REAL-TIME COMPOSITE SIGNAL DECOMPOSITION

By

DAVID PRESTON SKINNER

A DISSERTATION PRESENTED TO THE GRADUATE COUNCIL
OF THE UNIVERSITY OF FLORIDA IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF DOCTOR OF PHILOSOPHY

UNIVERSITY OF FLORIDA

1974

Dedicated to Betty because I love her.

ACKNOWLEDGEMENTS

The author wishes to thank Dr. D.G. Childers, chairman of his supervisory committee, for his guidance during the preparation of this dissertation. The example set by him, both intellectually and personally, has been a highlight of this author's experience at the University of Florida. The author also wishes to express his gratitude to the other members of his supervisory committee for their assistance.

Finally, the author would like to express his appreciation to his parents, Mr. and Mrs. J.P. Skinner, for their encouragement and love throughout the years, and to his wife for her unflinching support.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Abstract of Dissertation Presented to the Graduate Council
of the University of Florida in Partial Fulfillment
of the Requirements for the Degree of Doctor of Philosophy

REAL-TIME COMPOSITE SIGNAL DECOMPOSITION

By

David Preston Skinner

December, 1974

Chairman: Donald G. Childers
Major Department: Electrical Engineering

The purpose of this research is to investigate the recovery in real-time of an unknown wavelet from a composite signal in the presence of additive noise. Specifically, the composite signal consists of a wavelet and its echoes.

Two techniques, the power and complex cepstra, are used in the wavelet recovery procedure. Historically, these techniques have been computed separately. An alternate definition of the power cepstrum is presented which closely relates the two techniques, and makes computation of the power cepstrum from the complex cepstrum trivial. This unification led to the discovery of a third cepstrum technique, the phase cepstrum, which like the power cepstrum is useful in the determination of echo epoch times.

Several problems inherent in cepstrum computation which must be overcome to achieve satisfactory wavelet recovery are examined. These

problems include linear phase terms, phase unwrapping errors, aliasing, and oversampling.

The effects of windowing (as used to reduce leakage in spectral analysis) at various stages in the wavelet recovery system are examined. In general, windowing of the input data or log spectrum is detrimental to wavelet recovery. A windowing of the complex cepstrum may result in some improvement in wavelet recovery.

The addition of zeroes to the input data record is explored. This is observed to reduce the phase unwrapping and aliasing problems.

Finally, algorithms suitable for use in a real-time wavelet recovery system are examined. A system based on the selected algorithms should be able to achieve sampling rates of around 10^5 samples/sec.

CHAPTER I

INTRODUCTION

Composite signals consisting of the convolution of two or more simple signals arise in a number of physical situations. The separation of the contributions of the components of such signals often yields important information about the underlying physical processes. Several techniques have been devised for this purpose, these include inverse filtering [1], Wiener filtering [1], decision theory [2] and the power and complex cepstrum techniques to be discussed herein. Each of these techniques is applicable under its own circumstances.

In particular the power and complex cepstrum have been shown to be invaluable when the composite signal consists of a basic wavelet convolved with a train of impulses [3]. One example of this is a composite signal consisting of a basic wavelet plus echoes as might arise in sonar or radar applications. The power cepstrum, defined as the power spectrum of the logarithm of the power spectrum of the given signal, has been shown to be successful in the determination of echo arrival times when the composite signal consists of a basic wavelet and its echoes [3]. Of course, since the power cepstrum is derived from the power spectrum there is no hope of reconstructing the original wavelet from it. The complex cepstrum technique, defined as the Fourier transform of the complex logarithm

of the Fourier transform of the given composite signal, overcomes this limitation since the phase information is preserved. The success of both techniques stems from the fact that convolution in the time domain is equivalent to multiplication in the frequency domain. Thus by taking the logarithm of the frequency domain function the contributions from the basic wavelet and impulse train may be separated into a sum. In the case of the complex cepstrum appropriate filtering techniques may then achieve separation of the components, and by performing the inverse operation (to the complex cepstrum) we may extract the basic wavelet. There is, in fact, no other technique either linear or nonlinear, which allows the extraction of an unknown wavelet from a composite signal consisting of the wavelet and its echoes.

This investigation will primarily be concerned with the cepstrum techniques as applied to echo detection and extraction. However, the results are in most cases equally applicable to the other areas in which cepstrum techniques have found applicability. Among these areas are seismology [4,5], and speech [6,7].

In the past the cepstrum techniques have been limited to non-real-time applications. It seems clear that the performance of these techniques in real-time would aid the research effort in the above-mentioned fields, and perhaps open new areas of interest. This investigation is the first into the area of real-time computation of the power and complex cepstrum, and wavelet recovery. This is a formidable problem since these techniques involve the computation of several Fourier transforms together with nonlinear operations.

The purpose of this research was:

(1) to identify and investigate the problems inherent in computation of the cepstra, and in wavelet recovery,

(2) to investigate the use of ordinary data handling techniques (windowing and the addition of zeroes) at various points in the wavelet recovery algorithm to improve the performance of these techniques both in noise free, and additive noise environments, and

(3) to select algorithms suitable for the real-time computation of the cepstra, and for real-time wavelet recovery, and finally, to estimate the possible data rate of a system based on the selected algorithms.

Chapter II reviews the power and complex cepstrum techniques. Historically these two techniques have been treated separately. An alternate definition of the power cepstrum is presented which closely relates the two techniques, and makes computation of the power cepstrum from the complex cepstrum trivial. This unification of the two techniques led to the discovery of a third cepstrum technique, the phase cepstrum, which like the power cepstrum is useful for the determination of echo epoch times. This technique is introduced and discussed. A comparison of this new technique with the complex and power cepstra for the estimation of the echo amplitudes, and epochs in the presence of noise is reported in Appendix A.

Chapter III presents some problems of cepstrum computation which must be overcome to achieve satisfactory wavelet recovery. Some of these have been reported by other authors ; others were previously

unmentioned. Problems of linear phase terms, phase unwrapping errors, aliasing, and oversampling are discussed, and methods for their alleviation are indicated.

Chapter IV analyzes (both theoretically and experimentally) the effects of windowing at various stages in the wavelet recovery process, and the effects of extending the input data record with zeroes. Considerable insight is gained into the performance of cepstrum techniques in additive noise.

Chapter V discusses the algorithms suitable for use in a real-time wavelet recovery system, and roughly estimates the performance of such a system. The computation of the power cepstrum (only) in real-time is also discussed. And, finally, Chapter VI collects and summarizes the results of the previous chapters.

CHAPTER II

THE CEPSTRA

This chapter presents a brief history of, and introduction to, the power and complex cepstrum techniques. In addition, a new cepstrum technique, the phase cepstrum, is introduced along with an alternate definition of the power cepstrum which serves to unify the cepstra, and is of computational interest.

THE POWER CEPSTRUM

History

The power cepstrum was first presented by Bogert et al. [4] in 1963 as a heuristic technique for finding the echo epoch times of a composite signal. Subsequently, A.M. Noll [6] in 1964 successfully applied the power cepstrum to speech data to determine the pitch period and for voiced-unvoiced detection. Noll also simulated a technique for short time power cepstrum computation utilizing the direct computation of the discrete Fourier transform (DFT). In 1966, Bogert and Ossanna [8] examined the statistical properties of the power cepstrum of a Gaussian signal in Gaussian noise. Halpeny [9] examined the properties of the cepstrum of a composite signal consisting of a basic wavelet plus multiple echoes in the presence of additive noise. Kemerait [3] then extended Halpeny's results and examined the effects of echo truncation on the power cepstrum

(in conjunction with his work on the complex cepstrum). Kemerait's dissertation shows the power cepstrum to be invaluable in the detection and estimation of echo amplitudes and epochs.

Introduction

Basically, the power cepstrum, usually referred to in the literature as simply the cepstrum, is just a clever way of separating the contributions of two simple signals to the power spectrum of a composite signal (which is the convolution of the two simple signals). Of course to effectively separate any two signals there must be some difference on which we may base the separation; for the power cepstrum (and analogously for the complex cepstrum) this difference is that the power spectra (more properly the logarithms of the power spectra) of the two simple signals must vary (as a function of ω) at different rates; i.e., in Tukey's terminology [4] they must occupy different ranges of quefrency. Historically the power cepstrum has been defined as the power spectrum of the logarithm of the power spectrum of the given signal. In actuality, the power spectrum of the logarithm of the power spectrum does not exist for most signals. The power cepstrum is only meaningful when defined in a sampled data sense (as is the complex cepstrum). Thus the following definition is proposed: The power cepstrum is the square of the inverse z-transform of the logarithm of the magnitude squared of the z-transform of the given sequence. Evaluated on the unit circle this definition (except for the normalization factors associated with the power spectrum) is precisely the procedure historically used for estimating the power cepstrum. Thus we may write

$$x_{pc}(nT) = (Z^{-1}(\log |X(z)|^2))^2 \qquad (2\text{-}1)$$

where $x_{pc}(nT)$ is the power cepstrum of the sequence $x(nT)$ and $X(z)$ is the z-transform of $x(nT)$. Consider the signal given below

$$y(nT) = x(nT) * e(nT) \qquad (2\text{-}2)$$

where * denotes convolution. The magnitude squared of the z-transform of the above equation is then

$$|Y(z)|^2 = |X(z)|^2 |E(z)|^2 . \qquad (2\text{-}3)$$

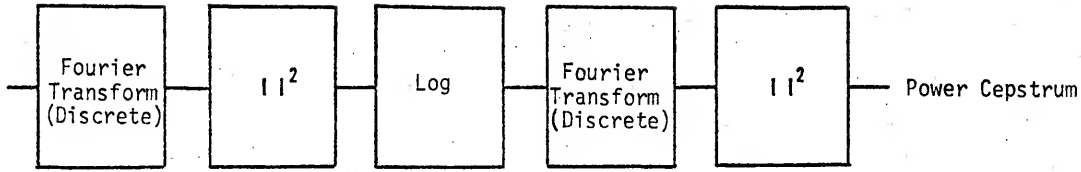
Taking the logarithm of equation (2-3), we obtain

$$\log |Y(z)|^2 = \log |X(z)|^2 + \log |E(z)|^2 . \qquad (2\text{-}4)$$

We may now return to the time domain by taking the square of the inverse z-transform of equation (2-4) with the contributions due to the two signals additively combined (provided they occupy different ranges of quefrency).

$$y_{pc}(nT) = x_{pc}(nT) + e_{pc}(nT) . \qquad (2\text{-}5)$$

Figure 1 shows the computation of the power cepstrum as outlined above. It is of interest to note that the second squaring operation will again mix the contributions from the two signals if they do not occupy distinctly different quefrency ranges (there will usually be some overlap in practice).



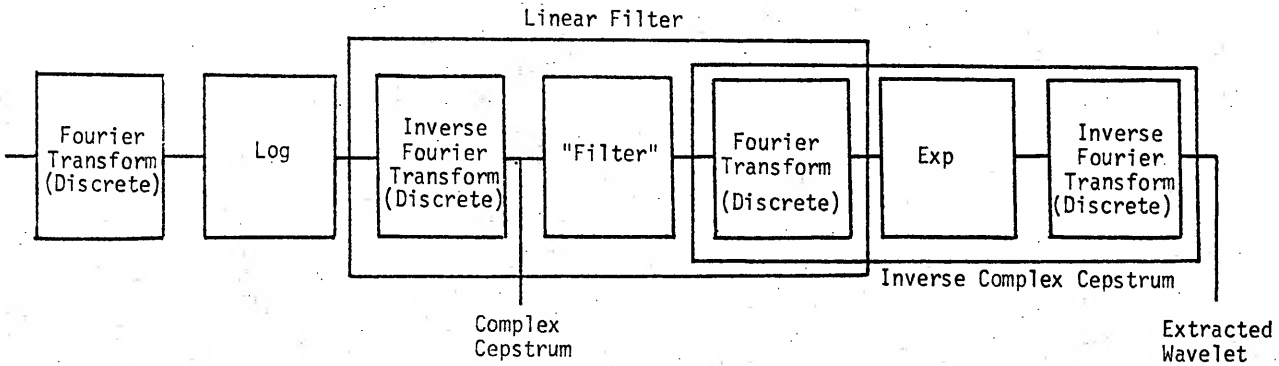


Figure 1 Computation of the Power and Complex Cepstrum.

Note that computational considerations lead to the replacement of the z-transform with the discrete Fourier transform.

To further clarify this technique consider a composite signal consisting of a basic wavelet and a single echo

$$y(nT) = x(nT) + ax(nT-n_oT) \tag{2-6}$$

$$= x(nT) * (\delta(nT) + a\delta(nT-n_oT)). \tag{2-7}$$

Taking the magnitude squared of the z-transform of equation (2-7), we have

$$|Y(z)|^2 = |X(z)|^2 \ |(1+az^{-n_o})|^2 \ . \tag{2-8}$$

Evaluating the above equations on the unit circle ($z = e^{j\omega T}$) and taking the logarithm, we obtain

$$\log |Y(e^{j\omega T})|^2 = \log |X(e^{j\omega T})|^2 + \log(1+a^2+2a\cos(\omega n_oT)) \tag{2-9}$$

$$= \log |X(e^{j\omega T})|^2 + \log(1+a^2) + \log(1+\frac{2a}{1+a^2}\cos(\omega n_oT)). \tag{2-10}$$

We may now expand the third term on the right hand side of (2-10) in a power series (except for the point values $a = \pm 1$ and $\cos \omega n_oT = \pm 1$) to obtain

$$\log (1+\frac{2a}{1+a^2}\cos(\omega n_oT)) = \sum_{m=1}^{\infty} (-1)^{m+1} (\frac{2a}{1+a^2}\cos \omega n_oT)^m/m \ . \tag{2-11}$$

Thus, the logarithm of the magnitude squared of the z-transform of the composite signal will contain cosinusoidal ripples whose amplitude and quefrency (that is, the "frequency" of the ripples) are related to the echo amplitude and delay respectively. If we take the inverse z-transform of (2-10) and square, we obtain peaks at quefrencies of

n_0T seconds (note, the units of quefrency are seconds, since (2-10) is a function of frequency the inverse z-transform brings us back into the time domain) and multiples thereof. These peaks should be detectable provided $\log|X(e^{j\omega T})|^2$ is approximately quefrency limited to less than n_0T, so as not to obscure the peaks. Thus ripples in $\log|X(e^{j\omega T})|^2$ should have a ripple length greater than $1/n_0T$. It is now apparent that the power cepstrum is extremely useful in both echo epoch detection, and amplitude estimation. It is not obvious from the expansion of (2-11) what the relationship is between the echo amplitude a and the heights of the peaks in the power cepstrum, but it will be seen in the discussion of the relation between the power and complex cepstra that the relationship is simple. This will be further discussed in Appendix A. It should be noted that if we can remove the ripple peaks from the power cepstrum, it is possible to invert the procedure described to obtain an estimate of the power spectrum of the basic wavelet, but the basic wavelet itself can not be recovered since the phase information is discarded (the power cepstrum is totally independent of the signal phase).

THE COMPLEX CEPSTRUM AND WAVELET RECOVERY

History

The complex cepstrum technique originated as an outgrowth of the homomorphic system theory set forth by A. Oppenheim [10] in 1965. The earlier work of Bogert *et al.* [8] on the power cepstrum has also been shown to be a specific application of homomorphic system theory. The complex cepstrum technique was first presented in R.W. Shafer's doctoral dissertation [7] in which he derived the technique from homomorphic system theory, analyzed many aspects of the complex

cepstrum in detail, and used the technique in echo detection, wavelet recovery and speech analysis. In 1971, Kemerait [3] studied echo detection and wavelet recovery utilizing the complex cepstrum in the presence of additive noise, and in the case of an echo distorted by truncation.

Introduction

The primary difference between the power and complex cepstrum techniques is that the complex cepstrum technique retains the phase information of the composite signal. Thus the complex cepstrum technique may be utilized not only for echo detection, but also in wavelet recovery.

Formally we define the complex cepstrum as the inverse z-transform of the (complex) logarithm of the z-transform of a given input sequence

$$X(z) = \sum_{n=-\infty}^{\infty} x(nT)z^{-n} \tag{2-12}$$

$$\hat{X}(z) = \log X(z) \tag{2-13}$$

$$\hat{x}(nT) = \frac{1}{2\pi j} \oint_c \log(X(z))z^{n-1}dz \tag{2-14}$$

where c lies within an annular region in which log X(z) has been defined as single valued and analytic.

Let us now consider application of the complex cepstrum to a composite signal formed by convolving two simpler signals

$$y(nT) = x(nT) * e(nT) \quad . \tag{2-15}$$

Z-transforming (2-15), we obtain the familiar result

$$Y(z) = X(z)E(z) \ . \tag{2-16}$$

Now taking the logarithm of both sides of (2-16), we have

$$\hat{Y}(z) = \log Y(z) = \log X(z) + \log E(z). \tag{2-17}$$

Observe that the contributions from the two signals are now additively combined (just as for the power cepstrum), and that the phase information of the original signal is retained. We can now use familiar time domain techniques to achieve separation of the signals [3]. To complete the calculation of the complex cepstrum we inverse z-transform equation (2-17) and obtain

$$\hat{y}(nT) = \hat{x}(nT) + \hat{e}(nT) \ . \tag{2-18}$$

It should be noted that computation of the complex cepstrum is an invertible operation; thus if the log E(z) contribution can be "filtered" from $\hat{Y}(z)$, so that the complex cepstrum becomes

$$\hat{y}_R(nT) = \hat{x}(nT) \tag{2-19}$$

then we may z-transform (2-19), exponentiate the result and inverse z-transform to obtain the signal x(nT). Figure 1 shows the overall system for wavelet recovery. Note that the encircled operation may be replaced by a digital filtering operation on the log spectrum treating it as if it were a time domain function. In fact if this "filtering" operation can be accomplished by multiplying the complex cepstrum by some function of (nT) (and z-transforming the result), we may call it frequency invariant (by analogy with time-invariant systems),

since the output of the filter is just the convolution of its input with the "impulse" response.

It is of interest to note that filtering the original signal $y(nT)$ is accomplished in the quefrency domain by adding the complex cepstrum of the impulse response of the filter to the complex cepstrum of $y(nT)$. Also note that filters which are ordinarily unrealizable in the time domain (that is, the impulse response is nonzero for negative time) are implementable in this manner.

One problem present in the computation of the complex cepstrum arises from the fact that the complex logarithm is a multivalued function. If we compute the imaginary part of the logarithm modulo 2π, that is, evaluated at its principal value then discontinuities appear in the phase curve. This is clearly not allowable since the $\log (X(z))$ is the z-transform of $\hat{x}(nT)$ and thus must be analytic in some annular region of the z-plane. This problem may be rectified by making the following observations:

(1) The imaginary part of $\log X(z)$ must be continuous, and periodic (evaluated on the unit circle) as a function of ω with period $\frac{2\pi}{T}$ since it is the z-transform of $\hat{x}(nT)$.

(2) Since it is required that the complex cepstrum of a real function be real, it is required that the imaginary part of $\log (X(z))$ be an odd function of ω.

Subject to the above conditions (and provided the phase curve is sampled at a sufficient rate) we may compute an unwrapped phase curve having the correct properties with the following algorithm. Consider the phase curve (modulo 2π) shown in Figure 2(a). If we

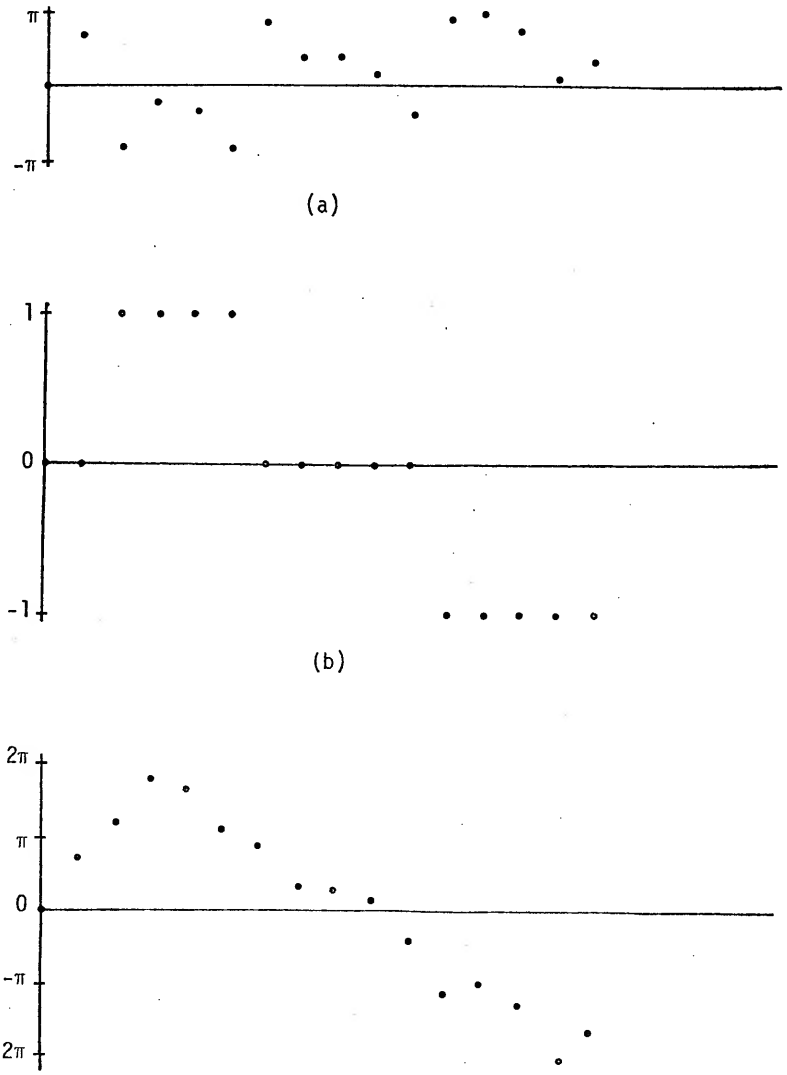


Figure 2 Phase Unwrapping. (a) phase modulo 2π, (b) c(k), the correction sequence, and (c) unwrapped phase curve.

are sampling the phase at a rate sufficient to assure that it never changes by more than π between samples then the phase may be unwrapped by adding the correction sequence C(k) to the phase modulo 2π sequence P(k) where C(k) is determined as follows

$$C(0) = 0$$
$$C(k) = C(k-1) - 2\pi \qquad \text{If } P(k)-P(k-1) > \pi$$
$$C(k) = C(k-1) + 2\pi \qquad \text{If } P(k-1)-P(k) > \pi$$

Figure 2(b) shows the correction sequence, and Figure 2(c) shows the unwrapped phase curve. Alternatively the phase may be unwrapped by computing the relative phase between adjacent samples of the spectrum. These phases may then be added to achieve a cumulative (unwrapped) phase for each point. Both methods have the drawback that the computation must be done sequentially, i.e., the phase at each point must be computed before the phase at the next point can be computed. It should also be noted that if the phase never changes by more than $\pi/2$ between samples, the phase modulo π could be computed and unwrapped with algorithms similar to the above. This is interesting since it is slightly easier to calculate the phase modulo π than the phase modulo 2π (the arctangent algorithm is simpler), and many signals of interest have this property (though noise does not).

There is one class of signals for which phase unwrapping is unnecessary, namely minimum phase signals. A minimum phase sequence is a sequence whose z-transform has no poles or zeroes outside the unit circle. Thus from equation (2-14)

$$\hat{x}(nT) = 0 \quad n < 0 \; . \tag{2-20}$$

Let $\hat{x}_e(nT)$ denote the even part of $\hat{x}(nT)$. Then

$$\hat{x}_e(nT) = \frac{1}{2}\left(\hat{x}(nT) + \hat{x}(-nT)\right) \qquad (2\text{-}21)$$

$$\hat{x}(nT) = 2u(nT)\,\hat{x}_e(nT) \qquad (2\text{-}22)$$

where $u(nT) = 1 \qquad n > 0$

$$= \frac{1}{2} \qquad n = 0$$

$$= 0 \qquad n < 0$$

but

$$\hat{x}_e(nT) = Z^{-1}(\log |X(z)|) .$$

and thus $x(nT)$ can be computed from a transform involving only the real part of $\log X(z)$. The complex cepstrum determined in this manner is equivalent to reconstructing the phase associated with the magnitude of the spectrum using the Hilbert transform. We also see that for $n \geq 0$ the complex cepstrum is identical to the power cepstrum in this case (except for a factor of 2 and the squaring operation). From equation (2-20), we see that the complex cepstrum of a minimum phase sequence is zero at negative quefrencies. Analogously a maximum phase sequence may be defined (the z-transform has no poles or zeroes inside the unit circle) which is zero at positive quefrencies. As will be seen shortly the impulse trains generated in the complex cepstrum by the presence of a single additive echo are nonzero only on one side of the origin, and thus will often be referred to as minimum (or maximum) phase impulse trains. To conclude this section an example of the application of the complex cepstrum to the single additive echo case is presented.

In equation (2-15), let $e(nT) = \delta(nT)+a\delta(nT-n_oT)$.

That is

$$y(nT) = x(nT) * e(nT) = x(nT)+ax(nT-n_oT) \quad . \tag{2-23}$$

Taking the z-transform and evaluating it on the unit circle, we have

$$Y(e^{j\omega T}) = X(e^{j\omega T}) \quad (1+ae^{-j\omega n_o T}) \quad . \tag{2-24}$$

Taking the logarithm of both sides of (2-24), we obtain

$$\hat{Y}(e^{j\omega T}) = \log\,(Y(e^{j\omega T})) = \log\,(X(e^{j\omega T})) + \log\,(1+ae^{-j\omega n_o T}). \tag{2-25}$$

If $a < 1$ we may expand the right most term in (2-25) in a power series; thus

$$\hat{Y}(e^{j\omega T}) = \log X(e^{j\omega T}) + ae^{-j\omega n_o T} - \frac{a^2}{2} e^{-2j\omega n_o T} + \frac{a^3}{3} e^{-3j\omega n_o T} \ldots . \tag{2-26}$$

Inverse z-transforming (2-26) we find the complex cepstrum

$$\hat{y}(nT) = \hat{x}(nT) + a\delta(nT-n_oT) - \frac{a^2}{2}\delta(nT-2n_oT)..... \tag{2-27}$$

Thus the complex cepstrum of the composite signal consists of the complex cepstrum of the basic wavelet plus a train of δ functions located at positive quefrencies at the echo delay (and its multiples) whose heights are simply related to the echo amplitude. Shafer [7] has shown that these δ functions can be effectively removed by multiplying the complex cepstrum by a sequence which is unity everywhere except at multiples of n_oT, and zero at these points. After this comb filter has been applied the complex cepstrum is smoothed at the zeroed points by averaging adjacent points. The filtered

complex cepstrum is then used to recover the basic wavelet by inverting the operations used to compute the complex cepstrum. Had the echo amplitude (a) been greater than 1, (2-25) could have been rewritten

$$\hat{Y}(e^{j\omega T}) = \log\,(ae^{-jn_o\omega T}X(e^{j\omega T})) + \log\,(1+\frac{1}{a}e^{j\omega n_o T}) \qquad (2\text{-}28)$$

which may be expanded

$$\hat{Y}(e^{j\omega T}) = \log\,(ae^{-j\omega n_o T}\,X(e^{j\omega T})) + \frac{1}{a}e^{jn_o\omega T} - \frac{1}{2a^2}e^{j2\omega n_o T} \ldots\ldots \qquad (2\text{-}29)$$

Thus the complex cepstrum will again have peaks at the echo delay (and multiples), but these peaks occur at negative rather than positive quefrencies and their amplitudes will be related to $\frac{1}{a}$ rather than a. If these peaks are filtered out and the wavelet recovery procedure is followed we note that the echo rather than the basic wavelet is recovered.

From the above discussions, we note that the peaks of the impulse train in the complex cepstrum may never have an amplitude of greater than unity regardless of the value of (a). It is also interesting to note that multiplying the original signal by a scale factor only changes the n=0 term of the complex cepstrum, since the scale factor appears as a shift in the mean of the log spectrum. Thus the complex cepstrum is virtually independent of the signal power.

The Relation Between the Complex and Power Cepstrum

It is obvious that the complex and power cepstrum are closely related; however, the exact nature of this relationship has never been fully exploited. This is primarily due to the differing definitions for the two techniques. From equation (2-1) the

relationship between the two cepstra becomes apparent,

$$x_{pc}(nT) = (Z^{-1}(\log X(z) \cdot X^*(z))^2 \qquad (2\text{-}30)$$

$$= (Z^{-1}(\log X(z) + \log X^*(z)))^2 \; . \qquad (2\text{-}31)$$

Assuming $x(nT)$ is real and evaluating its z-transform on the unit circle, we find $X^*(z) = X(z^{-1})$, thus we may write

$$x_{pc}(nT) = (\frac{1}{2\pi j} \oint \log X(z) \cdot z^{n-1} dz + \frac{1}{2\pi j} \oint \log X(z^{-1}) \cdot z^{n-1} dz)^2 \; . \quad (2\text{-}32)$$

Letting $z' = z^{-1}$, we obtain

$$x_{pc}(nT) = (\frac{1}{2\pi j} \oint \log X(z) \cdot z^{n-1} dz + \frac{1}{2\pi j} \oint \log X(z') \cdot z'^{-n-1} dz')^2 \quad (2\text{-}33)$$

but the complex cepstrum is by definition

$$\hat{x}(nT) = \frac{1}{2\pi j} \oint \log X(z) \cdot z^{n-1} dz \; .$$

Thus we may write (2-33) as

$$x_{pc}(nT) = (\hat{x}(nT)) + \hat{x}(-nT)^2 \; . \qquad (2\text{-}34)$$

The power cepstrum is just 4 times the square of the even part of the complex cepstrum. This also follows from the fact that the power cepstrum is the square of the inverse transform of twice the real part of the log spectrum. Equation (2-34) is of interest since as pointed out in [3] the power cepstrum is often superior to the complex cepstrum for epoch detection. Thus in a wavelet recovery system we may wish to compute both the power and complex cepstrum. Rather than using the straightforward method outlined in the section

on the power cepstrum, we can compute the complex cepstrum, and then use the simple relation given by (2-34). This allows the use of one less FFT in the computation process.

The Phase Cepstrum

The inverse transform of the log phase yields peaks at multiples of the echo epoch in much the same way that the inverse transform of the log magnitude does. To see this, let us once again examine the log spectrum for the single additive echo case given in equation (2-25)

$$\hat{Y}(e^{j\omega T}) = \log\,(X(e^{j\omega T})) + \log\,(1+ae^{-j\omega n_o T}) \qquad (2\text{-}25)$$

$$= \log|X(e^{j\omega T})| + j\,\text{Phase}[X(e^{j\omega T})] \qquad (2\text{-}35)$$

$$+ \frac{1}{2}\log\,(1+a^2+2a\,\cos\,\omega n_o T) + j\tan^{-1}\left(-\frac{a\sin\omega n_o T}{1+a\cos\omega n_o T}\right) .$$

The 4th term in equation (2-35) produces ripples in the log phase, just as the third term produces ripples in the log magnitude. Since $Y(e^{j\omega T})$ is the transform of a real sequence, its magnitude ($Re\hat{Y}(e^{j\omega T})$) is an even function of ω, and its phase ($Im\hat{Y}(e^{j\omega T})$) is an odd function of ω. Thus the inverse transform of $Re(\hat{Y}(e^{j\omega T}))$ will yield the even portion of the complex cepstrum and the inverse transform of $jIm(\hat{Y}(e^{j\omega T}))$ will produce the odd portion of the complex cepstrum. Since the inverse transform of the term $\log\,(1+ae^{-j\omega n_o T})$ produces peaks on one side of the origin only, the peaks produced by its real and imaginary parts must be equal in magnitude and must cancel on one side of the origin while reinforcing on the other (according to whether (a) is greater or less than one).

From the above argument it is obvious that a quantity called the phase cepstrum may be defined which should be of use in the estimation of echo epoch times, and amplitudes. Formally the phase cepstrum may be defined as the magnitude squared of the inverse z-transform of twice the phase of the z-transform (or equivalently the imaginary part of the logarithm of the z-transform) of a given input sequence, which may be written

$$x_{phc}(nT) = |Z^{-1}(2 \log X(z) - 2 \log |X(z)|)|^2 \qquad (2\text{-}36)$$

where the factor of 2 has been introduced to eliminate any normalization factors in the relation between the phase and complex cepstrum. It is easy to show from (2-36) that the phase cepstrum is related to the complex cepstrum as below

$$x_{phc}(nT) = (\hat{x}(nT) - \hat{x}(-nT))^2 \ . \qquad (2\text{-}37)$$

Thus the phase cepstrum is to the log phase just what the power cepstrum is to the log magnitude. Empirically it has been determined that the phase cepstrum is less useful than the power cepstrum for the determination of echo epoch times because of its greater sensitivity to noise, but it has proven to be a valuable guide in the determination of the effects of noise on the signal phase, and may find application in problems of signal identification. Computation of the phase cepstrum alone is almost as difficult as that of the complex cepstrum, since both require the phase unwrapping procedure.

SUMMARY

A new cepstrum technique, the phase cepstrum, has been introduced

which should be useful in the detection of echo epoch times. The power and phase cepstra have been shown to be (except for a constant) the square of the even and odd parts of the complex cepstrum. Because of this close relationship between these techniques, only the effects of the data handling procedures on the complex cepstrum are discussed in Chapter IV. The extension of the results to the power and phase cepstra is obvious. Since the effect of noise on the phase cepstrum has not been examined previously, this technique is compared with the power and complex cepstra (in the presence of additive noise) for the detection of echo epoch times and the estimation of echo amplitudes in Appendix A. Some insight is thereby gained into the results of previous authors.

CHAPTER III

THE PITFALLS OF CEPSTRUM COMPUTATION

Since the computation of the complex cepstrum and subsequent wavelet recovery involve a number of discrete Fourier transforms together with nonlinear operations, it is not surprising to find problems associated with these computations not present in ordinary spectral analysis. This chapter is devoted to the study of these problems, some of which have been noted by other authors; others were previously unknown.

LINEAR PHASE TERMS

The phase of the z-transform of a composite signal will often contain a linear trend. This may be a property of the signal under analysis, or may be due to the signal being delayed (that is, not starting at the beginning of the record). In such cases the z-transform of a typical input signal can be rewritten

$$X(z) = z^{-r}X'(z) \qquad (3\text{-}1)$$

where z^{-r} is the term which contributes the linear part of the phase curve. Then

$$\hat{X}(z) = \log X(z) = \log z^{-r} + \log X'(z) \ . \qquad (3\text{-}2)$$

Consider the contribution of the term $\log z^{-r}$ to the complex cepstrum.

$$\hat{x}_{LP}(nT) = \frac{1}{2\pi j} \oint \log z^{-r} z^{n-1} dz \quad . \tag{3-3}$$

Evaluating the above expression on the unit circle $z = e^{j\omega T}$, we obtain

$$\hat{x}_{LP}(nT) = -\frac{1}{2\pi j} \int_{-\pi}^{\pi} (j\omega T r) e^{jn\omega T} j d(\omega T) \quad . \tag{3-4}$$

Integrating, we find that the contribution of the linear phase term to the complex cepstrum is

$$\hat{x}_{LP}(nT) = 0 \qquad n=0 \tag{3-5}$$

$$= -\frac{r}{n} \cos \pi n \qquad n \neq 0 \quad .$$

The linear phase component thus causes a rapid oscillation in the complex cepstrum on which the wavelet and echo information rides. If the linear phase component is sufficiently large, it will dominate the complex (and phase) cepstrum, masking important characteristics.

Figure 3 shows the mean square error (MSE) of the recovered wavelet as a function of the delay in a composite signal. This delay only affects the phase of the composite signal (introducing a linear component). Note the rapid increase in MSE as a function of the delay. For comparison the MSE of the recovered wavelet is shown when the linear phase term is removed prior to computing the complex cepstrum. No explanation is available for the very low MSE observed at the odd delay times. For the composite signal considered (echo amplitude = .5) the echo peak becomes undetectable in both the phase and complex cepstra at a delay of 30 sample times (when no linear

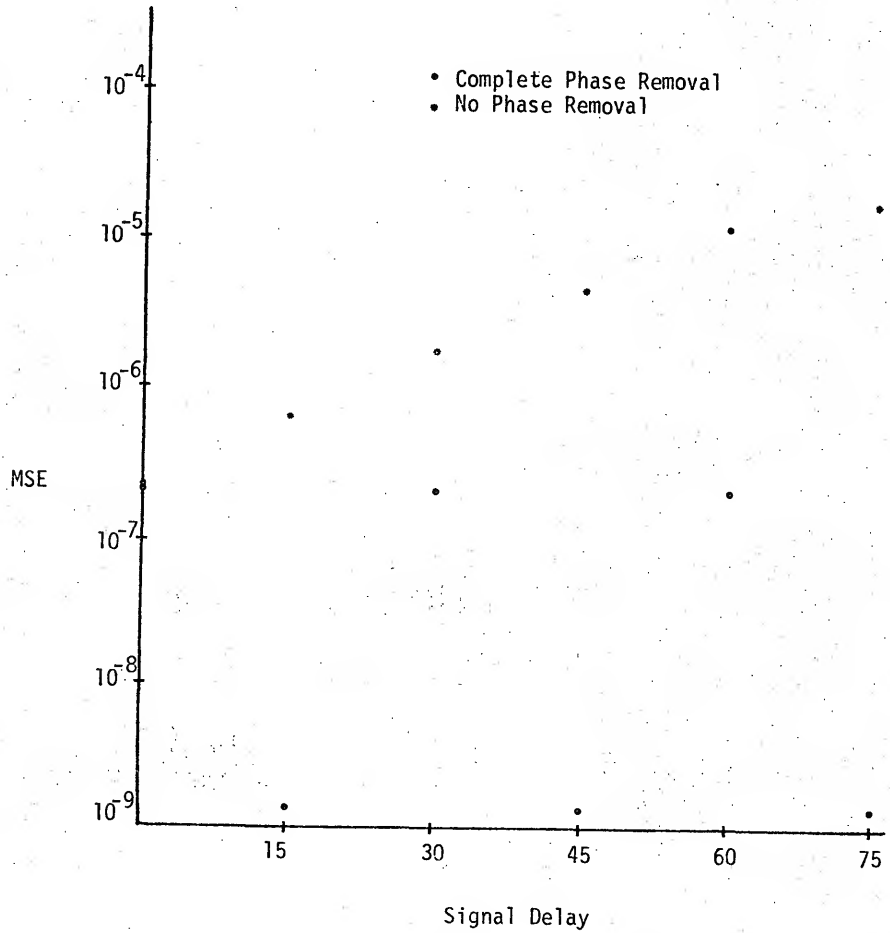


Figure 3 Effect of a Delay in the Composite Signal on the MSE of the Recovered Wavelet.

phase removal is performed). Clearly the removal of linear phase terms is required for an effective wavelet recovery system.

Linear phase removal results in a dramatic change in the appearance of the complex cepstrum and also displaces the recovered wavelet in time (unless corrected for). The linear phase component is removed by calculating the unwrapped phase of the N/2-1 point, using this to compute the slope of the linear phase term, and then subtracting the linear phase term from the signal phase. For the purpose of linear phase removal the unwrapped phase of the N/2-1 point can be truncated to an integer multiple of π [3]. This allows the correction of the recovered wavelet (for linear phase removal) by shifting it an integer number of sample times. This generally still leaves a small linear phase term present in the log spectrum. Figure 7 shows the complex cepstrum for this type of linear phase removal. Note the effects of the remaining linear phase term are clearly visible.

