MULTI-CHANNEL HOMOMORPHIC WAVELET ESTIMATION

by

MARK CHRISTOPHER LANE

B.Sc., The University Of Victoria, 1980

A THESIS SUBMITTED IN PARTIAL FULFILMENT OF
THE REQUIREMENTS FOR THE DEGREE OF
MASTER OF SCIENCE

in

THE FACULTY OF GRADUATE STUDIES
Department Of Geophysics And Astronomy

We accept this thesis as conforming
to the required standard

THE UNIVERSITY OF BRITISH COLUMBIA
November 1983

Department of Geophysics and Astronomy

The University of British Columbia
1956 Main Mall
Vancouver, Canada
V6T 1Y3

Date Nov. 17, 1983

Abstract

Wavelet estimation can be posed as a multi-channel common information problem. Each channel of data is modeled as the convolution of a wavelet with an impulse sequence. A homomorphic transform maps the data from a convolutional to an additive space. The mapping may also effect partial separation of wavelet and impulses. In the additive space the wavelet can be estimated using averaging. This is termed cepstral averaging.

This thesis reviews the homomorphic transform and provides a synthesis and comparison of the techniques available for its realization. The method of principal components for wavelet estimation is proposed as an alternative to cepstral averaging. The effect of noise on this method is investigated. The investigation shows that noise may cause principal components to produce estimates which are inferior to cepstral averaging. For these cases an alternate solution is proposed in which principal components are used in the original convolutional space. A wavelet is estimated by homomorphic separation for each data channel. Principal components may then be used to define a best estimate from this suite of estimates.

Table of Contents

List of Tables

List of Figures

Acknowledgement

At this point in a thesis, space is allocated for panegyrization. Therefore, following the course of an untold multitude, I will do so.

My supervisor, Professor Tadeusz Ulrych, has managed to guide me through the labyrinth of academic research to the completion of this thesis. This he has done by sharing freely his wisdom, expertise and philosophies. I am grateful for the opportunities given me, by Tad, to meet many people who make research what is is.

I thank Brad Prager, particularly for sharing his vast knowledge of computers and his programs.

Shlomo Levy, Iain Jones and Kerry Stinson provided me with many stimulating discussions about the topics dealt with in this thesis, as well as myriad others.

The comraderie which accompanies students progressing through the system together is important. Accordingly, and gladly, I thank Julian (Bandito) Cabrera, Don White, Lynda Fisk and Bob Jeffrey. Without them life would have been much duller.

Christian Sutherland has survived my many lucubrations and provided much needed balance and frame. I am deeply grateful.

I thank the University of British Columbia and the Natural Engineering and Research Council for their financial support in the form of fellowships.

I. INTRODUCTION

In any investigation, it is the understanding and insight gained which is paramount.

"Let all things be done unto edifying."[1]

The original purpose of this thesis was to investigate wavelet estimation by combining a non-linear mapping and an optimal linear system. However, it has become primarily an investigation of the mapping, the system and their interaction.

Wavelet estimation can be posed as a multi-channel common information problem. Each channel of data is modeled as the convolution of a wavelet with an impulse sequence. The wavelet remains constant while the impulse sequence varies from channel to channel. A non-linear homomorphic transform can be used to map the data from a convolutional to an additive space. This mapping may also effect partial or total separation of the wavelet and impulses.

In the additive space the problem can be analyzed with linear techniques. We propose the use of principal components to further separate the wavelet and impulses.

In practice this approach yielded some unexpected results. Resolution of these necessitated re-examination of the homomorphic transform and principal components. This led to a clearer understanding of their interaction.

[1] St. Paul, 1 Corinthians 14,26, King James Version of the Holy Bible.

Chapter II of this thesis examines the homomorphic transform: its underlying theory, development, properties and practical realization. Realization includes the calculation of the continous phase of a Fourier transform. This is termed phase unwrapping and is discussed in Chapter III. Phase unwrapping is essentially a numerical analysis problem although it may incorporate diverse mathematical relations. A new approach, of theoretical interest, is examined in detail.

Having mapped from a convolutional to an additive space, averaging can be used for wavelet estimation. An alternate method is provided by principal component analysis. This is developed in Chapter IV in terms of optimal information extraction.

The application of principal components to wavelet estimation is dealt with in Chapter V. For the case of data corrupted by additive noise, principal components may yield poorer estimates than averaging. To avoid this problem, principal components may be used in the original convolutional space. For each channel of data a wavelet estimate may be found by homomorphic separation. Principal components may then be used to define a best estimate from these individual estimates. This procedure is also discussed in Chapter V.

Recent literature has cast doubts on the homomorphic transform's invertibility. Chapter VI briefly addresses this concern.

II. HOMOMORPHIC TRANSFORMS

2.1 Introduction

A problem which arises in signal processing is that of separating signals which have been combined in a known way. The theory of mappings has proven to be useful in its solution. Consider a transformation which maps combined signals into different regions of a space. These regions may be separated and individually mapped back to the original space, yielding the separate input signals. Even if the signals are not separated, various techniques may be more amenable to this separation in the transform space than in the original space. The problem then becomes that of finding an appropriate transformation. We do not know, a priori, that such a transform exists and must consider each case individually. One approach to finding an appropriate transform is provided by the theory of homomorphic systems.

The theory of homomorphic systems was formalized by Oppenheim (1965). He considered classes of nonlinear systems which obey a principle of generalized superposition. These systems may be represented by algebraically linear transformations between input and output vector spaces and are thus called homomorphic systems (Lipschutz, 1974, p. 123). The generalized superposition defines the relationship between a rule for combining signals in the input space and that for combining them in the output space. Linear systems are a special case of homomorphic systems.

2.2 Generalized Superposition

For the purposes of this discussion, all input signals are considered to be discrete and thus may be considered as sequences or as multidimensional vectors. The notation x(n) indicates that x is a function of the discrete variable n.

Consider a system defined by the linear transform T. Let $x_1(n)$ and $x_2(n)$ be input signals and a and b be constant scalars. Then the system T, by definition, satisfies the superposition relation

$$T[a\cdot x_1(n) + b\cdot x_2(n)] = a\cdot T[x_1(n)] + b\cdot T[x_2(n)] \qquad (1)$$

Note that the order of scalar multiplication, signal addition and system transformation of signals is irrelevant. This superposition relation shows the suitability of a linear transform to the separation of signals combined by addition.

It is possible to define more general rules for combining signals and for combining signals with scalars. In this case the corresponding rules for combining the signal transforms will be, in general, different from those used to combine the input signals. Let * represent a rule for combining input signals and : represent a rule for combining a scalar with an input signal. Similarly, let $\underline{+}$ represent a rule for combining output signals and $\underline{\cdot}$ represent a rule for combining output signals with scalars. Then a system H satisfies a generalized principle of superposition if

$$H[a:x_1(n)*b:x_2(n)] = a\underline{\cdot}H[x_1(n)] \underline{+} b\underline{\cdot}H[x_2(n)] \qquad (2)$$

Comparison of equations (1) and (2) shows that linear systems satisfy a generalized superposition with the operations : and $\underline{\cdot}$ as multiplication and the operations * and $\underline{+}$ as addition.

There are a variety of systems which obey a generalized principle of superposition. The mathematical restrictions and the formalism of such systems were developed by Oppenheim (1975) and the details will not be given here.

The class of systems specified by equation (2) can be interpreted as algebraically linear transformations from an input vector space to an output vector space. The rules for combining signals correspond to vector addition and those for combining scalars with signals correspond to scalar multiplication. All systems of this class can be represented as a cascade of three systems known as the canonic representation. In this thesis we will consider that homomorphic system in which * represents discrete convolution and $\underline{+}$ represents addition. The operation : represents a rule for combining scalars with input signals and is best defined through the output operation $\underline{\cdot}$ which represents scalar multiplication. In the special case that the scalar is an integer m, : corresponds to the convolution of the signal with itself m times.

In the interest of readability, let us denote the output operations $\underline{+}$ and $\underline{\cdot}$ by + and $\cdot$ respectively. Then equation (2) becomes

$$H[a{:}x_1(n)*b{:}x_2(n)] = a\cdot H[x_1(n)] + b\cdot H[x_2(n)] \qquad (3)$$

The canonic representation of this homomorphic system is shown in Figure 1.

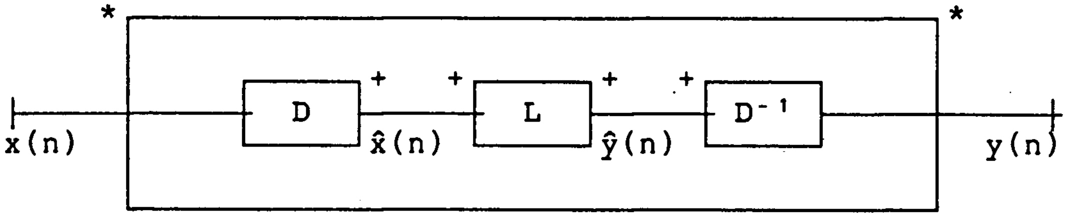


Figure 1 - Canonic Representation of a Homomorphic System
D maps from a convolutional space to an additive space.

D is a homomorphic system which transforms from a convolutional space to an additive space. It is defined by the relationship

$$D[a{:}x_1(n)*b{:}x_2(n)] = a\cdot D[x_1(n)] + b\cdot D[x_2(n)] \qquad (4)$$

L is a linear system, defined by the relation in equation (1). The system D^{-1} is the inverse of D and is defined by the relationship

$$D^{-1}\{a\cdot D[x_1(n)] + b\cdot D[x_2(n)]\} = a:x_1(n)*b:x_2(n) \qquad (5)$$

In general the system D specifies the canonic representation and is called the characteristic system. The canonic representation shows that, once D is fixed, the problem of separating signals which have been combined by convolution is reduced to that of linear filtering. The solution, therefore, lies in the specification of the linear system L.

2.3 The Characteristic System

The characteristic system D serves to transform from a convolutional space to an additive space. It can be decomposed into its constituent transforms. Recall that the z-transform of the convolution of two signals is equal to the product of the z-transforms of the individual signals (Appendix A). That is

$$Z[x_1(n)*x_2(n)] = Z[x_1(n)]\cdot Z[x_2(n)] \qquad (6)$$

It is a homomorphic system which maps from a convolutional to a multiplicative space. The system D has a canonic representation shown in Figure 2 and may be realized in three steps.

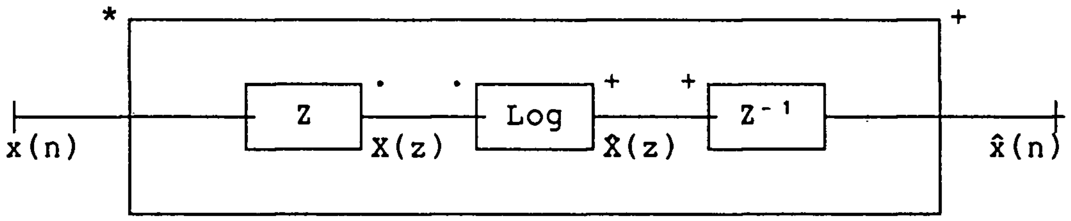


Figure 2 - Representation of the Characteristic System D

First, the z-transform maps an input sequence, x(n), from a convolutional space into a continuous function, X(z), in a multiplicative space. Next, the complex logarithm maps this function from the multiplicative space into another continuous function, $\hat{X}(z)$, in an additive space. Finally , the inverse z-transform maps this continuous function from the additive space into a sequence, $\hat{x}(n)$, in another additive space. This output, $\hat{x}(n)$, is called the complex cepstrum.

The word cepstrum was proposed by Bogert, Healy and Tukey (1963) as a paraphrase of the word spectrum. This, along with fourteen other words, was proposed for use in this additive space because "although strange at first sight, [they] considerably reduce confusion on balance."[1] The cepstrum is defined as the power spectrum of the logarithm of the power spectrum of a signal. Oppenheim, Schafer and

[1] Bogert et al, 1963, "The Quefrency Alanysis of Time Series for Echoes: Cepstrum, Pseudo-Autocovariance, Cross-Cepstrum and Saphe Cracking," in Proc. Symp. on Time Series Analysis (N.Y.: John Wiley and Sons), p. 209.

Stockham (1968) added the word complex to emphasize the use of the complex transforms and the complex logarithm in the system D.

We have implicitly assumed that the complex logarithm maps from a multiplicative space to an additive space. This means that is must be defined such that

$$\log[X_1(z)\cdot X_2(z)] = \log[X_1(z)] + \log[X_2(z)] \qquad (7)$$

We have also assumed that $\hat{X}(z)$ has an inverse z-transform. Thus $\hat{X}(z)$ must be the valid z-transform of some sequence, $\hat{x}(n)$. The unique definition of $\hat{x}(n)$ requires the specification of a region of convergence of $\hat{X}(z)$. We may restrict $x(n)$ and $\hat{x}(n)$ to be real and such that the regions of convergence of $X(z)$ and $\hat{X}(z)$ include the unit circle. That is, $X(z)$ and $\hat{X}(z)$ are analytic in a region including the unit circle. In the case that $X(z)$ or $\hat{X}(z)$ is not analytic on the unit circle, an alternate contour of integration may be used (Appendix A).

Let us denote the evaluation of $\hat{X}(z)$ on the unit circle, $z=\exp(j\omega)$, by $\hat{X}(j\omega)$. In terms of real and imaginary parts

$$\hat{X}(j\omega) = \hat{X}r(j\omega) + j\cdot\hat{X}i(j\omega) \qquad (8)$$

where j is the imaginary unit, $j^2 = -1$. The requirement that $\hat{x}(n)$ be real implies that $\hat{X}r(j\omega)$ is an even function of ω and $\hat{X}i(j\omega)$ is an odd function of ω. It further implies that $\hat{X}(j\omega)$

is periodic in ω with period 2π. Analyticity of $\hat{X}(z)$ on the unit circle implies that $\hat{X}(j\omega)$ must be a continous function of ω. We have

$$\hat{X}(j\omega) = \log[X(j\omega)] \tag{9a}$$

$$\hat{X}(j\omega) = \log|X(j\omega)| + j\cdot\arg[X(j\omega)] \tag{9b}$$

Comparison of equations (9a,b) with equation (8) implies that

$$\hat{X}r(j\omega) = \log|X(j\omega)| \tag{10a}$$

$$\hat{X}i(j\omega) = \arg[X(j\omega)] \tag{10b}$$

Therefore, $\log|X(j\omega)|$ and $\arg[X(j\omega)]$ must be continuous functions of ω. Continuity of $\hat{X}r(j\omega) = \log|X(j\omega)|$ is assured, provided $X(z)$ has no zeros on the unit circle, by the analyticity of $X(j\omega)$ on the unit circle. Note that zeros of $X(z)$ become poles of $\hat{X}(z)$. Continuity of $\hat{X}i(j\omega) = \arg[X(j\omega)]$ depends on the definition of the complex logarithm. Note that $X(j\omega)$, the z-transform evaluated on the unit circle, is just the discrete time Fourier transform.

The essential problem with uniqueness and analyticity of the complex logarithm result from the ambiguity in defining the argument (phase) of a complex function. Any integer multiple of 2π may be added to the principal phase without affecting the phase's being representative of that function.

However, only one continuous phase can be representative of the function. Also, in general, only this continuous phase will satisfy the property of addition specified in equation (7). The procedure of finding this unique phase is commonly termed phase unwrapping. This terminology results from the fact that, in the numerical computation of a phase, the principal value is obtained. Thus the phase is said to have wrapped back into the principal value's range of $(-\pi,\pi)$. Phase unwrapping has been, and continues to be, a major factor in the calculation of complex cepstra. From a theoretical point of view, the continuous phase can be uniquely defined. The definition and computation of the unwrapped phase are dealt with elsewhere in this thesis.

The requirements that $\hat{X}i(j\omega)$ be continuous, odd and periodic implicitly constrain the allowable input sequences $x(n)$. In particular, $x(n)$ must be such that $X(j\omega)$ has a phase of zero at $\omega=\pi$. It must also have a positive mean value. The former follows directly from the above requirements, noting that $\hat{X}i(j\omega)$ is the phase of $X(j\omega)$. The latter follows from the oddness and continuity requirements. These imply that $\hat{X}i(j\omega)$, and hence the phase of $X(j\omega)$, is zero at the origin. Thus the value of $X(j\omega)$ is equal to its magnitude and cannot be negative. The possibility of a zero magnitude is precluded because $X(z)$ has no zeros on the unit circle. Now, the value of $X(j\omega)$ at the origin is equal to the mean value of $x(n)$ which therefore must be positive. It is worth noting that, for a sequence $x(n)$ having a negative mean value, the phase of

$X(j\omega)$ at $\omega=0$ is an odd multiple of π and is discontinuous.

2.4 The Inverse Characteristic System

The inverse characteristic system D^{-1} is defined in equation (5). It transforms from an additive space to a convolutional space and has the canonic representation shown in Figure 3. The transform is realized by cascading a z-transform, complex exponential and inverse z-transform. The complex exponential is unique and, if $\hat{X}(z)$ is analytic on the unit circle, so is $\exp[\hat{X}(z)]$.

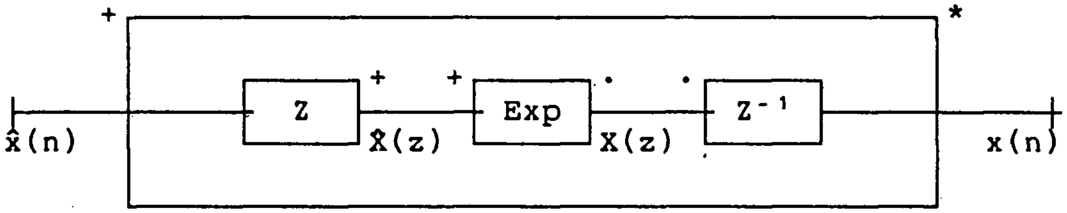


Figure 3 - Representation of the System D^{-1}

The inverse characterstic system D^{-1} maps from an additive space to a convolutional space.

2.5 The Linear System

The linear system L is the only system in H which is unspecified once D is fixed. Its specification will depend on the characteristics of the inputs, the mapping D and the intended outcome of the procedure.

2.6 Properties Of The Complex Cepstrum

We will consider here only finite length input sequences x(n) with positive index n. Such sequences have z-transforms with no singularities in the z plane and can be represented as (Schafer, 1969)

$$X(z) = A\cdot z^{-m_0}\cdot\prod_{k=1}^{m_1}(1-a(k)\cdot z^{-1})\prod_{k=1}^{m_0}(1-b(k)\cdot z) \qquad (11)$$

where $|a(k)|<1$ and $|b(k)|<1$. The a(k) are the m_1 zeros inside the unit circle and the b(k) are the m_0 zeros outside the unit circle. The first factor, A, is a constant and the second factor represents a shift of the input sequence by m_0 positions.

Let us consider the first two factors in equation (11). For real sequences A is real, and if A is positive it contributes only to $\hat{x}(0)$ (Oppenheim and Schafer, 1975). If A is negative, its contribution to $\hat{x}(n)$ is more complicated. It can be shown (Tribolet, 1979) that the sign of A is equal to the sign of the mean value of x(n). Thus, normalizing the input by multiplication by +1 or -1 to make A positive is consistent with satisfying the continuity requirement of $\hat{X}i(j\omega)$ at the origin.

In the second factor, the exponent of z, m_0, gives rise to a term $\hat{x}_1(n)$ in the complex cepstrum of

$$\hat{x}_1(n) = -[m_0 \cdot \cos(\pi n)]/n \qquad (12)$$

(Ulrych, 1971). If X(z) has many zeros outside the unit circle, m_0 can be large and $\hat{x}_1(n)$ may dominate the complex cepstrum. Thus it is important to shift the input sequence to remove this term. This is equivalent to removing the linear phase component of $X(j\omega)$ and is consistent with satisfying the requirements that $\hat{X}i(j\omega)$ be continuous and periodic.

Several properties of the complex cepstrum have been examined (Oppenheim and Schafer, 1975; Tribolet, 1979). Those most relevant are outlined below.

1. The complex cepstrum decays at least as fast as 1/n. That is

$$|\hat{x}(n)| < C \cdot |d^n/n| \qquad -\infty<n<\infty$$

where C is a constant and d is the maximum of |a(k)| and |b(k)|.

2. If x(n) is minimum phase (no zeros outside the unit circle) then

$$\hat{x}(n) = 0 \ , \quad n<0$$

and $\hat{x}(n)$ can be calculated recursively, directly from x(n).

3. If x(n) is maximum phase (no zeros inside the unit circle) then

$$\hat{x}(n) = 0 \ , \quad n>0$$

and $\hat{x}(n)$ can be calculated recursively, directly from $x(n)$.

4. If $x(n)$ is of finite duration, $\hat{x}(n)$ will nevertheless be of infinite duration.

5. The complex cepstrum of a sequence $x(n)$ whose spectrum is smooth tends to concentrate near $n=0$. This property is a result of the fact that a smooth spectrum results from a sequence whose zeros are far from the unit circle. Thus $|a(k)|$ and $|b(k)|$ are small and property 1 implies that $\hat{x}(n)$ decays rapidly with n.

6. Adding an impulse at the origin to a cepstrum is equivalent to scaling the time sequence. Consider a complex cepstrum which is non-zero only at the origin and has magnitude A. Then applying the inverse transform yields a time sequence which is exp[A] at the origin and zero elsewhere. Since addition maps to convolution and this convolution merely scales the sequence by exp[A], the property follows. Note that the scale factor is always positive.

Recall that we have excluded input sequences having zeros on the unit circle. In the case that $x(n)$ has finite duration, $X(z)$ has a region of convergence which includes the unit circle. In the general case the contour of integration of $X(z)$ can be changed by exponentially weighting the input sequence (Appendix A). That is, $x(n)$ becomes

$$w(n) = x(n) \cdot \exp[a \cdot n]$$

where a is a real constant. The sequence w(n) has the z-transform

$$W(z) = X(z\cdot\exp[-a])$$

or

$$W(z) = X(z\cdot b)$$

where b=exp[-a]. Thus, the zeros and poles of X(z) are moved radially by the factor b. If we have a convolutional input, $x(n)=x_1(n)*x_2(n)$, then

$$x(n)\cdot\exp[a\cdot n] = x_1(n)\cdot\exp[a\cdot n]*x_2(n)\cdot\exp[a\cdot n]$$

Thus, exponential weighting of the convolutional of sequences is equivalent to the convolution of exponentially weighted sequences. Exponential weighting can be used to make a mixed phase sequence either minimum phase or maximum phase by moving all its zeros and poles either inside the unit circle or outside the unit circle. Then the special properties of the minimum and maximum phase sequences can be exploited.

2.7 Computational Realization Of The Characteristic System

We have discussed the characteristic system, D, in terms of continuous transforms. In fact, the z-transform, when evaluated on the unit circle, is just the Fourier transform. In practice we cannot compute continous transforms and must use a discrete representation. We use the discrete Fourier transform to evaluate samples of $X(j\omega)$. This computation leads to an aliased cepstrum (Oppenheim and Schafer, 1975). That is, if $\hat{x}(n)$ is the true cepstrum, the calculated cepstrum, $\hat{x}_0(n)$, will be

$$\hat{x}_0(n) = \sum_{k=-\infty}^{\infty} \hat{x}(n+k \cdot N)$$

where N is the number of points in the original sequence x(n). Since, in general, $\hat{x}(n)$ is of infinite duration, $\hat{x}_0(n)$ will be an aliased version of $\hat{x}(n)$. However, as $\hat{x}(n)$ decays exponentially, appending zeros to the input sequence will increase N and reduce the aliasing error.

