AN ABSTRACT OF THE THESIS OF

Roy C. Rathja for the degree of Doctor of Philosophy

in Electrical and Computer Engineering presented on April 22, 1980

Title: Tone Burst Deconvolution

Abstract approved: Redacted for Privacy

James R. Herzog

The suitability of discrete least-squares optimal (Wiener) filters for deconvolution of tone-burst waveforms is studied. It is shown that: (1) such waveforms are not well suited to these filters without preprocessing, (2) the use of exponential weighting results in a significant improvement in filter performance and (3) that "phase-unwrapping" in the z-plane results in a useful procedure for estimating appropriate exponential weighting. Based on these results a procedure is developed and applied to artificial and real data to demonstrate the practicality of such methods.

TONE BURST DECONVOLUTION

by

Roy C. Rathja

A THESIS

submitted to

Oregon State University

in partial fulfillment of
the requirements for the
degree of

Doctor of Philosophy

Completed April 22, 1980

Commencement June 1980

APPROVED:

Redacted for Privacy

Professor of Electrical and Computer Engineering in charge of major

Redacted for Privacy

Head of Department of Electrical and Computer Engineering

Redacted for Privacy

Dean of Graduate School

Date thesis is presented: April 22, 1980

Typed by Shanda L. Smith for Roy C. Rathja

ACKNOWLEDGEMENTS

I would like to take this opportunity to express my sincere appreciation to the many people who have supported me during my development of this thesis. First and foremost is my wife, Marsha, who raised my children, Laura and Eric, with little help from her husband for the last few years. That situation will soon change.

There are many people in the Department of Electrical and Computer Engineering at Oregon State University who have been supportive. I would especially like to thank my major professor, Jim Herzog, and John Saugen for their support, guidance and most of all patience.

A considerable amount of support and the impetus for this thesis was provided by the Naval Undersea Warfare Engineering Station (NUWES) at Keyport, WA under contract N0096-78-0779. Without their financial support and motivation, this thesis never would have been developed.

TABLE OF CONTENTS

LIST OF FIGURES

List of Figures (Cont.)

LIST OF TABLES

TONE BURST DECONVOLUTION

I. INTRODUCTION

1.1 Purpose

It is frequently important to determine the characteristics of a waveform that is composed of a sum of more fundamental waveforms. This is generally posed as a detection and/or estimation problem and the parameters of interest are the time of arrival of the leading edge (epoch detection) and the amplitude of the various fundamental waveforms. The fundamental waveforms are often called wavelets.

The multiple occurrence ofoverlapping wavelets is frequently caused physically by an echo generating structure. The waveform to be analyzed is composed of a number of echos differing only in amplitude and time of arrival.

An application in which this problem frequently occurs is radar and sonar ranging systems. Returns from multiple targets, multiple returns from the same target and multiple transmission paths all lead to waveforms of this type. The most commonly used wavelet in radar and sonar is the tone burst. A tone burst is a finite duration sinusoidal waveform. It is the purpose of this thesis to investigate the application of discrete least-squares optimal (Wiener) filtering for analysis of multiple overlapping tone burst wavelets.

1.2 The Convolution Model

The formulation of this problem in mathematical terms is the purpose of this section. A simplified model will be developed which will closely approximate the (noise free) physical situation.

Since the ultimate goal of the thesis is to develop analysis methods to be implemented on a digital computer, the theory of discrete-time systems will be used extensively [14, pp. 9-74]. In this well known theory, information to be processed is treated as a sequence of numbers rather than as a more classical continuous function. Such a framework results in a more natural representation for computer processing with no loss of generality. Discrete-time signals are defined only for discrete values of time and continuous-time signals may be converted to discrete-time by measuring their amplitude at appropriate time intervals.

Certain characteristics of the signals to be analyzed will be assumed. These assumptions significantly simplify the analysis to follow and are reasonable for the class of signals to be studied. It will be assumed that a discrete-time linear time-invariant model is appropriate. In general, the wavelet composing the fundamental waveform will be denoted by W. For the specific case of a tone burst, we have a sequence with values given by:

$$w(kT) = \begin{cases} \cos\,(\omega kT + \theta) & 0 \leq k \leq n \\ 0 & \text{otherwise} \end{cases} \tag{1.1}$$

where: k is the index of the value in the sequence

T is the time between values in the sequence

ω is the angular frequency of the cosinusoid

θ is the phase angle of the cosinusoid

Throughout the thesis, upper case letters will represent a sequence of data and lower case letters will represent a single value from a sequence.

For simplicity of notation and with no loss of generality, the sample interval, T, will be omitted from the following developments. The composite signal to be analyzed is a sum of several basic wavelets and will be denoted by S. For the tone burst case we have:

$$S(k) = \sum_{i=1}^{N} b(i) \cos (\omega[k-\tau(i)] + \Theta) \qquad (1.2)$$

where b(i) is the amplitude and $\tau(i)$ is an integer specifying the delay of the wavelet in multiples of T. In this framework, the problem to be studied is the determination of the number of wavelets, N, in the composite signal S and the estimation of the individual wavelet amplitudes, b(i) and delays, $\tau(i)$.

The model of equation (1.2) provides a clear idea of the problem to be studied, but it is not in the most convenient mathematical form. It is possible to rewrite (1.2) as a discrete convolution of the basic wavelet, W, and a sequence often referred to as the reflector series, R, as follows:

$$s(k) = \sum_{j=o}^{m} r(j)w(k-j) \qquad k=0,1,\ldots n+m \qquad (1.3)$$

W(k) is defined as in (1.1) and R(k) is a sequence of length m+1 of N spikes of amplitude b(i) and time of occurrence $\tau(i)$.

$$R(k) = \sum_{i=1}^{N} b(i)\ \delta[k-\tau(i)] \qquad (1.4)$$

and

$$\delta[k-\tau(i)] = \begin{cases} 1 & k=\tau(i) \\ 0 & \text{otherwise} \end{cases} \qquad (1.5)$$

The convolution of (1.3) is more conveniently represented as:

$$S(k) = R(K)*W(k) \tag{1.6}$$

where the linear operator * represents convolution. Equation (1.6) is the basic data model. Figure 1.1 shows typical functions for S(k), R(k) and W(k).

It is important to recognize the significance of expressing the model in the form of equation (1.6) rather than equation (1.2). In (1.6), the reflector series, R(k), contains complete information concerning the amplitude and time of arrival of the various wavelets making up S(k). Furthermore, this information has been linearly separated from the basic wavelet characteristics. This linear separation allows linear methods to be used to remove W(k) from S(k) leaving an estimate of R(k).

1.3 Deconvolution

It is a conceptually simple matter to determine R(k), given S(k), as in (1.6) by applying a filter H(k) as follows:

$$S(k)*H(k) = R(k)*W(k)*H(k) \tag{1.7}$$

The right hand side of (1.7) may be written:

$$S(k)*H(k) = R(k)*[W(k)*H(k)] \tag{1.8}$$

The problem may now be seen to be: Determine H(k) such that

$$W(k)*H(k) = \delta(k) \tag{1.9}$$

where $\delta(k)$ is defined by (1.5).

If H(k) can be determined, then substituting (1.9) into (1.8) yields:

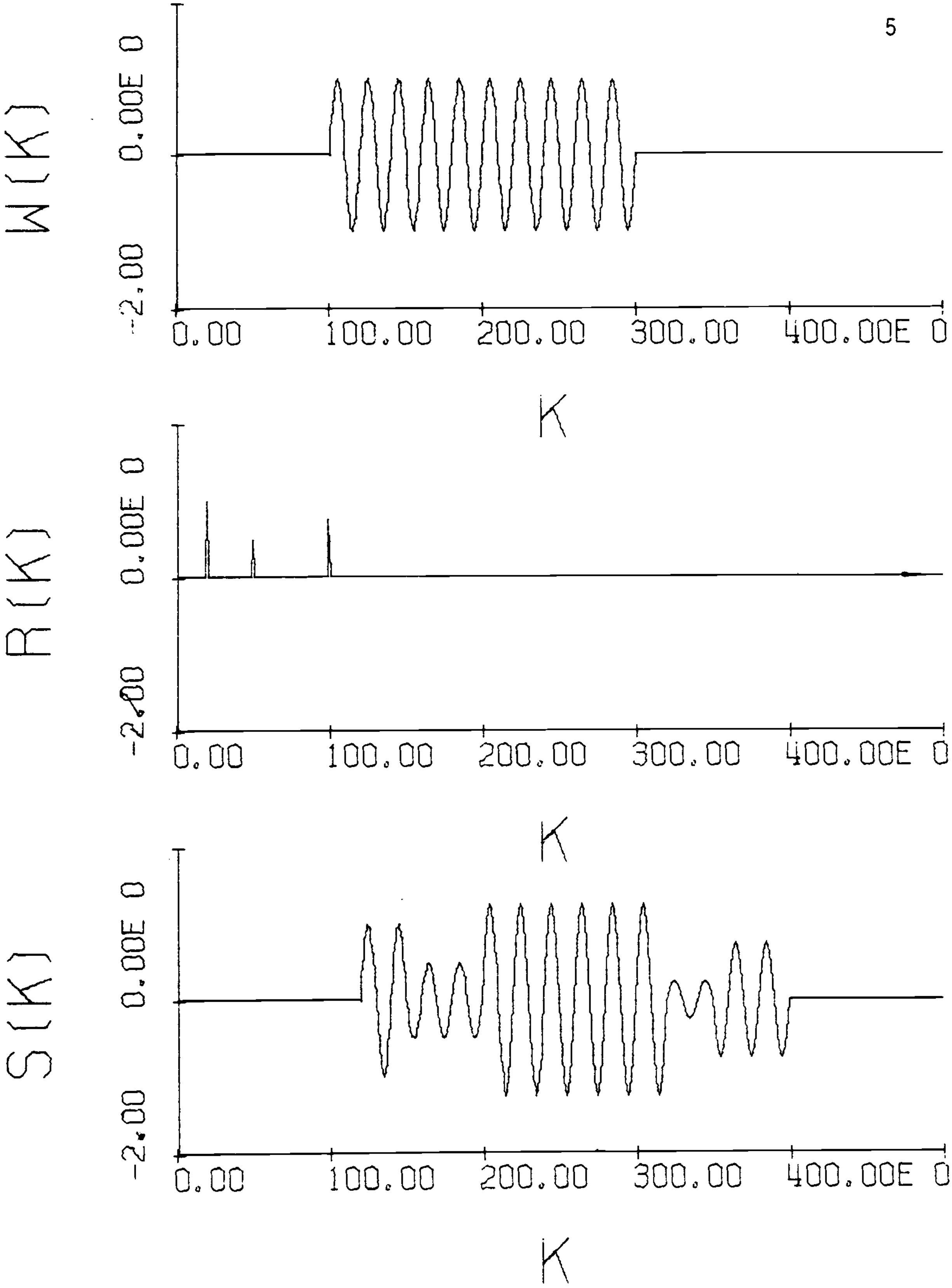


Figure 1.1

Typical Functions

$$S(k)*H(k) = R(k)*[\delta(k)] \tag{1.10}$$

Since convolution of a function with a spike is just the original function, we have the desired result.

$$S(k)*H(k) = R(k) \tag{1.11}$$

The application of a convolutional filter as in equation (1.7) for the purpose of removing a component of a signal is commonly called deconvolution. The filter to be designed, H(k),is an inverse filter.

1.4 Scope

Least-squares deconvolution is the topic of Chapter II. This chapter is a review of the mathematical development of the discrete time domain formulation of least-squares (Wiener) filters. Chapter II is primarily background material for the development of later chapters and notation and terminology are introduced. Additionally, the fundamental deficiencies of this filtering method are discussed. In Chapter II, it is shown that, for many signals, it is not possible to determine a convolutional filter that satisfies equation (1.9). In general (1.9) must be rewritten:

$$H(k)*W(k) = \delta(k-\beta) \tag{1.12}$$

That is, delay must be added to the required filter output.

The purpose of Chapter III is to study the characteristics of tone burst signals which affect their suitability for Wiener deconvolution. In the beginning of the chapter, several attributes of signals which may be used to study this suitability are presented.

Next, two methods are presented which transform unacceptable signal types to those more suited to Wiener deconvolution. Section 3.4 presents exponential weighting and intuitively discusses why this technique has proven useful. The last two sections of Chapter III deal with analytic and simulation results in the study of tone burst characteristics. The result of these sections is that tone burst waveforms are not well suited to Wiener deconvolution. However, it is shown that an exponentially weighted tone burst is suitable for this type of analysis.

The purpose of Chapter IV is to study in detail the influence of exponential weighting as a preprocess to Wiener deconvolution. The effect of exponential weighting is shown for analytic inverse filters and the conditions for improved filter performance are derived. Additionally, a computational method for estimating the optimal weighting factor is presented.

Chapter V presents computational results for artificial and real data that demonstrate the concepts developed in Chapters III and IV. Chapter VI is a summary of the new results from this work and also includes suggestions for future research.

II. LEAST-SQUARES DECONVOLUTION

2.1 Introduction

The early concepts of least-squares filtering were developed by Norbert Wiener [1] in the 1940s and applied to the theory of statistical communication. These concepts were successfully adapted and extended during the 1960s to digital processing of seismic recordings by Enders A. Robinson and Sven Treitel [2-6]. Wiener filters have now been successfully applied in several fields of digital signal processing including image [7, pp. 206-210] and speech processing. This section develops the mathematical basis for understanding discrete Wiener filtering.

2.2 Wiener Shaping Filters

A Wiener shaping filter is a linear filter that, when applied to a given input sequence, will (with minimum squared error) generate a desired output. The use of these filters is a two-step process as shown in Figure 2.1. The estimation of the basic wavelet, W', is in some cases a difficult problem which will not be addressed in this thesis. It will be assumed that a good estimate of W is available. In many cases W is directly available, while in others it must be estimated [8, pp. 222-241].

Of primary interest will be a special case of this general filtering technique - namely that in which a spike is the required output. The general digital shaping filter will be developed and then reduced to the spike case.

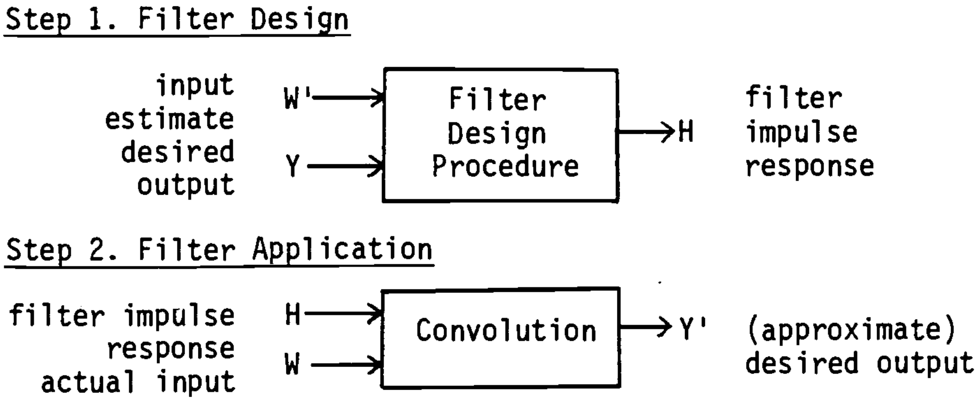


Figure 2.1

The discrete time domain formulation of the general Wiener filter design can be developed as follows. Given an input W which is to be transformed into an output Y, design a filter with finite impulse response H which minimizes in the least squares sense the error between the desired output Y and the attainable output Y'.

Let $H = [h(0),h(1),...,h(m)]$	filter impulse response
$W = (w(0),w(1),...,w(n)]$	filter input
$Y = (y(0),y(1),...,y(n+m)]$	desired filter output
$Y' = (y'(0),y'(1),...,y'(n+m)]$	actual filter output

Then

$$H * W = Y'$$

where H * W is the discrete convolution of H and W defined by:

$$\sum_{j=0}^{m} h(j)w(i-j) = y'(i), \quad i=0,1,...,n+m \qquad (2.1)$$

It is required that the following error be minimized.

$$I = \sum_{i=0}^{m+n} [y(i) - y'(i)]^2 \qquad (2.2)$$

Substituting (2.1) into (2.2) yields:

$$I = \sum_{i=0}^{m+n} [y(i) - \sum_{j=0}^{m} h(j)w(i-j)]^2 \tag{2.3}$$

The error is minimized if each of the partial derivatives, with respect to the filter impulse response coefficients, are zero.

$$\frac{\partial I}{\partial h(k)} = \sum_{i=0}^{m+n} 2[y(i) - \sum_{j=0}^{m} h(j)w(i-j)]w(i-k) = 0 \tag{2.4}$$

where k = 0, 1, ..., m. Expanding (2.4) yields:

$$- \sum_{i=0}^{m+n} y(i)w(i-k) + \sum_{i=0}^{m+n} [\sum_{j=0}^{m} h(j)w(i-j)]w(i-k) = 0 \tag{2.5}$$

Rearranging the double summation leads to:

$$\sum_{j=0}^{m} h(j) \sum_{i=0}^{m+n} w(i-j)w(i-k) = \sum_{i=0}^{m+n} y(i)w(i-k) \tag{2.6}$$

It is not difficult to recognize

$$\sum_{i=0}^{m+n} w(i-j)w(i-k) = R_{ww}(k-j) \tag{2.7}$$

as the autocovariance of W for lag k-j and

$$\sum_{i=0}^{m+n} y(i)w(i-k) = R_{wy}(k) \tag{2.8}$$

as the cross-covariance of W and Y for lag k. Substitution of (2.7) and (2.8) into (2.6) yields:

$$\sum_{j=0}^{m} h(j)R_{ww}(k-j) = R_{wy}(k) \qquad k = 0, 1, \ldots, m \tag{2.9}$$

This set of m equations in m unknowns is called the normal equations [3]. They may be written in matrix notation to more clearly display their form.

$$\begin{bmatrix} R_{ww}(0) & R_{ww}(1) & R_{ww}(2) & \dots & R_{ww}(m) \\ R_{ww}(1) & R_{ww}(0) & R_{ww}(1) & \dots & R_{ww}(m-1) \\ R_{ww}(2) & R_{ww}(1) & R_{ww}(0) & \dots & R_{ww}(m-2) \\ \cdot & & & & \\ \cdot & & & & \\ \cdot & & & & \\ R_{ww}(m) & R_{ww}(m-1) & R_{ww}(m-2) & .. & R_{ww}(0) \end{bmatrix} \bullet \begin{bmatrix} h(0) \\ h(1) \\ h(2) \\ \cdot \\ \cdot \\ \cdot \\ h(m) \end{bmatrix} = \begin{bmatrix} R_{wy}(0) \\ R_{wy}(1) \\ R_{wy}(2) \\ \cdot \\ \cdot \\ \cdot \\ R_{wy}(m) \end{bmatrix} \qquad (2.10)$$

This set of equations can be solved by a standard matrix inversion technique or it is possible to take advantage of the Toplitz character of the autocovariance matrix to efficiently solve these equations for the filter impulse response H. A recursive method due to Levinson [9] significantly reduces this computation.

It is also possible to derive a simple measure of the error for a given filter implementation. From (2.3), the error measure is given by:

$$I = \sum_{i=0}^{m+n} [y(i) - \sum_{j=0}^{m} h(j)w(i-j)]^2 \qquad (2.11)$$

Expanding and introducing a dummy variable yields:

$$I = \sum_{i=0}^{m+n} [y(i)^2 - 2y(i) \sum_{j=0}^{m} h(j)w(i-j)$$

$$+ \sum_{j=0}^{m} h(j)w(i-j) \sum_{k=0}^{m} h(k)w(i-k)] \qquad (2.12)$$

$$I = \sum_{i=0}^{m+n} y(i)^2 - 2 \sum_{j=0}^{m} h(j) \sum_{i=0}^{m+n} y(i)w(i-j)$$

$$+ \sum_{j=0}^{m} h(j) \sum_{k=0}^{m} h(k) \sum_{i=0}^{m+n} w(i-j)w(i-k) \qquad (2.13)$$

Using the expressions for the autocovariance and cross-covariance from equations (2.7) and (2.8) leads to:

$$I = R_{yy}(0) - 2 \sum_{j=0}^{m} h(j)R_{wy}(j) + \sum_{j=0}^{m} h(j) \sum_{k=0}^{m} h(k)R_{ww}(k-j) \qquad (2.14)$$

The minimum value of the error I may be derived by substituting the normal equations (2.9) into (2.14).

$$I_{min} = R_{yy}(0) - 2 \sum_{j=0}^{m} h(j)R_{wy}(j) + \sum_{j=0}^{m} h(j)R_{wy}(j) \qquad (2.15)$$

$$I_{min} = R_{yy}(0) - \sum_{j=0}^{m} h(j)R_{wy}(j) \qquad (2.16)$$

It is common to normalize this minimum error by the zero lag coefficient of the autocovariance of the desired output.

$$E = \frac{I_{min}}{R_{yy}(0)} = 1 - \frac{\sum_{j=0}^{m} h(j)R_{wy}(j)}{R_{yy}(0)} \qquad (2.17)$$

E is bounded by 0 (no error between desired and actual filter outputs) and 1 (no agreement between desired and actual filter outputs).

2.3 Wiener Spike Filters

A special case of the Wiener shaping filter will be of particular interest. The desired filter output will be restricted to a spike $\delta(k-\beta)$ as in equation 1.5.

$$\delta(k-\beta) = \begin{cases} 1 & k = \beta \\ 0 & \text{otherwise} \end{cases} \tag{2.18}$$

That is, $\delta(k-\beta)$ is an all zero sequence except for a value of 1 at position β. For this particular output, the normal equations (2.9) reduce to:

$$\sum_{j=0}^{m} h(j)R_{ww}(k-j) = w(k+\beta) \qquad k = 0, 1, \ldots, m \tag{2.19}$$

and it is no longer necessary to compute the cross-covariance of the filter input and desired output. These shaping filters when restricted to spike outputs have been called Wiener (or least squares) spike filters and also Wiener (or least squares) inverse filters. They are called inverse filters since

$$H * W = \delta(k-\beta) \tag{2.20}$$

and taking a z-transform yields:

$$H(z)W(z) = z^{-\beta} \tag{2.21}$$

or

$$H(z) = \frac{z^{-\beta}}{W(z)} \tag{2.22}$$

It can be seen that the z-transform of the filter impulse response is the inverse of the z-transform of the input signal delayed by β samples. The problem with (2.22) is that H(z) is generally an infinite length two-sided filter. A finite length estimate of H must be developed to allow practical implementations.