Figure 14 shows the complex cepstrum of the same signal as Figure 7, but here the linear phase term was removed entirely prior to computation of the complex cepstrum. Note the radically different appearance of the complex cepstrum. In this case, the linear phase term must be reinserted in the log spectrum of the recovered wavelet during the inverse complex cepstrum operations.

It should be noted that if the linear phase term is not entirely removed from the log spectrum of the composite signal then the echo peaks cannot be smoothed (without introducing considerable error into the recovered wavelet) by averaging the points adjacent to them since these points have contributions from the linear phase

Example

Incomplete Linear Phase Removal

Figures 4 through 10

This group of figures illustrates the computation of the cepstra, and wavelet recovery when the linear phase term removed from the unwrapped phase curve is based on the unwrapped phase of the N/2-1 point truncated to an integer multiple of π.

The composite signal (256 points) is

$$y(nT) = x(nT) + .5x(nT-30T)$$

where $x(nT) = nT\, e^{-nT}$ $\quad 0 \leq n < 64$

Figure Number	Figure Title
4	Composite Signal SNR = 40 dB
5	Unwrapped Phase Curve
6	Log Magnitude
7	Complex Cepstrum
8	Phase Cepstrum
9	Power Cepstrum
10	Recovered Wavelet MSE = 7.97×10^{-7}

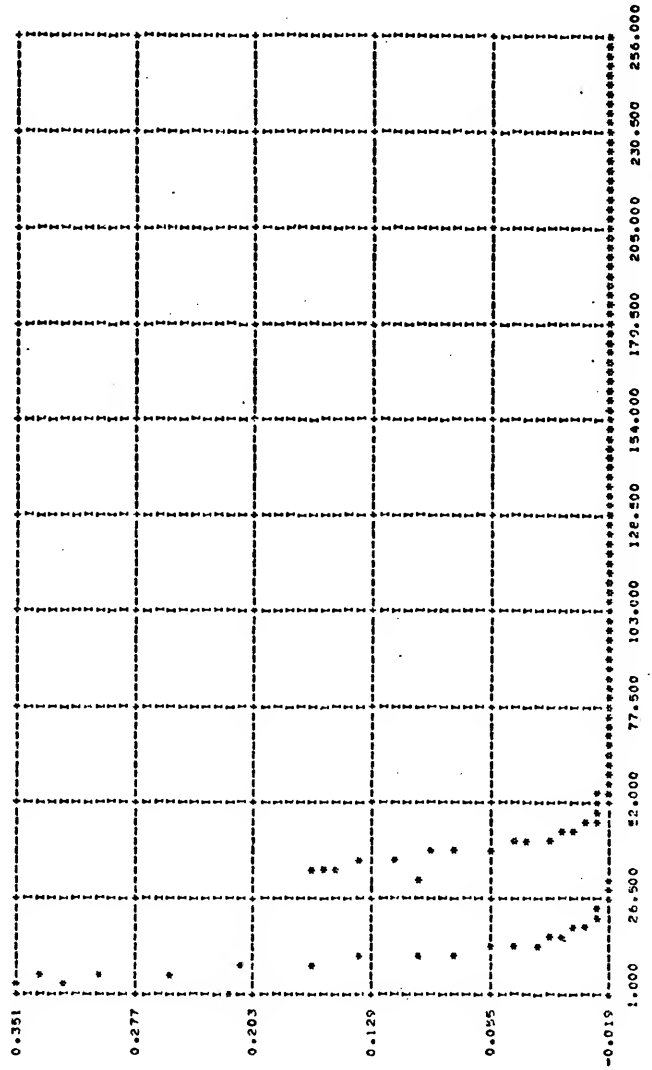


Figure 4 Composite Signal
SNR=40 dB

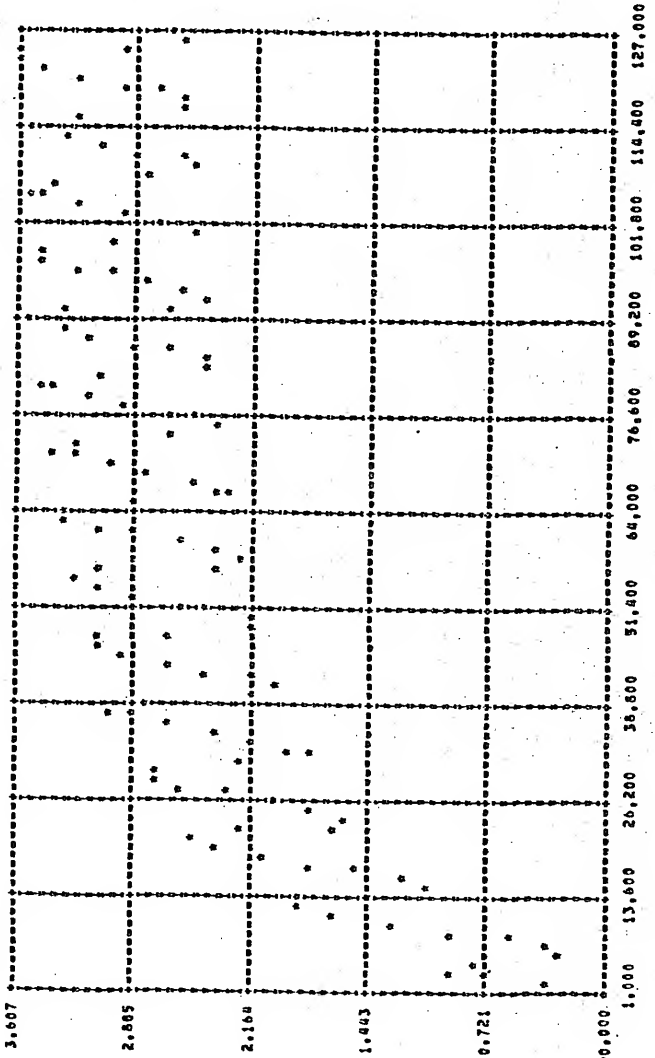


Figure 5 Unwrapped Phase Curve

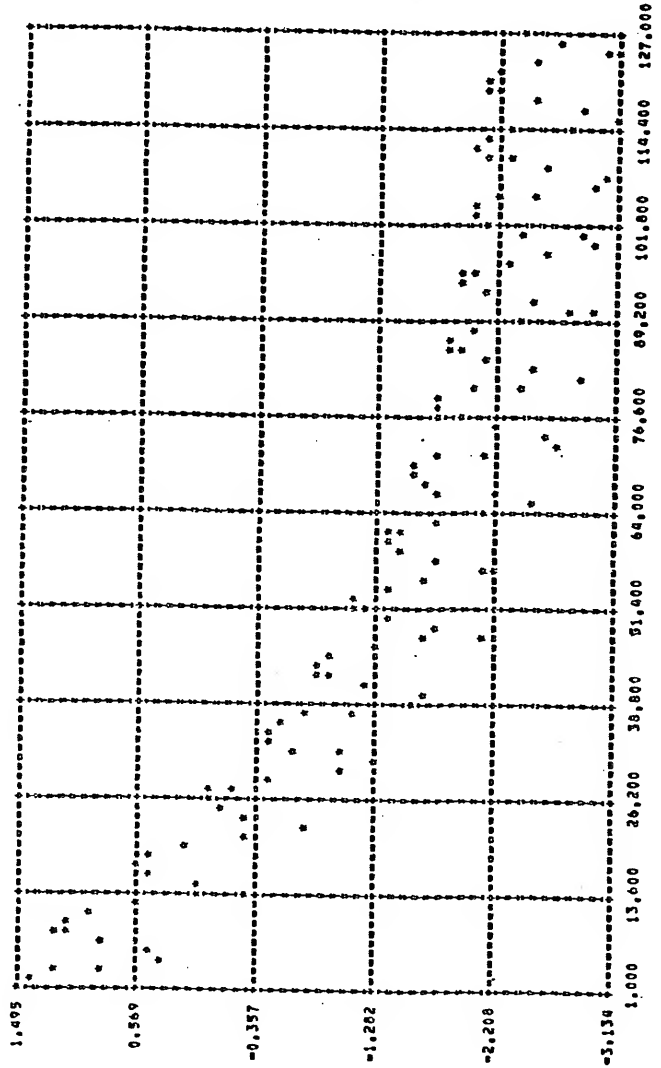


Figure 6 Log Magnitude

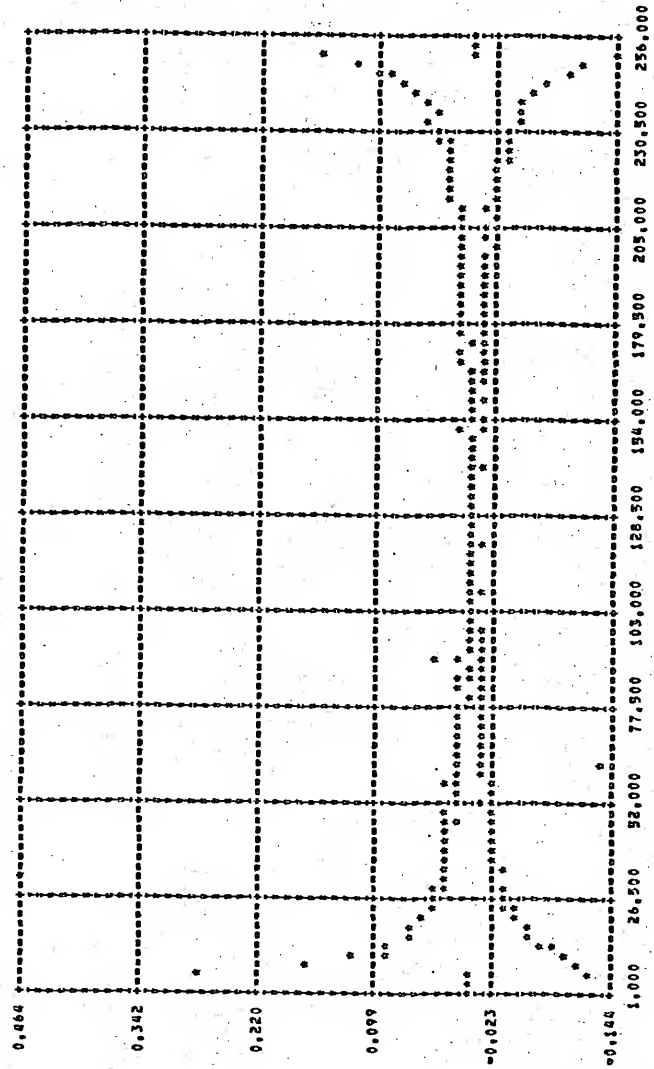


Figure 7 Complex Cepstrum

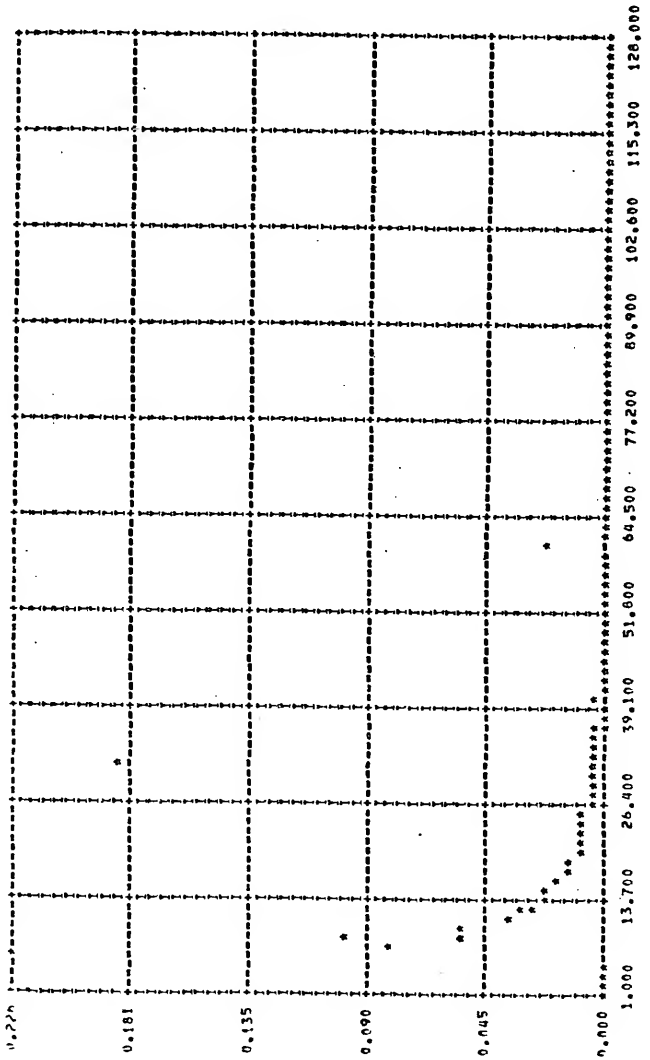


Figure 8 Phase Cepstrum

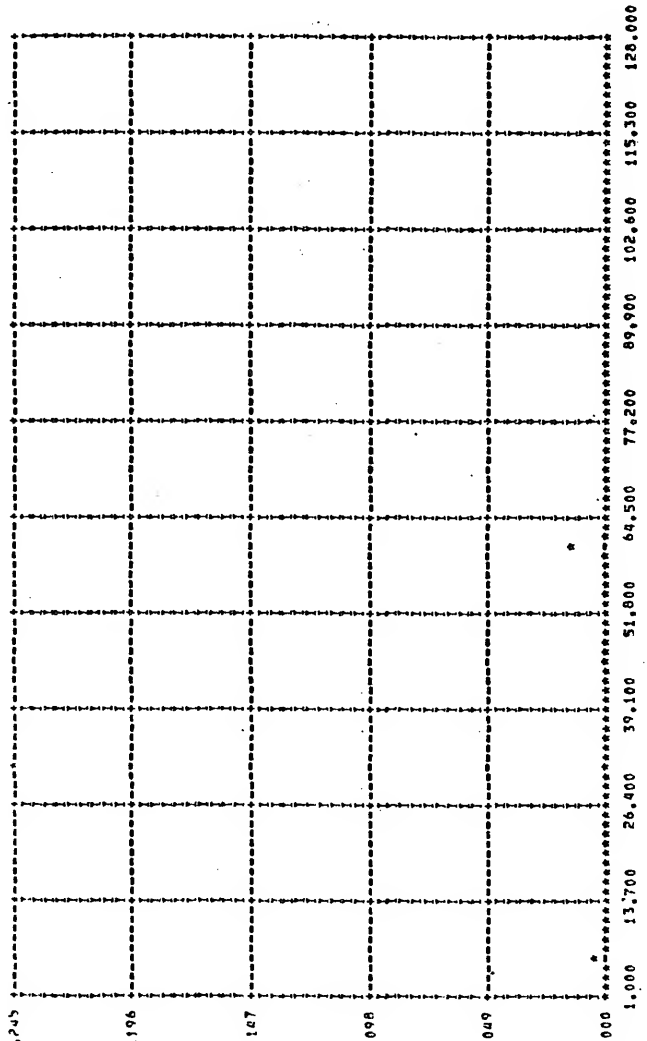


Figure 9 Power Cepstrum

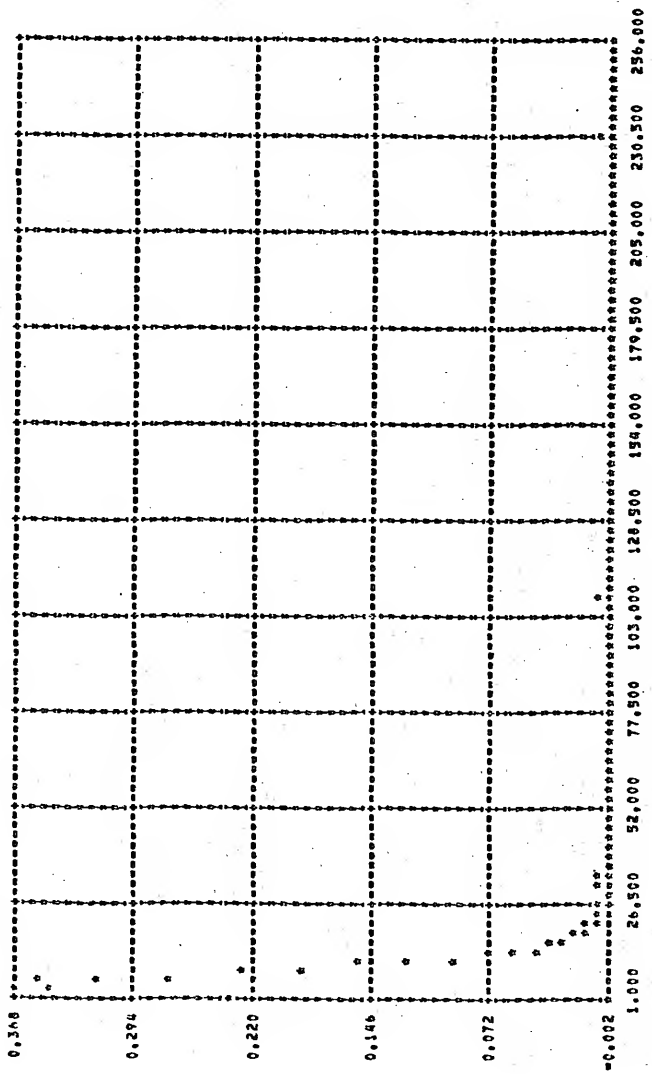


Figure 10 Recovered Wavelet
MSE = 7.97×10^{-7}

Example

Complete Linear Phase Removal

Figures 11 through 17

This group of figures illustrates the computation of the cepstra, and wavelet recovery when the linear phase term removed from the unwrapped phase curve is based on the unwrapped phase of the N/2-1 point (no truncation).

The composite signal (256 points) is

$$y(nT) = x(nT) + .5x(nT-30T)$$

where $x(nT) = nT\, e^{-nT}$ $\quad 0 \leq n < 64$

Figure Number	Figure Title
11	Composite Signal SNR = 40 dB
12	Unwrapped Phase Curve
13	Log Magnitude
14	Complex Cepstrum
15	Phase Cepstrum
16	Power Cepstrum
17	Recovered Wavelet MSE = 8.01×10^{-7}

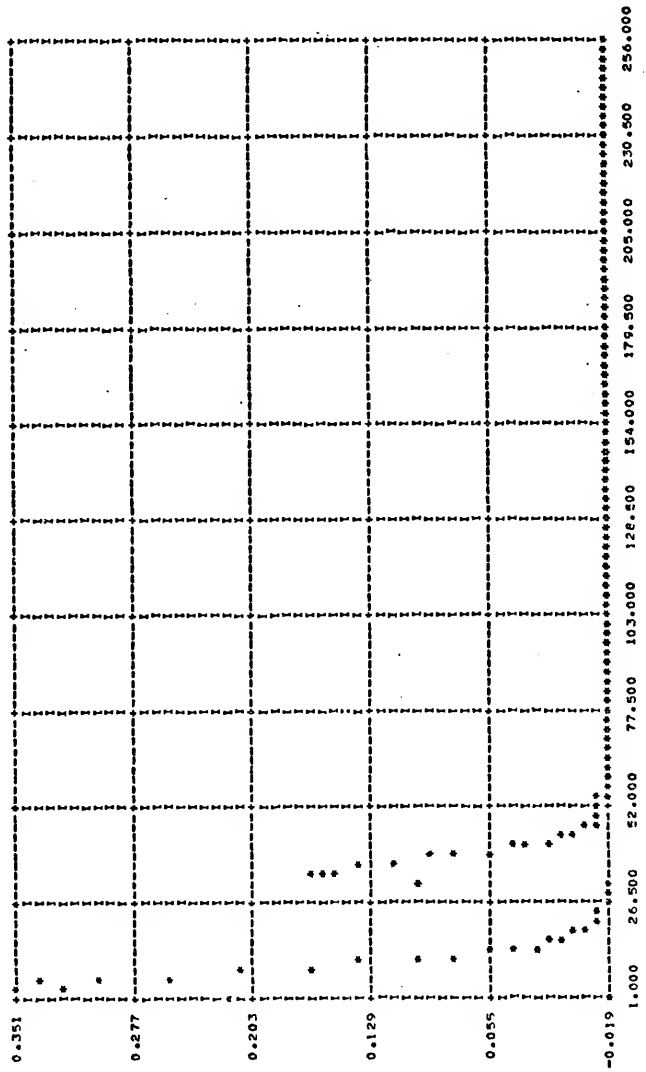


Figure 11 Composite Signal
SNR=40 dB

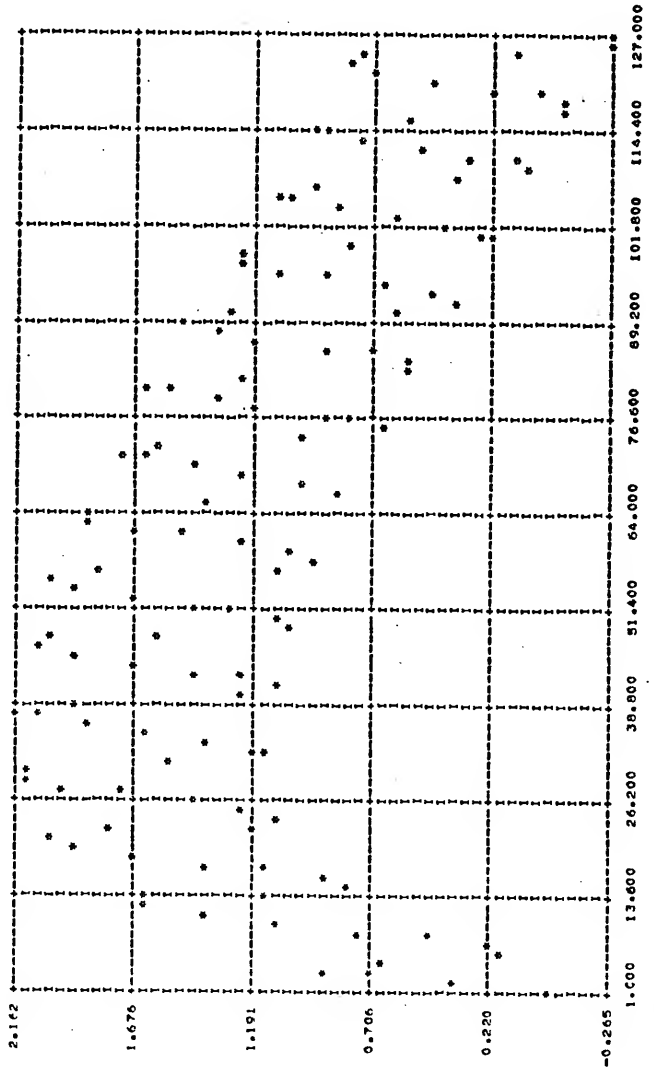


Figure 12 Unwrapped Phase Curve

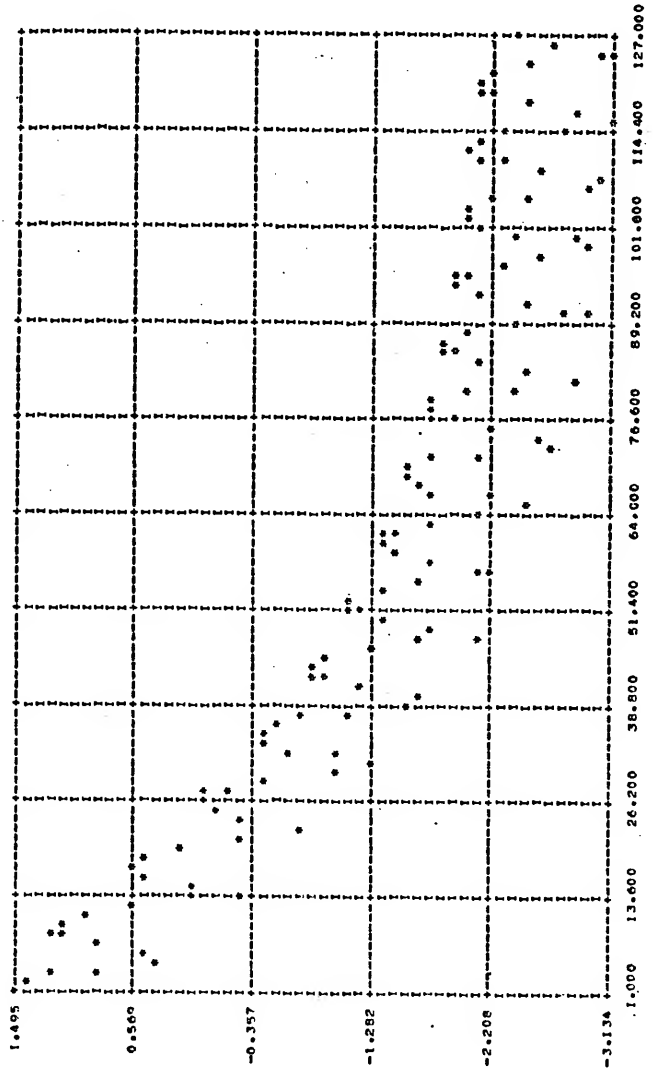


Figure 13 Log Magnitude

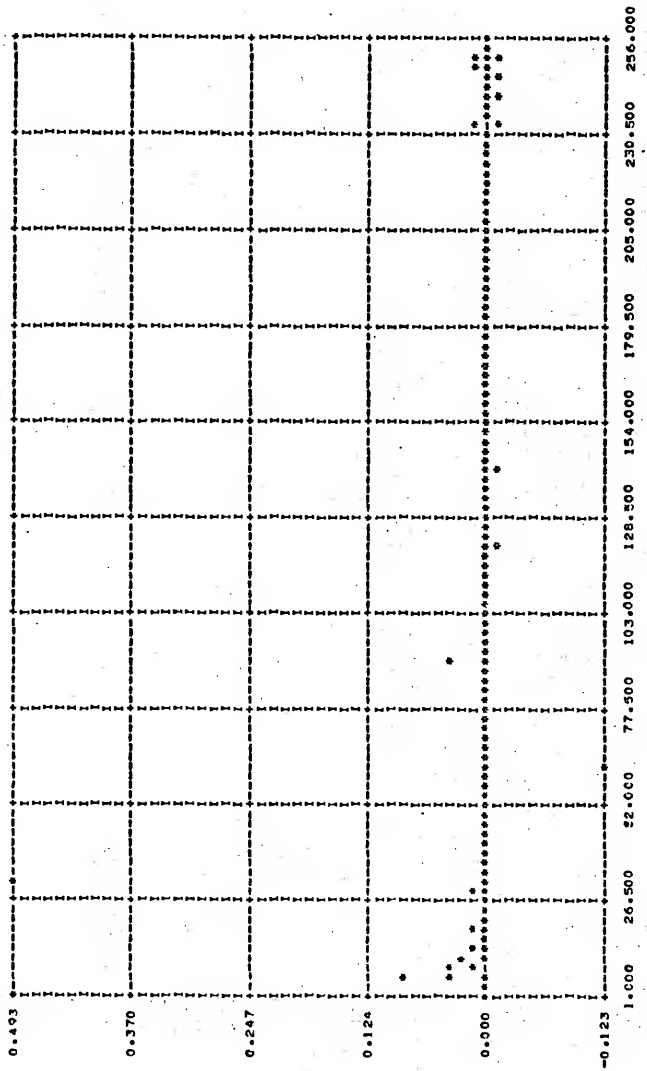


Figure 14 Complex Cepstrum

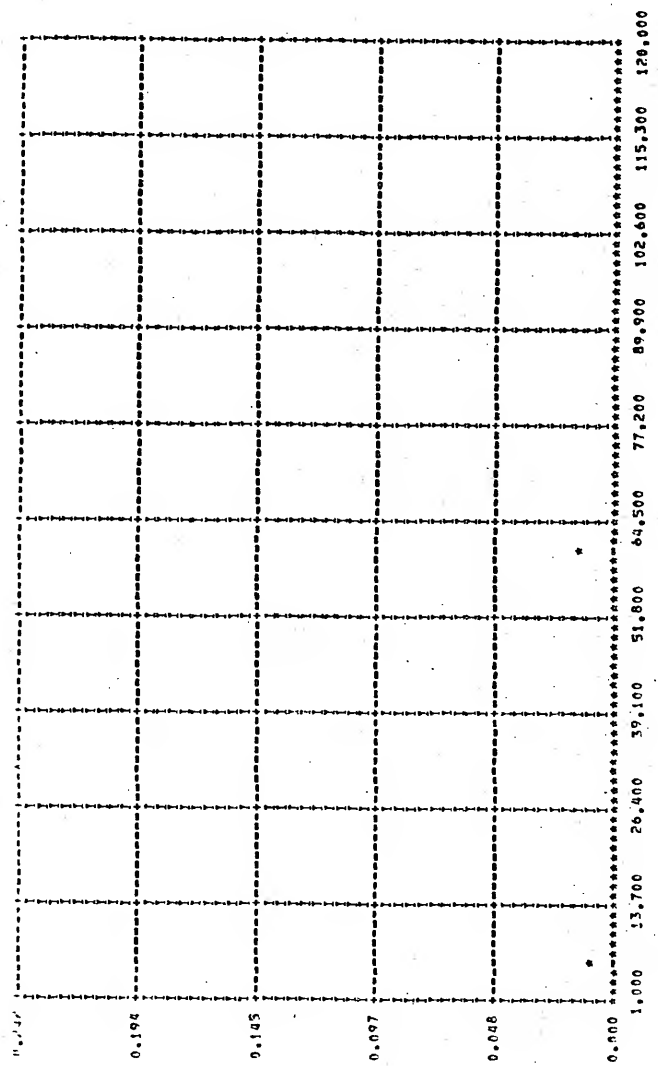


Figure 15 Phase Cepstrum

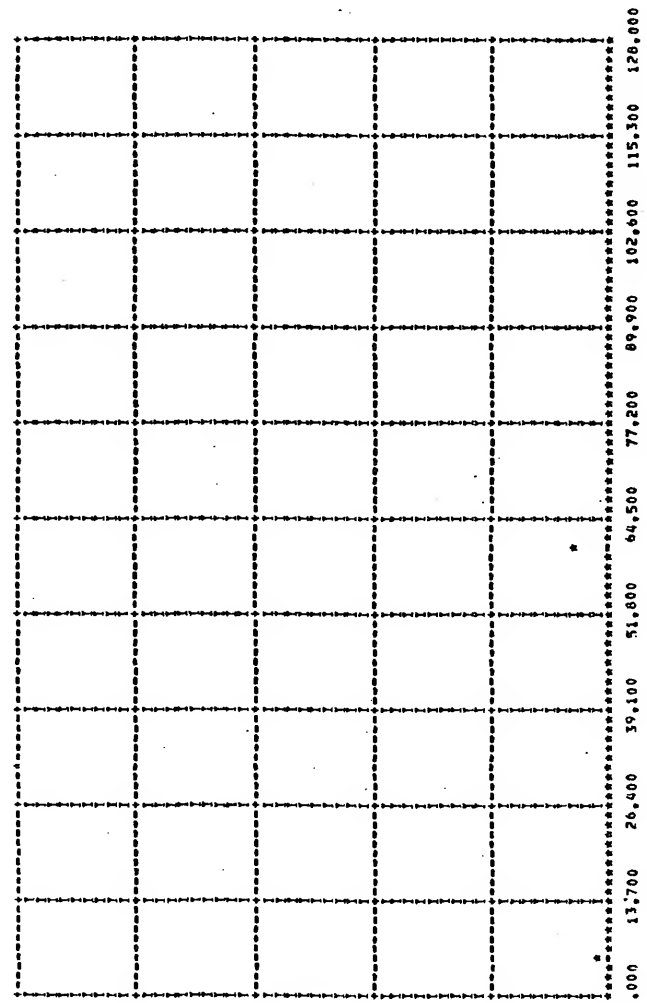


Figure 16 Power Cepstrum

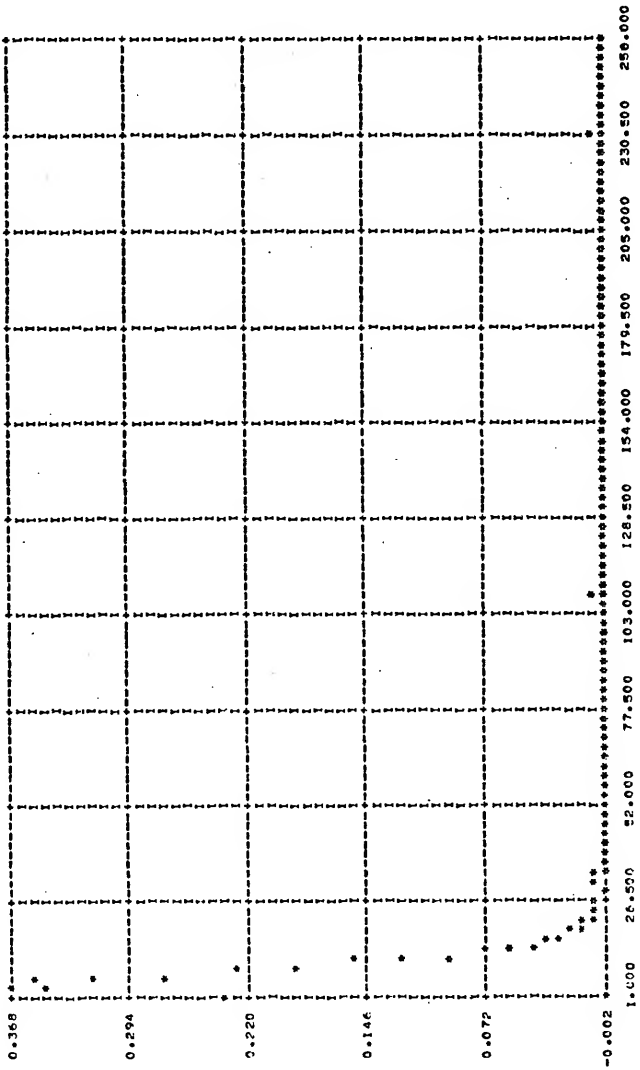


Figure 17 Recovered Wavelet
MSE = 8.01×10^{-7}

component which are opposite in sign to the contribution of the point being smoothed. To smooth the nth point properly when a linear phase term is present the contributions of the n+2 and n-2 points must be averaged. This form of smoothing results in a somewhat smaller MSE (than averaging the points adjacent to the echo peaks) even when the linear phase term is completely removed.

Subject to the above smoothing considerations it has been determined that the MSE of the recovered wavelet is not significantly different for the two forms of linear phase removal, though as mentioned previously the appearance of the complex (and phase) cepstrum is quite different. This undoubtedly will affect the detection of small amplitude echo peaks and echo peaks occuring at low quefrencies. Thus the second form of linear phase removal is preferred, and is employed in the computer experiments discussed in this dissertation.

PHASE UNWRAPPING ERRORS

Previous authors have tacitly assumed that the phase of the DFT of a given sequence may be unwrapped unambiguously. Only recently has it been pointed out that errors may occur in the phase unwrapping process [5]. Since the arctangent routine used in the phase unwrapping algorithm computes the phase modulo 2π, it is evident that phase unwrapping (explained in Chapter II) is performed correctly only if the change in phase between samples of the z-transform of the given sequence is less than π. The DFT is just the z-transform (evaluated on the unit circle) of the sequence sampled at values $\omega = n\Delta\omega = n2\pi/T_R$ where $T_R = NT$ is the record length under consideration.

The sampling theorem (in the frequency domain) indicates that $\Delta\omega$ must be less than or equal to $2\pi/T_R$ in order to reconstruct $X(z)$ exactly from its samples. Thus the DFT samples the z-transform at the minimum rate necessary for reconstruction. Sampling at this rate does not, however, assure us that the phase of the z-transform (evaluated on the unit circle $z=e^{j\omega T}$) does not change by more than π between samples as can be seen by the following simple examples.

Example 1

$$\text{Let } x(nT) = -\frac{1}{2}\delta((n-N/2)T) + \delta((n-N)T) \quad . \tag{3-6}$$

$$X(z)\Big|_{z=e^{j\omega T}} = X(e^{j\omega T}) = -\frac{1}{2}e^{-j\omega\frac{NT}{2}} + e^{-j\omega NT} \quad . \tag{3-7}$$

Consider the phase diagram of $X(e^{j\omega T})$ shown in Figure 18. It is evident from Figure 18 that even if the z-transform is sampled at twice the minimum rate ($\Delta\omega = \frac{\pi}{T_R} < \frac{2\pi}{T_R}$) the change in phase can be greater than π. Thus the phase unwrapping algorithm will yield incorrect results. To see the effects of this on the computed phase curve consider Figure 19. Obviously the computed phase curve differs considerably from the true phase curve.

Example 2

$$\text{Let } x(nT) = y(nT-n_oT) \qquad [0,N) \tag{3-8}$$
$$= 0 \qquad \text{otherwise} \quad .$$

$$X(e^{j\omega T}) = e^{-j\omega n_o T} Y(e^{j\omega T}) \quad . \tag{3-9}$$

As expected the phase of $X(e^{j\omega T})$ is the sum of a linear phase component and the phase of $Y(e^{j\omega T})$. If ω is sampled at the minimum rate, that is, $\omega = n\Delta\omega = \frac{n2\pi}{NT}$, then

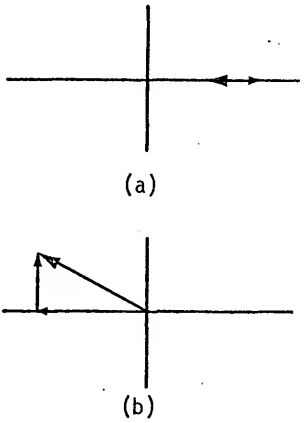


Figure 18 Phase of $X(e^{j\omega T})$. (a) $\omega=0$, and (b) $\omega=\pi/T_R$.

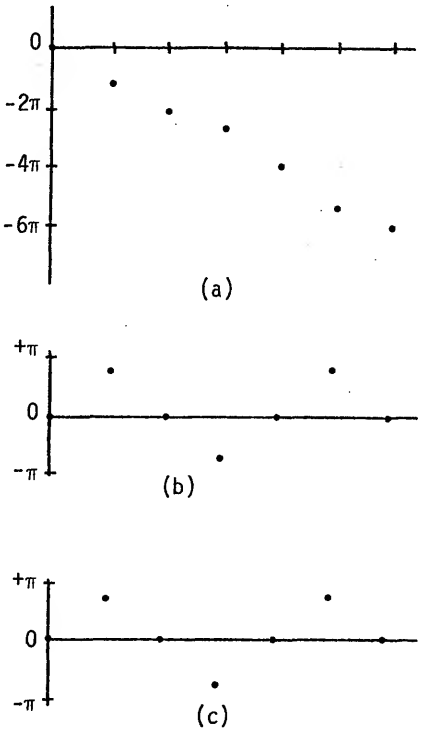


Figure 19 Phase Unwrapping Errors. (a) true phase, (b) phase modulo 2π, and (c) unwrapped phase.

$$X_s(e^{j\omega T}) = X(e^{jn\frac{2\pi}{N}}) = e^{j2\pi n(n_o/N)} Y_s(e^{j\omega T}) \quad . \tag{3-10}$$

Thus if $n_o > N/2$ the linear component of the phase will change by more than π between samples and unless the phase of $Y(e^{j\omega T})$ counteracts this change, the phase unwrapping algorithm will yield erroneous results. This was observed in computer experiments when the composite signal is delayed by more than half the record length. As expected this not only reduces the echo detectability in both the phase and complex cepstra, but also severely distorts the recovered wavelet.

Fortunately, provided the composite signal starts relatively early in the record, many signals of interest can be unwrapped correctly since their phase changes slowly. Furthermore many signals only occupy a portion of the total record, and thus the DFT yields samples of the z-transform spaced more closely than dictated by the minimum sampling considerations. Even in the absence of the above conditions, this problem can be alleviated by extending the data record with zeroes as this increases the sampling rate of the z-transform (this will be shown in the next chapter).

