

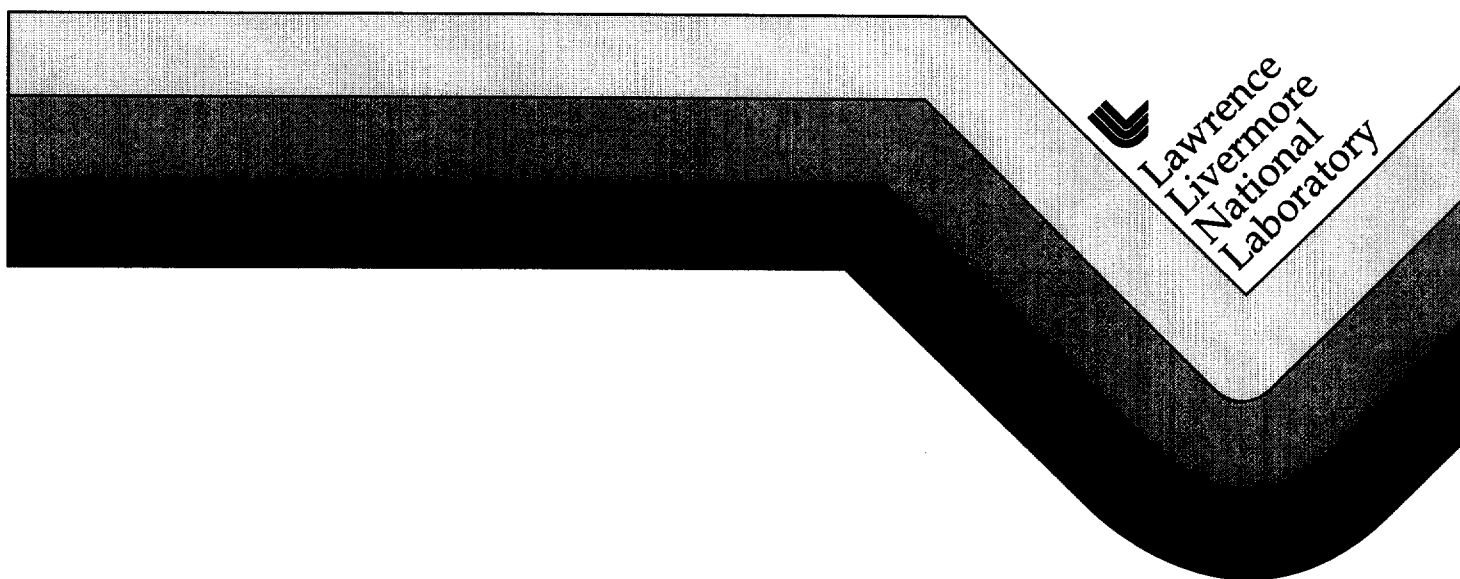
23-11-97

UCRL-LR-127700

# **Wavelet Transform Analysis of Transient Signals: The Seismogram and the Electrocardiogram**

**Kanwaldip Singh Anant  
PhD Thesis**

**June 1997**



# DISCLAIMER

This document was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor the University of California nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or the University of California. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or the University of California, and shall not be used for advertising or product endorsement purposes.

This report has been reproduced  
directly from the best available copy.

Available to DOE and DOE contractors from the  
Office of Scientific and Technical Information  
P.O. Box 62, Oak Ridge, TN 37831  
Prices available from (615) 576-8401, FTS 626-8401

Available to the public from the  
National Technical Information Service  
U.S. Department of Commerce  
5285 Port Royal Rd.,  
Springfield, VA 22161

# **Wavelet Transform Analysis of Transient Signals: The Seismogram and the Electrocardiogram**

**Kanwaldip Singh Anant**

**Doctor of Philosophy  
Thesis**

**Manuscript date: June 1997**





***Wavelet Transform Analysis of Transient Signals: the  
Seismogram and the Electrocardiogram***

by

**Kanwaldip Singh Anant**

**B.S. (University of Oklahoma) 1987**

**M.S. (University of California, Davis) 1993**

**DISSERTATION**

**Submitted in partial satisfaction of the requirements for the degree of**

**DOCTOR OF PHILOSOPHY**

in

**Engineering - Applied Science**

in the

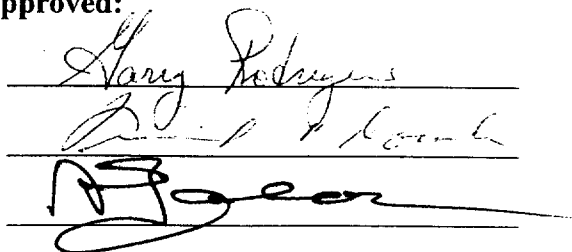
**OFFICE OF GRADUATE STUDIES**

of the

**UNIVERSITY OF CALIFORNIA**

**DAVIS**

**Approved:**



**Committee in Charge**

**1997**

# *Wavelet Transform Analysis of Transient Signals: the Seismogram and the Electrocardiogram*

**Kanwaldip Singh Anant**  
**Department of Applied Science**  
**University of California, Davis**

**June 1997**

**Dissertation Committee:**  
**Garry H. Rodrigue, Chair (University of California, Davis)**  
**Farid U. Dowlah (University of California, Davis and Lawrence Livermore National Laboratory)**  
**Andrew J. Gabor (University of California, Davis)**

## **Abstract**

In this dissertation I quantitatively demonstrate how the wavelet transform can be an effective mathematical tool for the analysis of transient signals. The two key signal processing applications of the wavelet transform, namely feature identification and representation (i.e., compression), are shown by solving important problems involving the seismogram and the electrocardiogram.

The seismic feature identification problem involved locating in time the P and S phase arrivals. Locating these arrivals accurately (particularly the S phase) has been a constant issue in seismic signal processing. In Chapter 3, I show that the wavelet transform can be used to locate both the P as well as the S phase using only information from *single station* three- component seismograms. This is accomplished by using the basis function (wavelet) of the wavelet transform as a matching filter and by processing information *across*

scales of the wavelet domain decomposition. The “pick” time results are quite promising as compared to analyst picks.