2.4 The Phase Problem

It is important to consider some limitations placed on a filter designed by the normal equations (2.9). The most severe deficiency of this approach is that the only information used to design the filter is the autocorrelation of the filter input and the cross correlation of the filters input and output. Since only the amplitude spectrum of a signal can generally be derived from correlations, the lack of phase information may lead to problems. In fact, this limitation generally requires a search of all possible delays (lags) of the output waveform for the optimal filter design [6, pp. 77-78]. The abstract from a paper by Claerbout and Robinson [10] succinctly states:

> Least-squares inverse filtering always involves consideration of the error. Under certain conditions the error will go to zero as the length of the filter tends to infinity. It is shown that the error will go to zero if either: 1) the waveform being inverted is minimum-phase, or 2) if the output is chosen to come after a sufficiently long time delay. If the waveform being inverted is not minimum-phase and if in addition the output is not chosen to be delayed, then the error will be finite and may be large.

If one chooses to search for the best delay of the output, it is possible to solve for the filter coefficients for delayed output waveforms without repetitively solving the normal equations. A significant computational saving is attained by solving the normal equations for an output with no delay and then using Simpson sideways recursion [11] to solve for the delayed output filter coefficients.

Since, in many cases, Simpson sideways recursion still results in excessive computation or unacceptable numerical errors, the majority of applications involving least-squares filtering require that waveforms involved have the property of minimum-phase. Least squares filters that assume the property of minimum-phase for their input require no time delay in their output and are known as zero-lag Wiener filters.

Before presenting the requirements for "minimum-phase" signals, it is instructive to note the relationships between linear prediction and Wiener filters in order to better appreciate the scope of this "phase problem." Linear predictive filters have found widespread application in the areas of Neurophysics, Geophysics and Speech Communication [12]. A predictive filter is a filter that has as its output, an estimate of the future values of its input. A development similar to equations (2.1) through (2.10) starting with the defining filter equation

$$\sum_{j=0}^{m} a(j)w(i-j) = w(i+\alpha) \qquad (2.23)$$

where A = prediction filter impulse response at length m+1

W = filter input of length n+1

i = 0, 1, 2, ..., n+m

α = prediction distance

leads to the normal equations:

$$\sum_{j=0}^{m} a(j)R_{ww}(k-j) = R_{ww}(k+\alpha) \qquad k = 0, 1, \ldots, m \qquad (2.24)$$

for least squares linear predictive filters. These normal equations may be solved in the same manner as the Wiener shaping filter case. It is common to define a prediction error series (residual) which is just the difference between the predicted future value of the input and the actual future value of the input.

$$e(i+\alpha) = w(i+\alpha) - \sum_{j=0}^{m} a(j)w(i=j) \qquad i = 0, 1, \ldots, m+n \qquad (2.25)$$

Since the linear prediction normal equations (2.24) only involve the autocovariance of the filter input, this class of filters suffers from the same lack of phase information as the Wiener filters. Therefore, minimum-phase input signals play a key role in the application of these filters as well. In fact, Peacock and Treitel [13] have shown that: "The least-squares prediction filter with unit prediction distance is equivalent within a scale factor to the least-squares, zero lag inverse filter."

The minimum-phase property plays a significant role in many other areas of signal processing. For example, it is only for minimum-phase signals that a Hilbert Transform exists relating the logarithm of the magnitude of the frequency response and the phase for a causal system [14, pp. 70-73]. Minimum-phase sequences also play an important role in the theory of Homomorphic systems [15, pp. 345-353]. Chapter III will be concerned with the phase properties of signals.

III. TONE BURST CHARACTERIZATION

3.1 Introduction

It was shown in Chapter II that least-squares filtering methods suffer from a lack of the use of phase information. In this chapter, a representation of phase information using polynomial zero behavior will be developed. It will be shown that not all sequences meet the requirements of zero-lag Wiener filters and two preprocessing transformations will be presented to correct this problem. Finally, tone burst signals will be studied in detail to determine if Wiener filters may be successfully used on this class of signals.

In this and following sections, the z-transform will be utilized extensively. Since many good texts in signal processing cover the fundamentals for z-transform theory, several well-known results will be used without reference (c.f. [14-17]).

As in Chapter II, it will be assumed that a finite length discrete sequence is to be analyzed. It is given by:

$$W = [w(0), w(1), w(2), \ldots, w(n)] \tag{3.1}$$

Finite sequences of data such as 3.1 may be considered to belong to a class of signals called wavelets. Wavelets have been defined by Robinson [17, p. 111] to have two properties:

a. "The one-sided property: A wavelet has a definite origin (or arrival) time in the sense that all values of the wavelet before its origin time are zero." A sequence with this property is commonly called a causal sequence.

b. "The stability property: A wavelet has finite energy so that it is a transient or dying-out phenomenon."

A finite sequence such as 3.1 obviously meets both of these properties.

For the finite length sequence given by (3.1), the z-transform is given by:

$$W(z) = \sum_{i=o}^{n} w(i)z^{-i} \qquad (3.2)$$

where z is a complex variable. Equation (3.2) can be written in expanded form as:

$$W(z) = w(0) + w(1)z^{-1} + \ldots + w(n)z^{-n} \qquad (3.3)$$

When the z-transform is expressed in polar coordinates, the value of the function may be expressed in terms of magnitude and phase. The power spectrum of W is given by the magnitude squared of the z-transform of W evaluated along a circle of radius one centered at the origin (the unit circle). The phase spectrum is similarly defined as the phase along the same contour.

3.2 The Pole/Zero Model

It is possible to consider W to be the impulse response of some system,in which case the poles and zeros of the polynomial W(z) take on great significance in characterizing the system. Equation (3.3) may be rewritten to display its pole-zero characteristics by factoring z^{-n}.

$$W(z) = z^{-n}[w(0)z^{n} + w(1)z^{n-1} + \ldots + w(n)] \qquad (3.4)$$

From Equation (3.4), it can be seen that a wavelet of length n+1 is characterized by n poles located at z=0 and n zeros located at the roots of the polynomial given by the expression in brackets. The fundamental theorem of algebra insures that the polynomial in brackets may always be factored. Furthermore, if the values of W are real, then the roots occur either in conjugate pairs or as real roots. Using the basic factorability of polynomials (3.4) may be rewritten in product form as:

$$W(z) = z^{-n}[c(z-\omega(0))(z-\omega(1)) \ldots (z - \omega(n-1)] \qquad (3.5)$$

A wavelet can be characterized to within a scale factor by the complex values of the zeros of the z-transform of that wavelet:

$$\Omega = \{\omega(0), \omega(1) \ldots \omega(n-1)\} \qquad (3.6)$$

As a numerical example, assume that a wavelet is given as follows:

$$W = [-2, 6, -5, 2] \qquad (3.7)$$

From (3.3), the z-transform of W is given by:

$$W(z) = -2 + 6z^{-1} - 5z^{-2} + 2z^{-3} \qquad (3.8)$$

which when factored as in (3.4), becomes:

$$W(z) = z^{-3}[-2z^3 + 6z^2 - 5z + 2] \qquad (3.9)$$

It is easy to verify that the zeros of the polynomial in brackets in (3.9) are:

$$\Omega = \{(\tfrac{1}{2} + \tfrac{1}{2}j), (\tfrac{1}{2} - \tfrac{1}{2}j), 2\} \quad \text{where } j = \sqrt{-1} \qquad (3.10)$$

and that the wavelet given by (3.7) can be represented as in (3.5) as:

$$W(z) = z^{-3}[-2(z-\tfrac{1}{2}-\tfrac{1}{2}j)(z-\tfrac{1}{2}+\tfrac{1}{2}j)(z-2)] \tag{3.11}$$

A pole-zero plot of the z-transform of W given by (3.11) would be as follows:

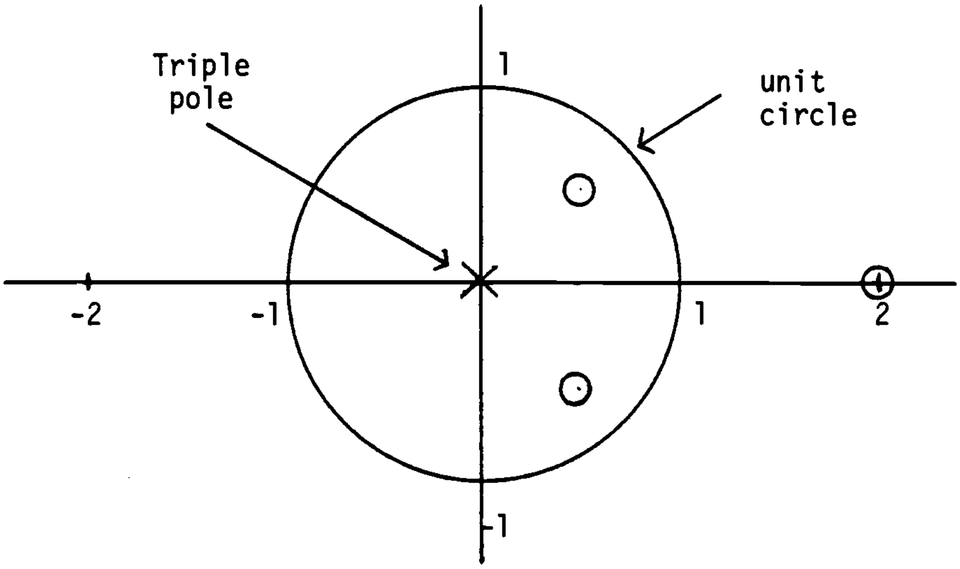


Figure 3.1

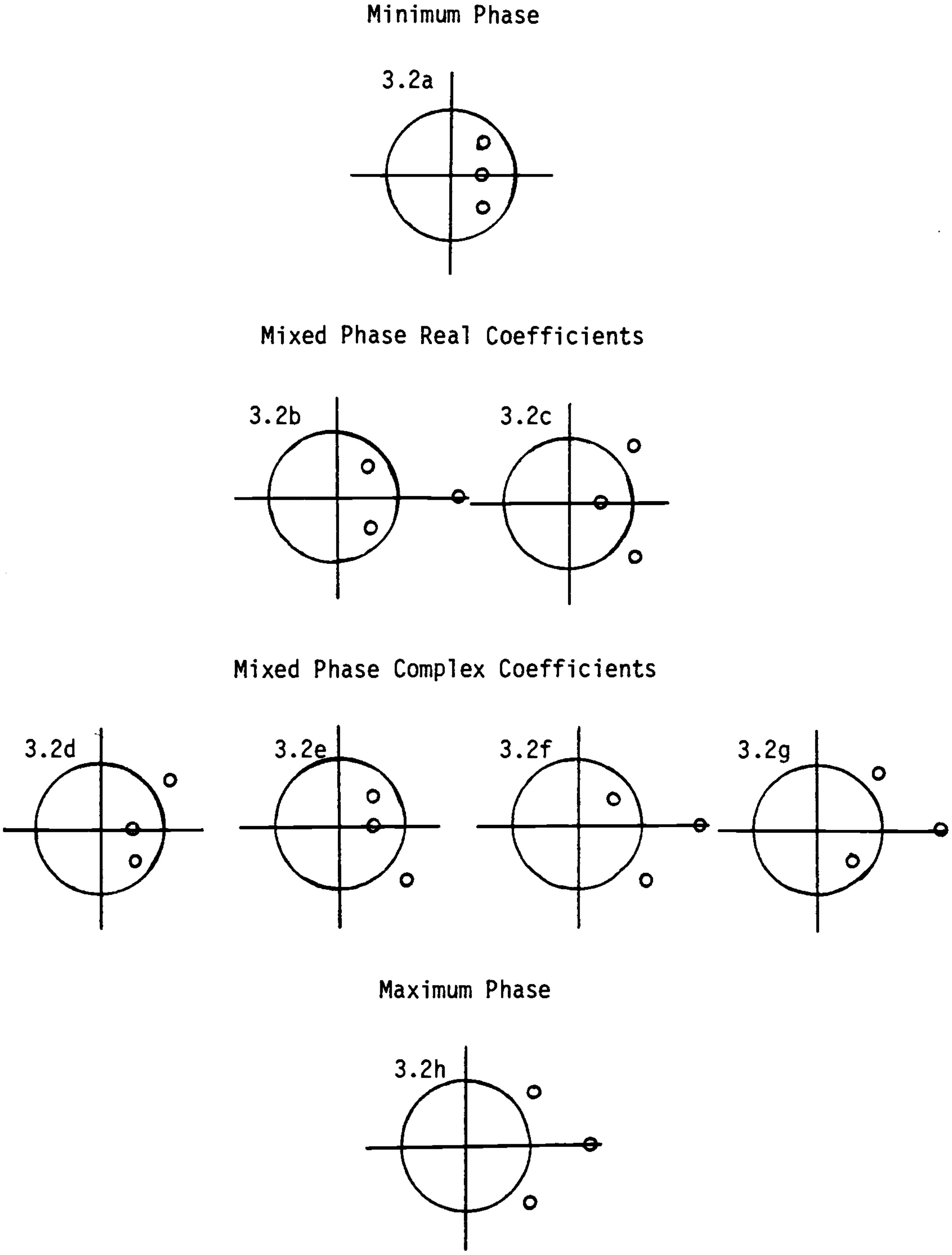
where x indicates the location of poles and 0 indicates the location of zeros.

It is convenient to characterize wavelets (and more general signals as well) by the location of their zeros [15, pp. 354-353]. For a finite length sequence, all poles are located at the origin as presented in (3.4). Therefore, poles contribute no knowledge concerning the signals' characteristics. A wavelet with all its zeros inside the unit circle is called a <u>minimum phase wavelet</u>. A wavelet with all its zeros outside the unit circle is called a <u>maximum phase wavelet</u>.

A wavelet with some zeros inside and some zeros outside the unit circle is called a mixed phase wavelet. This terminology is not intuitively obvious. It must be considered that for a given power spectrum, there is a class of signals all differing in phase spectrum. If the phase spectrum were computed for all the possible wavelets with a given power spectrum, it would turn out that the signal with the minimum phase spectrum would be the only such signal with all its zeros inside the unit circle. A similar statement holds with respect to the wavelet with the maximum phase spectrum - it would have all its zeros outside the unit circle. The remainder of the wavelets with the common power spectrum would have phase spectra intermediate between the two extremes and would have some zeros inside the unit circle and some zeros outside the unit circle. It is, in fact, possible to order all possible wavelets with a given power spectrum by their phase.

To make these ideas more concrete, the numerical example of (3.7) will be continued. All the length four, real-coefficient sequences with the same power spectra as (3.7) will be found. A simple method of constructing sequences with common power spectra is to make use of the property that if a zero of a sequence, located at radius r, is replaced by another zero located at radius 1/r, the power spectra does not change [18, p. 441]. Thus, we may geometrically construct eight possible zero plots of which four have real coefficients for their corresponding sequences as in Figure 3.2.

Note that Figure 3.2b is the zero plot corresponding to the wavelet of (3.7). The minimum phase wavelet corresponding to Figure 3.2a may be derived by writing the product form of the z-transform as follows:



Minimum Phase

Mixed Phase Real Coefficients

Mixed Phase Complex Coefficients

Maximum Phase

Figure 3.2

Zero Plots for Sequences Having a Common Power Spectra

$$W(z) = z^{-3}[c(z-\tfrac{1}{2}-\tfrac{1}{2}j)(z-\tfrac{1}{2}+\tfrac{1}{2}j)(z-\tfrac{1}{2})] \tag{3.12}$$

Equation (3.12) may be expanded yielding:

$$W(z) = z^{-3}\ [c(z^3 - 3/2z^2 + z - 1/4)] \tag{3.13}$$

The value of the constant c may be determined from the requirement that the power spectra (the magnitude of the z-transform evaluated on the unit circle) be the same as the original wavelet (3.8). For simplicity, (3.13) and (3.8) may be evaluated at z=1 and equated. From (3.13) we have:

$$W(1) = 1/4c \tag{3.14}$$

From (3.8) we have:

$$W(1) = 1 \tag{3.15}$$

Equating (3.14) and (3.15) yields a value of c=4. Thus, (3.13) becomes:

$$W(z) = 4 - 6z^{-1} + 4z^{-2} - z^{-3} \tag{3.16}$$

and the minimum phase sequence with the same power spectra as (3.7) is:

$$W = [4, -6, 4, -1] \tag{3.17}$$

Wavelets corresponding to Figures 3.2c and 3.2h may be derived in a similar manner. A simpler method is to make use of the property that the time reversal of a wavelet has the effect of replacing all zeros with zeros located at the inverse of the original locations.

Therefore, the sequence corresponding to Figure 3.2c can be readily derived from the sequence (3.7) corresponding to Figure 3.2b as:

$$W = [2, -5, 6, -2] \tag{3.18}$$

and the maximum phase wavelet corresponding to Figure 3.2h can be determined directly from the minimum phase sequence (3.17) as:

$$W = [-1, 4, -6, 4] \tag{3.19}$$

Figure 3.3 is the power spectrum common to these wavelets and Figure 3.4 is the phase spectra for each of the wavelets with real coefficients. The first point plotted in Figures 3.3 and 3.4 corresponds to the magnitude and phase values on the unit circle at the intersection with the positive real axis. The last point plotted corresponds to the respective values on the unit circle at the intersection with the negative real axis. Points between the first and last are equally spaced values along the unit circle in a counterclockwise direction.

It was shown in Section 2.4 that one procedure for designing optimal least-squares filters is with zero-lag Wiener filters. A requirement of such filters is that the sequence to be filtered must be minimum phase. It is easy to see that, in general, wavelets do not have this property. Most successful least-squares filtering applications such as seismology have been forced to assume the phase characteristics of their signals and have generally shown qualitatively that their signals are approximately minimum phase. Those signal types that are not generally minimum phase have typically been preprocessed with a heuristically developed transformation to make the signals approximately minimum phase.

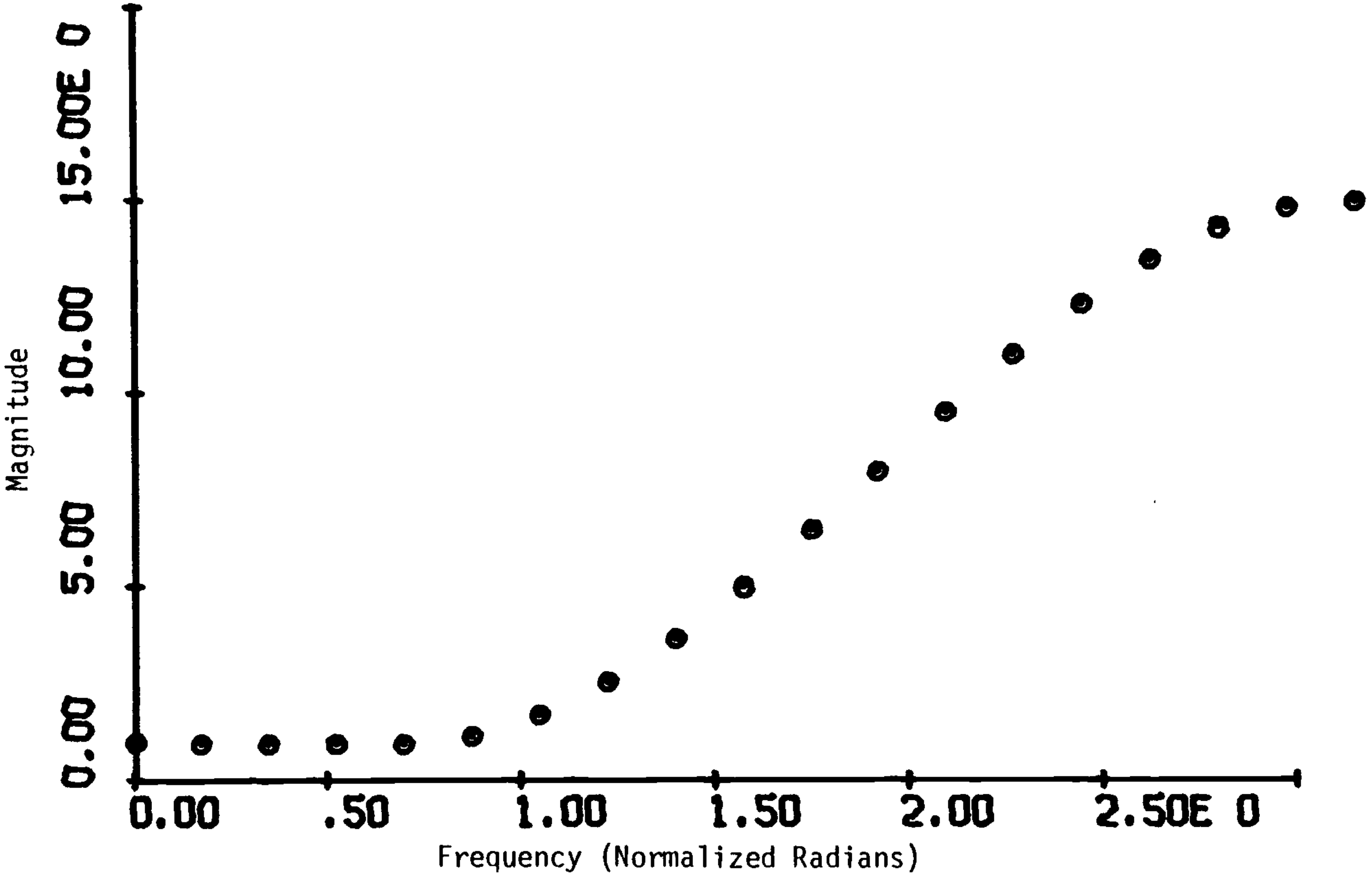


Figure 3.3

Power Spectrum

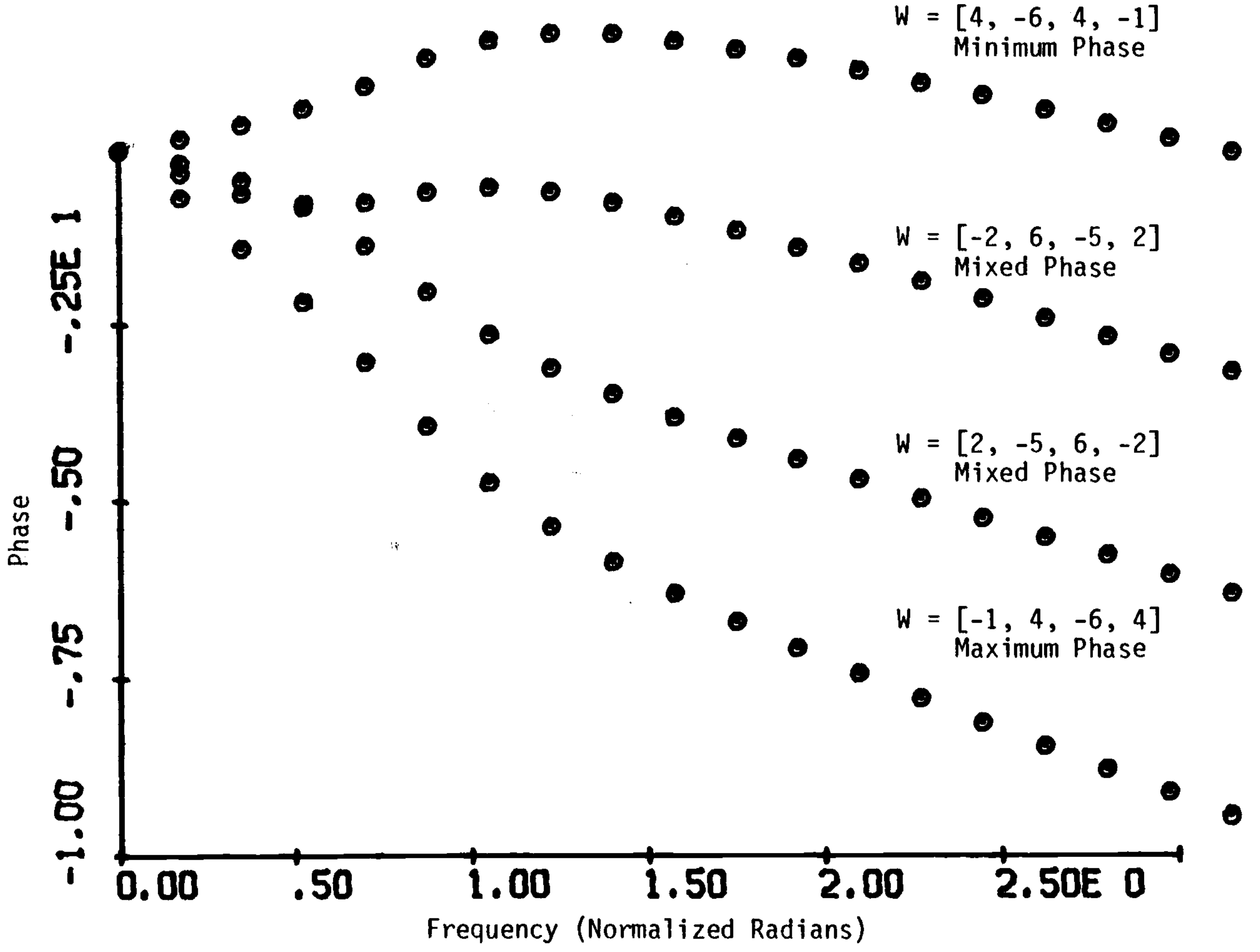


Figure 3.4

Phase Spectrum

3.3 Dispersive All-Pass Filters

Undoubtedly, the most theoretically pleasing approach to minimum phase transformations have been the so-called dispersive all-pass filters [18 and 15, p. 348]. These filters are very simple to design if the exact location of all zeros are known. The essential operation of these filters involves positioning a pole exactly on top of any zeros outside the unit circle. This removes all zeros outside the unit circle by pole-zero cancellation. Next, new zeros are added inside the unit circle at locations which are the inverse of the added poles. Thus, the result of applying a dispersive filter to a non-minimum phase sequence is to generate a minimum-phase sequence with the same power spectrum. Unfortunately, such a procedure is of little practical importance. It is not possible to compute the exact location of polynomial zeros for sequences as long as those typically encountered in practical situations. Numerical errors prevent root finding routines from working for sequences much longer than 100 data points. Even if the location of the zeros outside the unit circle could be found, numerical errors in attempting exact pole-zero cancellation may be severe.