These errors often occur when unwrapping the phase of a composite signal in additive noise at low signal-to-noise ratios (SNR), since the noise may dominate portions of the spectrum and the phase of the noise spectrum will change rapidly from sample to sample. This error in phase unwrapping seems to be actually beneficial in this case in that it always limits the jumps in phase (to less than π) when in actuality the jumps may be somewhat larger. This topic

will be discussed in more detail in the section on the addition of zeroes in the following chapter.

ALIASING

Even though the DFT yields adequately spaced samples of the z-transform (X(z)), it is not surprising that aliasing may be a problem in computation of the cepstra, since the nonlinear complex logarithm operation will introduce harmonics into $\hat{X}(z)$.

Consider, for example, the complex cepstrum. Because of the harmonics introduced into $\hat{X}(z)$, the complex cepstrum will be infinite in extent. The DFT will yield a finite N point sequence, thus the contributions from quefrencies $nT > \frac{NT}{2}$ and $nT < -\frac{NT}{2}$ will alias into the interval $(-\frac{NT}{2}, \frac{NT}{2})$.

The aliasing of the cepstra of the basic wavelets considered did not prove to be a problem. This might have been expected since their time duration was never more than 25% of the total record length.

One manifestation of aliasing is an ambiguity in the determination of the echo epoch time. A peak at n_0T in the complex cepstrum may be caused either by an echo delayed by $(N-n_0)T$ or by an echo delayed by n_0T. This ambiguity cannot be resolved unless the relative echo amplitude is known to be either greater or less than unity (unless the original data record is extended by the addition of zeroes).

Aliasing of the echo impulse train can also be a problem. The aliasing of the higher order terms in the log expansion given in equation (2-25) is often observed. This is especially troublesome

if (a) is close to one since the amplitudes of the peaks in the impulse train do not fall rapidly in this case.

Aliasing is also evident when additive noise is introduced into the observed record. Since noise is present throughout the observed record it is not surprising that it produces contributions at high quefrencies in the complex cepstrum and thus may cause aliasing.

In the noise analysis of Appendix B, it is shown that aliasing is the primary mechanism through which noise is introduced into the complex cepstrum at negative quefrencies for high SNR.

As will be seen in the next chapter, adding zeroes to the input data sequence reduces the aliasing of the complex cepstrum. In fact, if the number of zeroes added is greater than or equal to the original record length the ambiguity discussed above can never occur since the echo delay will always be less than one half the augmented record length.

Aliasing can also be reduced (for a given composite signal) by choosing the record length NT as large as possible subject to the constraints on the number of points analyzable and on the minimum sampling rate. That this is the case becomes evident if we recall that the samples of the log spectrum are spaced $\Delta f = \frac{1}{T_R} = \frac{1}{NT}$ apart; thus increasing NT increases the "sampling rate" of the log spectrum, and should minimize aliasing in its inverse DFT. This gain is offset somewhat by the fact that increasing the duration in time of the observed data increases the "long time" (high quefrency) contributions to the cepstra from noise (assuming the composite signal remains constant). Furthermore the shorter the interval of the data record

that the composite signal occupies, the lower the SNR for a given noise environment.

OVERSAMPLING

One problem caused by additive noise occurs when the composite signal is oversampled. Outside the signal band (although these components may have been greatly attenuated prior to sampling) noise often dominates the spectrum. This presents no problem in ordinary spectral analysis as these components contain little power, but they may have a marked effect on the cepstra (since the complex logarithm of the spectrum is taken). Because of this nonlinearity the regions of low power may contribute as much or more to the cepstra as the signal band, and therefore will degrade both echo detectability and eventual wavelet recovery. This effect is especially evident in the phase cepstrum since the phase of the spectrum outside the signal band may change rapidly from point to point. It is difficult to assess the effect of oversampling on the recovered wavelet, but the appearances of the complex, phase and power cepstra are observably degraded by oversampling.

Oversampling also aggravates the phase unwrapping and aliasing problems previously noted, since it shortens the data record (if the number of points is fixed) which in turn implies that the samples of the log spectrum are spaced farther apart.

SUMMARY

Four problems connected with cepstral analysis have been examined. These problems are linear phase terms, phase unwrapping errors, aliasing of the cepstra, and oversampling.

The presence of a linear phase term distorts the appearance of the complex and phase cepstra even when partially removed as in reference [3]. Such a term degrades not only echo epoch detectability in the phase and complex cepstra, but also wavelet recovery. The complete removal of linear phase contributions to the cepstra is a necessity for satisfactory wavelet recovery.

Phase unwrapping errors are observed to occur if the log phase changes by more than π between samples. This can be quite detrimental to echo epoch detectability in the complex and phase cepstra, and to wavelet recovery.

Some aliasing of the cepstra is always present since the cepstra are derived from samples of the log spectrum. This results in an ambiguity in the echo epoch determination. Both the aliasing and phase unwrapping problems can be alleviated by extending the original data record with zeroes.

Finally, oversampling the input data record is observed to degrade wavelet recovery when additive noise is present. Oversampling also aggravates the phase unwrapping and aliasing problems, and thus should be avoided.

CHAPTER IV

THE EFFECTS OF SOME DATA PROCESSING TECHNIQUES ON THE CEPSTRA

Care must be exercised when computing the complex, phase, and power cepstra. The repeated use of the DFT may raise the problems of aliasing, leakage, and the picket fence effect. Since the complex, phase, and power cepstra are nonlinear techniques, it is not clear that ordinary data processing techniques, normally used to improve spectral analysis, are applicable. Two such techniques, windowing (normally used to reduce leakage) and the addition of zeroes (normally used to reduce the picket fence effect), are examined to determine the effects of their application at various stages in the echo detection and wavelet recovery process. Primarily the effects of these techniques on the complex cepstrum are discussed, since the extension of the results to the phase and power cepstra is usually obvious.

WINDOWING THE COMPOSITE SIGNAL

When the input data record is windowed the applicability of cepstrum techniques is no longer evident. Consider for a single additive echo

$$y(nT) = [x(nT-k_oT) + ax(nT-k_oT-n_oT)]w(nT) \qquad (4\text{-}1)$$

where $w(nT) = 0 \qquad n > L-1 \quad \text{or} \quad n < 0$

$x(nT) = 0 \qquad n < 0$

$w(nT)$ is the windowing sequence. The basic wavelet $x(nT)$ has the argument $(n-k_o)T$, since we wish it to begin at some unspecified point (k_oT) because it is not known where (if at all) in the data record the composite signal begins. Equation (4-1) is rewritten

$$y(nT) = [x(nT)*(\delta(nT-k_oT)+a\delta(nT-k_oT-n_oT))]w(nT) \; . \tag{4-2}$$

Z-transforming (4-2), we obtain

$$Y(z) = [z^{-k_o}(X(z)(1+az^{-n_o}))]*W(z) \; . \tag{4-3}$$

Thus, the contributions of the basic wavelet $x(nT)$ and the echo delay cannot be separated by taking the logarithm since the term in brackets is convolved with $W(z)$. Fortunately many practical situations permit the extraction of the basic wavelet (though there is some error) and detection of the echo arrival time.

Experiments have been undertaken to determine the effects of windowing the composite signal on the wavelet recoverability (as measured by the MSE between the recovered wavelet and the true wavelet) and on the echo epoch detectability. These effects are ascertained in both the noise free and the additive noise case. Unless otherwise noted the composite signal examined is of the form

$$y(nT) = x(nT-k_oT) + ax(nT-k_oT-n_oT) \quad 0 \leq n \leq L-1 \tag{4-4}$$

and

$$x(nT) = nTe^{-bnT} \qquad 0 \leq n < 64 \tag{4-5}$$

with $L = 256$

$a = .5$

$n_o = 30$

$k_o = 0$

$b = .1, .5, 1.0$

$T = .333$.

A typical experiment is conducted as follows: a composite signal as given above is generated. Bandlimited random noise is added to this sequence and it is windowed. The complex, phase, and power cepstra are computed. Peaks due to the echo train are smoothed in the complex cepstrum, which is then inverse transformed to obtain an estimate of the basic wavelet. This estimate is corrected for the windowing by multiplication by $(w(nT))^{-1}$ where $w(nT)$ is the windowing sequence. The mean square error (MSE) is computed between the recovered and the initial basic wavelet over the entire record length.

The Exponential Window

The exponential window

$$w(nT) = \alpha^n \quad 0 \leq n < L \qquad \alpha < 1 \qquad (4\text{-}6)$$

$$= 0 \quad \text{otherwise}$$

has been proposed by Shafer (7) to reduce any error associated with truncating the echo. To some extent this window also reduces leakage, but since its properties are quite different from those of the commonly used windows, it will be studied separately.

Experiments were undertaken to determine the effects of this window on wavelet recoverability and to determine its effects on

echo detectability both in the noise free and the additive noise cases.

The following observations are made on the effects of this window:

(1) No difficulty is encountered detecting the echo epoch, though the peaks in the power, phase and complex cepstrum are reduced in height. The echo detectability threshold in the case of additive noise is about the same as observed for the rectangular window (about a signal to noise ratio (SNR) of 8dB).

(2) The MSE of the recovered wavelet is somewhat better than that obtained with the rectangular window at low SNR (14 dB and below), but not nearly as good at higher SNR. A comparison of the MSE is given in Figure 20. At high SNR the MSE in the recovered wavelet is comparable to that for the rectangularly windowed case if it is computed only over the known duration of the basic wavelet. Thus the MSE results shown in Figure 20 reflect the distortions introduced into the recovered record outside the signal duration by correcting for the exponential window.

Interpretation of Results

The following derivation is presented in order to clarify the effects of the exponential window. Consider again the z-transform of equation (4-1), it is:

$$Y(z) = \sum_{n=0}^{L-1} [w(nT)x(nT-k_oT)+aw(nT)x(nT-k_oT-n_oT)]z^{-n} . \qquad (4-7)$$

$$Y(z) = X_w(z)+az^{-n_o} \sum_{n=-n_o}^{L-n_o-1} x(nT-k_oT)w(nT+n_oT)z^{-n} . \qquad (4-8)$$

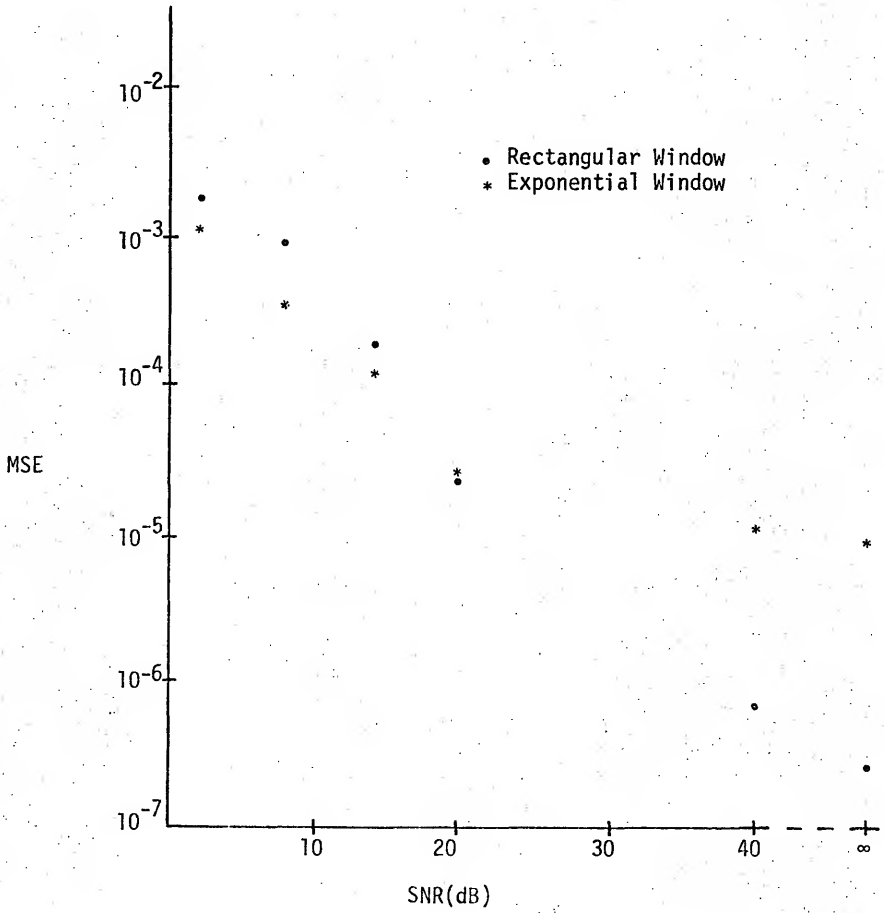


Figure 20 MSE of Recovered Wavelet when the Input Data Record is Exponentially Windowed

where $X_w(z) = \sum_{n=0}^{L-1} w(nT)x(nT-k_oT)z^{-n}$.

$$Y(z) = X_w(z) + az^{-n_o} \sum_{n=0}^{L-n_o-1} x(nT-k_oT)w(nT+n_oT)z^{-n}+E_1 \qquad (4\text{-}9)$$

where $E_1 = az^{-n_o} \sum_{n=-n_o}^{-1} x(nT-k_oT)w(nT+n_oT)z^{-n}$.

Now letting $w(nT) = \alpha^n \qquad 0 \leq n < L$

$$= 0 \qquad \text{otherwise}$$

$$Y(z) = X_w(z) + az^{-n_o}\alpha^{n_o} \sum_{n=0}^{L-n_o-1} x(nT-k_oT)\,\alpha^n z^{-n}+E_1 \quad . \qquad (4\text{-}10)$$

$$= X_w(z)+az^{-n_o}[\alpha^{n_o} \sum_{n=0}^{L-1} x(nT-k_oT)\alpha^n z^{-n} - z^{n_o-L}\alpha^L \sum_{n=0}^{n_o-1} x(nT+LT-n_oT-k_oT)\alpha^n z^{-n}]$$

$$+ E_1 \; . \qquad (4\text{-}11)$$

The term denoted E_1 is an error term. A close examination reveals that the error E_1 arises because it has not been assumed that the basic wavelet begins within the windowed record. If this assumption is made (that is, k_o must be positive and less than L-1), then the error term E_1 vanishes. Henceforth this assumption is made unless otherwise noted. The second term in brackets also is an error term. It arises because the echo has been truncated. It is the error due to this term that this window minimizes (due to the presence of the factor α^L). At present it is assumed that the duration of x(nT) is less than $(L-n_o-k_o)T$ so that the truncation error vanishes. With the above assumption equation (4-11) becomes

$$Y(z) = X_w(z) + \alpha^{n_o}az^{-n_o}\,X_w(z) = X_w(z)(1+\alpha^{n_o}az^{-n_o}) \qquad (4\text{-}12)$$

and

$$\hat{Y}(z) = \log Y(z) = \log X_w(z) + \log (1+\alpha^{n_o} a z^{-n_o}) \quad . \qquad (4\text{-}13)$$

Clearly the presence of the factor α^{n_o} alters the heights of the peaks which are observed in the complex cepstrum. If $a < 1$, the peaks in the complex cepstrum are reduced by the factor $(\alpha^{n_o})^m$ where m is the order of the peak (as compared to the rectangularly windowed case). This is precisely what is observed in the experimental section. If $a > 1$, and $\alpha^{n_o} a > 1$ then the peaks in the complex cepstrum are enhanced by the factor $(\alpha^{n_o})^{-m}$. If $a > 1$, and $\alpha^{n_o} a < 1$, then the peaks may be either reduced or enhanced, and they occur at positive rather than negative quefrencies. The basic wavelet rather than the echo is then recovered.

It can be seen from the above discussion that aliasing of the echo peaks can be reduced by choosing α properly. Another possible advantage to using this window occurs in the case of multiple echos. Kemerait [3] has indicated that when the sum of the echo amplitudes is near one, ambiguous peaks occur in the cepstrum. Again by choosing α properly this situation can be avoided. Assuming that the contribution to the log spectrum from the term $\log(1+\alpha^{n_o} z^{-n_o})$ can be filtered out (by transforming and smoothing the corresponding peaks in the complex cepstrum), the inverse wavelet recovery procedure should, after correcting for the windowing, yield the basic wavelet. Experimentally this is found to be true over the duration of the basic wavelet, but the windowing correction appears to introduce some error into the recovered record outside this interval and thus the MSE results shown in Figure 20 are somewhat greater than observed for the rectangular window at high SNR. As stated previously, at low SNR

the MSE is substantially lower than that obtained using the rectangular window. It is also noted that the echo detectability threshold is about the same as observed for the rectangular window, even though the peaks in the complex cepstrum are reduced in amplitude. Both of the above results are undoubtedly due to the fact that the signal occurred near the beginning of the record and thus is not reduced as much by the windowing as is the noise which is spread throughout the record. Had the signal occurred much later in the record, it is expected that the MSE would increase considerably, and that the echo detectability would suffer. This has subsequently been verified experimentally. Essentially, the cost of reducing the error due to truncation with the exponential window is that the portion of the signal occurring late in the data record is also discriminated against.

The Common Window

Three windows (Hamming, Hanning, and Tapering) commonly used to reduce leakage in spectrum analysis are examined to determine the effects of their application. Since these windows exhibit similar properties, they are discussed as a group in this section with the Hamming window being discussed first, and the other windows then compared with it.

The following observations are made on the effects of the Hamming window.

(1) Several signals of the form given in equations (4-4) and (4-5) with (b) ranging from .1 to 1.0 were generated without additive noise. In all cases the echo epoch is detectable, though the height of the peaks found in the complex, phase, and power cepstrum are

quite different than are found when the rectangular window is used. Substantial peaks at the echo epoch time are often found at both positive and negative quefrencies in the complex cepstrum with the negative quefrency peaks being larger (the rectangularly windowed sequence has peaks at positive quefrencies only). The peaks are somewhat smeared. For the echo delay used (n_0=30), satisfactory wavelet extraction is not possible except when b=1.0; in this case extraction of the echo rather than the basic wavelet produces the least error (since the negative quefrency peaks dominate the complex cepstrum) though considerable distortion is still evident in the recovered wavelet.

(2) Since wavelet recovery is only possible when b=1.0, a number of experiments were conducted with b=1.0 to determine the effects of the window when noise is added. Figure 21 shows the MSE of the recovered wavelet as a function of the SNR when the input data record is Hamming windowed. For reference the MSE obtained with the rectangular window is also shown. It is noted that the MSE is sometimes reduced for the Hamming windowed composite signal by smoothing not only the peaks, but several adjacent points as well.

For this Hamming windowed composite signal the peaks in the complex cepstrum occur at negative rather than positive quefrencies. In the no noise case the negative quefrency peaks are larger than for the corresponding rectangularly windowed signal. Even with this increased peak height in the no noise case the SNR at which the echo epoch becomes undetectable is about 12 dB greater than found for the rectangularly windowed case.

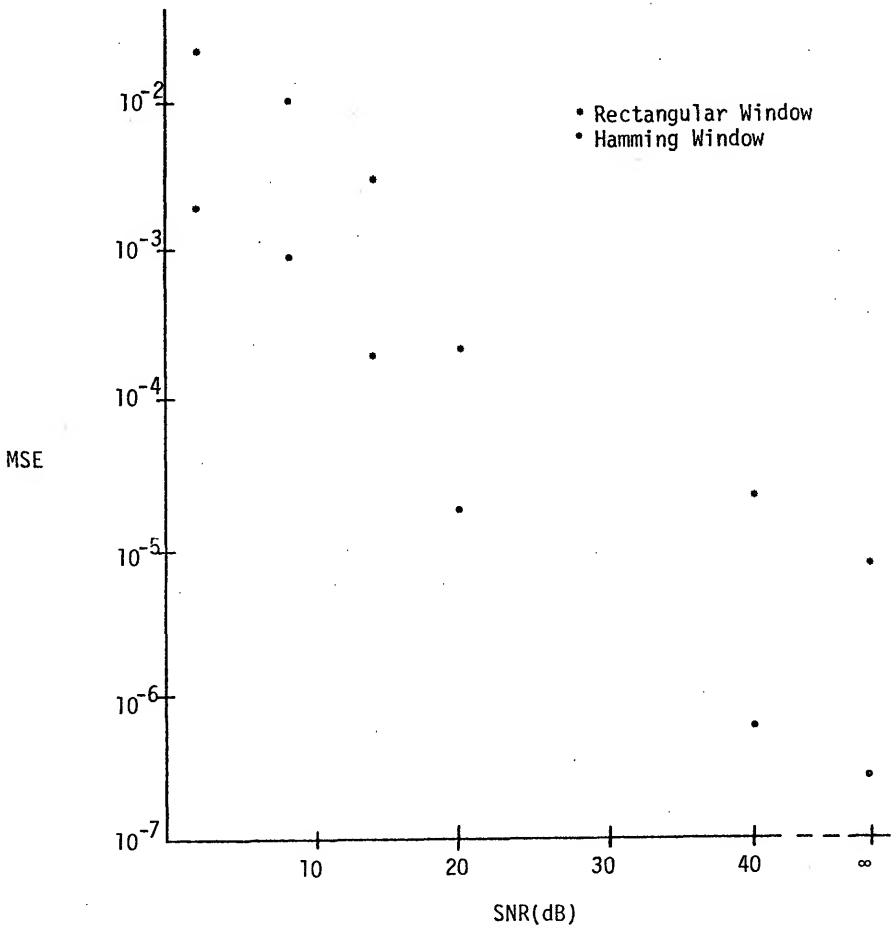


Figure 21 MSE of the Recovered Wavelet when the Input Data Are Hamming Windowed.

Only a few computer computations were made using the Hanning and Tapering windows, but general trends have been ascertained from the results. The Hanning window produces considerably worse results than the Hamming. Likewise, the Tapering window gives poor results if a major portion of the composite signal lies in the region of tapering; however, if the composite signal lies in the untapered region the results are almost identical to those for the rectangular window. One additional drawback of these two windows is that they are zero at the endpoints and thus it is impossible to correct the recovered wavelet at these points. It is noted that correction for the window when $w(nT)<<1$ (that is, near the ends of the record) appears to introduce some distortion.

Interpretation of results

The following theoretical analysis is intended to clarify the conditions under which wavelet extraction is possible after using one of the foregoing windows. Consider (4-7) which is rewritten as

$$Y(z) = w(k_oT)\sum_{n=0}^{L-1} x(nT-k_oT)z^{-n}+aw(n_oT+k_oT)\sum_{n=0}^{L-1} x(nT-k_oT-n_oT)z^{-n}$$

$$+\sum_{n=0}^{L-1} x(nT-k_oT)[w(nT)-w(k_oT)]z^{-n}+a\sum_{n=0}^{L-1} x(nT-k_oT-n_oT)[w(nT)-w(n_oT+k_oT)]z^{-n}. \qquad (4\text{-}14)$$

The third and fourth terms of equation (4-14) can be considered error terms. It will be seen below that extraction of the wavelet is possible if these terms are small. This requires the window to be approximately constant over the duration of $x(nT)$. For the windows under consideration this requires the duration of $x(nT)$ to be much

less than the record length. The error terms vanish completely when the window is rectangular. The sum of the third and fourth terms will be designated as error term E_2. Equation (4-14) is now rewritten as

$$Y(z) = [w(k_oT) + az^{-n_o} w(n_oT+k_oT)]\, X(z) + E_2 \qquad (4\text{-}15)$$

where $X(z) = \sum_{n=0}^{L-1} x(nT-k_oT)z^{-n}$.

If $\beta = \dfrac{aw(n_oT+k_oT)}{w(k_oT)} > 1$, then

$$\hat{Y}(z) = \log Y(z) = \log[az^{-n_o} w(n_oT+k_oT)X(z) + \frac{E_2}{1+z^{n_o}/\beta}] + \log(1+z^{n_o}/\beta). \qquad (4\text{-}16)$$

The term $\log (1+z^{n_o}/\beta)$ may now be expanded as we have done previously to obtain

$$\hat{Y}(z) = \log[az^{-n_o} w(n_oT+k_oT)X(z) + \frac{E_2}{1+z^{n_o}/\beta}] + \frac{z^{n_o}}{\beta} - \frac{z^{2n_o}}{2\beta^2} \ldots . \qquad (4\text{-}17)$$

Thus the peaks in the complex cepstrum occur at negative quefrencies. If these peaks are removed, then the recovered wavelet will be

$$y_{RW}(nT) = Z^{-1}(az^{-n_o} w(n_oT+k_oT)X(z) + \frac{E_2}{1+z^{n_o}/\beta}) \ . \qquad (4\text{-}18)$$

Expanding $\dfrac{1}{1+z^{n_o}/\beta}$ in a series, and noting the origin of the error term E_2, we may rewrite equation (4-18) as

$$y_{RW}(nT) = aw(n_oT+k_oT)x(nT-k_oT-n_oT)+[x(nT-k_oT)(w(nT)-w(k_oT))$$

$$+ ax(nT-k_oT-n_oT)(w(nT)-W(n_oT+k_oT))]*[\delta(nT)-\beta^{-1}\delta(nT+n_oT)$$

$$+ \beta^{-2}\delta(nT+2n_oT)...] = w(nT)ax(nT-k_oT-n_oT)+[x(nT-k_oT)(w(nT)$$

$$- w(k_oT))]*[\delta(nT)-\beta^{-1}\delta(nT+n_oT)+\beta^{-2}\delta(nT+2n_oT)...]$$

$$+ [x(nT-k_oT-n_oT)(w(nT)-w(n_oT+k_oT))]*[-\beta^{-1}\delta(nT+n_oT)$$

$$+ \beta^{-2}\delta(nT+2n_oT)...] \quad . \qquad (4\text{-}19)$$

After correction for windowing, and a few simple manipulations, equation (4-19) becomes

$$y_R(nT) = ax(nT-k_oT-n_oT)+x(nT-k_oT)(1 - \frac{w(k_oT)}{w(nT)}$$

$$- \beta^{-1}\frac{w(nT+n_oT)}{w(nT)} + \beta^{-1}\frac{w(n_oT+k_oT)}{w(nT)})$$

$$- \beta^{-1}\, x(nT+n_oT-k_oT)(\frac{w(nT+n_oT)}{w(nT)} - \frac{w(k_oT)}{w(nT)}$$

$$- \beta^{-1}\frac{w(nT+2n_oT)}{w(nT)} + \beta^{-1}\frac{w(n_oT+k_oT)}{w(nT)}) + \ldots \qquad (4\text{-}20)$$

Thus the recovered wavelet consists of the echo plus a number of distorted and shifted replicas of the basic wavelet. The above analysis agrees well with the experimental results. For the case considered (n_o=30, a=.5 with the Hamming window), β=1.25. Thus peaks are expected at negative quefrencies and since $\frac{1}{\beta}$ = .8>a, these peaks should be larger than for the corresponding rectangularly windowed case. The echo can be recovered by filtering these peaks from the complex cepstrum. The distortions suggested by equation (4-20) are evident in the recovered wavelet shown in Figure 28. The recovered

Example

Hamming Windowed Data Record

Figures 22 through 28

This group of figures illustrates the computation of the cepstra, and wavelet recovery when the input data record is Hamming windowed. Note the distortions present in the recovered wavelet.

The composite signal (256 points) is

$$y(nT) = x(nT) + .5x(nT-30T)$$

where $x(nT) = nT\, e^{-nT}$ $\quad 0 \leq n < 64$

Figure Number	Figure Title
22	Composite Signal SNR = 40 dB
23	Unwrapped Phase Curve
24	Log Magnitude
25	Complex Cepstrum
26	Phase Cepstrum
27	Power Cepstrum
28	Recovered Wavelet MSE = 4.24×10^{-3}

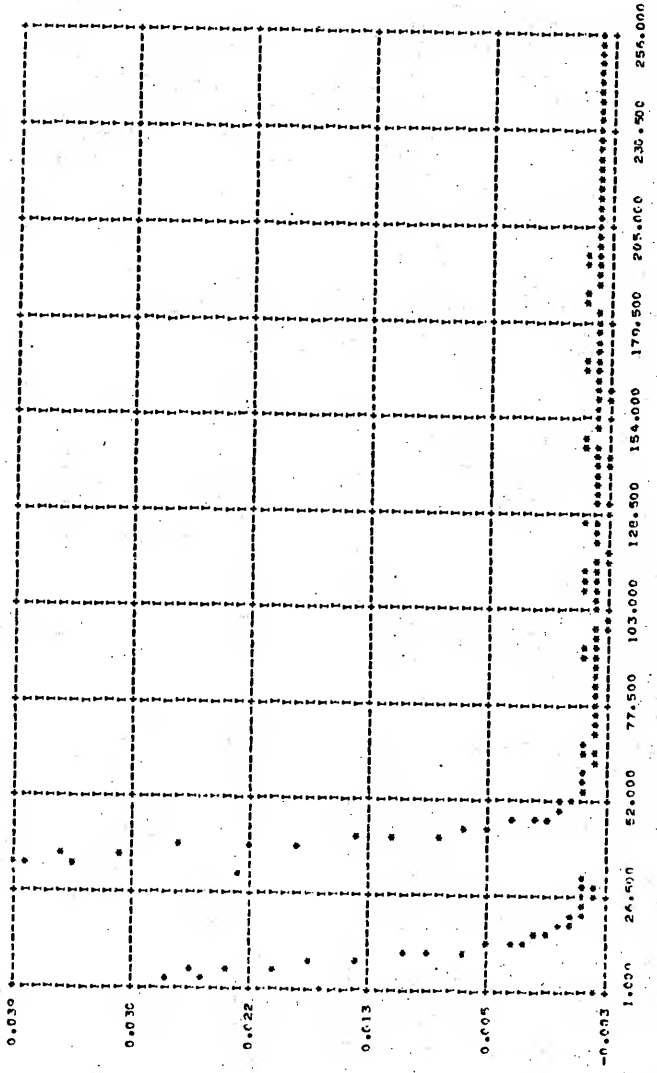


Figure 22 Composite Signal
SNR = 40 dB

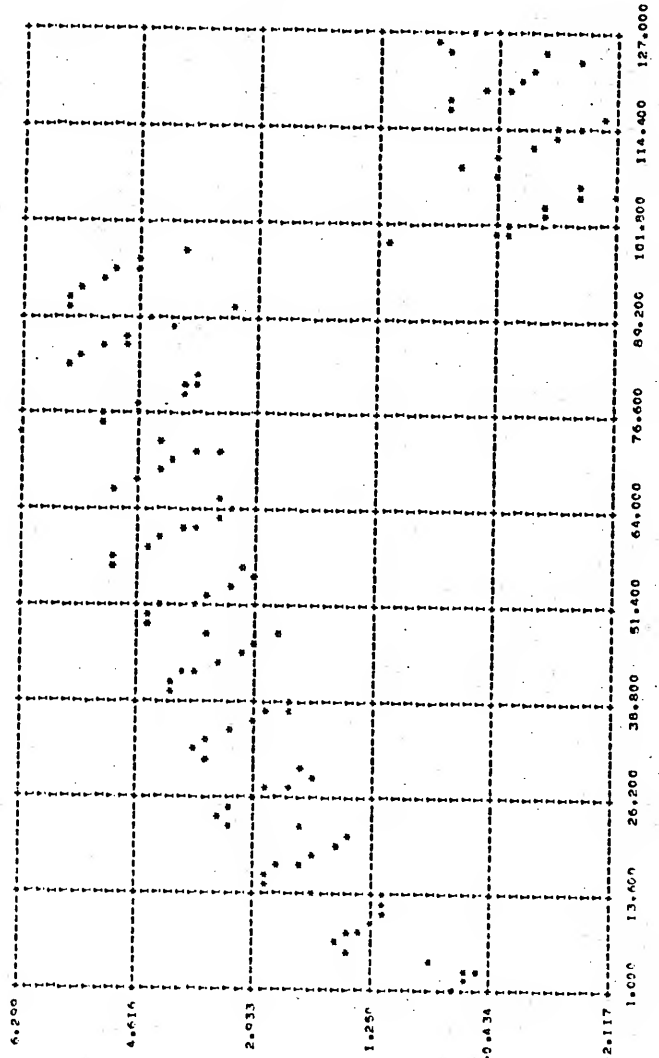


Figure 23 Upwrapped Phase Curve

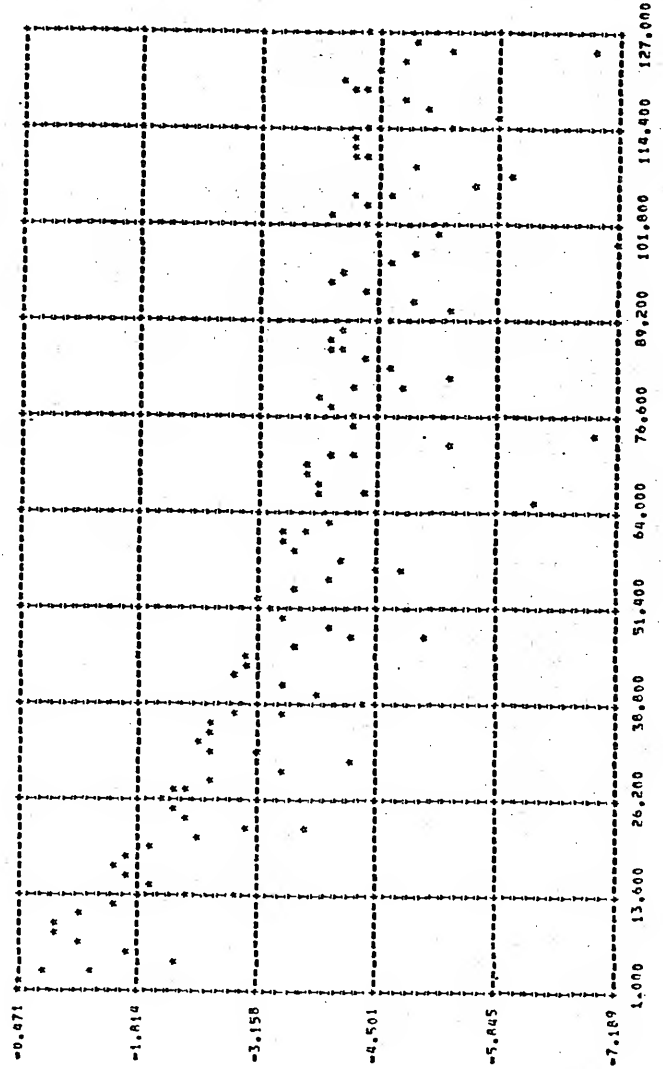


Figure 24 Log Magnitude

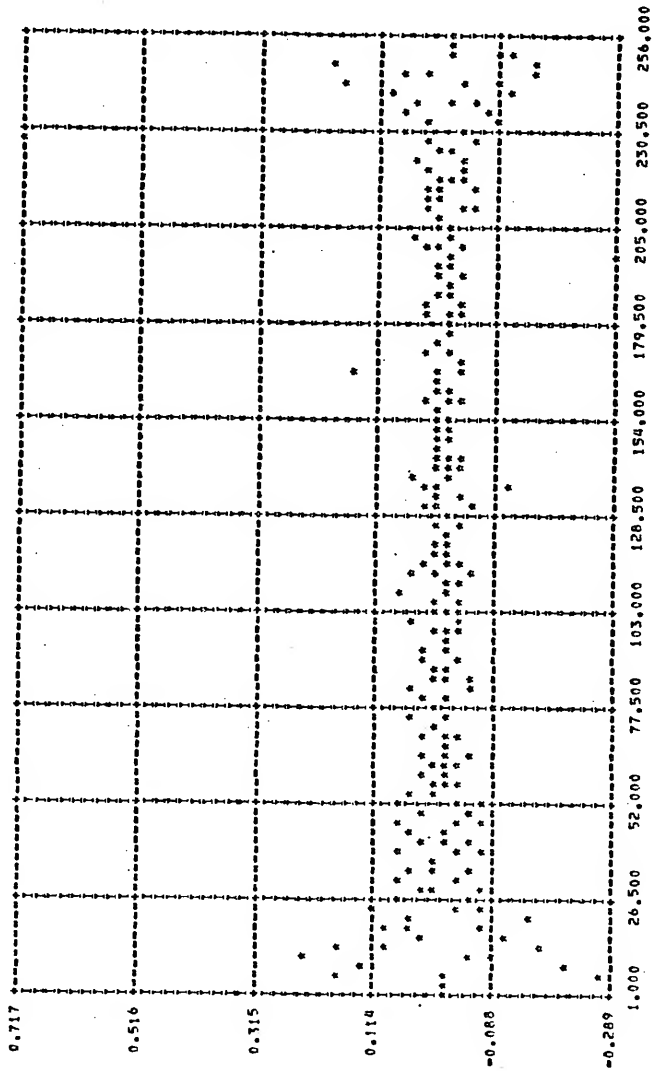


Figure 25 Complex Cepstrum

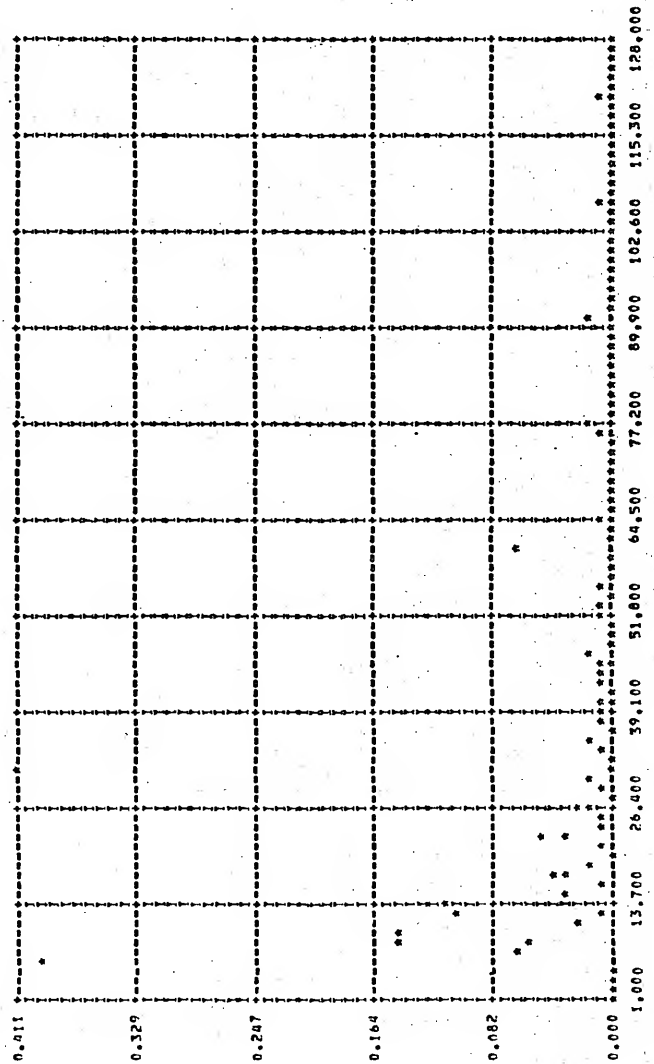


Figure 26 Phase Cepstrum

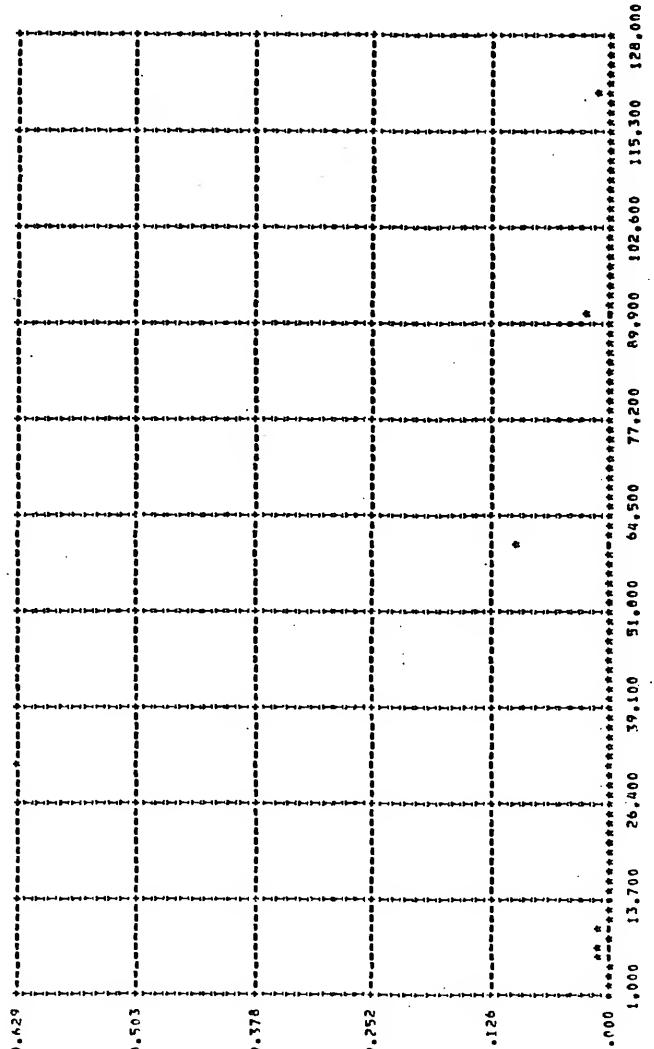


Figure 27 Power Cepstrum

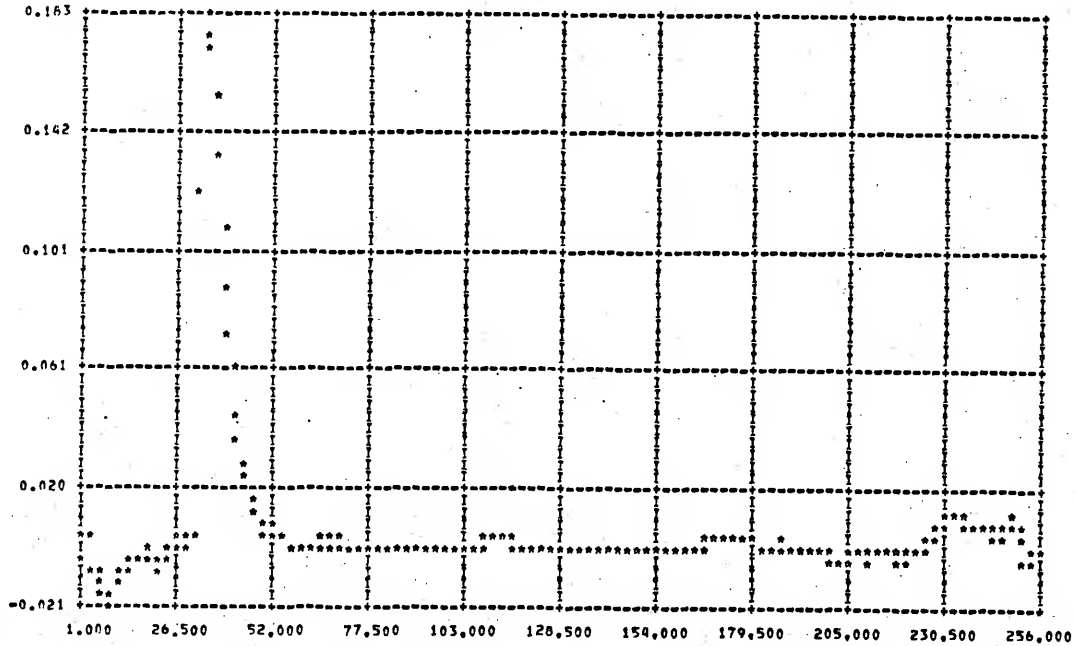


Figure 28 Recovered Wavelet
MSE = 4.24×10^{-5}

wavelet of Figure 28 is a result of smoothing only the peaks in the complex cepstrum. When not only the peaks but several adjacent points were smoothed the distortions clearly diminished. It is to be noted from equation (4-18) that if E_2 vanishes then the echo may be recovered exactly; thus E_2 is indeed an error term.