The representation application involved the compression of the electrocardiogram which is a recording of the electrical activity of the heart. Compression of the electrocardiogram is an important problem in biomedical signal processing due to transmission and storage limitations. In Chapter 4, I develop an electrocardiogram compression method that applies vector quantization to the wavelet transform coefficients. The best compression results were obtained by using *orthogonal* wavelets, due to their ability to represent a signal efficiently.

Throughout this thesis the importance of choosing wavelets based on the problem at hand is stressed. In Chapter 5, I introduce a wavelet design method that uses *linear prediction* in order to design wavelets that are geared to the signal or feature being analyzed. The use of these designed wavelets in a test feature identification application led to positive results.

The methods developed in this thesis; the feature identification methods of Chapter 3, the compression methods of Chapter 4, as well as the wavelet design methods of Chapter 5, are general enough to be easily applied to other transient signals.

## Acknowledgments

I would like to first thank the members of my dissertation committee: Farid Dowla, Garry Rodrigue, and Andrew Gabor. Farid Dowla provided the motivation for what is now a deep interest in signal processing. He has been patient as I have learned the fundamentals of the field, and has helped me to develop and refine my research skills. Farid's support and guidance have been critical to the success of this research. Garry Rodrigue was my initial guide in choosing a research area and has been an important teacher of applied math concepts. He also provided a great deal of support as I prepared for my written and oral qualifying examinations. Though I did not get the opportunity to interact a great deal with Andrew Gabor, he has helped me with my understanding of biological signal processing, particularly with regards to my work on the electrocardiogram. I would also like to thank him for participating on my oral qualifying examination committee. My thanks also goes to Gary Ford, Nelson Max, and Ann Orel for sitting on the oral committee.

Others who have influenced my research work include (but are not limited to) the following. Ronald Fearing and Martin Vetterli introduced me to the basics of signal processing theory through their excellent courses at the University of California, Berkeley. Martin Vetterli's papers were also a strong influence as I developed an understanding of wavelet theory. Dave Harris, Richard Leach, Flori Ryall, and Craig Schultz have provided important information and feedback regarding my work on the seismogram. Many fellow students have served as informal advisors during my graduate studies, the most notable being Paul Amala, Colin Aro, and Abner Bello.



### *Dedication*

I would like to dedicate this dissertation to my parents: Mrs. Darshan Kaur and the late Dr. Santokh Singh, and my two brothers: Janmeet Singh and Birjinder Singh. They have been my role models in life and I have tried to follow their leads as they exemplify the Sikh way of life through honest work, sharing with others, and a strong belief in universal brotherhood. Without the inspiration that they provided this work would not have been possible. This dissertation belongs more to them than to me.

# Table of Contents

|                      |    |
|----------------------|----|
| Abstract .....       | ii |
| Acknowledgments..... | iv |

## CHAPTER 1

### Introduction

|  |   |
|--|---|
| 1.1 Background .....                   | 1 |
| 1.2 Overview of this Research .....    | 2 |
| 1.3 Organization of Dissertation ..... | 4 |

## CHAPTER 2

### The Wavelet Transform

|  |    |
|--|----|
| 2.0 Motivation and Introduction .....  | 5  |
| 2.1 Definition of the Wavelet Transform .....  | 8  |
| 2.2 Properties of the Wavelet Transform .....  | 11 |
| 2.3 Discrete-time Wavelet Transform (DTWT) and Perfect Reconstruction Filter Banks (PRFB)..... | 13 |
| 2.3.1. <i>Basics of PRFB</i> .....   | 13 |
| 2.3.2 <i>The Discrete-time Wavelet Transform (DTWT)</i> .....                                  | 16 |
| 2.3.3 <i>Linearity and Shift Properties of DTWT</i> .....                                      | 19 |
| 2.4 Designing Wavelet Filters .....  | 20 |
| 2.5 Applications .....   | 22 |
| 2.5.1. <i>Feature Detection</i> .....  | 22 |
| 2.5.2. <i>Compression</i> .....  | 23 |
| 2.6 Discussion and Conclusions .....   | 23 |
| Appendix 2.A The Resolution of Identity Theorem and the Inverse Wavelet Transform.....         | 24 |
| Appendix 2.B Linear Convolution, Downsampling, Upsampling, etc.....                            | 27 |

## CHAPTER 3

### Wavelet Transform Methods for Phase Identification in Three-component Seismograms

|   |    |
|---|----|
| 3.0 Summary .....   | 34 |
| 3.1 Introduction.....   | 35 |
| 3.2 Theoretical Background.....   | 37 |
| 3.2.1 <i>The Wavelet Transform and Feature Identification</i> .....         | 37 |
| 3.2.2 <i>Characterization of Seismograms by the Wavelet Transform</i> ..... | 40 |
| 3.2.3 <i>The Algorithm</i> .....  | 43 |
| 3.2.4 <i>Possible Refinements of Algorithm</i> .....                        | 47 |
| 3.3. Results.....   | 49 |
| 3.3.1 <i>P Locator Results</i> .....  | 49 |
| 3.3.2 <i>S Arrival Locator Results</i> .....                                | 51 |
| 3.4 Discussion and Conclusions .....  | 54 |
| Appendix 3.A Example of Conservation of Polarization Through Scales .....   | 55 |
| Appendix 3.B Determination of Window for Polarization Calculation.....      | 59 |
| Appendix 3.C Use of the Wavelet Transform as a Glitch Detector .....        | 60 |