3.4 Exponential Weighting

Another transformation for moving zeros is exponential weighting. Exponential weighting, unlike dispersive filtering, affects both the power and phase spectrum. However, it does not seem that for inverse filtering applications it is necessary or even desirable to use transformations that maintain a given power spectrum. The effect of exponential weighting is easy to observe as follows:

Let A be the exponential weight sequence of length n+1:

$$A = [1, \alpha^{-1}, \alpha^{-2}, \ldots, \alpha^{-n}] \tag{3.20}$$

and as before, let W be the signal to be analyzed of length n+1:

$$W = [w(0), w(1), \ldots, w(n)] \tag{3.21}$$

then the weighted sequence is given by:

$$W' = [w(0), \alpha^{-1}w(1), \alpha^{-2}w(2), \ldots, \alpha^{-n}w(n)] \tag{3.22}$$

The z-transform of W' is:

$$W'(z) = \sum_{i=0}^{n} \alpha^{-i}w(i)z^{-i} = \sum_{i=0}^{n} w(i)(\alpha z)^{-i} \tag{3.23}$$

From (3.23) it is easy to see that the z-transform of an exponentially weighted sequence is just the z-transform of the unweighted sequence with the complex variable scaled by α.

Thus, exponential weighting has the effect of linear scaling of the z domain. It is also easy to see that the distance from the origin of the z-plane to any zero is also linearly scaled by α. For α greater than one, the zero locations are moved on radials toward the origin. For α less than one, the zeros are moved along radials away from the origin.

Because of this very simple linear scaling, the exponential transformation is a good candidate for transforming non-minimum phase sequences into minimum phase sequences. By the proper selection of α, it is possible to move all or nearly all zeros inside the unit circle, thereby conditioning general sequences so they will be compatible with the requirements of zero-lag Wiener filters. There are three major problems to be overcome when using this transformation. They are:

1) If the value of α is not close to one, then the exponential weights will exceed the numerical range of most computers for even moderate length sequences. As an example for α = 1.2 and a sequence of length 500, the last weight is given by:

$$\alpha^{-500} = (1.2)^{-500} = 2.57 \times 10^{-40}$$

For a sequence of length 1,000, this factor becomes 6.59×10^{-80}.

2) It is necessary to determine the radius of the zero most distant from the origin to transform a sequence into a minimum phase sequence. If this distance is too great and would lead to the numerical errors discussed in 1) above, then a method for selecting a "best" value of α must be used.

3) Since exponential weighting is a non-linear process, it is very difficult to determine the effect of unwanted signals (noise) on the filter performance. Experimentally, it appears that as the value of α approaches one, these problems become less severe. This topic is more completely discussed in Chapter V.

The remainder of this chapter will be devoted to studying a particular signal type for compatibility with these requirements. Chapter IV will be devoted to the development of an algorithm to determine the parameters necessary to make a useful selection of the value of α.

3.5 Tone Burst Analysis

3.5.1 Analytic Analysis

To demonstrate the practicality of the methods developed, least-squares inverse filters will be applied to pulse modulated continuous wave (tone burst) signals much like those commonly encountered in radar and sonar. As a simplified model, it will be assumed that this signal may be described by:

$$W(k) = B(m)\cos(\omega kT + \theta) \tag{3.24}$$

where B(m) is the well-known boxcar function given by:

$$B(m) = \begin{cases} 1 & 0 \leq k \leq m \\ 0 & \text{otherwise} \end{cases} \tag{3.25}$$

To determine the zero behavior of this class of wavelets, it is possible to take the z-transform and then attempt to evaluate the resultant polynomial for the zero locations. The z-transform of (3.24) is:

$$W(z) = \sum_{k=0}^{\infty} B(m)\cos(\omega kT + \Theta)z^{-k} \tag{3.26}$$

The boxcar function may be expressed in terms of unit step functions as:

$$B(m) = u_{-1}(0) - u_{-1}(m) \tag{3.27}$$

where

$$u_{-1}(\ell) = \begin{cases} 0 & k < \ell \\ 1 & k \geq \ell \end{cases} \tag{3.28}$$

Substituting (3.27) and (3.28) into (3.26), yields:

$$W(z) = \sum_{k=0}^{\infty} \cos(\omega kT + \Theta)z^{-k} - \sum_{k=0}^{\infty} \cos[\omega T(k+m) + \Theta]z^{-(k+m)} \tag{3.29}$$

The right-hand expression in (3.29) can be evaluated by using the trigonometric identity:

$$\cos(\beta + \alpha) = \cos\alpha\cos\beta - \sin\alpha\sin\beta \tag{3.30}$$

The left-hand expression in (3.29) can be found in any good table of z-transforms. It will be notationally convenient to make the substitution:

$$G(z) = \sum_{k=0}^{\infty}\cos(\omega kT + \theta)z^{-k} = \frac{z[z\cos\theta - \cos(\omega T - \theta)]}{(z - e^{j\omega T})(z - e^{-j\omega T})} \tag{3.31}$$

Substituting (3.30) and (3.31) into (3.29) yields:

$$W(z) = G(z) - z^{-m}\sum_{k=0}^{\infty}\cos(\omega Tk + \theta)\cos(\omega Tm)z^{-k}$$

$$+ z^{-m}\sum_{k=0}^{\infty}\sin(\omega TK + \theta)\sin(\omega Tm)z^{-k} \tag{3.32}$$

$$W(z) = G(z) - z^{-m}\cos(\omega Tm)\sum_{k=0}^{\infty}\cos(\omega Tk + \theta)z^{-k}$$

$$+ z^{-m}\sin(\omega Tm)\sum_{k=0}^{\infty}\sin(\omega Tk + \theta)z^{-k} \tag{3.33}$$

$$W(z) = G(z) - z^{-m}\cos(\omega Tm)G(z) + z^{-m}\sin(\omega Tm)\sum_{k=0}^{\infty}\sin(\omega Tk + \theta)z^{-k} \tag{3.34}$$

In a manner similar to (3.31) we define:

$$H(z) = \sum_{k=0}^{\infty}\sin(\omega Tk + \theta)z^{-k} = \frac{z[z\sin\theta + \sin(\omega T - \theta)]}{(z - e^{j\omega T})(z - e^{-j\omega T})} \tag{3.35}$$

Substituting (3.35) into (3.34) yields:

$$W(z) = G(z)[1 - z^{-m}\cos(\omega Tm)] + z^{-m}\sin(\omega Tm)H(z) \tag{3.36}$$

$$W(z) = z^{-m}[G(z)(z^{m} - \cos(\omega Tm)) + \sin(\omega Tm)H(z)] \qquad (3.37)$$

It is possible to express H(z) in terms of G(z). By inspection it is clear that:

$$H(z) = G(z)\left[\frac{z\sin\theta + \sin(\omega T - \theta)}{z\cos\theta - \cos(\omega T - \theta)}\right] \qquad (3.38)$$

(3.38) can be substituted into (3.37) yielding:

$$W(z) = G(z)z^{-m}[z^{m} - \cos(\omega Tm) + \sin(\omega Tm)\left(\frac{z\sin\theta + \sin(\omega T - \theta)}{z\cos\theta - \cos(\omega T - \theta)}\right)] \qquad (3.39)$$

Some tedius manipulation of (3.39) involving substitution of (3.31) into (3.39), expansion of multiplicative terms and the use of the identity (3.30) yields the expression:

$$W(z) = \frac{z^{-m+1}[\cos\theta z^{m+1} - \cos(\omega T - \theta)z^{m} - \cos(\omega Tm + \theta)z + \cos(\omega T(m-1) + \theta)]}{(z - e^{j\omega T})(z - e^{-j\omega T})} \qquad (3.40)$$

It can now be seen that the zeros of the z-transform of a tone burst signal are given by the roots of the following polynomial:

$$\cos\theta z^{m+1} - \cos(\omega T - \theta)z^{m} - \cos(\omega Tm + \theta)z + \cos(\omega T(m - 1) + \theta) \qquad (3.41)$$

A general solution to (3.41) has not, to the author's knowledge, been determined. A particular solution of (3.41) will give some insight into the zero behavior of this type signal. If the width of the tone burst were chosen such that an integral number of cycles of the cosinusoid were in the pulse, then:

$$\omega Tm = 2\pi n \qquad n = 1, 2, \ldots \qquad (3.42)$$

Substituting (3.42) into (3.41) yields

$$\cos(\theta)z^{m+1} - \cos(\omega T - \theta)z^{m} - \cos\theta z + \cos(-\omega T + \theta) = 0 \qquad (3.43)$$

We may use the fact that $\cos(\omega T - \theta) = \cos(-\omega T + \theta)$ and then factor (3.43) into

$$(z^{m} - 1)(\cos\theta z - \cos(\omega T - \theta)) = 0 \qquad (3.44)$$

and it can be seen that there are m-1 zeros on the unit circle and a real zero located at

$$z = \frac{\cos(\omega T - \theta)}{\cos\theta} \qquad (3.45)$$

Physically, a tone burst must have a phase angle θ equal to either $\pi/2$ or $3\pi/2$, in which case the pulse starts with a value of 0. In this situation, the real root (3.45) is very large. Based on this case, it is not to be expected that this signal type is minimum phase in practice. What is worse, it is to be expected that at least one zero will be found at exceptionally large distances outside the unit circle. Additionally, noise (always found in practice) will have the effect of moving some zeros toward the origin and some zeros away from the origin in a random fashion.

3.5.2 Simulation Analysis

Since (3.43) does not yield to complete analytic solution, it is reasonable to use simulation to determine the qualitative behavior of the roots. There are three parameters in (3.43). M is the number of samples minus 1 in the tone burst. The product ωT is essentially the ratio of the signal frequency to the sample frequency. That is, since

$$T = \frac{1}{f_s} = \frac{2\pi}{\omega_s} \qquad f_s = \text{the sample frequency} \qquad (3.46)$$

we have

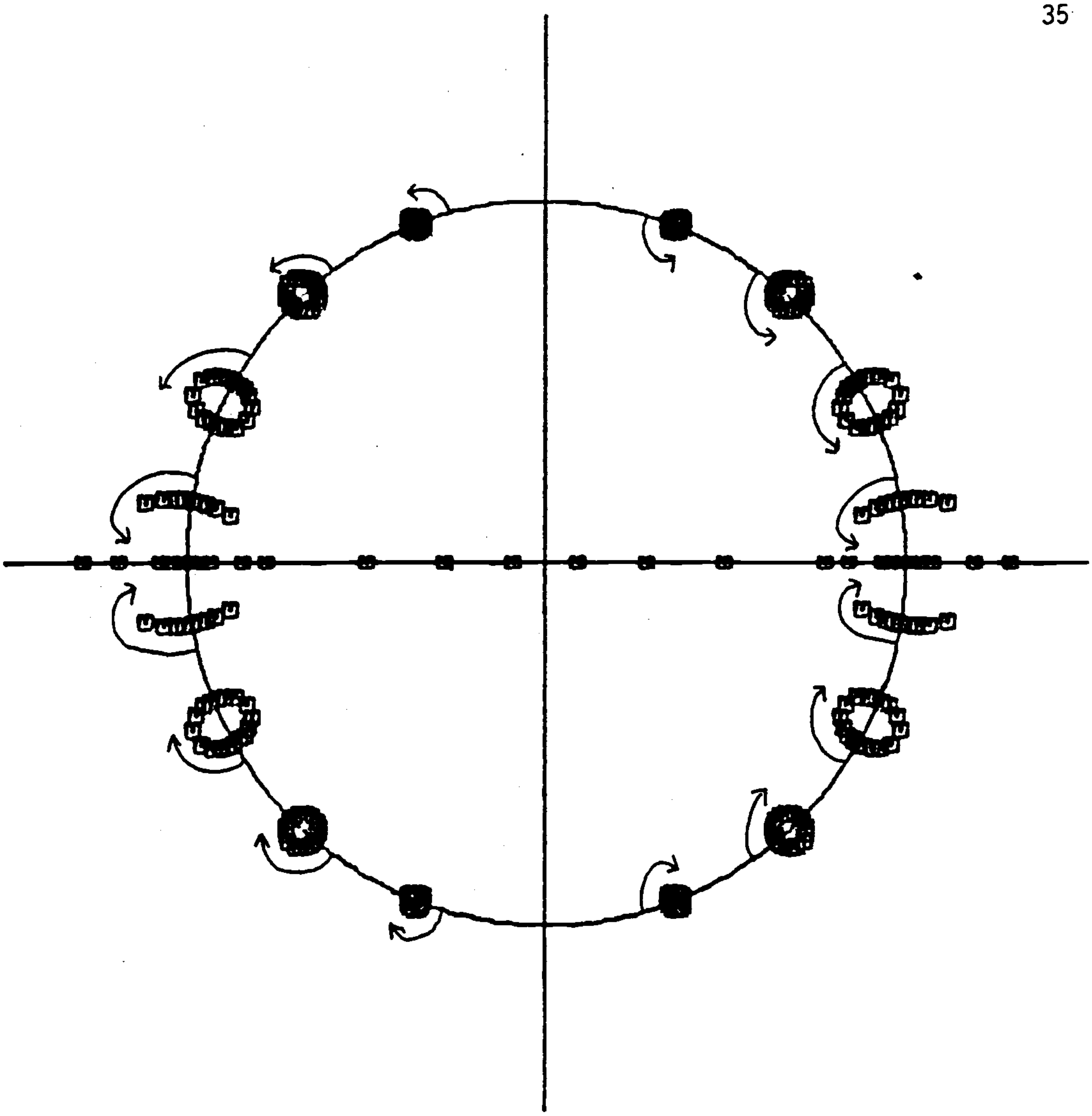
$$\omega T = \frac{2\pi\omega}{\omega_s} \qquad \omega_s = \text{the angular sample frequency} \qquad (3.47)$$

The factor θ is just the phase angle of the cosinusoid.

To study the interaction of these three variables on the zero locations, a computer program was written to numerically determine the zero locations of the z-transform of wavelets generated by the model of equation (3.24). The purpose of the program was to solve and plot the zero locations for fixed ωT and m while θ was varied between 0 and π. Figures 3.5, 3.6 and 3.7 are typical output from this program.

Each symbol plotted in Figures 3.5-3.7 is the location of a single zero for a value of ωT, m and π. It is apparent that the zero behavior as θ is varied is generally an elliptical motion centered on the unit circle.

It is possible to make some general statements concerning the effect of these three parameters based on this simulation. A comparison of Figures 3.5 and 3.6 indicates that a variation in ωT has the principal effect of relocating the center of rotation of the zeros around the unit circle. There is little if any affect on the distance inside or outside the unit circle of the most widely varying zero. One may, therefore, assume that ωT plays a minor role when trying to determine the number of zeros outside a radius in the z-plane. It is clear from all three figures that the most important of the three factors for determining if a given pulse CW signal is minimum phase is the phase angle θ. A variance in θ has a dramatic effect on zero location.



