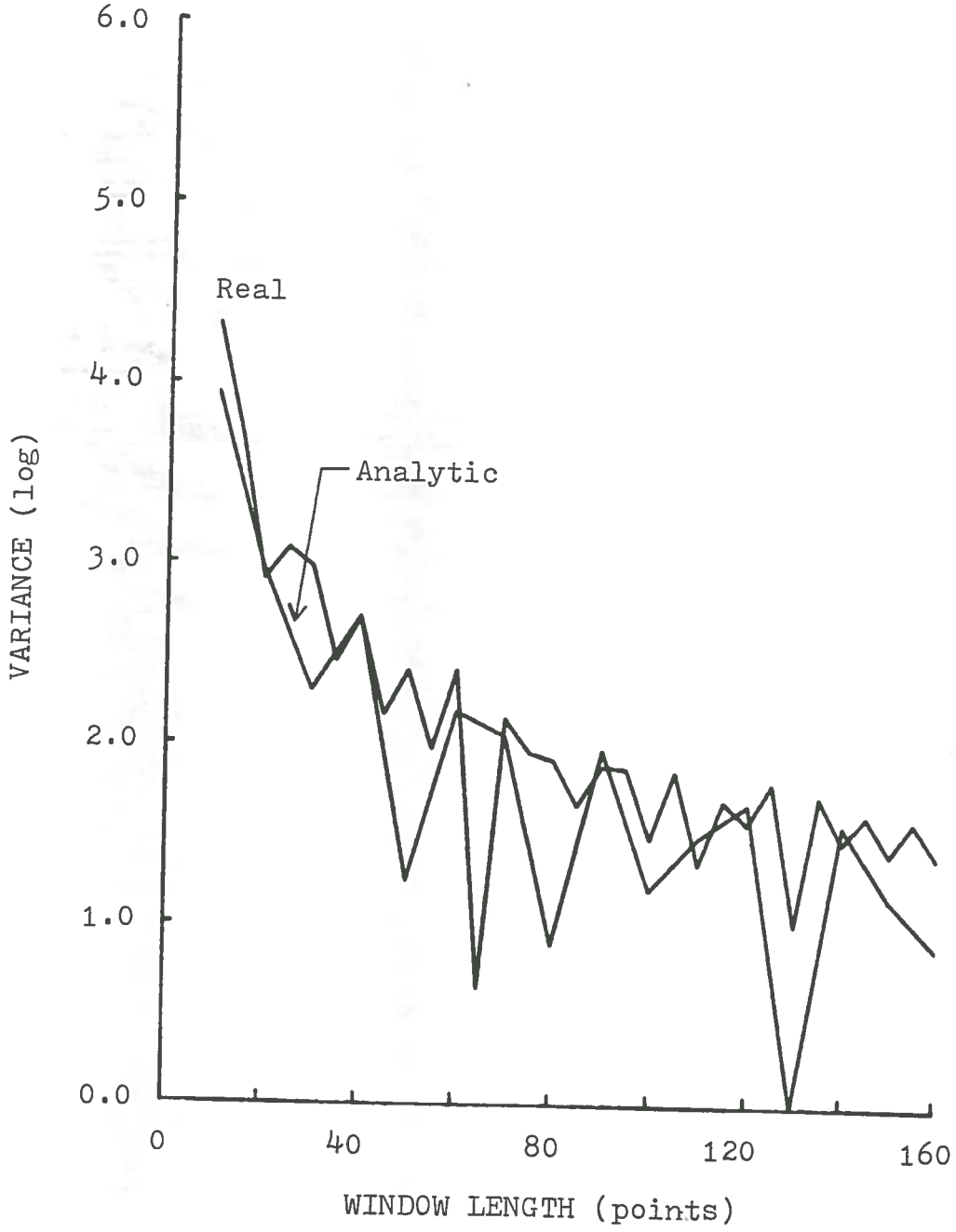


Figure 31. Dependence of the Variance on Window Length for the Autocorrelation Method (SNR = 60 dB)