Schafer (1969) has shown that one may reduce cepstral aliasing by exponentially weighting the input sequence. This has the effect of exponentially weighting the complex cepstrum, causing it to decay more rapidly. Examples of this effect may be found in Stoffa et al.(1974).

The most difficult part, in practice, in calculating the complex cepstrum lies in the computation of the argument or phase of $X(j\omega)$. We must compute samples of the continous phase of $X(j\omega)$ to define $\hat{X}i(j\omega)$ as in equation (10b). If we use the relation $\mathrm{ARG}[X(j\omega)] = \tan^{-1}[Xi(j\omega)/Xr(j\omega)]$, where $Xi(j\omega)$ and $Xr(j\omega)$ represent the imaginary and real parts of $X(j\omega)$, we obtain the principal value of $\arg[X(j\omega)]$. In general, the principal value of the phase is discontinuous and thus will not suffice. Methods of computing samples of the continuous phase are dealt with in the next chapter.

For the special case that the input sequence is either minimum or maximum phase, the complex cepstrum can be calculated, without aliasing, directly from $x(n)$. Oppenheim and Schafer (1975) provide details of the method used.

2.8 Summary

We have discussed a non-linear homomorphic transform which maps convolutions into additions. This transform may effect the separation of convolved signals. Several properties of this transform were presented. There are restrictions on the allowable input signals due to the use of complex logarithms and inverse z-transforms. There are computational problems in computing the complex logarithm which will be dealt with in the next chapter.

III. PHASE UNWRAPPING

3.1 Introduction

The problem of phase unwrapping can be approached from many directions. Let us use a geometrical approach to elucidate the concepts.

Say we have a real sequence, x(n), with Fourier transform $X(\omega)$. $X(\omega)$ is a complex valued function of ω and may be represented by a path in the complex plane. Let $Xr(\omega)$ and $Xi(\omega)$ represent the real and imaginary parts of $X(\omega)$ and let them define the axes of a complex plane as in Figure 4. As ω increases from zero, $X(\omega)$ traces out a path in the complex plane as shown. The principal phase of $X(\omega)$, $\mathrm{ARG}[X(\omega)]$, is defined as the angle a straight line from the origin to $X(\omega)$ makes with the positive real axis. Traditionally this angle is measured positive upward and negative downward. The angles $+\pi$ and $-\pi$ coincide on the negative real axis. If the path of $X(\omega)$ crosses the negative real axis, $\mathrm{ARG}[X(\omega)]$ undergoes a jump discontinuity of 2π. If $X(\omega)$ is moving down, the jump is from $+\pi$ to $-\pi$. If it is moving up, the jump is from $-\pi$ to $+\pi$.

We wish to define a unique phase, $\arg[(X(\omega)]$, which changes continuously everywhere including the negative real axis. This phase will generally be unbounded and will increase or decrease continuously across this axis. $\arg[X(\omega)]$ can be defined by adding to $\mathrm{ARG}[X(\omega)]$ an integer number of multiples of 2π. That is

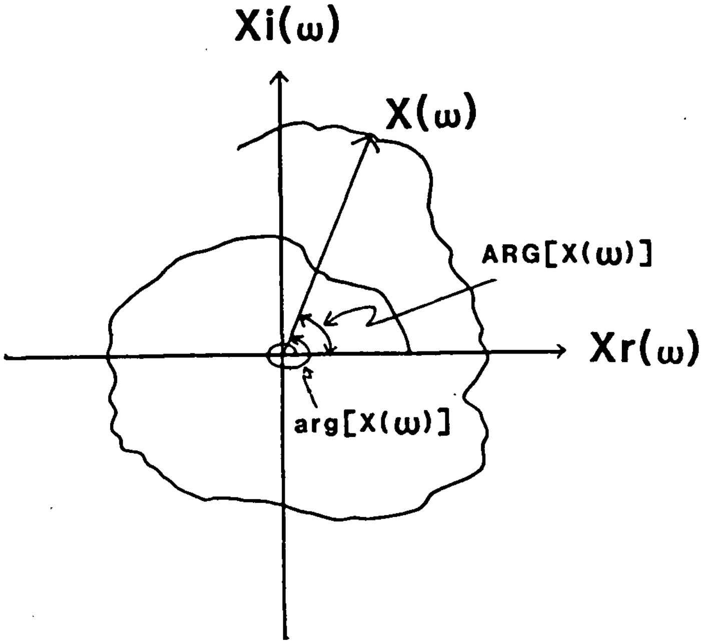


Figure 4 - Complex Function in the Complex Plane

As ω increases, F(ω) traces out a path in the complex plane.

$$\arg[X(\omega)] = \mathrm{ARG}[X(\omega)] + L(\omega)\cdot 2\pi$$

where $L(\omega)$ is an integer which makes arg[·] continuous. $L(\omega)$ can change only at discontinuities in ARG[·]. These occur only when $Xr(\omega)=0$ and $Xi(\omega)$ changes sign. The continuous phase, $\arg[X(\omega)]$, is called the unwrapped phase of $X(\omega)$ and the act of finding $L(\omega)$ is called phase unwrapping. Note that if $X(\omega)=0$ for some ω there can be no unique phase.

This problem is intimately related to the unique definition of the complex logarithm. This follows as $\arg[X(\omega)]$ is the imaginary part of the logarithm of $X(\omega)$. It is also related to the requirement that, if two complex functions are multiplied, their phases will be added. In general

$$\mathrm{ARG}[X_1(\omega)] + \mathrm{ARG}[X_2(\omega)] \neq \mathrm{ARG}[X_1(\omega) + X_2(\omega)]$$

but

$$\arg[X_1(\omega)] + \arg[X_2(\omega)] = \arg[X_1 + X_2(\omega)]$$

Note that if one uses the definition

$$\mathrm{ARG}[X(\omega)] = \tan^{-1}[Xi(\omega)/Xr(\omega)]$$

then ARG[·] is defined on $(-\pi/2,\pi/2)$ and the discontinuities occur at points on the imaginary axis. This follows because the ratio $Xi(\omega)/Xr(\omega)$ does not have the sign information of its constituents and thus $\tan^{-1}[\cdot]$ is projected onto the right half plane. Then $L(\omega)$ can change when $X(\omega)$ crosses the imaginary axis ($Xr(\omega)=0$). An example of the principal value of a phase and the corresponding unwrapped phase are shown in Figure 5.

Various approaches have been put forth to determine the unwrapped phase. Schafer (1969) used an algorithm which searched for discontinuities in the principal phase, ARG(ω), and removed them by addition of $n\cdot 2\pi$ using the appropriate n. A technique to exploit the analytic definition of the phase as the integral of its derivative was developed by Tribolet (1977). Bhanu (1977) used a different technique to exploit

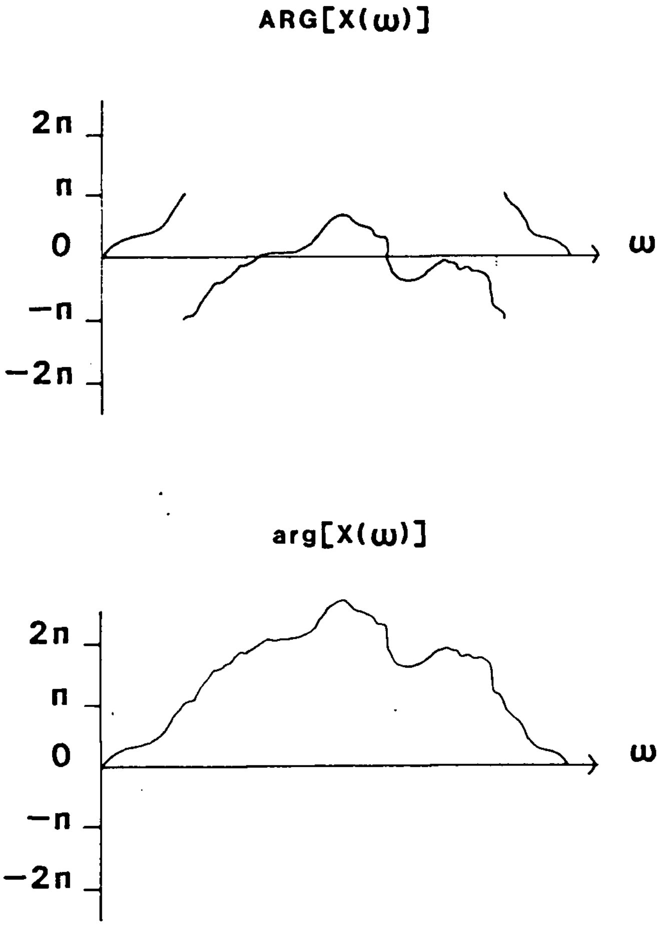


Figure 5 - Principal Value and Unwrapped Phase

The principal value of a phase (ARG) is wrapped into (-π),π) while the unwrapped phase (arg) is continuous.

this same definition. Steiglitz and Dickinson (1982) proposed polynomial factorization as a means of computing the phase. A novel approach, of some theoretical interest, has been developed by McGowan and Kuc (1982) using number theory applied to polynomials.

These methods will be presented briefly. The method of McGowan and Kuc will be discussed in more detail to elucidate the concepts involved. Some discussion of the properties of functions which affect phase unwrapping will be presented.

3.2 Principal Value

The method of phase unwrapping by processing the principal value is due to Schafer (1969). The relation between the principal value, $ARG(\omega)$, and the continuous phase, $arg(\omega)$, is

$$ARG(\omega) = \{arg(\omega)\}MOD2\pi$$

That is, the continuous phase, modulo 2π, is the principal phase. Schafer's algorithm searches for discontinuities in $ARG(\omega)$ caused by the modulo operation and removes them by the appropriate addition or subtraction of 2π. The algorithm starts at $arg(0)$, which is 0 for real $x(n)$, and searches for discontinuities at increasing ω. The user must specify a threshold within which adjacent samples must lie for the phase to be considered continuous. In general, the algorithm becomes more accurate as the spacing between adjacent samples is reduced. Any mis-identified discontinuities will affect the unwrapping at larger ω.

More recently, Poggiagliolmi et al. (1982) proposed another algorithm for processing the principal phase. It also searches for discontinuities, but adaptively recalculates phase values between the original values of the Fourier transform. They suggest shifting the input sequence such that its first moment is zero before calculating the transform. This is an attempt to remove part of the linear phase component and may reduce the number of discontinuities.

3.3 Integration Of The Derivative

Various methods to compute the unwrapped phase use a numerical solution of an analytic expression. Essentially, these methods are attempts to do numerical integration with constraints.

Let x(n) be a sequence and X(ω) be its Fourier transform. Then, as in the previous chapter, we define

$$\hat{X}(\omega) = \log[X(\omega)] \qquad (2a)$$

or

$$\hat{X}(\omega) = \log|X(\omega)| + j\cdot\arg[X(\omega)] \qquad (2b)$$

where the complex logarithm is defined as in the previous chapter and equation (2b) represents a proper Fourier transform. The derivative of $\hat{X}(\omega)$ is well defined and is given by

$$\hat{X}'(\omega) = d\{\log[X(\omega)]\}/d\omega \qquad (3a)$$

$$\hat{X}'(\omega) = \log'|X(\omega)| + j\cdot\arg'[X(\omega)] \tag{3b}$$

$$\hat{X}'(\omega) = X'(\omega)/X(\omega) \tag{3c}$$

where the prime denotes differentiation with respect to ω. Comparison of equations (3b) and (3c) shows that the derivative of $\arg[X(\omega)]$ is equal to the imaginary part of $X'(\omega)/X(\omega)$. It can be shown (Oppenheim and Schafer, 1975) that

$$\arg'[X(\omega)] = \{Xr(\omega)\cdot Xi'(\omega) - Xi(\omega)\cdot Xr'(\omega)\}/|X(\omega)|^2 \tag{4}$$

The phase, $\arg[X(\omega)]$, can be unambiguously defined as

$$\arg[X(\omega)] = \int_0^{\omega} \arg'[X(y)]\, dy \tag{5}$$

with initial condition $\arg[X(0)]=0$. As mentioned in the previous chapter, $\arg[X(\omega)]$ must be continuous and odd. Equation (5) establishes continuity. Oddness will result if the mean phase derivative is zero. That is

$$0 = 1/2\pi\cdot\int_0^{2\pi} \arg'[X(\omega)]\, d\omega \tag{6}$$

If this is not true, removal of the linear phase component will make it so. Implicity, it has been assumed that we have independent knowledge of $\arg'[X(\omega)]$. In fact, this can be computed directly from $x(n)$ using the relation (Tribolet, 1977)

$$Xr'(\omega) + j\cdot Xi'(\omega) = -j\cdot F\{n\cdot x(n)\} \qquad (7)$$

where $F\{\cdot\}$ denotes the Fourier transform. Thus, computing $arg'[X(\omega)]$ by using equations (7) and (4) and integrating as in equation (5) yields the unwrapped phase.

The computational problem is the numerical integration of equation (5). The most straight-forward solution is to use a trapezoidal rule for integration. Again, this method improves as the spacing between adjacent samples of $arg'[X(\omega)]$ decreases. This approach also suffers from error propagation in that an error in integration at $\omega=\omega_1$ will affect the result at $\omega>\omega_1$.

A more sophisticated method of carrying out the integration was developed by Tribolet (1977) and is called adaptive integration. The principal phase $ARG(\omega)$ as well as the phase derivative, $arg'(\omega)$ are calculated. The phase derivative is then integrated as in (5) using a trapezoidal rule between, say, ω_1 and ω_2. This computed value of the phase is then compared to the principal value. The constraint that these values must be close to each other to within an integer multiple of 2π is added. That is

$$|\{ARG[X(\omega)]+2\pi\cdot L(\omega)\} - arg[(X(\omega)]| < E$$

for some integer $L(\omega)$, where E is some small positive number supplied by the user. The algorithm checks values of $L(\omega)$ to satisfy the inequality. If no $L(\omega)$ suffices, another point of the phase derivative is calculated midway between ω_1 and ω_2. The trapezoidal rule is applied between ω_1 and $(\omega_2-\omega_1)/2$ and the procedure iterated until the inequality is satisfied. The procedure is initiated by taking a DFT of $x(n)$ and $n \cdot x(n)$ to obtain $ARG[X(\omega)]$ and $arg'[X(\omega)]$. Then the integration starts at $\omega=0$ and proceeds to higher ω, recalculating points as needed. This method also suffers from error propagation. An error in determining $L(\omega_1)$ will carry through for $\omega>\omega_1$.

Bhanu (1977) investigated Tribolet's algorithm and compared it to alternate techniques. He also proposed and investigated the use of piecewise polynomial interpolation of the phase derivative as a better rule of integration. This requires the evaluation of the second derivative of the phase, computed from $F\{n^2 \cdot x(n)\}$ as well as the first derivative and the principal value. There is a trade-off of more initial computation for increased accuracy in the integration.

Note that, although the principal value of the arctangent is contained in $(-\pi/2,\pi/2)$, by retaining the sign information of the terms in $\tan^{-1}[Xi(\omega)/Xr(\omega)]$ a value unique within $(-\pi,\pi)$ can be assigned.

3.4 Factorization

Steiglitz and Dickinson (1982) proposed the use of polynomial factorization of the z-transform as a "reliable and acurate"[1] method of phase computation. Once the roots of a z-transform are known, it can be factored into the product of polynomials of degree one (dipoles). These dipoles may then be evaluated on the unit circle to form the Fourier transform. The phase of the product is equal to the sum of the phases of each of the dipoles. The continuous phase of each dipole, and thus the product, can be calculated unambiguously. With this method, the problem becomes that of finding roots of a complex polynomial. Steiglitz and Dickinson used a Newton-Raphson root-finding algorithm. They ignored the problem of repeated roots and saddle points on the grounds that these rarely occur in practice.

3.5 Number Theory

McGowan and Kuc (1982) proposed a method of defining the continuous phase based on number theory. The theory will be discussed in detail and relevant proofs presented.

In linear system control theory the problem of stability is important. There are methods which will determine the stability of a system without actually finding which part of the system is unstable. A particular method of determining

[1] Steiglitz and Dickinson, 1982, "Phase Unwrapping by Factorization," I.E.E.E. Trans. A.S.S.P., v. ASSP-30, No. 6, p. 984.

stability gives information about the number of roots of a polynomial which are inside or outside the unit circle without actually locating them. This same method may be used to determine the continuous phase.

Consider a real sequence x(n), n=0,1,...N-1 with the discrete Fourier transform (DFT)

$$X(\omega) = \sum_{n=0}^{N-1} x(n)\cdot\exp[-j\cdot n\omega] \qquad (8)$$

The phase can then be written as

$$\arg[X(\omega)] = \tan^{-1}[Xi(\omega)/Xr(\omega)] + L(\omega)\cdot\pi$$

for $0<\omega<\pi$. $\tan^{-1}[\cdot]$ is the principal value arctangent, that is $-\pi/2<\tan^{-1}<\pi/2$, and $L(\omega)$ is an integer function of ω which makes arg[·] a continuous function. Finding $L(\omega)$ is the phase unwrapping problem. Consider the arctangent function in Figure 6. Consider a point on the principal branch, L=0, defined by $F(\omega) = Xi(\omega)/Xr(\omega)$ (and assuming $Xi(\omega)$ to have no singularities). This point can change branches to L=1 only if $F(\omega)$ goes to $+\infty$. It can then 'wrap' onto the L=1 branch at $-\infty$ without exhibiting any discontinuity. Likewise, if $F(\omega)$ goes to $-\infty$, this point can change to the L=-1 branch. The arctangent can change branches only at singularities of $F(\omega)$, or equivalently, at zeros of $Xr(\omega)$. However, a branch change is not required at these singularities. $F(\omega)$ may go to infinity and return on the same

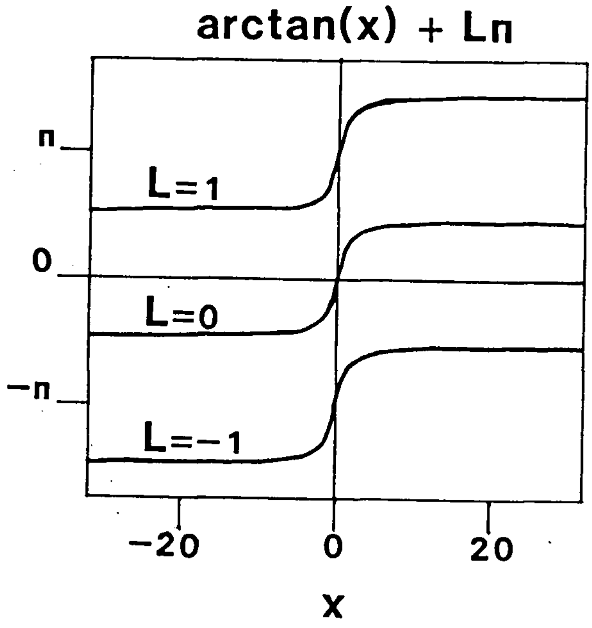


Figure 6 - Branches of arctangent Function

The function can continuously change branches at infinities.

branch. Thus, whether or not $F(\omega)$ changes sign and the direction of the sign change provides the information necessary to determine $L(\omega)$.

Let us assume that $X(\omega)\neq 0$ on $0\leq\omega\leq\pi$. This is the constraint that there are no zeros on the unit circle in the z-plane. The product $Xi(\omega)\cdot Xr(\omega)$ has the same sign as the ratio $Xi(\omega)/Xr(\omega)$ but has no singularities. Consider the sign of $Xi(\omega)\cdot Xr(\omega)$ as ω increases through a zero of $Xr(\omega)$. (Note that $Xi(\omega)$ doesn't change sign for ω close enough to the zero of $Xr(\omega)$.) If the product's sign changes from positive to negative, $L(\omega)$ is increased by one. If an opposite sign change occurs. $L(\omega)$ is decreased by one. If there is no sign change $L(\omega)$ is not changed. $L(\omega_1)$ is generated, in principle, by letting ω increase from $\omega=0$ to $\omega=\omega_1$ and incrementing or decrementing $L(\omega)$ as required. The condition arg[0]=0 provides the starting value, $L(0)=0$. Now the problem of phase unwrapping has become that of finding the sign changes of $Xi(\omega)\cdot Xr(\omega)$ through the zeros of $Xr(\omega)$. A solution to this problem can be obtained using Sturm's theorem (Beaumont and Pierce, 1963). Sturm's theorem arises in number theory and provides a method of finding the number of distinct real roots of a polynomial between two arguments. The method involves the generation of a Sturm sequence, a sequence of polynomials of decreasing degree, generated recursively from two given polynomials (Marden, 1966). McGowan and Kuc use the real and imaginary parts of the DFT to start the sequence. They found it convenient to express the Sturm sequence in terms of Chebyshev polynomials. The Sturm sequence provides sufficient information to determine $L(\omega)$ and hence the unwrapped phase.

The method proceeds as follows. The real and imaginary parts of $X(\omega)$ are expressed in terms of Chebyshev polynomials of the second kind. Chebyshev polynomials of the first kind, $T(n)$, and of the second kind, $U(n)$, as functions of $\cos(\omega)$ are defined as (Snyder, 1966, p.11-24)

$$T(n) = T(n,\cos\omega) = \cos(n\cdot\omega) \qquad n\geq 0 \qquad (9)$$

$$U(n) = U(n,\cos\omega) = \sin[(n+1)\omega]/\sin(\omega) \qquad n\geq 0 \qquad (10)$$

Relevant recursion relations between these polynomials are, for $n \geq 2$,

$$T(n) = [U(n) - U(n-2)] \tag{11a}$$

$$T(1) = U(1) \tag{11b}$$

$$T(0) = U(0) = 1 \tag{11c}$$

For $n \geq 1$ there is the recursion relation

$$U(n+1) = U(n) \cdot U(1) - U(n-1) \tag{11d}$$

where it is understood that U(n) and T(n) are functions of cos(ω). (Note that Chebyshev polynomials are an orthogonal set (Snyder, 1966)). From the identities (9) and (10), X(ω) in (8) can be expressed as

$$X(\omega) = \sum_{n=0}^{N-1} a(n) \cdot U(n) + j \cdot \sin(\omega) \sum_{n=0}^{N-2} b(n) \cdot U(n) \tag{12}$$

where the a(n) are linear combinations of x(n) and b(n) = -x(n). Details on calculating a(n) and b(n) are given in Appendix B. Equation (12) may be written as

$$X(\omega) = Xr(\omega) + j \cdot Xi(\omega) \tag{13a}$$

$$X(\omega) = P(0,\omega) + j \cdot \sin(\omega) \cdot P(1,\omega) \tag{13b}$$

$$X(\omega) = P(0) + j\cdot\sin(\omega)\cdot P(1) \qquad (13c)$$

where it is understood that P(0) and P(1) represent polynomials which are functions of ω. That is

$$P(0) = P(0,\omega) = Xr(\omega) = \sum_{n=0}^{N-1} a(n)\cdot U(n,\cos\omega) \qquad (14a)$$

and

$$P(1) = P(1,\omega) = Xi(\omega)/\sin\omega = \sum_{n=0}^{N-2} b(n)\cdot U(n,\cos\omega) \qquad (14b)$$

Note that the sign of P(1) is the same as that of $Xi(\omega)$ on $0<\omega<\pi$.