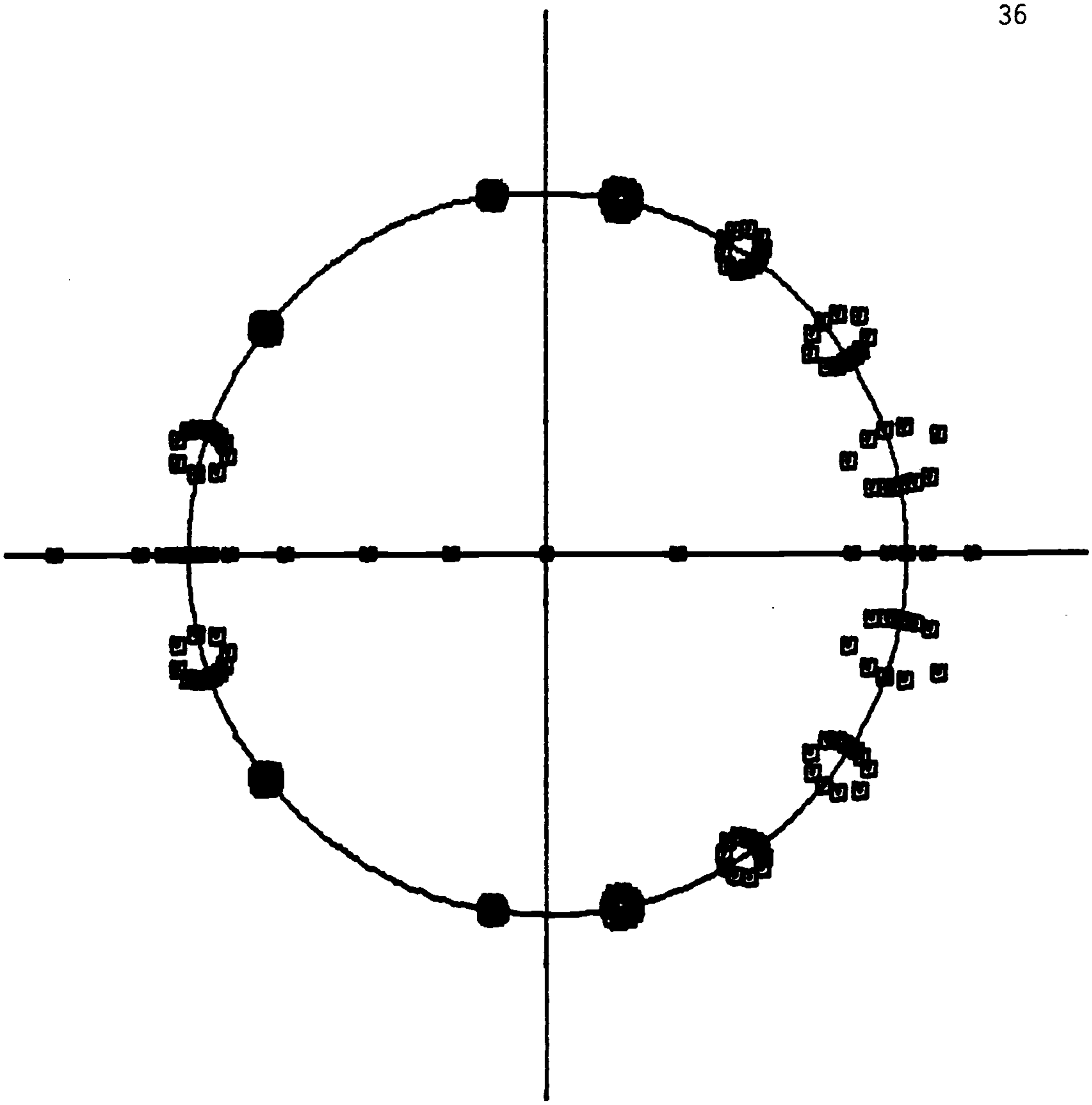
$$\omega T = \frac{2\pi}{4}$$

$$m = 17$$

$$\theta = 0, \frac{\pi}{12}, \frac{2\pi}{12}, \ldots, \frac{11\pi}{12}$$

Figure 3.5

The Zeros of $B_m(k)\cos(\omega kT + \theta)$



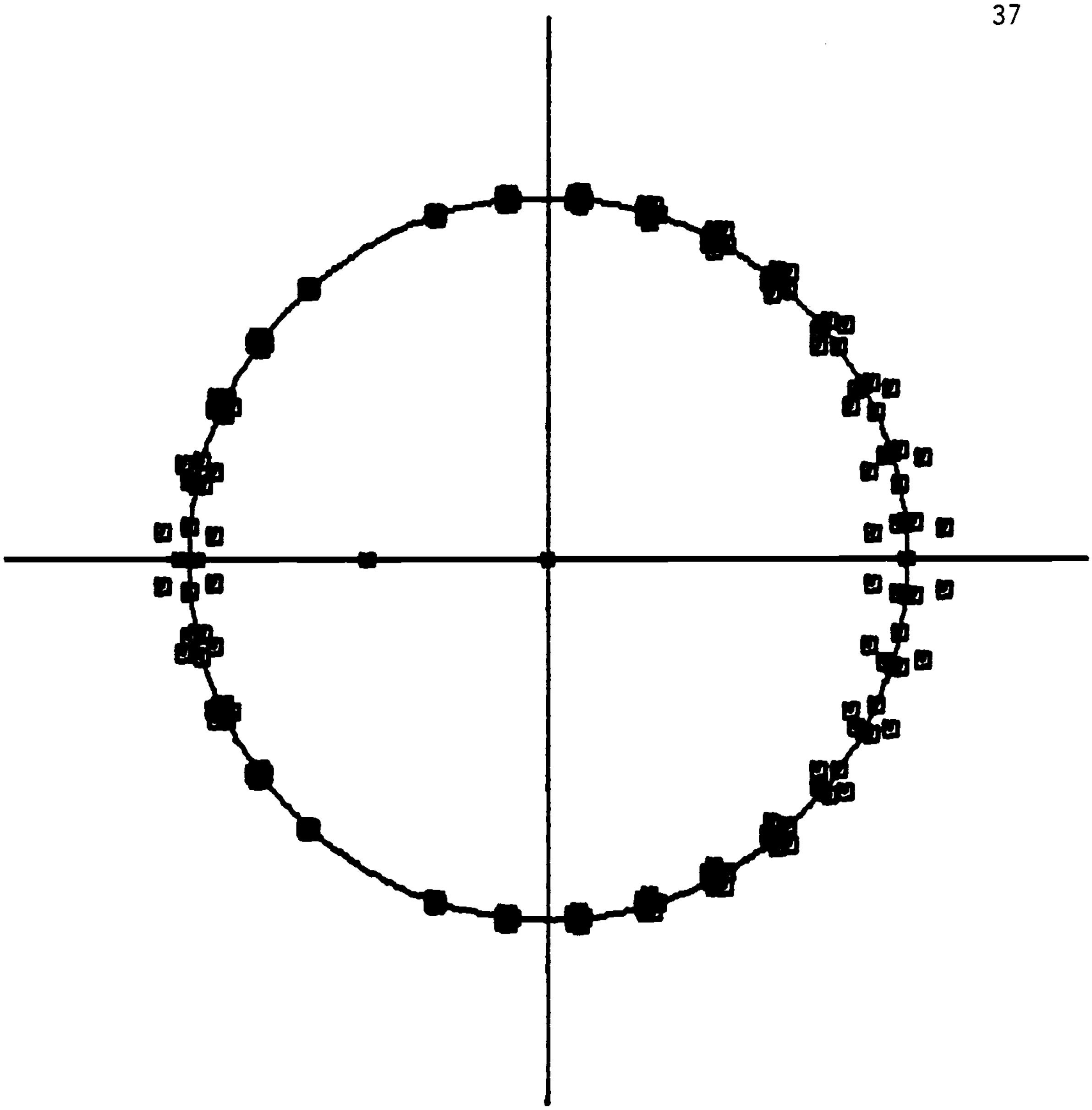
$\omega T = \frac{2\pi}{3}$

$m = 17$

$\theta = 0, \frac{\pi}{12}, \frac{2\pi}{12}, \ldots, \frac{11\pi}{12}$

Figure 3.6

The Zeros of $B_m(k)\cos(\omega kT + \theta)$



$\omega T = \frac{2\pi}{3}$

$m = 31$

$\theta = 0, \frac{\pi}{6}, \frac{2\pi}{6}, \dots \frac{5\pi}{6}$

Figure 3.7

The Zeros of $B_m(k) \cos(\omega kT + \theta)$

Figure 3.5 has arrows indicating the direction of motion of the zeros as θ is increased from 0 to π. The tail of the arrow indicates the $\theta = 0$ zero locations. These are all on the unit circle (except the real zero).

The most interesting aspect of Figures 3.5-3.7 is that for $0 \leq \theta \leq \pi/2$ all zeros (except the real zero) in the right-half plane are inside the unit circle and all zeros in the left-half plane are outside the unit circle, resulting in a mixed phase configuration. For $\pi/2 < \theta < \pi$, the opposite is true, still resulting in a mixed phase signal. In general, it should be expected that a tone burst signal will be mixed phase.

The effect of the third parameter, m, is clear from Figures 3.6 and 3.7. As m increases, the maximum deviation from the unit circle decreases. In other words, as the sequence being analyzed increases in length, the zeros approach the unit circle. Since sequences of practical importance are generally much greater in length than the sequences of Figures 3.5-3.7, it is reasonable to expect that most zeros are located very near the unit circle. Transformations that move zeros inside the unit circle need only to move zeros a short distance.

3.6 Summary

To summarize this Chapter, it has been shown that tone burst waveforms do not meet the minimum phase requirement for zero-lag digital Wiener filters. However, an exponential transformation can be used to move zeros of a sequence inside the unit circle. It has been shown that tone burst signals lend themselves well to such transformations since the zeros of the z-transform of tone burst signals lie nearly on the unit circle. This means that only light exponential weighting need be used and numerical errors due to the dynamic range limitations of most digital computers should not be a serious problem.

IV. EXPONENTIAL WEIGHTING

4.1 Introduction

It is clear from the previous two chapters that it is not generally practical to apply digital Wiener filtering methods directly to pulsed continuous wave (tone burst) signals. Since this signal type is generally mixed phase, zero-lag Wiener filters will not perform satisfactorily. Further, the process of searching for an appropriate lag to match the signal's characteristics requires excessive computation. It was also shown that it is not generally possible to transform tone burst signals into minimum phase sequences using exponential weighting since at least one zero occurs at a great distance from the unit circle on the real axis. It is the purpose of this chapter to develop an efficient algorithm that determines an appropriate lag and exponential weighting for the pulsed continuous wave waveform (CW).

It will first be shown that it is possible to select a small range of (good) filter lags given that a few zeros of the z-transform of a wavelet are outside the unit circle. Treitel and Robinson [3, p. 31] have heuristically suggested positioning the desired output spike with a delay that corresponds to the relative location of most of the energy in the input waveform. They argue, most reasonably, that the best chance of detecting a given waveform is by utilizing the information in the highest energy portion of that waveform. A problem exists with this procedure whenever the length of the filter is short with respect to the required delay. This becomes evident when one inspects analytic inverse filters.

4.2 Analytic Inverse Filters

An analytic inverse filter, H, for a sequence, W, is a convolutional filter with the property that H convolved with W results in a spike. As will be shown in this section, these filters are generally infinite in length. The Wiener spike filter developed in Chapter II is a finite length least-squares approximation to the analytic inverse filter.

In section 3.2, the product form for the z-transform of a wavelet was derived as:

$$W(z) = z^{-n}[c(z - \omega(0))(z - \omega(1)) \ldots (z - \omega(n-1))] \qquad (4.1)$$

Applying an inverse z-transform to (4.1) and using the product-convolution property yields:

$$W(n) = C(1, \omega(0)) * (1, \omega(1)) * \ldots * (1, \omega(n)) \qquad (4.2)$$

where * is the convolution operator defined by equation 2.1. It is possible to calculate the inverses for each of the doublets (1, $\omega(i)$), i=0, n and then convolve the inverses to yield the inverse of W. Because of this, it is of interest to study in detail the properties of a doublet (1,k) [19].

It is easy to show that there are two cases when determining the exact inverse of (1,k):

Case 1. If $|k| < 1$ (minimum phase doublet), then the inverse H is of the form:

$$h(n) = \begin{cases} 0 & n < 0 \\ (-k)^n & n \geq 0 \end{cases} \qquad (4.3)$$

That is, the inverse is an infinite realizable filter. A realizable filter is one that depends only on past and present filter inputs to determine the present output. Such filters are intuitively appealing since in real-time processing only past and present values of the filter input are available.

Case 2. If $|k| > 1$ (maximum phase doublet), then the inverse is of the form:

$$h(n) = \begin{cases} -(-k)^{-n} & n < 0 \\ 0 & n \geq 0 \end{cases} \qquad (4.4)$$

In this case, we have an infinite purely non-realizable filter since only future values of the filter input are used in determining the current output. Historically, non-realizable inverses were not practical, but with the advent of digital computers, it is a simple matter to collect and store entire data records prior to processing. This makes future values of a signal available for processing at a later time.

If a wavelet is composed of both minimum and maximum phase doublets, it is mixed phase and the analytic inverse filter is a convolution of both realizable and non-realizable filters. This class of filters is frequently called two-sided. In the general analytic case, one must make use of all previous, present and future filter inputs to calculate the inverse. As presented in Chapter II, the digital Wiener filter is of finite length. In computing practical implementations of digital Wiener inverse filters, we are in fact attempting to approximate in the least-squares sense the analytic infinite inverse filter with a finite length filter. Thus, a two-sided filter must be used for best performance for mixed phase signals.

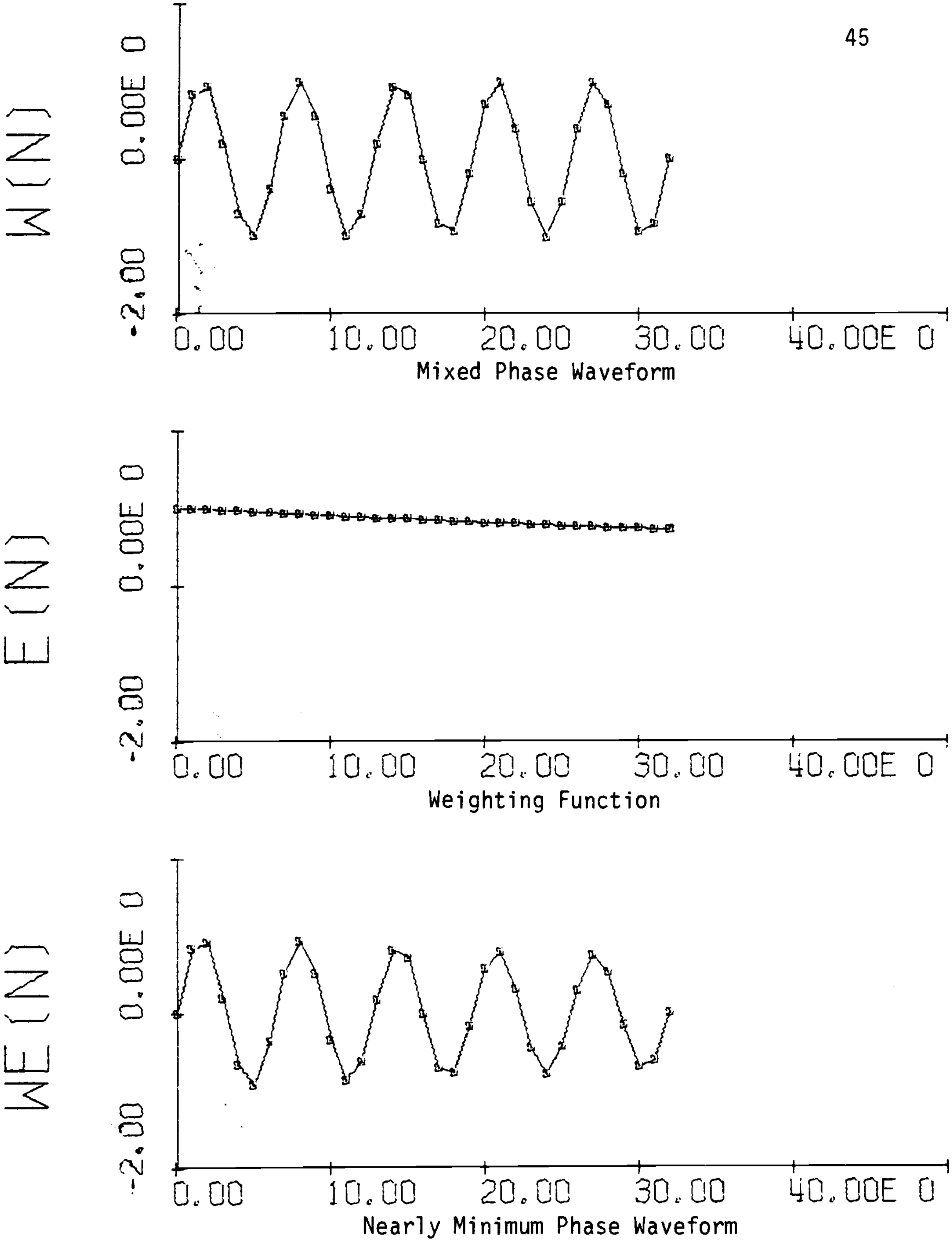
In practice, the two-sided nature of the filter to be designed is accounted for by a delay in the required output as in equation (2.20), which is repeated here:

$$H * W = \delta(k-\beta). \tag{4.5}$$

It is important to note that if the delay β is greater than or equal to the length of the inverse filter, then the filter must be purely non-realizable. This places an upper bound on the allowable output delay. The output delay for a mixed phase input must be less than the length of the filter to be designed. It is now clear that Treitel and Robinson's suggestion to position the output relative to the most energy in the input is only valid if the maximum energy in the input is delayed less than the length of the inverse filter. For short filter lengths, this requires that the energy of the input must be delayed very little - i.e., for short inverse digital Wiener filters, the input signal must be minimum phase or nearly so.

A tone burst signal does not meet the criteria for minimum phase, nor is it even nearly so since approximately half the zeros of the z-transform of such a signal are outside the unit circle. In the z domain, an exponential weighting which moves zeros toward the origin will move energy in the time domain toward the start of the pulse as in Figure 4.1. As discussed previously, it is not practical to transform all tone burst signals into minimum phase sequences. It was also previously shown that for long sequences, it is not possible to determine the zero locations precisely. It will be shown later in this chapter that a reasonable method exists for determining the number of zeros outside the unit circle in the z-domain for a given exponential weighting. A natural question to ask is if it is analytically possible to determine the optimum delay for the filter output if the number of zeros outside the unit circle for the input sequence is known. The answer, unfortunately, is no.

If the precise locations of the zeros were known, one could use equations (4.3) and (4.4) to determine inverses for each doublet. Convolving these inverses would leadto the infinite length inverse filter. Of course, in practice, one would allow a certain small error in the inverse and truncate the analytic inverse filter to the minimum number of terms necessary to meet the error criterion. Even this scheme would not allow the selection of an optimum delay for a least-squares estimate of the analytic inverse filter. It is possible and even quite probable that a least-squares analysis will lead to a better estimate of the analytic inverse filter for a given number of terms than truncation of the analytic series. One could find that the number of realizable and non-realizable coefficients which were best for a truncation of the analytic series would not be best for a least-squares estimate.



Mixed Phase Waveform

Weighting Function

Nearly Minimum Phase Waveform

Figure 4.1

Exponential Weighting

There is a special case for which the optimum delay can be determined - that of minimum phase. For minimum phase the analytic inverse filter is realizable. This is easily seen by noting that the convolution of two realizable sequences is realizable as follows:

Let sequence A be given by:

$$A = (\ldots\ 0,\ 0,\ a(0),\ a(1),\ a(2),\ \ldots) \tag{4.6}$$

Let sequence B be given by:

$$B = (\ldots\ 0,\ 0,\ b(0),\ b(1),\ b(2),\ \ldots) \tag{4.7}$$

then

$$A * B = (\ldots\ 0,\ 0,\ a(0)b(0),\ a(1)b(0) + a(0)b(1), \ldots). \tag{4.7}$$

Thus, the optimum delay for the inverse of a minimum phase sequence is zero, since no non-realizable components should be included.

4.3 Convergence

It is possible to qualitatively understand an important effect of exponential weighting by studying the convergence properties of the inverses of exponentially weighted doublets. In section 3.4, it was shown that exponential weighting corresponds to a scaling of the complex variable by α. A weighted doublet would be given by:

$$W = (1, \tfrac{k}{\alpha}). \tag{4.9}$$

If it is assumed that $\alpha > 1$ (i.e., the weighting function is a decaying exponential) then there are four cases.

Case 1. $|k| < 1$ - the minimum phase case. The analytic inverse is given by:

$$h(n) = \begin{cases} 0 & n < 0 \\ (-\frac{k}{\alpha})^n & n \geq 0 \end{cases} \tag{4.10}$$

Comparing equations (4.10) and (4.3), it is easy to see that exponential weighting causes more rapid convergence of the analytic inverse. Less terms would be required to approximate the analytic inverse for a truncated series to within a given error.

Case 2. $|k| > 1$ and $|\frac{k}{\alpha}| > 1$ - the maximum phase case where the exponential weighting does not change the doublet to minimum phase. The analytic inverse is given by:

$$h(n) = \begin{cases} -(\frac{k}{\alpha})^n & n < 0 \\ 0 & n \geq 0 \end{cases} \qquad (4.11)$$

Inspection of equations (4.11) and (4.4) indicates that exponential weighting in this case causes less rapid convergence of the analytic inverse. This is not a desired result.

Case 3. $|k| > 1$ and $|\frac{1}{k}| < |\frac{k}{\alpha}| < 1$ - the maximum phase case where the exponential weighting changes the doublet to minimum phase. The analytic inverse is given by equation (4.10). Comparison of (4.10) and (4.4) indicates that while we have a realizable rather than a non-realizable inverse, the new sequence is less rapidly convergent than the original sequence. This is also not a desirable result.

Case 4. $|k| > 1$ and $|\frac{k}{\alpha}| < |\frac{1}{k}|$ - this is the same as case 3 except the minimum phase inverse will converge more rapidly than the original maximum phase inverse.

To minimize the length of the total inverse filter, it is desirable to only have cases 1 and 4 occur. In practice, this may not be possible for all doublets - particularly those maximum phase doublets that are not converted to minimum phase. Exponential weighting will have the desirable characteristic of resulting in a more rapidly converging analytic inverse filter whenever most of the doublets making up the original signal have their convergence properties improved. It has been found in this study and reported for several other signal types that exponential weighting improves the performance of Wiener filters. It is likely that the major cause of such an improvement is the improved convergence properties just discussed.

4.4 Determining the Number of Zeros

It has now been shown that exponential weighting can improve Wiener inverse filtering of tone burst signals. A natural question is how much weighting is best. It was pointed out in section 3.4 that too much weighting may result in serious problems. A consequence of the previous material in this chapter is that too little weighting will require a lengthy filter. This is true since the filter output must be delayed more as the signal phase characteristic becomes less minimum. An approximate measure of how close a signal is to minimum phase is the number of signal zeros outside the unit circle. If there are no zeros outside the unit circle, then the signal is minimum phase and no delay is required. If all zeros are outside the unit circle, then the signal is maximum phase, and the output delay must be the length of the signal.

In order to determine, then, how nearly minimum phase a given signal is, a procedure will be developed to determine the number of zeros outside the unit circle for a given exponential weighting. To do this use may be made of a result from complex variable theory. To quote from Churchill [20, p. 271]:

> Let a function f be analytic inside and on a closed contour C except for at most a finite number of poles interior to C. Also, let f have no zeros on C and at most a finite number of zeros interior to C. Then, if C is described in the positive sense, it is true that
>
> $$\frac{1}{2\pi i} \int_c \frac{f'(z)}{f(z)} \quad dz = N_o - N_p, \tag{4.12}$$
>
> where N_o is the total number of zeros of f inside C, a zero or order m_o being counted m_o times, and N_p is the total number of poles of f inside C, if a pole of order m_p is counted m_p times.

The complex function f(z) is the z-transform of the basic wavelet.

The proof of this result is given by Churchill and will not be repeated here. The contour integral in (4.12) may be simplified to an argument principle, which can also be found in Churchill. A less formal development of the simplified form will be presented here to provide insight into the application of (4.12).

Assume that the contour C will be chosen as the unit circle. This particular contour corresponds to the Fourier integral and allows the use of the Fast Fourier Transform (FFT) algorithm to increase the speed of evaluation of (4.12) for discrete finite length data. Note that:

$$\frac{f'(z)}{f(z)} = \frac{d}{dz} \text{Log } f(z) \tag{4.13}$$

substituting (4.13) into (4.12) yields:

$$\frac{1}{2\pi j} \int_c \frac{d}{dz} \text{Log } f(z)dz = N_o - N_p \tag{4.14}$$

where $j = \sqrt{-1}$ has been substituted for i to conform to more conventional engineering notation. Restricting (4.14) to the unit circle requires that $z = e^{j\omega}$ and (4.14) becomes:

$$\frac{1}{2\pi j} \int_c \frac{d}{d\omega} \text{Log } f(e^{j\omega})d^{\omega} = N_o - N_p \tag{4.15}$$

F may be represented in polar form:

$$f(e^{j\omega}) = |f(e^{j\omega})| \exp(j \arg(e^{j\omega})) \tag{4.16}$$

where $|f(e^{j\omega})|$ is the magnitude of the Fourier transform and $\arg(e^{j\omega})$ is the phase of the Fourier transform. Since

$$\text{Log } f(e^{j\omega}) = \text{Log} |f(e^{j\omega})| + j \arg[f(e^{j\omega})]. \tag{4.17}$$

(4.15) may be rewritten as:

$$\frac{1}{2\pi j} \int_o^{2\pi} \frac{d}{d\omega} \text{Log} |f(e^{j\omega})| d\omega + \frac{1}{2\pi} \int_o^{2\pi} \frac{d}{d\omega} \arg[f(e^{j\omega})]d\omega = N_o - N_p \tag{4.18}$$

It is easy to show that:

$$\frac{1}{2\pi j} \int_o^{2\pi} \frac{d}{d\omega} \text{Log} |f(e^{j\omega})| d\omega = 0 \tag{4.19}$$

by recognizing that the function $\text{Log}|f(e^{j\omega})|$ is an even function on the interval $[0,2\pi]$. If $f(e^{j\omega})$ is explicitely defined as:

$$f(e^{j\omega}) = f_R(e^{j\omega}) + jf_I(e^{j\omega}) \tag{4.20}$$

then

$$|f(e^{j\omega})| = \sqrt{f_R(e^{j\omega})^2 + f_I(e^{j\omega})^2}. \tag{4.21}$$

From the basic properties of Fourier transforms for real data, it is known that the real part of the transform is an even function, and the imaginary part is an odd function. Using this fact, $|f(e^{j\omega})|$ may be seen to be an even function, and the identity (4.19) readily follows.

Substituting (4.19) into (4.18) yields:

$$\frac{1}{2\pi}\int_{0}^{2\pi}\frac{d}{d\omega}\arg[f(e^{j\omega})]d\omega = N_o - N_p. \tag{4.22}$$

With $f(e^{j\omega})$ defined as in (4.20) and

$$\arg[f(e^{j\omega})] = \tan^{-1}\left(\frac{f_I(e^{j\omega})}{f_R(e^{j\omega})}\right) \quad . \tag{4.23}$$

it is true that $\arg[f(e^{j0})] = 0$ since $f_I(e^{j0}) = 0$. Then (4.22) can be rewritten:

$$\frac{1}{2\pi}\arg f(e^{j2\pi}) = N_o - N_p \; . \tag{4.24}$$

This is the argument form of equation (4.12).