## CHAPTER 4

### Compression of the ECG using the Wavelet Transform and Vector Quantization

|   |    |
|---|----|
| 4.0 Summary .....   | 75 |
| 4.1. Compression Methods using Linear Transforms .....                    | 76 |
| 4.2. The Wavelet Transform and Compression.....                           | 79 |
| 4.3. The ECG as a Representation of Heart Function.....                   | 82 |
| 4.4. Background on ECG Compression Techniques.....                        | 84 |
| 4.5. The Wavelet Transform, Vector Quantization, and ECG Compression..... | 86 |
| 4.6. Method .....   | 88 |
| 4.6.1. <i>Some Notation and Nomenclature</i> .....                        | 88 |
| 4.6.2. <i>Development of Method</i> .....                                 | 89 |
| 4.6.3. <i>Application of Method to Test Set</i> .....                     | 91 |
| 4.7. Results.....   | 93 |
| 4.8. Discussion and Conclusions .....                                     | 96 |
| Appendix 4.A K-means clustering.....                                      | 98 |

## **CHAPTER 5**

### **Wavelet Design by Linear Prediction Lifting**

|   |     |
|---|-----|
| 5.0 Summary .....   | 111 |
| 5.1 Introduction .....  | 112 |
| 5.2 Wavelet Design Methods: Fourier Transform Based .....                               | 113 |
| 5.3 Wavelet Design Methods: Lifting Scheme .....  | 114 |
| 5.4 Linear Prediction and Lifting for Wavelet Design .....                              | 118 |
| 5.4.1. <i>Linear Prediction for Predict Step of Lifting Scheme</i> .....                | 118 |
| 5.4.2 <i>Using Linear Prediction and Lifting to Construct Filter Bank Filters</i> ..... | 119 |
| 5.5 Test Application: Problem Cases from Chapter 3 .....                                | 122 |
| 5.6 Discussion and Conclusions .....  | 123 |
| Appendix 5.A Fourier Based Wavelet Design Method .....                                  | 124 |
| Appendix 5.B Calculation of the Linear Prediction Coefficients .....                    | 129 |

## **CHAPTER 6**

### **General Discussion**

|                            |     |
|----------------------------|-----|
| 6.1 Conclusions .....      | 137 |
| 6.2 Future Direction ..... | 139 |

|                           |     |
|---------------------------|-----|
| <b>Bibliography</b> ..... | 141 |
|---------------------------|-----|

**CHAPTER 1*****Introduction*****1.1 Background**

During the last several years the wavelet transform (WT) has emerged as an important signal processing research tool. A group of French geophysicists were the first to formally introduce the idea of the wavelet transform as a result of their work on seismic signals (Grossmann and Morlet, 1984; Goupillaud et al., 1984). Wavelet transform theory as it relates to filter banks was refined by several researchers in the late 1980s and early 1990s, including Mallat (1989), Daubechies (1992), Vetterli and Herley (1992), and Vaidyanathan (1993). Though work on wavelet theory has been extensive, research into wavelet transform applications is still in its infancy. Of particular interest is the use of the wavelet transform for the analysis of transient signals, since it is seen as an improvement over the traditionally used short-time Fourier transform (STFT). This is based on the fact that by using different *scales* of the analyzing *wavelet*, the wavelet transform is able to locate in

time short duration, high frequency behavior, while resolving in frequency longer duration lower frequency behavior.

## 1.2 Overview of this Research

The two main areas of signal processing where the wavelet transform is seen as having the greatest potential impact are feature identification and data compression. This dissertation studies the wavelet analysis of transient signals in two parts. The first part (Chapter 3) focuses on the use of the wavelet transform for feature identification in three component single station seismograms. The features of interest are the P and S phase arrivals. Methods are developed to accurately locate these arrivals using information that is present *across* the scales of the wavelet decomposition. The second part (Chapter 4) reports on the development of an electrocardiogram (ECG) compression method using vector quantization (VQ) of the wavelet coefficients. The common issue among both applications is the importance of wavelet choice. For compression, the wavelet must be chosen to produce the most compact representation of the original signal. For feature detection, the wavelet should best “match” the feature in question. The wavelet can be chosen from a “library” of standard wavelets constructed mathematically and not based on signal or feature matching, or they can be designed independently using signal features as design guidelines. In Chapter 5, I introduce a design method that uses *linear prediction* to design wavelets that are geared to the particular signal being studied.

Thus, overall, the purpose of this research can be broken down as such:

- Investigate the use of the wavelet transform as a feature identification tool. Develop methods to identify features in single station seismograms by taking advantage of the freedom of basis function choice as well as the multiscalar decomposition of the wavelet transform.
- Develop a compression technique using the wavelet transform and vector quantization. Compare compression results of different wavelets.
- Develop a wavelet design method that uses information about signal features in order to design wavelets that are “best” suited to a particular application.

The main contributions of this research are:

- Development of a method to locate seismic arrivals in single station seismograms using information that is present *across* wavelet scales. Comparison of performance of various wavelets in this feature identification problem. Finding that earlier P phase arrival can be used to design wavelet to locate S phase arrival. Establishing the wavelet transform as a valuable seismic analysis tool.
- Development of ECG compression method by using vector quantization on the wavelet transform coefficients. Finding that higher order orthogonal wavelets lead to best compression results due to the resulting nonredundant decomposition. Achievement of storage rates as low as 1.1 bits-per-sample while retention of ECG feature integrity.
- Development of a wavelet design method that uses linear prediction to design wavelets by using strictly time domain feature properties. Application of this method to design wavelets for seismic feature identification.

These contributions are of use not only for the analysis of seismic signals and ECGs, but also in the analysis of other types of transient signals, including images. Most of the algorithms developed in this research are not dependent on the type of signal being analyzed, and those that are need only minor modifications to be applied to other situations.

### **1.3 Organization of Dissertation**

This dissertation is arranged as follows. Chapter 2 presents a brief tutorial on the wavelet transform, highlighting its advantages over traditional signal processing techniques, and presenting its main properties. Also presented is much of the notation that is used throughout this dissertation. A discussion of wavelet transform applications is found at the end of Chapter 2, providing a springboard for the guts of the dissertation: Chapters 3, 4, and 5, the contents of which have already been described in the previous section. Each of the chapters contains a thorough theoretical justification for the methods developed. At the end of each chapter are appendices providing more detailed derivations and code for those interested. Chapter 6 contains conclusion about this research as well as ideas about the possible directions of future work.