P(0) and P(1) are polynomials in U(n). They are used to generate a Sturm sequence, denoted by $\{P(0),P(1),\ldots,P(M)\}$ where $M\leq N-1$. This Sturm sequence is a sequence of polynomials of decreasing degree. Each polynomial is a function of ω. The sequence is generated using the 'negative remainder' relationship (Marden, 1966). In fact, this relationship is merely Euclid's algorithm for finding the greatest common divisor of integers or polynomials (Beaumont and Pierce, 1963). The relationship is

$$P(k-1) = Q(k)\cdot P(k) - P(k+1) \qquad (15)$$

where $1\leq k\leq M-1$. P(k) and Q(k) are polynomials of degree k. The degree of the polynomials P and Q is defined to be that of the highest degree Chebyshev polynomial, U, in their sum. If a P contains a term U(K) where K is the highest degree in the sum, P is of degree K.

The recursion (15) proceeds, given P(k-1) and P(k), by finding Q(k) and P(k+1) such that

$$\deg\{Q(k)\} = \deg\{P(k-1)\} - \deg\{P(k)\} > 0 \qquad (16)$$

and

$$\deg\{P(k+1)\} < \deg\{P(k)\}$$

The existence of Q(k) and P(k+1) are guaranteed (Beaumont and Pierce, 1963) and, in fact, this algorithm is equivalent to the long division of polynomials. We choose Q(k) such that (16) is satisfied and such that the highest degree term of P(k-1) is identical to that of Q(k)•P(k). It may happen that other terms of lesser degree are also identical. This results in the elimination of these terms. That is, writing (15) as

$$P(k+1) = Q(k) \cdot P(k) - P(k-1)$$

then P(k+1) will be of degree less than P(k-1). For example, if P(0) is degree N and P(1) is of degree N-R, then we take Q(1) of degree N-(N-R) = R so that Q(1)•P(1) is of degree N and

$$P(2) = Q(1)\cdot P(1) - P(0)$$

The highest degree term in P(0) is cancelled by the highest degree term in $Q(1)\cdot P(1)$. (Possibly some other terms will cancel, but this is not guaranteed.) The algorithm continues, discarding Q, by increasing k until $k=M-1$ where $P(M+1)=0$. Then $P(M-1) = Q(M)\cdot P(M)$ and P(M) is the greatest common divisor of P(0) and P(1). If $M=N-1$, $P(N-1) = P(M)$ is a constant. Details of the recursion are given in Appendix C.

Let us define the operator $V(\omega)$ which, when applied to the Sturm sequence, gives the number of sign changes in the sequence for a fixed ω. To calculate $V(\omega)$, fix an ω and count sign changes from adjacent P(n) in $\{P(0),P(1),...,P(M)\}$. If adjacent P(n) have opposite signs $V(\omega)$ is incremented by one, otherwise it is unchanged. The number of sign changes in the Sturm sequence, as a function of ω, will determine $L(\omega)$, the function required to unwrap the phase. Specifically, $L(\omega) = V(\omega)$. This follows from the properties of the Sturm sequence and is proved below.

We require two theorems to lead up to the final result. These are presented here with proofs.

Theorem: If $P(0,\omega)$ and $P(1,\omega)$ have no common zeros for $0<\omega<\pi$, then $\text{sign}[P(M,\omega)]$, where P(M) is the greatest common divisor of P(0) and P(1), cannot change for $0<\omega<\pi$.

Proof: P(M) was generated by Euclid's algorithm:

$$P(0) = Q(1)\cdot P(1) - P(2)$$

$$P(1) = Q(2) \cdot P(2) - P(3)$$

$$\vdots$$

$$P(M-2) = Q(M-1) \cdot P(M-1) - P(M)$$

$$P(M-1) = Q(M) \cdot P(M)$$

If $P(M)=0$ for some ω, then this recursion implies, by induction, $P(M-1)=0$, $P(M-2)=0$, ..., $P(1)=0$, $P(0)=0$. But, by hypothesis, $P(1)$ and $P(2)$ have no common zeros on $0<\omega<\pi$. Therefore $P(M)\neq 0$ for any ω on $0<\omega<\pi$ and cannot change sign. Q.E.D.

Theorem: If $P(0,\omega)$ and $P(1,\omega)$ have no common zeros, then $V\{P(k,\omega)\}$ does not change, as a function of ω, except at a zero of $P(0,\omega)$.

Proof: Say $V\{P(k,\omega)\}$ changes. This can only happen at a zero of some $P(k,\omega)$, say $\omega=\omega_0$. That is, $P(k,\omega_0)=0$. Euclid's algorithm then implies (equation (5))

$$P(k-1,\omega_0) = -P(k+1,\omega_0)$$

or

$$P(k-1,\omega_0) \cdot P(k+1,\omega_0) < 0$$

That is, members of the Sturm sequence separated by one other member have opposite signs. (This must be true because otherwise $P(k-1,\omega_0) = P(k+1,\omega_0) = 0$ which would imply $P(J,\omega_0)=0$ for every $J>k$ including $J=M$ which cannot be by the previous theorem.) Therefore $V\{P(k,\omega)\}$ cannot change through a zero of $P(k,\omega)$, $0<k<M$.

Q.E.D.

This theorem can be illustrated by considering the signs of $\{P(k-1),P(k),P(k+1)\}$ near $\omega \doteq \omega_0$. Say they are $\{-,+,+\}$. Then if $P(k)$ changes sign, from + to -, the sequence becomes $\{-,-,+\}$ and the number of sign changes is the same (i.e. 1). The other possible cases are

$$\{-,-,+\} \text{ becoming } \{-,+,+\}$$
$$\{+,-,-\} \text{ becoming } \{+,+,-\}$$
$$\{+,+,-\} \text{ becoming } \{+,-,-\}$$

By inspection we see that the number of sign changes is unchanged. If more than one of the $P(n)$ are zero the above applies to each $P(n)$ individually. Note that $P(k)$ and $P(k+1)$ cannot both be zero for some ω_0 as the division algorithm would imply $P(0)$ and $P(1)$ are also both zero, which is contrary to our hypothesis. Note that when $P(k)$ is actually zero, sign changes cannot be counted by comparing the signs of adjacent terms as zero has no sign. However, we can ignore the 0 and compare the signs of the terms on either side (Beaumont and Pierce, 1963). This follows directly from Euclid's algorithm.

Now let us consider the first term of the sequence, $P(0,\omega)$, as ω passes through a zero, ω_0, of $P(0,\omega)$. That is $P(0,\omega_0)=0$. Since $P(1,\omega_0)\neq 0$ by hypothesis, three situations of interest can occur. First, $P(0)\cdot P(1)$ does not change sign, and $V(\omega)$ does not change. Second, $P(0)\cdot P(1)$ changes from - to

+ and $V(\omega)$ decreases by one. That is, $P(0)$ and $P(1)$ change from having opposite signs to having the same sign. Third, $P(0)\cdot P(1)$ changes from + to - and $V(\omega)$ increases by one. That is, $P(0)$ and $P(1)$ change from having the same sign to having opposite signs. By the previous theorem these are the only conditions under which $V(\omega)$ can change. Therefore, $V(\omega_0)$ is equal to the number of times $P(0)\cdot P(1)$ changes from + to -, minus the number of times it changes from - to +, as ω increases from 0 to ω_0.

Because $P(0,\omega) = Xr(\omega)$ and $P(1,\omega) = Xi(\omega)/\sin(\omega)$ it follows that $V(\omega)$ contains the branch number of the arctangent of $Xi(\omega)/Xr(\omega)$. In fact, $L(\omega) = V(\omega)$. $V(\omega)$ may also be used to keep track of the relative number of branch changes in the phase between two points, say ω_1 and ω_2, as ω goes from ω_1 to ω_2.

Because the relation between $X(\omega)$ and $V(\omega)$ is direct, calculating $V(\omega)$ does not involve sampling $X(\omega)$. We compute the Sturm sequence which is a purely algebraic recursion (Appendix C). Then we evaluate $P(n,\omega)$ for all n from 0 to M at those ω for which we want the phase. We then calculate the number of sign changes in the P at each ω, yielding $V(\omega)$. We can choose as many or as few ω as we like. The method is not recursive and does not depend on previous calculations as did the integration methods.

3.5.1 Computational Details

McGowan and Kuc (1982) present a computer program for carrying out the evaluation of $L(\omega)$ using this method. In general this algorithm will be computationally slow when compared to a single Fast Fourier Transform (FFT) (Brigham, 1974, p 177). This results from the computation of many trigonometric functions and multiplications in the evaluation of the Sturm sequence. Say we have an input sequence of length N. Then we calculate a Sturm sequence of, at most, N polynomials of lengths N, N-1, N-2,..., 1. Thus we evaluate $N + (N-1) + (N-2) + \ldots + 1$ or about $N^2/2$ coefficients. Each coefficient is multiplied by a trigonometric function. Thus, to evaluate the sequence of polynomials at one value of ω we must carry out about $N^2/2$ trigonometric function evaluations and multiplications. For B values of ω we need $B \cdot N^2/2$ such operations. If $B = N$ we need $N^3/2$ such operations. This compares with about $N \cdot \log_2 N$ such operations for a single (power of 2) FFT. In fact, we may use the FFT to evaluate these polynomials and increase efficiency. Consider the polynomial

$$P(k,\omega) = a(0) \cdot U(0,\omega) + a(1) \cdot U(1,\omega) + \ldots$$

where $U(n,\omega) = \sin[(n+1)\omega]/\sin(\omega)$. Bringing $\sin(\omega)$ out of the sum yields

$$\sin(\omega) \cdot P(k,\omega) = a(0) \cdot \sin[\omega] + a(1) \cdot \sin[2\omega] + \ldots$$

which can be written as

$$\sin(\omega)\cdot P(k,\omega) = 0\cdot\sin[0\omega] + a(0)\cdot\sin[\omega] + \ldots$$

which is in the form of a Fourier transform. Using the FFT will reduce the number of operations. For example, if N is a power of 2, these operations will number about $N^2\log_2 N$.

3.6 Difficulties Of Phase Unwrapping

We will consider only the problem of accurate data. That is, we want the phase of the input sequence x(n), where we know x(n) accurately. As we have seen, the unwrapped phase can be defined analytically. However, finite numerical evaluation of the analytic expressions introduces computational errors. These may lead to an incorrect result. The question naturally arises as to which types of sequence are more susceptible to these errors. Generally speaking, when the zeros of X(z) are far from the unit circle, the phase will be well determined. It will vary slowly and any discontinuities will be easily detected. As a zero comes closer to the unit circle, the phase develops a rapid change near that zero. If the zero is on the unit circle the amplitude becomes zero and the phase is undefined. The difference in the phase curve if a zero is just inside (minimum phase) or just outside (maximum phase) the unit circle can be large (Bhanu, 1977; Clayton and Wiggins, 1976; Poggiagliomli et al., 1982). Thus, if computations shift a zero just across the unit circle, serious inaccuracies could result. For zeros near the unit circle the problem of phase determination is ill-conditioned. This is because the ratio $Xi(\omega)/Xr(\omega)$ used in the arctangent becomes poorly determined. If both $Xi(\omega)$ and $Xr(\omega)$ are small, small errors in the evaluation of $Xr(\omega)$ can cause large changes in the ratio.

The question becomes that of which types of sequence have zeros close to the unit circle. A type of sequence which will be relevant in this thesis consists of a random impulse train. Steiglitz and Dickinson (1982) discuss such a sequence. If the sequence values are independent random variables then, as the sequence length increases, its zeros tend to become evenly distributed in angle and tightly clustered near the unit circle. Also, the probability increases that there will be a zero so close to the unit circle that a given algorithm will not be able to place it on the correct side of the circle. If this sequence is convolved with a short sequence, a finite number of extra zeros are introduced and the above result still holds.

The question of how close a zero must be to the unit circle to be incorrectly identified was explored by Bhanu (1977). For simple short sequences, zeros with a magnitude of 1.00001 were correctly identified although magnitudes between about 0.9 and 1.1 are typically considered close to the unit circle.

3.7 Comparison Of Phase Unwrapping Techniques

The comparison of phase unwrapping techniques is difficult. On well conditioned data they will all work well. On poorly conditioned data, their accuracy will depend on the individual data set. The interaction between the zeros of even a short sequence is complicated and not amenable to direct analysis. Thus, the comparison of various techniques must be empirical. Computational time requirements of some techniques has been evaluated empirically or analytically. The computational time of McGowan and Kuc's technique does not depend of the input data, only on its length. Using a power of 2 FFT to evaluate Sturm sequences, run time is of the order of $N^2 \log_2 N$. That is $T = O(N^2 \log_2 N)$. Steiglitz and Dickinson found that their algorithm typically ran in $T = O(N^2)$. The algorithms of Tribolet, Bhanu and Poggiagliolmi et al. initially run in $T = O(N \cdot \log_2 N)$ before adaptation. Adaptation time depends on the data and the program parameters.

McGowan and Kuc's algorithm was implemented using a FORTRAN program. Tribolet's algorithm was published as a FORTRAN program (Tribolet, 1977). On the data used for this thesis the programs yielded identical results except for computation time. Tribolet's algorithm almost always took less time. For example, for an input sequence of 90 points padded with zeros to 256 points, McGowan and Kuc's algorithm took 0.048 ms while Tribolet's algorithm had an average run time of about 0.02 ms on an Amdahl 470 V/8 computer.

3.8 Summary

Phase unwrapping can be approached from many directions. Iterative methods include search algorithm, adaptive integration and polynomial factorization. Number theory provides a direct relation between a time sequence and its phase. In principle, the phase determination is unstable for zeros near the unit circle.

IV. PRINCIPAL COMPONENT ANALYSIS

4.1 Introduction

We have examined theoretical aspects of the homomorphic transform as applied to convolutional problems. The transform changes convolutional problems to additive ones. Techniques developed for additive problems may then be applied. Using this transform, wavelet estimation can be stated as a multi-channel common information problem. The formalism is presented in the next chapter. This chapter shows how this problem can be approached in the general case. It thus provides a solution in anticipation of its application. The reader's forbearance is requested.

Suppose we have a suite of sequences, each sequence a function of n. Write $\{x_1(n), x_2(n), \ldots\}$. Suppose further that each of these sequences contains some common signal in addition to a component which is, in general, different in each sequence. We seek a way to extract the common signal from the suite in an optimal fashion. We may measure the amount of common signal between sequences by their covariance. By seeking a linear combination of the sequences which optimizes the common signal we are led to a technique known as principal component analysis (Dhrymes, 1970, p. 53). In fact, this technique can be shown to be equivalent to the Karhunen-Loève, Hotelling, or eigenvector transforms (Hall, 1979, p. 115). Because there exist a variety of ways to derive this result, that which lends the most insight to the problem dealt with in this thesis will be derived.

4.2 Principal Components

Let us begin by defining several terms. The terminology is reminiscent of the statistical literature, reflecting the origins of this development, although the definitions will be for deterministic data. Say we have a suite of M sequences, each sequence a function of n and of length N. In vector form we write $\{x_1(n), x_2(n), \ldots x_M(n)\} = \underline{x}'(n)$ where underscore denotes a vector and prime denotes the transpose of a vector or matrix. We define the variance of an element of $\underline{x}(n)$, say $x_i(n)$, by

$$\mathrm{var}(x_i) = 1/N \sum_{n=1}^{N} [x_i(n) - m_i]^2 \qquad (1)$$

where m_i is the mean of $x_i(n)$, defined by

$$m_i = 1/N \sum_{n=1}^{N} x\ (n) \qquad (2)$$

We define the covariance of two sequences $x_i(n)$, $x_j(n)$ as

$$\mathrm{cov}(x_i, x_j) = 1/N \sum_{n=1}^{N} [x_i(n) - m_i][x_j(n) - m_j] \qquad (3)$$

Equation (3) can be extended to define the covariance matrix C, of $\underline{x}(n)$, whose elements are given by $C_{ij} = \mathrm{cov}(x_i, x_j)$. That is

$$C = 1/N \sum_{n=1}^{N} [\underline{x}(n)-\underline{m}][\underline{x}(n)-\underline{m}]' \qquad (4)$$

where $\underline{m}$ is the vector of means $\underline{m}' = \{m_1, m_2, \ldots m_M\}$ and $[\cdot][\cdot]'$ represents the outer product of vectors (Gelb, 1974). C is symmetric and can be shown to be positive semidefinite (Dhrymes, 1970, p 3). For compactness of notation let us define the operator $E\{\cdot\}$ to be the sum $1/N\sum_{n=1}^{N}\{\cdot\}$. (Note that this is not the expectation operator in probability theory, as we are not dealing with random variables.) Then equations (1),(2), and (3) become

$$\mathrm{var}(x_i) = E\{[x_i(n)-m_i]^2\} \qquad (5a)$$

$$m_i = E\{x_i(n)\} \qquad (5b)$$

$$C = E\{[\underline{x}(n)-\underline{m}][\underline{x}(n)-\underline{m}]'] \qquad (5c)$$

We can consider the covariance of two sequences to be a measure of the signal common to both, where that signal is defined in terms of fluctuations about a mean.

Let us attempt to maximize the common signal of the suite of sequences by taking a linear combination of the $\underline{x}(n)$, say $y(n) = \underline{a}'\underline{x}(n)$, where $\underline{a}' = \{a_1, a_2, \ldots\}$ is a vector of constant coefficients. Our measure of the signal in y(n) is its variance. Thus we attemt to find that $\underline{a}$ which maximizes the variance of y(n). If $y(n) = \underline{a}'\underline{x}(n)$ then, from equations (5)

$$\mathrm{var}(y) = \underline{a}'C\underline{a} \qquad (6)$$

(Dhrymes, 1970) where C is the covariance matrix of $\underline{x}(n)$. However, we see that there is no maximum of $\underline{a}'C\underline{a}$. If $\underline{a}$ is a solution, it could be multiplied by any constant scalar s such that $\underline{b} = s\underline{a}$ and, with $\underline{a}$ fixed, s could be increased to make var(y) arbitrarily large. We have a problem of scale. It is the structure of $\underline{a}$ which is important, not its magnitude. Thus we fix the scale by arbitrarily requiring $\underline{a}$ to be of unit length, that is $\underline{a}'\underline{a} = 1$.

Now the problem is to maximize $\mathrm{var}(y) = \underline{a}'C\underline{a}$ subject to the constraint $\underline{a}'\underline{a} = 1$. This can be done with the method of Lagrange multipliers. We define the Lagrangian

$$L = \underline{a}'C\underline{a} + \lambda(1-\underline{a}'\underline{a}) \qquad (7)$$

where λ is a Lagrange multiplier. L is maximized with respect to $\underline{a}$ and λ by setting its partial derivatives equal to zero. This yields

$$dL/d\underline{a} = \underline{0} = 2C\underline{a} - 2\lambda\underline{a} \qquad (8a)$$

$$dL/d\lambda = 0 = 1 - \underline{a}'\underline{a} \qquad (8b)$$

where we have used the rules of vector differentiation (Gelb, 1974; Hall, 1970) and $\underline{0}$ denotes the zero vector. Premultiplying equation (8a) by $\underline{a}'$ yields

$$\underline{a}'C\underline{a} = \lambda \tag{9}$$

which, from (6), equates the variance of y with the Lagrange multiplier λ.

Equation (8a) implies

$$C\underline{a} = \lambda\underline{a} \tag{10}$$

which is the eigenvalue problem (Strang, 1980, p. 179-239) with eigenvalues λ and eigenvectors $\underline{a}$.

Because we want the variance of y to be maximal, and this is equal to λ, we must choose the largest λ which satisfies equation (10), say λ_1, and its associated eigenvector, say $\underline{a}_1$. Thus, the solution, denoted by $y_1(n)$, is

$$y_1(n) = \underline{a}_1'\underline{x}(n) \tag{11}$$

This is defined to be the first principal component of $\underline{x}(n)$ (Dhrymes, 1970, p. 34). The mean and variance of y_1 are easily derived. The mean, my_1, is

$$my_1 = E\{y_1(n)\}$$
$$my_1 = E\{\underline{a}_1'\underline{x}(n)\}$$
$$my_1 = \underline{a}_1'E\{\underline{x}(n)\}$$

or finally

$$my_1 = \underline{a}_1'\underline{m} \tag{12}$$

Thus the mean of the first principal component is the weighted sum of the means of $\underline{x}(n)$. The variance of y_1 is, from (6) and (9),

$$\mathrm{var}(y_1) = \lambda_1 \tag{13}$$

Thus the variance of the first principal component is equal to the largest eigenvalue of equation (10).

We expect the first principal component to contain the most common signal of the suite. We may then investigate other linear combinations of the sequences to see what properties they have. Accordingly, we seek the normalized linear combination of $\underline{x}(n)$, say $\underline{y}(n) = \underline{a}'\underline{x}(n)$, which has maximal variance but is uncorrelated with the first principal component $y_1(n)$. By uncorrelated, we mean $\mathrm{cov}(y_1,y) = 0$. We can rewrite this covariance as (using equation (11))

$$\begin{aligned}
\mathrm{cov}(y_1,y) &= E\{[y_1(n)-my_1][y(n)-my]'\} \\
\mathrm{cov}(y_1,y) &= E\{[\underline{a}_1'(\underline{x}(n)-\underline{m})][\underline{x}(n)-\underline{m}]'\underline{a}\} \\
\mathrm{cov}(y_1,y) &= \underline{a}_1'E\{[\underline{x}(n)-\underline{m}][\underline{x}(n)-\underline{m}]'\}\underline{a} \\
\mathrm{cov}(y_1,y) &= \underline{a}_1'C\underline{a}
\end{aligned}$$

so that the constraint may be written as

$$\underline{a}_1'C\underline{a} = 0 \tag{14}$$

Proceeding as before we define the Lagrangian

$$L = \underline{a}'C\underline{a} + \lambda(1-\underline{a}'\underline{a}) + u\underline{a}_1'C\underline{a}$$

where λ and u are Lagrange multipliers. Setting partial derivatives to zero yields

$$dL/d\underline{a} = \underline{0} = 2C\underline{a} - 2\lambda\underline{a} + uC\underline{a}_1 \qquad (15a)$$

$$dL/d\lambda = 0 = 1 - \underline{a}'\underline{a} \qquad (15b)$$

$$dL/du = 0 = \underline{a}_1'C\underline{a} \qquad (15c)$$

Premultiplying equation (15a) by $\underline{a}'$ and using equations (15b) and (15c) yields

$$\underline{a}'C\underline{a} = \lambda$$

This is just the same result derived earlier: the variance of y is equal to the Lagrange multiplier λ. From the previous derivation we have

$$C\underline{a}_1' = \lambda_1\underline{a}_1$$

Premultiplying by $\underline{a}'$ and using equation (15c) yields

$$\underline{a}'C\underline{a}_1 = \lambda_1\underline{a}'\underline{a}_1$$

or

$$0 = \underline{a}'\underline{a}_1 \tag{16}$$

which is an expression requiring the orthogonality of the eigenvectors. Now, premultiplying equation (15a) by $\underline{a}'$ yields

$$2\underline{a}_1'C\underline{a} - 2\lambda\underline{a}_1'\underline{a} + u\underline{a}_1'C\underline{a}_1 = 0$$

However, from equations (15c) and (16), the first two terms are zero. Hence

$$u\underline{a}_1'C\underline{a}_1 = 0$$

and from equation (13)

$$u\lambda_1 = 0$$

Now, in general, $\lambda_1 > 0$ which implies $u = 0$ and equation (15a) becomes

$$C\underline{a} = \lambda\underline{a}$$

and we have returned to the eigenvalue problem. However, now we must choose the second largest eigenvalue, say λ_2, and its associated normalized eigenvector, say $\underline{a}_2$. Thus, the second principal component is defined as

$$y_2(n) = \underline{a}_2'\underline{x}(n)$$

and has the properties that

$$\mathrm{mean}(y_2) = \underline{a}_2'\underline{m}$$

$$\mathrm{var}(y_2) = \lambda_2$$

$$\mathrm{cov}(y_1,y_2) = 0$$

This procedure can be continued (Dhrymes, 1970, p 56) by finding that linear combination of $\underline{x}(n)$ having maximal variance but which is uncorrelated with the previous principal components. Doing so gives M linear combinations of $\underline{x}(n)$, say

$$y_i(n) = \underline{a}_i'\underline{x}(n) \qquad i = 1,2,\ldots M$$

such that

$$\mathrm{var}(y_i) = \lambda_i$$

$$\mathrm{cov}(y_i,y_j) = 0 \qquad i \neq j$$

where the λ_i are the ordered (largest to smallest) eigenvalues of C and $\underline{a}_i$ are their associated orthonormal eigenvectors. The $y_i(n)$ form a set of sequences which are mutually uncorrelated linear combinations of $\underline{x}(n)$ having maximal variance. In general we define $\underline{y}'(n) = \{y_1(n), y_2(n), \ldots, y_M(n)\}$ as the vector of principal components of $\underline{x}(n)$ with $y_i(n)$ as the i-th principal component.