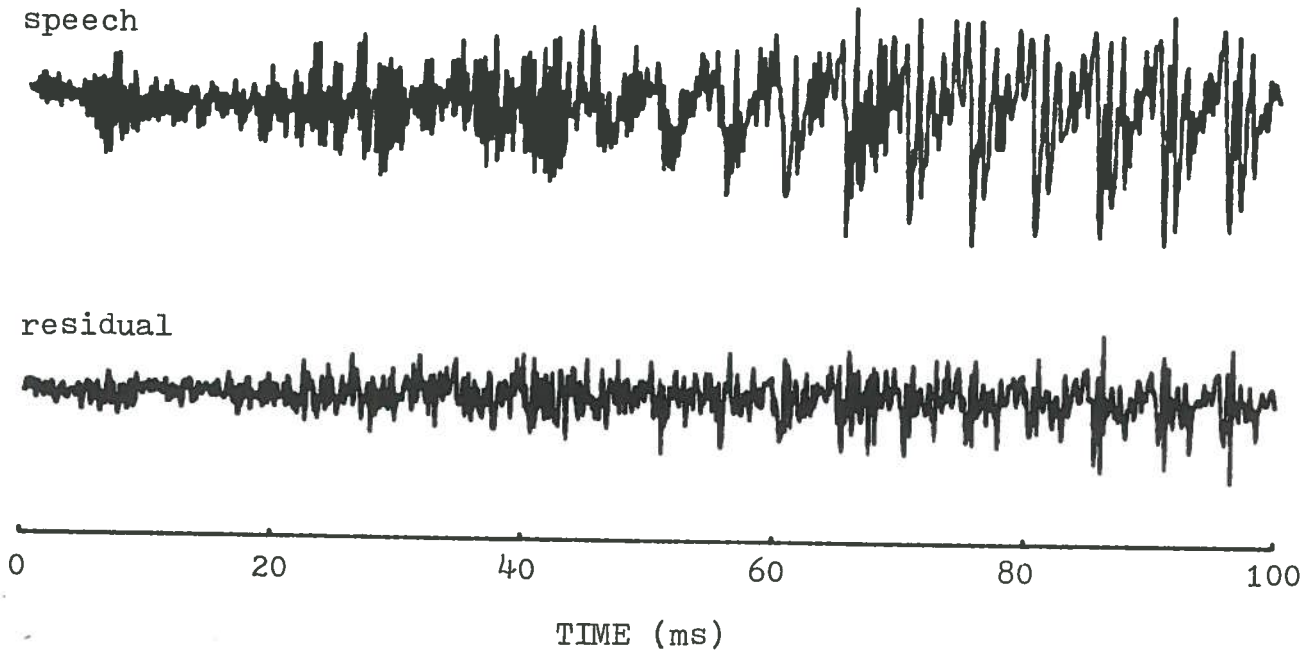


Figure 32. Irregularly Spaced Pitch Pulses at the Onset of a Voiced Interval

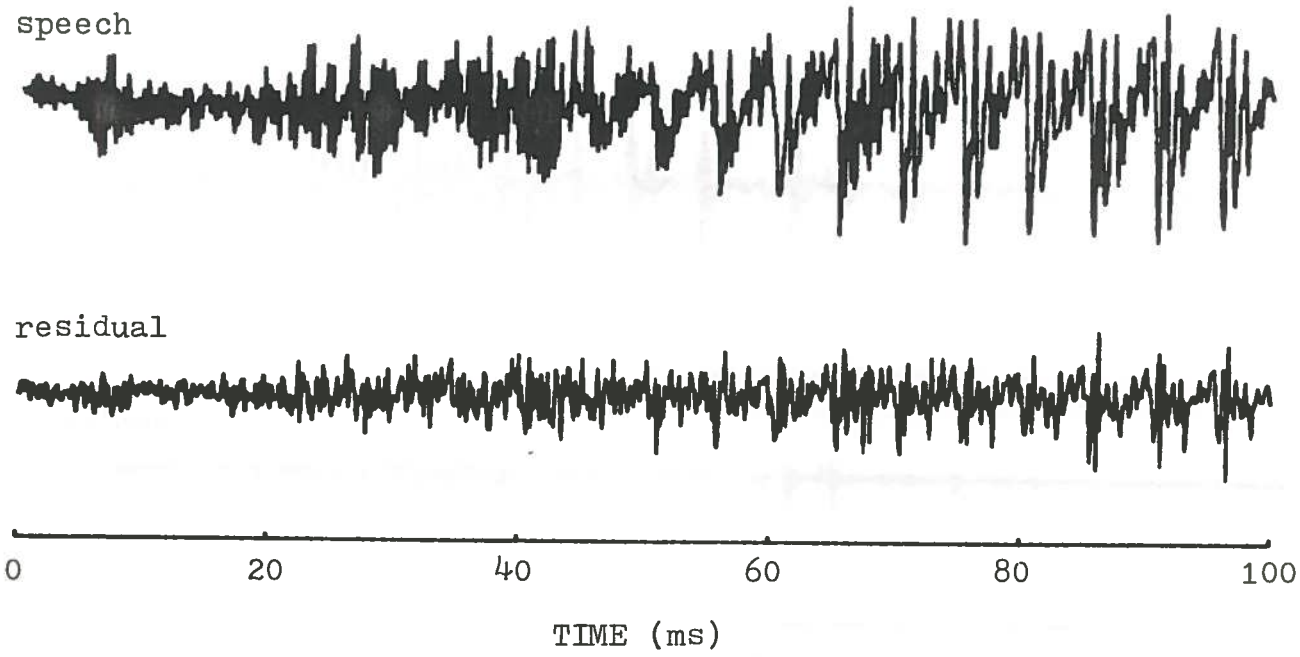


Figure 32. Irregularly Spaced Pitch Pulses at the Onset of a Voiced Interval

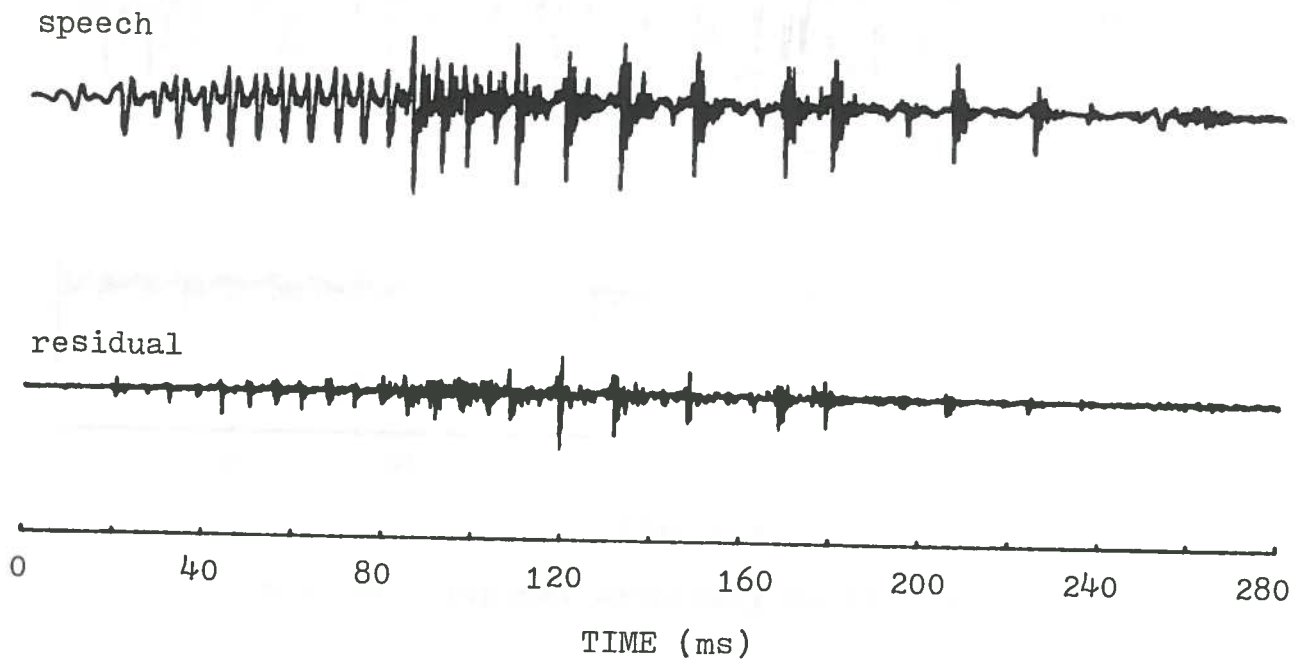


Figure 33. Discontinuous Pitch Period Behavior