From equation (3.4), it was determined that for a wavelet of length n + 1, there are n poles all located at z = 0 and n total zeros. Since z = 0 is within the chosen contour of integration that led to (4.24), it must be true that:

$$N_p = n. \tag{4.25}$$

If N_o is the number of zeros inside the contour of integration and n is the total number of zeros, then the number of zeros outside the unit circle, $\hat{N}_o$, is:

$$\hat{N}_o = n - N_o \tag{4.26}$$

substituting (4.25) and (4.26) into (4.24) yields the very simple relationship that the number of zeros outside the unit circle is given by a constant times the value of the phase of the Fourier transform of a sequence evaluated at $\omega = 2\pi$

$$\hat{N}_o = - \frac{1}{2\pi} \arg [f(e^{j2\pi})]. \tag{4.27}$$

Unfortunately, this very simple analytic result is relatively complex to evaluate numerically. The evaluation of (4.27) is not simply a matter of substituting real and imaginery parts of the discrete Fourier transform at $\omega = 2\pi$ into (4.23). The problem lies in the fact that the inverse tangent function ($\tan^{-1}$) is multivalued, i.e.,

$$\tan(\theta \pm 2\pi m) = \tan\theta \qquad m = 0, 1, 2, \ldots \tag{4.28}$$

To resolve this ambiguity, inverse tangent functions on a computer, generally return a value limited to the range $\omega = [0,2\pi]$ (equivalently, $\omega = [-\pi,\pi]$). This readily computed phase value is normally called the principal value. The actual phase value is related to the principal value by the relationship:

$$\arg[f(e^{j\omega})] = \text{principal value} \pm 2\pi m; \quad m = 0, 1, 2, \ldots \tag{4.29}$$

The process of computing the phase from the phase principal value has become known as phase unwrapping. An important property of a correctly unwrapped phase is that it should be continuous, except at those values of ω corresponding to the location of a zero in which case a jump of 2π should occur as the zero is crossed.

4.5 Phase Unwrapping

Two fundamentally different approaches have been taken to unwrap phase. Both methods utilize the FFT algorithm with its resultant computational efficiency and its equally spaced values of the Discrete Fourier Transform on the unit circle. One approach, attributable to Schafer [21], involves adding or subtracting multiples of 2π to the principal value until the change in phase between the currently processed phase value and its immediate predecessor is minimized. An essential assumption in this scheme is that the phase between two adjacent phase values never differs by more than 2π. Generally, it is not possible to determine, a priori, the spacing between phase values that can guarantee such a limited change. Furthermore, using only the principal value of the phase, it is not possible to determine whether the phase should increase or decrease, nor is it possible to estimate the magnitude of the phase change.

The second approach involves integration of the phase derivative. The phase derivative may be readily computed by using another basic property of the Discrete Fourier Transform (DFT) - namely that the derivative of the DFT is given by:

$$\frac{d}{d\omega} f(e^{j\omega}) = -j \text{ FT}(kf(k)) \qquad k = 0, 1, \ldots, n-1 \qquad (4.30)$$

where FT indicates the Discrete Fourier Transform and $f(k)$ is the sequence of data being analyzed. An expression for $\frac{d}{d\omega} \arg[f(e^{j\omega})]$

which can readily be obtained in terms of the coefficients of the DFT and the derivative of the DFT [22, p. 170] is:

$$\frac{d}{d\omega}\arg[f(e^{j\omega})] = \frac{f_R(e^{j\omega})\,\frac{d}{d\omega}f_I(e^{j\omega}) - f_I(e^{j\omega})\,\frac{d}{d\omega}\,f_R(e^{j\omega})}{|f(e^{j\omega})|^2}. \tag{4.31}$$

Numerical integration of (4.31) with the initial condition that $\arg[f(e^{j0})] = 0$ yields an estimate of the phase. This integration method has the same approximation problems that are inherent in all digital integration methods - error accumulation. The error accumulation can be particularly severe in this case since fixed step size is mandated by the use of the Fast Fourier Transform (FFT), and once again no a priori estimate of the maximum phase derivative is available to allow pre-selection of a suitable step size. Conventional variable step size methods are not practical due to the excessive computation generally required by a direct calculation of DFT.

A recent phase unwrapping procedure proposed by Tripolet [22] combines these two methods and, when necessary, also implements variable step size to produce an accurate unwrapped phase. This method uses integration to generate an initial estimate of the phase. At each integration step, the integrated value is corrected by using Schafer's method. If the difference between the integration estimate and Schafer's estimate is too great, then the integration step size is halved. Whenever, the step size is modified, the direct computation of the DFT is performed to provide phase and phase derivative estimates. This method only uses direct computation of the DFT when the phase is rapidly changing and appears to be the most efficient and accurate phase unwrapping method currently available. A phase unwrapping problem may occur due to the discontinuity of the phase at zero locations. To prevent the possibility of an attempt to evaluate the phase at a zero location, the integration step size is not allowed to be reduced indefinitely. Should the algorithm attempt to reduce the step size too much, an error return is taken. Because of the adaptive step size, it is very difficult to estimate the computational complexity of applying this technique to a particular data set. It is, however, clear that the procedure is quite efficient in regions of the z plane in which few zeros occur. Fortunately, as will be seen, these are the regions of most interest in determining exponential weight factors.

A modification to Tripolet's method has been proposed by Bonzanigo [23]. This small change involves using Goertzel's algorithm to improve the speed of computation for the Discrete Fourier Transform whenever the step size is modified. A second modification, useful in many applications, is to keep track of the number of times the step size is reduced. This information could then be used to modify the length of similar data sequences being analyzed. Since the speed of the FFT is considerably greater than the DFT, such an adaptive procedure could result in significant reduction in the number of computations required in computing unwrapped phase.

It is possible to utilize (4.27) to determine the number of zeros for the z-transform of a data sequence outside any circular region of radius r centered on $z = 0$. To accomplish this, it is only necessary to exponentially weight the sequence to be analyzed with the exponential factor $\alpha = r$ as in equation (3.22). This weighting will not affect the poles (since they are at $z = 0$), but will move all zeros inside radius r, inside the unit circle, and will leave all zeros outside radius r outside the unit circle. The phase may then be evaluated on the unit circle, assuming no zero is on the unit circle, using Tripolet's phase unwrapping method and equation (4.27) may be applied to determine the number of zeros outside radius r. This procedure is presented in the flow chart of Figure 4.2.

4.6 Weight Estimation

The procedure presented in Figure 4.2 has been used in two different ways for estimating a good value of exponential weighting coefficient α. Both methods involve searching in the z-plane for information concerning the location of zeros. The two methods used were linear and binary search. In both cases, the variable searched was the exponential weight value (equivalent to the radius in the z plane) and the variable determined was the number of zeros outside the radius.

In the linear search method, the value of radius, r, in the z plane was initially set to some minimum value, say r_0, and varied in even increments of Δr to a maximum value, say r_1. At each value of r, the number of zeros outside the radius was determined by the procedure of Figure 4.2. In this way, an estimate of the number of zeros as a function of radius was developed. From this, a minimum value of exponential weighting could be selected such that most of the zeros of the weighted sequence would be inside the unit circle.

In order to test the validity of this method, several data sets with known zero locations in the z domain were generated. The search procedures were tested against these known data sets. Figure 4.3 is

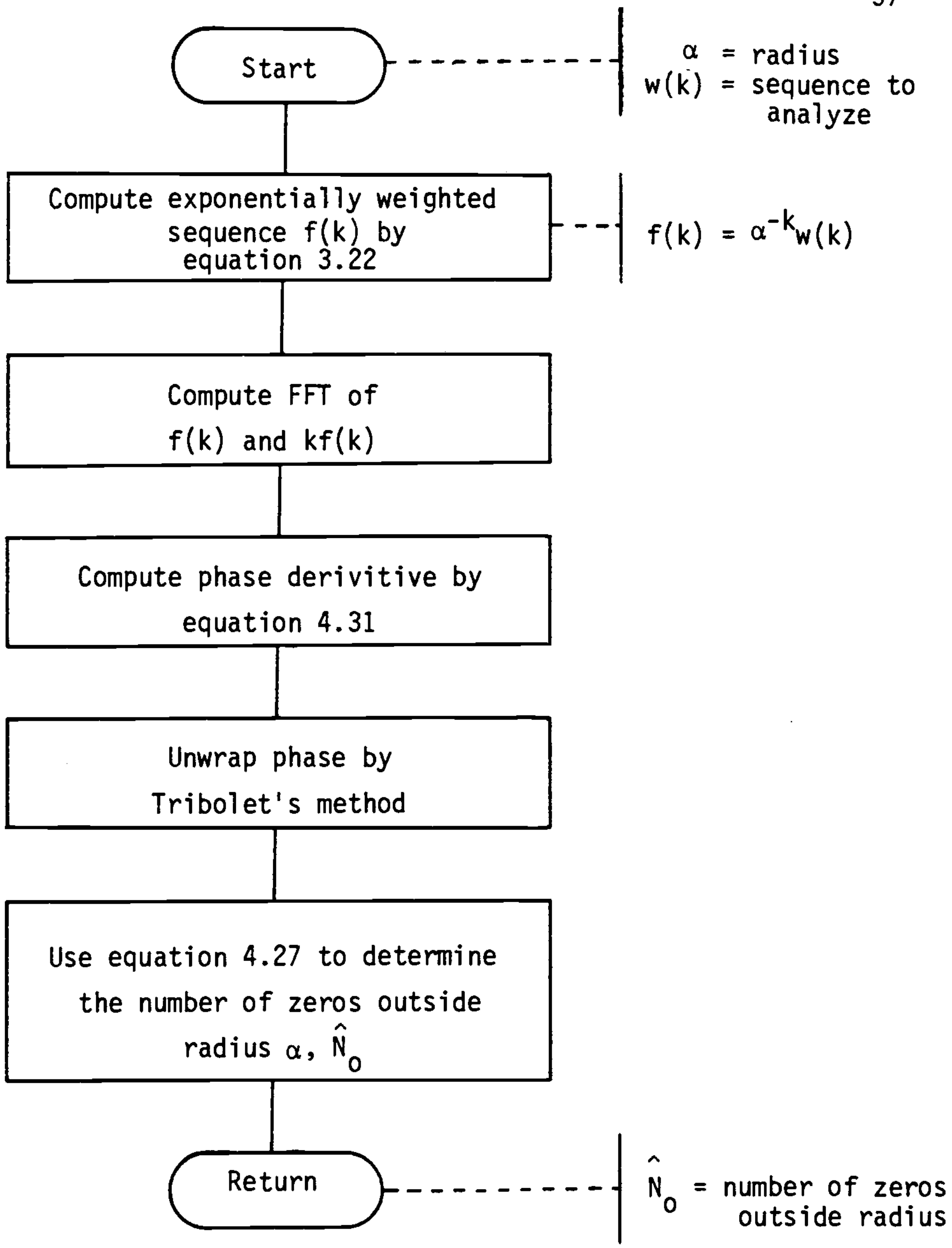

Start
α = radius
w(k) = sequence to analyze
Compute exponentially weighted sequence f(k) by equation 3.22
f(k) = α^-k w(k)
Compute FFT of f(k) and kf(k)
Compute phase derivitive by equation 4.31
Unwrap phase by Tribolet's method
Use equation 4.27 to determine the number of zeros outside radius α, N̂o
Return
N̂o = number of zeros outside radius

Figure 4.2

Zero Location Estimate Flowchart

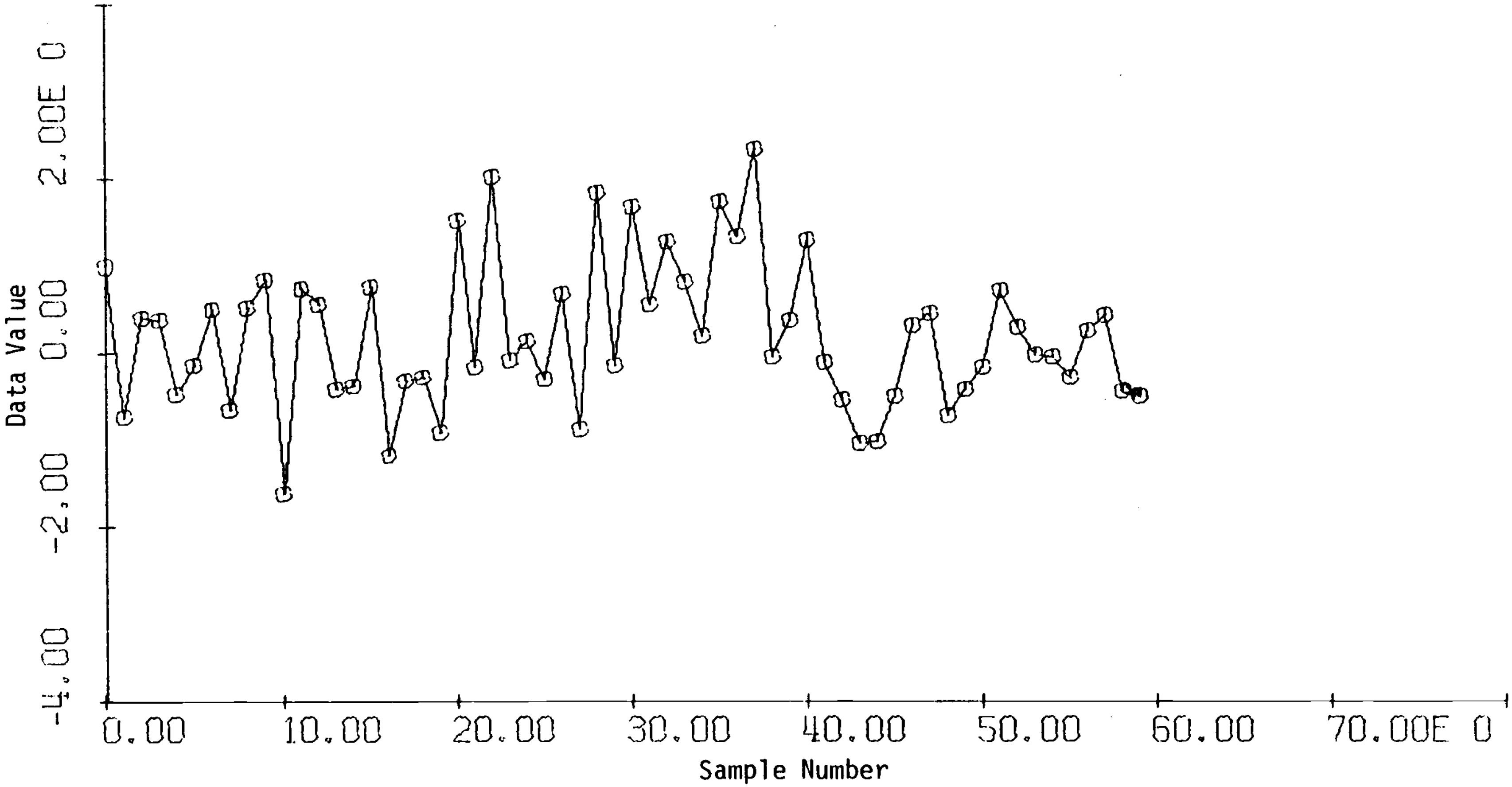


Figure 4.3

Example Data Used for Testing

the time series for one such example. In this example, the data to be analyzed was selected such that zeros of the z transform occurred with uniform probability between radius .9 and 1.1 in the z plane. The locations were selected such that zeros occurred evenly spaced in angle around the the unit circle. Zeros were chosen as conjugate pairs or as single zeros on the real axis to insure that the artificial data was not complex. Table 4.1 displays the set of zero locations corresponding to the data of Figure 4.3 Table 4.2 is the result of a FORTRAN program which implemented the linear search algorithm on this data. The program was run on a CDC Cyber 70 Model 73 computer. Table 4.3 summarizes the computer time <u>per search step</u> required to implement the algorithm for a few examples.

The first entry in Table 4.3 corresponds to the example of Figure 4.3. For the 40 search steps required to generate Table 4.2, a total of 13.2 seconds of CPU time was required.

The purpose of the binary search method is to determine the minimum radius in the z plane which contains a given number of zeros of the z transform. A flow chart of the procedure is presented in Figure 4.4. As with the linear search, the range of the search is from some minimum radius in the z plane, r_0, to a maximum radius, r_1. In this case, however, the actual radius searched is the midpoint of the radii known to contain the searched for value. In this way, the interval to be searched is reduced by one-half each search step with the resultant rapid convergence characteristic of binary searches. This procedure can be a significant computational savings since the cost per search step is the same as the linear search but significantly less

Table 4.1. Zero Locations for Test Data.

Zero Pair	Radius	Angles (radians)
1	0.919	0,3.142
2	1.090	±0.105
3	0.902	±0.209
4	1.053	±0.314
5	1.083	±0.419
6	0.967	±0.524
7	0.953	±0.628
8	1.097	±0.733
9	0.968	±0.838
10	0.953	±0.943
11	1.073	±1.047
12	1.083	±1.152
13	1.086	±1.257
14	0.961	±1.361
15	1.002	±1.466
16	0.940	±1.571
17	1.026	±1.676
18	1.087	±1.780
19	0.948	±1.885
20	0.942	±1.990
21	0.934	±2.094
22	0.907	±2.199
23	0.930	±2.304
24	1.059	±2.409
25	0.939	±2.513
26	1.057	±2.618
27	1.017	±2.723
28	0.910	±2.827
29	0.907	±2.932
30	1.067	±3.037

Table 4.2. Results of Linear Search.

Radius in z-Plane	Number of Zeros Outside Radius
.900	60
.905	58
.910	54
.915	52
.920	50
.925	59
.930	48
.935	46
.940	42
.945	40
.950	38
.955	34
.960	34
.965	32
.970	28
.975	28
.980	28
.985	28
.990	28
.995	28
1.000	28
1.005	26
1.010	26
1.015	26
1.020	24
1.025	24
1.030	22
1.035	22
1.040	22
1.045	22
1.050	22
1.055	20
1.060	16
1.065	16
1.070	14
1.075	12
1.080	12
1.085	8
1.090	2
1.095	2
1.100	0

Table 4.3. Computer Time for Some Linear Search Examples

Data Length	FFT Size	CPU Seconds
61	128	0.33
101	256	0.53
201	256	0.92

Table 4.4. Results of Binary Search

Search Step	Radius	Zeros Outside Radius
	.9000	60
	1.1000	0
1	1.0000	28
2	1.0500	22
3	1.0750	12
4	1.0875	4
5	1.0813	12
6	1.0844	8
7	1.0828	12
8	1.0836	8
9	1.0832	10
10	1.0830	10

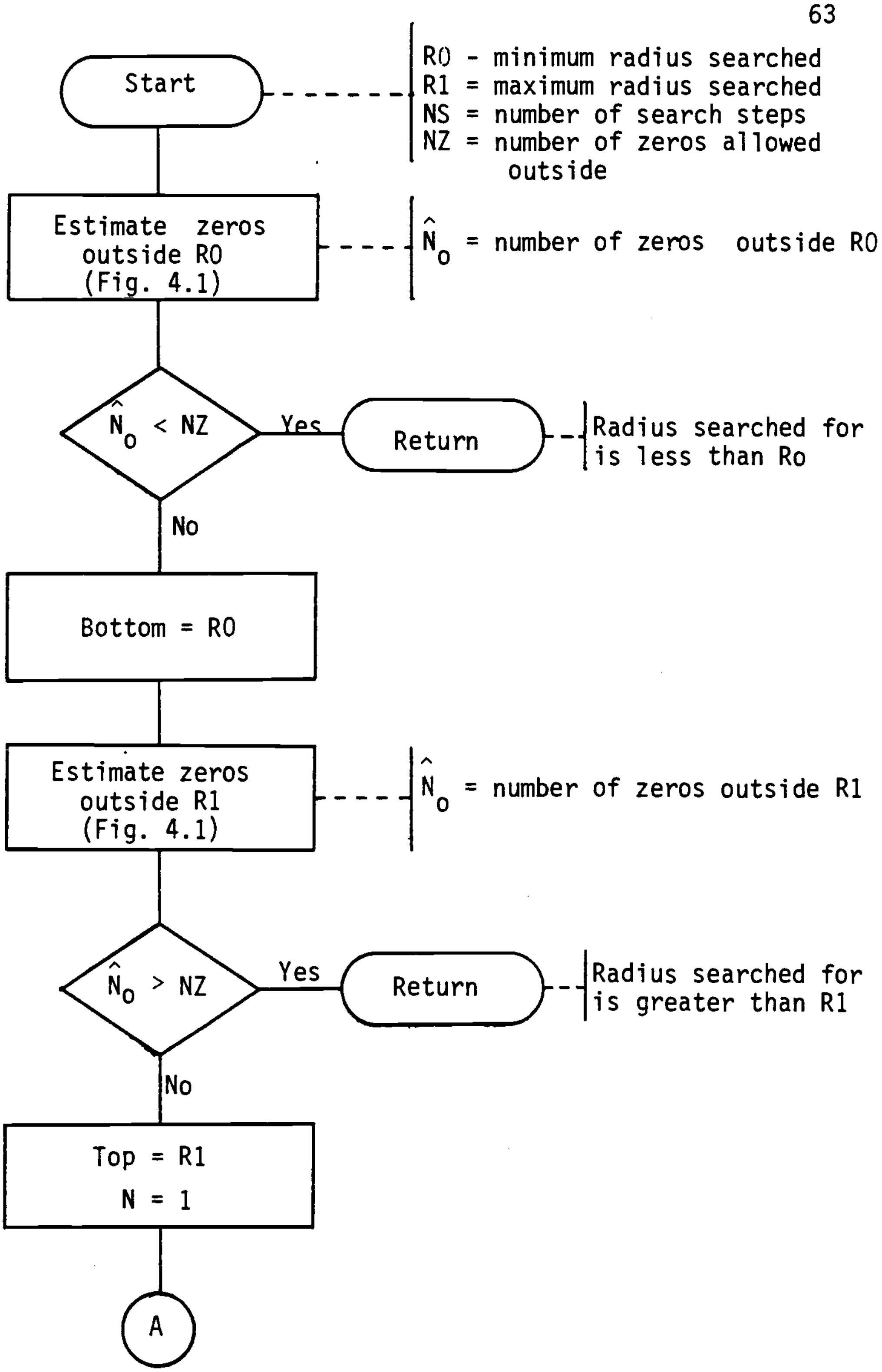

Start
R0 - minimum radius searched
R1 = maximum radius searched
NS = number of search steps
NZ = number of zeros allowed outside
Estimate zeros outside R0 (Fig. 4.1)
$\hat{N}_o$ = number of zeros outside R0
$\hat{N}_o$ < NZ
Yes
Return
Radius searched for is less than Ro
No
Bottom = R0
Estimate zeros outside R1 (Fig. 4.1)
$\hat{N}_o$ = number of zeros outside R1
$\hat{N}_o$ > NZ
Yes
Return
Radius searched for is greater than R1
No
Top = R1
N = 1
A