If we write the transpose of the eigenvectors as the rows of a matrix A we get

$$\underline{y}(n) = A\underline{x}(n) \tag{17}$$

A has the property that the sum of the variances of $\underline{y}(n)$, the eigenvalues of C, equals the sum of the variances of $\underline{x}(n)$ (Ready and Wintz, 1973).

Equation (17) defines a linear transformation from $\underline{x}(n)$ to $\underline{y}(n)$. It is known as the eigenvector, Hotelling, Karhunen-Loève, or principal component transform. This transform can be alternately derived as the solution to the optimal data compression problem. This derivation provides additional insight into the transform. The essential ideas will be presented briefly. For a detailed derivation see Hall (1979, p. 115), Ahmed and Rao (1975) or Kramer and Mathews (1956).

Say we have $\underline{x}(n)$ and expect a high mutual correlation between its elements. This correlation is indicative of redundant information, or signal, common to several sequences. In order to reduce storage space or transmission channel utilization we wish to represent $\underline{x}(n)$ using fewer data. We separate $\underline{x}(n)$ into different components, $\underline{y}(n) = A\underline{x}(n)$, discard some of these components, then store or transmit those remaining. The original $\underline{x}(n)$ are then reconstructed as well as possible using the remaining components. In general $\underline{x}(n)$ cannot be reconstructed exactly unless all of the components are retained.

Let us denote the reconstructed $\underline{x}(n)$ by $\underline{x}_0(n)$ and define the error in reconstruction as $\underline{E}(n) = \underline{x}(n) - \underline{x}_0(n)$. We also define the total error, T, as the sum of the variances of $\underline{E}(n)$. If A is constructed such that the total error is minimized when discarding successive elements of $\underline{y}(n)$ the principal component transform results. It follows that, if some of the $\underline{y}(n)$ are replaced by their mean values, T is equal to the sum of the variances of these replaced $\underline{y}(n)$. However, the variances of the $\underline{y}(n)$ are just the eigenvalues of C. Thus the eigenvalues contain information regarding the reconstruction error. In particular, the ratio of the first eigenvalue to the sum of the rest will give the proportion of the reconstruction which would have been obtained using the first principal component to that obtained using the rest of the principal components.

Obviously, if we want the optimal reconstruction using only one principal component, this component must be the most common to all of the $\underline{x}(n)$. Thus this ratio may also be considered as the ratio of the correlated or common signal in $\underline{x}(n)$ to the uncorrelated signal or noise.

4.3 Some Properties Of Principal Components

Let us examine some limiting cases to illustrate the behaviour of this transform as a common signal estimator. Let us write the suite of sequences $\underline{x}(n)$ as a common signal s(n) plus a 'noise' component $\underline{u}(n)$, in general different for each sequence. That is,

$$\underline{x}(n) = s(n) + \underline{u}(n)$$

Let us assume that the mean values of s(n) and $\underline{u}(n)$ are zero and that the variances of the $\underline{u}(n)$ are identical.

A normalized linear combination of the $\underline{x}(n)$ which uses no information about the $\underline{x}(n)$ uses identical weights. Ready and Wintz (1973) call this the coherent sum of the $\underline{x}(n)$ and use it for comparison to the first principal component.

Consider the case where the $\underline{u}(n)$ are identically correlated with each other ($\mathrm{cov}(u_i,u_j) = f$, f = constant) and uncorrelated with s(n). Then the covariance matrix C has elements

$$C_{ij} = \mathrm{cov}(x_i,x_j) = E\{[s(n)+u_i(n)][s(n)+u_j(n)]\}$$

Expanding the sum yields

$$\mathrm{cov}(x_i,x_j) = \mathrm{var}(s) + f$$

or, as var(s) is a constant,

$$\mathrm{cov}(x_i, x_j) = \text{constant}$$

Thus C is a constant matrix of unit rank having only one non-zero eigenvalue. The first eigenvector will have identical elements and the first principal component will equal the coherent sum. All of the variance will be concentrated into the first principal component. In fact, if the noises were all zero the same result would follow. This indicates that the correlations between noises makes them indistinguishable from signal.

Now consider the case where the noises are uncorrelated with each other and with the signal. Then

$$C_{ij} = \mathrm{var}(s) + d_{ij} \cdot f$$

where d_{ij} is the Kronecker delta and f is a constant. Thus C is a constant matrix plus a another constant added to the diagonal. The first eigenvalue of C is $\lambda_1 = \lambda + f$ where λ is the eigenvalue of $C_{ij} = \mathrm{var}(s)$. All the other eigenvalues are identical and equal to the constant f (Ready and Wintz, 1973). We can say that the signal has concentrated in the first principal component and that the noise has spread uniformly among all the principal components. The first principal component is equal to the coherent sum. All the other eigenvectors are orthogonal to the first and therefore the sum of their elements is zero. This implies that there is no signal in these principal components.

In the general case noises are correlated with signal and with each other. We can say that the most correlated signal is in the first principal component and the uncorrelated noise is spread uniformly over all the principal components. Signals having lesser and lesser correlation appear in successive principal components.

Because of this concentration of signal in the first principal component, and the spreading out of the noise over all the principal components, we can use the eigenvalues as a measure of how much signal was in the original data. In particular, the ratio of the first to the sum of the rest can be used as a measure of the signal to noise ratio.

Note that there is no restriction on the sign of the elements of $\underline{a}$. Thus, in a principal component, one or more of the $\underline{x}(n)$ may be inverted. This could occur if some of the $\underline{x}(n)$ had an inverted signal $-s(n)$, or if the noise led to negative correlations.

4.4 Summary

Signal common to several sequences can be estimated using linear combinations. Interpreting variance as an information measure leads to the covariance matrix. The eigenvectors of this matrix define principal components and the eigenvalues show where the information lies.

V. WAVELET ESTIMATION

5.1 The Problem

In the field of seismic exploration the recorded data are often modeled as the convolution of a wavelet with an impulse sequence. The wavelet is a time limited function, usually having a smooth spectrum (zeros far from the unit circle) (Tribolet, 1979, p. 9). The impulse sequence consists of isolated non-zero values.

Physically, the wavelet models a wave propagating in the Earth. The impulse sequence is a consequence of the Earth's structure and the physical geometry of the problem. The data are often available in the form of a suite of time sequences called traces. The wavelet is assumed constant from trace to trace while the impulses are, in general, different. (See Robinson and Treitel (1980) or Telford et al. (1976) for further information.)

This chapter deals with the problem of estimating the wavelet, given the above data.

5.2 A Solution

Three methods of wavelet estimation were discussed by Lines and Ulrych (1977): the Weinér-Levinson, Wold-Kolmogorov and homomorphic methods. The former two methods require a minimum delay wavelet and stationary impulses while the latter requires the separation of the complex cepstra of the wavelet and the impulse sequence. The method of minimum entropy deconvolution (Wiggins, 1978), while not designed as a wavelet estimater, can yield such an estimate (Ooe and Ulrych, 1979). It makes an assumption about the entropy, or simplicity, of the impulse sequence. We would like to avoid making these assumptions on an arbitrary basis. In general, we would like to include as much a priori information while excluding as many arbitrary restrictions as possible. For example, if we have an a priori estimate of the wavelet cepstrum's length we would like to include this.

We assume a random, not necessarily stationary, impulse sequence. If the data are a single time sequence, this sequence can be segmented. If the data are already in the form of a suite of sequences this segmentation may not be necessary. Each sequence is modeled as the convolution of a wavelet with an impulse sequence.

We write the data as

$$x_k(n) = w(n)*i_k(n) \qquad k=1,...M \tag{1a}$$

or in vector notation

$$\underline{x}(n) = w(n)*\underline{i}(n) \tag{1b}$$

where $\underline{x}(n)$, $w(n)$ and $\underline{i}(n)$ represent, respectively, the data, the wavelet and the impulse sequences. Each $\underline{x}(n)$ is thus formed by the convolution of the same $w(n)$ with a different

$i(n)$.

The convolutional operation in (1) is not in a form most amenable to analysis. However, by using the homomorphic transformation discussed earlier, the equations can be mapped from a convolutional space to an additive space. That is

$$D[w(n)*\underline{i}(n)] = D[w(n)] + D[\underline{i}(n)] \tag{2a}$$

or

$$\underline{\hat{x}}(n) = \hat{w}(n) + \underline{\hat{i}}(n) \tag{2b}$$

where the circumflex denotes a complex cepstrum. The data are now in the form of a suite of sequences, each containing a common component, $\hat{w}(n)$, and a different component, $\underline{\hat{i}}(n)$. We wish to estimate $\hat{w}(n)$ from the $\underline{\hat{x}}(n)$. This has been done by averaging the $\underline{\hat{x}}(n)$ (Clayton and Wiggins, 1976; Otis and Smith, 1977). Averaging is a data-independent technique which ignores any information in the data. Principal component analysis, on the other hand, is data-dependent and uses information in an optimal fashion (Ulrych et al, 1983). Thus it may be an improvement over averaging. The homomorphic transform provides us with the possibility of separating, at least partially, $\hat{w}(n)$ from the $\underline{\hat{i}}(n)$ by low quefrency windowing (Ulrych, 1971).

Before discussing the principal component (P.C.) method, let us examine averaging. Assume that, for each fixed n, the elements of $\underline{\hat{i}}(n)$ can be considered as random variables with identical probability distributions and zero mean. Then $1/M \sum_{k=1}^{M} \hat{i}_k(n)$ will tend to zero for large M (Otis and Smith, 1977). The estimate of $\hat{w}(n)$, denoted by $\hat{w}_e(n)$, is defined as

$$\hat{w}_e(n) = 1/M \sum_{k=1}^{M} \hat{x}_k(n) \tag{4a}$$

$$\hat{w}_e(n) = 1/M \sum_{k=1}^{M} [\hat{w}(n)+\hat{i}_k(n)] \tag{4b}$$

$$\hat{w}_e(n) = \hat{w}(n) + 1/M \sum_{k=1}^{M} \hat{i}_k(n) \tag{4c}$$

The estimated cepstrum equals the wavelet cepstrum if the sum in equation (4c) is zero. The inverse homomorphic transform of $\hat{w}_e(n)$ yields $w_e(n)$, the wavelet estimate.

Proceeding as above, we define the P.C. estimate of $\hat{w}(n)$ as

$$\hat{w}_e(n) = [\sum_{k=1}^{M} a_k \hat{x}_k(n)] / \sum_{k=1}^{M} a_k \tag{5a}$$

where the a_k are the first P.C. weights. The reason for normalizing the estimate can be shown as follows. Equation (5a) can be written as

$$\hat{w}_e(n) = [\sum_{k=1}^{M} a_k \hat{w}(n) + \sum_{k=1}^{M} a_k \hat{i}_k(n)] / \sum_{k=1}^{M} a_k \tag{5b}$$

or

$$\hat{w}_e(n) = \hat{w}(n) + \sum_{k=1}^{M} a_k \hat{\imath}_k(n) / \sum_{k=1}^{M} a_k \qquad (5c)$$

which is analogous to equation (4c). Again, if the sum is zero, the estimated cepstrum equals the wavelet cepstrum.

It should be noted that, in the original convolution, the wavelet's scale and time origin were lost. In principle they cannot be recovered and we will not be concerned with them. If the sum in equations (4c) or (5c) is an impulse at the origin, $w_e(n)$ will differ from $w(n)$ only by a multiplicative scale factor (see property 6 of the complex cepstrum). For the examples used in this thesis we arbitrarily set the zero quefrency point to zero to normalize the scale factor and prevent it from affecting cepstral correlations.

5.2.1 The Smoothing Function

In practice we have a finite number of data and cannot expect the impulse cepstra to average to zero. Suppose that, in equation (4c) or (5c), the sum is not zero. Write the sum as $\hat{g}(n)$. Then (4c) or (5c) becomes

$$\hat{w}_e(n) = \hat{w}(n) + \hat{g}(n)$$

with inverse homomorphic transform

$$w_e(n) = w(n) * g(n)$$

where $\hat{g}(n)$ is the complex cepstrum of $g(n)$. The estimate is the convolution of the true wavelet with g(n). g(n) may be considered as a smoothing function which distorts the estimate. (It is worth noting the similarity of this to a result from the field of linear inversion where the estimated model is the true model viewed through a resolution function (Wiggins, 1972; Oldenburg and Samson, 1979).

In order to illustrate this smoothing function consider the following example. (One may consider the data as samples of a time function with a 1 ms interval. Then one would read 'ms' for 'points'.) We generate a suite of six data sequences, i(n), of length 70 ms. The data are impulses which are Poisson distributed in spacing and Gaussian distributed in amplitude. The Poisson parameter is 11 ms, which is both the mean and variance of the distribution. Figure 7 shows the data and the corresponding smoothing function obtained by cepstral averaging and low quefrency windowing with a Hanning window of length 40 ms (centred on the origin). The window length is a result of hindsight and will be shown to be appropriate in subsequent examples. Note that the smoothing function approximates an impulse. The complex cepstra and their average are shown in Figure 8. Note the exponential decay indicating the lack of cepstral aliasing.

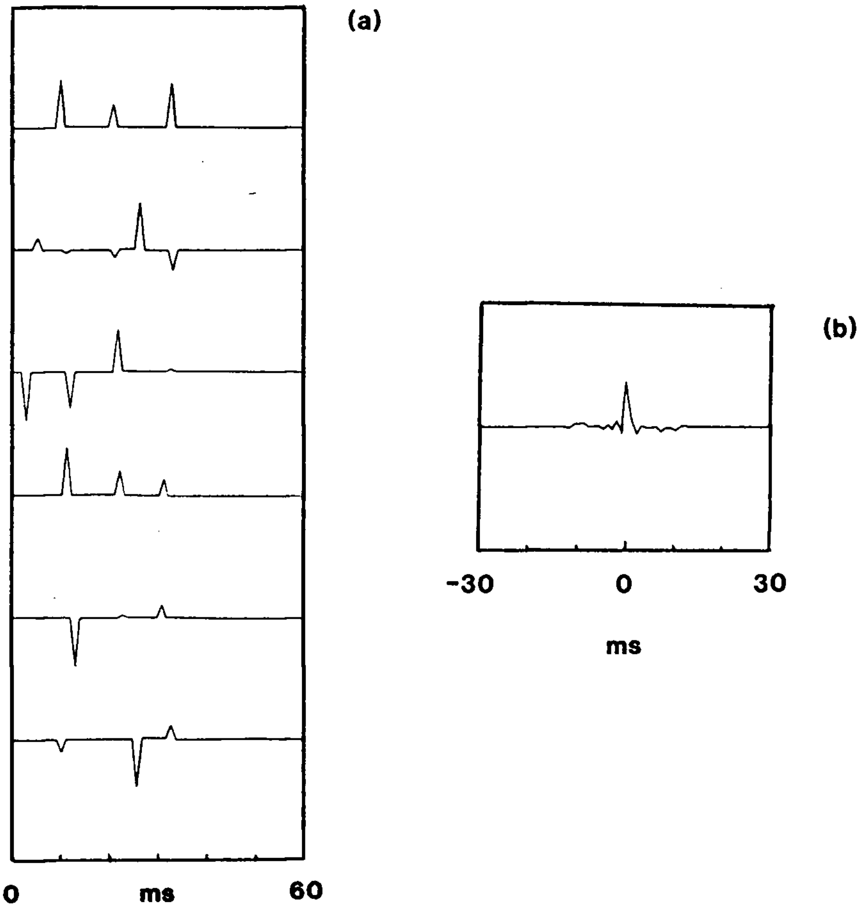


Figure 7 - Smoothing Function

Six random impulse sequences (a) and their smoothing function due to cepstral avaraging and windowing (b).

5.2.2 The Wavelet

Now let us consider the wavelet. We assume a mixed delay wavelet, shown with its cepstrum in Figure 9. The wavelet is time limited to 20 ms and its cepstrum is quefrency limited to about 25 ms. This justifies the use of a 40 ms window for comparisons using this wavelet. Note the cepstrum's higher amplitude than the previous impulse cepstra. Plotting is to the same scale. The power spectrum of the wavelet is also shown. Note its low pass nature. Small changes to the high frequency spectrum due to convolution or noise may dominate

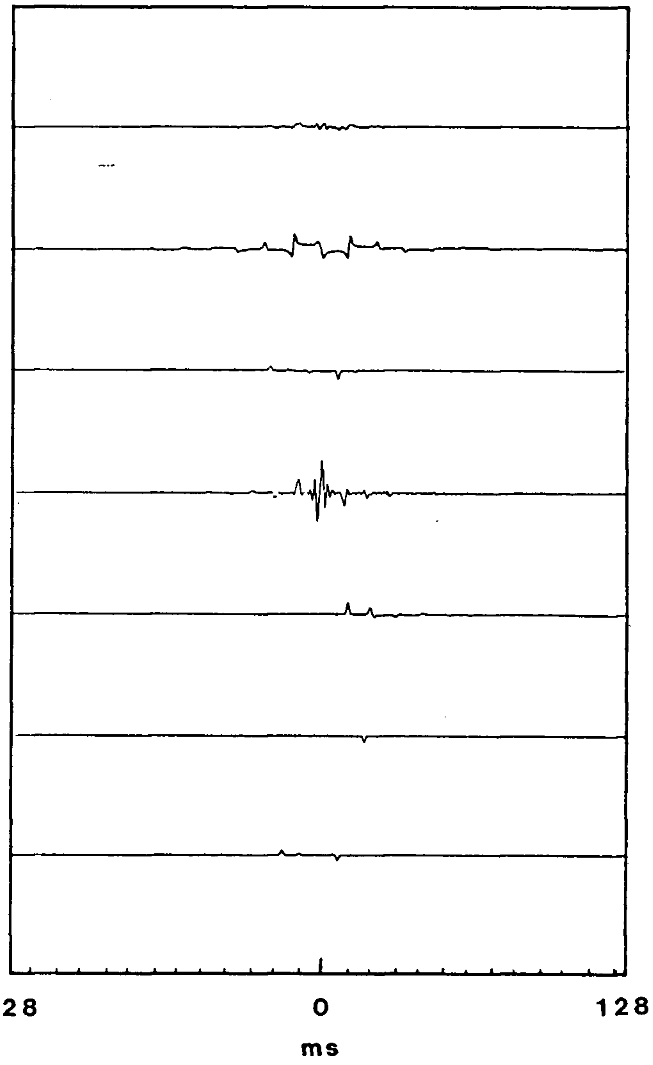


Figure 8 - Complex Cepstra Due to Impulses

These cepstra correspond to the impulse sequences of Figure 7. Note the exponential decay.

the linear phase computation. This can be avoided by setting the phase to zero above some frequency and removing the linear trend to that point. If the energy content for greater frequencies is small, changes to the wavelet and its cepstrum will be small.

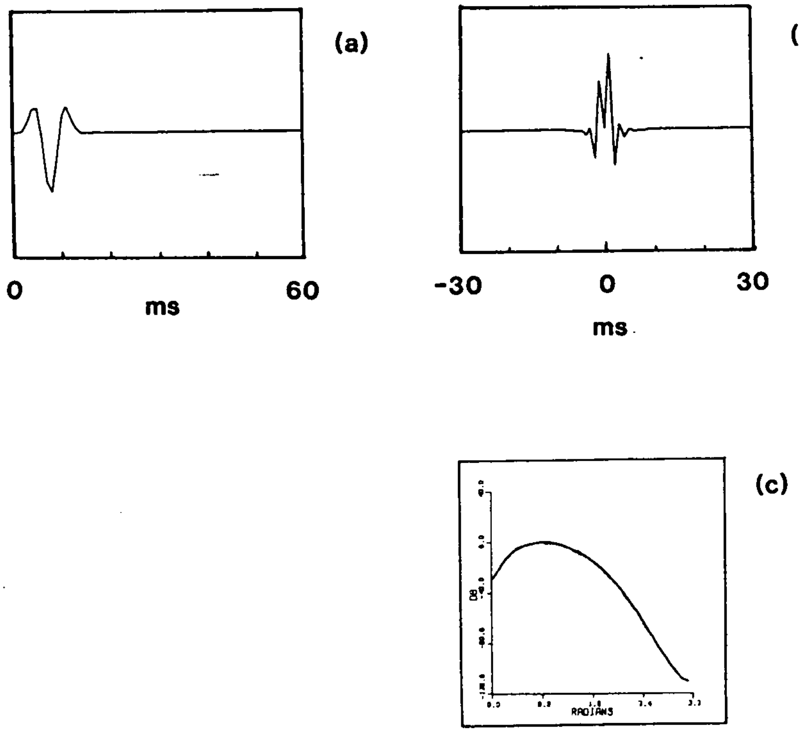


Figure 9 - An Assumed Wavelet

The assumed wavelet (a) is mixed delay. Its cepstrum (b) is time limited and its power spectrum (c) is low pass.