If β is less than one, a parallel analysis to that given above predicts peaks at positive quefrencies, and reveals that a distorted version of the basic wavelet rather than the echo will be recovered.

The theoretical analysis presented above gives considerable insight into the wavelet recovery process when the window is approximately constant over the duration of the basic wavelet. A second analysis is presented below which gives further insight into the effects of windowing on epoch detection and wavelet recovery under different conditions.

Consider equation (4-9), it may be rewritten as

$$Y(z) = X_w(z)+az^{-n_o}\sum_{n=0}^{L-1} x(nT-k_oT)w(nT)z^{-n}+E_1+E_3 \qquad (4\text{-}21)$$

$$\text{where } E_3 = az^{-n_o}\sum_{n=0}^{L-1} x(nT-k_oT)(w(nT+n_oT)-w(nT))z^{-n} \qquad (4\text{-}22)$$

$$\text{thus } Y(z) = X_w(z)(1+az^{-n_o})+E_1+E_3 . \qquad (4\text{-}23)$$

Both E_1 and E_3 are considered error terms in that if they are not present, then one could obtain the complex cepstrum, remove the contributions due to the echo impulse train, perform the inverse complex cepstrum operation, correct for the windowing by multiplying the recovered sequence by $(w(nT))^{-1}$ and obtain the basic wavelet.

The error term E_1 has already been discussed in the section on the exponential window, and will again be assumed to vanish. A close examination of the error E_3 reveals it arises from two distinct sources. E_3 can be written as

$$E_3 = az^{-n_o} \sum_{n=0}^{L-n_o-1} x(nT-k_oT)[w(nT+n_oT)-w(nT)]z^{-n} + az^{-n_o} \sum_{n=L-n_o}^{L-1} x(nT-k_oT)w(nT)z^{-n}.$$

The first term in the equation above represents an error introduced by the window because the window weights the basic wavelet and echo differently, the second term is again the truncation error due to the fact that the echo may extend beyond the record under consideration, and as in the section on the exponential window it is neglected. This leaves only the windowing error to be considered; obviously E_3 vanishes when $w(nT)=w(nT+n_oT)$. This again suggests the rectangular window for wavelet extraction. Other common windows for should be acceptable, provided $x(nT-k_oT)(w(nT+n_oT)-w(nT))$ is small. For most commonly used windows (Hamming, Hanning, etc.) this requires n_o to be small (i.e., the delay between the basic wavelet and echo must be short). Carrying the analysis further we may rewrite equation (4-23) as

$$Y(z) = (X_w(z) + \frac{E_3}{1+az^{-n_o}})(1+az^{-n_o}) \qquad (4\text{-}24)$$

thus

$$\hat{Y}(z) = \log Y(z) = \log(X_w(z)+ \frac{E_3}{1+az^{-n_o}}) + \log (1+az^{-n_o}) . \qquad (4\text{-}25)$$

From equation (4-25) we see that the side of the complex cepstrum

on which the echo peaks occur will be determined by the echo amplitude (a) just as in the rectangularly windowed case.

Assuming that the term log $(1+az^{-n_o})$ can be filtered out, we can extract the wavelet by (assuming a<1)

$$Y_{RW}(nT) = Z^{-1}\left(X_w(z) + \frac{E_3}{1+az^{-n_o}}\right)$$

$$= Z^{-1}(X_w(z) + E_3(1-az^{-n_o}+a^2z^{-2n_o}\ldots)) \quad . \tag{4-26}$$

$$Y_{RW}(nT) = w(nT)x(nT-k_oT)+[ax(nT-k_oT-n_oT)(w(nT+n_oT)-w(nT))]$$

$$*[\delta(nT)-a\delta(nT-n_oT)+a^2\delta(nT-2n_oT)\ldots] \quad . \tag{4-27}$$

Multiplying (4-27) by $(w(nT))^{-1}$, we obtain

$$Y_R(nT) = x(nT-k_oT) + [ax(nT+k_oT-n_oT)(\frac{w(nT+n_oT)}{w(nT)} -1)]-a^2x(nT+k_oT-2n_oT)$$

$$(1- \frac{w(nT-n_oT)}{w(nT)}) + \ldots. \tag{4-28}$$

Thus the basic wavelet is recovered though it is distorted by the windowing error terms. A similar derivation applies if a>1, yielding recovery of the echo rather than the basic wavelet, but with similar distortion terms.

From the two analyses considered it is evident that there are two cases when we may expect satisfactory wavelet recovery after windowing with one of the windows commonly used to reduce leakage:

(1) When w(nT) is relatively constant over the duration of the basic wavelet. This requires the signal duration to be short compared to the total record length. The experimental results presented for the signal of the form given in equations (4-4) and (4-5)

(with b=1.0) are an example of this case. Even though the actual signal duration is a considerable portion of the total record length, 90% of the signal energy lies in a region only 3 or 4% of the total record length. Thus the effective duration is quite short. The first theoretical analysis presented accurately predicts the results of these experiments.

(2) When the window w(nT) is approximately constant over the echo delay n_o, that is $\frac{w(nT+n_oT)}{w(nT)} \simeq 1$. This requires n_o to be short compared to the total record length. Additional computer computations were conducted to verify the findings of the second analysis. Utilizing the signal given in equations (4-4) and (4-5) with b=.1, it is found that echo epoch detection and successful wavelet recovery (in the no noise case) are possible for echo delays as short as 5 sample times when the rectangular window is used. When the same signal is Hamming windowed, echo detection and satisfactory wavelet recovery are possible for small echo delays ($n_o<10$) though even for n_o equal 5 the MSE is considerably greater for this window than for the rectangular window.

It should be noted that no assumptions (other than regarding the magnitudes of a and β) were made in the analyses presented until the filtering of the peaks in the complex cepstrum was considered. Essentially both analyses are correct up to this point though the nature and magnitude of the error terms differ. This leads to the observation that when $a<1$ and $\beta>1$, and neither approximation is valid, peaks should be present at both positive and negative quefrencies.

One set is predicted by the first theoretical analysis and the other set is predicted by the second analysis. This is precisely what was observed in many of the experimental outputs.

Conclusions

The exponential window

Windowing the input data with the exponential window was expected from theoretical considerations to perform as well as the rectangular window in the no noise case (and somewhat better if echo truncation is a problem). Over the duration of the signal this is verified by the experimental results at high SNR; however, correction for the exponential window introduces some distortion into the recovered record outside the signal duration. At low SNR the exponential window performed better than the rectangular window. This is undoubtedly due to the fact that the composite signal examined occurred at the beginning of the data record and thus the exponential window tends to reduce the noise content of the total record much more than the signal content. Correspondingly when the composite signal occurs near the end of the data record, the exponential window degrades the recovered wavelet. It is similarly noted that while the echo epoch peak in the complex cepstrum is diminished, the detection threshold is unchanged when the composite signal occurs near the beginning of the record but increases when the composite signal occurs later in the record. Finally it was noted that the exponential window may be used to reduce aliasing of the echo impulse train, and to prevent ambiguous peaks in the multiple echo case. Unless echo truncation

or one of the problems cited above is present, an exponential windowing of the input data sequence appears to offer no advantages over the rectangular window.

The common windows

Application of the Hamming, Hanning, and Tapering windows is very detrimental to wavelet recovery. Satisfactory recovery is not possible unless:

(1) the echo delay is small compared with the total record, or

(2) the duration of the basic wavelet is short compared with the record length.

Even in these cases wavelet recovery is degraded. It is interesting to note that in spite of the distortion caused by windowing, echo epoch detection is always possible (at least in the no noise case) and that while peaks may be introduced on both sides of the complex cepstrum, no shifting of the peaks from the proper echo epoch is present. Windowing does however, increase the SNR at which echo epoch determination is possible. This indicates that cepstral techniques may be useful in resolving signals of similar, but not identical, shapes. The theoretical analyses presented predict the errors produced in the recovered wavelet by windowing with the Hamming window quite well. A similar analysis may give some insight into the distortion introduced in the recovered wavelet when the echo is truncated.

WINDOWING THE LOG SPECTRUM

Since the complex cepstrum contains well defined peaks, it seems reasonable that windowing the log spectrum be considered to

reduce the leakage due to these peaks. Only the effect of the Hamming window on the log spectrum is studied, as this window seems representative of the windows used to reduce leakage and it is nonzero at its endpoints making correction less difficult. After windowing the log spectrum, the complex cepstrum is computed and the peaks present due to the echo are removed ; this sequence is then inverse transformed to obtain the recovered log spectrum which is (unless otherwise noted) corrected by multiplying the recovered log spectrum by the inverse of the windowing sequence.

The following observations are made on the effects of windowing the log spectrum.

(1) A spreading of the peaks in the complex cepstrum is noted; e.g., a single positive peak is reduced in amplitude and has two adjacent negative peaks.

(2) Empirically, it is found that the "filtering" of the complex cepstrum is critical. Smoothing the complex cepstrum at single peak points as used previously proves to be totally inadequate, as the recovered wavelet is virtually unrecognizable even when no noise is present. When both the peak and the two points adjacent to it are smoothed, recovery is possible. Figure 29 compares the MSE of the recovered wavelet (when adjacent points are smoothed) to that obtained when only rectangular windowing of the log spectrum is used. As can be seen for the no noise case, windowing the log spectrum results in about the same MSE as for the rectangularly windowed case, but performance rapidly deteriorates when noise is added. The wavelet becomes unrecoverable at a SNR of 14 dB. The echo detectability threshold (20 dB) is also increased by 12 dB over the rectangularly windowed case.

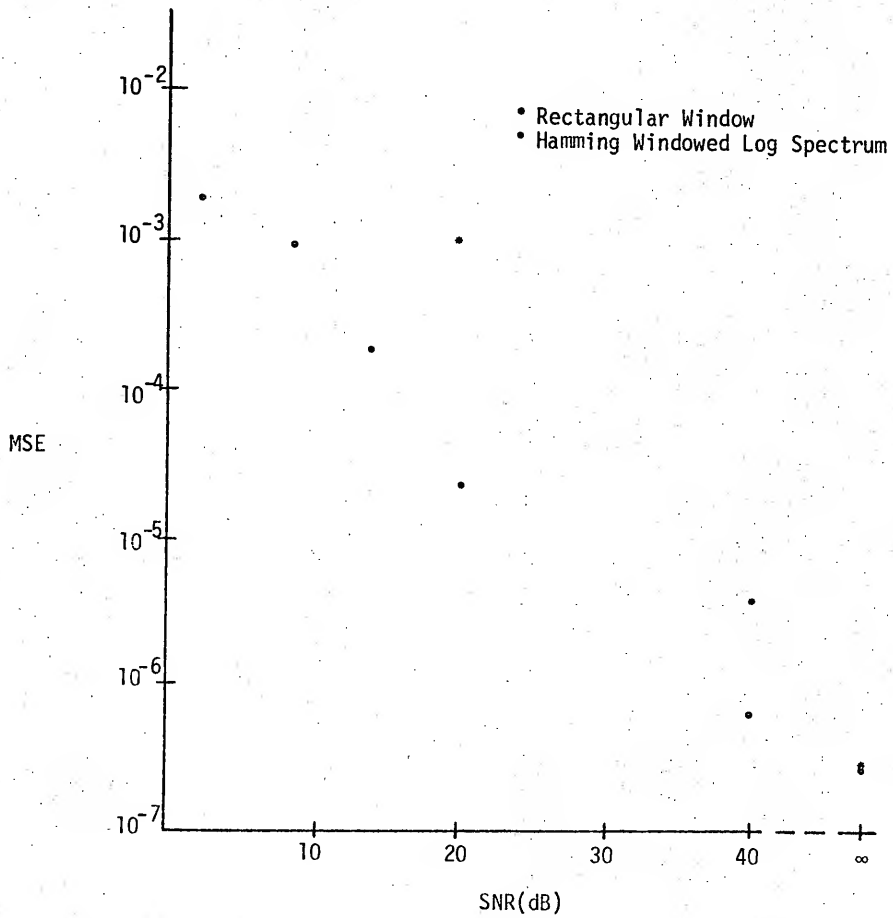


Figure 29 MSE of the Recovered Wavelet when the Log Spectrum Is Hamming Windowed.

(3) Several runs have been made with no correction of the recovered log spectrum, and it was found that wavelet recovery is impossible if correction is not made.

Interpretation of the Results

Let us consider the effects of windowing the log spectrum in the single echo case.

$$\hat{Y}(z) = \log Y(z) = \log X(z) + \log (1+az^{-n_o}) . \qquad (4\text{-}29)$$

Windowing $\hat{Y}(z)$, we obtain

$$W(z)\hat{Y}(z) = W(z) \log (1+az^{-n_o}) + W(z) \log X(z) .$$

where $W(z)$ is the window.

Assuming $a<1$, so that we may expand the logarithm in the familiar power series, and inverse z-transforming equation (4-29), we obtain the complex cepstrum.

$$\hat{y}_w(nT) = w(nT)*\hat{x}(nT)+w(nT)*(a\delta(n-n_o)- \frac{a^2}{2} \delta(n-2n_o)+ \frac{a^3}{3} \delta(n-3n_o)\ldots) . \qquad (4\text{-}30)$$

Thus, it is observed that windowing the log spectrum convolves the complex cepstrum with the transform of the window. In the case of the Hamming window this means the complex cepstrum is convolved with the sequence

$$w(nT) = -.22\delta(nT+T)+.56\delta(nT)-.22\delta(nT-T) .$$

This accounts for the spreading of the peaks noted experimentally. If the peaks due to the echo can be "filtered" from the complex cepstrum, then we may obtain the basic wavelet by correcting the recovered

log spectrum and completing the remainder of the wavelet recovery algorithm. It is observed that the contributions in the complex cepstrum due to the echo are no longer isolated peaks but in fact extend to the points immediately adjacent to the main peaks thus we expect that "filtering" over a single peak will be inadequate for wavelet recovery as is observed.

The rapid deterioration of the MSE with increasing SNR is probably also connected with the filtering problem but this is not well understood. The increase in the echo detectability threshold is undoubtedly due to the fact that the peaks are reduced in height by convolution with the sequence given above.

Conclusions

While windowing the log spectrum with the Hamming window reduces leakage, it is apparent that other factors (principally the problem with "filtering" the complex cepstrum) tend to degrade wavelet recovery especially at low SNR. Thus windowing the log spectrum with the Hamming window cannot be recommended as part of the general wavelet recovery algorithm. The performance of other similar windows is expected to be the same.

HANNING THE LOG SPECTRUM

It has been reported that the Hanning smoothing of the log magnitude (that is, the real part of the log spectrum) results in a decrease in both the MSE and the echo epoch detectability threshold in the presence of additive noise [3]. Since this is closely related to windowing the complex cepstrum, it is appropriate to discuss this topic at this time. Smoothing consists of convolving the sequence to

be smoothed with a smoothing sequence (e.g., the Hanning smoothing sequence is .25, .50, and .25). Smoothing is ordinarily used after an FFT to reduce leakage, as this is equivalent to windowing the input prior to the FFT with a Hanning window (convolution in the frequency domain is equivalent to multiplication in the time domain). In the present context it serves a quite different function, smoothing the log spectrum is essentially a frequency invariant low (short) pass filtering operation. Consider the single additive echo case.

$$\hat{Y}_s(z) = [\log X(z) + \log (1+az^{-n_o})]*W(z) \qquad (4\text{-}31)$$

where W(z) is the smoothing function or equivalently it may be regarded as the impulse response of a filter. Inverse z-transforming the above equation, we find

$$\hat{y}_s(nT) = (\hat{x}(nT)+\hat{e}(nT))w(nT) \qquad (4\text{-}32)$$

where $\hat{x}(nT)$ is the complex cepstrum of the basic wavelet, $\hat{e}(nT)$ is the train of δ functions due to the presence of an echo, and w(nT) is the inverse z-transform of the smoothing function.

For the Hanning weights given above, $w(nT) = \frac{1}{2}(1+\cos\frac{2\pi n}{N})$. Thus we see that Hanning smoothing tends to reduce the high quefrency (near $\frac{NT}{2}$) contributions to the complex cepstrum, and is equivalent to windowing the complex cepstrum.

If only the log magnitude is smoothed, then

$$\hat{Y}_s(z) = \mathrm{Re}(\log X(z)+ \log(1+az^{-n_o}))*W(z)+j\mathrm{Im}(\log X(z)+ \log(1+az^{-n_o})$$

$$= (\log|X(z)|+ \log|1+az^{-n_o}|)*W(z)+j(\text{Phase } X(z)+\text{Phase } (\log(1+az^{-n_o}))). \qquad (4\text{-}33)$$

Evaluating the inverse z-transform on the unit circle $z=e^{j\omega T}$ and noting that

$$\mathrm{Re}\hat{Y}(z) = \log|X(e^{j\omega T})| + \log(1+a^2+2a\cos\omega n_0 T) \tag{4-34}$$

is an even function of ω, and

$$j\mathrm{Im}\hat{Y}(z) = j\mathrm{Phase}\, X(e^{j\omega T}) + j\tan^{-1}\left(-\frac{a\sin(n_0\omega T)}{1+a\cos(n_0\omega T)}\right) \tag{4-35}$$

is an odd function of ω, we see that the transform of equation (4-34) is the even part of the complex cepstrum $\hat{y}(nT)$ (i.e., it is $\hat{y}_e(nT) = \frac{1}{2}(\hat{y}(nT) + \hat{y}(-nT))$ and the transform of equation (4-35) is the odd part of the complex cepstrum $\hat{y}(nT)$ (i.e., it is $\hat{y}_o(nT) = \frac{1}{2}(\hat{y}(nT)-\hat{y}(-nT))$). Thus the smoothed complex cepstrum is

$$\hat{y}_s(nT) = \hat{y}_e(nT)w(nT) + \hat{y}_o(nT) \tag{4-36}$$

but

$$\hat{y}_e(nT) = \frac{1}{2}(\hat{y}(nT)+\hat{y}(-nT)) = \frac{1}{2}(\hat{x}(nT)+\hat{x}(-nT)+\hat{e}(nT)+\hat{e}(-nT)) \tag{4-37}$$

and

$$\hat{y}_o(nT) = \frac{1}{2}(\hat{y}(nT)-\hat{y}(-nT)) = \frac{1}{2}(\hat{x}(nT)-\hat{x}(-nT)+\hat{e}(nT)-\hat{e}(-nT)) \tag{4-38}$$

where $\hat{e}(nT)$ is the train of impulses in the complex cepstrum due to the echo. From the above analysis it is clear that the log magnitude terms produce δ functions on both sides of the origin, and the log phase terms produce δ functions which reinforce on one side of the origin and cancel on the other to produce the one-sided $\hat{e}(nT)$. When

the log magnitude is smoothed the δ functions produced by it are reduced by the window w(nT); thus they no longer cancel on one side with the δ functions due to the phase portion, and peaks may be expected on both sides of the origin (though the peaks on one side should be substantially greater). Thus we observe that a minimum phase sequence (normally zero on the negative side of the complex cepstrum) may produce contributions on both sides of the complex cepstrum if the log magnitude alone is smoothed. If both the log magnitude and phase are smoothed their contributions will be reduced equally and thus they will still cancel completely on the negative side of the complex cepstrum.

Experimental Results

A Hanning smoothing performed on the log magnitude and phase produces a windowing of the complex cepstrum as expected. For the case considered (n_0=30) the echo epoch detection threshold appears to be slightly lower than for the unsmoothed case. This is undoubtedly due to the fact that the primary echo epoch peak is only slightly reduced by the window while the peaks associated with noise located at higher quefrencies are considerably reduced. Had the echo epoch been at a higher quefrency (e.g., n_0=100) the detection threshold would have increased greatly. As can be seen from Figure 30, the MSE of the recovered wavelet is considerably reduced by the smoothing process. Only in the no noise case did the unsmoothed process prove to be superior.

Smoothing the log magnitude (only) produced similar results as seen in Figure 30, but this is clearly inferior to smoothing both the log magnitude and phase when additive noise is present. Since the power

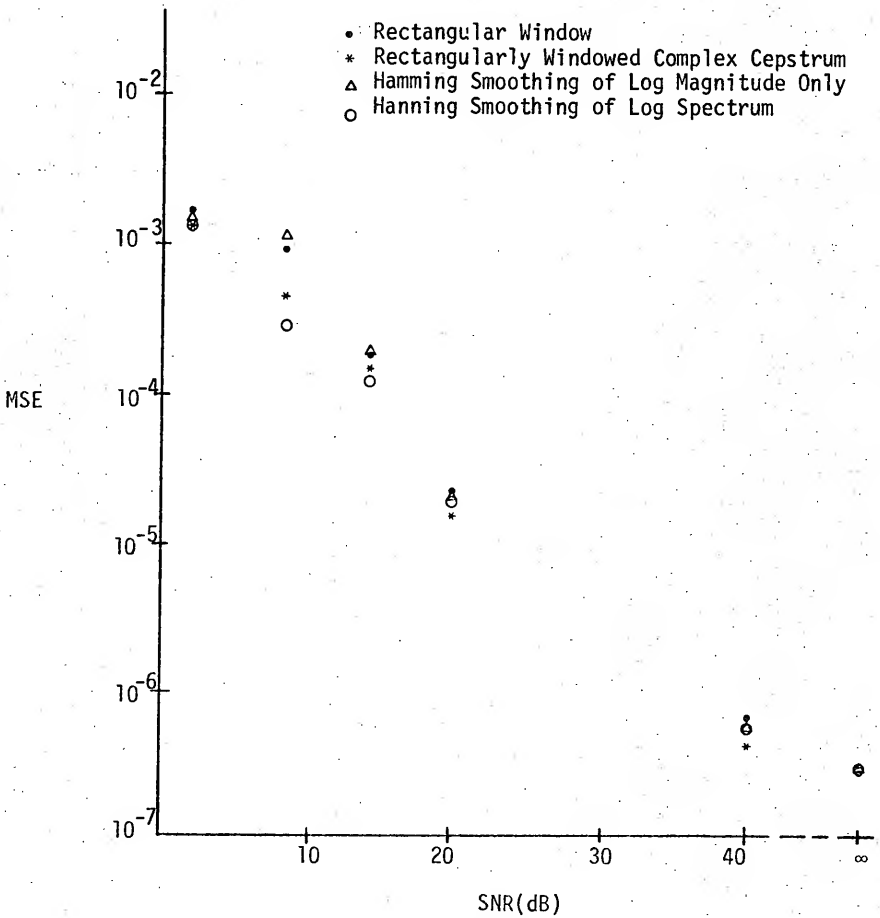


Figure 30 MSE of Recovered Wavelet when the Log Spectrum Is Hanning Smoothed.

cepstrum is independent of phase information, the power cepstrum in this case is identical to the power cepstrum produced above. Kemerait [3] has reported observing echo peaks at both positive and negative quefrencies only when $a>1$. Our results clearly contained echo peaks at positive and negative quefrencies whether $a>1$ or $a<1$. The positive peaks are dominant when $a<1$, and the negative peaks are dominant when $a>1$. This clearly confirms the theoretical analysis presented. Since echo peaks are present at both positive and negative quefrencies the complex cepstrum must be smoothed on both sides. If only the dominant peaks are filtered the MSE of the recovered wavelet is degraded slightly.

An attempt was made to improve the MSE performance by not smoothing the log spectrum at low frequencies (where most of the signal power is present) while continuing to smooth the higher frequency components. This is again essentially a filtering process but in this case the filtering is no longer frequency invariant. The results from this type of smoothing process were found to be considerably worse than those found above, and its use was discontinued.

Conclusions

The Hanning smoothing of the log spectrum is essentially a frequency invariant low (short) pass filtering operation on the log spectrum, or equivalently it may be regarded as a windowing of the complex cepstrum. In this light the effect of smoothing on echo epoch detection is easily understood, as it does in fact simply reduce echo peaks occurring at high quefrencies more than those at low quefrencies.

Since the noise is interspersed throughout the data record, and the signal is limited to only a portion of the data record, it is reasonable that the high quefrency components of the complex cepstrum should contain more noise than signal information and thus the filtering of these from the log spectrum by smoothing improves the MSE of the recovered wavelet. Additional experiments indicate that similar results can be obtained when the complex cepstrum is rectangularly windowed (i.e., portions of the complex cepstrum are zeroed) to eliminate the high quefrency noise contributions. These results are included in Figure 30. For signals having durations approaching the total record length these smoothing techniques may not improve the MSE as much as in the case considered.

THE ADDITION OF ZEROES

The addition of zeroes to the input data record has been shown to be an effective method of improving wavelet extraction and echo detectability [3]. The purpose of this discussion is to clarify, theoretically justify, and extend the results of [3]. Adding zeroes to the input data record necessarily reduces the throughput of the overall wavelet recovery system and thus may sometimes be impossible in a real-time system. In previous discussions z-transform analysis has been heavily relied upon. The z-transform is a function of the continuous complex variable z (ω, when evaluated on the unit circle). Computational considerations lead us to use the sampled z-transform which is equivalent to the DFT. It is in this context that the addition of zeroes to the input data record is meaningful and in which the effects of adding zeroes may be examined (note, the

z-transform (continuous) of a signal extended by zeroes is precisely the same as that of the unextended signal).

In ordinary discrete spectral analysis zeroes are normally added to the data record to correct for the picket fence effect (described in [11]). This improves the frequency resolution of components whose frequency lies in between the frequencies of the DFT coefficients. This probably results in some improvement in cepstral analysis, but it is difficult to ascertain the nature of this improvement. In the analysis below other effects of the addition of zeroes which affect cepstrum analysis, and are in fact crucial in wavelet recovery are discussed.

Consider the z-transform of the sequence $x(nT)$ where $x(nT) = 0$ outside $[0,N)$

$$X(z) = \sum_{n=0}^{N-1} x(nT)z^{-n} \quad . \tag{4-39}$$

Evaluating the z-transform on the unit circle

$$X(e^{j\omega T}) = \sum_{n=0}^{N-1} x(nT)e^{-j\omega nT} \tag{4-40}$$

and sampling it at uniformly spaced intervals around the unit circle, we obtain

$$X_s(e^{j\omega T}) = [\sum_{n=0}^{N-1} x(nT)e^{-j\omega nT}][\sum_{m=-\infty}^{\infty} \delta(\omega - \frac{2\pi m}{NT})] = \sum_{n=0}^{N-1} x(nT)e^{-j\frac{2\pi mn}{N}} \tag{4-41}$$

which is just the DFT of $x(nT)$. Since the logarithm of a sampled function is equivalent to sampling the logarithm of the function,

we find

$$\hat{X}_s(e^{j\omega T}) = \log X_s(e^{j\omega T}) = \log X(e^{j\omega T}) \sum_{m=-\infty}^{\infty} \delta(\omega - \frac{2\pi m}{NT}) \quad . \tag{4-42}$$

Evaluating the inverse z-transform on the unit circle, we obtain

$$\hat{x}_s(nT) = \frac{T}{2\pi j} \int_{-\pi}^{\pi} \log X(e^{j\omega T}) \sum_{m=-\infty}^{\infty} \delta(\omega - \frac{2\pi m}{NT}) e^{j\omega nT} j d\omega \tag{4-43}$$

$$= \frac{T}{2\pi} \sum_{m=0}^{N-1} \log X(e^{j\frac{2\pi m}{N}}) e^{j\frac{2\pi mn}{N}} \tag{4-44}$$

$$= \frac{NT}{2\pi} \text{ (inverse DFT of } \hat{X}_s(e^{j\omega T})) \quad . \tag{4-45}$$

Noting equation (4-42) and the fact that $Z^{-1}(X(z)Y(z)) = x(n)*y(n)$ we may write

$$\hat{x}_s(nT) = \hat{x}(nT) * \frac{T}{2\pi j} \int_{-\pi}^{\pi} \sum_{m=-\infty}^{\infty} \delta(\omega - \frac{2\pi m}{NT}) e^{j\omega nT} j d\omega \tag{4-46}$$

Now since $\sum_{m=-\infty}^{\infty} \delta(\omega - \frac{2\pi m}{NT})$ is a periodic function of ω with period $\frac{2\pi}{NT}$, we may expand it in a Fourier series, thus we obtain

$$\sum_{m=-\infty}^{\infty} \delta(\omega - \frac{2\pi m}{NT}) = \frac{NT}{2\pi} \sum_{m=-\infty}^{\infty} e^{-j\omega mNT}$$

and equation (4-46) becomes

$$\hat{x}_s(nT) = \hat{x}(nT) * \frac{T}{2\pi j} \int_{-\pi}^{\pi} \sum_{m=-\infty}^{\infty} \frac{NT}{2\pi} e^{-j\omega mNT} e^{j\omega nT} j d\omega \tag{4-47}$$

$$= \frac{NT}{2\pi} \hat{x}(nT) * \sum_{m=-\infty}^{\infty} \delta[(n-mN)T] \quad . \tag{4-48}$$

Comparing equations (4-45) and (4-48), we see that

$$\text{the IDFT of } \hat{X}_s(e^{j\omega T}) = \hat{x}(nT) * \sum_{m=-\infty}^{\infty} \delta[(n-mN)T] \qquad (4\text{-}49)$$

where $\hat{x}(nT)$ is the true (not sampled) complex cepstrum of $x(nT)$. Surveying the above discussion, we immediately note that the only effect of adding zeroes to the input data record is to increase the value of N. From equation (4-42) we observe that one of the consequences of increasing the value of N is to sample the log spectrum at smaller intervals. If, for example, we doubled the value of N we would halve the interval between samples of the log spectrum. We also see that the samples of the log spectrum of the extended sequence at m=0,2,4,6... should be precisely the same as the samples of the log spectrum of the original sequence at m=0,1,2,3... respectively, provided that the original sequence is phase unwrapped correctly. As was noted in the section on phase unwrapping it is possible for the phase of $X(e^{j\omega T})$ to change by more than π between samples, and thus for errors to occur in the phase unwrapping. By sampling $X_s(e^{j\omega T})$ at smaller intervals the occurrence of such errors can be reduced. In fact the errors caused by the presence of a strong linear phase component (described in Chapter III) may be eliminated by simply doubling the original record length by the addition of zeroes. From equation (4-48), we see that sampling the z-transform causes the complex cepstrum to fold back onto itself, e.g., quefrencies between $\frac{NT}{2}$ and NT fold into the interval $-\frac{NT}{2}$ to 0. Aliasing of the complex cepstrum can be a serious problem (as observed in Chapter III), but as is seen from equation (4-48), it

may be reduced by making N as large as necessary. The ambiguity problem reported in Chapter III may be eliminated completely, just as the linear phase problem, by doubling the record length with zeroes.

Experimental Results

A composite linear FM signal of the form

$$y(nT) = x(nT) + .5x(nT-30T) \quad 0 \leq n < 256$$

where $x(nT) = \sin((.4 + .11 \cdot nT) \cdot nT) \quad 0 \leq n < 64$ is generated and noise is added. The record length is doubled by the addition of zeroes, the cepstra are computed, and wavelet recovery is performed. The log spectrum, cepstra, and MSE of the recovered wavelet are compared with the results obtained when no zeroes were added. A linear FM signal is used to demonstrate the applicability of cepstrum techniques to signals more complicated than the simple exponential considered so far, and because it may be of more interest in a sonar application. The following observations are made.

(1) As expected the samples of the log magnitude of the unextended record are identical with the even numbered samples of the extended record. Similarly the log phases are identical at high SNR, but differ somewhat for low (14dB or less) SNR. As can be seen by comparing Figures 32 and 39 the phase curves match over about three quarters of the frequency spectrum but are somewhat different near the folding frequency. This difference in the phase curves always begins outside the signal band (for the SNR's observed), and thus the log spectrum at these points is dominated by noise. This

Example

Composite Linear FM Signal

Figures 31 through 37

This group of figures illustrates the computation of the cepstra, and wavelet recovery when the composite signal is a linear FM signal with a single additive echo.