**2.0 Motivation and Introduction**

In this chapter the mathematical tool that is the focus of this dissertation is introduced. Most discussions on the *wavelet transform* (WT) begin with a review of the *Fourier transform* (FT). Although this is not necessary, as the WT is not dependent in any manner on the FT, starting with the FT helps to draw attention to the ways in which the WT is different than, and in certain cases (dependent on the application), superior to the FT. Also, when introducing a new subject it is helpful to start with something that is familiar. Thus, the first step is to present the definition of the FT. The FT of a function  $x(t)$  is defined as:

$$X(\omega) = \int_{-\infty}^{\infty} x(t) \exp(-j\omega t) dt \quad (1)$$

where  $x(t)$  is in  $L^2(\mathfrak{R})$ .

The inverse Fourier Transform (IFT) is:

$$x(t) = \frac{1}{2\pi} \int_{-\infty}^{\infty} X(\omega) \exp(j\omega t) d\omega \quad (2)$$

Looking at Eq. (2) it can be seen that the function  $x(t)$  is represented as a sum of infinite length sinusoids. The FT is a powerful tool that has been used extensively in various signal processing applications. Nonetheless, it does have some serious limitations. The FT provides a good representation for stationary signals especially signals that are made up of a combination of pure sinusoids. However, problems arise if the signal to be represented contains sudden sharp transitions (spikes or bursts). These “transients” need an infinite number of sinusoidal components to be accurately represented using the FT. Thus, this type of behavior is spread out over the entire frequency axis, and therefore is difficult to isolate in the frequency domain.

A classic method used to overcome this type of problem is to window the signal before taking the FT. This type of procedure is called the *short-time Fourier transform* (STFT) (Gabor, 1946; Allen and Rabiner, 1977), which is just a windowed version of the FT:

$$STFT(\tau, \omega) = \int_{-\infty}^{\infty} x(t) g(t-\tau) e^{-j\omega t} dt \quad (3)$$

The discrete-time version is:

$$STFT(n, \omega) = \sum_{m=-\infty}^{\infty} x[m] w[n-m] e^{-j\omega m} \quad (4)$$

where  $g$  ( $w$  in the discrete version) is a window function which can be rectangular, Gaussian, Hanning, etc. (Oppenheim and Schaffer, 1989). The STFT divides a signal into segments and takes the FT of each resulting section. The size of the window to use depends on what type of features need to be identified. To locate short-time high frequency behavior, narrow time windows are needed, while to resolve low frequency behavior wider time windows are necessary. This is a result of the fact that time localization and frequency res-

olution cannot simultaneously be determined to arbitrary precision. This can be stated in terms of a redefinition of the Heisenberg uncertainty principle (which is better known as one of the basic principles of quantum physics):

$$\Delta t \Delta f \geq 1 \quad (5)$$

where  $\Delta t$  and  $\Delta f$  are the uncertainties in time and frequency (Gabor, 1946).

To illustrate this with an example, let us analyze a synthetic signal containing a Dirac delta function superimposed on a sinusoid (Figure 1). There are 1024 sample points in the signal, and the frequency of the sinusoid is  $\omega = 0.03\pi$  radians/second. Therefore, the signal is characterized by a relatively long duration low frequency sinusoid and an extremely short duration high frequency transient represented by the delta function. Based on the discussion of the previous paragraph, it should be expected that in order to detect the transient a narrow window STFT will be necessary. The STFT of the signal was calculated using a wide rectangular window of 256 points and a narrow rectangular window of 8 points (an overlap of half the window length was used). The resulting time-frequency plots are presented in Figures 2 and 3, respectively. These plots show the frequency response versus window position. For example for the wide window case, six overlapping windows were used and in Figure 2 are shown the frequency responses for each of these windows, from the first to the sixth.

In Figure 2 it can be seen that along the frequency axis there is only one sharp peak at the sinusoidal frequency for each of the windows. From the vantage point of the time axis there is no indication of the presence of the transient. Thus, the presence of the delta func-

tion cannot be determined from the “wide window” STFT, but the frequency of the sinusoid is well resolved. On the other hand in Figure 3 the peaks for each window at the sinusoidal frequency have magnitudes that are significantly less than the peaks in Figure 2. Also, there are “ripples” at the other frequencies. However, along the time axis the presence and *location* of the transient is clearly represented. Thus, the “narrow window” STFT has resulted in detection and localization of the delta function at the cost of losing frequency resolution of the sinusoid. So, to analyze various types of behavior in a signal the STFT would have to be taken several times, each time with a different size window function. This is especially problematic when the location and/or duration of transient behavior is unknown, which is normally the case. What is needed is a mapping that uses one initial window function and varies its size automatically so that all frequency behavior can be analyzed with just one pass (narrow windows at high frequencies and wide windows at low frequencies). This is where the *wavelet transform* (WT) comes in.

## 2.1 Definition of the Wavelet Transform

Though some of the ideas behind the wavelet transform have been around since the early part of the 20th century, the formalization of wavelet theory was only recently initiated by geophysicists involved in seismic signal analysis (Goupillaud, et al., 1984; Grossmann and Morlet, 1984, Morlet, et al., 1982). The theory was refined by researchers in the mathematical community (Daubechies, 1992) and in the signal processing community (Mallat, 1989; Vetterli and Herley, 1992; Vetterli and Kovacevic, 1995). In this and pro-

ceeding sections the wavelet transform is defined (both continuous-time and discrete-time) and the advantages of this type of transform are highlighted.