5.2.3 Principal Components vs Averaging

Let us now compare the P.C. method to averaging. A feature of the P.C. method is its ability to discriminate between cepstra. This is demonstrated in Figure 10. The input data are the convolution of our wavelet with three sequences of random impulses. Random noise has been added to the first sequence. The noise is zero mean Gaussian with a variance of 25% of that of the noise free sequence. The cepstrum of the noisy input is noticeably different from the others. The P.C. estimate yielded weights of [0.1,0.7,0.7]

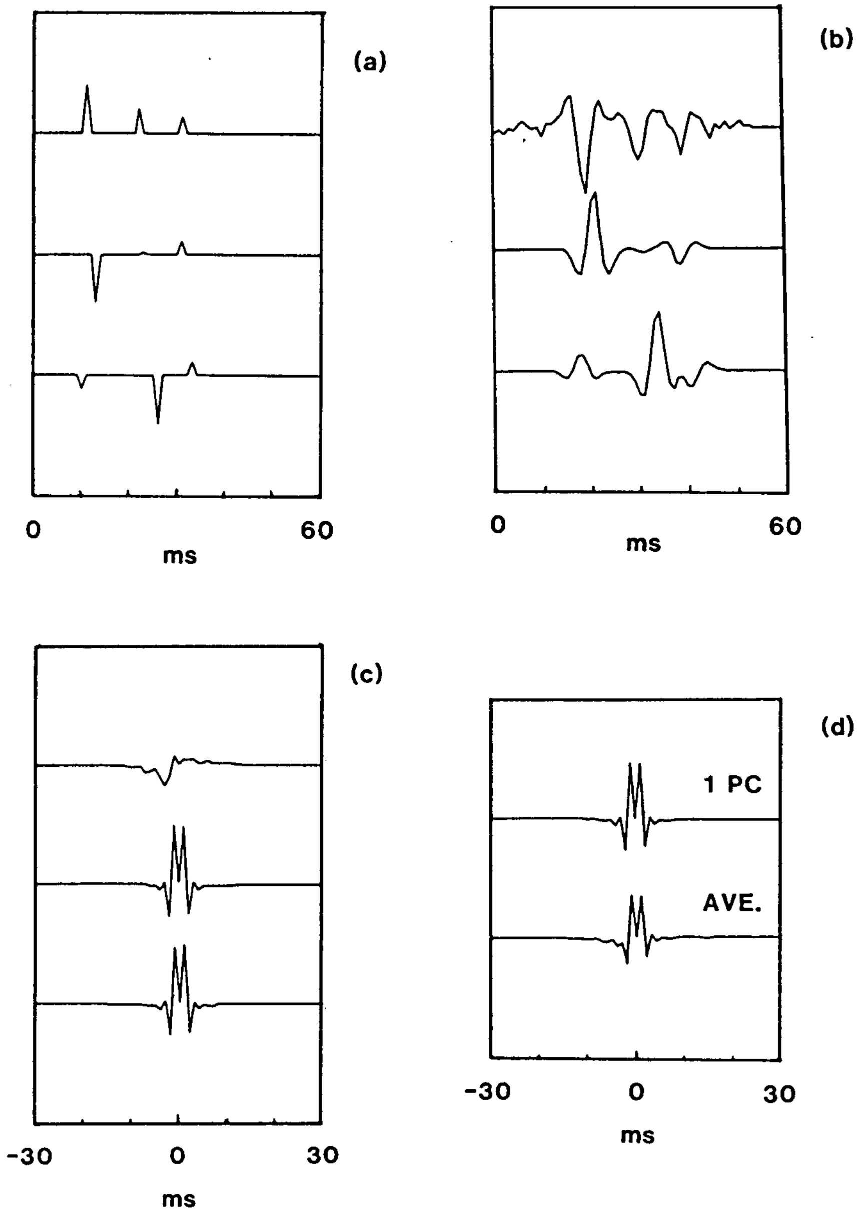


Figure 10 - Discrimination by Principal Components

Input impulse sequences (a) and data derived by their convolution with a wavelet (b). The top data sequence has additive noise. Cepstra (c) show the noisy input's difference. The average and first principal component of (c) are shown in (d).

indicating successful discrimination of the noisy input. The wavelet estimates due to averaging and the first P.C. are shown in Figure 11.

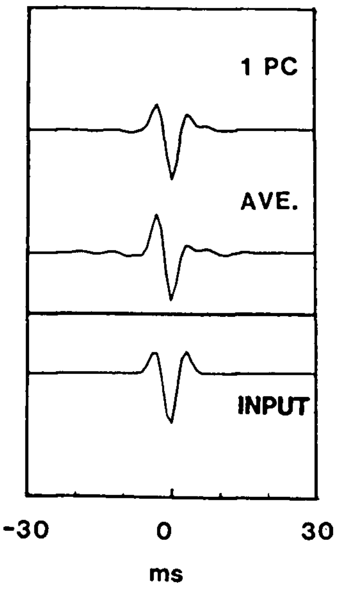


Figure 11 - Wavelet Estimates

Wavelet estimates derived from the first principal component (1 PC) and averaging (AVE.) of cepstra of Figure 10. The input wavelet is shown at the bottom.

To obtain a numerical comparison of the estimates we define a misfit error. The misfit is the sum of squares of the difference between the wavelet and the estimate. The wavelet and estimate are first scaled to unit variance, then time shifted for minimum misfit. Since we are calculating a 256 point cepstrum, the wavelet estimate is 256 points long. Thus the null estimate has a misfit of 256 and the wavelet itself has a zero misfit. In this example the average estimate had a misfit of 78 versus the principal component's 40. These indicate an improvement of the P.C. method over averaging.

5.2.4 Multiple Sequence Data

Let us now apply these methods to a more general example. The data are 15 sequences generated by the convolution of our wavelet with random impulse sequences of length 70 ms (Figure 12). The Poisson parameter is 6 ms implying an appreciable wavelet overlap due to the convolution. The wavelets estimated from averaging and the first two P.C.s are shown in Figure 13. Their respective misfits are 8.1, 5.9 and 370. The first P.C. estimate has a smaller misfit than the average and the second P.C. bears little resemblance to the wavelet. These results are encouraging and lend credence to our expectations of the P.C. method.

On the basis of the previous result for a noisy input we proceed by adding random noise. The noise is Gaussian with a variance of 5% of that of the data. The noisy data are shown in Figure 14 along with the wavelet estimates. Misfits of the average and first two P.C.s are 20, 280 and 16. Note that the misfits are generally larger than for the noise free case, which is expected. However, the behaviour of the first two P.C.s is unexpected and bears closer examination. It appears that the proper wavelet estimate is in the second P.C. rather than the first. This is not predicted by our theoretical development and examples. In fact, it seems paradoxical. To resolve this apparent paradox, let us re-examine the homomorphic transform and its interactions with the time and quefrency domains.

The behaviour of the homomorphic transform in the

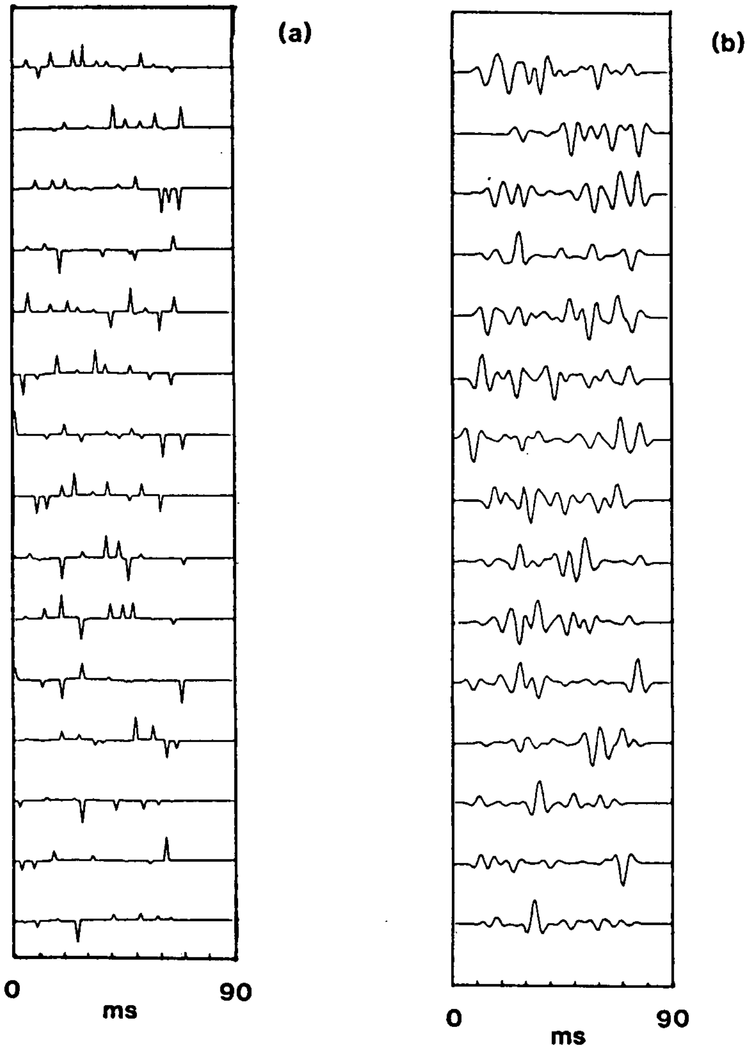


Figure 12 - Multiple Sequence Data

Fifteen random impulse sequences (a) and convolved with a wavelet (b).

presence of additive noise is not well understood. However, the phase spectrum may be strongly affected (Buttkus, 1975; Clayton and Wiggins, 1976; Jin and Rogers, 1983). It is possible that P.C. analysis can aid in our understanding of how noise affects the cepstrum. The appearance of the wavelet cepstrum in the second P.C. may imply that noise has induced

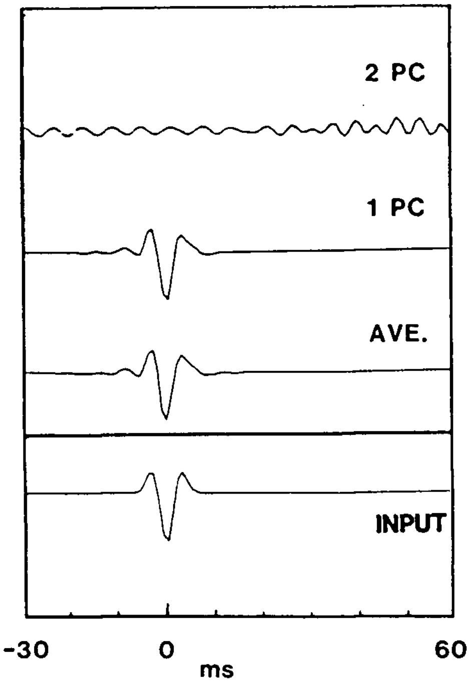


Figure 13 - Wavelet Estimates

Wavelet estimates due to first two principal components (1 PC, 2 PC) and average (AVE.) of cepstra of data in Figure 12. The input wavelet is shown at the bottom.

a common signal in the cepstra. This signal appears in the first P.C. and the next most common signal, due to the wavelet, appears in the second P.C. This hypothesis can be tested by examining the time-quefrencey relationship for pure noise inputs.

In practice we cannot generate completely uncorrelated noise. However, we can compare the correlations of noise in the time domain (input) with those in the quefrency domain (output). Poorly correlated sequences have a covariance matrix with similar eigenvalues. This is due to the lack of a common signal and the distribution of uncorrelated noise over

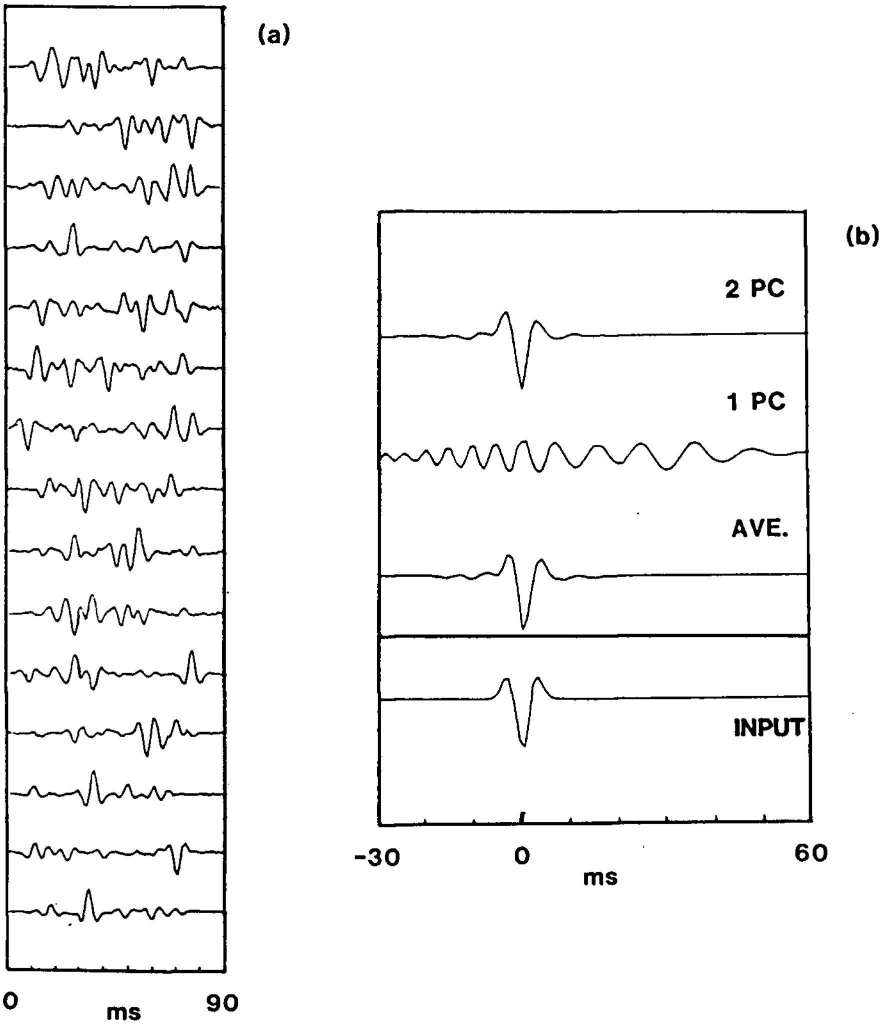


Figure 14 - Noisy Data

The data of Figure 12 but with 5% additive noise (a). (b) shows the wavelet estimates due to the first two principal components (1 PC, 2 PC) and averaging of cepstra. The input wavelet is shown at the bottom.

all the P.C.s. On the other hand, highly correlated sequences yield (ordered) eigenvalues which decrease rapidly. This occurs because the sequences contain mostly common signal. Thus, the eigenvalues of inputs and outputs are indicative of changes in relative correlations.

Consider the suite of six input noise sequences shown in Figure 15. The noise is zero mean Gaussian. Its variance is irrelevant as scale factors are removed in the cepstra (Figure

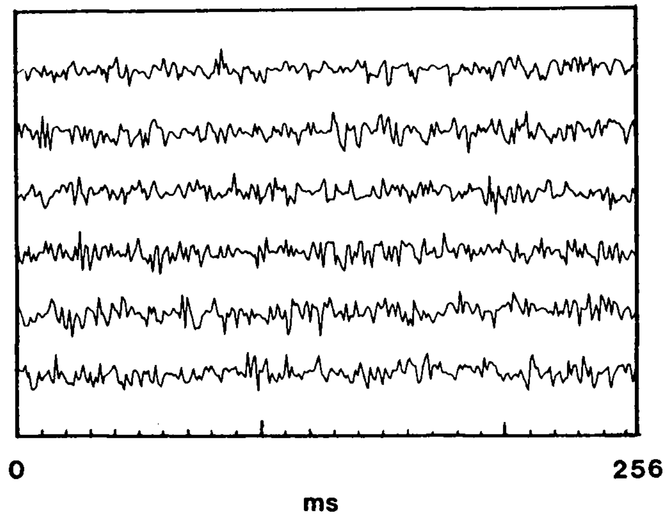


Figure 15 - Noise Inputs
A suite of random noise inputs.

16). Note the high amplitude of the cepstra and the strong odd component due to the phase. The cepstra are plotted to the same scale as the previous two examples. For comparison, each set of eigenvalues is normalized by dividing by the largest value. The eigenvalues for the output decrease more rapidly than for the input (Figure 17). This implies that the outputs (cepstra) are more highly correlated than the inputs (time). One must be cautioned against inferring that the transform has produced non-random outputs from random inputs. Correlation may not be a suitable measure of similarity in the quefrency domain for time domain noise. It is likely that this noise induced correlation dominates the first P.C. The observation that the wavelet estimate appears in the second P.C. implies that, although the noise may induce a common component, this is uncorrelated with the wavelet cepstrum.

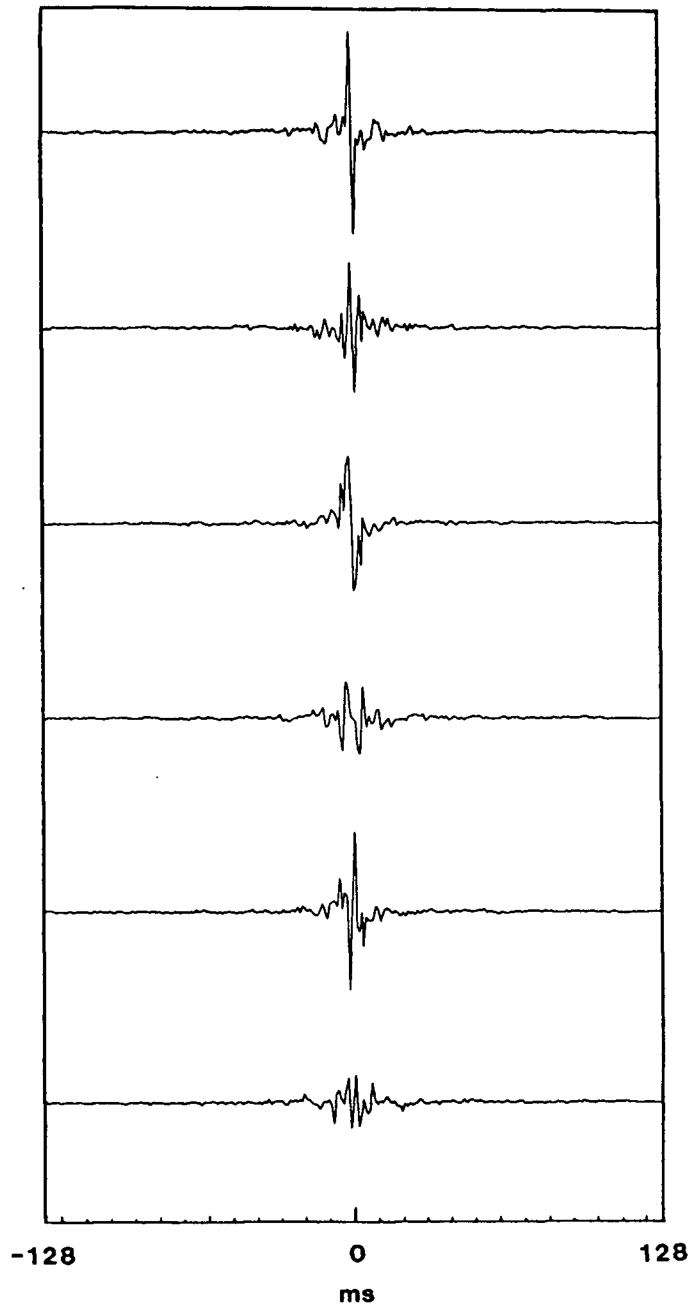


Figure 16 - Cepstra of Noise

The complex cepstra of the noise inputs of Figure 15.

This follows because the P.C.s are mutually uncorrelated.

We can gain further insight by examining cepstral covariance matrices. For comparison, the wavelet alone yields

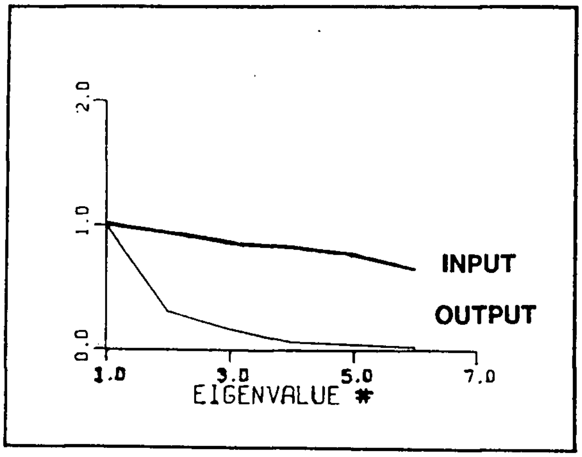


Figure 17 - Eigenvalues of Input and Output

The eigenvalues of input (dark line) decrease more slowly that those of the output. This implies that the output is more correlated than the input.

a constant matrix with the value 0.063. The covariance matrix due to input noise alone is shown in Figure 18. Note the range of values of both signs. Several elements have a

$$\begin{bmatrix} 0.633 & 0.314 & 0.400 & 0.079 & -0.411 & -0.130 \\ 0.314 & 0.359 & 0.241 & -0.097 & -0.365 & -0.143 \\ 0.400 & 0.241 & 0.574 & 0.068 & -0.219 & -0.029 \\ 0.079 & -0.097 & 0.068 & 0.244 & 0.042 & 0.065 \\ -0.411 & -0.365 & -0.219 & 0.042 & 0.496 & 0.166 \\ -0.130 & -0.143 & -0.029 & 0.065 & 0.166 & 0.146 \end{bmatrix}$$

Figure 18 - Covariance Matrix of Noise

The covariance matrix of the noise cepstra of Figure 16. These elements compare with the wavelet cepstrum's correlation of 0.063.

magnitude greater than the covariance of the wavelet cepstrum (0.063). Thus covariances of cepstra due to noise alone can be much larger than those due to signal alone.

Now let us examine the matrices from the previous example. For clarity, only the result from the first six inputs will be considered. Consider the matrices derived from the cepstra of the impulse sequences, C_0; from the impulses convolved with the wavelet with no noise, C_1; and with 5% noise, C_2 (Figure 19). The impulse cepstra, C_0, generally

$$\begin{bmatrix} 0.002 & 0.002 & -0.000 & -0.001 & -0.001 & -0.000 \\ 0.002 & 0.114 & -0.009 & -0.036 & -0.046 & -0.011 \\ -0.000 & -0.009 & 0.017 & 0.014 & 0.007 & 0.008 \\ -0.001 & -0.036 & 0.014 & 0.039 & 0.035 & 0.021 \\ -0.001 & -0.046 & 0.007 & 0.035 & 0.092 & 0.022 \\ -0.000 & -0.011 & 0.008 & 0.021 & 0.022 & 0.023 \end{bmatrix} \quad \textbf{(a)}$$

$$\begin{bmatrix} 0.064 & 0.065 & 0.064 & 0.064 & 0.064 & 0.066 \\ 0.065 & 0.127 & 0.075 & 0.055 & 0.023 & 0.071 \\ 0.064 & 0.075 & 0.069 & 0.062 & 0.051 & 0.065 \\ 0.064 & 0.055 & 0.062 & 0.068 & 0.069 & 0.065 \\ 0.064 & 0.023 & 0.051 & 0.069 & 0.104 & 0.061 \\ 0.066 & 0.071 & 0.065 & 0.065 & 0.061 & 0.072 \end{bmatrix} \quad \textbf{(b)}$$

$$\begin{bmatrix} 0.009 & 0.006 & 0.009 & 0.008 & 0.002 & 0.009 \\ 0.006 & 0.016 & 0.016 & -0.001 & -0.005 & 0.006 \\ 0.009 & 0.016 & 0.028 & -0.000 & -0.017 & 0.009 \\ 0.008 & -0.001 & -0.000 & 0.012 & 0.003 & 0.010 \\ 0.002 & -0.005 & -0.017 & 0.003 & 0.044 & -0.008 \\ 0.009 & 0.006 & 0.009 & 0.010 & -0.008 & 0.015 \end{bmatrix} \quad \textbf{(c)}$$

Figure 19 - Covariance Matrices

The covariance matrices due to impulse cepstra (a), impulses convolved with a wavelet (b) and impulses convolved with a wavelet, with 5% additive noise (c).

yield off diagonal terms less than the wavelet's cepstral covariance (0.063). The covariance matrix due to impulses convolved with the wavelet, C_1, has less structure than C_0 and generally higher correlations, all positive. The matrix due to noisy data, C_2, has generally smaller correlations than C_1

and more structure, including negative corrrelations.