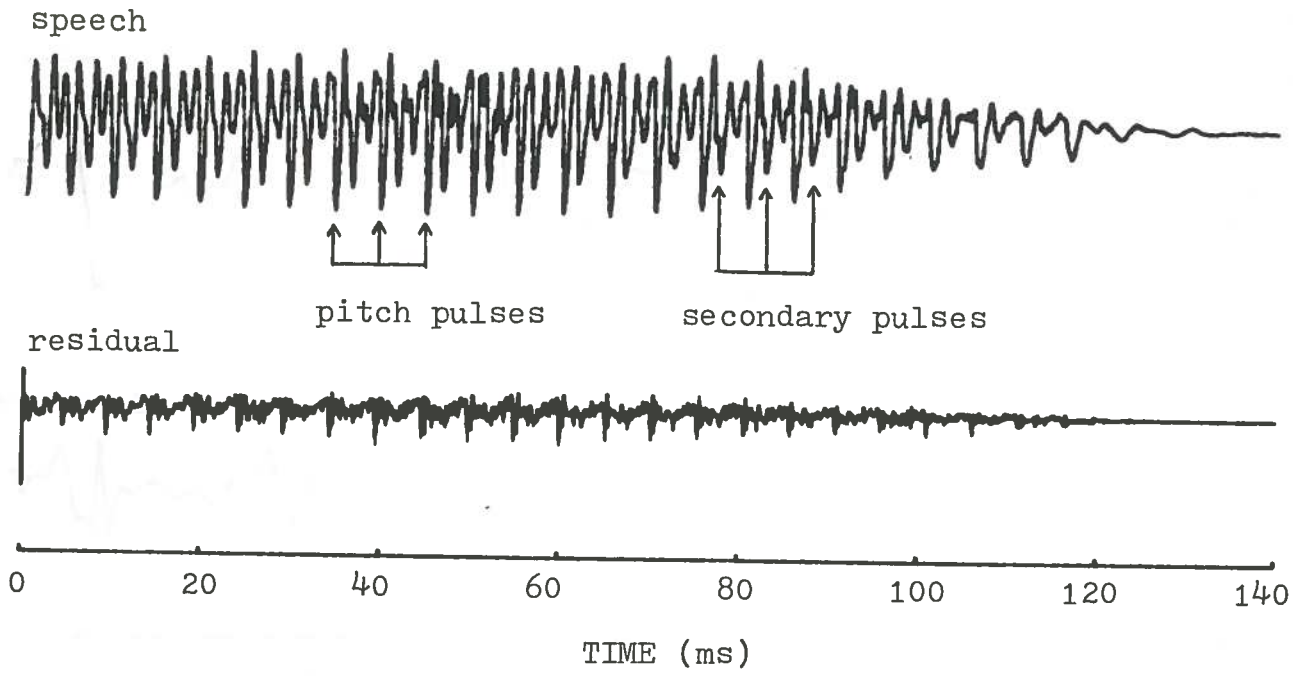


Figure 34. Typical Secondary Excitation Pulses

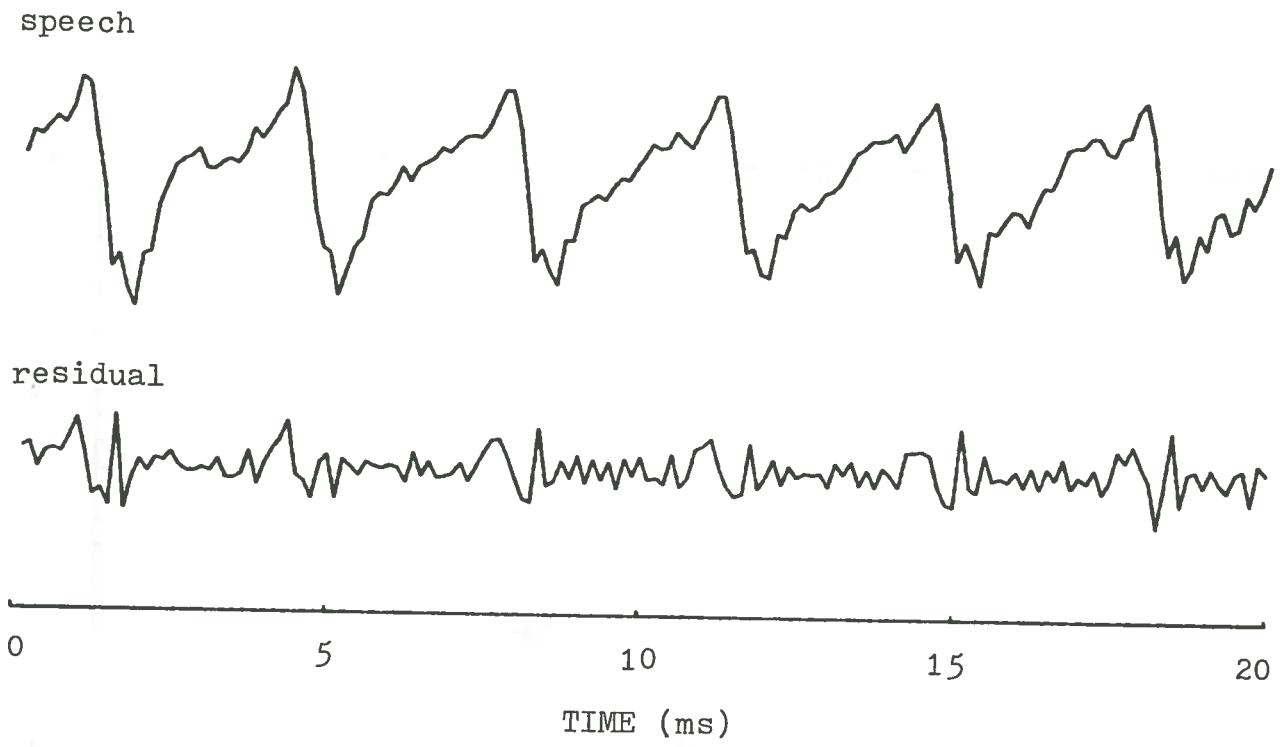


Figure 35. A Female Speech Signal and Its Residual

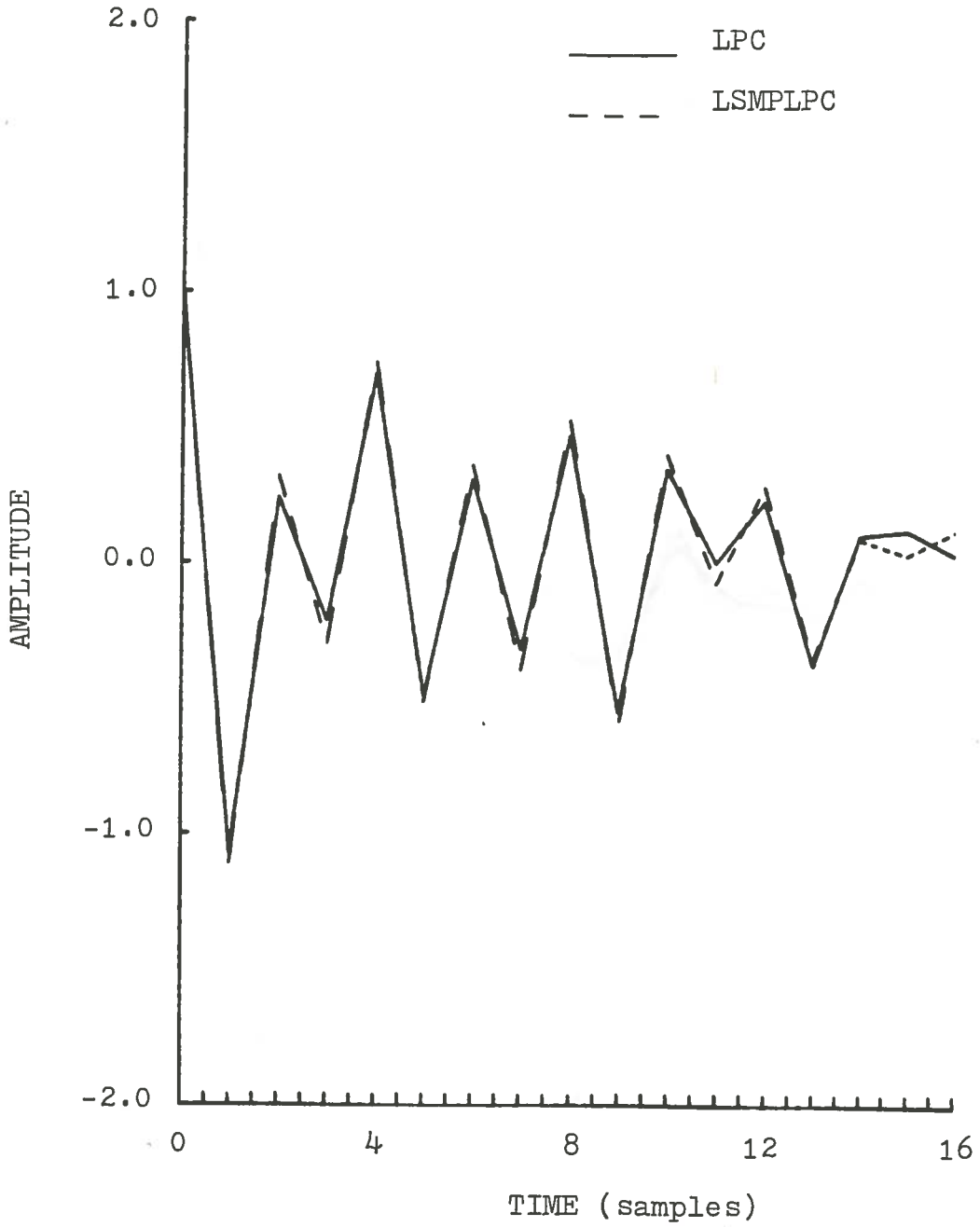


Figure 36. Comparison of LPC and LSMPLPC Impulse Responses