The Continuous-time Wavelet Transform (CWT) is defined as:

$$W_f(a, b) = \frac{1}{\sqrt{|a|}} \int_{-\infty}^{\infty} f(t) g^*\left(\frac{t-b}{a}\right) dt \quad (6)$$

$$a, b \in \mathbb{R}; a \neq 0 \quad f, g \in L^2(\mathbb{R})$$

where  $a$  is called the scaling factor,  $b$  is the translation parameter (has the same meaning as  $\tau$  which is in the STFT), and  $g$  is the window function or “wavelet”. The choice of  $g$  is restricted by the following relation:

$$C_g = \int_{-\infty}^{\infty} \frac{|G(\omega)|^2}{|\omega|} d\omega < \infty \quad (7)$$

which is known as the *admissibility condition*. If it is assumed that  $g(t)$  is in  $L^1(\mathbb{R})$  (which is true for all cases of practical interest), then  $G(\omega)$  is continuous, and therefore the admissibility condition is met only if  $G(0) = 0$ . It easily follows from (1) that if  $G(0) = 0$ , then  $\int_{-\infty}^{\infty} g(t) dt = 0$  and so the necessary and *sufficient* conditions on the continuous wavelet are that it 1) be finite energy, and 2) have zero mean. Thus, unlike the STFT where one is limited to a single basis function (sinusoid), with the wavelet transform one can choose or design a wavelet to fit the application at hand, as long as the admissibility condition is met. The possibilities are infinite!

Also unlike the STFT where the window size is independent of  $\omega$ , with the WT the scaling factor,  $a$ , controls the size of the analysis wavelet. Looking at (6) it can be seen that as  $a$  decreases the function  $g$  becomes more compact (the window narrows), while as

$a$  increases the window widens (refer to Figure 5 for an illustration of how the wavelet broadens with increasing  $a$ ). So, keeping in mind that  $a$  is inversely proportional to frequency, for small  $a$  (high frequency) there is good time localization, while for large  $a$  (low frequency) there is better frequency resolution. This is often termed a constant-Q

$$Q = \frac{\Delta f}{f_o} = \frac{\text{frequency bandwidth}}{\text{center frequency}} \quad (8)$$

type of analysis and is one of the main advantages of the WT. This advantage is illustrated in the wavelet domain representation of the example signal of Figure 1 (sinusoid plus Dirac). Figure 4 shows five scales of that signal with the lowest scales at the top of the plot. Notice how the location of the Dirac is clearly represented in the low scales, while in the higher scales the sinusoid is well resolved. Another important (often unstated) advantage of the WT is the fact that the wavelet domain is a time based domain, while the Fourier domain of the STFT is somewhat abstract (compare the representations in Figures 2 and 3 with the representation in Figure 4). This can be a very important issue, particularly when one is dealing with feature identification problems.

The reason for having the admissibility condition is so that an inverse wavelet transform (IWT) can be defined. The IWT is defined as:

$$f(t) = \frac{1}{C} \int_{-\infty}^{\infty} \frac{1}{a^2} \int_{-\infty}^{\infty} W_f(a, b) \frac{1}{\sqrt{a}} g\left(\frac{t-b}{a}\right) db da \quad (9)$$

This relation can be derived from the resolution of identity theorem:

$$\int_{-\infty}^{\infty} \frac{1}{a^2} \int_{-\infty}^{\infty} \langle f_1, g_{ab} \rangle \langle g_{ab}, f_2 \rangle db da = C_g \langle f_1, f_2 \rangle \quad (10)$$

where  $g_{ab}(t) = \frac{1}{\sqrt{|a|}}g\left(\frac{t-b}{a}\right)$  (refer to Appendix 2.A for a complete derivation). From (9) it can be seen that the original function  $f(t)$  is expressed as a combination of weighted, scaled, and translated wavelets, where the wavelet coefficients  $W_f(a, b)$  are the weight factors. This is a much more natural way to represent real signals than the representation given in equation (2), since many real signals (transient signals in particular) are more effectively represented as a combination of basis functions that are concentrated in time than by a combination of infinite duration sinusoids.

## 2.2 Properties of the Wavelet Transform

In this section some important properties of the CWT are presented. Notice that the CWT is linear and shift invariant.

1) *Linearity*:

$$Af_1(t) + Bf_2(t) \rightarrow AW_{f_1}(t) + BW_{f_2}(t) \quad (11)$$

Proof:

$$\begin{aligned} & \frac{1}{\sqrt{|a|}} \int_{-\infty}^{\infty} (Af_1(t) + Bf_2(t)) g\left(\frac{t-b}{a}\right) dt \\ &= \frac{A}{\sqrt{|a|}} \int_{-\infty}^{\infty} f_1(t) g\left(\frac{t-b}{a}\right) dt + \frac{B}{\sqrt{|a|}} \int_{-\infty}^{\infty} f_2(t) g\left(\frac{t-b}{a}\right) dt \end{aligned}$$

2) *Translation invariance*: The CWT is shift invariant in terms of the translation parameter,  $b$ . Thus, if  $f_1(t) = f(t-t_0)$ , then:

$$W_{f_1}(a, b) = W_f(a, b - t_0) \quad (12)$$

Proof: Using  $f_1(t)$  as defined above we have:

$$W_{f_1}(a, b) = \frac{1}{\sqrt{|a|}} \int_{-\infty}^{\infty} f(t - t_0) g\left(\frac{t - b}{a}\right) dt \quad (13)$$

Using the substitution,  $t' = t - t_0$ , one gets:

$$\begin{aligned} W_{f_1}(a, b) &= \frac{1}{\sqrt{|a|}} \int_{-\infty}^{\infty} f(t') g\left(\frac{t' + t_0 - b}{a}\right) dt' \\ &= \frac{1}{\sqrt{|a|}} \int_{-\infty}^{\infty} f(t') g\left(\frac{t' - (b - t_0)}{a}\right) dt' = W_f(a, b - t_0) \end{aligned} \quad (14)$$