The composite signal (256 points) is

$$y(nT) = x(nT) + .5x(nT-30T)$$

where $x(nT) = \sin((.4 + .11nT)nT) \quad 0 \leq n < 64$

Figure Number	Figure Title
31	Composite Signal SNR = 14 dB
32	Unwrapped Phase Curve
33	Log Magnitude
34	Complex Cepstrum
35	Phase Cepstrum
36	Power Cepstrum
37	Recovered Wavelet MSE = 7.93×10^{-3}

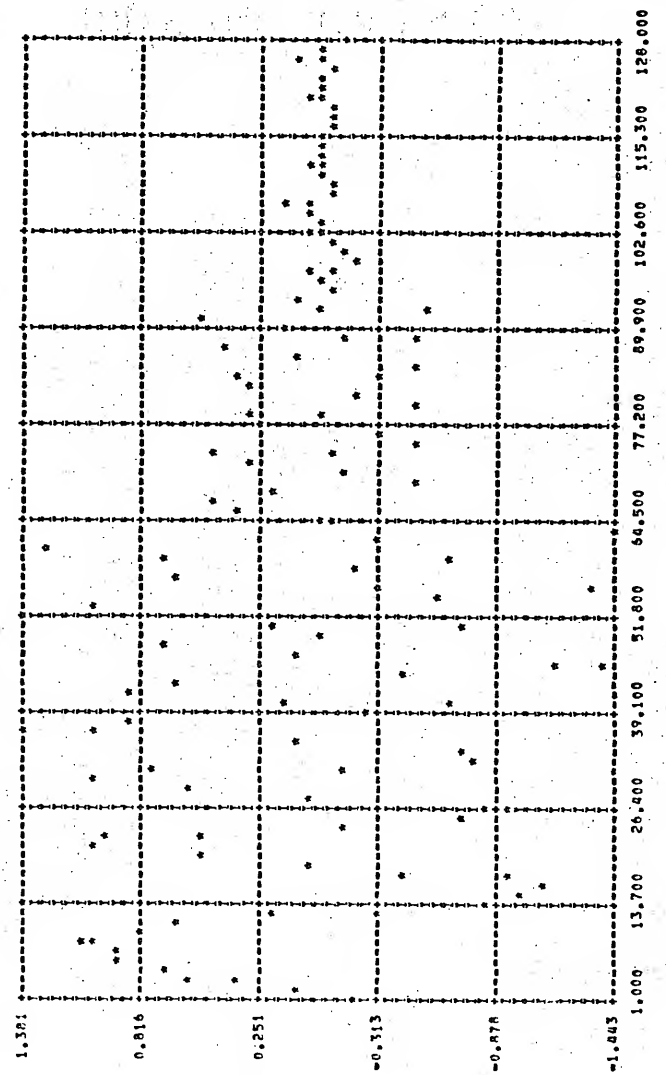


Figure 31 Composite Signal
SNR=14 dB

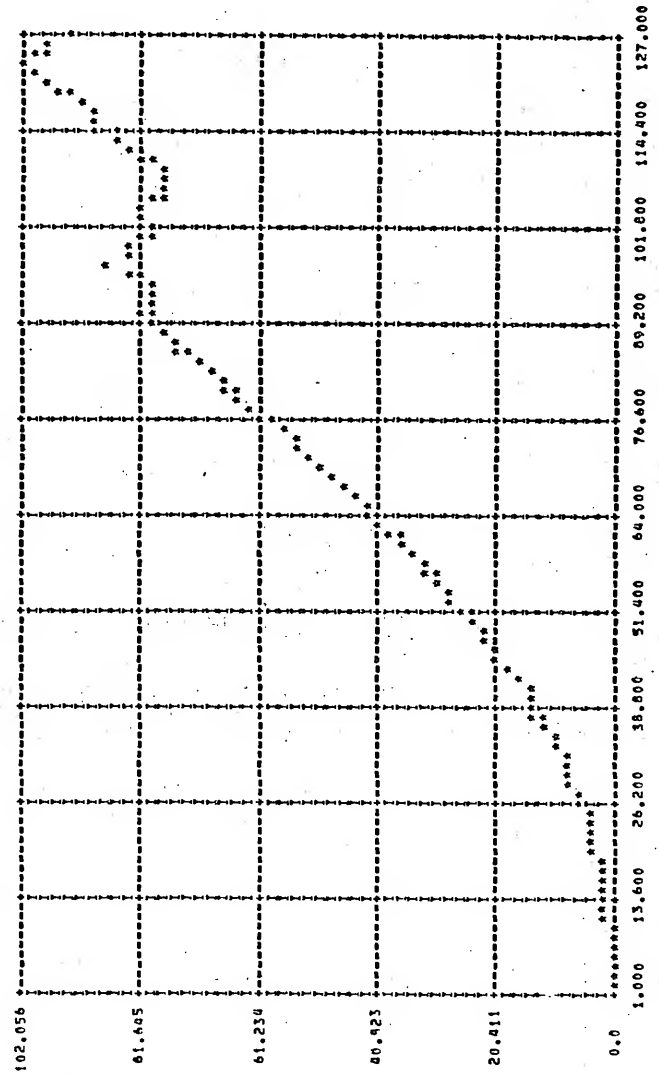


Figure 32 Unwrapped Phase Curve

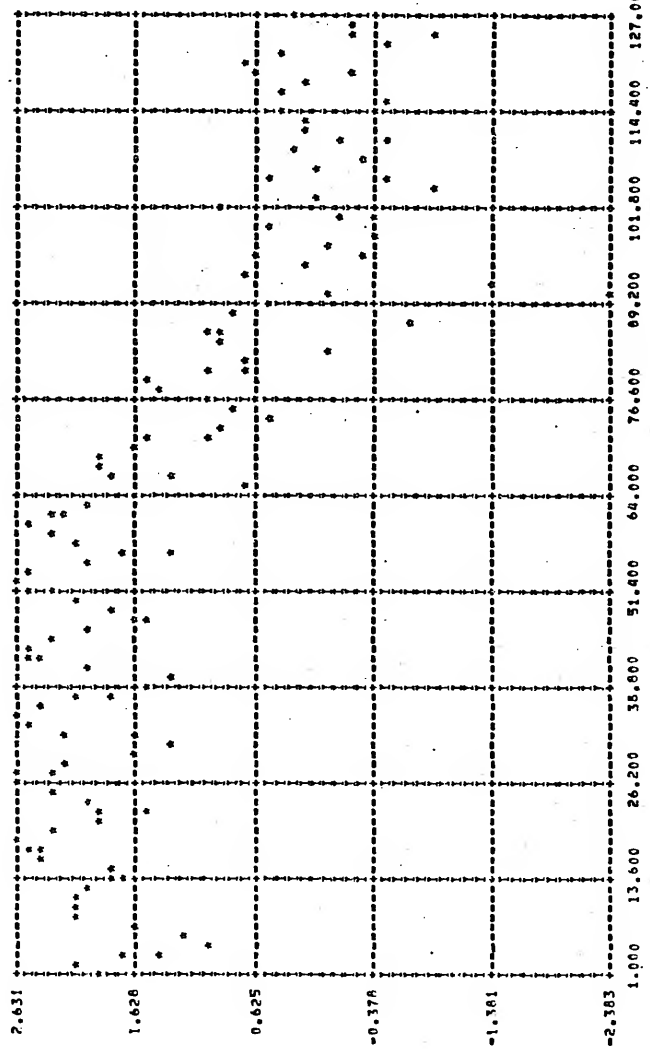


Figure 33 Log Magnitude

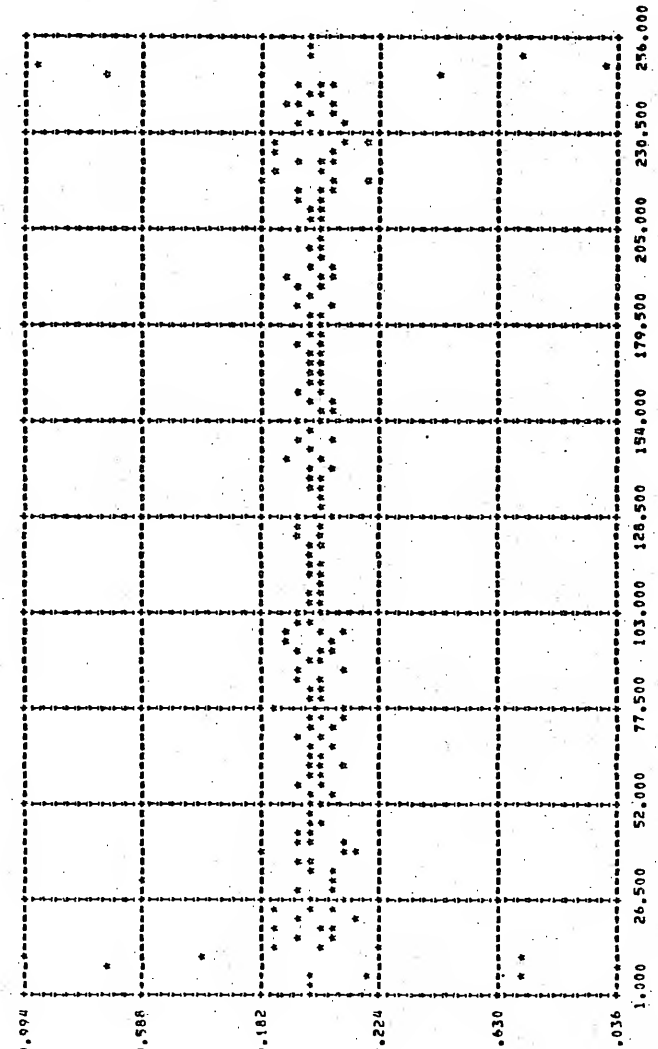


Figure 34 Complex Cepstrum

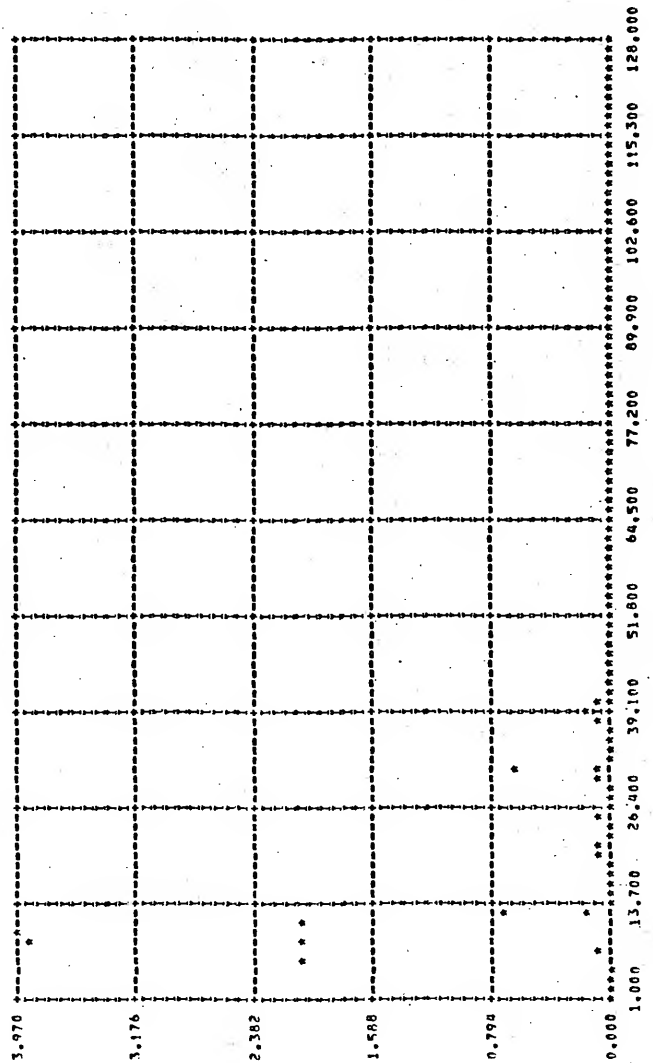


Figure 35 Phase Cepstrum

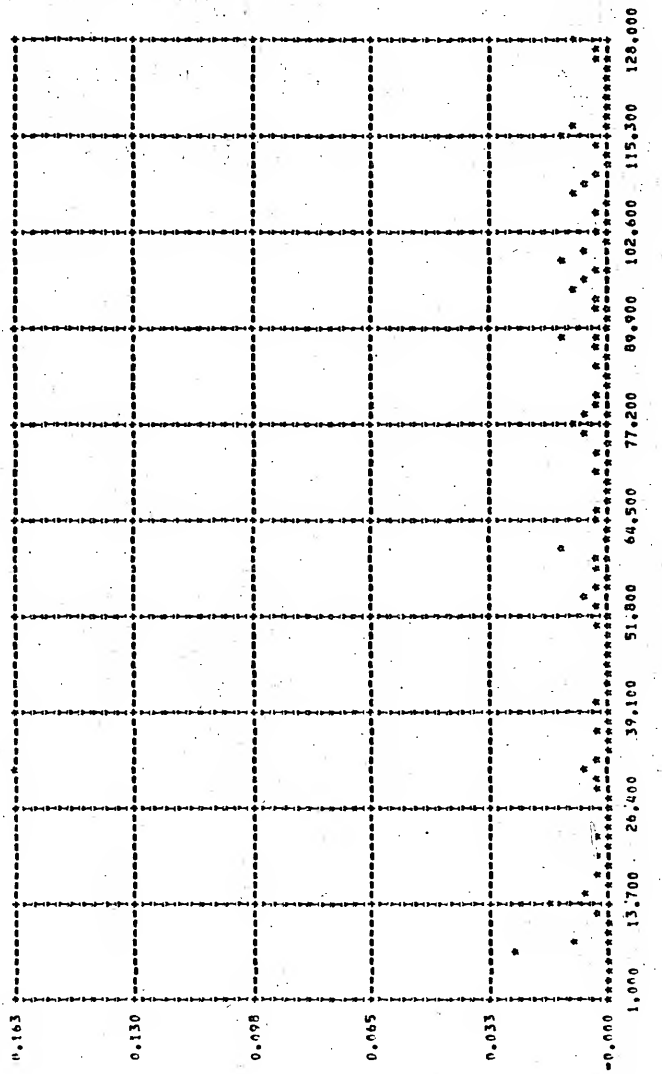


Figure 36 Power Cepstrum

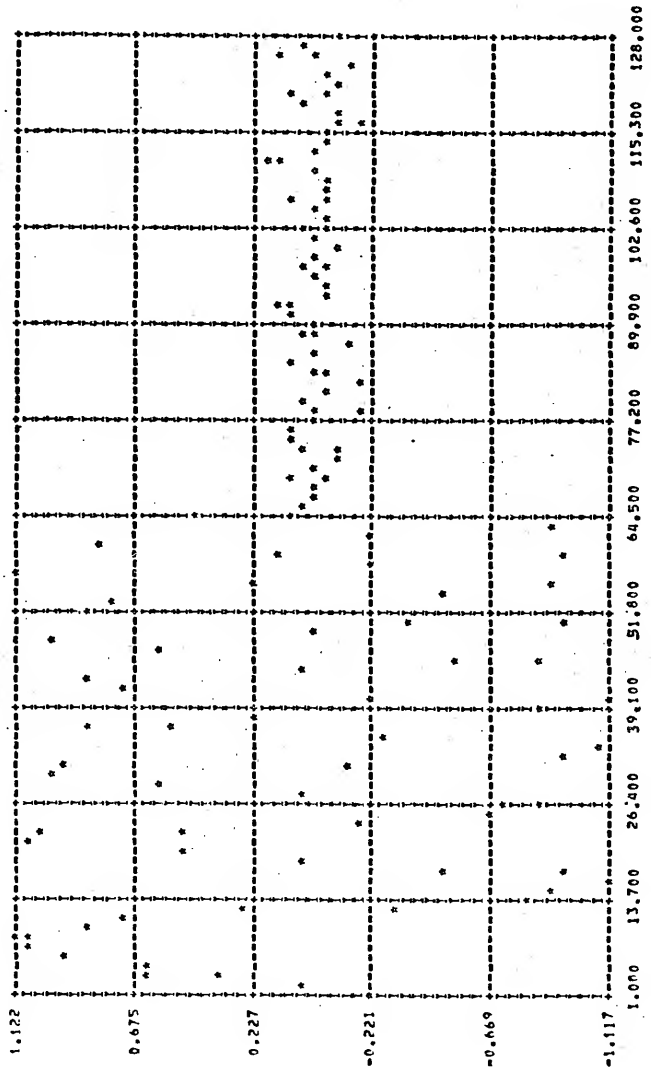


Figure 37 Recovered Wavelet
MSE = 7.93×10^{-3}

Example

Data Record Extended With Zeroes

Figures 38 through 44

This group of figures illustrates the computation of the cepstra, and wavelet recovery when the data record is extended by adding 256 points.

The composite signal (256 points) is

$$y(nT) = x(nT) + .5x(nT-30T)$$

where $x(nT) = \sin((.4 + .11nT)nT) \quad 0 \leq n < 64$

Figure Number	Figure Title
38	Composite Signal (Zeroes Added, 512 Points) SNR = 14 dB
39	Unwrapped Phase Curve
40	Log Magnitude
41	Complex Cepstrum
42	Phase Cepstrum
43	Power Cepstrum
44	Recovered Wavelet MSE = 6.31×10^{-3}

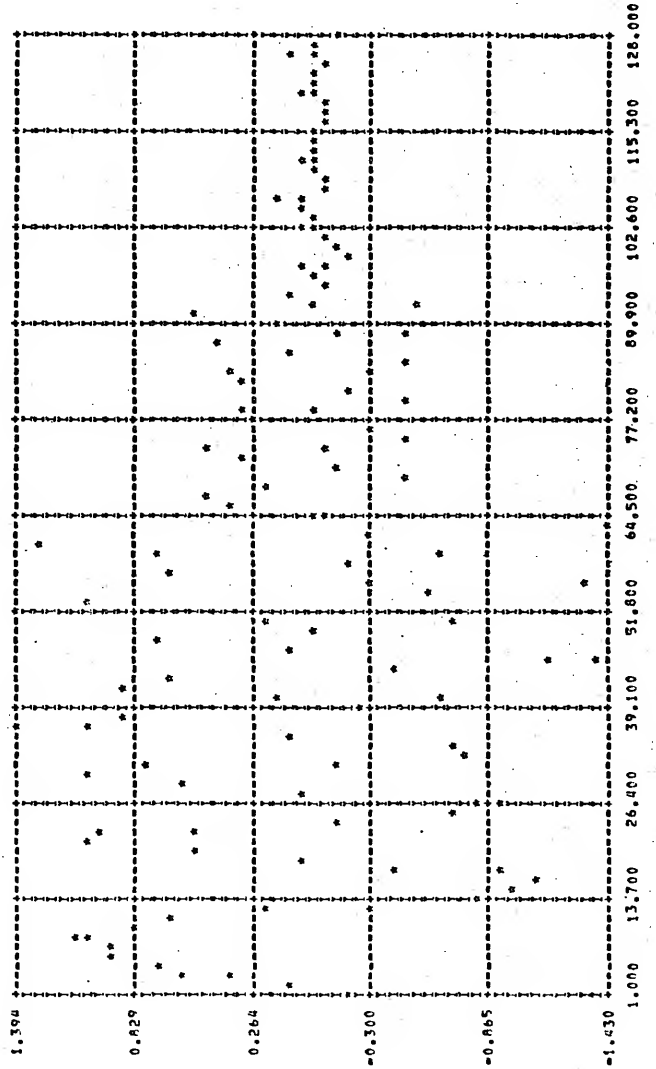


Figure 38 Composite Signal (Zeroes Added, 512 Points)
SNR = 14 dB

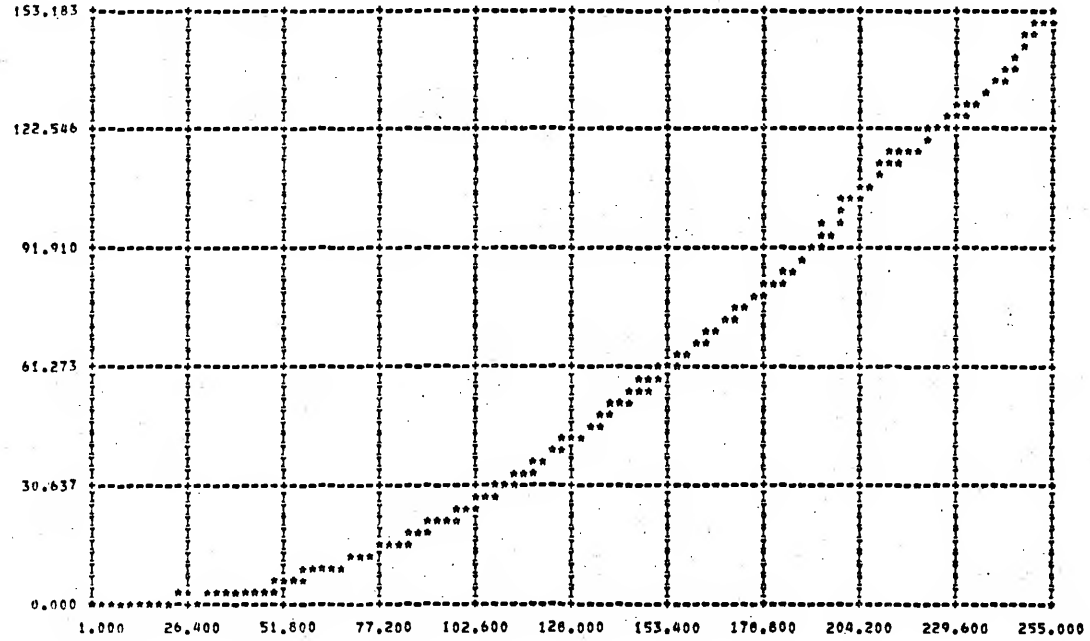


Figure 39 Upwrapped Phase Curve

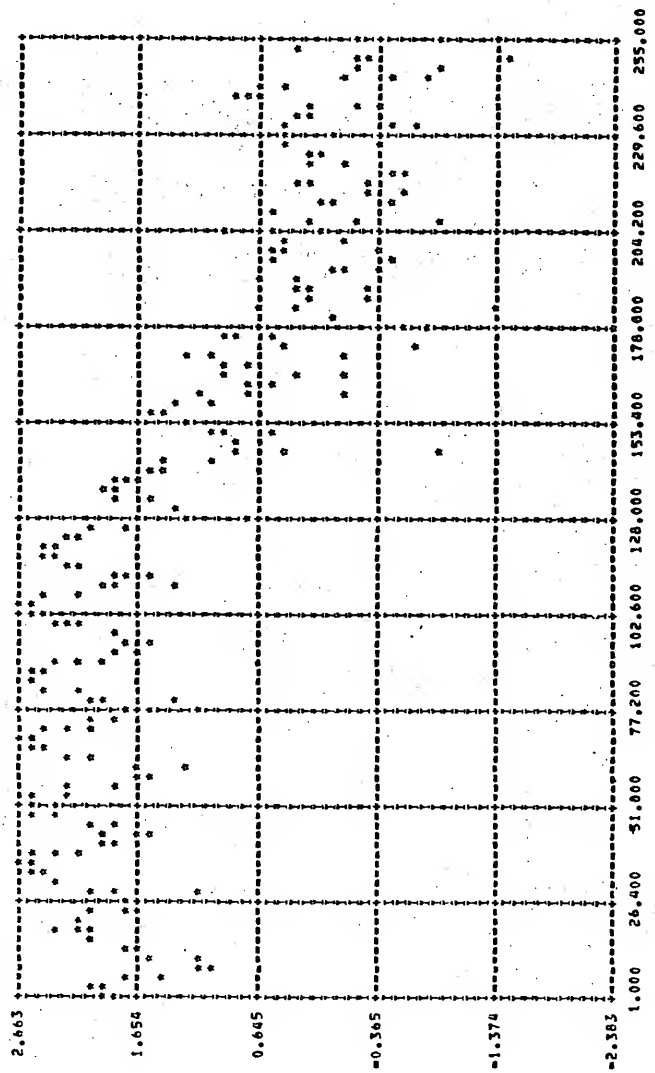


Figure 40 Log Magnitude

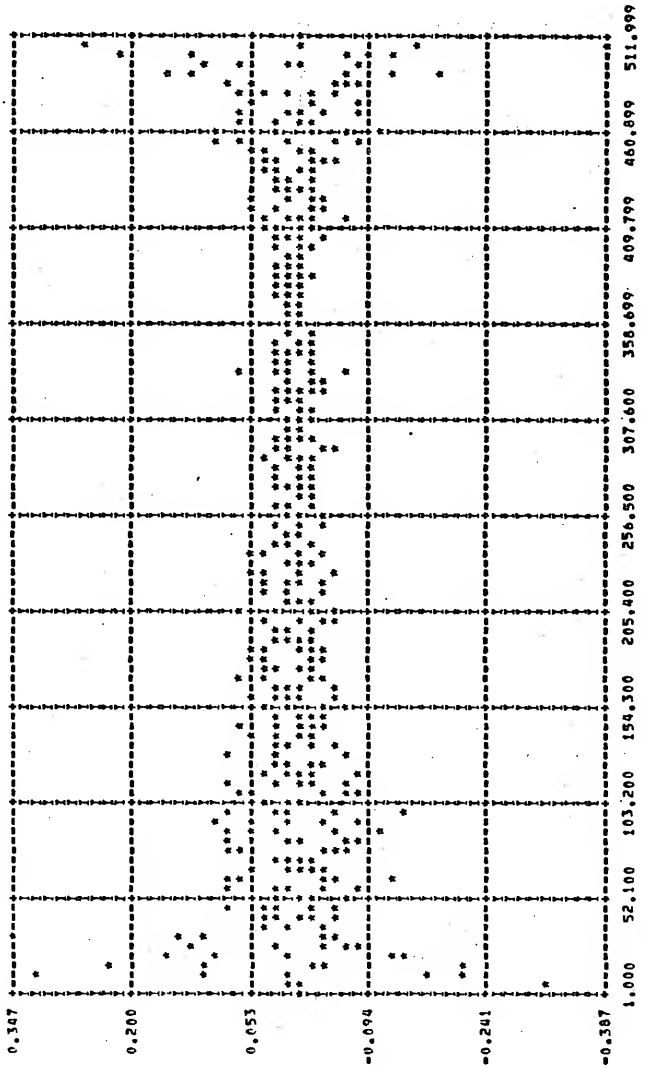


Figure 41 Complex Cepstrum

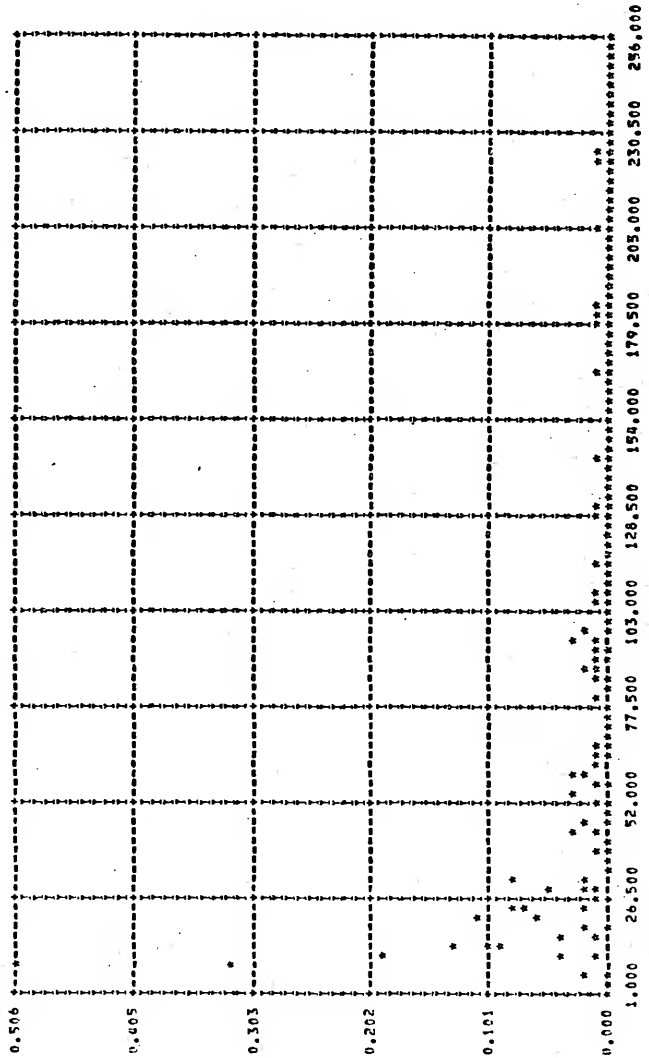


Figure 42 Phase Cepstrum

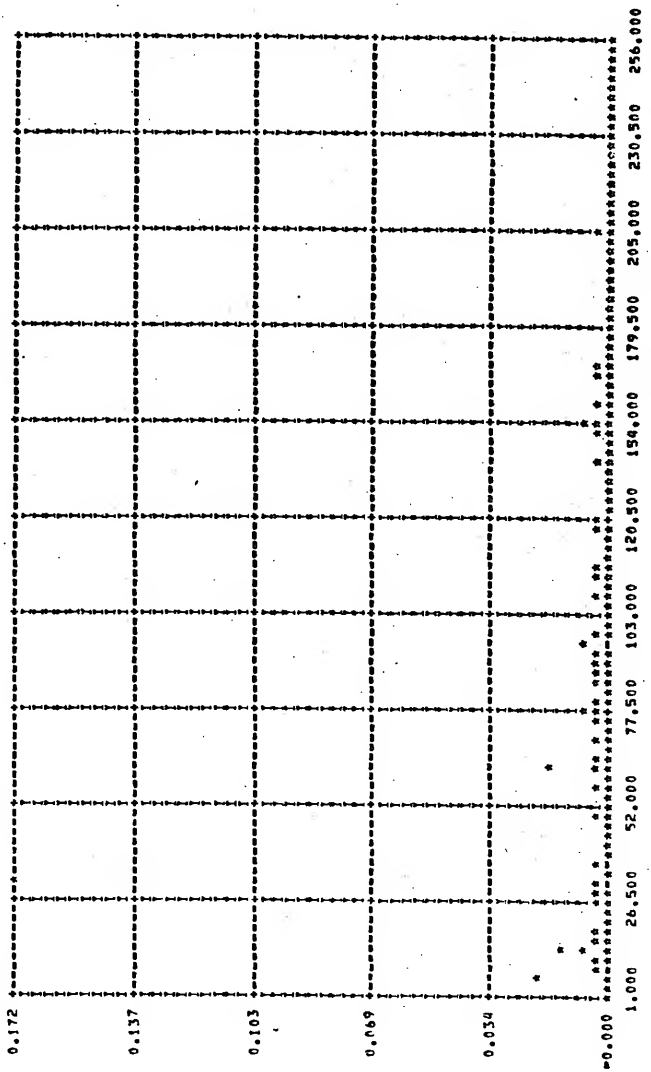


Figure 43 Power Cepstrum

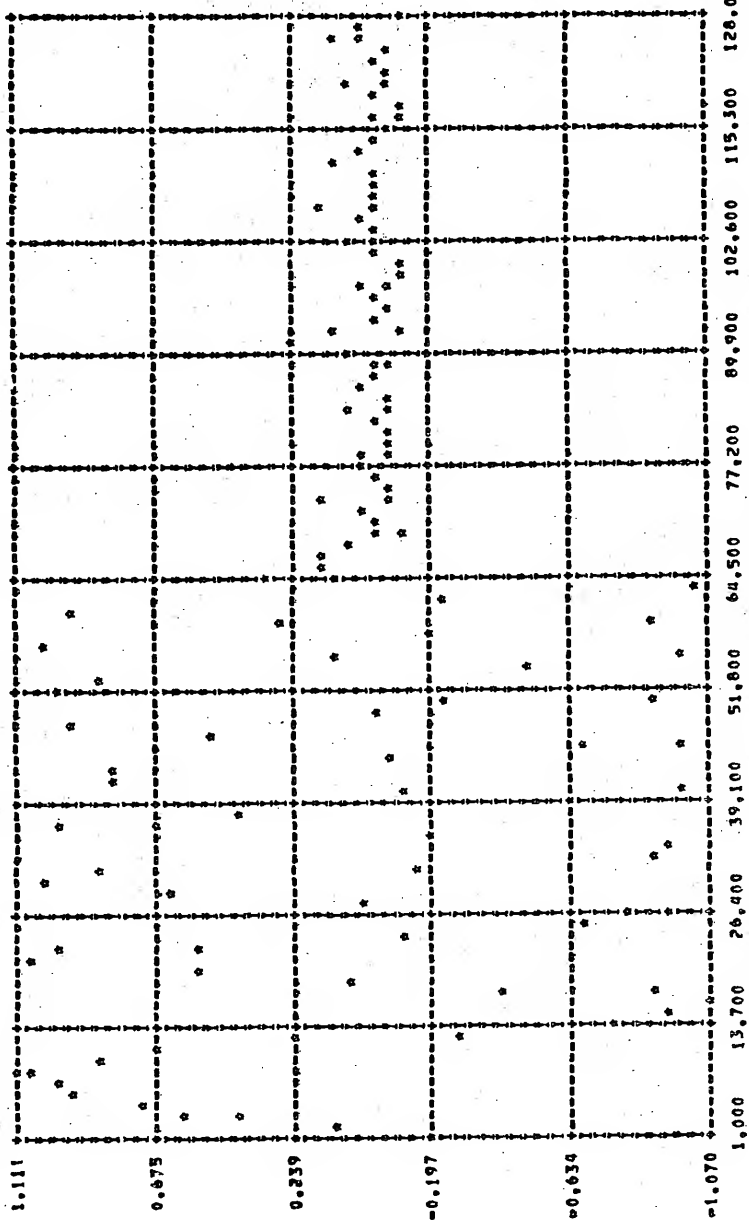


Figure 44 Recovered Wavelet.
MSE=6.31 x 10^{-3}

substantiates the observation that the phase unwrapping algorithm produces erroneous results when the phase jumps rapidly between samples of the log spectrum (as in the case when noise dominates).

(2) The reduction of aliasing in the cepstra by extending the record with zeroes is quite noticeable. This seems to improve the echo detectability. As can be seen from Figure 45, there is a substantial improvement in the MSE of the recovered wavelet caused by the addition of zeroes at all SNR.

(3) Similar experiments were conducted using the exponential signal. The results are similar to those found above with one notable exception. At low SNR the MSE of the recovered wavelet sometimes increases when the original record is extended. This is apparently connected to the fact that the phase curves produced are quite different (outside the signal band). This superiority of the original record seems to arise because the phase errors introduced in phase unwrapping the original record effectively limit the excursions of the phase; since this occurs over a portion of the log spectrum dominated by noise there is a corresponding improvement in wavelet recovery. When the record is extended by zeroes, larger phase jumps may be tracked, and thus more noise may enter the cepstra. In these cases, the appearance of the phase and complex cepstrum are degraded by extending the record. The power cepstrum, however, did show the expected improvement due to the reduction of aliasing, since it is independent of the phase information. These results are atypical and only were found in experiments where the SNR was low and the signal was considerably oversampled (allowing noise to dominate large portions of the log spectrum).

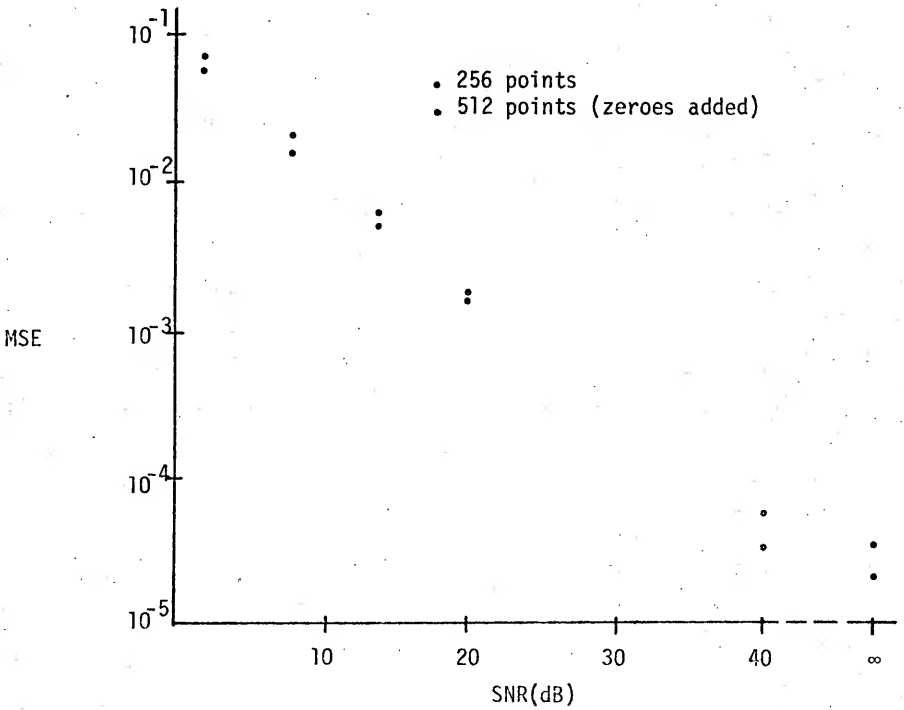


Figure 45 MSE of the Recovered Wavelet when Data Record Is Extended with Zeroes.

Conclusions

The addition of zeroes to a data record increases the sampling rate of the log spectrum and thus substantially reduces the aliasing of the complex cepstrum. In most cases of interest this results in a modest improvement in the MSE of the recovered wavelet and echo detectability. Changing the sampling rate of the log spectrum may (especially at low SNR's) change the unwrapped phase curve. Adding zeroes will ordinarily decrease the number of errors made by the phase unwrapping algorithm, and may be an absolute must for adequate phase unwrapping when the signal phase contains a strong linear trend. Doubling the effective record length by the addition of zeroes will also eliminate the ambiguities found when the echo delay is large.

It is of interest to compare the conclusions reached here with those of reference [3]. Reference [3] concludes that the addition of zeroes adds a linear phase component to the log phase and that this tends to override the noise distortions. As can be seen from the derivation presented above the addition of zeroes will not introduce such a linear phase component (unless of course the zeroes are added in front of the data record). The linear phase component observed in reference [3] is probably due to the difference in phase unwrapping caused by the addition of zeroes, and the reduction in MSE results from the reduction of aliasing.

In a real-time wavelet recovery system it may be impossible to avoid the strong linear phase terms producing the phase unwrapping errors described in Chapter III. To avoid these errors it is recommended that the original record be doubled by the addition of zeroes. This together with the other positive benefits including

the improvement in wavelet recovery, and elimination of echo epoch ambiguities should in many cases outweigh the necessary reduction by a factor of two in the maximum data rate.

CHAPTER V

ALGORITHMS FOR A REAL-TIME WAVELET RECOVERY SYSTEM

Wavelet recovery in real-time proves to be a formidable problem due to the numerous forward and inverse transformations, and nonlinear functions necessary in the complete system. In this chapter the major algorithms necessary for the real-time computation of the complex cepstrum and its inverse are examined. Algorithms are selected on the basic of simplicity, economy, and speed. This is the first look at a real-time wavelet recovery system (based on cepstrum techniques). No attempt is made to actually design a system, rather the intent is to outline methods suitable for implementation in real-time and to roughly estimate the performance which can be expected from the overall system. The computation of the forward and inverse transforms are considered first, and then the computation of the nonlinear functions is examined. As will be seen in the subsequent discussions, it is the computation of the nonlinear functions which offers the greatest barrier to high data rates.

A cursory look at Figure 1 indicates that wavelet recovery is naturally divisible into a number of distinct stages. The first three stages are devoted to the computation of the complex cepstrum, and the last three to the inverse complex cepstrum. The subsequent discussion treats each of these stages separately,assuming only that they all must accept data at the same rate so that there is no

"bottleneck" in the overall system. While most of the discussion is directed toward computation of the complex cepstrum (since the power and phase cepstrum may be derived from it), and the inverse complex cepstrum, many of the considerations are the same for a system dedicated to computation of the power cepstrum only and a short section is devoted to this consideration.

THE DFT ALGORITHMS

Since an all-digital system will be considered, the z-transform methods of Chapter II must be modified. In particular since only finite discrete sequences may be used in an actual computational realization, we are led naturally to use the sampled z-transform rather than its continuous counterpart. It is easily shown that the z-transform sampled at a finite number of equispaced points around the unit circle is just the discrete Fourier transform (DFT). Thus in the following paragraphs the discussion will be of algorithms for performing the DFT and its inverse.

The Cooley-Tukey (decimation in time) and Sande-Tukey (decimation in frequency) Algorithms

These are the "standard" algorithms used in performing the fast Fourier transform (FFT), which is merely a computationally efficient method for performing the discrete Fourier transform (or its inverse). The discrete Fourier transform and its inverse are defined as follows:

$$X(n) = \sum_{k=0}^{N-1} x(k)\, W_N^{-kn} \qquad n = 0,1,2\ldots N-1 \qquad (5\text{-}1)$$

$$x(k) = 1/N \sum_{n=0}^{N-1} X(n)\, W_N^{kn} \qquad n = 0,1\ldots N-1 \qquad (5\text{-}2)$$

where $W_N = \exp(j2\pi/N)$, $\{x(k)\}$ is the input data sequence, and $\{X(n)\}$ are the discrete Fourier transform coefficients.

The Cooley-Tukey and Sande-Tukey algorithms are diagrammed in Figures 46 and 47 respectively. It can be seen that both algorithms possess a remarkable order and symmetry. Each algorithm consists of a single basic operation repeated N/2 log N times. It should be noted that both algorithms output the coefficients in a bit reversed sequence. This implies a reordering (a second bit reversal) before phase unwrapping may take place. Bit reversal is an especially simple form of reordering since it may be accomplished in hardware utilizing only shift registers [12].

For real data the discrete Fourier coefficients are somewhat redundant since from equation (5-1) we see that $X(n) = X^*(N-n)$. Thus if we know $X(n)$, $n = 0,1,\ldots N/2$ then we may compute the remaining coefficients from the above relationship. Algorithms which take advantage of this redundancy are known. Since we are interested only in real data these should prove useful.