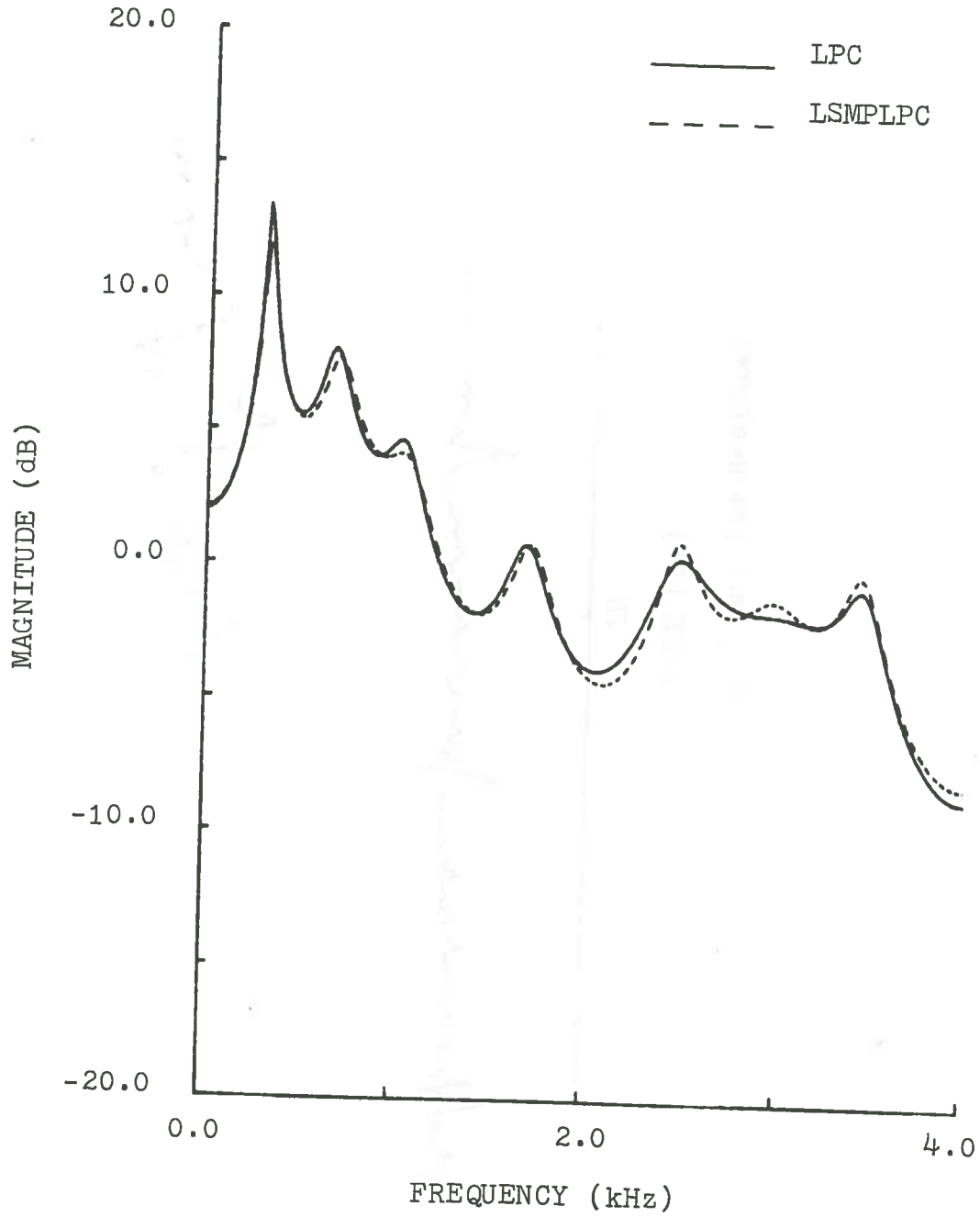


Figure 37. Comparison of LPC and LSMPLPC Spectra

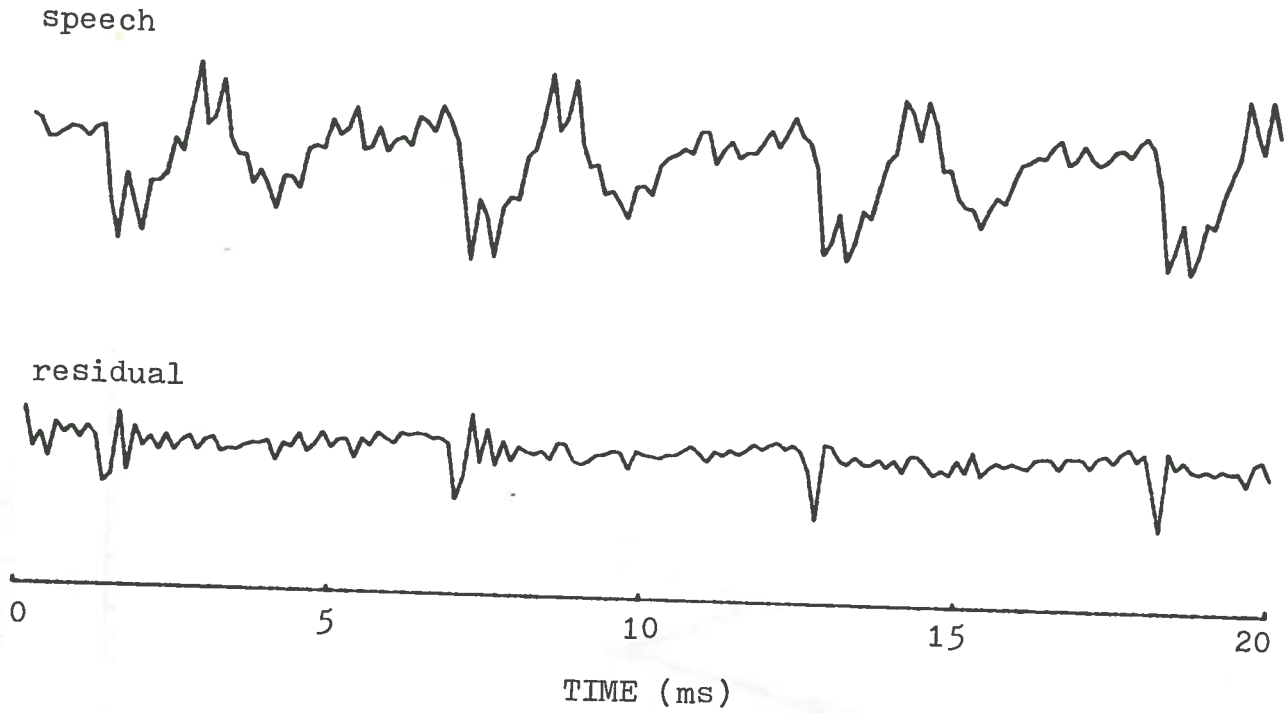


Figure 38. A Male Speech Signal and Its Residual

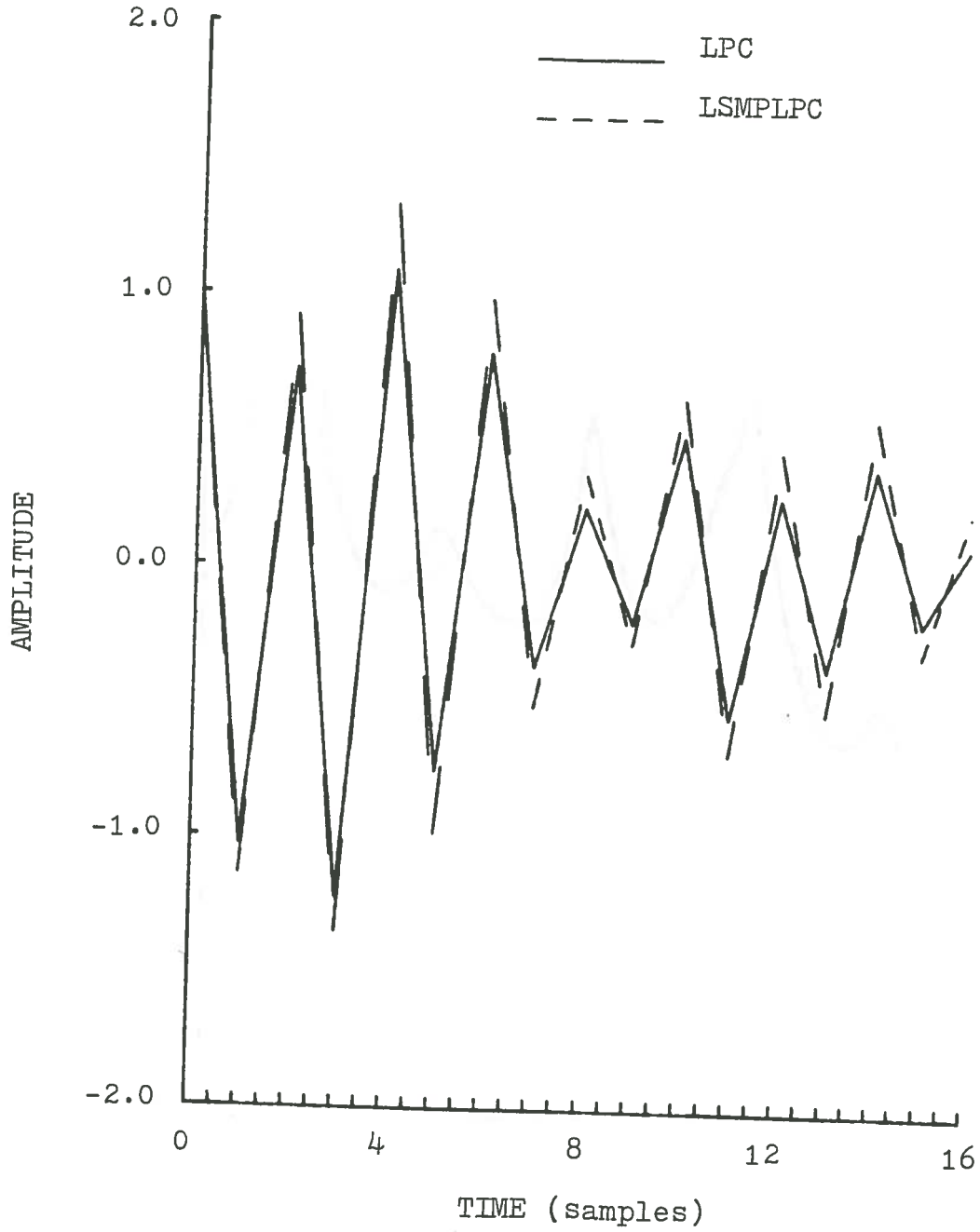


Figure 39. Comparison of LPC and LSMPLPC Impulse Responses

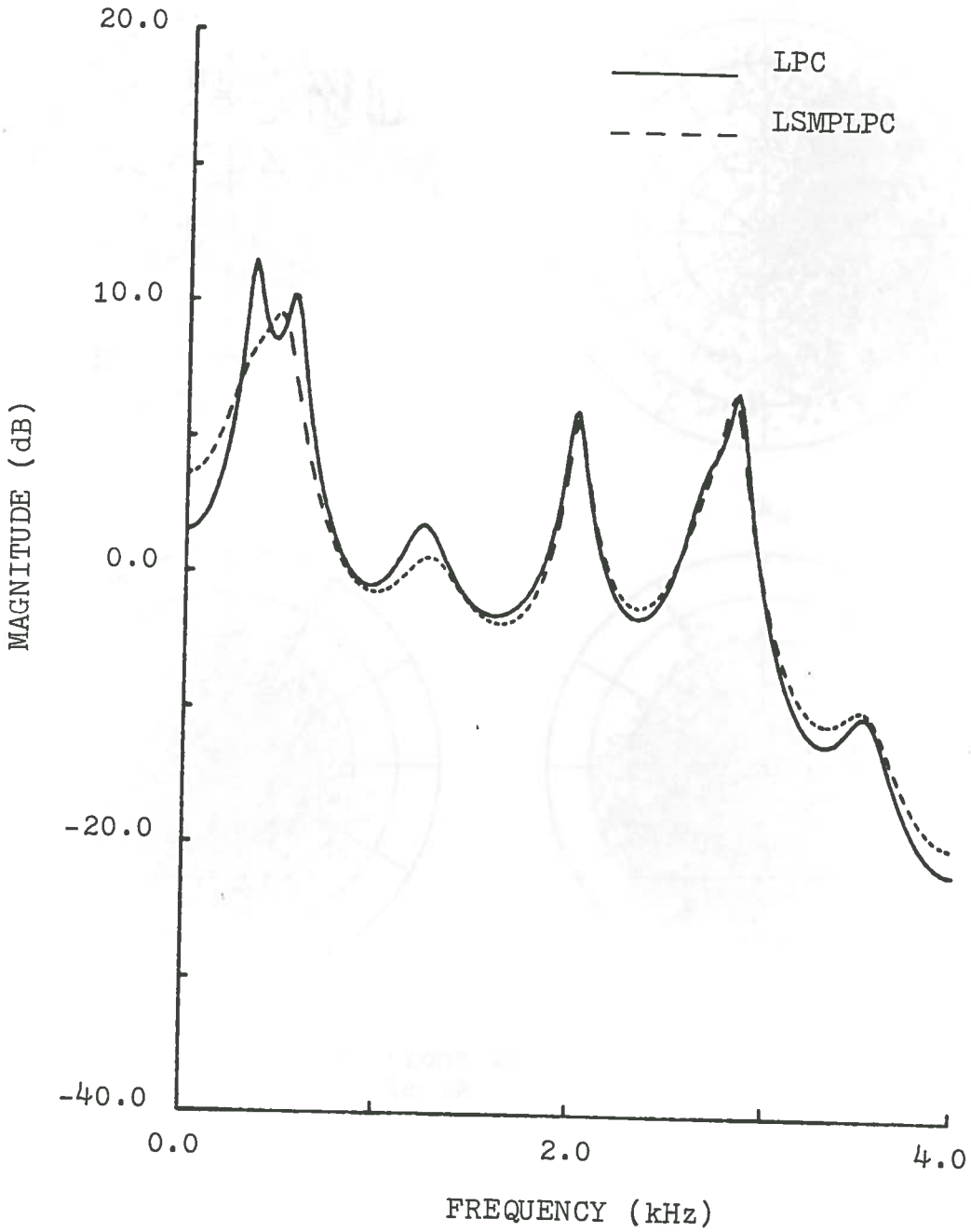