3) *Scalar variance*: If  $f_2(t) = f(2t)$ , then:

$$W_{f_2}(a, b) = \frac{1}{\sqrt{2}} W_f(2a, 2b) \quad (15)$$

Proof:

$$W_{f_2}(a, b) = \frac{1}{\sqrt{|a|}} \int_{-\infty}^{\infty} f(2t) g\left(\frac{t - b}{a}\right) dt \quad (16)$$

By making the substitution,  $\tau = 2t$ , one gets:

$$W_{f_2}(a, b) = \frac{1}{\sqrt{2}(|2a|)} \int_{-\infty}^{\infty} f(\tau) g\left(\frac{\tau - 2b}{2a}\right) d\tau = \frac{1}{\sqrt{2}} W_f(2a, 2b) \quad (17)$$



## 2.3 Discrete-time Wavelet Transform (DTWT) and Perfect Reconstruction

### Filter Banks (PRFB)

Though the CWT is useful for the mathematical derivation of wavelet transform theorems and properties, in computational applications (where signals and filters are discrete), it is the DTWT that is used. The DTWT is generally implemented using filter banks (Vetterli and Herley, 1992; Akansu, 1992; Vaidyanathan, 1993).

#### 2.3.1. Basics of PRFB

In Figure 6 is illustrated one stage of a two-band filter bank, with the squares representing linear convolution, circles with downward arrows representing downsampling by a factor of two (removal of every other sample), and circles containing upward arrows representing upsampling by a factor of two (inserting of zeros between samples). A mathematical description of these operations is given in Appendix 2.B. In the figure,  $x[n]$  is the input while  $g$  and  $h$  are called the analysis filters, and  $\tilde{g}$  and  $\tilde{h}$  are known as the reconstruction filters. The sequence  $\hat{x}[n]$  is the reconstructed version of the input. Let us analyze the filter bank to determine the conditions needed to attain perfect reconstruction (i.e.  $\hat{x}[n] = x[n]$ ). To make things easier, the z-transform representation shall be used in the analysis.

The z-transform is defined as:

$$X(z) = \sum_{n=-\infty}^{\infty} x[n] z^{-n} \quad (18)$$

Now, using some of the properties of the z-transform (Appendix 2.B) and referring to Figure 6, we have the following:

$$X_1(z) = X(z) G(z) \quad (19)$$

$$X_2(z) = \frac{1}{2} (X(z^{1/2}) G(z^{1/2}) + X(-z^{1/2}) G(-z^{1/2})) \quad (20)$$

$$X_3(z) = \frac{1}{2} (X(z) G(z) + X(-z) G(-z)) \quad (21)$$

$$X_a(z) = X_3(z) \tilde{G}(z) = \frac{1}{2} (X(z) G(z) \tilde{G}(z) + X(-z) G(-z) \tilde{G}(z)) \quad (22)$$

In a similar fashion, the following can be obtained:

$$X_b(z) = \frac{1}{2} (X(z) H(z) \tilde{H}(z) + X(-z) H(-z) \tilde{H}(-z)) \quad (23)$$

Now, finally the output is found to be:

$$\hat{X}(z) = X_a(z) + X_b(z) \quad (24)$$

$$\hat{X}(z) = \left( \frac{1}{2} X(z) (G(z) \tilde{G}(z) + H(z) \tilde{H}(z)) + \frac{1}{2} X(-z) (G(-z) \tilde{G}(z) + H(-z) \tilde{H}(z)) \right) \quad (25)$$

The  $X(-z)$  term is due to aliasing and can be cancelled by setting

$G(-z) \tilde{G}(z) + H(-z) \tilde{H}(z)$  to zero. This can be achieved by assuming the following relation between filters:

$$\begin{aligned} \tilde{G}(z) &= H(-z) \\ \tilde{H}(z) &= -G(-z) \end{aligned} \quad (26)$$

In the time domain the above two relations are (refer to Appendix 2.B):

$$\begin{aligned}\tilde{g}[n] &= (-1)^n h[n] \\ \tilde{h}[n] &= (-1)^{n+1} g[n]\end{aligned}\tag{27}$$

Using (26) the output becomes:

$$\hat{X}(z) = \frac{1}{2} X(z) (G(z) \tilde{G}(z) + H(z) \tilde{H}(z))\tag{28}$$

It should be obvious that for perfect reconstruction it is necessary that

$$G(z) \tilde{G}(z) + H(z) \tilde{H}(z) = 2\tag{29}$$

This condition together with the earlier found condition:

$$G(-z) \tilde{G}(z) + H(-z) \tilde{H}(z) = 0\tag{30}$$

are the only requirements necessary to achieve perfect reconstruction. Equations (29) and (30) together determine the filters used in the PRFB. A system just satisfying (29) and (30) is commonly termed a *biorthogonal* (Vetterli and Herley, 1992; Cohen et al., 1992) system. Additional conditions are necessary in order to achieve an *orthogonal* system. In such a system the filters are orthogonal to their even translates, that is:

$$\langle g(n-2l), h(n-2k) \rangle = 0\tag{31}$$

$$\langle g(n-2l), g(n-2k) \rangle = \delta_{kl}\tag{32}$$

$$\langle h(n-2l), h(n-2k) \rangle = \delta_{kl}\tag{33}$$

where  $l, k$  are integers. In the z-transform domain this means that:

$$G(z) H(z^{-1}) + G(-z) H(-z^{-1}) = 0\tag{34}$$

$$G(z) G(z^{-1}) + G(-z) G(-z^{-1}) = 2\tag{35}$$

$$H(z) H(z^{-1}) + H(-z) H(-z^{-1}) = 2\tag{36}$$

So, for an orthogonal PRFB, conditions (34) - (36) along with (29) and (30) have to be met. There are two interesting facts resulting from (34) - (36). One is that the length of  $g[n]$  has to be even. In addition the one rather elegant way to satisfy the conditions above is to impose the following:

$$H(z) = G(-z^{-1}) \quad (37)$$

which in the time-domain is:

$$h[n] = (-1)^n g[-n] \quad (38)$$

While in a biorthogonal system the reconstruction filters ( $\tilde{g}, \tilde{h}$ ) are different than the analysis filters ( $g, h$ ), in an orthogonal system the reconstruction filters are simply flipped versions of the analysis filters, i.e.,  $\tilde{g}[n] = g[-n]$  and  $\tilde{h}[n] = h[-n]$ . This is a result of the combination of equations (26) and (37).