These observations can be interpreted as follows. Correlations due to the impulse sequence contribute less to the matrix structure than those due to the wavelet. Time domain noise can dominate both of these in the cepstral covariance matrix and strongly affect the P.C. estimate. Thus, in a sense, P.C.s may amplify noise effects compared with averaging.

The presence of both positive and negative P.C. weights implies that the wavelet cepstrum will tend to cancel. These may thus be used as an indicator of noise dominance.

5.2.5 Single Sequence Data

Having achieved an understanding of some of the interactions between the homomorphic transform and P.C. analysis, let us proceed.

Consider the case where the data are a single sequence. This can be converted to a multi-sequence input by segmentation. Segmentation can be done on the basis of the envelope (Ulrych et al, 1983). The envelope is defined as the magnitude of the complex signal (Bracewell, 1965, p 271; Clayton and Wiggins, 1976) and its maxima can be used as an indicator of the locations of the underlying impulses (Taner and Sheriff, 1977). The envelope is smoothed to reduce noise effects and merge closely spaced maxima. Centering windows on these maxima yields the trace segments, which may overlap. This windowing will, in general, destroy the original convolutional model and has a poorly understood effect on the homomorphic transformation (Tribolet, 1979). However, if the window changes on a longer scale than the wavelet, we expect it to affect only the impulse sequence. Once the data have been converted into the desired form we proceed as before.

Let us consider a single sequence example. The data are the convolution of a single 500 ms impulse sequence convolved with our wavelet (Figure 20). Envelope maxima yield window locations. A 90 ms Hanning window was used to extract 18 segments (Figure 21). The wavelet and estimates due to averaging and the first and second P.C.s are also shown. Their respective misfits are 60, 59, and 409. Note that the average and first P.C. have similar misfits, the P.C.'s being marginally better. The second P.C. estimate is dissimilar to the wavelet. The corresponding cepstra are shown in Figure 22 with the weights yielded by the first P.C. The lack of variability among these weights and the misfit's similarity to the average's indicate that the impulse cepstra did not strongly affect the covariance structure.

Let us now examine the effect of additive noise in this context. The data with 5%, 10% and 30% noise are shown in Figure 23. (Note that individual segments may have different signal to noise ratios.) The wavelet, average estimate and first two P.C. estimates are shown in Figure 24. The segments' cepstra and the corresponding first P.C. weights are shown in Figure 25. (Note that, for 30% noise, 17 segments were defined.) Misfits of the various estimates are

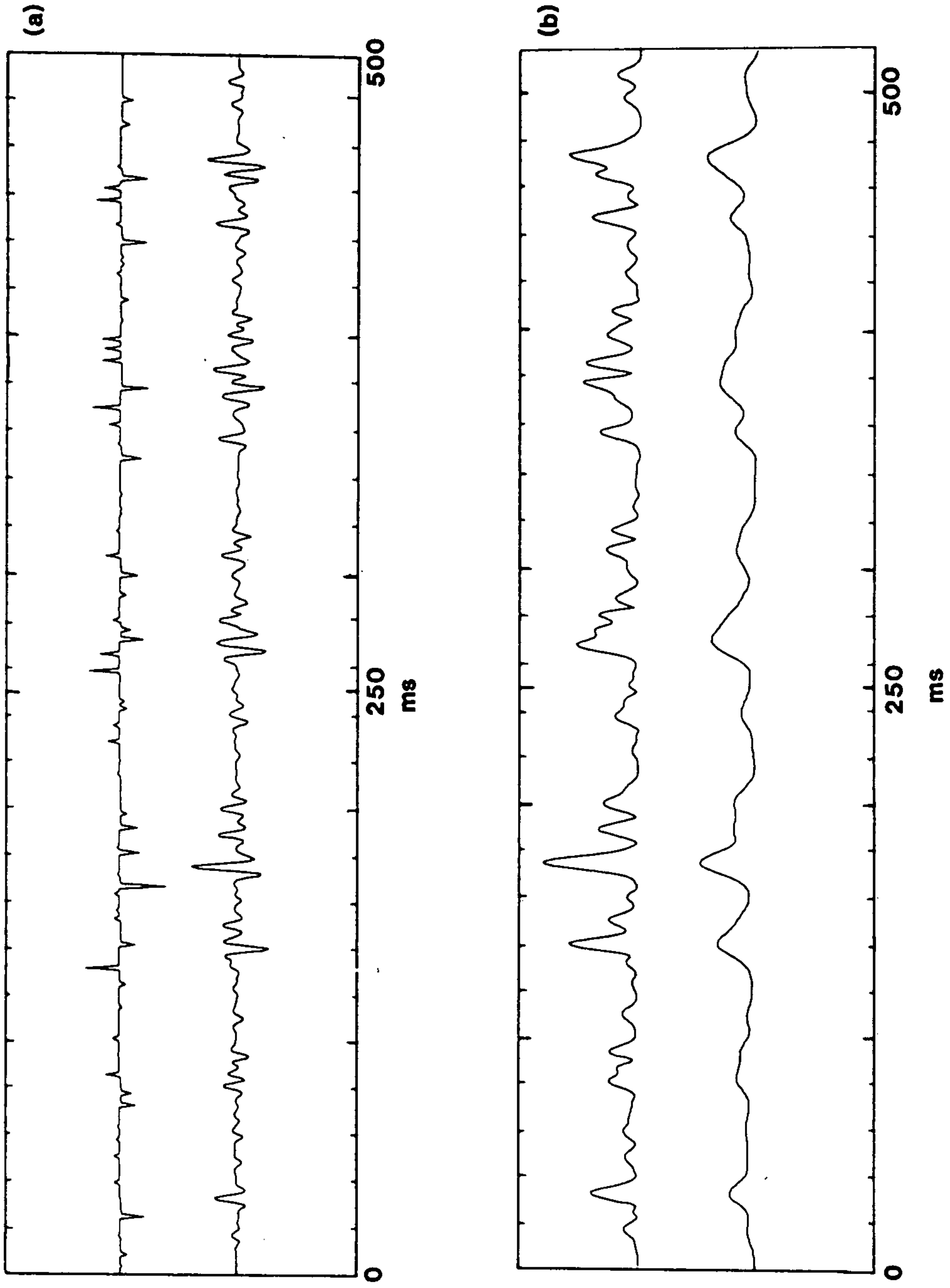


Figure 20 - Single Sequence Data

A random impulse sequence and data abtained by convolution with a wavelet (a). The envelope before and after smoothing (b).

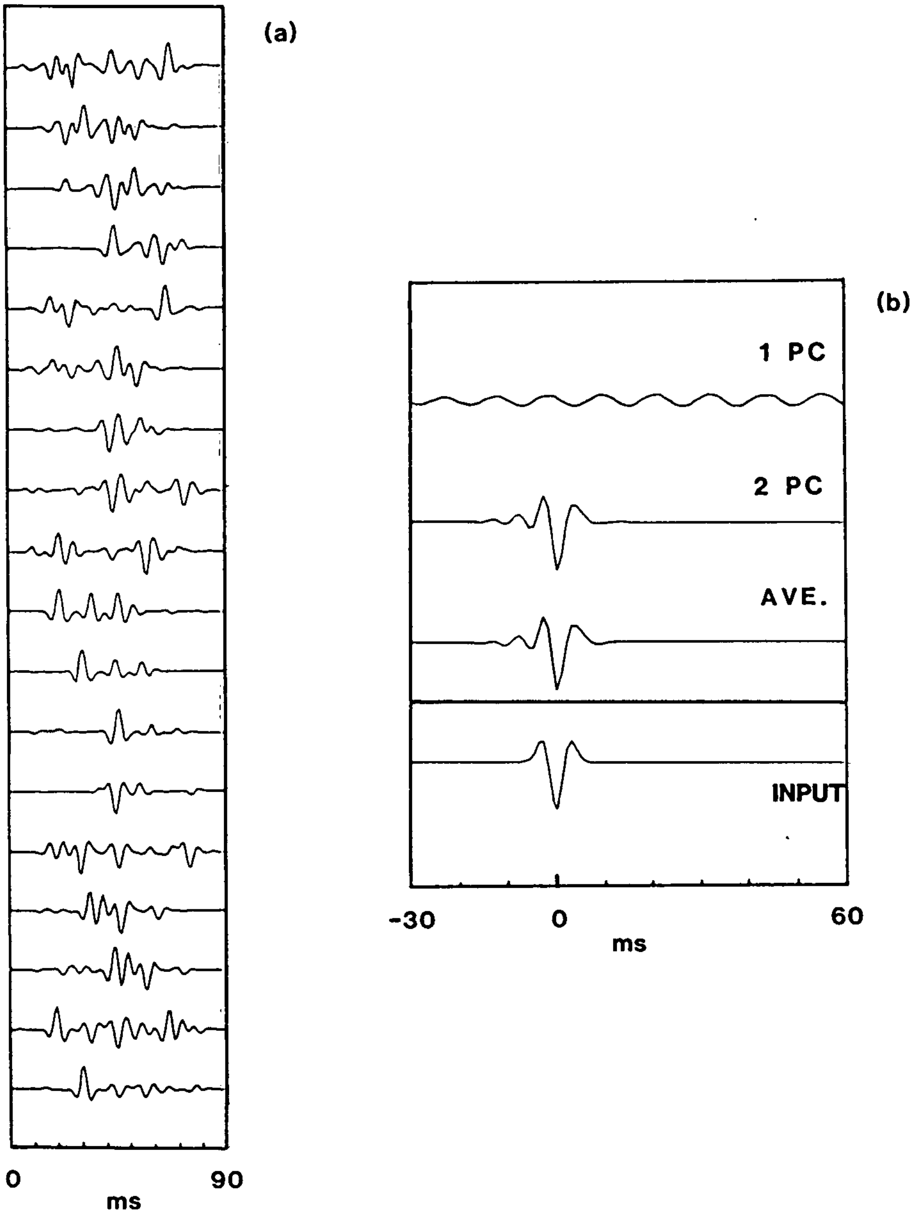


Figure 21 - Segmentation and Estimation

Segments of the data in Figure 20 (a) and wavelet estimates due to first two principal components (1 PC, 2 PC) and averaging (AVE.) of cepstra (b). The input wavelet is shown at the bottom of (b).

shown in Table 1. The second P.C. becomes a better estimator than the first as the noise level increases. In fact, for 30% noise, it is better than the average. Examination of the

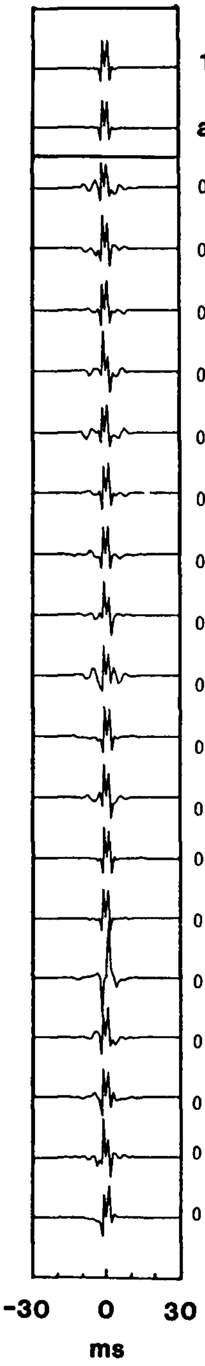


Figure 22 - Cepstra of Segments

The cepstra of the segments shown in Figure 21. The first principal component (1 PC) and average (AVE.) cepstra are also shown. The weights used to form the first P.C. weights are shown next to the corresponding cepstra.

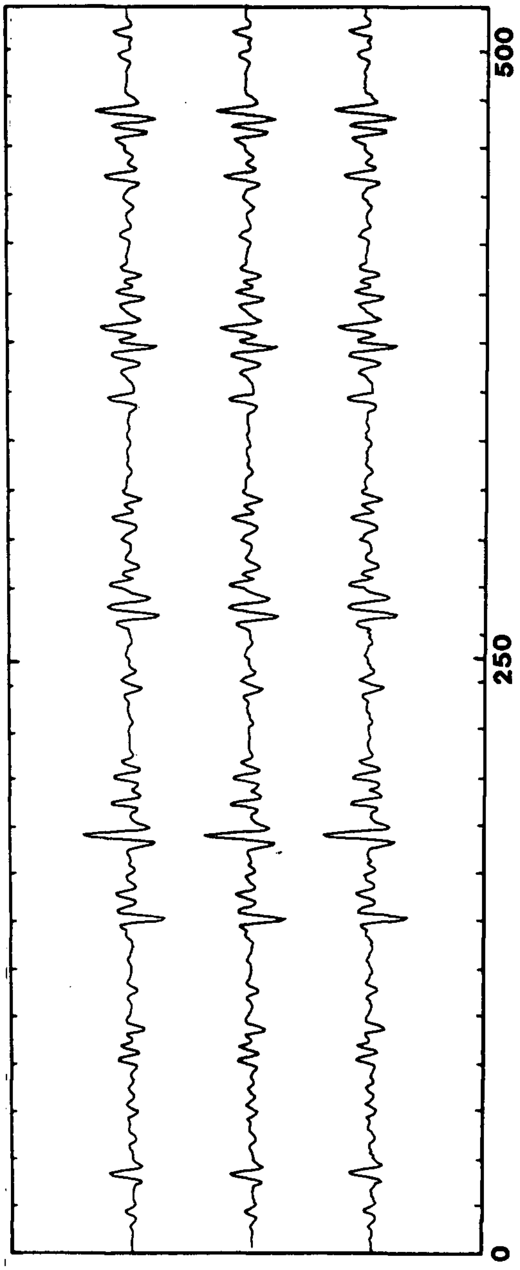


Figure 23 - Noisy Single Sequence Data

A data sequence with 5%, 10% and 30% additive noise (top to bottom).

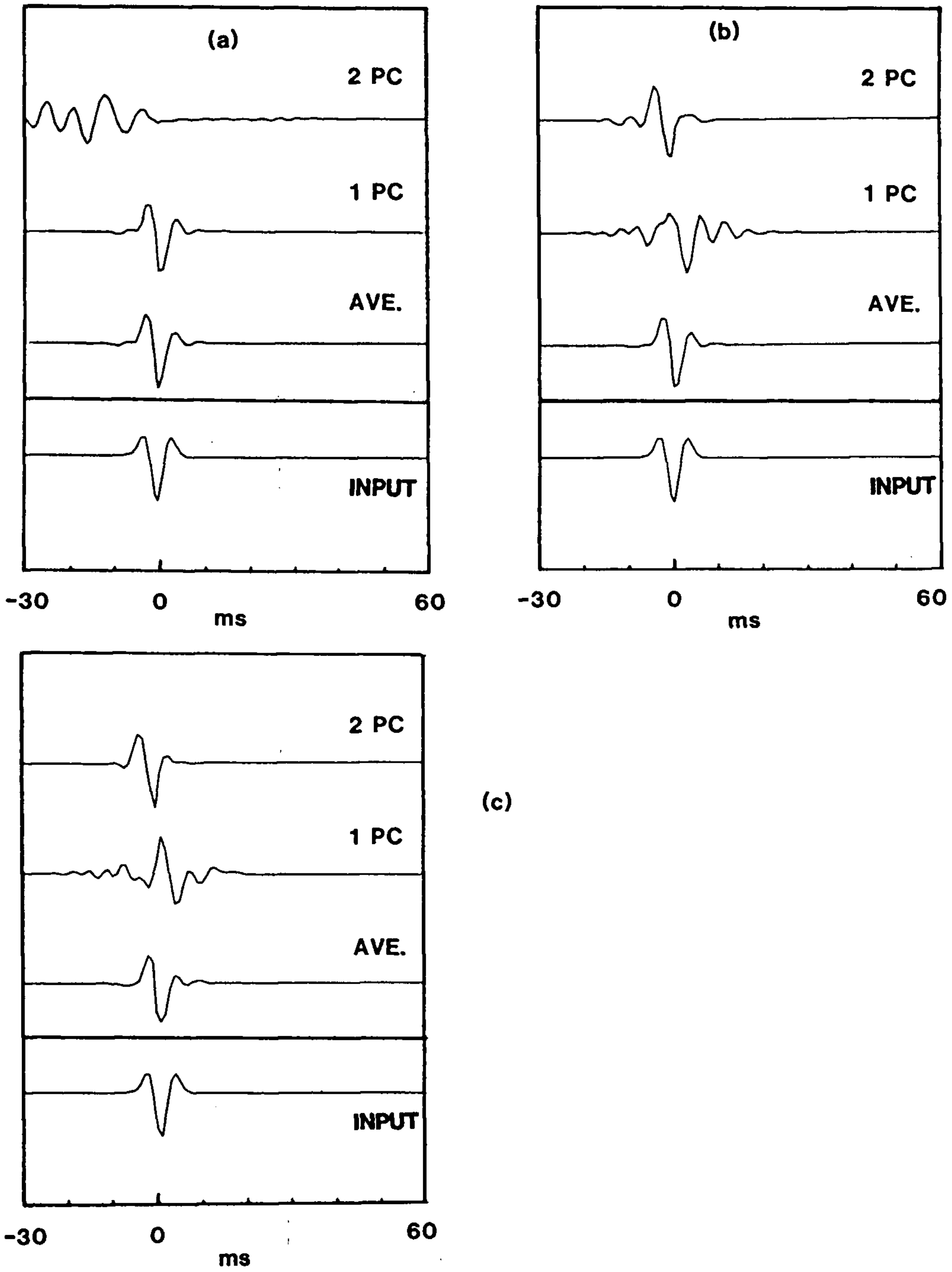


Figure 24 - Wavelet Estimates

The wavelet estimates due to the noisy data of Figure 23. The first (1 PC), second (2 PC) and average (AVE.) estimates are shown for 5% (a), 10% (b) and 30% (c) noise. The input wavelet is shown at the bottom of each panel.

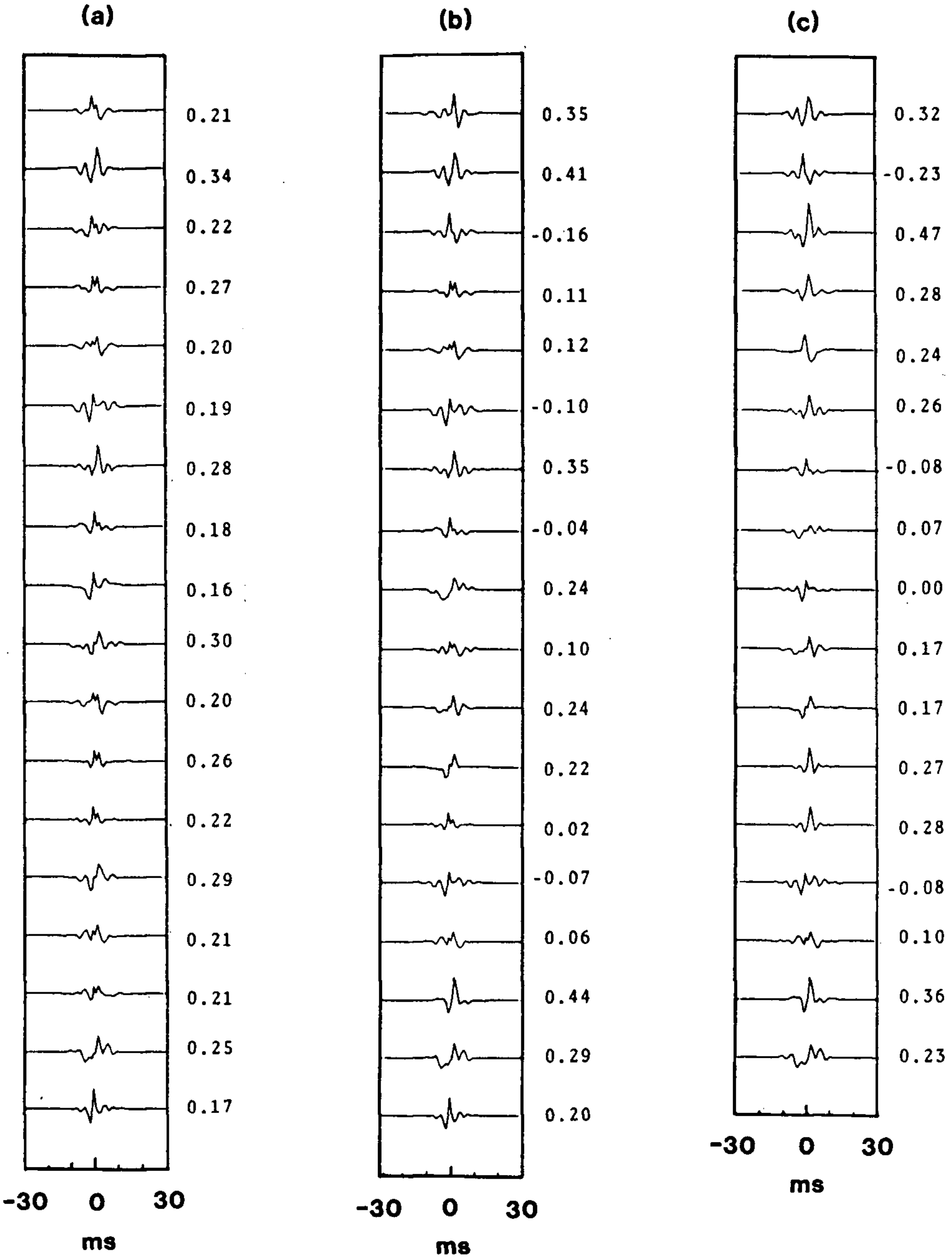


Figure 25 - Segment's Cepstra

The cepstra due to the noisy data of Figure 24. Inputs have 5% (a), 10% (b), and 30% (c) noise. The first principal component weights are shown beside the corresponding cepstra.

first P.C. weights and misfits shows that, as the weight's variability increases, the wavelet estimate tends to shift from the first to the second P.C. Note also that, for low

Wavelet Estimate Misfits			
Noise	Ave.	1 P.C.	2 P.C.
5%	39	19	190
10%	18	74	56
30%	75	134	32

Table 1 - Wavelet Misfits

noise (5%), the first P.C. yielded a better wavelet estimate than averaging. Also, for high noise (30%), the second P.C. yielded a better estimate than averaging. At an intermediate noise level, the P.C. method yielded a worse result than averaging.