Figure 40. Comparison of LPC and LSMPLPC Spectra

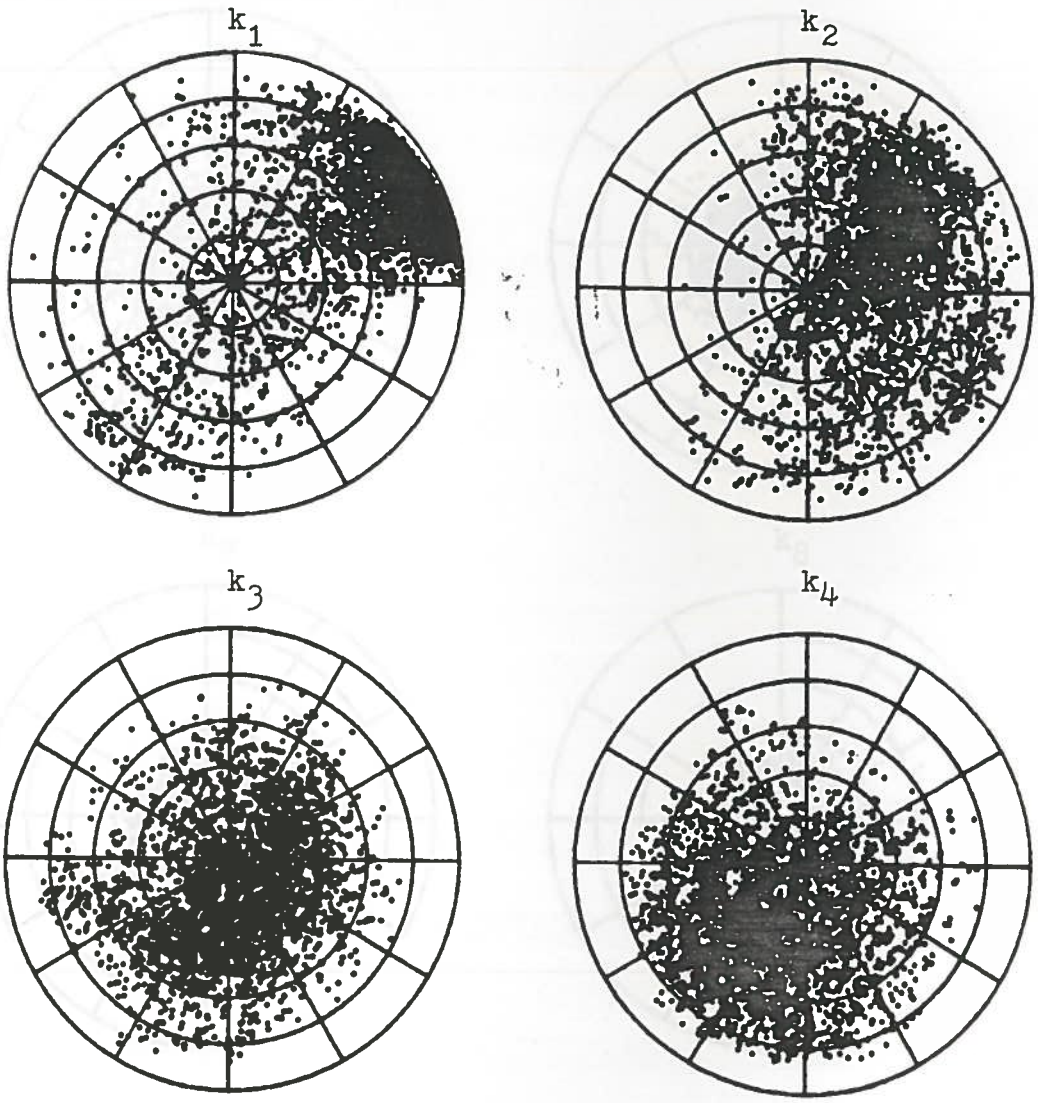


Figure 41. Distributions of the First Four Complex Reflection Coefficients

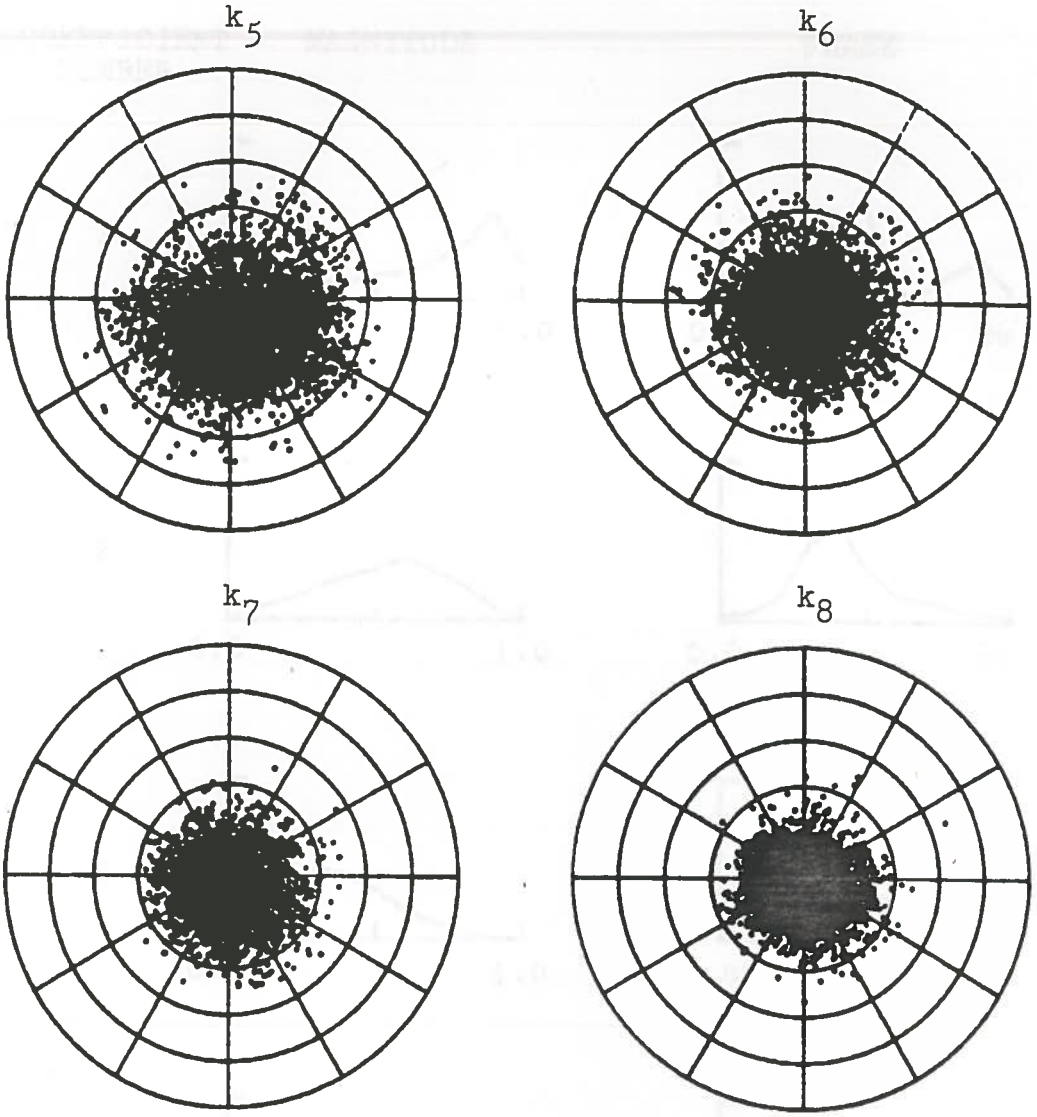


Figure 42. Distributions of the Second Four Complex Reflection Coefficients