In both the orthogonal and the biorthogonal systems, the two analysis filters can be seen as a *lowpass-highpass* pair.

### 2.3.2 The Discrete-time Wavelet Transform (DTWT)

How is the wavelet transform implemented using the type of filter bank discussed above? What is shown in Figure 6 is just one scale of decomposition and reconstruction. For more scales, the processing shown on  $x[n]$  would be repeated on the lowpass output. For the filters of a PRFB to be wavelet filters they must be *regular* (this term will be explained further below). The highpass output are the *wavelet coefficients* for that scale. (see Figure 7). This is continued until the desired number of scales are obtained. This is

known as the *pyramid* algorithm. For reconstruction, all that is needed are the wavelet coefficients at all scales and the lowpass output (*scaling coefficients*) at the last scale. This whole process can be written mathematically in the following manner:

Decomposition:

$$\begin{aligned} c_n^{j+1} &= \sum_m h(2n-m) c_m^j \\ d_n^{j+1} &= \sum_m g(2n-m) c_m^j \end{aligned} \quad (39)$$

Reconstruction:

$$c_n^{j-1} = \sum_m \tilde{h}(n-2m) c_m^j + \sum_m \tilde{g}(n-2m) d_m^j \quad (40)$$

where  $d_n^j$  are the wavelet coefficients and  $c_n^j$  are the scaling coefficients, with  $c_n^0$  being the original signal. The index  $j$  represents scale number, and is the discrete-time equivalent of the continuous-time scale factor  $a$ . The output scaling coefficients become the input to the next stage in the DTWT. Figure 7 illustrates two stages of such a scheme. The scaling coefficients,  $c^{j+1}$ , result from the convolution of  $c^j$  with a lowpass filter, followed by downsampling. The coefficients,  $c^{j+1}$ , are thus known as the “coarse” approximation of  $c^j$ . The wavelet coefficients,  $d^{j+1}$ , are a result of convolution with a highpass filter, and are commonly known as the “detail” coefficients. The detail coefficients along with  $c^J$  (where  $J$  equals the total number of scales) make up the wavelet representation of the signal  $c^0$ .

Finally, how is it that the filters in the pyramid algorithm are equivalent to wavelets?

The answer is that the cascade of filters converge to wavelet functions. To illustrate this let us rewrite (39) in terms of the original signal,  $c^0$ :

$$\begin{aligned} c_n^j &= \sum_m h_j(2n-m) c_m^0 \\ d_n^j &= \sum_m g_j(2n-m) c_m^0 \end{aligned} \quad (41)$$

where

$$\begin{aligned} g_1[n] &= g[n] & h_1[n] &= h[n] \\ g_{j+1}[n] &= \sum_k g_j[k] h[2k-n] & h_{j+1}[n] &= \sum_k h_j[k] h[2k-n] \end{aligned} \quad (42)$$

As mentioned above, in order for the filters of the filter bank to be considered wavelet filters they must be *regular* which means that as  $j \rightarrow \infty$  the function  $g_j$  should converge to a continuous function. Figure 5 illustrates a wavelet at three different scales resulting from the operations shown in Figure 8. As stated by Rioul (1993), the regularity requirement imposes some “smoothness” on the filter bank filters. The faster the convergence is of  $g_j$  to a “smooth” function, the higher the *degree of regularity*. There are advantages of having filters with high degrees of regularity which will be discussed in Chapters 4 and 5.

Now, how does one go about choosing or designing the wavelet filters (perfect reconstruction and regular) used in the pyramid algorithm? That is the issue that is addressed in Section 2.4 and in more detail in Chapter 5.

### 2.3.3 Linearity and Shift Properties of DTWT

In this section the linearity and shift properties of DTWT are presented and the DTWT and CWT are compared in light of these.

*Linearity:* Given two inputs  $c_1^j[n]$  and  $c_2^j[n]$  and using (39) the following holds:

$$Ac_1^j[n] + Bc_2^j[n] \rightarrow Ad_{c_1}^{j+1}[n] + Bd_{c_2}^{j+1}[n] \quad (43)$$

where  $d_{c_1}^{j+1}[n] = \sum_m g(2n-m) c_1^j[m]$  are the wavelet coefficients.

*Shift variance:* The DTWT is *not* shift invariant, i.e., if  $c_2^j[n] = c_1^j[n-n_0]$  then,

$$d_{c_2}^{j+1}[n] \neq d_{c_1}^{j+1}[n-n_0] \quad (44)$$

Proof:

$$\begin{aligned} d_{c_2}^{j+1}[n] &= \sum_m g(2n-m) c_1^j[m-n_0] \\ &= \sum_{m'} g(2n-m'-n_0) c_1^j[m'] = \sum_{m'} g((2n-n_0)-m') c_1^j[m'] \\ &= d_{c_1}^{j+1}[2n-n_0] \neq d_{c_1}^{j+1}[n-n_0] \end{aligned} \quad (45)$$

Therefore, unlike the CWT, the DTWT is linear but not shift invariant. This is due to the fact that the DTWT is a critically sampled representation, and the shift variance is due to the downsampling step of the filter bank implementation. The CWT on the other hand is

a highly redundant representation. Thus, in terms of sampling, the CWT and DTWT are at two extremes.