It may seem paradoxical that, for the averaged estimate, the misfit is lower for 10% noise than for 5% noise. However, there are many factors to be considered. Segmentation is affected by noise, and cepstral windowing may affect the noise as well as the wavelet. Also, for this particular wavelet, setting the phase to zero above some frequency may affect the noise's relation to the wavelet. These do not detract from the fact that, for a fixed noise level, the P.C. estimates may be compared to the average estimate. One must be cautioned against inferring that adding noise to an input will, in general, improve the estimate.

5.2.6 Low Pass Inputs

In previous examples the homomorphic transform was stabilized by setting the phase to zero above some angle ω_1. It may seem obvious that, having admitted ignorance of this phase, we should do likewise with the magnitude. This would prevent the shifting of energy above ω_1 to zero phase. For cepstral averaging this is a logical thing to do. However, this will induce a common signal in the cepstra which may adversely affect the P.C. method of analysis. Let us examine this further.

Say a time sequence has a Fourier transform with magnitude $M(\omega)$. Setting $M(\omega)$ to a small value above $\omega=\omega_1$ is equivalent to setting $\log[M(\omega)]$ to a large negative value. This may be done in two steps. First, multiply $\log[M(\omega)]$ by a boxcar function (i.e. unity on $0<\omega<\omega_1$ and zero on $\omega_1\leq\omega\leq\pi$). Second, add a bandpass function which is zero on $0<\omega<\omega_1$ and a large negative value, say A, on $\omega_1\leq\omega\leq\pi$. This procedure is shown graphically in Figure 26. The Fourier transform of the boxcar is a sinc function and that of the bandpass function is a cosine modulated sinc function with peak value A, call it $A\cdot S(n)$ (Bracewell, 1965). In the quefrency domain this is equivalent to convolution with a sinc function and addition of $A\cdot S(n)$. A becomes larger as $M(\omega)$ is set to smaller values above ω_1. Figure 27 shows the cepstrum of our wavelet with the log magnitude set to various values above $\omega_1 = 0.62\cdot\pi$. The values were derived by subtracting 13, 10, 7, 3 and 0 from the original value at ω_1. Note the dominance of $A\cdot S(n)$ as A

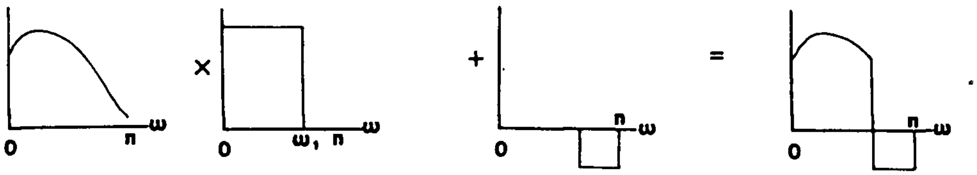


Figure 26 - Low Pass Signals

The energy above some angle may be set to a small constant by setting the log magnitude to a large value. This can be done by multiplying the log magnitude by a boxcar function, then adding a bandpass function.

becomes large. (The zero quefrency point has been set to zero.)

This effect and the arbitrariness of A suggest that this is not an appropriate procedure when P.C. analysis is to be used on the cepstra.

5.3 An Alternate Solution

We have seen that, in the presence of additive noise, the use of P.C.s on cepstra is inappropriate. However, the principal component method may still be utilized.

The homomorphic transform allows for wavelet estimation by low quefrency windowing. Exponential weighting of inputs and de-weighting of outputs may improve the estimate (Buhl, et al, 1974). A wavelet estimate can be obtained from each input sequence by this method. This yields a suite of estimates from which the most common wavelet can be extracted by principal components.

This is illustrated in Figure 28 using the noisy multiple sequence data generated previously. An exponential weighting

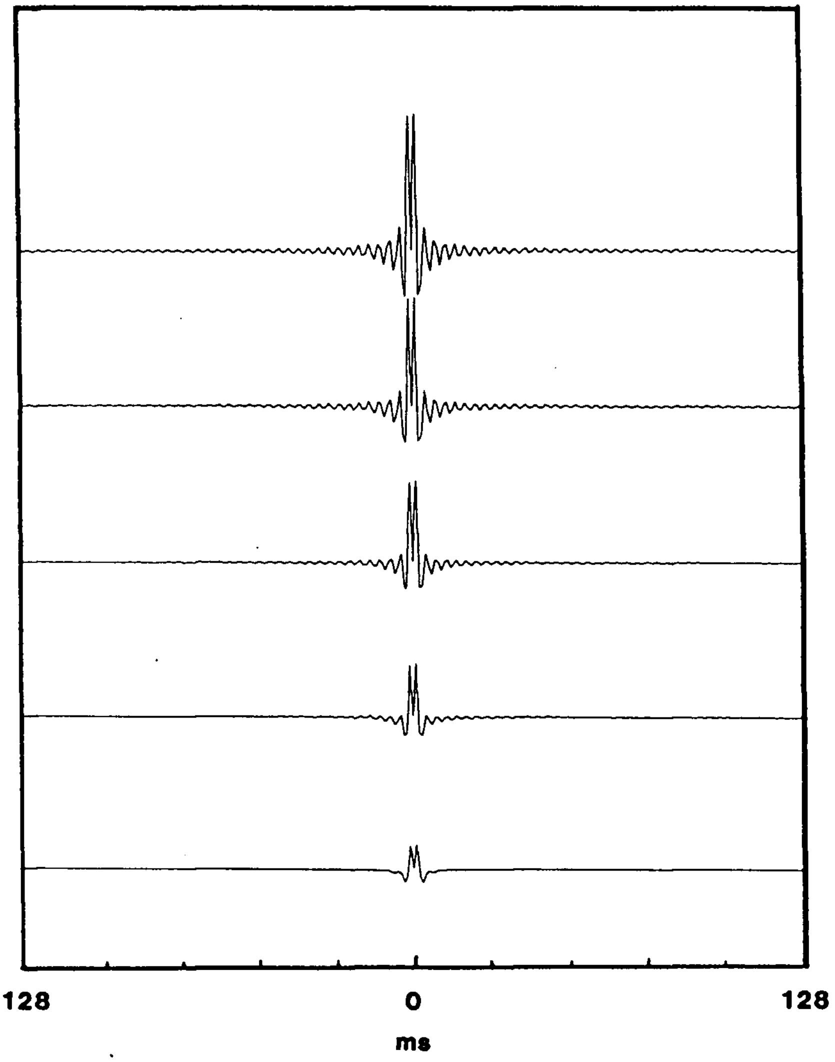


Figure 27 - Out of Band Energy

Cepstra due to the wavelet of Figure 9 but setting the log magnitude constant above the angle $\omega_1=0.62\cdot\pi$. This constant has been set to the original value minus 13, 10, 7, 3, and 0 (top to bottom).

factor of 0.98 and a 5 ms cepstral window were used to produce wavelet estimates. These estimates were normalized to unit variance and shifted to align their envelope peaks. The most common estimate may be defined as the average or the first P.C. Their misfits are 163 and 20 respectively which shows a

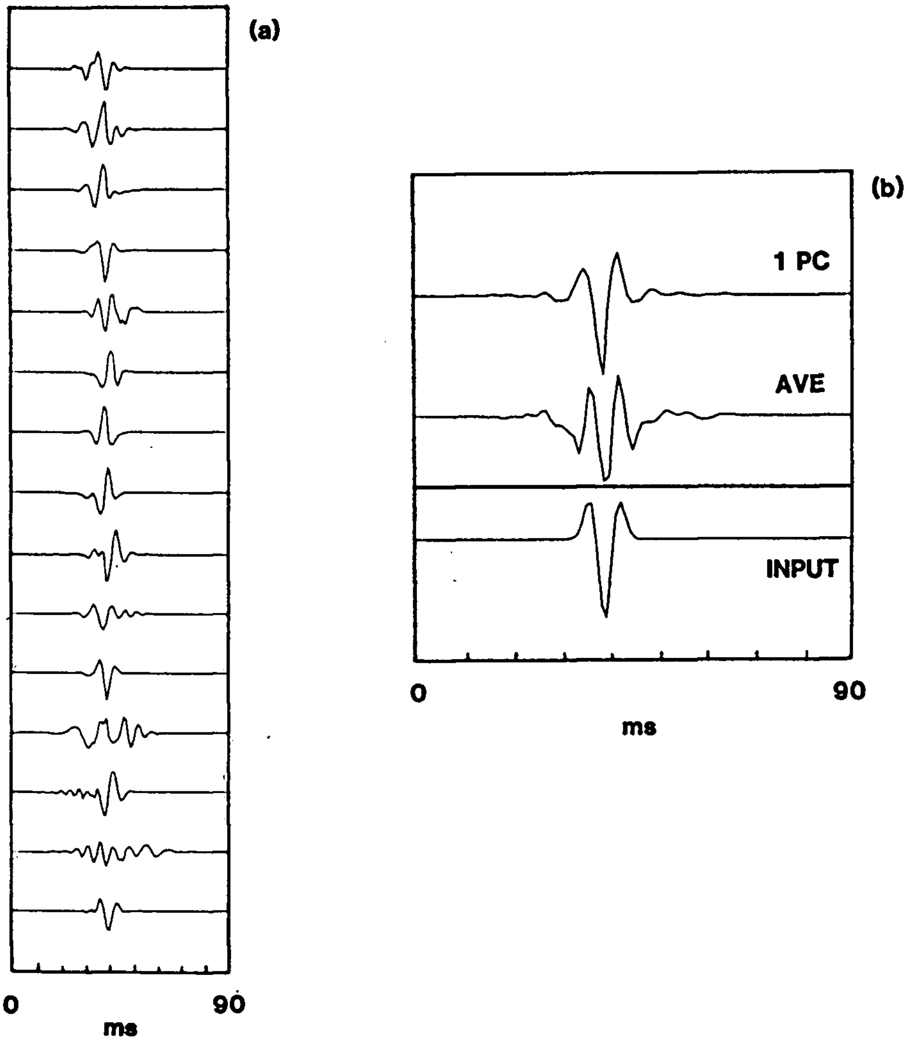


Figure 28 - Time Domain Principal Components

Wavelets estimated from the data of Figure 14 using cepstral windowing (a). The first principal component (PC) and average (AVE) of these estimates (b). Input wavelet is shown at the bottom.

significant improvement of the P.C. method over averaging. These misfits compare with 20, 280 and 16 for the cepstral average and first two cepstral P.C.s. Thus this technique can yield an estimate which is comparable to cepstral averaging.

5.4 Practical Considerations

The previously discussed wavelet estimation scheme has several parameters which must be specified. These include the segmentation window shape and length, cepstral window shape and length and exponential weighting factor. Specification of these must be done carefully, on the basis of a priori information and experience.

The segmentation window should be slowly varying. Its length is a trade-off between wavelet distortion and sequence length. Shorter sequences have greater distortion but tend to yield more reliable phase unwrapping. The cepstral window should probably not have sharp discontinuities. Its length can be adjusted by monitoring the corresponding estimates. Exponential weighting is again a trade-off between distorting the wavelet's cepstrum and separating it from the impulse cepstra.

Due to the statistical nature of the scheme, more data will tend to improve the estimate.

This technique is not user independent. Its flexibility and power can be fully realized only with careful and judicious use.

5.5 Summary

The application of the principal component method for wavelet estimation has been demonstrated. The method may be used in the quefrency domain or in the time domain after cepstral windowing. Additive noise may induce a common signal in the cepstra. Setting out of band energy to a small value will add a common term to the cepstra. In the presence of these common signals the P.C. method is more appropriately applied in the time domain.

VI. INVERTIBILITY OF THE HOMOMORPHIC TRANSFORM

A recently published note suggests that, due to phase unwrapping, the homomorphic transform may not be invertible (Jin and Rogers, 1983). This note is supported by examples illustrating their failure to obtain a successful inversion. This failure should not occur. To set our minds at ease these same examples are presented here demonstrating their successful forward and inverse transformation.

Except for inputs with zeros on the unit circle, if the forward transform can be calculated, so can its inverse, to within the precision of the computation. If the phase is unwrapped incorrectly an incorrect cepstrum will be calculated. However, even if this occurs, the inverse transform will still yield the original input. This follows from the fact that phase unwrapping adds integer multiples of 2π to the principal phase (except the method of McGowan and Kuc (1983) which adds multiples of π). These are transparent to the inverse operation of exponentiation. (Note that the whole inverse system is realized by this exponentiation and Fourier transforms, which are invertible.)

The first example consists of two impulses, of magnitude 2000 and 1999, separated by 20 points (Figure 29). The sequence is extended to 1024 points with zeros. Its cepstrum and the cepstrum's inverse are shown. Note the successful inversion. (Calculations were performed with single precision arithmetic in a FORTRAN program.) The cepstrum is minimum delay and exhibits some aliasing.

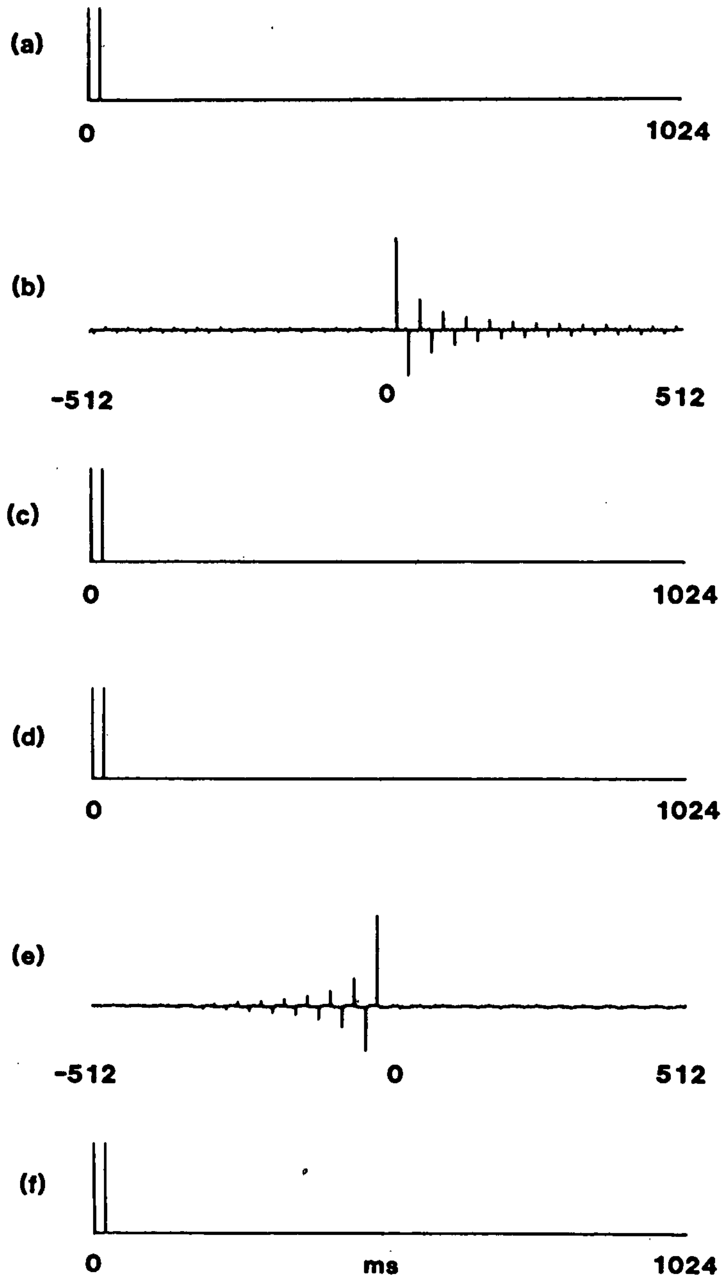


Figure 29 - Inversion of Cepstra

An input (a) consists of two impulses of amplitude 2000 and 1999. Its cestrum (b) and the cepstrum's inverse (c) show the return of the original input. (d) is the input sequence of (a) with additive noise. (e) is the cepstum of (d) and (f) is the inverse of (e).

The effect of additive noise with a standard deviation of 5 is also shown. This noise level is too low to be perceptible in the plots. It has, however, changed the relative magnitude of the impulses such that the sequence is nearly minimum delay. Again the cepstrum inverted correctly.

In the second example, the first impulse has a magnitude of 2000 while the second is reduced from this by 33% (Figure 30). The input, cepstrum and the cepstrum's inverse are shown. When noise having a standard deviation of 20 is added, the cepstrum's appearance changes dramatically. However, it still inverts to the original input.

The results of Jin and Rogers (1983) are reproduced in Figures 31 and 32 for comparison.

All of the inputs dealt with in the production of this thesis were invertible. We can be satisfied that the results of Jin and Rogers are anomalous.

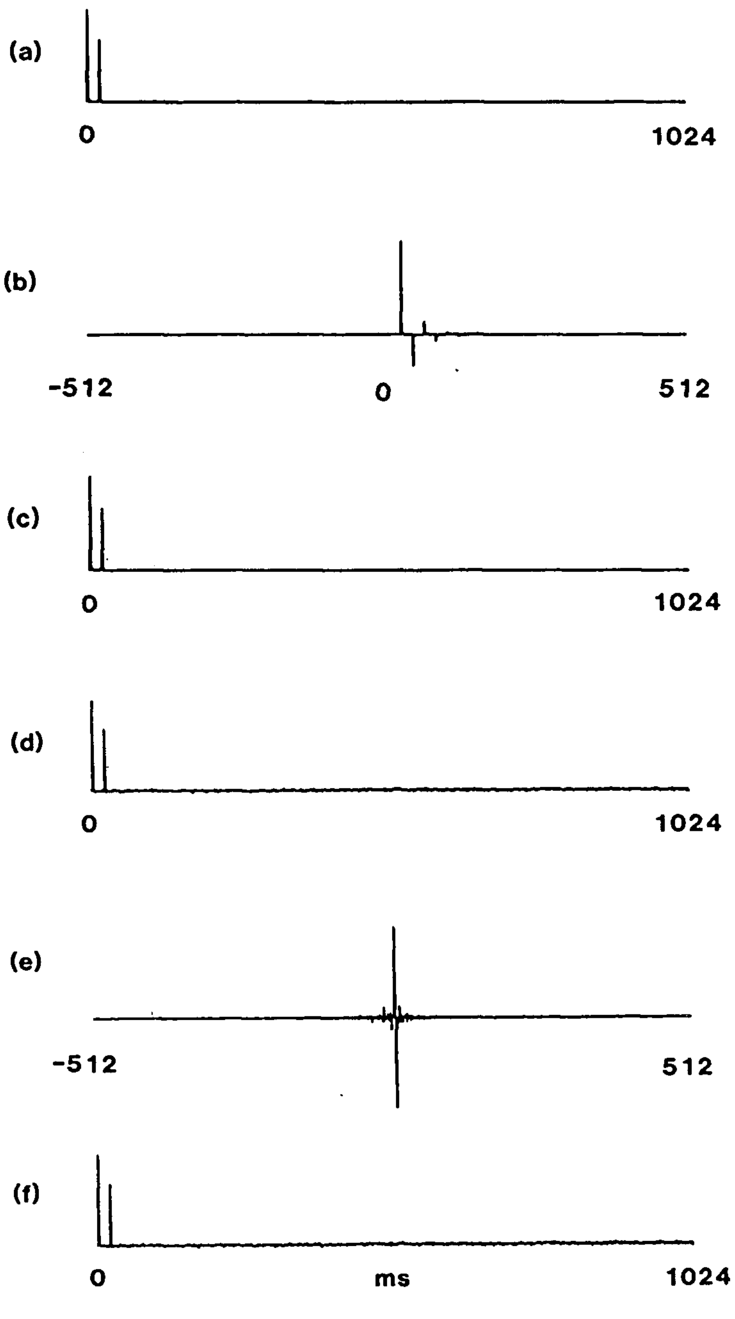
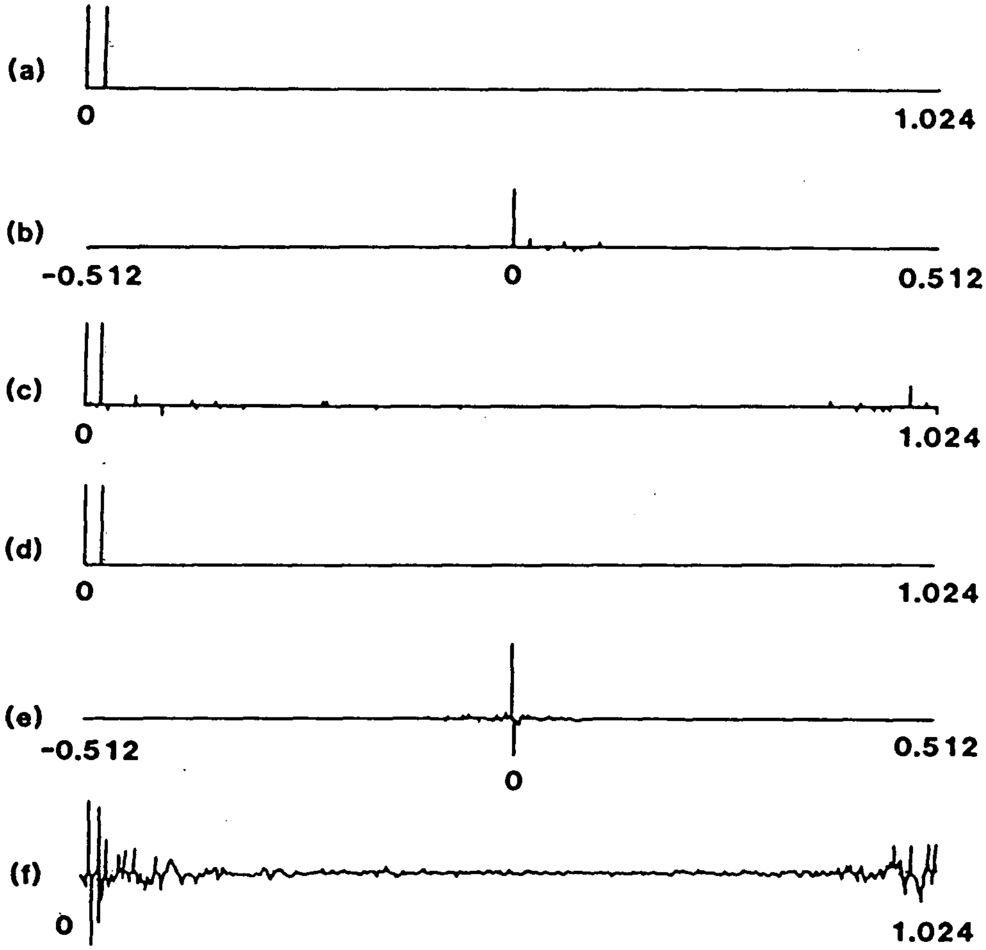


Figure 30 - Inversion of Cepstra

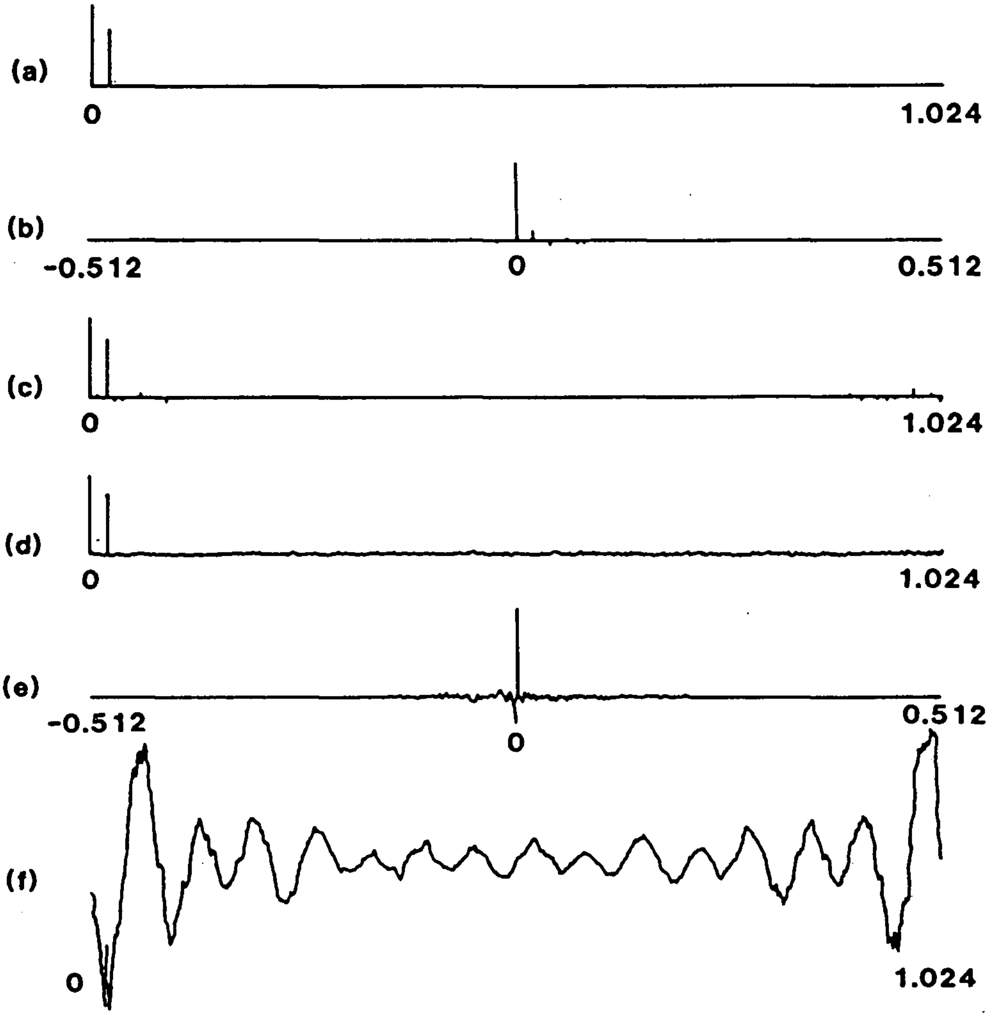
An input (a) consists of two impulses with different amplitude. Its cepstrum (b) and the cepstrum's inverse (c) show the return of the input. (d) is the input (a) with additive noise. (e) is the cepstrum of (d) and (f) is the inverse of (e).