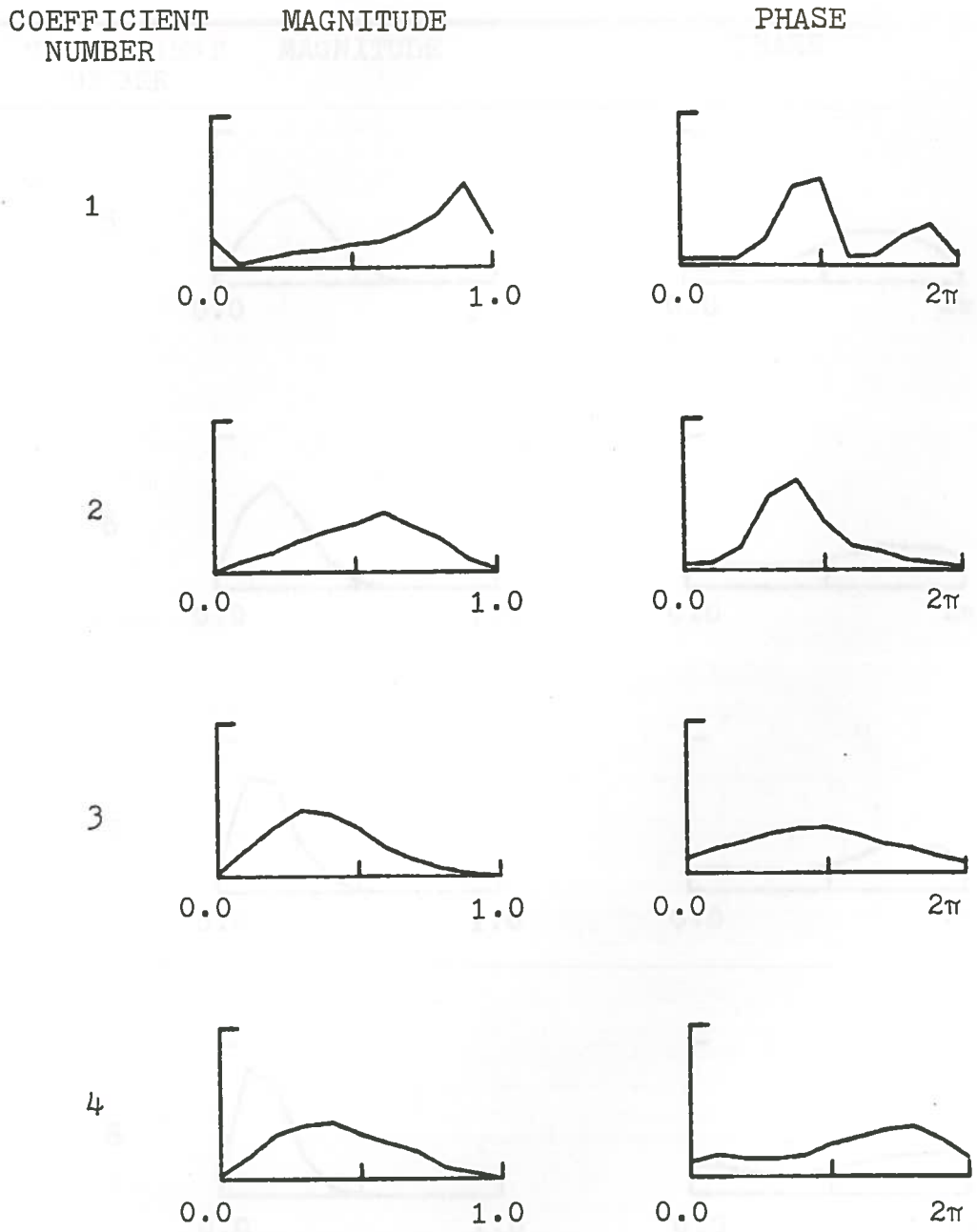


Figure 43. Magnitude/Phase Distributions of the First Four Complex Reflection Coefficients

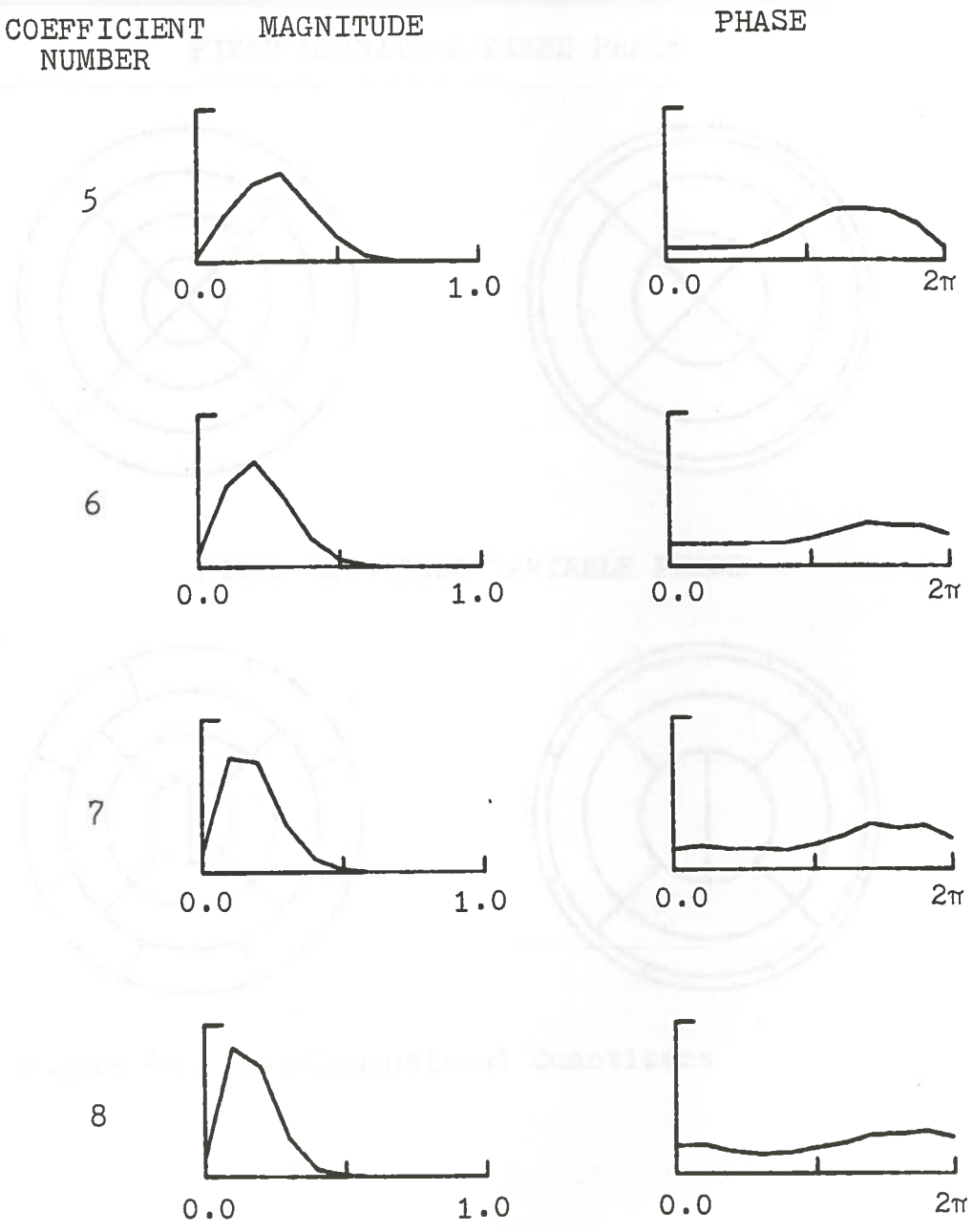
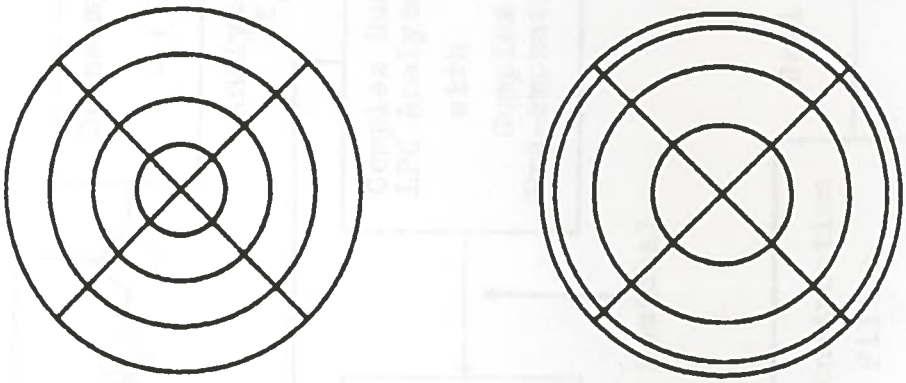


Figure 44. Magnitude/Phase Distributions of the Second Four Complex Reflection Coefficients