Modifying the Basic FFT Algorithms

One way to reduce the computations required by the FFT algorithms outlined above is to take advantage of the fact that $X(n) = X^*(N-n)$ by forming an artificial N/2 point complex array from an N point real array by treating the even and odd samples of the data as the real and imaginary parts of the new complex array. This is accomplished as follows:

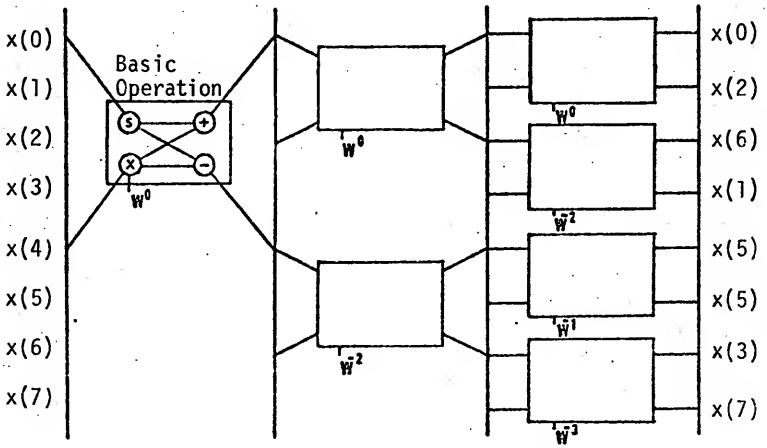


Figure 46 Cooley-Tukey FFT Algorithm.
S - temporary storage

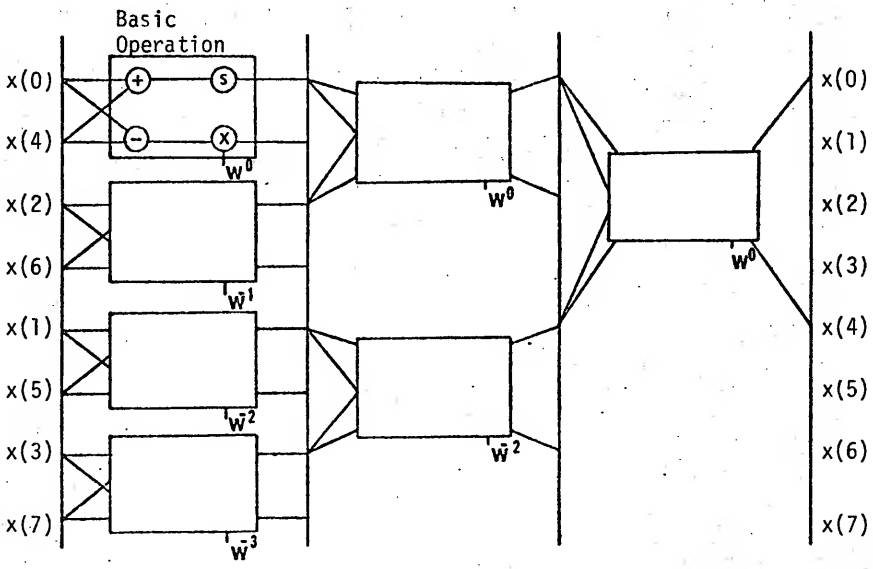


Figure 47 The Sande-Tukey FFT Algorithm (with bit input reversed).

Let $x(k) \leftrightarrow X(n)$

$x(2k) \leftrightarrow X_e(n)$

$x(2k+1) \leftrightarrow X_o(n)$

$c(k) \leftrightarrow C(n)$

where the double-headed arrow indicates the two sequences form a discrete Fourier transform pair.

Let $c(k) = x(2k) + jx(2k+1)$ where $k = 0,1 \ldots N/2-1$. (5-3)

Then it can be shown that

$$X_e(n) = Re[(C(n) + C(N/2-n))/2] + j[Im((C(n) - C(N/2-n))/2)] \quad (5\text{-}4)$$

and

$$X_o(n) = Im[(C(n) + C(N/2-n))/2] - j[Re((C(n) - C(N/2-n))/2)] \quad (5\text{-}5)$$

where $n = 0,1,2 \ldots N/2$ and

$$X(n) = X_e(n) + W_N^n X_o(n) \qquad n = 0,1 \ldots N/2 \qquad (5\text{-}6)$$

(or $\ldots N-1$ if desired).

Noting that an N/2 point transform requires only N/4 log N/2 basic operations and that the computations required by (5-4), (5-5) and (5-6) are approximately equivalent to N/4 basic operations, we see that the total number of operations

$$\cong N/4 \log_2 N/2 + N/4$$

$$\cong N/4 \log_2 (N/2) (2)$$

$$\cong N/4 \log_2 N$$

$\cong$ 1/2 number of operations for the unmodified algorithm.

Thus this algorithm achieves a 2:1 reduction in computation and required storage. Unfortunately, it does not retain the order and

symmetry of the standard Cooley-Tukey and Sande-Tukey algorithms (at least in the last stage) and thus to some extent complicates the hardware implementation.

The Bergland Algorithm

Bergland [13] has proposed an FFT algorithm for real valued series in which he specializes the Cooley-Tukey algorithm by eliminating redundant operations at each stage of the computation. This algorithm results in a 2:1 reduction in computation and storage while preserving the order and symmetry of the original Cooley-Tukey algorithm. Figure 48 diagrams the Bergland algorithm. It should be noted that the basic operation present in this algorithm is identical to that of the Cooley-Tukey algorithm except that one of the results is complex conjugated before being stored back in memory. One slight drawback of this algorithm is that the coefficients are produced in an unusual order requiring a more complicated reordering scheme than that used for bit reversal. Bergland has also proposed an algorithm for inverting the transform of real valued data. This is shown in Figure 49 and achieves a 2:1 reduction in storage and a better than 2:1 savings in computations over the ordinary inverse algorithm.

The Hartwell Modification

Hartwell [14] has proposed an algorithm for inverting the transform of real valued data utilizing the unaltered algorithm for the forward transformation (either the Bergland or the modified Cooley-Tukey). Let $X(n)$, $n = 0,1,2...N-1$ be the DFT coefficients for the real valued data $x(k)$, $k = 0,1...N-1$.

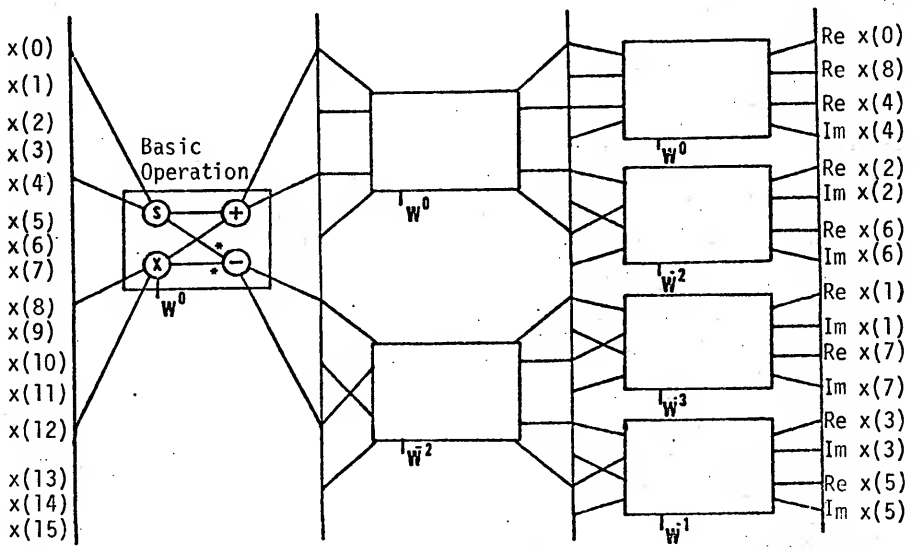


Figure 48 Bergland Real Valued Input Algorithm

Note: The circled multiplication performs the operation $(x_8+jx_{12})*W^0$; that is, the two real inputs are used as the real and imaginary parts of one of the complex multiplicands. Similarly, the sum and difference operations yield two outputs, the real and imaginary parts of their results.

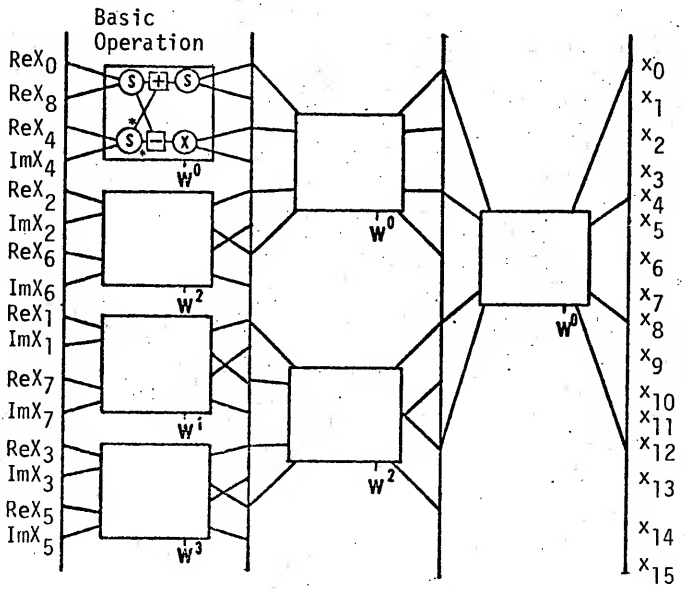


Figure 49 Bergland Inverse Algorithm

The inverse transformation defined in equation (5-2) is

$$x(k) = \frac{1}{N} \sum_{n=0}^{N-1} X(n)\, W_N^{nk} \qquad \text{where } k = 0,1 \ldots N-1.$$

Construct a new sequence of real numbers from the Fourier coefficients

$$B(n) = \mathrm{Re}(X(n)) + \mathrm{Im}(X(n)) \qquad (5\text{-}7)$$

$$k = 0,1 \ldots N/2$$

$$B(N-n) = \mathrm{Re}(X(n)) - \mathrm{Im}(X(n)) \,. \qquad (5\text{-}8)$$

Thus we may rewrite $\{x(n)\}$ in terms of $\{B(n)\}$

$$X(n) = [B(n) + B(N-n)]/2 + j[B(n) - B(N-n)]/2 \qquad (5\text{-}9)$$

inserting this into equation (5-2) and noting

$$W_N^{nk} = \cos(2\pi nk/N) + j \sin(2\pi nk/N)$$

we obtain

$$x(k) = \frac{1}{N} \sum_{n=0}^{N-1} B(n) \cos(2\pi nk/N) - B(n) \sin(2\pi nk/N) \qquad (5\text{-}10)$$

but these two sums may be computed using the forward transformation for real valued data. The sums being the real and imaginary parts of the forward transformation (except for a factor of N).

Thus let $$x'(k) = \sum_{n=0}^{N-1} B(n)\, W_N^{-nk} \qquad (5\text{-}11)$$

then $$x(k) = \frac{1}{N}[\mathrm{Re}(x'(k)) + \mathrm{Im}(x'(k))] \qquad k = 0,1 \ldots N/2 \qquad (5\text{-}12)$$

$$x(N-k) = \frac{1}{N}[\mathrm{Re}(x'(k)) - \mathrm{Im}(x'(k))] \,. \qquad (5\text{-}13)$$

Note that except for a factor of N the modifications performed both before and after the forward algorithm are identical. Thus, the inverse transformation may be obtained by executing a simple procedure both before and after the forward algorithm.

An added advantage to this algorithm results from noting that

$$\frac{1}{N} \text{Re } x'(k) = x(k) + x(N-k) \tag{5-14}$$

$$\frac{1}{N} \text{Im } x'(k) = x(k) - x(N-k) \quad . \tag{5-15}$$

If the complex cepstrum is being computed (as opposed to its inverse being computed), the squares of equations (5-14) and (5-15) are just the power and phase cepstrum respectively. Thus the steps leading to the computation of the complex cepstrum via this algorithm yields the power and phase cepstra as natural byproducts.

Which Algorithm for Complex Cepstrum Computation?

Table 1 outlines the basic properties of the various forward and inverse FFT's for computation of the complex cepstrum, and its inverse. The third combination (Bergland and Hartwell-Bergland) is preferred for the following reasons:

1) It achieves a two-to-one saving in storage and a better than 2:1 reduction in computation over Cooley-Tukey and Sande-Tukey algorithms.

2) The regularity of the Bergland algorithm facilitates its implementation and makes it especially adaptable to a cascade implementation which at present seems the most suitable for achieving high throughputs.

Table 1 Algorithms for Cepstrum Computation

CT - Cooley-Tukey
ST - Sande-Tukey

	FFT Algorithm		Number of	Comments
	Forward	Inverse	Basic Operations	
(1)	CT ST	CT ST	$(N/2)\log_2(N)$	Requires more computation and storage than other algorithms.
(2)	Bergland	Inverse Bergland	$(N/4)\log_2(N/2)$	Difference in forward and inverse algorithms would complicate hardware.
(3)	Bergland	Hartwell-Bergland	$(N/4)\log_2(N/2)$	Same basic algorithm used in both forward and inverse transforms with only minor modifications of data required. Inverse algorithm lends itself to computation of phase and power cepstra.
(4)	Modified CT	Hartwell-Modified CT	$(N/4)\log_2(N)$	Algorithm is not as regular as (3). Inverse algorithm lends itself to computation of phase and power cepstra.

3) Since essentially the same algorithm is used for the forward and inverse transform, it permits a time sharing of the FFT processor between input and output if dictated by economic considerations.

4) The inverse algorithm facilitates the computation of the power and phase cepstra directly from the complex cepstrum.

Of course other considerations may dictate the choice of the Cooley-Tukey or Sande-Tukey algorithms; e.g., if the FFT portion of the processor is to double as a spectrum analyzer where the DFT of complex data may be required. Appendix C demonstrates the ease with which the inverse transform of the transform real valued data can be computed utilizing the forward (Cooley-Tukey or Sande-Tukey) algorithms. This should prove useful in any wavelet recovery system utilizing the above-mentioned algorithms.

Having chosen suitable FFT algorithms for computation of the forward and inverse discrete Fourier transforms, we may now proceed to examine the performance that may be expected from the two FFT realizations most prevalent in the literature, namely the sequential and the cascade. While even faster realizations are known at this time, their construction would be prohibitively expensive [15], and as will be seen shortly, unnecessary. In the sequential realization a single arithmetic unit is used to perform the basic operation $N/4 \log_2 N/2$ times (as shown for N = 16 in Figure 48). Since only a single arithmetic unit is used, and the accessing pattern is quite regular a relatively small amount of hardware is involved. The cascade (or pipeline) realization dedicates an arithmetic unit to each of the $\log_2 N/2$ levels resulting in a $\log_2 N/2$ increase in the throughput. Table 2 compares the performance of the two

Table 2 FFT Processor Performance
Where T_{BO}=Time Required to Perform One Basic Operation

	Arithmetic Units	Execution Time	Throughput
Sequential	1	$T_{ES}=(N/4)\log_2(N/2)*T_{BO}$	$N/T_{ES}=4/(\log_2(N/2)T_{BO})$
Cascade	$\log_2(N/2)$	$T_{EC}=2*(N/4)*T_{BO}$	$2N/T_{EC}=4/T_{BO}$

realizations (neglecting any housekeeping operations in the computation of the execution time as these generally require relatively small amounts of time compared to the performance of the basic arithmetic operations).

From Table 2 we observe that the throughput of the cascade processor is greater than the sequential by a factor of $\log_2(N/2)$. Subsequently it will be seen that this gain is of no use due to limitations imposed by other stages in the computation process.

Now that the basic FFT realizations have been analyzed, we are in a position to examine the algorithms suitable for use in the computation of the nonlinear functions. As will be seen in the next section these computations, more than any other factor, limit the sampling rates to which the overall processor will be applicable.

COMPUTATION OF NONLINEAR FUNCTIONS

High speed computation of the nonlinear functions used in the wavelet recovery system is quite difficult. In this section a class of algorithms is presented which should allow wavelet recovery up to sampling rates of around 100 kHz using a single hardware unit for each nonlinear function. This is more than sufficient for the processing applications that cepstrum techniques are currently being used for (i.e., in the speech range). While higher data rates are possible by utilizing multiple hardware units or stored tables, the expense of such techniques would be excessive.

Computation of Nonlinear Functions

There are five nonlinear functions necessary in the entire wavelet recovery algorithm. These are log, arctangent, exponential,

sine and cosine. Specker [16] has described a class of algorithms for the evaluation of all of the above functions. The principle behind this class of algorithms is that tables of stored constants are used to transform the argument into a range where the function may be approximated by a simple polynomial. These algorithms have a number of interesting features:

1) All real multiplications reduce to one addition and one shift operation.

2) Only a relatively small number of stored constants are needed.

3) All of the above-mentioned functions may be computed with a hardware configuration having 3 adders, 2 shift networks, and a table of stored constants in approximately the time to perform R additions and R shift operations [16].

4) The maximum absolute error is less than 2^{-R}, where R is the number of iterations in the computation algorithm.

The computation of log(x) and exp(x) are outlined in the flow and block diagrams shown in Figures 50 and 51. Computation of arctangent (x) is similar to the logarithm computation, but involves complex operations and requires one additional adder and shift register. Simultaneous computation of sines and cosines is possible, and resembles computation of the exponential, but again complex operations are involved and an additional adder and shift register are required in the hardware configuration. Each of these nonlinear functions must be computed for N/2 ($\pm$1) points (recalling we are taking advantage of the redundancy in the transform of real data). Thus the execution

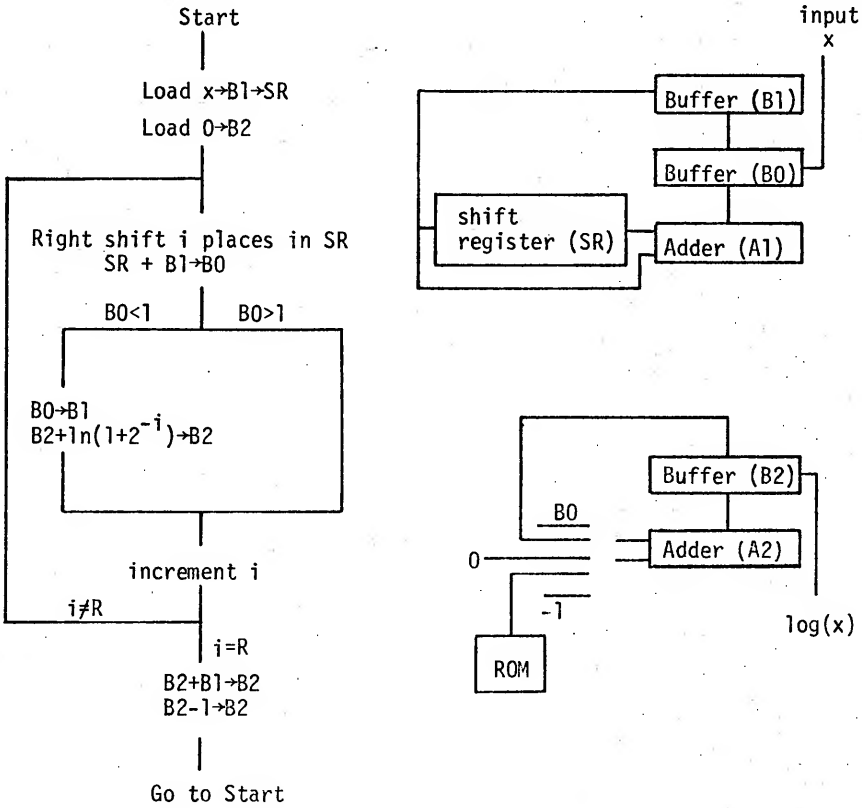


Figure 50 Computation of log(x).
Where ½ ≤x<1; maximum absolute error is 2^{-2R-1}.

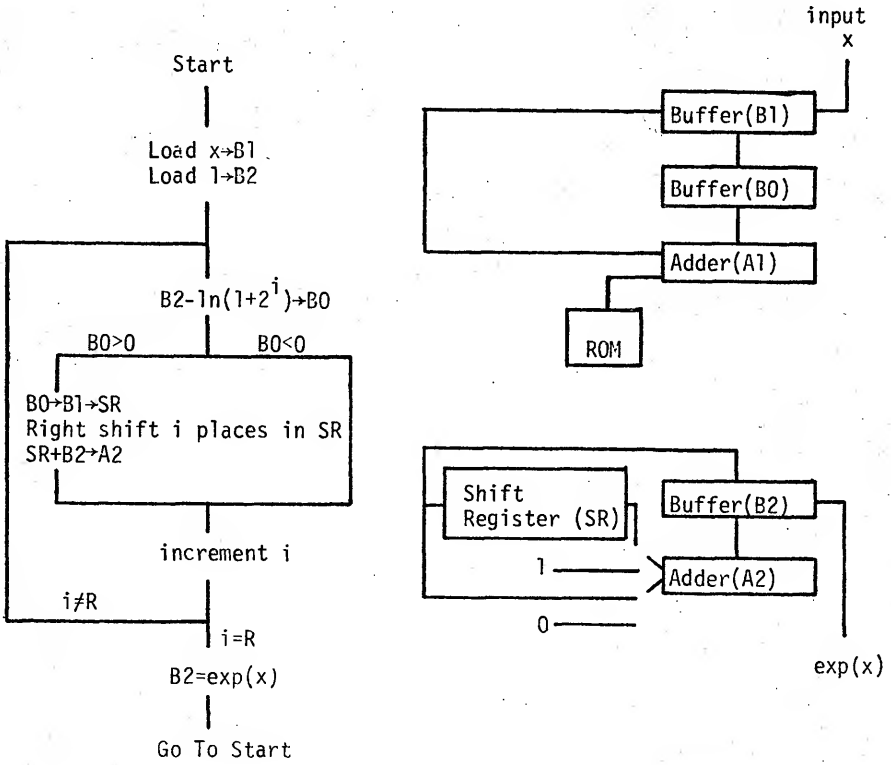


Figure 51 Computation of exp(x).
Where $1/2 \leq x < 1$; maximum absolute error is $2^{-R}\exp(1+2^{-R})$.

time for the computation of these functions (for the full $\frac{N}{2}$ points) is approximately

$$T'_{EN} = \frac{N}{2} R T_I$$

where R is the number of iterations, and T_I is the time per iteration. This estimate must be adjusted somewhat to obtain the execution time for the intermediate stages in the overall wavelet recovery system. The actual execution time for these stages must include time for the various computations associated with the particular stage under consideration such as the resolution of large arguments, correct assignment of the quadrant (in the case of the arctangent), computing the magnitudes of complex quantities, etc., and in the case of stage 2, the phase unwrapping and linear phase removal. It should be noted that the computations listed above (except for linear phase removal) can be cascaded with the computation of the nonlinear functions. Thus only a relatively small correction must be made in the execution time given for the nonlinear functions to estimate the total execution time of the intermediate stages.

An examination of the nature of the operations performed by the various stages of the wavelet recovery system reveals that stage 2 (principally because it involves not only the computation of nonlinear functions but also phase unwrapping and linear phase removal) will be the stage limiting the throughput of the overall system. The estimate given below should provide sufficient time for the computations of this stage and is thus somewhat conservative for the other intermediate stages.

Let the execution time for the intermediate stages be written

$$T_{EN} = T'_{EN} + T_A$$

where T_A is included to account for the time required by the operations (other than computing the nonlinear functions) associated with the stage under consideration. For reasonable values of R (say R=16), T_A should be somewhat less than T'_{EN}. For the purpose of making a rough estimate of the system performance we shall assume $T_A = \frac{1}{2} T'_{EN}$; thus the total execution time (which is assumed the same for all of the intermediate stages) is

$$T_{EN} = 3/2\ T'_{EN} = 3/4\ NRT_I \quad .$$

This may then be used to estimate the throughput of the intermediate stages, thus the expression

$$S_I = \frac{N}{T_{EN}} = \frac{4}{3RT_I}$$

provides a rough estimate of the throughput obtainable from the intermediate stages (involving the computation of nonlinear functions). This is the limiting throughput of the system.

PHASE UNWRAPPING AND LINEAR PHASE REMOVAL

As seen in Chapter II, the imaginary part of the log spectrum (the signal phase) must be "unwrapped" in order to make it continuous. A close look at the algorithm presented in Chapter II reveals that the phase unwrapping may begin as soon as the phase modulo 2π has been computed (via an arctangent routine) for the first two points. Thus we may phase unwrap each data point while the phase modulo 2π is being

computed for the next point. Inspecting the algorithm presented in Chapter II we see that the phase unwrapping should be easily implemented in hardware, and its computation time should be substantially less than that of the arctangent routine. Thus phase unwrapping should add little additional time to the computations of stage 2 of the complex cepstrum realization if it is cascaded after the computation of the arctangents.

A second problem with the phase (which was examined in Chapter III) is concerned with the presence of a linear phase term as might originate when the composite signal does not start at the beginning of the data record under consideration. Such a trend must be removed for satisfactory wavelet recovery. Removal of the trend requires that it be determined from the phase of the N/2 point and then subtracted point by point from points 1 thru N/2. This requires 1 division (to determine the trend), N/2-2 additions (to determine the quantity subtracted from each point) and N/2-1 subtractions (to remove the trend). Comparing the operations involved in phase unwrapping and linear phase removal with those necessary for the computation of the nonlinear functions indicates that these computations should be easily accomplished in the time T_A used in the previous section to estimate the execution time of the intermediate stages. It should also be noted that this same linear phase term must be restored during the inverse operation to avoid shifting the recovered wavelet.

LINEAR FILTERING

The detection and smoothing of echo peaks in the complex cepstrum should prove no problem at high SNR provided the basic wavelet and

echo are separated by 3 to 4% of the total record length. The computation time for the "filtering" stage should be considerably less than the nonlinear stages, thus this stage does not limit the throughput of the overall wavelet recovery system.

For low SNR, or in the case of very small echo delays, the automatic detection of the peaks will be quite difficult. Further research is necessary to determine the limits of automatic detection in this case.

AN OVERALL LOOK AT SYSTEM PERFORMANCE

As is shown in Figure 1, wavelet recovery consists of seven basic stages:

1. the forward FFT
2. the intermediate stage involving computation of the logarithm, phase unwrapping, and linear phase removal
3. the inverse FFT
4. "filtering" the complex cepstrum
5. the forward FFT
6. the intermediate stage involving exponentiation, the computation of sines and cosines, and the insertion of the linear phase component removed in stage 2
7. the inverse FFT

The performance of the wavelet recovery system will be examined subject to the following assumptions:

1) The computation of each basic stage is completed before the next stage is entered.

2) The execution time for the forward and inverse transforms is the same. This is not strictly true since the inverse FFT involves

a reordering and the operations specified by equations (5-7), (5-8), (5-12) and (5-13), but is close enough for a rough estimate. The execution time will be estimated from Table 2 .

3) Since the throughput of the overall system is limited by stage 2, performing the computations of stages 4 and 6 in less than the execution time for stage 2 will result in no gain in performance; thus the execution time for these stages will be assumed the same as for stage 2.

4) A single FFT processor is time shared between stages 1, 3, 5 and 7 of the wavelet recovery system. As will be seen shortly, the throughput is limited to such an extent by stage 2 that this is feasible. This would probably be dictated by economic considerations in any event.

Subject to the above assumptions, we find the throughput of the overall system is just the throughput of stage 2, and the execution time will be 3 times the execution time for stage 2 plus the time required for the four FFT's (about 4 times the execution time of stage 2 in all).

Table 3 summarizes the throughputs of the sequential FFT, cascade FFT, and the intermediate stages.

Since the FFT processor is time shared between four stages, its throughput must be four times that of the intermediate stages. If we assume R=16 (16 iterations for the computation of the nonlinear functions) and note that T_{BO} and T_I are on the same order of magnitude, we see that the cascade realization has a throughput much higher than necessary. For this reason and reasons of economy the cascade realization will not be considered further.

Table 3 Throughput Comparison of Stages in the Wavelet Recovery System.

	Throughput
Sequential FFT	$4/(\log_2(N/2)T_{BO})$
Cascade FFT	$4/T_{BO}$
Intermediate (non-FFT) stage estimate	$4/(3RT_I)$

An Estimate of System Performance

Let us assume a reasonable value for the iteration time T_I = .75 μsec, and the number of iterations R=16. From Table 3 we may estimate the throughput of stage 2 (and thus the throughput of the entire wavelet recovery system) as

$$S_I = \frac{4}{3RT_I} = 1.1 \times 10^5 \text{ Hz}$$

From Table 3 we note that the throughput of the sequential FFT goes down with increasing N. If we assume a typical value of T_{BO} as .50 μsec (processors are being built with T_{BO} less than this value) we see that $N=2^{19}$ points is the maximum number of points which may be transformed in real time, since for N greater than this value the throughput of the sequential FFT will not be 4 times the throughput of the other stages as is necessary for time sharing (this is not particularly restrictive as N is commonly chosen less than this value).

REAL-TIME COMPUTATION OF THE POWER CEPSTRUM

A system designed to compute the power cepstrum in real-time could be implemented along the lines suggested in Figure 1. Such a system is easier to construct than the wavelet recovery system since no phase unwrapping or linear phase removal is required, and only the real logarithm is computed. Utilizing the algorithms presented in this chapter, sampling rates on the order of 2×10^5 samples/sec are possible. The sampling rate is about twice that of the wavelet recovery system because computation of the real logarithm (for a given accuracy) requires only half as many iterations as computation of the arctangent,

and only two FFT's are required. It should be noted that it does not matter whether the second FFT of Figure 1 is forward or inverse since only the magnitude squared of its output is used.

SUMMARY

Figure 52 outlines the entire wavelet recovery system. Conservatively it is estimated that such a system utilizing the Bergland algorithm for the forward FFT's, the Hartwell-Bergland algorithm for the inverse FFT's, and the class of algorithms presented by Specker for computation of the nonlinear functions (logarithms, arctangents, exponentials, sines and cosines) can be implemented in hardware to provide real-time echo detection and wavelet recovery at sampling rates of around 10^5 samples/sec. Only a single sequential FFT processor is necessary, since it can be time shared between the four FFT stages.

Since the power cepstrum is insensitive to the phase unwrapping problems which can severely degrade echo epoch detectability in the complex cepstrum, it is reasonable to include its computation in the wavelet recovery system. As has been shown in this chapter, this is easily accomplished as a byproduct of the first inverse FFT.

If the record length is doubled by the addition of zeroes to avoid the problems discussed in Chapter III, the sampling rate cited above is reduced by a factor of two. A rectangular windowing of the complex cepstrum could be included in the peak detection and removal stage to reduce the MSE of the recovered wavelet. This seems superior to Hanning the log spectrum, since its implementation is simpler, and it will have no effect on detecting long time delay echoes.

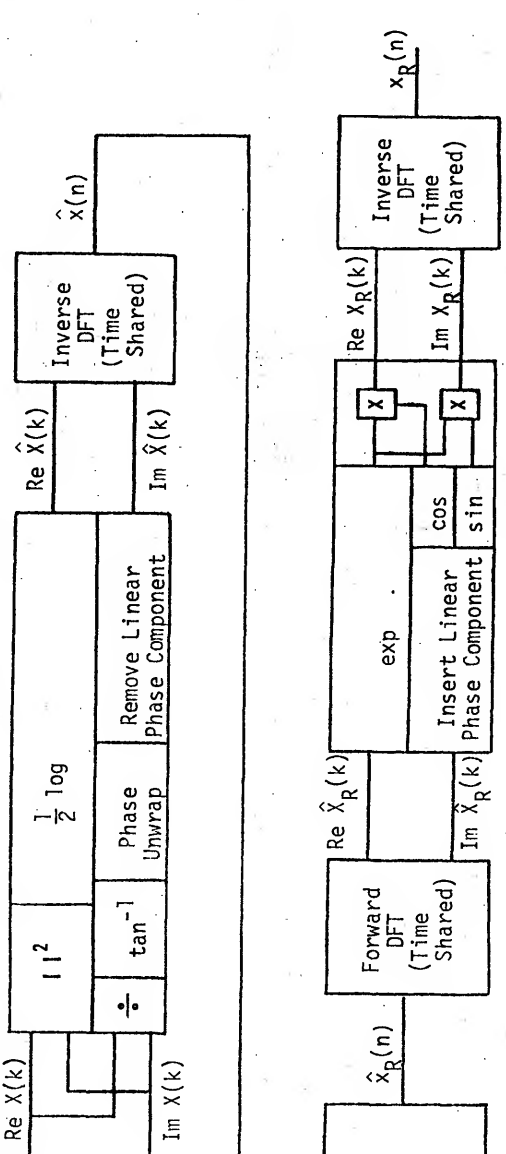


Figure 52 Overall Wavelet Recovery Algorithm.

CHAPTER VI

SUMMARY AND CONCLUSIONS

THE CEPSTRA

A new cepstrum technique, the phase cepstrum, has been introduced. The phase cepstrum, like the power cepstrum, has been shown to be useful for the detection of echo epoch times. In the presence of noise the phase cepstrum exhibits a somewhat (6 dB) higher epoch detection threshold than the power or complex cepstrum. The power and phase cepstra have been shown to be, respectively (except for a constant) the square of the even and odd portions of the complex cepstrum. This facilitates the computation of the power and phase cepstra, and gives added insight into the power cepstrum technique.

COMPUTATIONAL PROBLEMS

Four problems occurring in cepstral analysis were examined. The presence of a linear phase term distorts the appearance of the complex (and phase cepstrum) even when partially removed as in reference [3]. Such a term degrades both echo epoch detectability in the complex and phase cepstra, and wavelet recovery. The complete removal of linear phase contributions to the cepstra is a necessity for satisfactory wavelet recovery.

Phase unwrapping errors are observed to occur if the log phase changes by more than π between samples. This can be quite detrimental

to echo epoch detectability in the complex and phase cepstra, and to wavelet recovery. These errors can be reduced by adding zeroes to the data record. In fact doubling the record by adding zeroes completely eliminates phase unwrapping errors due to the presence of a strong linear phase term as may occur when the composite signal begins late in the record.

Aliasing of the complex cepstrum is another problem of note. One problem caused by aliasing is the ambiguity in echo epoch determination. This can be eliminated by doubling the data record with zeroes.

Finally oversampling the composite signal is observed to degrade wavelet recovery when additive noise is present.

WINDOWING

Windowing the input data record with the common windows used to reduce leakage is quite detrimental to wavelet recovery. However, wavelet recovery is possible in two cases:

(1) when the window w(nT) is approximately constant over the duration of the basic wavelet, or

(2) when the window w(nT) is approximately constant over the duration of the echo delay. The nature of the error introduced in these two cases into the recovered wavelet has been examined theoretically and verified experimentally. It is interesting to note that the determination of the echo epoch is possible even when wavelet recovery is not. Thus cepstrum techniques can be used to detect similar but not necessarily identical waveforms. Windowing does however increase the SNR at which echo epoch detection is possible.

The exponential window produces considerably different results than the windows used above. From theoretical considerations, the exponential window was expected to perform as well as the rectangular window at high SNR (and somewhat better if echo truncation is a problem). This is verified by the experimental results (provided the composite signal begins early in the record) when the MSE is computed over the signal duration; however, correction for the exponential window introduces some distortion into the recovered record outside the signal duration. At low SNR the exponential window performs better than the rectangular window. This is due to the fact that the composite signal examined starts near the beginning of the data record and thus the window weights the noise less than the signal. When the composite signal occurs near the end of the record the recovered wavelet is degraded by the exponential window both at low and high SNR. It is similarly noted that (for $a<1$) even though the echo impulse train is diminished by the exponential window the detection threshold is unchanged from the rectangularly windowed case when the composite signal occurs near the beginning of the record. The detection threshold increases when the composite signal occurs near the end of the record.

Windowing the log spectrum with the Hamming window has little effect on the recovered wavelet at high SNR, but it greatly degrades wavelet recovery at low SNR. The echo epoch detection threshold is also increased (12 dB over rectangularly windowed case) by windowing the log spectrum.

A Hanning smoothing of the log spectrum (both phase and magnitude) is equivalent to windowing the complex cepstrum with the Hanning window.

Thus, in general, it will not (as reported in reference [3]) decrease the echo epoch detection threshold. Hanning smoothing does, however, reduce the MSE of the recovered wavelet when additive noise is present. The MSE can also be reduced by rectangularly windowing the complex cepstrum with a window of length less than the record length. A Hanning smoothing of the log magnitude only (suggested in [3]) is equivalent to windowing only the even portion of the complex cepstrum. It is shown that this introduces echo epoch peaks at both positive and negative quefrencies. If both sets of peaks are smoothed, this also reduces the MSE of the recovered wavelet, but the MSE is not as low as obtained when both the log magnitude and phase are smoothed.

EXTENDING THE DATA RECORD WITH ZEROES

Adding zeroes to the data record is formally shown to increase the sampling rate of the log spectrum and to reduce the aliasing of the complex cepstrum. Doubling the data record with zeroes eliminates the echo epoch ambiguities and phase unwrapping errors associated with strong linear phase terms. Experimentally, the MSE of the recovered linear FM signal is significantly reduced by the addition of zeroes, and the echo epoch detectability is improved. It is also noted that at low SNR the unwrapped phase curve is changed in appearance by the addition of zeroes. This reflects the increased sampling rate of the log phase.

ALGORITHMS FOR REAL-TIME WAVELET RECOVERY

The primary difficulty with computing the complex cepstrum and performing wavelet recovery at high data rates is the computation of the nonlinear (logarithm, arctangent, etc.) functions. The class of

algorithms proposed by Specker [16] appears to offer a fast, efficient means of computing the functions required in wavelet recovery. The data rate is limited to such an extent by the computation of these functions that it appears that a single time shared FFT processor can be utilized to perform all four DFT's needed in the overall wavelet recovery system. The Bergland and Hartwell-Bergland algorithms would seem to be especially well adapted to the computation of the forward and inverse transforms, respectively. Utilizing these algorithms in the system outlined in Figure 52, data rates of around 100 KHz are possible. If it is necessary to double the record length by adding zeroes to avoid the phase unwrapping and aliasing problems discussed, then the data rate will be reduced by a factor of two.

SUGGESTIONS FOR FUTURE RESEARCH

Several problems inherent in the computation of the cepstra, and wavelet recovery remain to be studied. One problem concerns choosing the proper data record length. Too long a record will result in a low SNR (when noise is present), while a record length having a duration close to the duration of the basic wavelet may produce truncation problems. The techniques used herein to predict the distortion introduced by windowing the input data record should be useful in determining the nature of the errors introduced by truncation.

A second problem occurs when the echo is delayed by a noninteger number of sample times. In this case the echo delay ripples of the log spectrum are not at the discrete quefrencies of the sampled cepstra, thus the peaks associated with the echo will be subject to the picket fence effect. This will undoubtedly affect the echo detection threshold.

Finally, the actual hardware implementation of the wavelet recovery process should now be considered.

APPENDIX A

A COMPARISON OF THE ECHO DETECTION CAPABILITY OF THE PHASE, POWER AND COMPLEX CEPSTRA

Since the phase, like the complex and power cepstra, lends itself naturally to the estimation of echo epoch times and amplitudes, we seek in this section to compare the three techniques.

In the discussion of Chapter II, it was shown that for the single additive echo case the complex cepstrum will have a peak of height (a) at the echo epoch quefrency (for a<1 and provided the complex cepstrum of the basic wavelet is zero at the echo epoch quefrency). If the complex cepstrum of the basic wavelet is non-zero at the echo epoch, the peak height will be altered as can be seen from equation (2-26). From equations (2-33) and (2-36), it is now obvious that the peak height at the echo epoch will be (a^2) for both the phase and power cepstra (there will also be a contribution from the odd (for the phase cepstrum) or even (for the power cepstrum) portions of the complex cepstrum if they are non-zero at the echo epoch). The extension of these remarks to the case a>1 is obvious.

A series of computer experiments was conducted to compare the phase, power and complex cepstra for the estimation of the echo amplitude (a). The signal examined was approximately quefrency limited to less than the echo epoch time (n_0=30). As expected, the accuracy of the echo amplitude estimation decreases with decreasing SNR. For the complex cepstrum the amplitude estimate was quite good

down to a SNR of 8 dB (5% error at 8 dB for a=.5). The power and phase cepstra estimates were quite good at high SNR but not nearly as reliable at low SNR though the estimates should be useful down to a SNR of 14 dB.

Comparing the detectability threshold for the cepstra, it was found that the echo could be detected in the complex and power cepstrum down to a SNR of 8 dB. The phase cepstrum detectability was somewhat higher at a SNR of 14 dB. It is interesting to note that the log phase curve develops discontinuities at this SNR. A clear example of this is given in Figure 54. It is also noted in Chapter V that the log phase curve differs before and after the addition of zeroes at this SNR. The effects of additive noise apparently begin to affect the phase unwrapping at this SNR. This is undoubtedly connected with the fact that the detectability threshold of the phase cepstrum is higher than that of the power cepstrum. Since the power cepstrum is independent of phase, it is not affected by the phase unwrapping errors discussed in Chapter III. When the number of these errors is considerable the detection threshold of the phase and complex cepstra should increase, thus the power cepstrum may often prove superior in echo epoch detection.