(a) Signal with two spikes of nearly identical amplitudes, free of noise. (b) Complex cepstrum of the signal in (a). (c) Returned signal of (a). (d) Signal of (a) with noise added. (e) Complex cepstrum of the signal in (d). (f) Returned signal of (d).

Figure 31 - Inversion of Cepstra

The results of Jin and Rogers (1983). This compares with Figure 29.



(a) Signal with two spikes whose amplitudes are quite different, free of noise. (b) Complex cepstrum of the signal in (a). (c) Returned signal of (a). (d) Signal of (a) with noise added. (e) Complex cepstrum of the signal in (d). (f) Returned signal of (d).

Figure 32 - Inversion of Cepstra

The results of Jin and Rogers (1983). This compares with Figure 30.

VII. DISCUSSION AND CONCLUSIONS

The wavelet estimation problem has been formulated in terms of multi-channel common signal estimation. This does not require restrictive assumptions about the wavelet's phase. The formulation includes a homomorphic transform to map from a convolutional to an additive (cepstral) space. Realization of the transform requires the computation of a Fourier transform's continuous phase.

In the additive space the problem is one of common signal estimation. Properly formulated, the method of principal components yields an optimal estimate which can be compared to averaging.

The successful application of this technique has been demonstrated for noise free data. In the noisy data case principal components may yield a poorer estimate than averaging. This occurs because noise may induce a common signal in the cepstral space. Principal components then estimate this noise-induced signal instead of the wavelet cepstrum. If an attempt is made to reduce noise effects by setting out of band energy to a small value, another common signal is added to the cepstra.

Thus, for noisy data, we are led to an alternate solution. A wavelet is estimated from each input channel by exponential weighting and cepstral windowing. Principal components are then used to extract the most common estimate from this suite of estimates.

APPENDIX A - THE Z-TRANSFORM

The z-transform of a sequence x(n) is defined as

$$X(z) = \sum_{n=-\infty}^{\infty} x(n)\cdot z^{-n} \qquad (1)$$

where z is a complex number.

The inverse z-transform is defined as

$$x(n) = 1/2\pi j \oint_{C} X(z)\cdot z^{n-1}\,dz \qquad (2)$$

where C is a counterclockwise closed contour in the region of convergence of X(z), encircling the origin in the z-plane. Together, (1) and (2) form a z-transform pair.

If, in (1) and (2), C is taken as a circle centred on the origin, $z=r\cdot\exp[j\omega]$, on $-\pi<\omega<\pi$, then (1) and (2) become

$$X(r\cdot\exp[j\omega]) = \sum_{n=-\infty}^{\infty} [x(n)\cdot r^{n}]\cdot\exp[-j\omega] \qquad (3)$$

$$x(n)\cdot r^{-n} = 1/2\pi \int_{-\pi}^{\pi} X(r\cdot\exp[j\omega])\cdot\exp[j\omega n]\,d\omega \qquad (4)$$

These imply that $X(r\cdot\exp[j\omega])$ is the discrete time Fourier transform of $x(n)\cdot r^{-n}$. Thus the Fourier transform can be used to evaluate the z-transform on a circle centred on the origin. If r=1, equations (3) and (4) become a Fourier Transform pair. Setting $r\neq1$ is referred to as exponential weighting (Oppenheim and Schafer, 1975).

APPENDIX B - FOURIER TRANSFORMS AS CHEBYSHEV POLYNOMIALS

The Discrete Time Fourier Transform (DTFT) can be written in terms of Chebyshev polynomials of the second kind (McGowan and Kuc, 1982).

Consider a sequence x(n) where x(n)=0 for n<0 and n>N. The DTFT of x(n) is

$$F(\omega) = \sum_{n=0}^{N} x(n)\cdot\exp[-jn\omega] \tag{1}$$

or, in terms of real and imaginary parts

$$F(\omega) = \sum_{n=0}^{N} x(n)\cdot\cos(n\omega) - j\cdot \sum_{n=0}^{N} x(n)\cdot\sin(n\omega) \tag{2}$$

Chebyshev polynomials of the first kind, as a function of cos(ω), are defined as (Snyder, 1966)

$$T(n,\omega) = \cos(n\omega) \tag{3}$$

and those of the second kind as

$$U(n,\omega) = \sin[(n+1)\omega]/\sin(\omega) \tag{4}$$

There is a recursion relation betwen (3) and (4):

$$T(n,\omega) = [U(n,\omega) - U(n-2,\omega)]/2 \tag{5}$$

From equations (2) and (3) we write

$$F(\omega) = \sum_{n=0}^{N} x(n)\cdot T(n,\omega) - j\cdot \sum_{n=0}^{N} x(n)\cdot \sin(n\omega) \tag{6a}$$

Defining R = Real{F} amd I = Imaginary{F}, we write (6a) as

$$F = R + j\cdot I \tag{6b}$$

Consider the real part of (6):

$$R = \sum_{n=0}^{N} x(n)\cdot T(n,\omega) \tag{7}$$

Dropping the argument ω and using equation (5), (7) becomes

$$R = \{ \sum_{n=0}^{N} x(n)\cdot U(n) - \sum_{n=0}^{N} x(n)\cdot U(n-2)\}$$

or, using the defintition in (4),

$$R = \{ \sum_{n=0}^{N} x(n)\cdot \sin[(n+1)\omega] - \sum_{n=0}^{N} x(n)\cdot \sin[(n-1)\omega]\}/2\sin(\omega)$$

Rearranging terms and noting that $-x(0)\cdot\sin(-\omega) = x(0)\cdot\sin(\omega)$ and $x(1)\cdot\sin(0) = 0$, yields

$$\begin{aligned} R = \{&[2\cdot x(0)-x(2)]\cdot\sin(\omega) \\ &+ \sum_{n=2}^{N-1} [x(n-1) - x(n+1)]|\sin(n\omega) \\ &+ [x(N-1)]\cdot\sin(N\omega) + [x(N)\cdot\sin[(N+1)\omega]\}/2\sin(\omega) \end{aligned}$$

or, bringing the denominator inside the sum and using (4) yields

$$R = \sum_{n=0}^{N} a(n)\cdot U(n,\omega) \tag{8}$$

where

$$a(0) = x(0) - x(2)/2$$

$$a(n) = [x(n)-x(n+2)]/2 \qquad n=1,...N-2$$

$$a(N-1) = [x(N-1)]/2$$

$$a(N) = x(N)/2$$

Now consider the imaginary part of (6):

$$I = -\sum_{n=0}^{N} x(n)\cdot \sin(n\omega) \tag{9a}$$

or, as the first term is zero,

$$I = -\sum_{n=0}^{N-1} x(n+1)\sin[(n+1)\omega] \tag{9b}$$

Setting

$$b(n) = -x(n+1) \tag{10}$$

in (8) yields

$$I = \sum_{n=0}^{N-1} b(n)\cdot \sin[(n+1)\omega]$$

$$I = \sin(\omega) \cdot \sum_{n=0}^{N-1} \{ b(n) \cdot \sin[(n+1)\omega] / \sin(\omega) \}$$

$$I = \sin(\omega) \cdot \sum_{n=0}^{N-1} b(n) \cdot U(n,\omega)$$

where we have used the definition (4). Therefore we write (1) in terms of Chebyshev polynomials of the second kind as

$$F(\omega) = \sum_{n=0}^{N} a(n) \cdot U(n,\omega) + j \cdot \sin(\omega) \cdot \sum_{n=0}^{N-1} b(n) \cdot U(n,\omega)$$

where the a(n) and b(n) are given by equations (8) and (10) respectively.

APPENDIX C - GENERATION OF A STURM SEQUENCE

This appendix illustrates a method of generating a Sturm sequence from two Chebyshev polynomials of the second kind (McGowan and Kuc, 1982). The notation is designed to allow easy transcription to a computer program.

Chebyshev polynomials of the second kind, U(n), are defined in appendix B and have the recursion relation (Snyder, 1966)

$$U(n)\cdot U(1) = U(n+1) + U(n-1) \tag{1}$$

We start with the two polynomials

$$P(0) = \sum_{n=0}^{N} a(0,n)\cdot U(n) \tag{2}$$

$$P(1) = \sum_{n=0}^{N} a(1,n)\cdot U(n) \tag{3}$$

where a(k,n) refers to the n-th coefficient of the k-th polynomial P(k). The Sturm sequence is a sequence of polynomials of decreasing degree, generated recursively from P(0) and P(1) by Euclid's algorithm (Marden, 1966)

$$P(k+1) = Q(k)\cdot P(k) - P(k-1) \tag{4}$$

The general polynomial can be written as

$$P(k) = \sum_{n=0}^{N-k} a(k,n)\cdot U(n) \tag{5}$$

and the final polynomial as

$$P(N) = a(N,0)\cdot U(0) \tag{6}$$

In the recursion (4), let us define the Q(k) as the first degree polynomial

$$Q(k) = d(k,0)\cdot U(0) + d(k,1)\cdot U(1) \tag{7}$$

where the d are constants.

For clarity let us begin with an example by generating P(2). Inserting (2), (3) and (7) into (4) yields

$$\begin{aligned} P(2) = \{ \sum_{n=0}^{N-1} a(1,n)\cdot U(n)\}\cdot\{d(1,0)\cdot U(0)+d(1,1)\cdot U(1)]\} \\ -\{ \sum_{n=0}^{N} a(0,n)\cdot U(n)\} \end{aligned}$$

or, expanding,

$$\begin{aligned} P(2) = {} & \{ \sum_{n=0}^{N-1} a(1,n)\cdot d(1,0)\cdot U(n)\}\cdot U(0) \\ & + \{ \sum_{n=0}^{N-1} a(1,n)\cdot d(1,1)\cdot U(n)\}\cdot U(1) \\ & - \{ \sum_{n=0}^{N} a(0,n)\cdot U(n)\} \end{aligned}$$

The relation (1) and the fact that U(0)=1 yield

$$
\begin{aligned}
P(2) = & \sum_{n=0}^{N-1} a(1,n)\cdot d(1,0)\cdot U(n) \\
& + \{a(1,1)\cdot d(1,1)\}\cdot U(0) \\
& + \sum_{n=1}^{N-2} \{a(1,n-1)\cdot d(1,1)+a(1,n+1)\cdot d(1,1)\}\cdot U(n) \\
& + \{a(1,N-2)\cdot d(1,1)\}\cdot U(N-1) \\
& + \{a(1,N-1)\cdot d(1,1)\}\cdot U(N) \\
& - \sum_{n=0}^{N} a(0,n)\cdot U(n)
\end{aligned}
$$

Collecting like terms yields

$$
\begin{aligned}
P(2) = & \{a(1,0)\cdot d(1,0) + a(1,1)\cdot d(1,1) - a(0,0)\} \\
& + \sum_{n=2}^{N-2} \{a(1,n)\cdot d(1,0) + [a(1,n-1)+a(1,n+1)]\cdot d(1,1) \\
& \qquad - a(0,n)\}\cdot U(n) \\
& + \{a(1,N-1)\cdot d(1,0)+a(1,N-2)\cdot d(1,1)-a(0,N-1)\}\cdot U(N-1) \\
& + \{a(1,N-1)\cdot d(1,1)-a(0,N)\}\cdot U(N)
\end{aligned}
$$

We choose Q by finding d(1,0) and d(1,1) such that the terms in U(N) and (N-1) cancel. That is,

$$d(1,1) = a(0,N)/a(1,N-1)$$

$$d(1,0) = \{a(0,N-1) - a(1,N-2)\cdot d(1,1)\}/a(1,N-1)$$

By induction we can write the general term for the k-th polynomial as

$$P(k) = \{a(k-1,0)\cdot d(k-1,0)+a(k-1,1)\cdot d(k-1,1)-a(k-2,0)\} + \sum_{n=1}^{N-k} \{a(k-1,n)\cdot d(k-1,0)+[a(k-1,n-1)+a(k-1,n+1)] - a(k-2,n)\}\cdot U(n)$$

and the final polynomial as

$$P(N) = \{a(N-1,0)\cdot d(N-1,0) + a(N-1,1)\cdot d(N-1,1) - a(N-2,0)\}\cdot U(0)$$

The k-th polynomial can be written as

$$P(k) = \sum_{n=0}^{N-k} a(k,n)\cdot U(n) \qquad k=2,\ldots N$$

where

$$a(k,0) = a(k-1,0)\cdot d(k-1,0)+a(k-1,1)\cdot d(k-1,1)-a(k-2,0) \quad (8a)$$

and, for n>0

$$a(k,n) = a(k-1,n)\cdot d(k-1,0)+[a(k-1,n-1)+a(k-1,n+1)]\cdot d(k-1,1) - a(k-2,n) \quad (8b)$$

where, for k=0,...N-2,

$$d(k+1,1) = a(k,N-k)/a(k+1,N-k-1) \quad (9a)$$

and

$$d(k+1,0) = [a(k,N-k-1)-a(k+1,N-k-2)\cdot d(k+1,1)] / a(k+1,N-k-1) \qquad (9b)$$

The recursion relations (8) and (9) allow the computation of all the P(k), k>0, provided P(0) and P(1) are not relatively prime. If this is the case, the highest degree coefficient of a P(k) is zero and a division by zero in (9) will occur. For this case the next polynomial generated has a degree reduced by more than one. The recursion then skips one iteration and continues for subsequent calculations.

BIBLIOGRAPHY

1. Ahmed, N. and Rao, K.R., 1975, Orthogonal Transforms for Digital Signal Processing: Springer-Verlag, New York.

2. Beaumont, R.A. and Pierce, R.S., 1963, The Algebraic Foundations of Mathematics: Addison-Wesley, U.S.A.

3. Bhanu, B., 1977, Computation of Complex Cepstrum: M.Sc. and E.E. thesis, M.I.T., Massachusetts.

4. Bogert, B.P., Healy, M.J.R. and Tukey, J.W., 1962, The Quefrency Alanysis of Time Series for Echoes: Cepstrum, Pseudo-Autocovariance, Cross-Cepstrum and Saphe Cracking: Proc. Symp. on Time Series Analysis, Ed. M. Rosenblatt, John Wiley and sons, Inc., N.Y.

5. Bracewell, R., 1965, The Fourier Transform and Its Applications: McGraw-Hill, U.S.A.

6. Brigham, E.O., 1974, The Fast Fourier Transform: Prentice-Hall, Inc., New Jersey.

7. Buhl, P., Stoffa, P.L. and Bryan, G.M., 1974, The Application of Homomorphic Deconvolution to Shallow-Water Marine Seismology-Part II: Real Data: Geophysics, v. 39, p. 417-426.

8. Buttkus, B., 1975, Homomorphic Filtering - Theory and Practice: Geophysical Prospecting, v. 23, p. 712-748.

9. Clayton, R.W. and Wiggins, R.A., 1976, Source Shape Estimation and Deconvolution of Teleseismic Bodywaves: Geophys. J.R. astr. Soc., v. 47, p. 151-177.

10. Dhrymes, P.J., 1970, Econometrics, Statistical Foundations and Applications: Springer-Verlag, New York.

11. Gelb, A. et al., 1974, Applied Optimal Estimation: The Analytic Sciences Corporation, M.I.T., Massachusetts.

12. Hall, E.L., 1979, Computer Image Processing and Recognition: Academic Press, Inc., New York.

13. Jin, D.J. and Rogers, J.R., 1983, Homomorphic Deconvolution: Geophysics, v. 48, p. 1014-1016.

14. Kramer, H.P. and Mathews, M.V., 1956, Linear Coding For Transmitting a Set of Correlated Signals: I.R.E. Trans. on Information Theory, v. IT-2, p. 41-46.

15. Lines, L.R. and Ulrych, T.J., 1977, The Old and the New in Seismic Deconvolution and Wavelet Estimation:

Geophysical Prospecting, v. 25, p. 512-540.

16. Lipschutz, S., 1974, Linear Algebra: Schaum's Outline Series, McGraw-Hill, London.

17. Marden, M., 1966, Geometry of Polynomials: American Mathematical Society, Rhode Island.

18. McGowan, R. and Kuc, R., 1982, A Direct Relation Between a Signal Time Series and Its Unwrapped Phase: I.E.E.E. Trans. on A.S.S.P., v. ASSP-30, p. 719-726.

19. Oldenburg, D.W. and Samson, J.C., 1979, Inversion of Interferometric Data From Cylindrically Symmetric, Refractionless Plasmas: J. Opt. Soc. Am., v. 69, p. 927-942.

20. Ooe, M. and Ulrych, T.J., 1979, Minimum Entropy Deconvolution with an Exponential Transformation: Geophysical Prospecting, v. 27, p. 458-473.

21. Oppenheim, A.V., 1965, Superposition in a Class of Nonlinear Systems: Tech. Rept. 432, Research Laboratory of Electronics, M.I.T., Massachusetts.

22. Oppenheim, A.V. and Schafer, R.W., 1975, Digital Signal Processing: Prentice-Hall, Inc. New Jersey.

23. Oppenheim, A.V., Schafer, R.W. and Stockham, T.G., 1968, Nonlinear Filtering of Multiplied and Convolved Signals: Proc I.E.E.E., v. 56, p. 1264-1291.

24. Otis, R.M. and Smith, R.B., 1977, Homomorphic Deconvolution by Log Spectral Averaging: Geophysics, v. 42, p. 1146-1157.

25. Poggiagliolmi, E., Berkhout, A.J. and Boone, M.M., 1982, Phase Unwrapping, Possibilities and Limitations: Geophysical Prospecting, v. 30, p. 281-291.

26. Ready, P.J. and Wintz, P.A., 1973, Information Extraction, S.N.R. Improvement, and Data Compression in Multispectral Imagery: I.E.E.E. Trans. on Communications, v. COM-21, p 1123-1130.

27. Robinson, E.A. and Treitel, S., 1980, Geophysical Signal Analysis: Prentice-Hall, Inc., New Jersey.

28. Schafer, R.W., 1969, Echo Removal by Discrete Generalized Linear Filtering: Tech. Rept. 466, MIT Research Laboratory of Electronics, MIT, Massachusetts.

29. Snyder, M.A., 1966, Chebyshev Methods in Numerical Approximation: Prentice-Hall, Inc., New Jersey.

30. Steiglitz, K. and Dickinson, B., 1982, Phase Unwrapping by Factorization: I.E.E.E. Trans. on A.S.S.P., v ASSP-30, p. 984-991.

31. Stoffa, P.L., Buhl.,P. and Bryan, G.M., 1974, The Appications of Homomrphic Deconvolution to Shallow-Water Marine Seismology-Part I:Models: Geophysics, v. 39, p. 401-416.

32. Strang, G., 1980, Linear Algebra and Its Applications (2 ed.): Academic Press. New York.

33. Taner, M.T. and Sheriff, R.E., 1977, Application of Amplitude, Frequency, and Other Attributes to Stratigraphic and Hydrocarbon Determination: in Seismic Stratigraphy - Applications to Hydrocarbon Exploration, Ed. C.E. Payton, Memoir 26 of A.A.P.G, Tulsa, p. 301-328.

34. Telford, W.M., Geldart, L.P., Sheriff, R.E. and Keys, D.A., 1976, Applied Geophysics: Cambridge University Press, New York.

35. Tribolet, J.M., 1977, A New Phase Unwrapping Algorithm: I.E.E.E. Trans. on A.S.S.P., v. ASSP-25, p. 170-177.

36. Tribolet, J.M., 1979, Seismic Applications of Homomorphic Signal Processing: Prentice-Hall, Inc, New Jersey.

37. Ulrych, T.J., 1971, Application of Homomorphic Deconvolution to Seismology: Geophysics, v. 36, p. 650-660.

38. Ulrych, T.J., Levy, S., Oldenburg, D.W., and Jones, I.F., 1983, Applications of the Karhunen-Loève Transformation in Reflection Seismology: presented at the 53rd Annual Meeting of the S.E.G., Las Vegas.

39. Wiggins, R.A., 1972, The General Linear Inverse Problem: Implications of Surface Waves and Free Oscillations for Earth Structure: Reviews of Geophysics and Space Physics, v. 10, p. 251-285.

40. Wiggins, R.A., 1978, Minimum Entropy Deconvolution: Geoexploration, v. 16, p. 21-35.