FIXED MAGNITUDE/FIXED PHASE



FIXED MAGNITUDE/VARIABLE PHASE

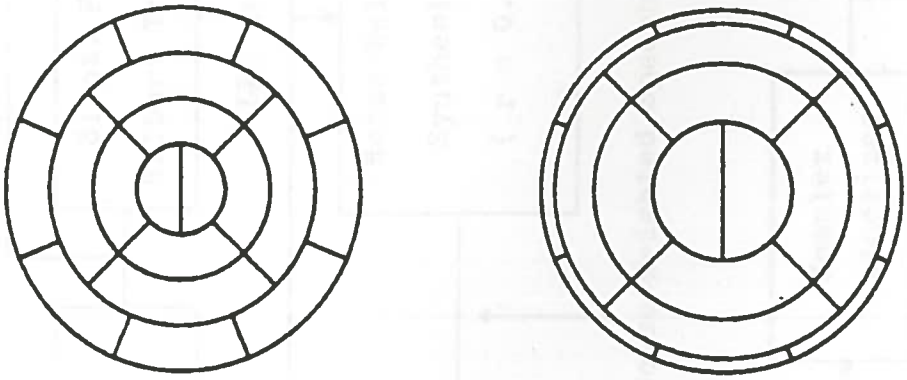


Figure 45. Two-Dimensional Quantizers

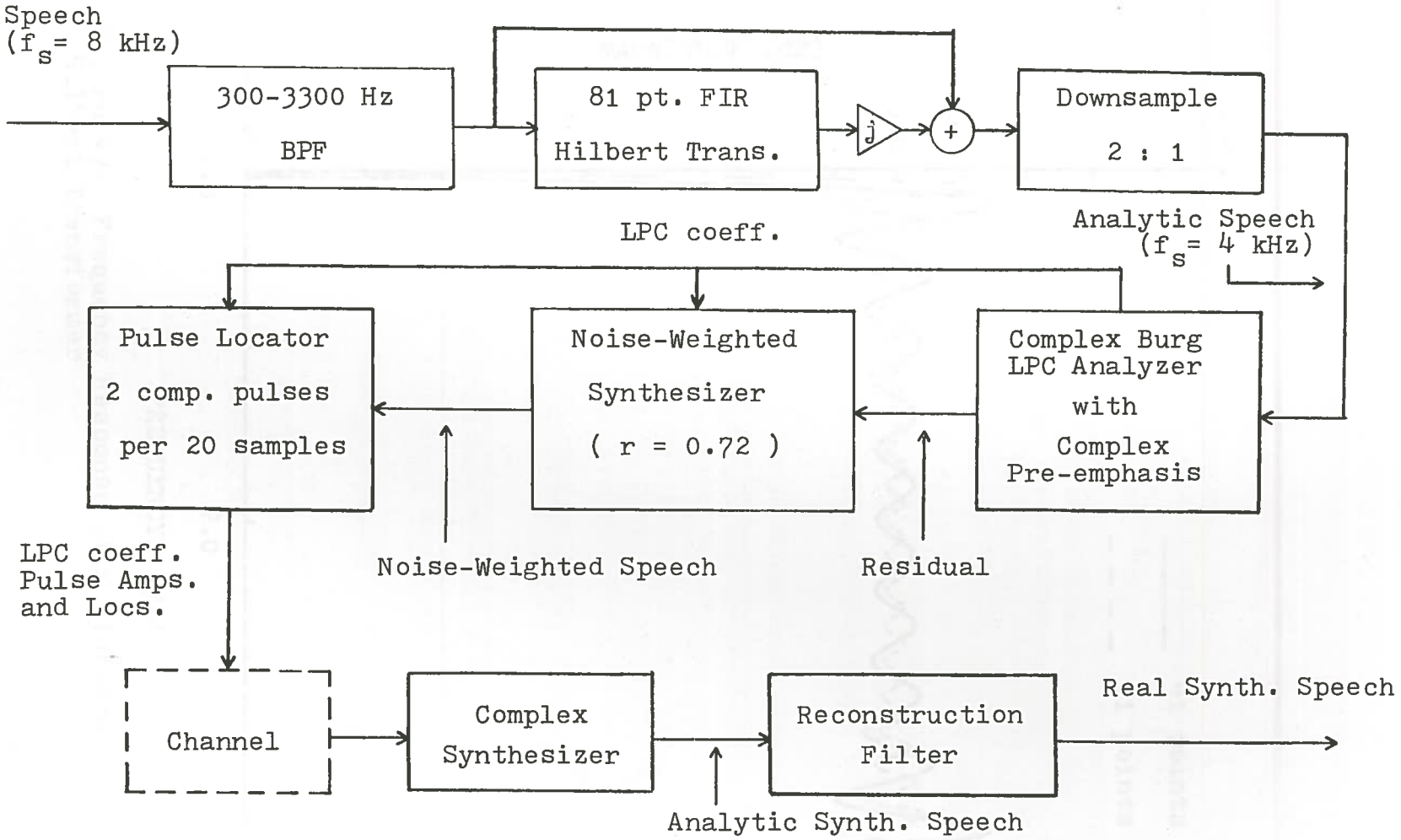


Figure 46. The AMPLPC System

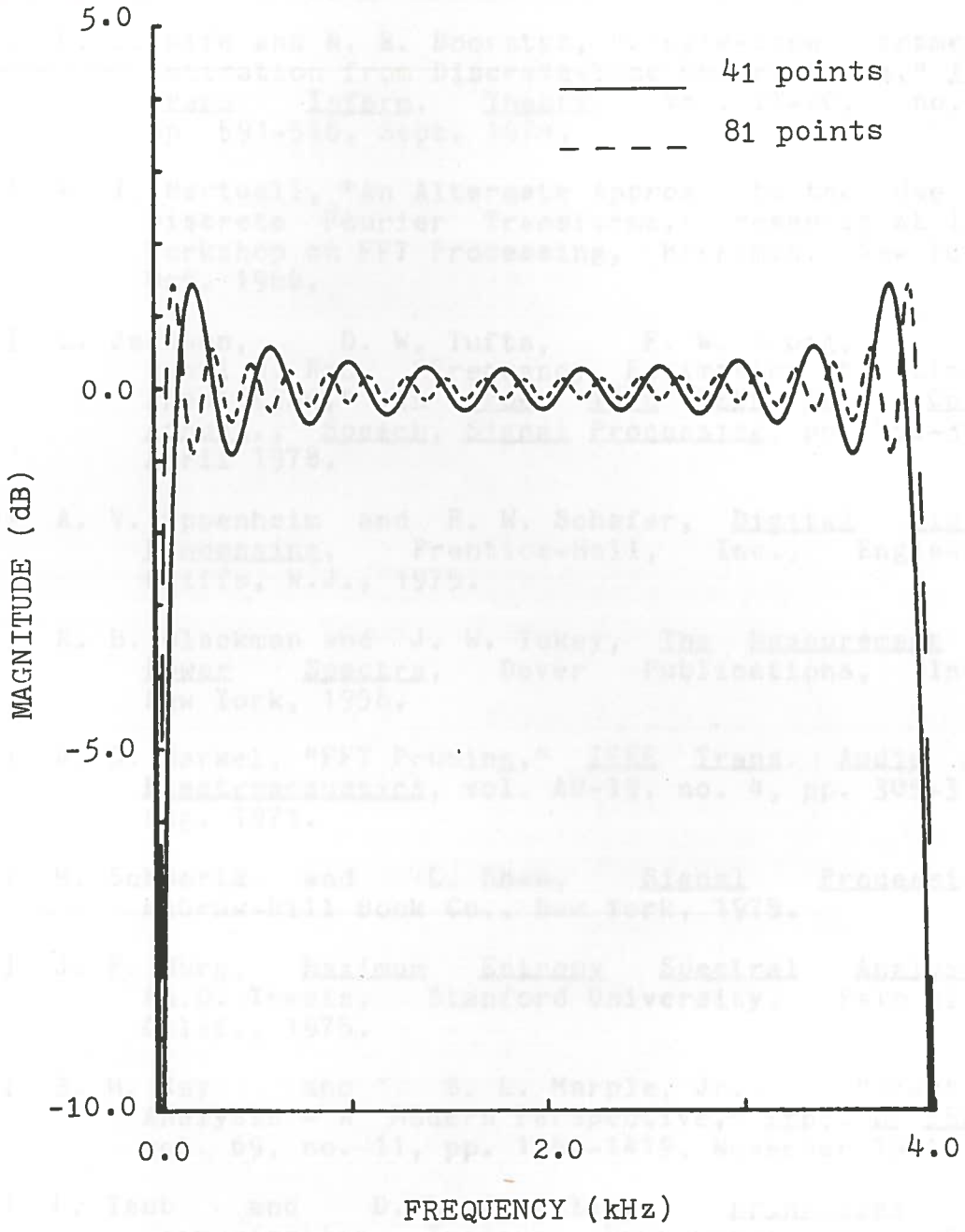


Figure 47. Frequency Response of a Discrete Hilbert Transformer

REFERENCES

[1] J. F. Claerbout, Fundamentals of Geophysicalp Processing, McGraw-Hill Book Co., New York, 1976.

[2] D. C. Rife and R. R. Boorstyn, "Single-Tone Parameter Estimation from Discrete-Time Observations," IEEE Trans. Inform. Theory, vol. IT-20, no. 5, pp. 591-598, Sept. 1974.

[3] W. T. Hartwell, "An Alternate Approach to the Use of Discrete Fourier Transforms," presented at IEEE Workshop on FFT Processing, Harriman, New York, Oct. 1968.

[4] L. Jackson, D. W. Tufts, F. W. Soong, and Rahul M. Rao, "Frequency Estimation by Linear Prediction," in Proc. 1978 IEEE Int. Conf. Acoust., Speech, Signal Processing, pp. 352-356, April 1978.

[5] A. V. Oppenheim and R. W. Schafer, Digital Signal Processing, Prentice-Hall, Inc., Englewood Cliffs, N.J., 1975.

[6] R. B. Blackman and J. W. Tukey, The Measurement of Power Spectra, Dover Publications, Inc., New York, 1958.

[7] J. D. Markel, "FFT Pruning," IEEE Trans. Audio and Electroacoustics, vol. AU-19, no. 4, pp. 305-311, Aug. 1971.

[8] M. Schwartz and L. Shaw, Signal Processing, McGraw-Hill Book Co., New York, 1975.

[9] J. P. Burg, Maximum Entropy Spectral Analysis, Ph.D. Thesis, Stanford University, Palo Alto, Calif., 1975.

[10] S. M. Kay and S. L. Marple, Jr., "Spectrum Analysis - A Modern Perspective," Proc. of IEEE, vol. 69, no. 11, pp. 1380-1419, November 1981.

[11] H. Taub and D. L. Schilling, Principles of Communication Systems, McGraw-Hill Book Co., New York, 1971.

[12] F. G. Stremler, Introduction to Communication Systems, Addison-Wesley Pub. Co., Reading, Mass., 1982.

[13] R. Bracewell, The Fourier Transform and Its Applications, McGraw-Hill Book Co., New York, 1965.

[14] D. H. Johnson, "The Application of Spectral Estimation Methods to Bearing Estimation Problems," Proc. IEEE, vol. 70, no. 9, pp. 1018-1029, Sept. 1982.

[15] F. E. Soong and A. M. Peterson, "Fast Spectral Estimation of a Speech Signal in Analytic Form," in Proc. 1980 IEEE Int. Conf. Acoust., Speech, Signal Processing, pp. 158-160, May 1978.

[16] S. M. Kay, "Maximum Entropy Spectral Estimation Using the Analytic Signal," IEEE Trans., Acoustics, Speech, Signal Processing, vol. ASSP-26, no. 5, pp. 467-469, Oct. 1980.