Example

Discontinuities in the Unwrapped Phase Curve

Figures 53 through 59

This group of figures illustrates the computation of the cepstra, and wavelet recovery in a case where additive noise causes discontinuities in the unwrapped phase curve.

The composite signal (256 points) is

$$y(nT) = x(nT) + .5x(nT-30T)$$

where $x(nT) = nT\, e^{-nT}$ $\quad 0 \leq n < 64$

Figure Number	Figure Title
53	Composite Signal SNR = 14 dB
54	Unwrapped Phase Curve
55	Log Magnitude
56	Complex Cepstrum
57	Phase Cepstrum
58	Power Cepstrum
59	Recovered Wavelet MSE = 2.63×10^{-4}

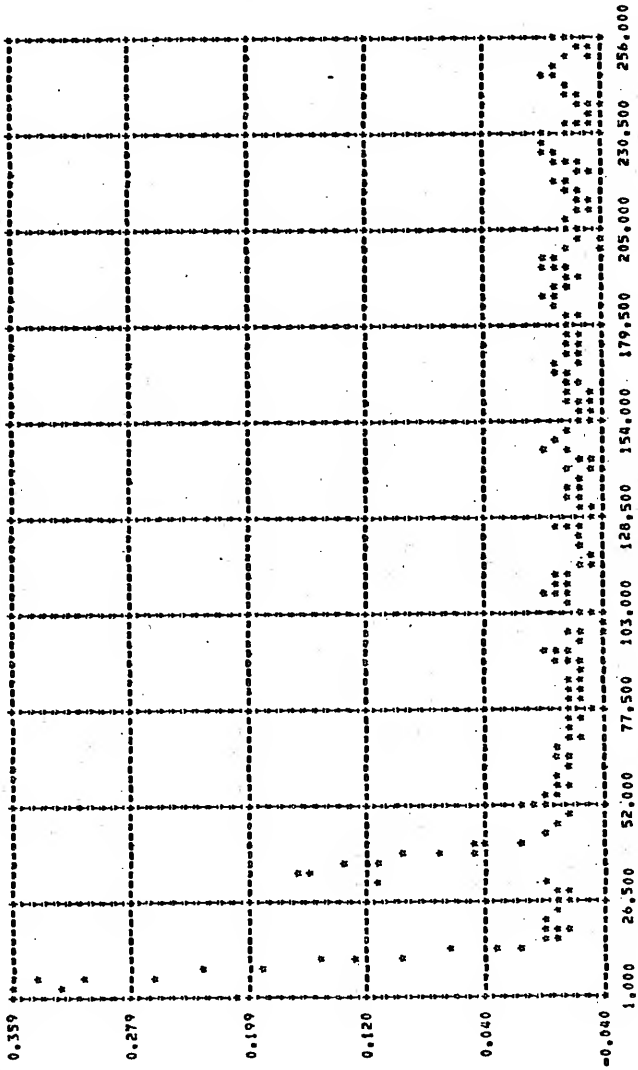


Figure 53 Composite Signal
SNR=14 dB

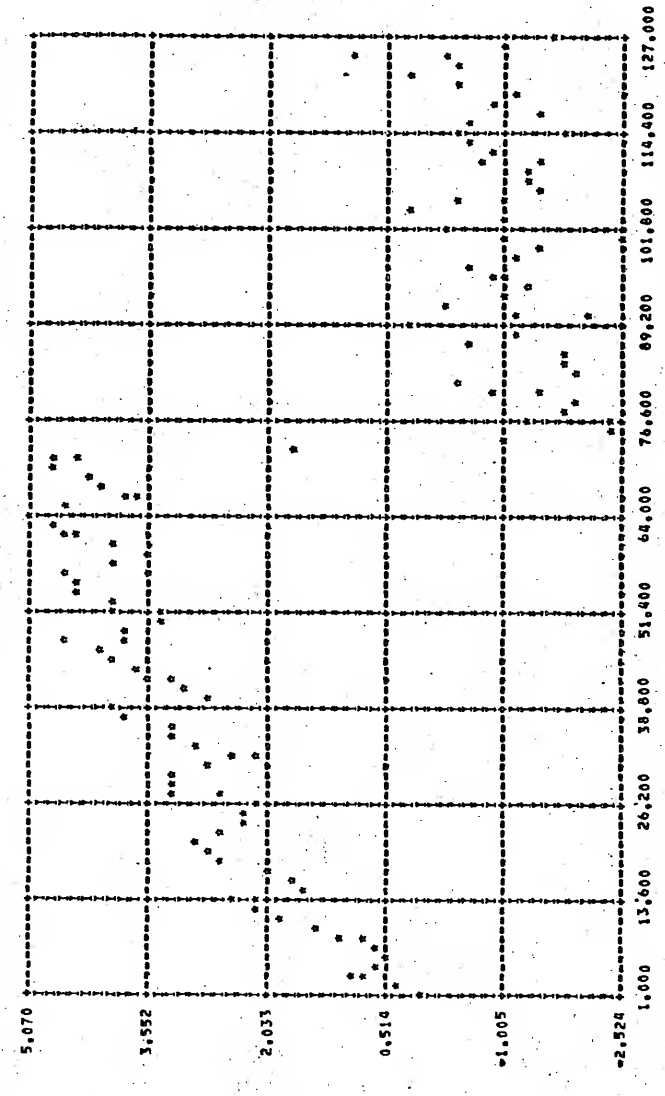


Figure 54 Unwrapped Phase Curve

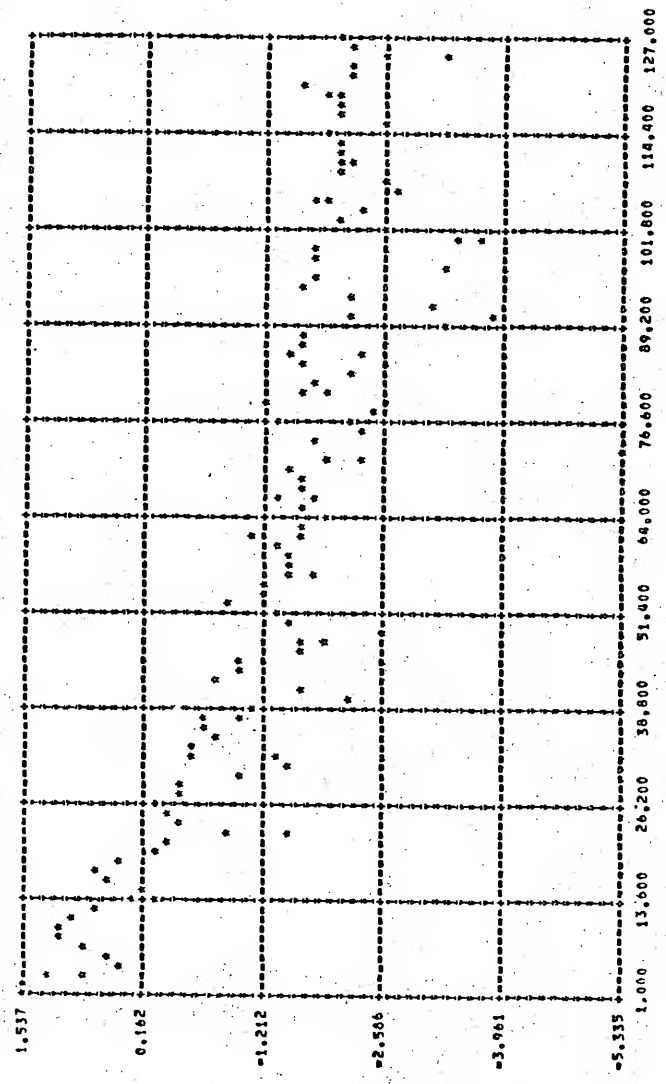


Figure 55 Log Magnitude

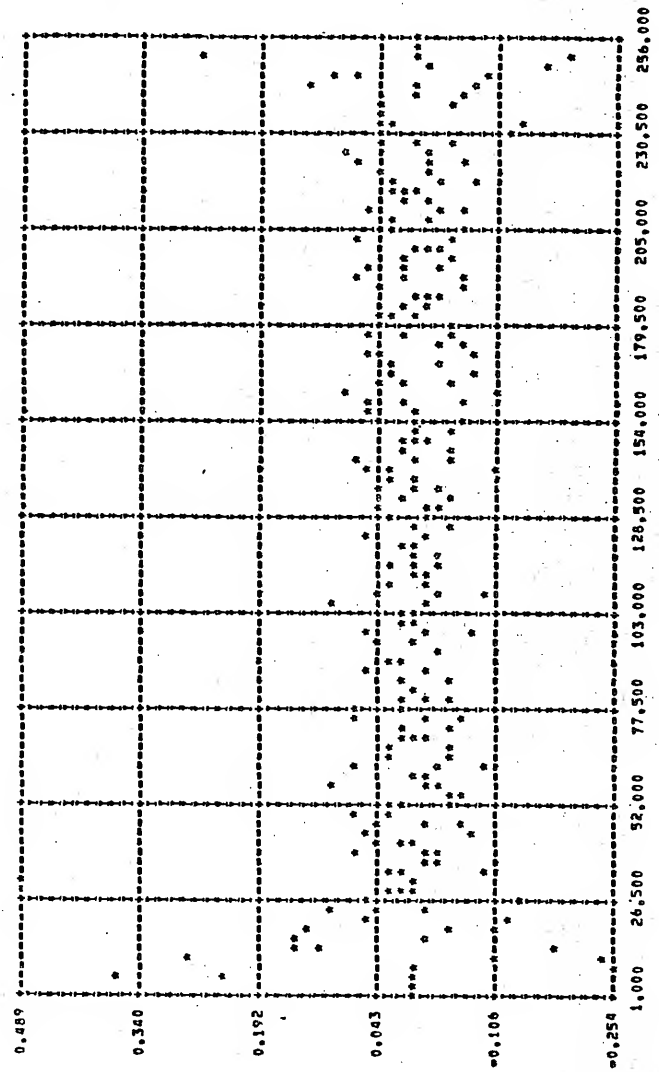


Figure 56 Complex Cepstrum

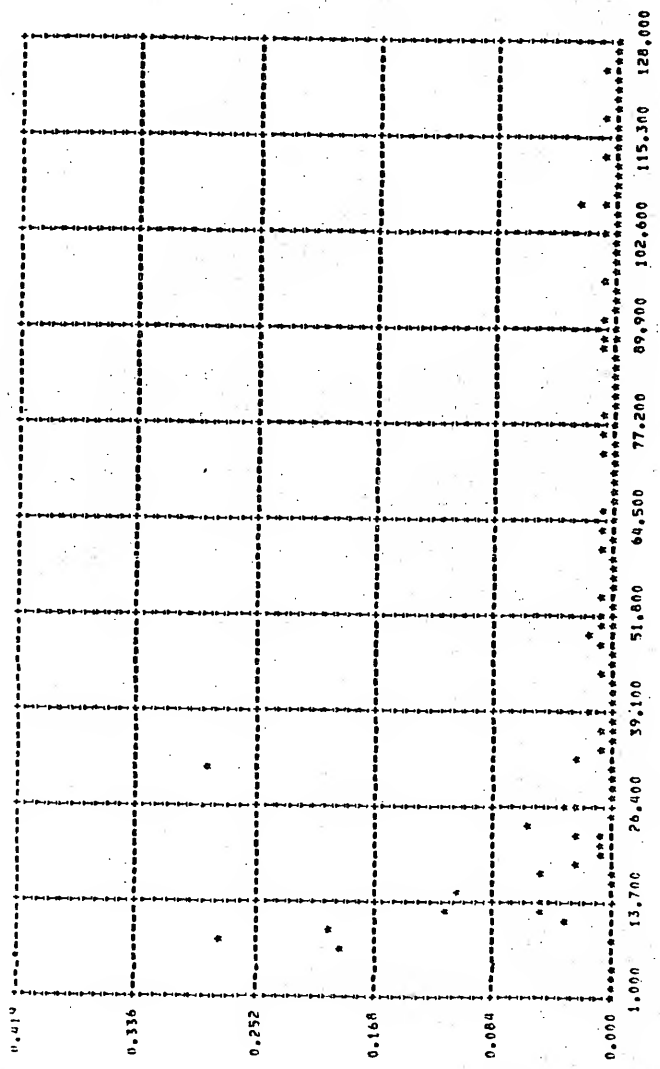


Figure 57 Phase Cepstrum

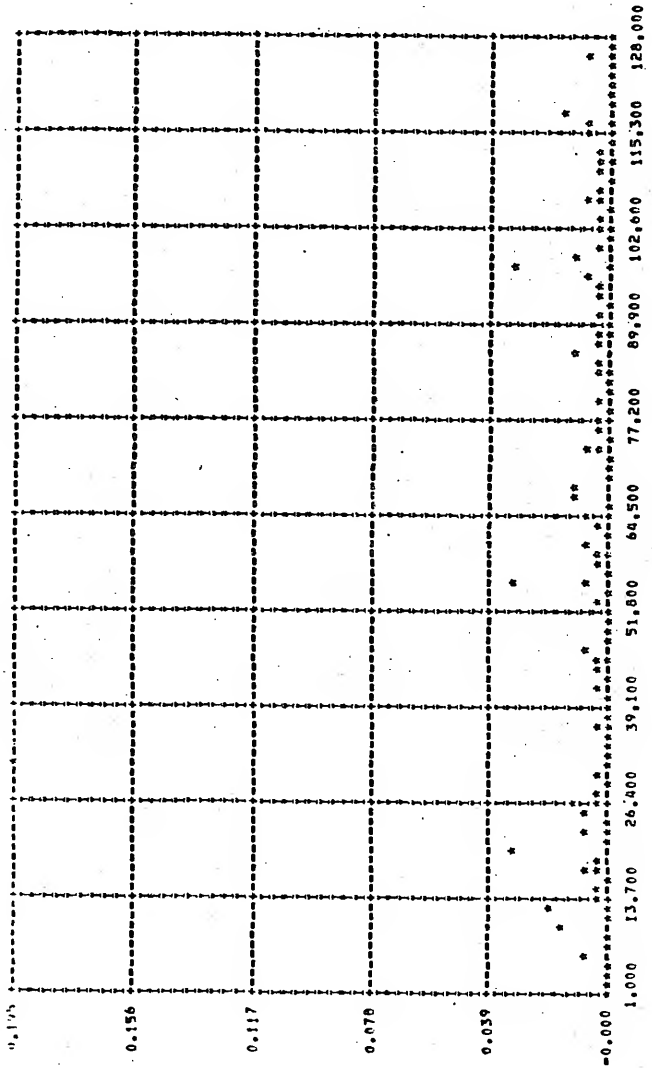


Figure 58 Power Cepstrum

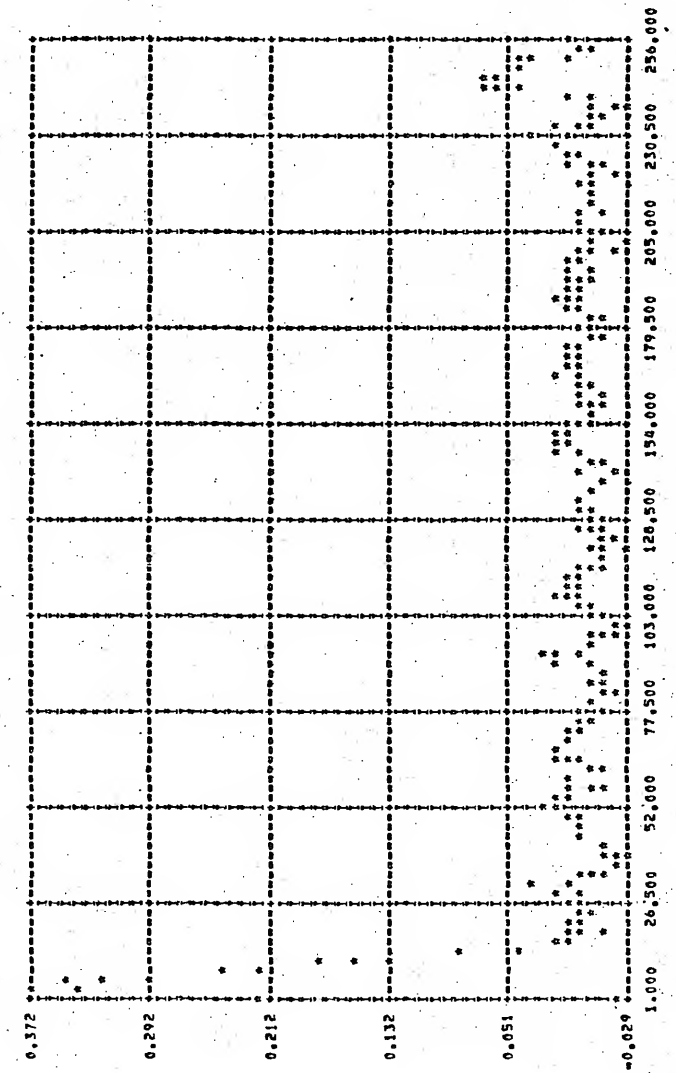


Figure 59 Recovered Wavelet
MSE = 2.63×10^{-4}

APPENDIX B

THE EFFECTS OF ADDITIVE NOISE ON THE CEPSTRA AT HIGH SNR

The effect of additive noise on the cepstra is extremely difficult to ascertain analytically. The following analysis provides some insight into the effect of additive noise at high SNR. The log spectrum of the composite signal plus noise is written

$$\hat{Y}(z) = \log Y(z) = \log[X(z)(1+az^{-n_o}) + N(z)] \quad . \tag{B-1}$$

$$\hat{Y}(z) = \log(X(z) + \frac{N(z)}{1+az^{-n_o}}) + \log(1+az^{-n_o}). \tag{B-2}$$

Assume $\frac{N(z)}{1+az^{-n_o}} << X(z)$, that is the SNR is large.

$$\hat{Y}(z) \simeq \log X(z) + \frac{N(z)}{1+az^{-n_o}} + \log (1+az^{-n_o}) \quad . \tag{B-3}$$

Assuming $a<1$, the terms $(1+az^{-n_o})^{-1}$ and $\log(1+az^{-n_o})$ may be expanded in a series, thus we obtain

$$Y(z) \simeq \log X(z) + N(z)(1-az^{-n_o}+a^2z^{-2n_o}\ldots) + az^{-n_o} - \frac{a^2}{2} z^{-2n_o} + \frac{a^3}{3}z^{-3n_o} \ldots \quad . \tag{B-4}$$

Inverse z-transforming the above equation, we obtain

$$\hat{y}(nT) \simeq \hat{x}(nT)+n(nT)*(\delta(n)-a\delta(n-n_o)+a^2\delta(n-2n_o)\ldots)+\hat{e}(nT)$$

where $\hat{y}(nT)$ and $\hat{x}(nT)$ are respectively the complex cepstra of composite

signal and the basic wavelet, n(nT) is the noise record, and $\hat{e}(nT)$ is the echo impulse train.

Thus the noise is spread throughout the complex cepstrum by convolution with the series of δ functions. Since $n(nT)=0$ for $n<0$ the presence of noise is the complex cepstrum at negative quefrencies will be due solely to aliasing. When aliasing is minimized (by the addition of zeroes to the input data) the complex cepstrum at negative quefrencies is observed to have virtually no distortion due to the addition of noise down to a SNR of 20 dB. This can be observed by comparing Figures 63 and 70. For $a>1$ a similar derivation to that given above yields

$$\hat{y}(nT) = \hat{x}(nT)+n(nT)*(\delta(n) - \frac{1}{a}\delta(n+n_o)+ \frac{1}{a^2}\delta(n+2n_o)\ldots)+\hat{e}(nT)$$

In this case regardless of aliasing noise will be present at both positive and negative quefrencies.

Example

Noise Contributions at Negative Quefrencies in the Complex Cepstrum

Figures 60 through 66

This group of figures illustrates the computation of the cepstra, and wavelet recovery at a SNR of 20 dB. Note the effects of noise throughout the complex cepstrum.

The composite signal (256 points) is

$$y(nT) = x(nT) + .5x(nT-30T)$$

where $x(nT) = nT\ e^{-nT}$ $\quad 0 \leq n < 64$

Figure Number	Figure Title
60	Composite Signal SNR = 20 dB
61	Unwrapped Phase Curve
62	Log Magnitude
63	Complex Cepstrum
64	Phase Cepstrum
65	Power Cepstrum
66	Recovered Wavelet MSE = 3.72×10^{-5}

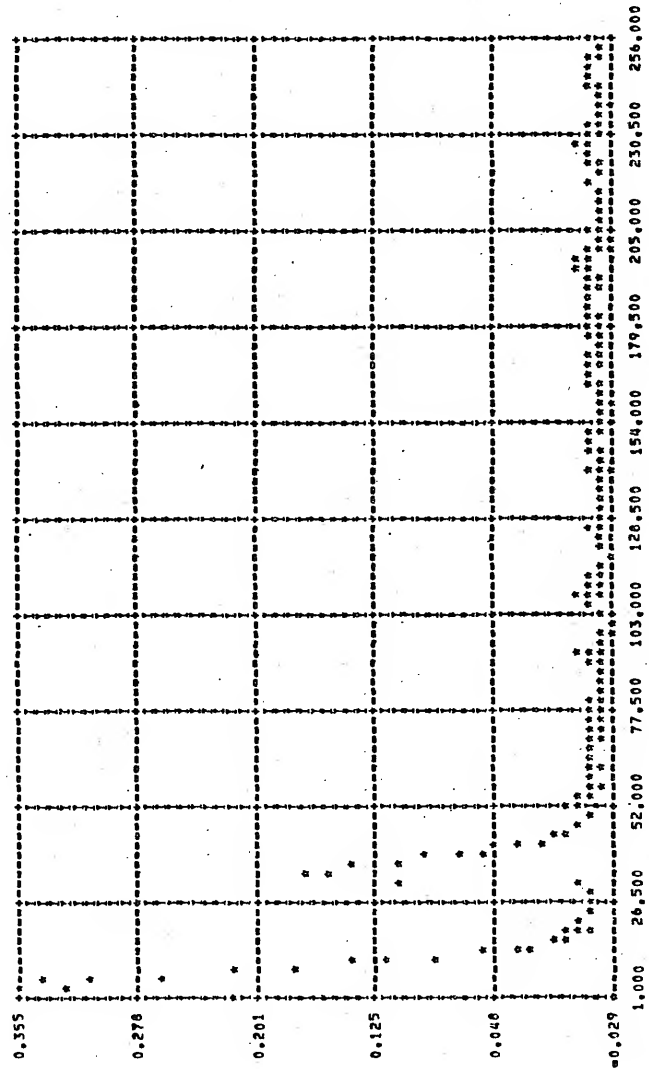


Figure 60 Composite Signal
SNR=20 dB

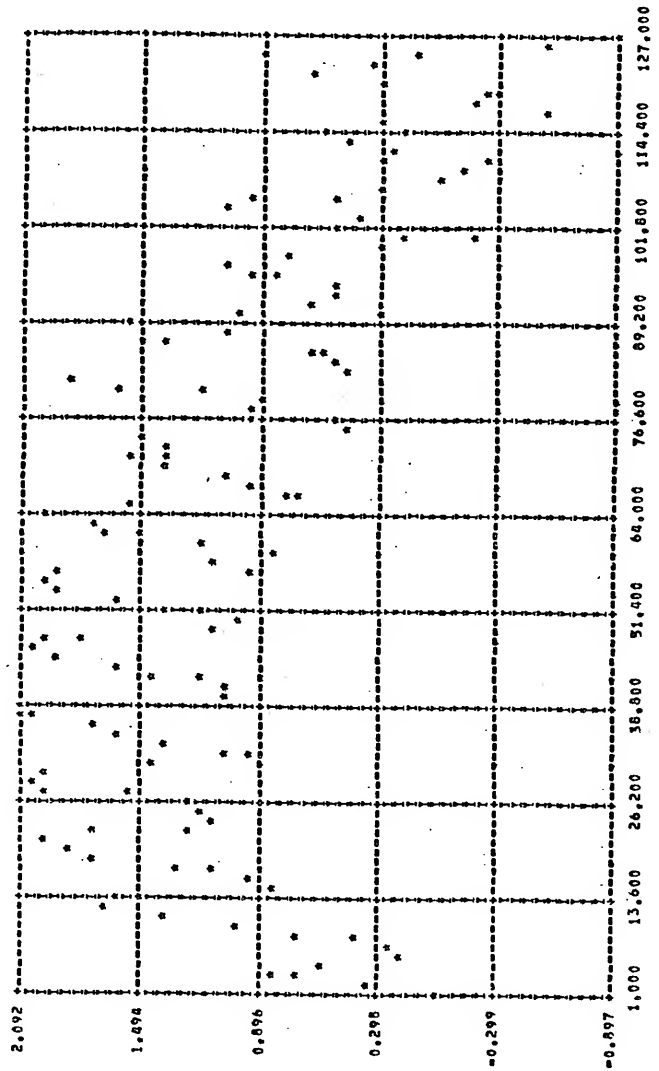


Figure 61 Unwrapped Phase Curve

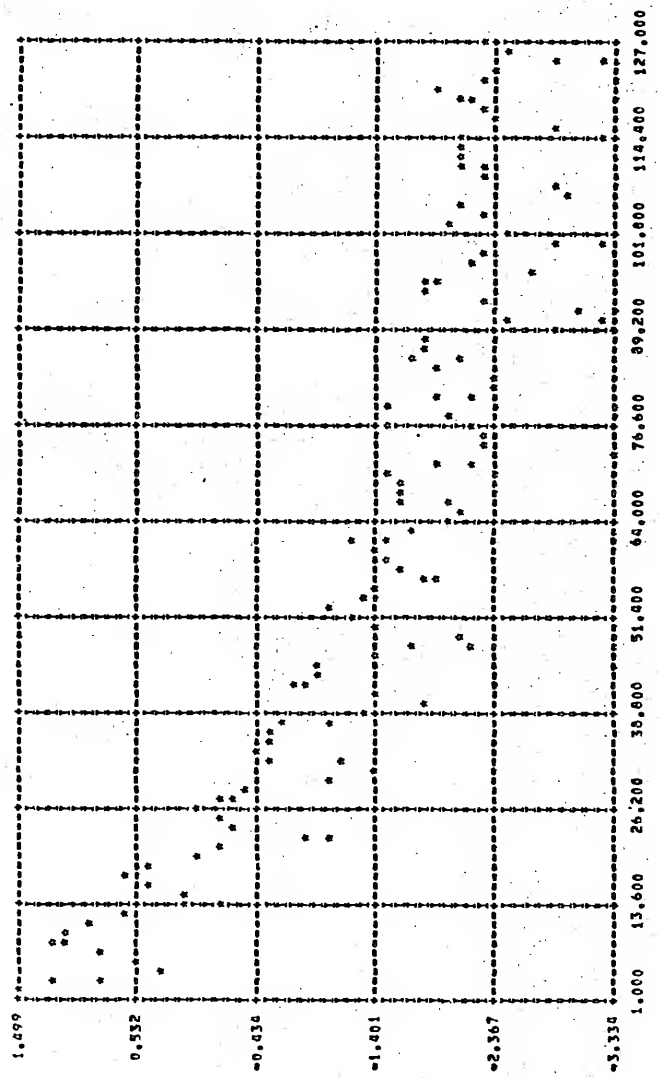


Figure 62 Log Magnitude

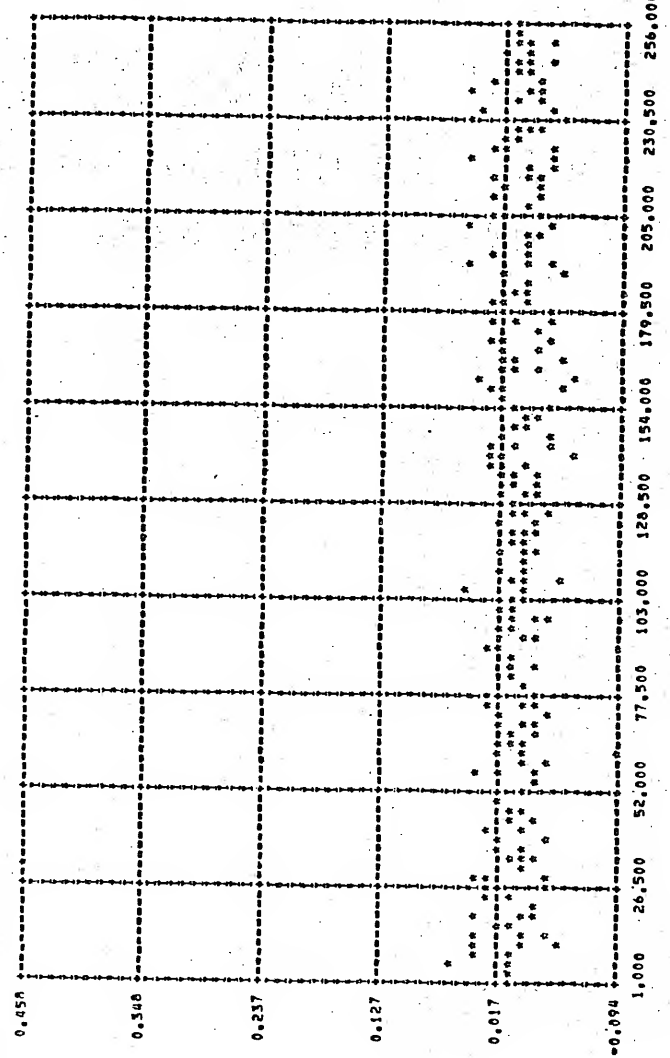


Figure 63 Complex Cepstrum

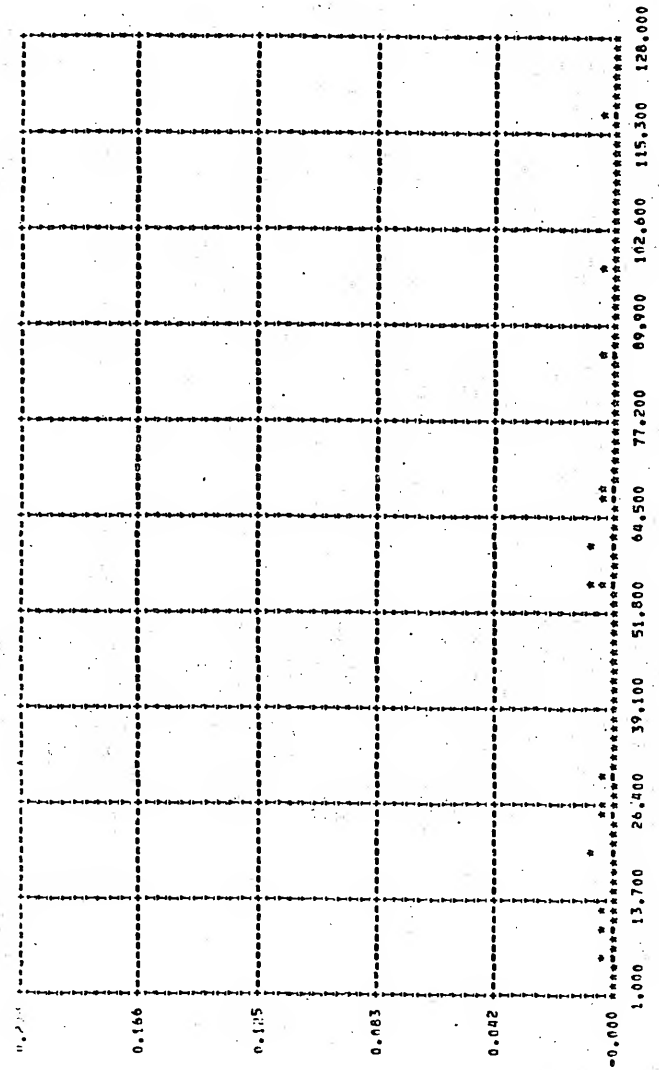


Figure 64 Phase Cepstrum

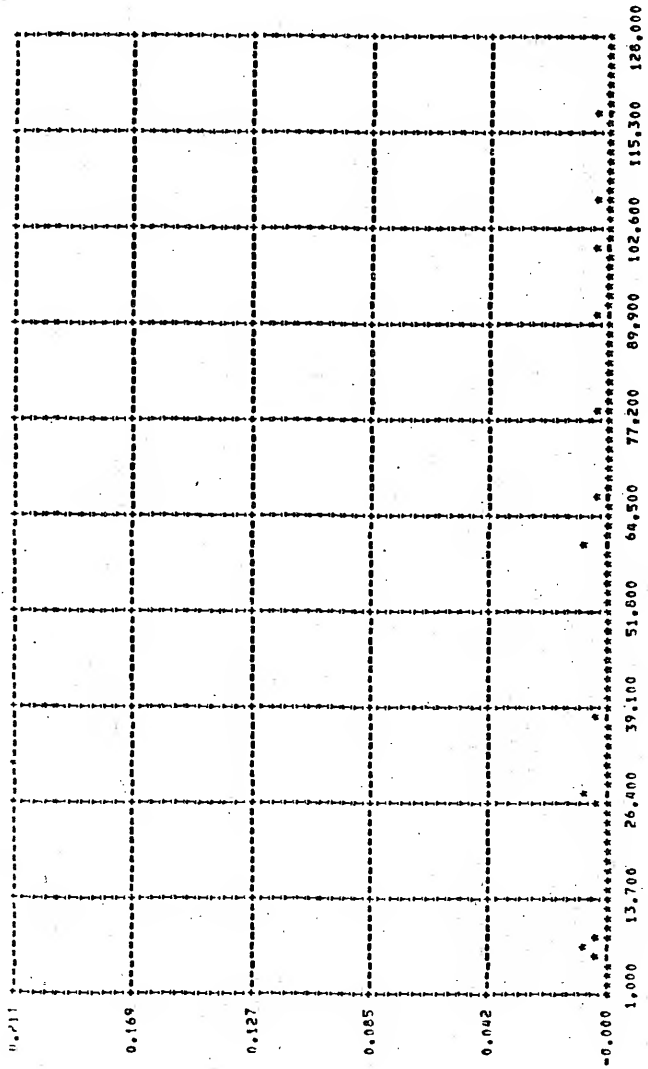


Figure 65 Power Cepstrum

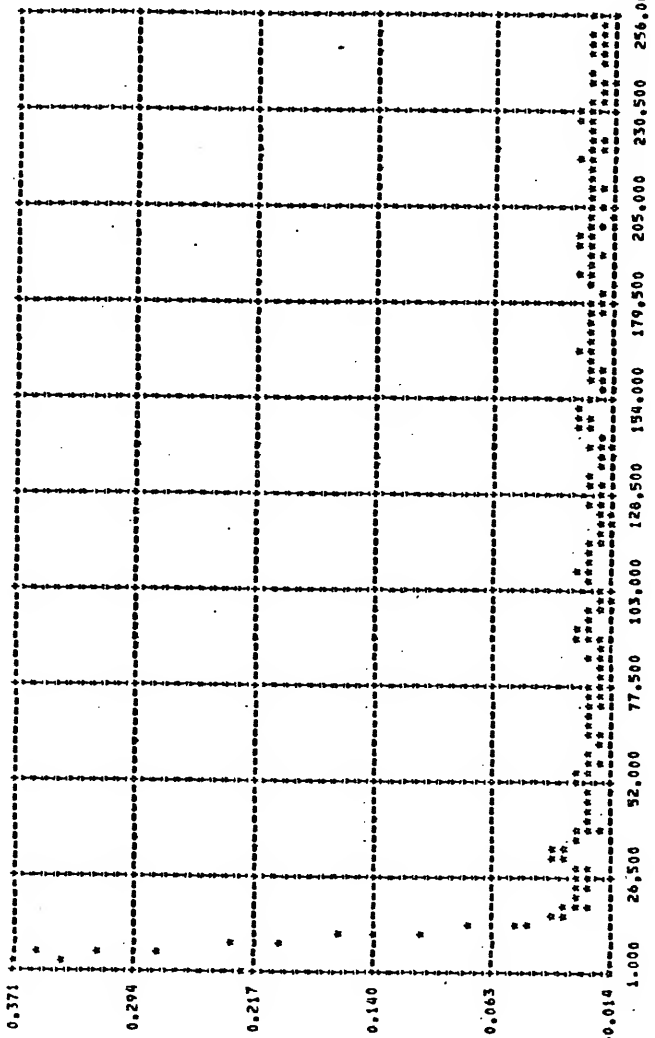


Figure 66 Recovered Wavelet
MSE = 3.72×10^{-5}

Example

Reduction of Noise Aliasing in the Complex Cepstrum
by Extending the Input Data Record with Zeroes

Figures 67 through 73

This group of figures illustrates the computation of the cepstra, and wavelet recovery when the input data record has been extended by adding 768 zeroes. Note the absence of noise effects at negative quefrencies in the complex cepstra.