Figure 4.4

Binary Search Method

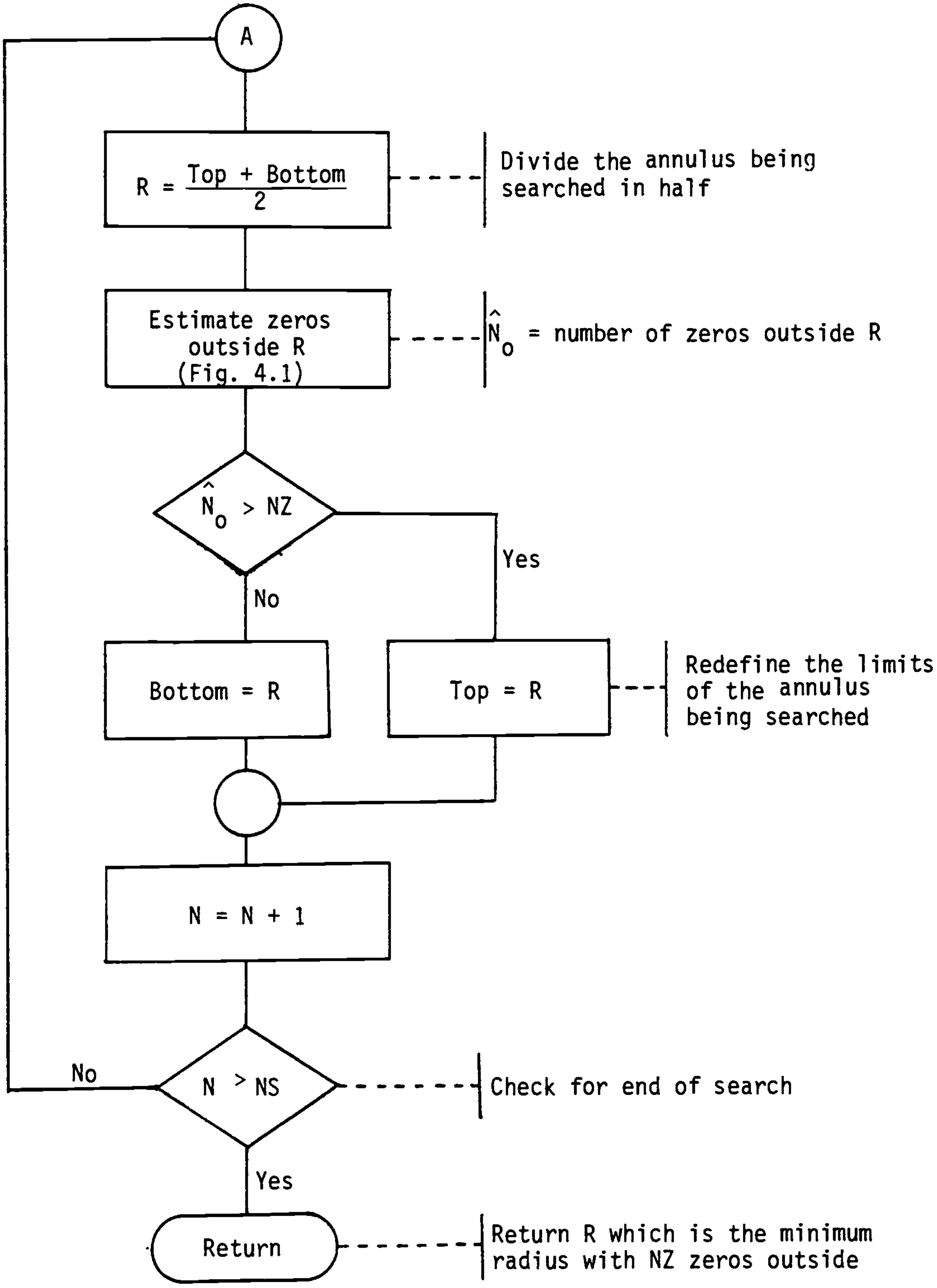

A
$R = \frac{\text{Top + Bottom}}{2}$
Divide the annulus being searched in half
Estimate zeros outside R (Fig. 4.1)
$\hat{N}_o$ = number of zeros outside R
$\hat{N}_o > NZ$
Yes
No
Bottom = R
Top = R
Redefine the limits of the annulus being searched
N = N + 1
No
N > NS
Check for end of search
Yes
Return
Return R which is the minimum radius with NZ zeros outside

Figure 4.4 (cont.)

Binary Search Method

search steps are required. Table 4.4 presents the results of such a binary search for the data of Figure 4.3. The desired number of zeros outside the searched for radius was 10.

V. WEIGHTED WIENER FILTERING

5.1 Introduction

In Chapter III, it was shown that the direct application of digital Wiener filtering to tone burst signals was not appropriate. To improve the performance of Wiener filters for this signal type, exponential weighting was proposed in Chapter IV. Also, a procedure for selecting a reasonable value of exponential weighting was presented. It is the purpose of this chapter to present a method utilizing exponential weighting and digital Wiener filtering for tone burst deconvolution and to demonstrate this method on some typical data.

5.2 Procedure

The addition of exponential weighting to digital Wiener filtering adds a third step to the application of these filters as presented in Figure 2.1. The three steps of the process become:

1. Exponential Weight Estimation
2. Filter Design
3. Filter Application

The remainder of this section is devoted to detailing these steps.

5.2.1 Exponential Weight Estimation

Phase unwrapping has been presented in section 4.5 as a reasonable method for estimating the optimal value of exponential weighting, α. With this technique, zero distribution as a function of radius in the z plane may be computed. This may be done either by linear search or binary search (Figure 4.4).

Ideally, these search techniques can be used to transform the data sequence to be deconvolved into a minimum phase sequence by selecting α such that all zeros of the weighted sequence will be inside the unit circle. Due to numerical range limitations, this is not always possible and α must be selected in such a way that the weighted sequence is "nearly" minimum phase. In these cases, the determination of α is not straight forward and considerable intuition based on experience and heuristics must be used.

5.2.2 Filter Design

A block diagram of the steps in the filter design phase is presented in Figure 5.1. The two results of this step are the filter impulse response, H, and an amplitude scale factor, C. The information required is the basic wavelet to be deconvolved, W, and the exponential weight, α, determined from step 1. The 5 parts of Figure 5.1 will now be discussed in some detail.

1. Exponential weight - the computation carried out at this stage is:

$$w_e(k) = \alpha^{-k} w(k) \qquad k = 0, 1, \ldots, n \qquad (5.1)$$

where: α is the exponential weight factor

w is the basic wavelet of length n+1

2. Filter design - the design implemented at this stage is the standard computation discussed in Chapter II. The implementation may consist of Robinson's SPIKE [6, p. 79] or SPIKER [6, pp. 80-82]

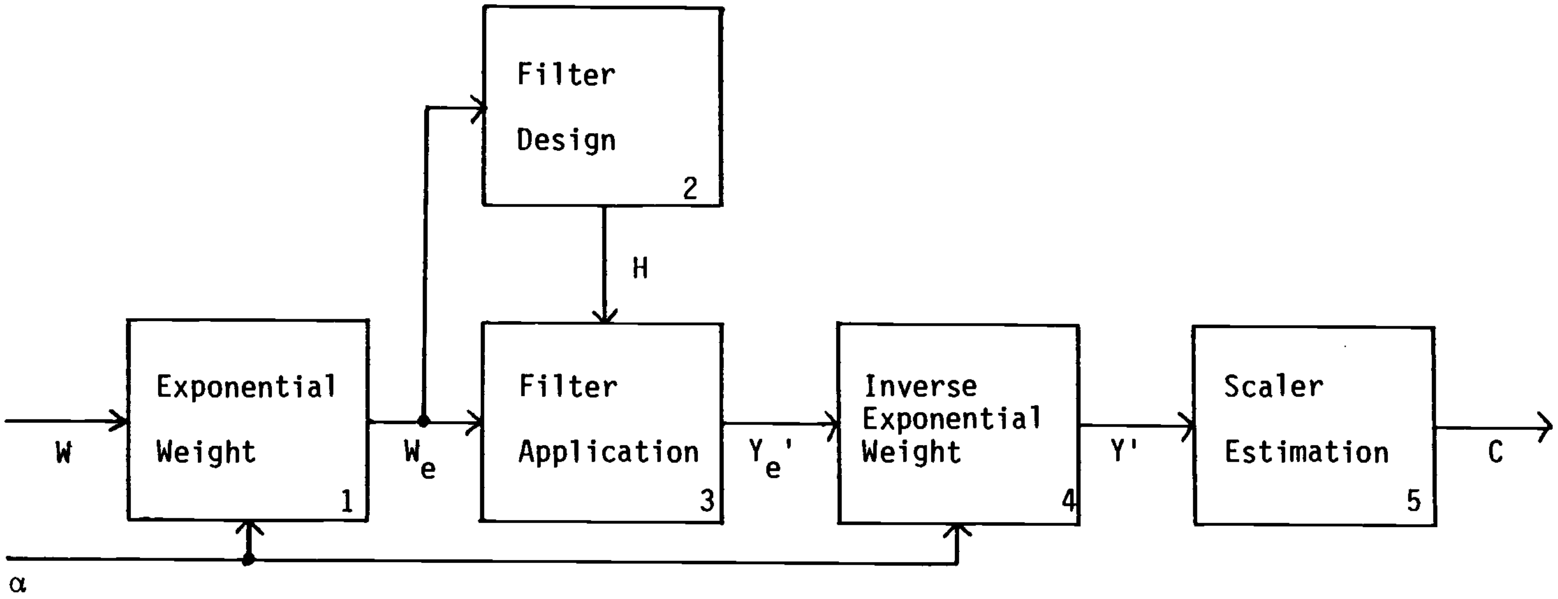


Figure 5.1

Filter Design

subroutines. These routines generate and solve the normal equations (equation 2.9) for all possible lags of the input. Considerable savings in computation may be realized by using knowledge of the wavelet developed by the phase unwrapping used to estimate exponential weight factors. If the exponentially weighted signal is minimum phase, then the optimal wavelet delay is zero and no searching is required. For nearly minimal phase weighted wavelets, only the first few lags need be computed. As long as the weighted wavelet is not maximum phase, then certainly it is not necessary to search for any lags greater than the filter length.

3. Filter application - the computation at this step is discrete convolution since the filter is a finite impulse response:

$$y_e'(k) = \sum_{j=0}^{m} h(j)w_e(k-j) \qquad k = 0, 1, \ldots, n+m \qquad (5.2)$$

where: H is the filter of length m+1

4. The computation at this step is the inverse of equation 5.1:

$$y'(k) = \alpha^k y_e(k) \qquad k = 0, 1, \ldots, n \qquad (5.3)$$

5. This step is necessary for two reasons. First, the least squares filter design minimizes the mean squared error and does not guarantee that the amplitude for the spike output for the basic wavelet is one. Second, if delay is required for the optimal filter design, then the inverse exponential weighting will not produce the desired unity amplitude estimate of the output Y. This scaler output may generally be computed by

$$c = \frac{1}{y'(\ell)} \text{ where } \ell \text{ is the optimal lag.} \qquad (5.4)$$

5.2.3 Filter application

A block diagram of the steps in the filter application phase is presented in Figure 5.2. The result of this step is an estimate of the reflector series R(k) in equation 1.4. The input is the composite signal S(k) in equation 1.3 and the exponential weight determined from step 1. The steps in implementing the filter application are:

1. Exponential weight - this step matches the composite signal to the basic wavelet by applying the same weighting. The computation is:

$$s_e(k) = \alpha^{-k} s(k) \qquad k = 0, 1, \ldots, n \tag{5.5}$$

The result of this computation is to weight each individual wavelet by a scale factor which is a function of the wavelet delay and the appropriate exponential weight sequence. This can be seen by expressing the composite signal, S, as a set of delayed wavelets, W, similar to equation (1.2) as follows:

$$S(k) = \sum_{i=1}^{N} b_i W(k-\tau_i) \tag{5.6}$$

Substituting (5.6) into (5.5) yields

$$S_e(k) = \sum_{i=1}^{N} b_i \ \alpha^{-k} \ W(k-\tau_i) \tag{5.7}$$

Any weighted W in (5.7) may be written as:

$$W_e(k) = \alpha^{-\tau_i} \ [b_i \alpha^{-(k-\tau_i)} W(k-\tau_i)] \tag{5.8}$$

From equation (5.8) it can be seen that the effect on individual delayed wavelets is to scale these wavelets by $\alpha^{-\tau_i}$ where τ_i is the wavelet delay.

2. Filter application - the computation at this step is the convolution of the filter designed and the weighted composite signal.

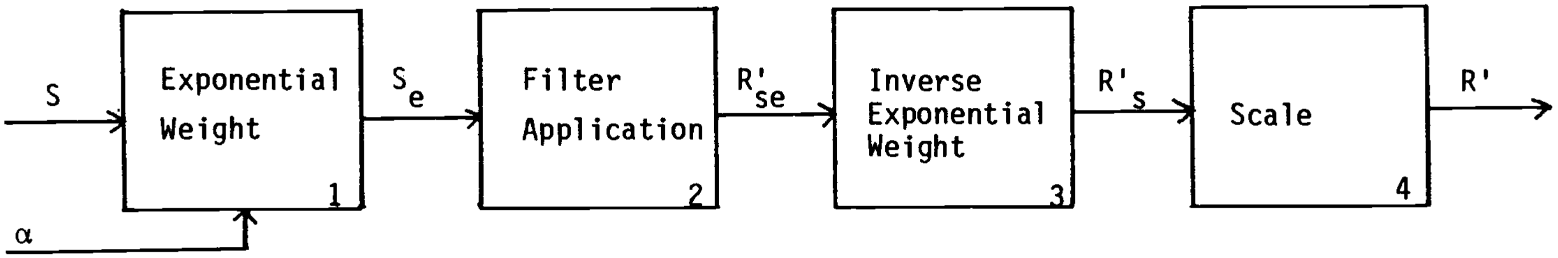


Figure 5.2

Filter Application

$$r'_{se}(k) = \sum_{j=0}^{m} h(j)s_e(k-j) \qquad k = 0, 1, \ldots, n+m \tag{5.9}$$

The result of this computation is a weighted and scaled estimate of the reflector series, R.

3. Inverse exponential weight - the computation at this step is:

$$r_s'(k) = \alpha^k \, r'_{se}(k) \qquad k = 0, 1, \ldots, n+m \tag{5.10}$$

4. Scale - the last step is to correct the scale error discussed in section 5.2.2. The computation is:

$$r'(k) = cr_s'(k) \qquad k = 0, 1, \ldots, n+m \tag{5.11}$$

and finally the estimated reflector series has been generated.

5.3 Examples of Tone Burst Deconvolution

The purpose of this section is to present examples of the application of the procedure described in section 5.2. The examples were chosen to demonstrate applications which as nearly as practical emulate realistic data and summarize the results of the many tests run to develop these procedures.

5.3.1 Example 1 (Artificial Data)

As a first example, the analysis of a short tone burst will be considered. A wavelet, W, of length 33 and comprising 5 cycles of a sinusoid was selected and is plotted in Figure 5.3a. This wavelet was short enough to allow direct computation of the zero locations and these are listed in Table 5.1. A weight sequence, E, with α = 1.01

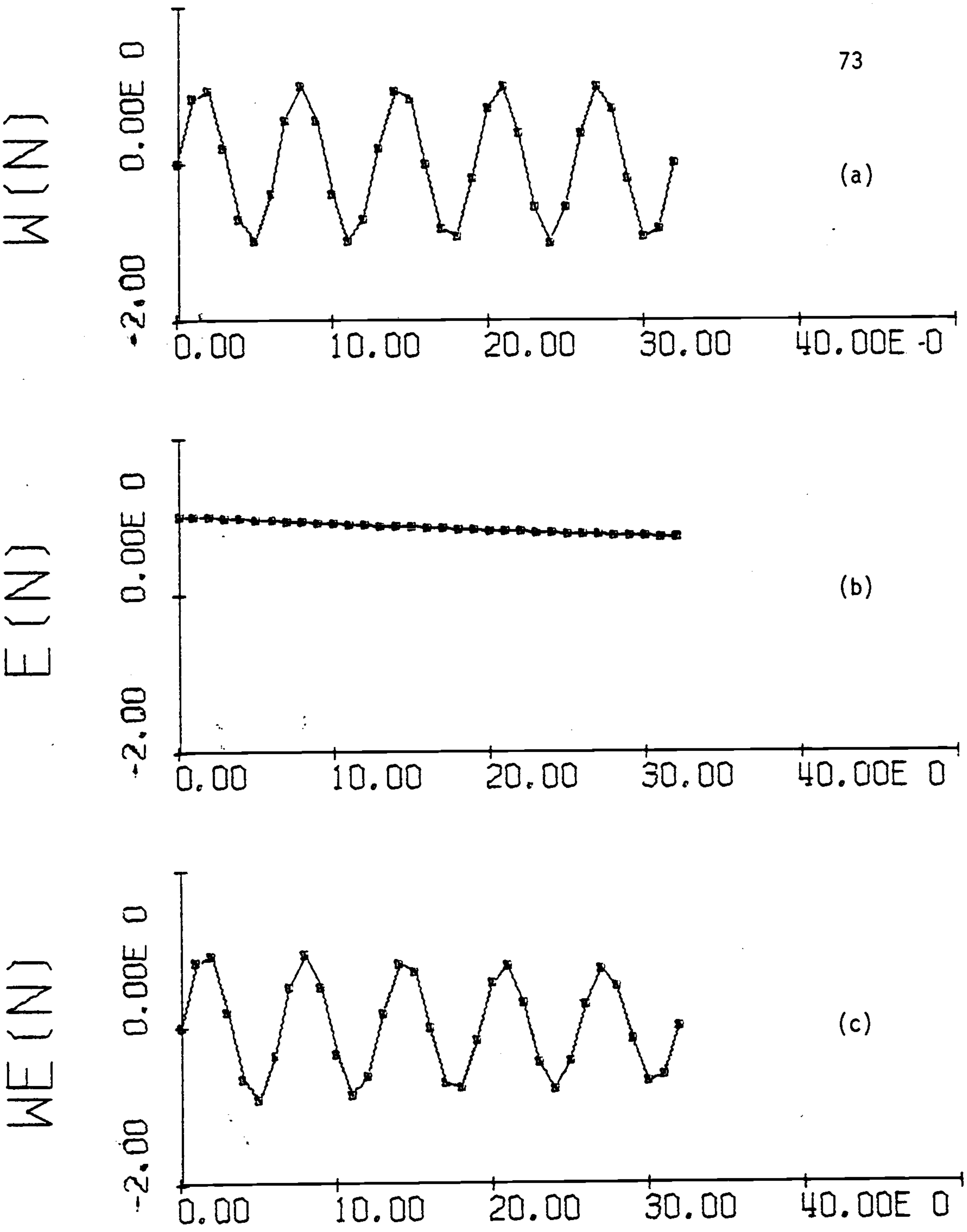


Figure 5.3

Example 1

Basic Wavelet		Zero Locations	
k	W(k)	R	Theta (Radians)
0	.0000	1.415×10^{-5}	0
1	.8315	3.866×10^{6}	0
2	.9239	1.0	0
3	.1951	1.0	± .1963
4	- .7071	1.0	± .3927
5	- .9808	1.0	± .5890
6	- .3827	1.0	± .7854
7	.5556	1.0	±1.178
8	1.0000	1.0	±1.347
9	.5556	1.0	±1.571
10	- .3827	1.0	±1.767
11	- .9808	1.0	±1.963
12	- .7071	1.0	±2.160
13	.1951	1.0	±2.356
14	.9239	1.0	±2.553
15	.8315	1.0	±2.749
16	- .5722	1.0	±2.945
17	- .8315	1.0	3.142
18	- .9239		
19	- .1951		
20	.7071		
21	.9808		
22	.3827		
23	- .5556		
24	-1.0000		
25	- .5556		
26	.3827		
27	.9808		
28	.7071		
29	- .1951		
30	- .9239		
31	- .8315		
32	.0000		

Table 5.1

Values and Zero Locations for Example 1

was selected and is plotted in Figure 5.3b. The weighted wavelet, W_e, is plotted in Figure 5.3c. The weighted sequence was not minimum phase due to the single zero remaining outside the unit circle after weighting. Because of the non-minimum phase characteristic of the weighted wavelet, it was necessary to search for the optimum filter lag which was determined to be a delay of one sample. A one sample delayed spike is then the desired filter output. A filter, H, was designed by solving the normal equations. The three samples comprising the impulse response of this filter are plotted in Figure 5.4a. The error for this filter (see equation 2.17 for the error measure definition) was 0.34. The result of applying H to W_e and inverse weighting is plotted in Figure 5.4c. The scale factor necessary to produce a spike output of amplitude 1.0 was 1.49.

It is interesting at this point to present some observations concerning expected Wiener filter characteristics for tone burst waveform deconvolution. The form of the filter is generally a weighted derivative which produces a spike output upon detection of the leading and trailing edges. This feature may be desirable for many applications since it allows an estimate to be produced of the tone burst duration. The filter has minimal response at the frequence of the tone burst as is demonstrated by the plot of the frequency response of the filter in Figure 5.4b.