[17] T. V. Ananthapadmanabha and B. Yegnanarayana, "Epoch Extraction from Linear Prediction Residual for Identification of Closed Glotts Interval," IEEE Trans. Acoustics, Speech, Signal Processing, vol. ASSP-27, no. 4, pp. 309-319, Aug. 1979.

[18] B. S. Atal and J. R. Remde, "A New Model of LPC Excitation for Producing Natural-Sounding Speech at Low Bit Rates," in Proc. 1982 IEEE Int. Conf. Acoust., Speech, Signal Processing, pp. 614-617, May 1982.

[19] L. R. Rabiner and R. W. Schafer, Digital Processing of Speech Signals, Prentice-Hall, Inc., Englewood Cliffs, N.J., 1978.

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

[21] T. E. Trumain, "The Government Standard Linear Predictive Coding Algorithm: LPC-10," Speech Technology, pp. 40-49, April 1982.

[22] R. V. Cox, R. E. Crochiere, and J. D. Johnston, "Real-Time Implementation of Time Domain Harmonic Scaling of Speech for Rate Modification and Coding," IEEE Trans. Acoustics, Speech, Signal Processing (special issue on integrated circuits for speech), vol. ASSP-31, no. 1, pp. 258-273, Feb. 1983.

[23] V. R. Viswanathan, A. L. Higgins, and W. H. Russell, "Design of a Robust Baseband LPC Coder for Speech Transmission Over 9.6 kbit/s Noisy Channels," IEEE Trans. Commun. (special issue on bit rate reduction and speech interpolation), vol. COM-30, no. 4, pp. 663-674, April 1982.

[24] R. V. Churchill, J. W. Brown, and R. F. Verhey, Complex Variables and Applications, McGraw-Hill Book Co., New York, 1976.

[25] C. D. McGillem and G. R. Cooper, Continuous and Discrete Signal and System Analysis, Holt, Rinehart, and Winston, Inc., New York, 1974.

[26] D. Graupe, Identification of Systems, Krieger Publishing Company Inc., New York, 1972.

[27] B. Friedlander, "Lattice Methods for Spectral Estimation," Proc. IEEE, vol. 70, no. 9, pp. 990-1017, Sept. 1982.

[28] J. D. Markel and A. H. Gray Jr., "On Autocorrelation Equations as Applied to Speech Analysis," IEEE Trans. on Audio and Electroacoustics, vol. AU-21, no. 2, pp. 69-79, April 1973.

[29] J. Makhoul, "Stable and Efficient Lattice Methods for Linear Prediction," IEEE Trans. Acoustics, Speech, and Signal Proc., vol. ASSP-25, no. 5, pp. 423-428, Oct. 1977.

[30] F. I. Itakura and S. Saito, "Analysis-Synthesis Telephony Based Upon the Maximum Likelihood Method," in Proc. 6th Int. Congress on Acoustics, pp. C17-20, Tokyo, 1968.

[31] J. P. Burg, "A New Analysis Technique for Time Series Data," NATO Advanced Study Institute on Signal Processing with Emphasis on Underwater Acoustics, Enschede, The Netherlands, Aug. 1968.

[32] A. Papoulis, Probability, Random Variables, and Stochastic Processes, McGraw-Hill Book Co., New York, 1965.

[33] N. Balabanian and T. A. Bickart, Electrical Network Theory, John Wiley and Sons, Inc., New York, 1969.

[34] O. Grenander and G. Szego, Toeplitz Forms and Their Applications, Univ. California Press, Berkeley, Calif., 1958.

[35] J. Durbin, "The Fitting of Time Series Models," Rev. Inst. Int. de Stat., vol. 28, no. 2, pp. 233-244, 1960.

[36] J. Makhoul, "Linear Prediction: A Tutorial Review," Proc. IEEE, vol. 63, no. 4, pp. 561-580, April 1975.

[37] A. H. Gray and D. Y. Wong, "The Burg Algorithm for LPC Speech Analysis/Synthesis," IEEE Trans. Acoustics, Speech, Signal Processing, vol. ASSP-28, no. 6, pp. 609-615, Dec. 1980.

[38] R. W. Herring, "The Cause of Line Splitting in Burg Maximum-Entropy Spectral Analysis," IEEE Trans., Acoustics, Speech, Signal Processing, vol. ASSP-28, no. 6, pp. 692-701, Dec. 1980.

[39] G. Winham and K. Steiglitz, "Input Generators for Digital Sound Synthesis Studies," J. Acoust. Soc. Am., vol. 47, no. 2, pp. 665-666, Feb. 1970.

[40] B. S. Atal and S. L. Hanauer, "Speech Analysis and Synthesis by Linear Prediction of the Speech Wave," J. Acoust. Soc. Am., vol. 50, no. 2 (Part 2), pp. 637-655, Aug. 1971.

[41] G. Fant, Acoustic Theory of Speech Production, Mouton, The Hague, 1970.

[42] D. W. Tufts and R. Kumaresan, "Estimation of Frequencies of Multiple Sinusoids: Making Linear Prediction Perform Like Maximum Likelihood," Proc. IEEE, vol. 70, no. 9, pp. 975-989, Sept. 1982.

[43] H. L. Van Trees, Detection, Estimation, and Modulation Theory, John Wiley and Sons, Inc., New York, 1968.

[44] L. Jackson, D. W. Tufts, F. W. Soong, and Rahul M. Rao, "Frequency Estimation by Linear Prediction," in Proc. 1978 IEEE Int. Conf. Acoust., Speech, Signal Processing, pp. 352-356, April 1978.

[45] B. S. Atal, M. R. Schroeder, and V. Stover, "Voice-Excited Predictive Coding System for Low Bit Rate Transmission of Speech," in Proc. 1975 IEEE Int. Conf. Commun., pp. 30-37 to 30-40, June 1975.

[46] B. S. Atal, "On Determining Partial Correlation Coefficients by the Covariance Method of Linear Prediction," J. Acoust. Soc. Amer., vol. 62, suppl. 1, pp. S64, Fall 1977.

[47] S. S. Singhal and B. S. Atal, "Optimizing LPC Filter Parameters For Multi-Pulse Excitation," in Proc. 1983 IEEE Int. Conf. Acoust., Speech, Signal Processing, pp. 781-784, April 1983.

[48] D. T. L. Lee, M. Morf, and B. Friedlander, "Recursive Least Squares Ladder Estimation Algorithms," IEEE Trans. Acoust., Speech, Signal Processing, vol. ASSP-29,no. 2, pp. 627-641, June 1981.

[49] J. L. Flanagan, Speech Analysis, Synthesis, and Perception, Springer-Verlag, New York, 1972.

[50] R. Sugarman and P. Wallich, "The Limits to Simulation," IEEE Spectrum, vol. 20, no. 4, pp. 36-41, April 1983.

[51] J. L. Flanagan, K. Ishizaka, K. L. Shipley, "Signal Models for Low Bit Rate Coding of Speech," J. Acoust. Soc. Am., vol. 68, no. 3, pp. 780-791, Sept. 1980.

[52] A. E. Rosenberg, "Effect of Glottal Pulse Shape on the Quality of Natural Vowels," J. Acoust. Soc. Am., vol. 49, no. 2, Part II, pp. 583-588, Sept. 1971.

[53] J. N. Holmes, "Formant Excitation Before and After Glottal Closure," in Proc. 1976 IEEE Int. Conf. Acoust., Speech, Signal Processing, pp. 39-42, April 1976.

[54] M. R. Shroeder, B. S. Atal, and J. L. Hall, "Optimizing Digital Speech Coders by Exploiting Masking Properties of the Human Ear," Jour. Acoust. Soc. Amer., vol. 66, no. 6, pp. 1647-1652, Dec. 1979.

[55] M. R. Schroeder, B. S. Atal, and J. L. Hall, "Objective Measure of Certain Speech Signal Degradations Based on Properties of Human Auditory Perception," Frontiers of Speech Communication Research, edited by B. Lindblom and S. Ohman, Academic Press, London, pp. 217-229, 1979.

[56] N. C. Gallagher, "Quantizing Schemes for the Discrete Fourier Transform of a Random Time-Series," IEEE Trans. Inform. Theory, vol. IT-24, no. 2, pp. 156-163, March 1978.

[57] A. P. Sage and C. C. White, III, Optimum Systems Control, Prentice-Hall, Inc., Englewood Cliffs, N.J., 1977.

[58] S. P. Lloyd, "Least Squares Quantization in PCM," IEEE Trans. Inform. Theory, vol. IT-28, no. 2, March 1982.

[59] J. Max, "Quantizing for Minimum Distortion," IRE Trans. Inform. Theory, vol. IT-6, no. 1, pp. 7-12, 1960.

[60] Y. Linde, A. B. Buzo, and R. M. Gray, "An Algorithm for Vector Quantizer Design," IEEE Trans. Commun., vol. COM-28, no. 1, Jan. 1980.

[61] "IEEE Recommended Practice for Speech Quality Measurements," IEEE Trans. Audio Electroacoust., vol. AU-17, no. 5, pp. 225-246, Sept. 1969.

[62] W. R. Daumer, "Subjective Evaluation of Several Efficient Speech Coders," IEEE Trans. Commun., vol. COM-30, no. 4, pp. 655-663, April 1982.