The composite signal (256 points) is

$$y(nT) = x(nT) + .5x(nT-30T)$$

where $x(nT) = nT\, e^{-nT}$ $\quad 0 \leq n < 64$

Figure Number	Figure Title
67	Composite Signal (Zeroes Added, 1024 Points) SNR = 20 dB
68	Unwrapped Phase Curve
69	Log Magnitude
70	Complex Cepstrum
71	Phase Cepstrum
72	Power Cepstrum
73	Recovered Wavelet MSE = 3.13×10^{-5}

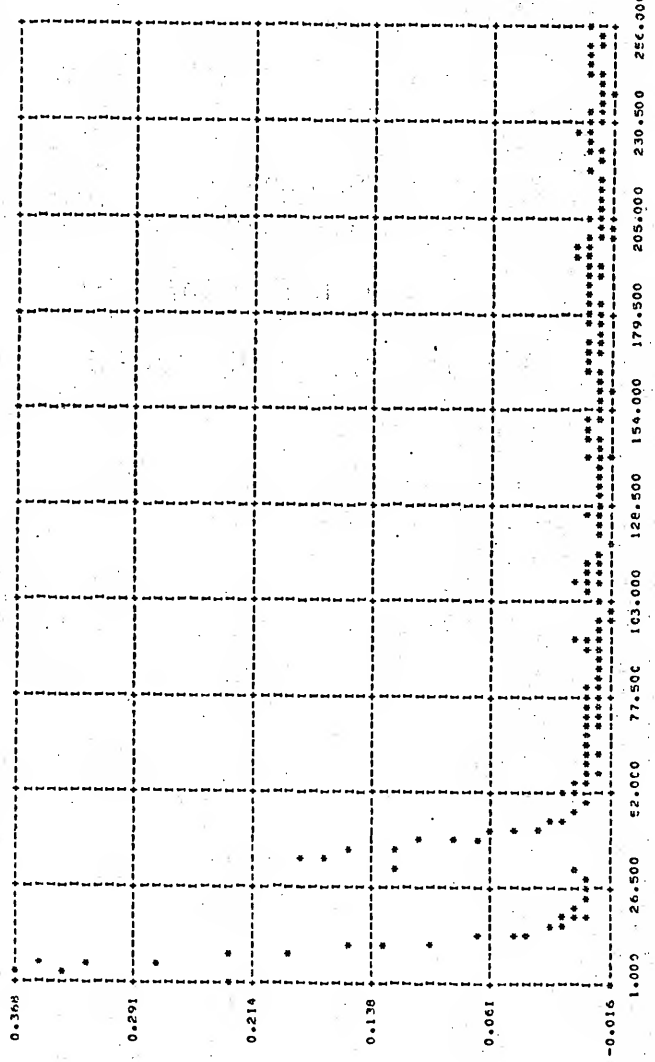


Figure 67 Composite Signal (Zeroes Added, 1024 Points)
SNR=20 dB

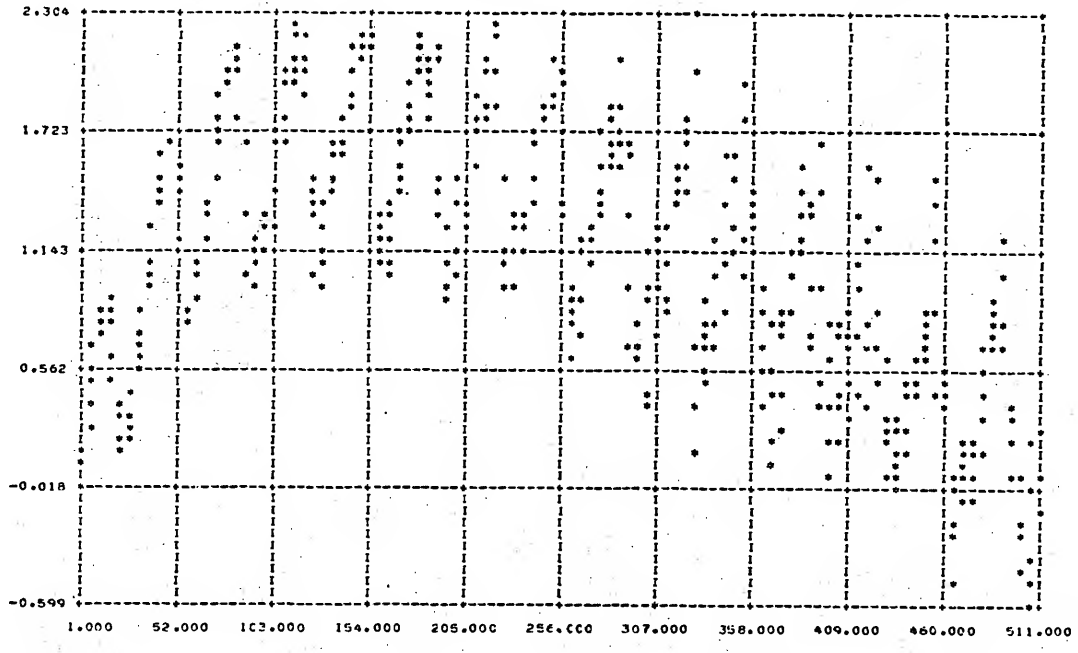


Figure 68 Unwrapped Phase Curve

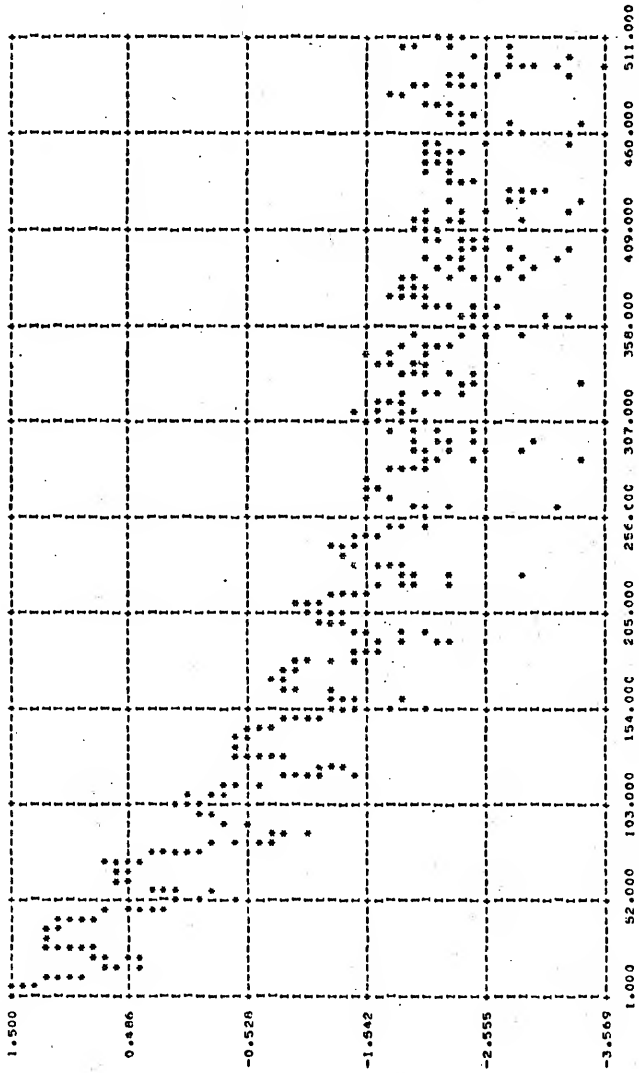


Figure 69 Log Magnitude

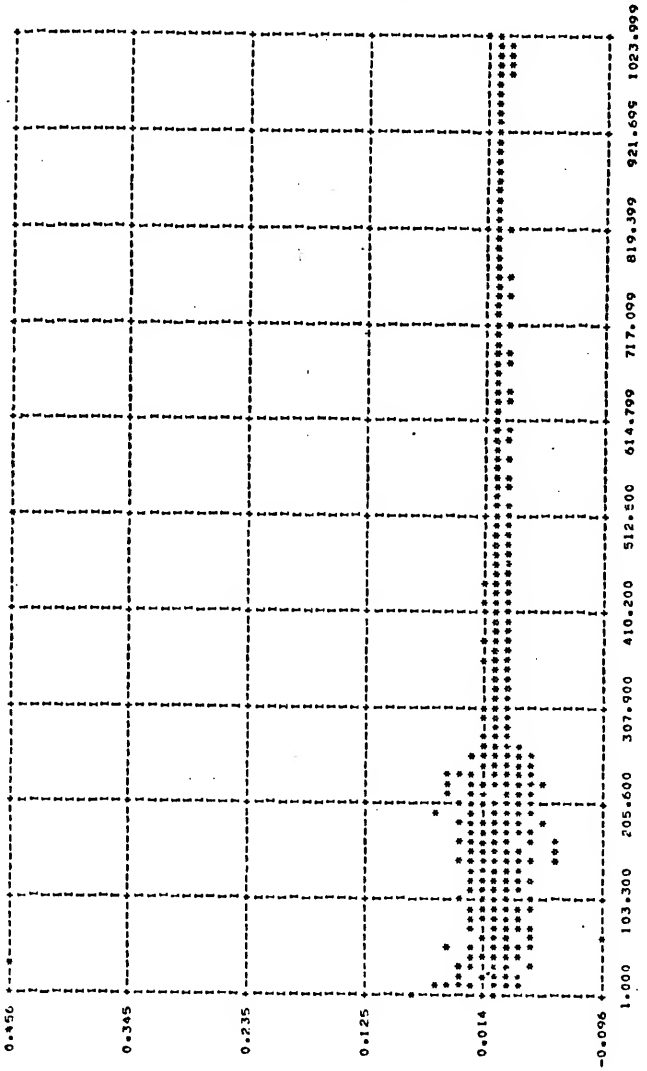


Figure 70 Complex Cepstrum

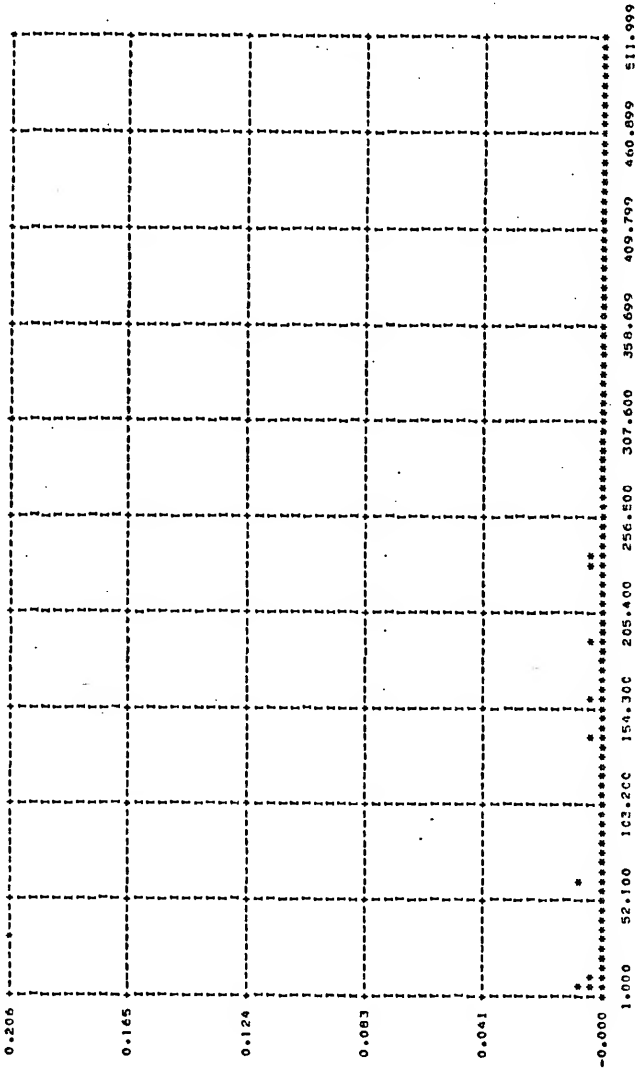


Figure 71 Phase Cepstrum

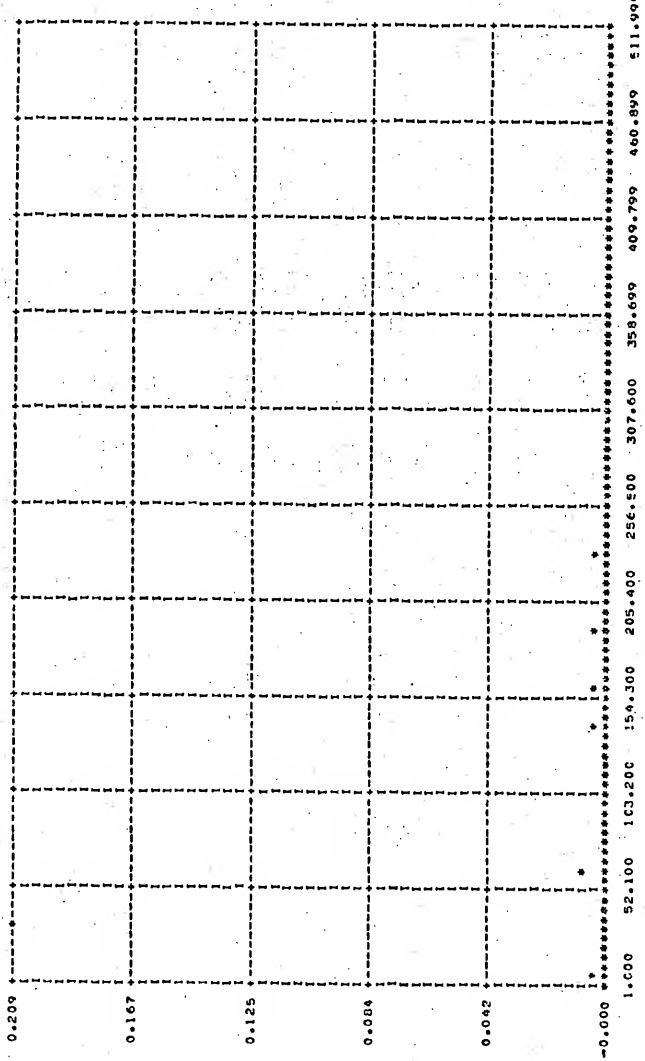


Figure 72 Power Cepstrum

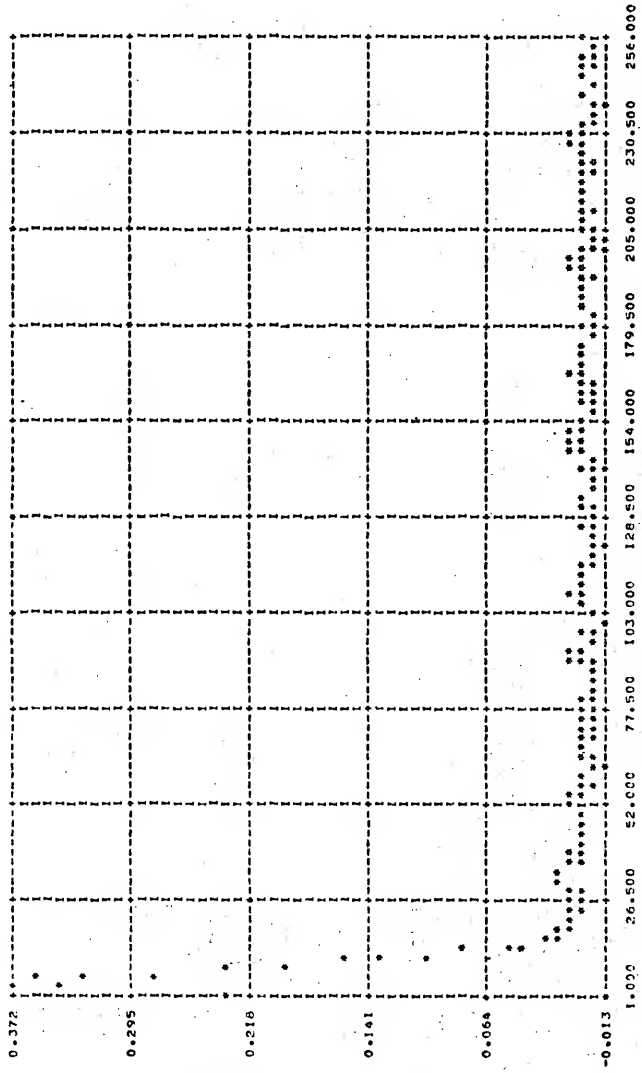


Figure 73 Recovered Wavelet
MSE = 3.13×10^{-5}

Appendix C

THE INVERSE TRANSFORM OF THE TRANSFORM OF A REAL VALUED SERIES UTILIZING THE FORWARD ALGORITHM

Consider the inverse DFT of the sequence X(n) where X(n) is the DFT of a real sequence.

$$x(k) + \frac{1}{N} \sum_{n=0}^{N-1} X(n)\, W_N^{nk} \qquad \text{(C-1)}$$

where $W_N^{nk} = e^{j\frac{2\pi kn}{N}} = \cos \frac{2\pi kn}{N} + j \sin \frac{2\pi kn}{N}$.

Noting Re X(n) = Re X(-n) and Im X(n) = -Im X(-n), we may rewrite equation (C-1) as

$$x(k) = \frac{1}{N} \sum_{n=0}^{N-1} \text{Re } X(n) \cos(\frac{2\pi kn}{N}) - \text{Im } X(n) \sin(\frac{2\pi kn}{N}) \; . \qquad \text{(C-2)}$$

The forward DFT of X(n) is

$$x'(k) = \sum_{n=0}^{N-1} X(n)\, W_N^{-kn} \qquad \text{(C-3)}$$

$$= \sum_{n=0}^{N-1} \text{Re } X(n) \cos(\frac{2\pi kn}{N}) + \text{Im } X(n) \sin(\frac{2\pi kn}{N}) \; . \qquad \text{(C-4)}$$

Comparing equations (C-2) and (C-4), we see

$$x(k) = \frac{1}{N} x'(-k) \quad . \qquad \text{(C-5)}$$

Thus the inverse transform of the transform of a real valued sequence can be easily computed utilizing the forward DFT algorithm.

APPENDIX D

PROGRAM LISTING

This appendix contains a listing of the main computer program and its associated subroutines used to compute the complex, phase and power cepstra, and to provide recovery of the basic wavelet.

```
0000        DIMENSION Y(258),SNOISE(256,10),X(256)
0001        COMMON C(64),S(64),KI(135),N,M
0002        VARSIG=.121218
0003        NREC=1
0004        CUTOFF=.75
0005        AMEAN=0.0
0006        PI=3.14159268
0007        M=8
0008        N=2**M
0009        NP2=N+2
0010        NS2=N/2
0011        NP1=N+1
0012        NM1=N-1
0013        ND2=N/2-1
0014        ND4=N/4
0015        JDELAY=30
0016        JDP1=JDELAY+1
0017        DELT=.3333333
0018        RECL=N
0019        EAMP=.5
0020        N0=60
0021        N0P1=N0+1
0022        JPN0=N0+JDELAY
0023        JNP1=JPN0+1
0024      C
0025      C ************************************************
0026      C INPUT MULTIPLIERS FOR FFT COMPUTATION.
0027      C ************************************************
0028      C
0029        DO 1 I=1,ND4
0030        READ 100,C(I),S(I)
0031      1 PRINT 100,C(I),S(I)
0032      C
0033      C ************************************************
0034      C INPUT REORDERING SEQUENCE FOR FFT COEFFICIENTS.
0035      C ************************************************
0036      C
0037        DO 8 J=1,ND2,15
0038        READ 200,(KI(J+I-1),I=1,15)
0039      8 PRINT 200,(KI(J+I-1),I=1,15)
0040        DO 9 I=1,ND2
0041      9 KI(I)=2*KI(I)+1
0042        DO 1000 IENT=1,6
0043        LSIG=256
0044        IF(IENT.GT.3) GO TO 2
0045        SN=100/(10.0**(IENT-1))
0046        IF(IENT.EQ.3) SN=10**9
0047        GO TO 3
0048      2 SN=5.0/(2**(IENT-4))
0049      3 CONTINUE
0050        STDEV=SQRT(VARSIG)/SN
```

```
0051        CALL BLRNDM(SNOISE,N,DELT,CUTOFF,STDEV,AMEAN,NREC)
0052      C
0053      C ************************************************
0054      C GENERATE COMPOSITE SIGNAL.
0055      C ************************************************
0056      C
0057        DO 4 I=1,64
0058        RI=I
0059      4 X(I)=SIN((.4+.11*(I-1)*DELT)*(I-1)*DELT)
0060        DO 6 I=65,N
0061      6 X(I)=0.0
0062        DO 10 I=1,N0
0063     10 CS(I)=SNOISE(I,1)
0064        DO 11 I=N0P1,JPN0
0065     11 CS(I)=X(I-N0)+SNOISE(I,1)
0066        DO 12 I=JNP1,N
0067     12 CS(I)=X(I-N0)+EAMP*X(I-JPN0)+SNOISE(I,1)
0068        CALL MEAN(CS,N,SM)
0069        CALL OPLOTN(CS,N)
0070      C
0071      C ************************************************
0072      C COMPUTE COMPLEX CEPSTRUM.
0073      C ************************************************
0074      C
0075        CALL BFFT(CS)
0076        CALL REO(CS)
0077        CALL MNINS(CS,N,SM)
0078        DO 13 I=1,NP2
0079     13 Z(I)=CS(I)
0080        DO 14 I=1,NP1,2
0081        CS(I)=(CS(I)*CS(I)+CS(I+1)*CS(I+1))
0082        IF (CS(I).LT..00000001) CS(I)=.00000001
0083        CS(I)=ALOG(CS(I))/2.0
0084     14 CONTINUE
0085        CALL PHASEU(Z,N)
0086        DO 15 I=2,NP2,2
0087     15 CS(I)=Z(I)
0088        CALL OPLOTN(CS,N)
0089        CALL PHREM(CS,N,SPH)
0090        CALL HARTI(CS)
0091        DO 20 I=1,N,2
0092        IP1=I+1
0093     20 PRINT 400,CS(I),I,CS(I+1),IP1
0094        DO 21 I=1,N
0095     21 Y(I)=CS(I)
0096        DO 22 I=1,5
0097        Y(I)=0.0
0098        Y(N-I+1)=0.0
0099     22 CONTINUE
0100        CALL OPLOTN(Y,N)
0101      C
```

```
0102      C ********************************************
0103      C COMPUTE POWER CEPSTRUM.
0104      C ********************************************
0105      C
0106         Y(1)=CS(1)*CS(1)*4
0107         DO 24 I=2,NS2
0108      24 Y(I)=(CS(I)+CS(N-I+2))**2
0109         DO 26 I=1,NS2,2
0110         IP1=I+1
0111      26 PRINT 400,Y(I),I,Y(I+1),IP1
0112         DO 27 I=1,5
0113         Y(I)=0.0
0114      27 Y(N-I+1)=0.0
0115         CALL OPLOTN(Y,NS2)
0116      C
0117      C ************************************************
0118      C COMPUTE PHASE CEPSTRUM.
0119      C ************************************************
0120      C
0121         DO 28  I=6,NS2
0122      28 Y(I)=(CS(I)-CS(N-I+2))**2
0123         CALL OPLOTN(Y,NS2)
0124         CALL FILTER(CS,JDELAY)
0125         CALL BFFT(CS)
0126         CALL REO(CS)
0127         CALL PHINS(CS,N,SPH)
0128         DO 30 I=1,NP2,2
0129         T1=CS(I)
0130         T2=CS(I+1)
0131         CS(I)=EXP(T1)*COS(T2)
0132         CS(I+1)=EXP(T1)*SIN(T2)
0133      30 CONTINUE
0134         CALL HARTI(CS)
0135         CALL OPLOTN(CS,N)
0136         DO 31 I=NOP1,N
0137      31 Y(I)=X(I-NO)
0138         DO 33 I=NOP1,N
0139      33 X(I)=Y(I)
0140         DO 32 I=1,NO
0141      32 X(I)=0.0
0142         PRINT 170,SN
0143      C
0144      C ************************************************
0145      C COMPUTE MSE BETWEEN RECOVERED AND BASIC WAVELET.
0146      C ************************************************
0147      C
0148         DO 1000 KJ=1,2
0149         IF(KJ.EQ.2) LSIG=64
0150         CALL MEAN(CS,LSIG,SM)
0151         CALL MEAN(X,LSIG,SM)
0152         DO 40 I=1,LSIG
```

```
0153     40 B(I)=(X(I)-CS(I))**2
0154        ERR=B(1)
0155        DO 42 I=2,LSIG
0156     42 ERR=B(I)+ERR
0157        ERR=ERR/LSIG
0158        PRINT 160, ERR
0159   1000 CONTINUE
0160    170 FORMAT('2','SNR=',E14.7)
0161    160 FORMAT('2','MSE=',E14.7)
0162    100 FORMAT(4E15.8)
0163    200 FORMAT(1514)
0164    300 FORMAT(1X,E16.8,2X,E16.8,2X,I4,2X,E10.4)
0165    400 FORMAT(1X,E16.8,2X,I4,5X,E16.8,2X,I4)
0166        END
```

```
0000        SUBROUTINE BFFT(B)
0001      C
0002      C ************************************************
0003      C THIS SUBROUTINE PERFORMS A FORWARD DFT UTILIZING
0004      C THE BERGLAND FFT ALGORITHM.
0005      C ************************************************
0006      C
0007        DIMENSION B(258)
0008        COMMON C(64),S(64),KI(135),N,M
0009        INTEGER REP,DISP,SEP
0010        MM1=M-1
0011        DO 1 I=1,MM1
0012        REP=2**(I+1)
0013        DISP=REP/4
0014        SEP=N/REP
0015        DO 1 J=1,DISP
0016        DO 1 K=1,SEP
0017        J1=K+(J-1)*N/DISP
0018        J2=J1+SEP
0019        J3=J2+SEP
0020        J4=J3+SEP
0021        IF (J.NE.1) GO TO 2
0022        T=B(J2)
0023        B(J2)=B(J3)
0024        B(J3)=T
0025      2 T1=B(J2)*C(J)+B(J4)*S(J)
0026        T2=-B(J2)*S(J)+B(J4)*C(J)
0027        B(J2)=B(J3)+T2
0028        B(J4)=-B(J3)+T2
0029        B(J3)=B(J1)-T1
0030      1 B(J1)=B(J1)+T1
0031        T=B(1)
0032        B(1)=B(1)+B(2)
0033        B(2)=T-B(2)
0034        RETURN
0035        END
0036        SUBROUTINE REO(B)
0037      C
0038      C ************************************************
0039      C THIS SUBROUTINE REORDERS THE DFT COEFFICIENTS
0040      C OUTPUT BY THE BERGLAND FFT ALGORITHM.
0041      C ************************************************
0042      C
0043        DIMENSION B(258),TEMP(258)
0044        COMMON C(64),S(64),KI(135),N,M
0045        NN1=N-1
0046        NP1=N+1
0047        TEMP(1)=B(1)
0048        TEMP(2)=0.0
0049        TEMP(N+1)=B(2)
0050        TEMP(N+2)=0.0
```

```
0051        DO 2 I=3,NM1,2
0052        TEMP(I)=B(KI(I/2))
0053        TEMP(I+1)=B(KI(I/2)+1)
0054      2 CONTINUE
0055        DO 4 I=1,NP1,2
0056        B(I)=TEMP(I)
0057      4 B(I+1)=TEMP(I+1)
0058        RETURN
0059        END
0060        SUBROUTINE HARTI(B)
0061      C
0062      C ************************************************
0063      C THIS SUBROUTINE PERFORMS AN INVERSE DFT WITH THE
0064      C THE HARTWELL MODIFIED BERGLAND FFT ALGORITHM.
0065      C ************************************************
0066      C
0067        DIMENSION B(258),F(258)
0068        COMMON C(64),S(64),KI(135),N,M
0069        NP1=N+1
0070        DO 2 I=1,NP1,2
0071        F((I+1)/2)=B(I)+B(I+1)
0072        F(N-(I+1)/2+2)=B(I)-B(I+1)
0073      2 CONTINUE
0074        CALL BFFT(F)
0075        CALL REO(F)
0076        DO 4 I=1,NP1,2
0077        B((I+1)/2)=(F(I)+F(I+1))/N
0078        B(N-(I+1)/2+2)=(F(I)-F(I+1))/N
0079      4 CONTINUE
0080        RETURN
0081        END
0082        SUBROUTINE PHASEU(A,NPT)
0083      C
0084      C ************************************************
0085      C THIS SUBROUTINE UNWRAPS THE PHASE OF THE LOG
0086      C SPECTRUM.
0087      C ************************************************
0088      C
0089        DIMENSION P(128),A(258)
0090        NS2=NPT/2
0091        NP2=NPT+2
0092        NPI=0
0093        OLDP=0.0
0094        PI=3.1415927
0095        A(NP2)=0.0
0096        DO 1 I=1,NS2
0097        P(I)=ATAN2(A(2*I),A(2*I-1))
0098        IF (I.EQ.1) GO TO 2
0099        IF ((P(I)-OLDP).GT.PI) NPI=NPI-1
0100        IF ((OLDP-P(I)).GT.PI) NPI=NPI+1
0101      2 OLDP=P(I)
```

```
0102        A(2*I)=P(I)+NPI*PI*2.0
0103      1 CONTINUE
0104        RETURN
0105        END
0106        SUBROUTINE PHREM(X,NPT,S)
0107      C
0108      C ************************************************
0109      C THIS SUBROUTINE REMOVES THE LINEAR PHASE TERM
0110      C FROM THE UNWRAPPED PHASE CURVE.
0111      C ************************************************
0112      C
0113        DIMENSION X(258)
0114        PI=3.14159268
0115        S=X(NPT)/PI
0116        DO 2 I=2,NPT,2
0117      2 X(I)=X(I)-PI*S*(I-2)/(NPT-2)
0118        PRINT 1,S
0119      1 FORMAT('2','S=',F10.6)
0120        RETURN
0121        END
0122        SUBROUTINE PHINS(X,NPT,S)
0123      C
0124      C ************************************************
0125      C THIS SUBROUTINE INSERTS THE LINEAR PHASE TERM
0126      C REMOVED BY SUBROUTINE PHREM TO AVOID SHIFTING
0127      C THE RECOVERED WAVELET.
0128      C ************************************************
0129      C
0130        DIMENSION X(258)
0131        PI=3.14159268
0132        DO 2 I=2,NPT,2
0133      2 X(I)=X(I)+PI*S*(I-2)/(NPT-2)
0134        RETURN
0135        END
0136        SUBROUTINE FILTER(F,JDELAY)
0137      C
0138      C ************************************************
0139      C THIS SUBROUTINE SMOOTHS THE ECHO PEAKS PRESENT
0140      C IN THE COMPLEX CEPSTRUM.
0141      C ************************************************
0142      C
0143        DIMENSION F(258)
0144        N=256
0145        DO 2 I=1,10
0146        IF(I*JDELAY.GT.192) GO TO 4
0147        F(I*JDELAY+1)=(F(I*JDELAY+3)+F(I*JDELAY-1))/2.0
0148      2 CONTINUE
0149      4 RETURN
0150        END
```

```
      SUBROUTINE MEAN(F,NPT,SUM)
C
C ************************************************
C THIS SUBROUTINE REMOVES THE MEAN OF THE REAL
C ARRAY F.
C ************************************************
C
      DIMENSION F(258)
      SUM=0.0
      DO 1 I=1,NPT
    1 SUM=SUM+F(I)
      SUM=SUM/NPT
      DO 2 I=1,NPT
    2 F(I)=F(I)-SUM
      SUM=SUM*NPT
      RETURN
      END
      SUBROUTINE MNINS(F,NPT,SUM)
C
C ************************************************
C THIS SUBROUTINE REINSERTS THE DC TERM REMOVED
C BY SUBROUTINE MEAN.
C ************************************************
C
      DIMENSION F(258)
      F(1)=SUM
      RETURN
      END
      SUBROUTINE BLRNDM(SNOISE,NSAMP,TAU,ALFANZ,STDEV,
     1ANEAN,NREC)
C
C ************************************************
C THIS SUBROUTINE PRODUCES BANDLIMITED RANDOM
C NOISE.
C ************************************************
C
      IMPLICIT INTEGER*4(I-N),REAL*4(A-H,O-Z,$)
      REAL*4 GAUSN(256,10),XNOISE(257,10),SNOISE(256,10)
      PI=3.14159268
      ALFARD=ALFANZ*2*PI
      ARG=-ALFARD*TAU
      NS1=NSAMP+1
      EXPARG=EXP(ARG)
      SIGMA=STDEV*SQRT(1-EXP(2.*ARG))
      DO 10 J=1,NREC
      ISTART=2*J-1
      DO 10 I=1,NSAMP
      CALL GAUSS(ISTART,SIGMA,ANEAN,GVAR)
   10 GAUSN(I,J)=GVAR
      DO 20 J=1,NREC
      XNOISE(1,J)=0.0
```

```
0202        XNOISE(1,J)=0.0
0203        DO 20 I=2,NS1
0204        XNOISE(I,J)=EXPARG*XNOISE(I-1,J)+GAUSN(I-1,J)
0205     20 SNOISE(I-1,J)=XNOISE(I,J)
0206        PRINT 5,SIGMA
0207        SIGMA=0.0
0208        DO 30 I=1,256
0209        SIGMA=SIGMA+SNOISE(I,1)*SNOISE(I,1)
0210     30 CONTINUE
0211        SIGMA=SIGMA/NSAMP
0212        SIGMA=SQRT(SIGMA)
0213        PRINT 5,SIGMA
0214        RETURN
0215      5 FORMAT(//,2X,'STANDARD DEVIATION OF OUTPUT
0216       1 VARIATES=',F14.6)
0217        END
0218        SUBROUTINE GAUSS(IX,S,AM,V)
0219      C
0220      C ************************************************
0221      C THIS SUBROUTINE GENERATES GAUSSIANLY DISTRIBUTED
0222      C RANDOM VARIABLES.
0223      C ************************************************
0224      C
0225        A=0.0
0226        DO 50 I=1,12
0227        CALL RANDU(IX,IY,Y)
0228        IX=IY
0229     50 A=A+Y
0230        V=(A-6.0)*S+AM
0231        RETURN
0232        END
0233        SUBROUTINE RANDU(IX,IY,YFL)
0234      C
0235      C ************************************************
0236      C THIS SUBROUTINE GENERATES RANDOM VARIABLES.
0237      C ************************************************
0238      C
0239        IY=IX*65539
0240        IF(IY)5,6,6
0241      5 IY=IY+2147483647+1
0242      6 YFL=IY
0243        YFL=YFL*.4656613E-9
0244        RETURN
0245        END
0246        SUBROUTINE EXW(DATA,NPT)
0247      C
0248      C ************************************************
0249      C THIS SUBROUTINE IS USED TO EXPONENTIALLY WINDOW
0250      C THE INPUT DATA RECORD.
0251      C ************************************************
0252      C
```

```
0253        DIMENSION DATA(258)
0254        A=.99
0255        PI=3.14159268
0256        DO 2 I=1,NPT
0257      2 DATA(I)=DATA(I)*(A**(I-1))
0258        RETURN
0259        END
0260        SUBROUTINE EXWCOR(F,NPT)
0261      C
0262      C ************************************************
0263      C THIS SUBROUTINE IS USED TO CORRECT THE RECOVERED
0264      C WAVELET FOR THE EXPONENTIAL WINDOW.
0265      C ************************************************
0266      C
0267        DIMENSION F(258)
0268        A=.99
0269        PI=3.14159268
0270        DO 2 I=1,256
0271      2 F(I)=F(I)/(A**(I-1))
0272        RETURN
0273        END
```

BIBLIOGRAPHY

[1] D.G. Childers, R.S. Varga, and N.W. Perry, Jr., "Composite Signal Decomposition," IEEE Trans. Audio Electroacoust., Vol. AU-18, pp.471-477, December, 1970.

[2] T.Y. Young, "Epoch Detection - A Method for Resolving Overlapping Signals," Bell Syst. Tech. J., Vol. 44, pp.401-425, March, 1965.

[3] R.C. Kemerait, "Signal Detection and Extraction by Cepstrum Techniques," Doctoral Dissertation, University of Florida, 1971.

[4] B. Bogert, M. Healy, and J. Tukey, The Quefrency Analysis of Time Series for Echoes, M. Rosenblatt, Ed., New York, Wiley, 1963, Ch. 15, pp.209-243.

[5] G.M. Bryan, P. Buhl, and P.L. Stoffa, "The Application of Homomorphic Deconvolution to Shallow-Water Marine Seismology - Part I: Models," Geophysics, Vol. 39, No. 4, pp.401-416, August, 1974.

[6] A.M. Noll, "Short-Time Spectrum and Cepstrum Techniques for Vocal Pitch Detection," J. Acoust. Soc. Am., Vol. 36, No. 2, pp.296-302, February, 1964.

[7] R.W. Shafer, "Echo Removal by Discrete Generalized Linear Filtering," Doctoral Dissertation, M.I.T., 1968.

[8] B.P. Bogert and J.F. Ossanna, "The Heuristics of Cepstrum Analysis of a Stationary Complex Echoed Gaussian Signal in Stationary Gaussian Noise," IEEE Trans. Infor. Theory, Vol. IT-12, No. 3, pp. 373-380, July, 1966.

[9] O.S. Halpeny, "Epoch Detection by Cepstrum Analysis," Master's Thesis, University of Florida, Gainesville, Florida, 1970.

[10] A.V. Oppenheim, "Superposition in a Class of Nonlinear Systems," Technical Report 432, Research Laboratory of Electronics, M.I.T., May, 1965.

[11] G.D. Bergland, "A Guided Tour of the Fast Fourier Transform," IEEE Spectrum, Vol. 6, No. 7, pp.41-52, July, 1969.

[12] G.D. Bergland and H.W. Hale, "Digital Real-Time Spectral Analysis," IEEE Trans. Electronic Comp., Vol. EC-16, pp. 180-185, April, 1967.

[13] G.D. Bergland, "A Fast Fourier Transform Algorithm for Real Valued Series," Comm. ACM, Vol. 11, No. 10, pp. 703-710, October, 1968.

[14] J.W. Hartwell, "A Procedure for Implementing Fast Fourier Transform on Small Computers," IBM J. Res. Develop., Vol. 15, No. 5, pp. 355-363, September, 1971.

[15] G.D. Bergland, "Fast Fourier Transform Hardware Implementations - An Overview," IEEE Trans. Audio Electroacoust., Vol. AU-17, No. 2, pp.104-108, June, 1969.

[16] W.H. Specker, "A Class of Algorithms for ln x, exp x, sin x, cos x, $\tan^{-1}x$, $\cot^{-1}x$," IEEE Trans. Electronic Computers, Vol. EC-14, pp. 85-86, January, 1965.

ADDITIONAL REFERENCES

T.C. Chen, "Automatic Computation of Exponentials, Logarithms, Ratios, and Square Roots," IBM J. Res. Develop., Vol. 16, pp.380-388, July, 1972.

D.G. Childers, R.S. VArga, and N.W. Perry, Jr., "Composite Signal Decomposition," IEEE Trans. Audio Electroacoust., Vol. AU-18, pp.471-477, December, 1970.

J.W. Cooley and J.W. Tukey, "An Algorithm for the Machine Calculation of Complex Fourier Series," Math. Computation, Vol.19, No.90, pp.297-301, April, 1965.

H.L. Groginsky and G.A. Works, "A Pipeline Fast Fourier Transform," IEEE Trans. on Comp., Vol. C-19, pp. 1015-1019, November, 1970.

R.C. Kemerait and D.G. Childers, "Signal Detection and Extraction by Cepstrum Techniques," IEEE Trans. Info. Theory, Vol. IT-18, No. 6, pp.745-759, November, 1972.

A.V. Oppenheim, R.W. Shafer, and T.G. Stockham, Jr., "Nonlinear Filtering of Multiplied and Convolved Signals," Proceedings of the IEEE, Vol. 56, No.8, pp.1264-1291, August, 1968.

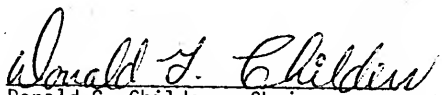
R.R. Shively, "A Digital Processor to Generate Spectra in Real Time," IEEE Trans. on Comp., Vol. C-17, pp.485-491, May, 1968.

BIOGRAPHICAL SKETCH

David P. Skinner was born on November 5, 1947, in Panama City, Florida. In August, 1969, he graduated Magna Cum Laude from Florida State University with the Bachelor of Science degree in Physics. He received the Master of Science degree in Physics from Florida State University in December, 1970. After attending Michigan State University for one year, he was employed by the Naval Research Laboratory in Washington, D.C. Since September, 1972, he has been pursuing the degree Doctor of Philosophy.

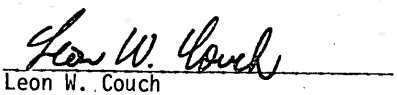
He is married to the former Betty Marquette of Miami, Florida. He is a member of the Institute of Electrical and Electronic Engineers, Phi Beta Kappa, and Sigma Pi Sigma.

I certify that I have read this study and that in my opinion it conforms to acceptable standards of scholarly presentation and is fully adequate, in scope and quality, as a dissertation for the degree of Doctor of Philosophy.



Donald G. Childers, Chairman
Professor of Electrical Engineering

I certify that I have read this study and that in my opinion it conforms to acceptable standards of scholarly presentation and is fully adequate, in scope and quality, as a dissertation for the degree of Doctor of Philosophy.



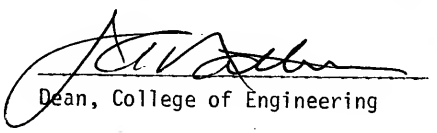
Leon W. Couch
Associate Professor of
Electrical Engineering

I certify that I have read this study and that in my opinion it conforms to acceptable standards of scholarly presentation and is fully adequate, in scope and quality, as a dissertation for the degree of Doctor of Philosophy.

Zoran Pop Stojanovic
Professor of Mathematics

This dissertation was submitted to the Graduate Faculty of the College of Engineering and to the Graduate Council, and was accepted as partial fulfillment of the requirements for the degree of Doctor of Philosophy.

December, 1974



Dean, College of Engineering

Dean, Graduate School