To test the designed filter, an artificial reflector series, R, was generated and is plotted in Figure 5.5a. The basic wavelet of Figure 5.3a was convolved with this reflector series to generate the composite signal, S, of Figure 5.5b. The procedure of section

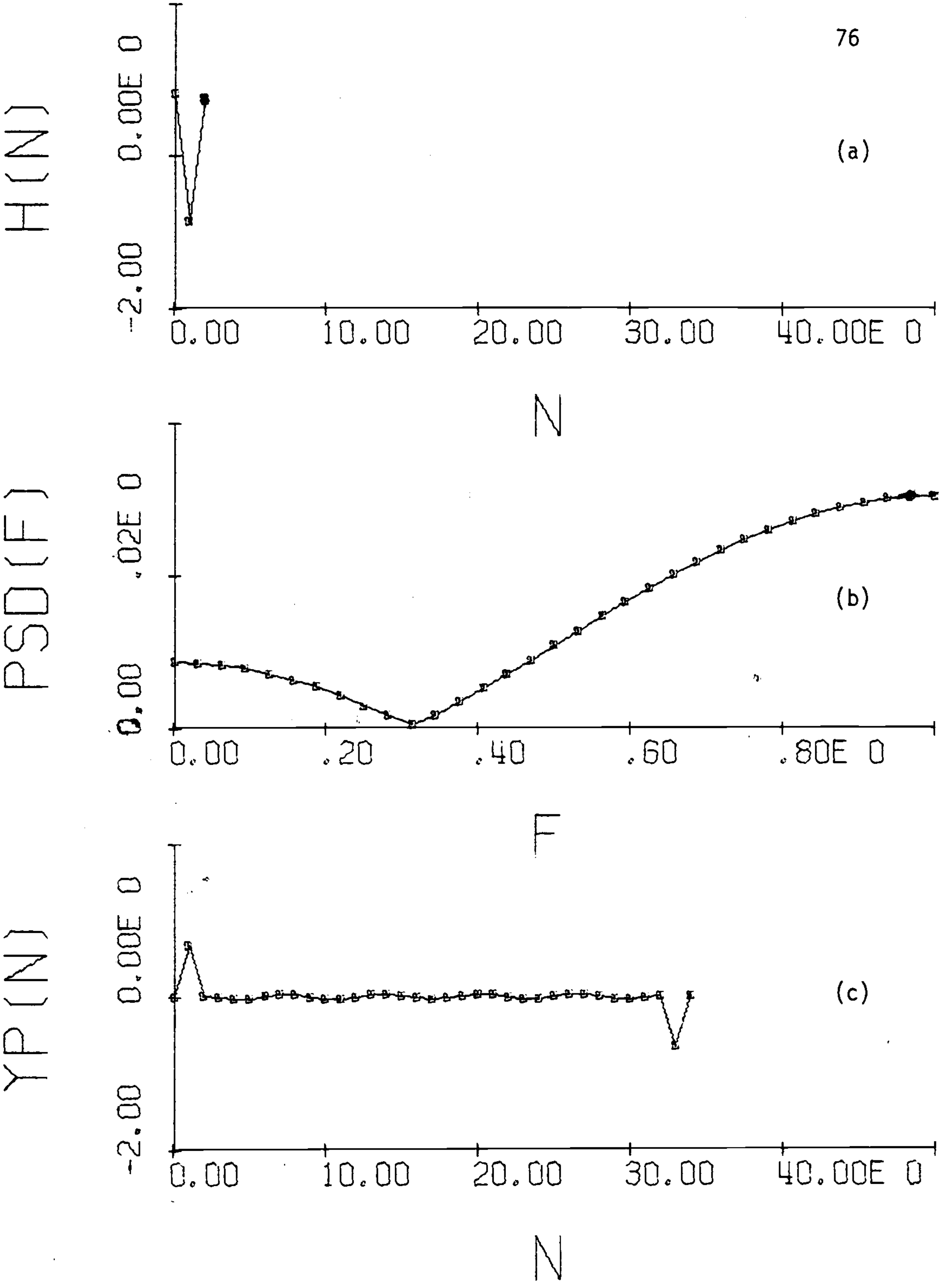


Figure 5.4

Example 1 (Cont.)

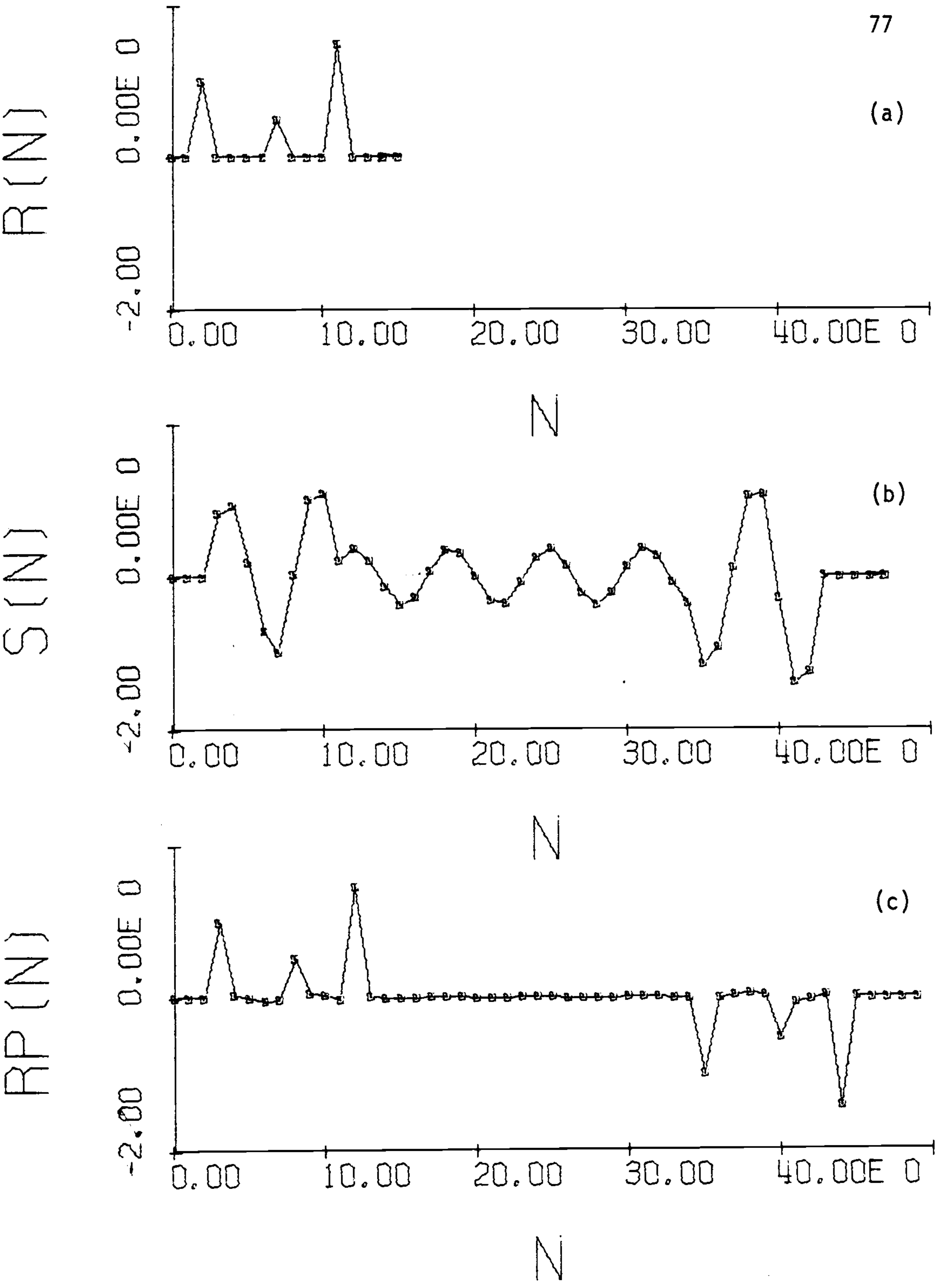


Figure 5.5

Example 1 (Cont.)

5.2.3 was then applied to S and the resultant reflector series estimate R' is plotted in Figure 5.5c. The following summarizes the results of this example:

	Actual Values		Estimated Values	
Echo Number	Magnitude	Delay	Magnitude	Delay
1	1.00	2	1.0	2
2	0.50	7	0.51	7
3	1.50	11	1.45	11

It should be noted that Figure 5.5c includes the 1 sample delay of the filter.

Generally, it appears that these filters, when correctly applied to noise-free overlapping tone burst signals have several characteristics. The delay of the tone burst can be accurately estimated - at least when the delay is an even multiple of the sample interval. The sample interval, of course, has a significant impact on the accuracy of delay estimation. The estimate of the magnitude is not as accurate in general. Interference from overlapping signals has some effect on the estimated magnitude when the filter length is greater than the spacing between echo occurrences. The actual computation of confidence intervals for these parameters in the noisy case is not within the scope of this thesis.

5.3.2 Example 2 (Real Data)

For this example, a sonar pulse collected during an in water test is used as the basic wavelet. This pulse is presented in Figure 5.6. As with all real signals generated by imperfect hardware and

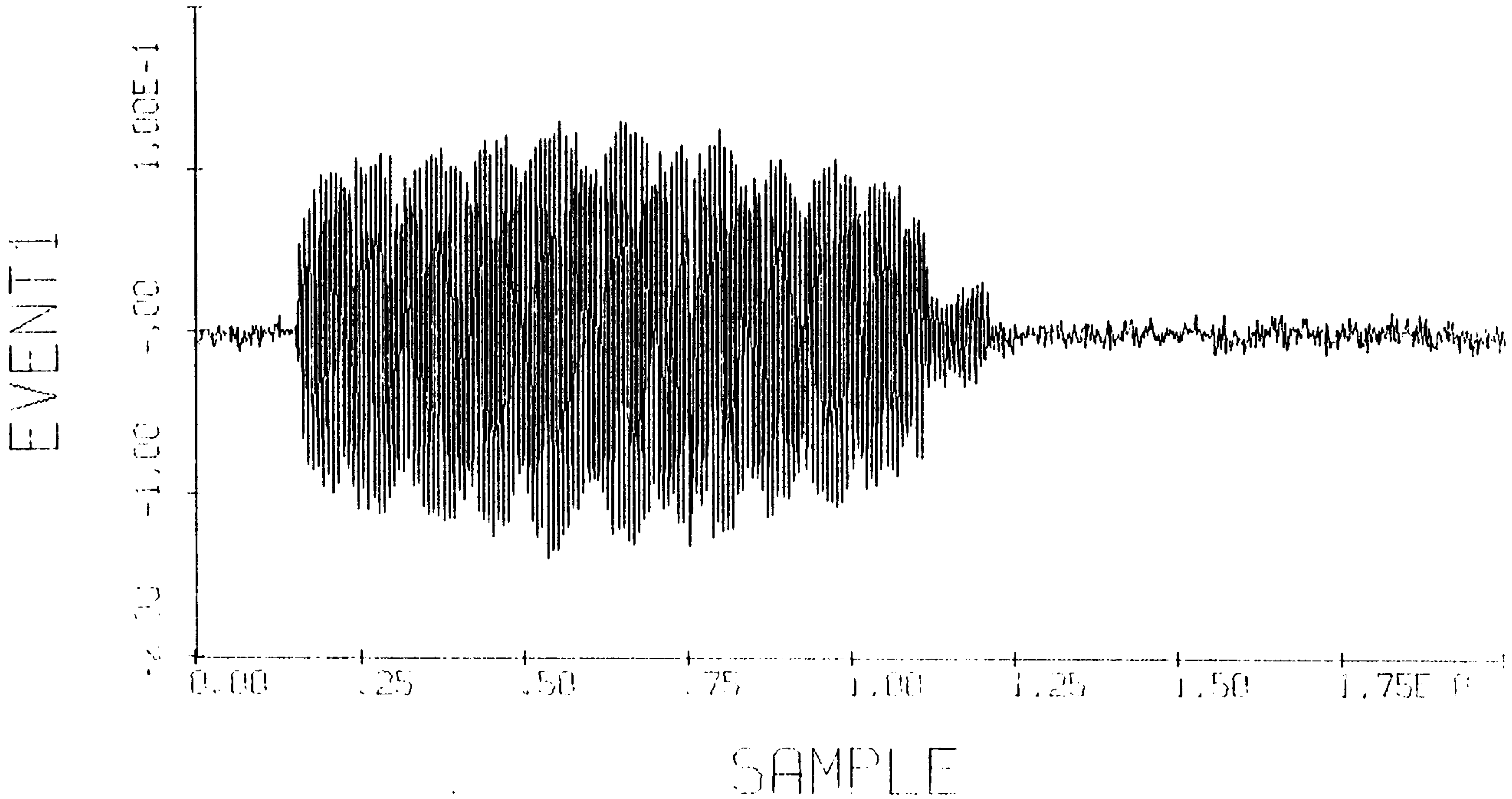


Figure 5.6

Sonar Pulse (Low Sampling Rate)

transmitted through complex environments, the pulse of Figure 5.46 deviates from the ideal tone-burst signals in a number of ways. The trailing edge of the pulse shows a considerable amount of noise due to volume reverberation. This noise is generally caused by a large number of very low energy reflections from biomass or other objects in the near vicinity of the receiving hydrophone. About two-thirds of the way through the pulse, a phase reversal occurs. This event is most likely a result of the transmitting hardware but the specific cause for it is unknown to the author.

It should be pointed out that the apparent amplitude modulation on the tone burst envelope is a result of the plotting method, not a real characteristic of the waveform. The plots were generated by connecting samples of the waveform by straight lines. Since the signal was sampled at only four times the highest frequency, this method of reconstruction is very crude, although sufficient in many respects. Figure 5.7 is the same signal as Figure 5.6 except the interpolation method of Crochiere and Rabiner [24] has been used to generate a sampling rate of 16 times the highest frequency. The plotting method of Figure 5.7 is still to connect sample points by straight lines but this figure appears to have much less amplitude modulation than Figure 5.6.

The basic waveform of Figure 5.6 was used to generate a simulated reflection from four separate reflectors. The characteristics of the reflectors were:

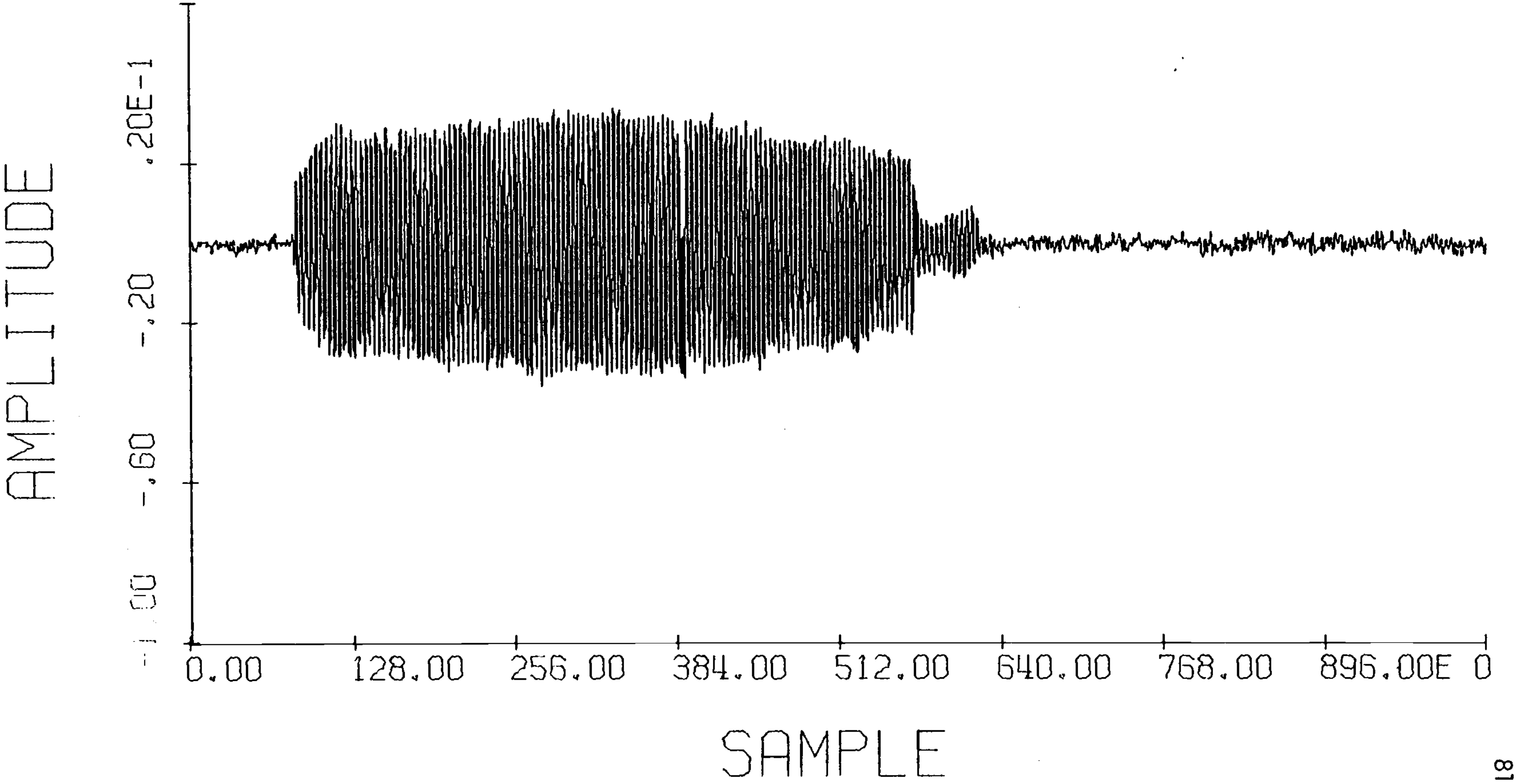


Figure 5.7

Sonar Pulse (High Sampling Rate)

Reflector	Delay (Samples)	Amplitude
1	1	1.0
2	50	1.2
3	80	1.5
4	120	0.5

Prior to generating the artificially reflected signal, the leading edge of the pulse was determined and the samples prior to this time were discarded. This was done since each sample prior to start of the signal results in a zero in the z-plane at infinity. Such zeros are not desirable when design of the filter is attempted and may result in larger than necessary filters being generated unnecessarily. Figure 5.8 shows the basic wavelet, the reflector series and the artificial reflected signal. The reflected signal, S, is the result of the convolution of the reflector series, R, and the basic wavelet, W, as presented previously in this thesis. That is:

$$S(n) = R(n) * W(n) \tag{5.12}$$

To determine a good exponential weighting for this wavelet, the linear search method of sections 4.5 and 4.6 were applied to the data. Selected results showing the number of zeros outside a given radius are presented in Table 5.2.

To demonstrate the improvement in filter performance due to exponential weighting, filters were designed with $\alpha = 1.00$, $\alpha = 1.01$ and $\alpha = 1.02$. A plot of the normalized mean squared error for the filter as a functionof filter length for these three cases is presented in Figure 5.9. It is evident from this figure that a significant

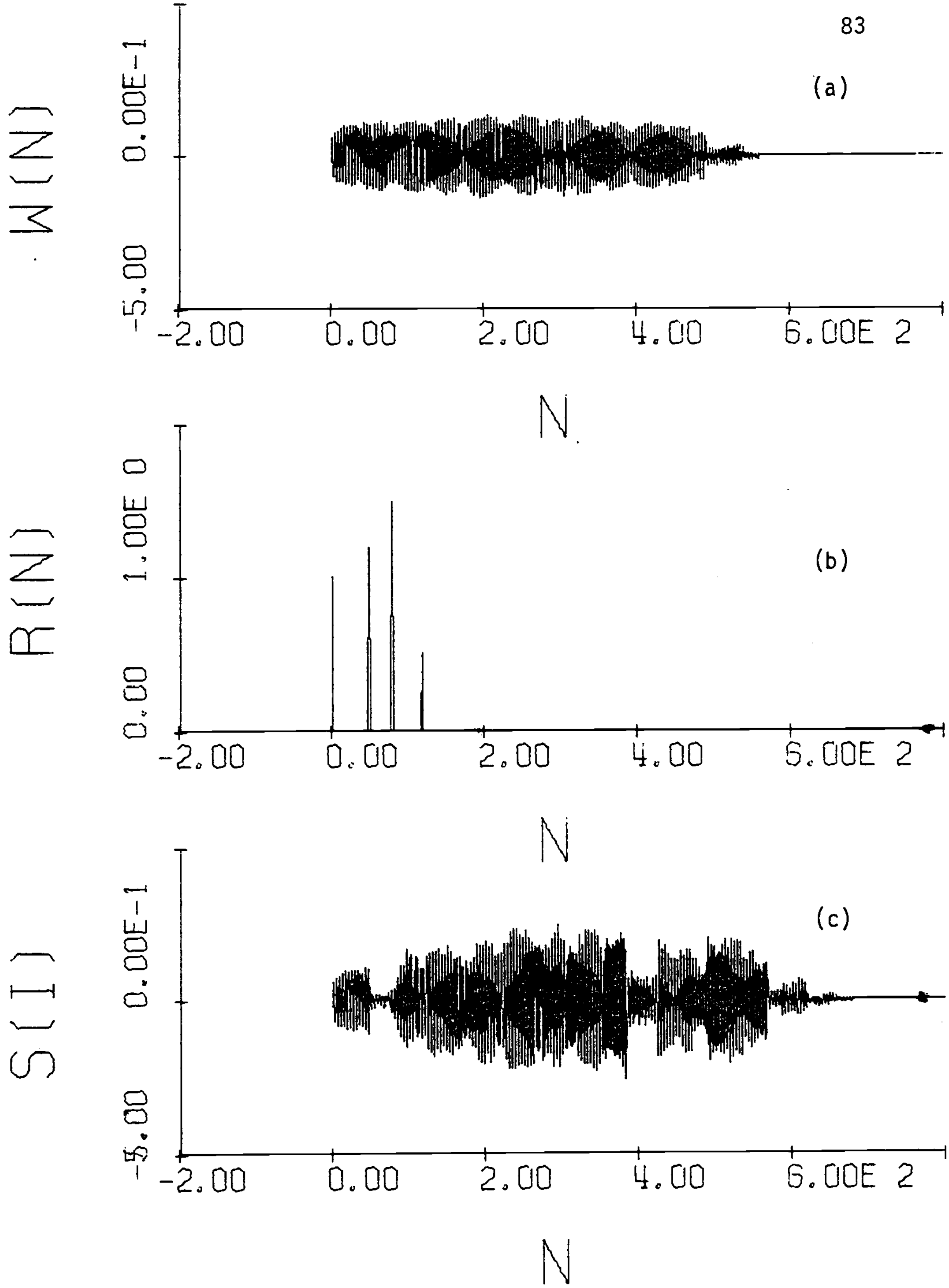


Figure 5.8

Example 2

Table 5.2. Z-Plane Zeros as a Function of Radius for Example 2.

Z-Plane Radius	Number of Zeros Outside
1.000	319
1.010	31
1.012	15
1.014	15
1.016	9
1.018	7
1.020	3
1.022	3
1.024	3
1.026	3
1.028	1
1.030	1
1.100	1

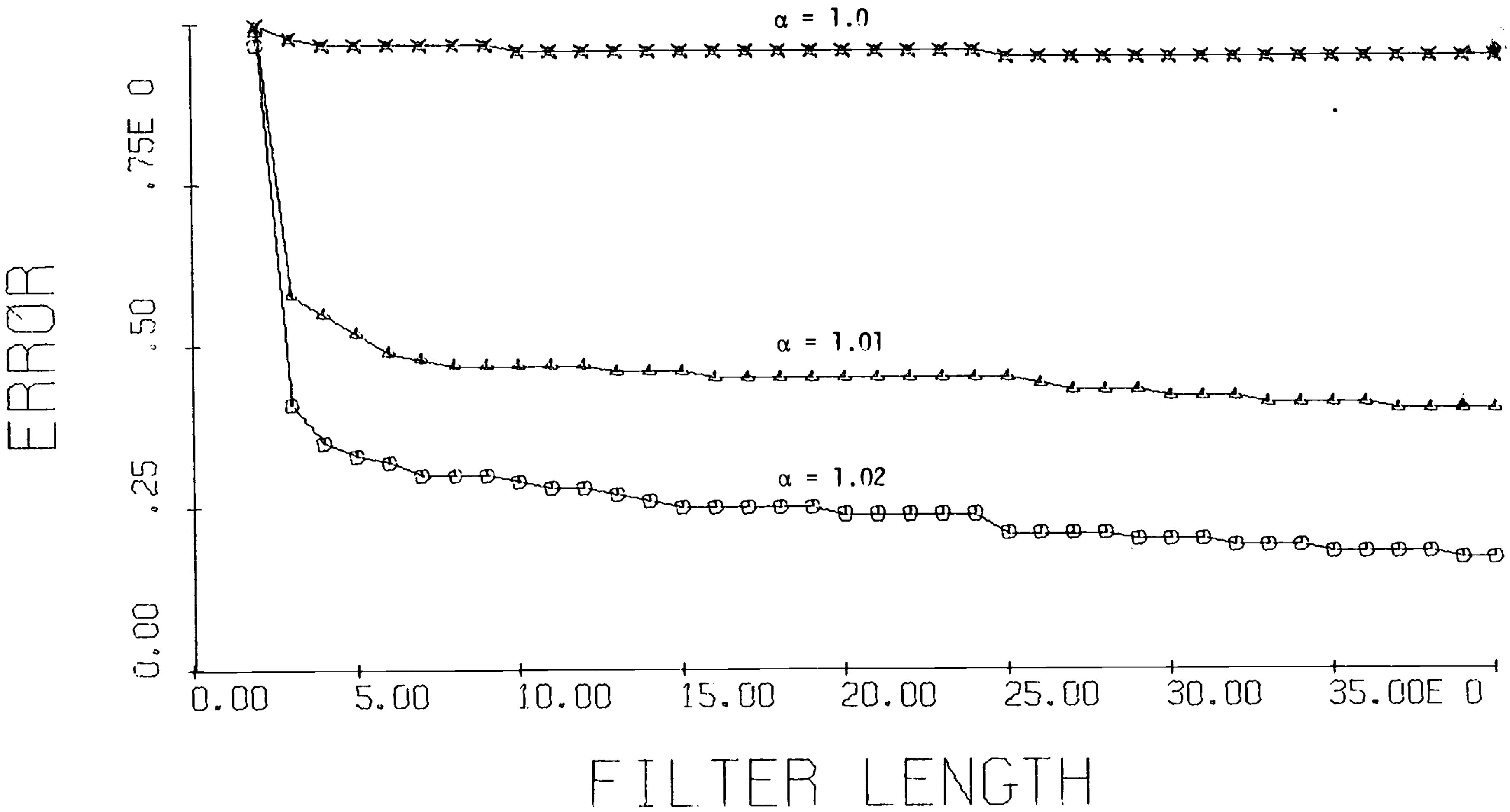


Figure 5.9

Error Versus Filter Length for Various α.

improvement in filter performance is derived from exponential weighting for this example.

The weighting value of $\alpha = 1.01$ was selected as a compromise between good filter error and numerical error due to the weighting of samples occuring near the end of the wavelet ($(1.02)^{600} = 1.4 \times 10^5$ while $(1.01)^{600} = 392$). Filters of length 8 and 150 were designed and are presented in Figure 5.10 and 5.11 respectively.

Figure 5.10 is the optimal length 8 filter for deconvolution of the basic wavelet of this example. Figure 5.10b is the frequency characteristic of this filter. For 5.10b and 5.11b, the independent axis has been normalized such that a value of .5 corresponds to the sample frequency divided by two. The other half of this characteristic is not presented since it is symmetric and easily derived from this figure if necessary. Figure 5.10c is the estimate of the reflector series generated by the procedure presented in this chapter. As can be seen from this figure, the leading edge of the first three reflections have been detected fairly well. The fourth reflected signal is not discernable due to noise. As with Example 1, the trailing edge of the tone burst also results in spike indications. All four trailing edge indications are evident in Figure 5.10c. The four negative going spikes occurring in this figure are caused by the previously mentioned phase reversal occurring in the basic wavelet.

Figure 5.11 presents the optimal length 150 filter in the same format as Figure 5.10. In this figure, it is interesting to note that all four spikes detecting the reflected wavelet leading edges are resolved and a significant reduction of noise in the region

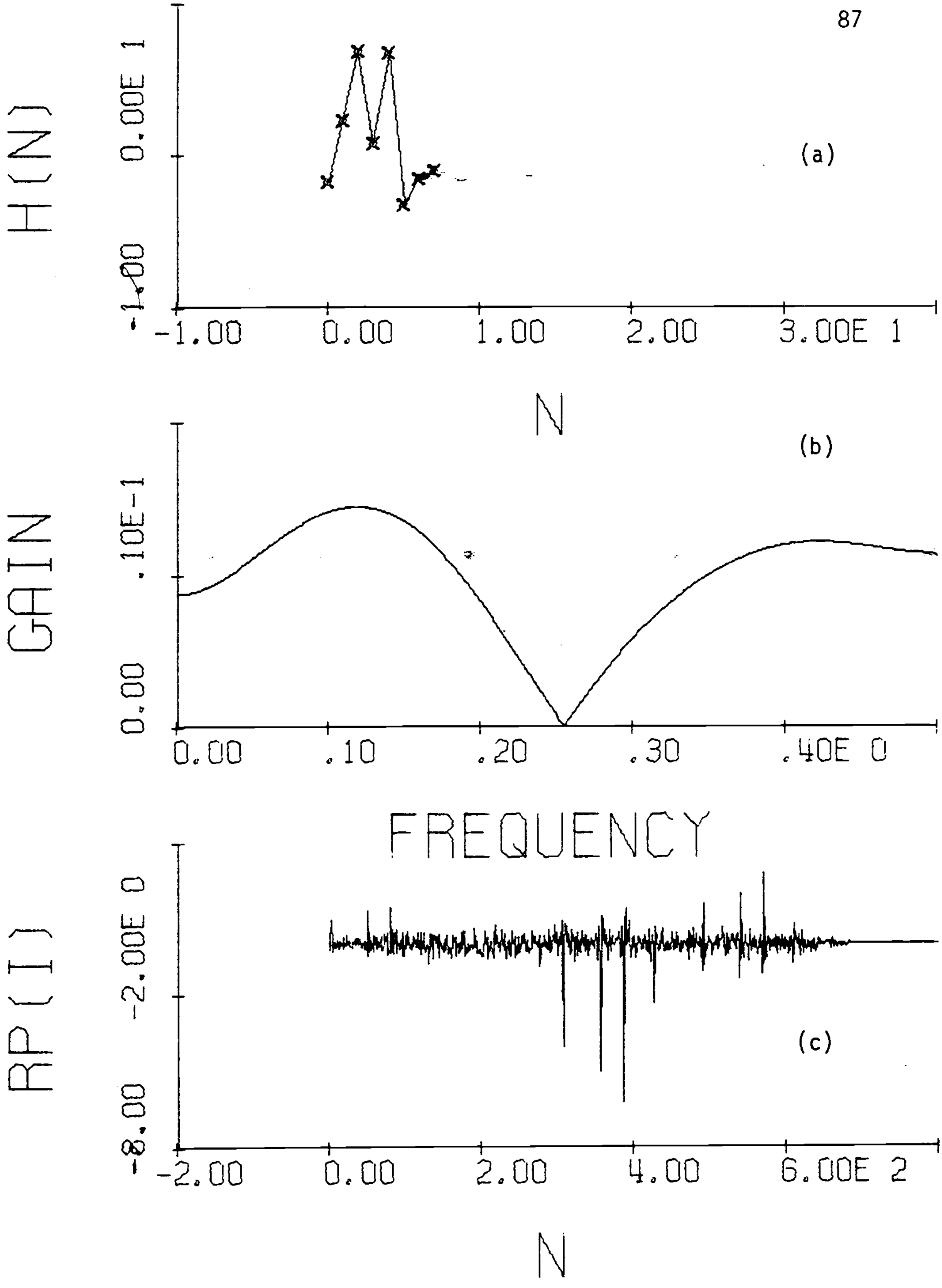


Figure 5.10

Example 2 Result for a Short Filter

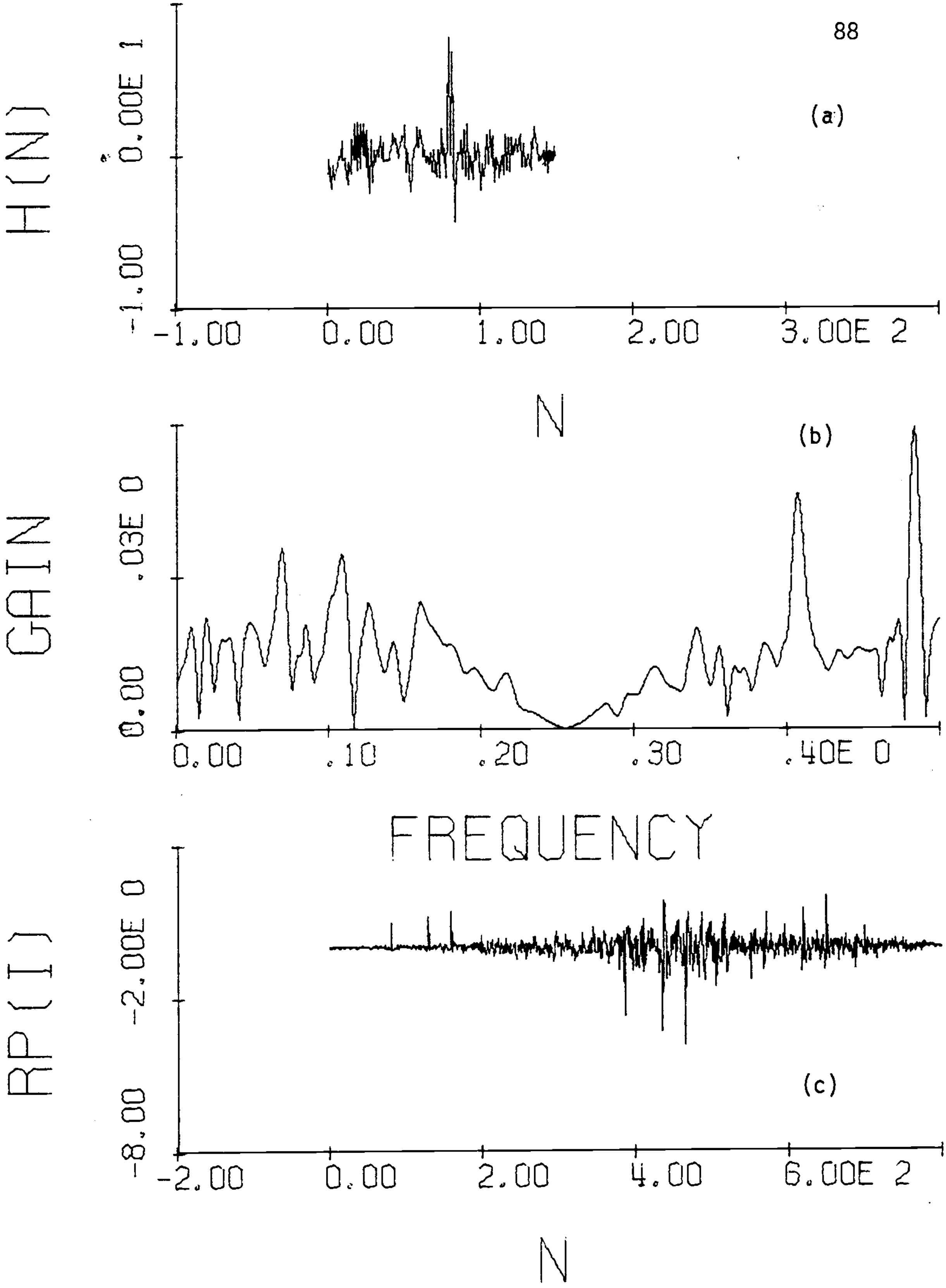


Figure 5.11

Example 2 Result for a Long Filter

corresponding to the leading edges is evident. Generally, a filter of length somewhat greater than the largest reflector delay will result in a very clear resolution of the leading edges. There is, of course, a price to pay for this improvement in filter complexity (design and application cost) and in increased noise at longer delays.

It is interesting to note the increased complexity of the frequency characteristic of Figure 5.11b over Figure 5.10b. As the filter length increases, the filter becomes more finely tuned to the job of detecting leading edges and the filter frequency characteristic approaches more closely the inverse of the characteristic of the basic wavelet. The price paid for this "improved" behavior is that such finely tuned filters will be less robust than shorter filters. This specialization causes undesirable behavior for data not precisely the same as the basic wavelet used to design the filter. This undesirable behavior is evident in Figure 5.11c as the increased noise occurring after the leading edges of the reflected wavelets.

VI CONCLUSIONS

6.1 Summary and Results

It has been the purpose of this thesis to study the application of least-squares optimal (Wiener) deconvolution filtering to multiple overlapping tone burst signals. The tone burst waveform is commonly used in ranging applications such as radar and sonar. Several overlapping return signals are frequently received when multiple reflections from the target or other objects are generated. The detection of these multiple reflections and estimation of their amplitudes and time of arrival allow a significant improvement in information derived from ranging systems.

A similar problem occurs in seismic studies when accoustic signals are transmitted into the earth and reflections are recorded to study the earth structure. Robinson, Treitel and others have been very successful at applying the original methods of Norbert Wiener in least squares filtering to the seismic problem using digital computers. These digital methods have been successfully applied to seismic processing for nearly 20 years. More recently, these methods have been successfully used in speech and image processing by a number of researchers. The methods of linear prediction especially appear to have a significant role to play in speech analysis and coding. Linear prediction was proposed by Robinson [25] in his Ph.D. thesis in 1954 as an extension of least-squares filtering methods.

It seems reasonable that since these methods have achieved such success in a wide range of deconvolution problems, they should also be appropriate for tone-burst deconvolution. The author is unaware of any previously published work involving efforts in this area.

As a first step toward evaluating the effectiveness of these methods for tone-burst deconvolution, least-squares filters were studied in Chapter II. Chapter II presented little in the way of insight into the use of these filters other than some general computational concepts and the basic formulation. One point discussed which indicates intuitively the direction of further study, is the phase problem with digital Wiener filters. It is shown that signals to be filtered must exhibit the property of minimum phase unless some delay is added to the filter. This property was studied in greater detail in Chapter IV.

In Chapter III, the characteristics of tone-burst signals were studied. Particular emphasis is placed on the phase of the signal since this characteristic plays a crucial role in a signal's suitability for Wiener deconvolution. It is shown that a signal's "phase" may be characterized by the location of zeros in the complex z-plane and that for tone-burst signals, these zeros are nearly all located on or very near the unit circle. As a result of this chapter, it is clear that tone-burst waveforms may be characterized as mixed phase. Since phase plays such a crucial role in applying Wiener filters, the method of exponential weighting is presented as a means of modifying the phase characteristic of a signal.

Chapter IV was the heart of the thesis. In this chapter the convergence properties of analytic inverse filters are used to demonstrate the desirability of phase modification for deconvolution of tone burst signals. It is shown why inverse Wiener filters (which are least-square optimal approximations of analytic inverse filters) have been successfully applied to minimum phase signals in seismology and speech in terms of the desirable convergence properties of minimum (or maximum) phase signals. Furthermore, Chapter IV develops the theoretical basis for understanding the characteristics of tone-burst signals which make them generally unsuitable for direct application of Wiener deconvolution. In Chapter IV, a computational method developed by Tribolet is shown to yield a reasonable estimate of the phase characteristic of a signal. This procedure allows a rough estimate of the exponential weighting necessary to modify a given tone-burst's phase characteristic to make the waveform more suitable for Wiener deconvolution.

Chapter V presented a computational procedure for implementing the concepts developed in Chapters II, III and IV. The application of this procedure to a representative artificial and real tone-burst signal is also presented in this chapter to give an idea of the implementation of these methods.

In summary, it was shown that tone-burst waveforms are not well suited to deconvolution via least-squares inverse (Wiener) filtering because of the waveforms phase characteristic. However, it is shown that this phase characteristic may be modified by exponential weighting. The resultant properly weighted waveform is more suitable for deconvolution and weighted Wiener filtering provides a reasonable approach for the detection and analysis of multiple overlapping tone-burst waveforms.

6.2 Future Directions

As with many detailed studies such as that undertaken for this thesis, more questions remain unanswered at the conclusion of the study than were believed to exist at its initiation. A clear understanding of these remaining problems is a significant result of this study. In the hope that these areas will be further examined in the future, a brief summary of some potential future directions will be presented here.

First of all, this study has not been concerned with the influence of ever-present noise on the analysis results. The effect of noise on an analysis procedure is of great importance to the successful application of the procedure to actual data. Many investigators have studied the noise characteristics of least-squares filtering techniques. The result of these studies indicate that this filtering method is optimal for certain types of noise (such as additive Gaussian). Furthermore, the broad application of this filtering method indicates the method works well for a range of different types of noise. However, the procedure proposed in this thesis requires that the non-linear operation of exponential weighting must be applied prior to least-squares filtering and a non-linear inverse weighting be applied after the filtering. The effect of noise on this procedure should be carefully studied. Non-linear operations frequently result in undesirable sensitivity to noise and one might suspect such a problem in this case.

Another direction for further study involves the development of "optimal" weighting functions for phase modification. These functions may be linear in which case the potential problem of noise sensitivity would be considerably reduced. It would seem that a large class of functions with more desirable characteristics than the exponential function must exist similar to the windows currently used to reduce leakage in discrete Fourier transform analysis. The development of a general theory of phase modification windows would allow a broad class of previously unsuitable signals to be analyzed by Wiener filters.

Another direction for future study is the improvement of the least-squares deconvolution filter itself. This area is considerably more difficult than the two previously mentioned, but also is significantly more useful in the long term. Various investigators have attempted to further develop optimal filters - many with significant advances. However, most of these "more optimal" filters pay the price of more restricted application. That is, they perform better than the least-squares optimal filter only for a narrow class of signals. A significant step in what may be the right direction has been made by Childers, Varga and Perry [26]. In their procedure, a waveform other than a spike is selected as the desired output of the filter. Such a waveform may be better matched to the phase characteristic of the signal to be analyzed. This procedure, if successfully developed, would alleviate the need for a phase modification preprocess entirely. A significant area of future study in deconvolution filters then might center on the optimal selection of the filter output waveform.

REFERENCES

[1] Wiener, N., Extrapolation, Interpolation and Smoothing of Stationary Time Series, The Technology Press, Cambridge, Mass. and John Wiley & Sons, Inc., New York, 1949.

[2] Robinson, Enders A. and Treitel, Sven, "Principles of Digital Wiener Filtering," Geophys. Prosp., V. 15, pp. 311-333, 1967.

[3] Treitel, S. and Robinson, E.A., "The Design of High-Resolution Digital Filters," IEEE Trans. Geos. Elect., Vol. GE-4, No. 1, pp. 25-38, June 1966.

[4] Wood, Lawrence C. and Treitel, Sven, "Seismic Signal Processing," Proc. IEEE, Vol. 63, No. 4, pp. 649-661, April 1975.

[5] Robinson, E.A. and Treitel, S., "Digital Signal Processing in Geophysics," in Applications of Digital Signal Processing, Alan V. Oppenheim, ed., Englewood Cliffs, New Jersey; Prentice-Hall, Inc., 1978, Chapter 7.

[6] Robinson, Enders A., Multichannel Time Series Analysis with Digital Computer Programs, San Francisco; Holden-Day, 1967, Chapter 2.

[7] Hunt, B.R., "Digital Image Processing," in Applications of Digital Signal Processing, Alan V. Oppenheim, ed., Englewood Cliffs, New Jersey; Prentice-Hall, Inc., 1978, Chapter 4.

[8] Tribolet, Jose Manuel, "Seismic Applications of Homomorphic Signal Processing," Ph.D. thesis submitted to the Department of Electrical Engineering and Computer Science at MIT on May 19, 1977.

[9] Levinson, N., "The Wiener RMS (Root Mean Square) Error Criterion in Filter Design and Prediction," J. Math. Phys., Vol. 25, pp. 261-278, January 1947.

[10] Claerbout, J.F. and Robinson, E.A., "The Error in Least Squares Inverse Filtering," Geophysics, Vol. 29, 1964, pp. 118-120.

[11] Wiggins, R.A. and Robinson, E.A., "Recursive Solution to the Multichannel Filtering Problem," J. Geophy. Research, Vol. 70, No. 8, pp. 1885-1891, April 1965.

[12] Makhoul, John, "Linear Prediction: A Tutorial Review," Proc. IEEE, Vol. 63, No. 4, April 1975, pp. 561-580.

[13] Peacock, K.L. and Treitel, Sven, "Predictive Deconvolution: Theory and Practice," Geophysics, Vol. 34, No. 2, April 1969, pp. 155-169.

[14] Rabiner, Lawrence R. and Gold,Bernard, Theory and Application of Digital Signal Processing, Prentice-Hall, Inc., 1975.

[15] Oppenheim, Alan V. and Schafer, Ronald W., Digital Signal Processing, Prentice-Hall, Inc., 1975.

[16] Cadzow, James A., Discrete-Time Systems, Prentice-Hall, Inc., Englewood Cliffs, New Jersey, 1973.

[17] Robinson, Enders A., Statistical Communication and Detection, Charles Griffin and Company, LTD., London, 1967.

[18] Robinson, Enders A. and Treitel, Sven, "Dispersive Digital Filters," Reviews of Geophysics, Vol. 3, No. 4, November 1965, pp. 433-461.

[19] Treitel, S. and Robinson, E.A., "The Stability of Digital Filters," IEEE Trans. Geo. Electronics, Vol. GE-2, pp. 6-18, November 1964.

[20] Churchill, Ruel V., Complex Variables and Applications, McGraw-Hill, Inc., 1960.

[21] Schafer, R.W., "Echo Removal by Discrete Generalized Linear Filtering," Tech. Rept. 466, MIT Research Laboratory of Electronics, MIT, Cambridge, Mass., February 1969. Also Ph.D. Thesis, Dept. of Elec. Engineering, MIT, February 1968.

[22] Tribolet, Jose M., "A New Phase Unwrapping Algorithm," IEEE Trans. Acoustics, Speech and Signal Processing, Vol. ASSP-25, No. 2, April 1977, pp. 170-177.

[23] Bonzanigo, F., "An Improvement to Tribolet's Phase Unwrapping Algorithm," IEEE Trans. Acoustics, Speech and Signal Processing, Vol. ASSP-26, No. 1, February 1978, pp. 104-105.

[24] Crochiere, R.E. and Rabiner, L.R., "Optimum FIR Digital Filter Implementation for Decimation, Interpolation, and Narrow-band Filtering," IEEE Trans. Acoustics, Speech and Signal Processing, Vol. ASSP-23, pp. 444-456, October 1975.

[25] Robinson, E.A., "Predictive Decomposition of Time Series with Applications to Seismic Exploration," Ph.D. Thesis, Department of Geology and Geophysics, Massachusetts Institute of Technology, Cambridge, Mass., 1954.

[26] Childers, D.G., Varga, R.S., and Perry, N.W., "Composite Signal Decomposition," IEEE Trans. Audio and Electroacoustics, Vol. AU-18, No. 4, December 1970, pp. 471-477.