## 2.4 Designing Wavelet Filters

As mentioned earlier, one of the main strengths of the wavelet transform is the flexibility with regards to the basis wavelet,  $g(t)$ , used in the WT, or equivalently the filters,  $g[n]$  and  $h[n]$ , used in the filter bank of Figure 7. This choice will depend on the type of signal being analyzed and the goal of the application. One way to make a choice of wavelet is to test existing or “library” wavelet functions (there are many to choose from) and use the wavelet that performs the best based on some type of metric. This is one of the approaches that I use in my solution of a feature detection problem (refer to Chapter 3) and a representation (compression) problem (refer to Chapter 4). For the feature detection problem the metric is the magnitude of an output detection function. The test wavelet that results in the detection function with the highest magnitude (high confidence) is chosen as the best wavelet. For the compression problem the metric is the degree of compression obtained versus the distortion in the reconstructed signal. The wavelet leading to the greatest degree of signal compression with the least distortion is chosen.

As an alternative or a complement to the “search through a library of wavelets” approach mentioned above, the wavelet to be used can be chosen or designed *a priori* based on inherent characteristics of the signal being processed. This approach is explored in Chapter 3 using a feature in the seismic signal (P phase arrival) as a starting point in



choosing the analyzing wavelet by comparing the feature with “library” wavelets, as well as designing what I call pseudo-wavelets (refer to Chapter 3). Designing actual wavelets (as opposed to pseudo-wavelets) can be done using either frequency based, or time based methods. A frequency based method to design the filters in a PRFB is described by Vaidyanathan (1993). This method starts with the design of a halfband finite impulse response (FIR) filter using the Parks-McClellan or Remez algorithm. The variables used in the design include passband, stopband, and length values. These can be chosen based on the application. For example, one might choose the passband and stopband values to reflect the characteristics of the frequency spectrum of the signal being analyzed. This resulting filter is modified and then the minimum phase part is extracted using spectral factorization. This minimum phase filter is the lowpass filter in the filterbank. The remaining filters can be found using the relationships presented in Section 2.3. The details of this design algorithm can be found in Chapter 5 which concentrates on the wavelet design issue.

Though a number of different wavelets can be designed using the *Fourier*-based design method outlined above, the different *type* of wavelets that can be designed is limited. The wavelets designed tend not to be that different from one another and are usually already represented by existing wavelets. This is due to the limitations of the frequency based design method (halfband, FIR, etc.). A more appealing method is to design the wavelet directly from features in the signal to be analyzed. This is illustrated in a rough manner in Chapter 3 with regards to the design of pseudo-wavelets. In that case the feature of interest is modified slightly (mean is removed and ends are tapered to zero) and used as an analyzing filter. More desirable would be to use the features of the signal to design actual perfect

reconstruction wavelets. A method to achieve this goal is presented in Chapter 5 utilizing *linear prediction* and a recently developed time-domain based design procedure.

## 2.5 Applications

The two main areas of signal processing where the application of wavelet analysis has shown promise are 1) feature detection and 2) compression. Feature detection refers to detection of some particular type of behavior in a signal. The problem of feature detection proves especially difficult when a significant amount of noise is present. Compression refers to reducing the amount of information needed to represent a signal. The main purpose of this thesis is to develop and test wavelet transform based methods for these two types of applications.

### 2.5.1. Feature Detection

If a signal feature of importance is not strong in a raw signal, it is often necessary to process the signal to enhance the feature in question. Even when a feature of interest is strong, it can be helpful to develop some type of automatic detection procedure. The fact that the basis function of the wavelet transform can be suited to features in a signal and that the resulting transform output is a representation of the signal over different scales makes the WT well suited to feature detection problems. The advantage of the WT over traditional Fourier based techniques is particularly important for transient signals. This idea is applied to a practical situation involving seismic signals and the results are pre-

sented in Chapter 3. Also in the same chapter more details about the WT and feature detection are given. In Chapter 5 wavelets are designed to help with some of the problem feature detection cases of Chapter 3.

### *2.5.2. Compression*

There are two types of data compression: lossy and lossless. In lossless compression no information is lost and these methods involve efficient bit allocation taking advantage of the relationships that exist between data points. For example the difference between neighboring data points is often significantly smaller than the actual data point values, which will in turn can be represented using fewer bits. Transform coding on the other hand is called lossy coding, since information is lost in the process. Wavelet transform compression techniques fall into this category. The advantage of the WT for compression is that the data in the wavelet domain is less correlated than the original data, and is often quite sparse. Thus, thresholding and quantization of the wavelet coefficients can be used to code the data. Chapter 4 contains the results of the application of a wavelet transform compression technique to electrocardiogram (ECG) data, along with a more thorough description of wavelet based compression methods.

## **2.6 Discussion and Conclusions**

In this chapter the wavelet transform has been introduced, focusing on its advantages over traditional signal analysis methods. In addition to basic definitions, a few important

properties, as well as methods of implementation are also presented. The purpose was to provide basis and motivation for the problems that are investigated in the following chapter. It should be noted that the wavelet transform is not always superior to traditional signal processing tools, and should not be viewed as a replacement of other transforms such as the Fourier transform. For example, for the representation of pure sinusoids such as those found in musical compositions, the Fourier representation is more than sufficient. The wavelet transform should instead be viewed as an additional tool that can be used to process signals, thus providing a different insight. In the next chapters it will be shown how the wavelet transform can be used to help in two important types of problems. Chapter 3 deals with detecting seismic arrivals in seismograms, while Chapter 4 shows how the WT is an effective tool in signal compression. An important issue in both Chapters 3 and 4 is how one chooses or designs a wavelet for a particular application. Chapter 5 presents a wavelet design algorithm that uses linear prediction to help address this issue.

## **Appendix 2.A The Resolution of Identity Theorem and the Inverse Wavelet Transform**

As stated in Section 2.1, the definition of the IWT (9) is derived using the resolution of identity theorem (10). In this appendix, first the resolution of identity theorem will be justified and then the IWT will be derived from it. Given that  $f_1(t)$ ,  $f_2(t)$ , and  $g_{ab}(t)$  are

finite energy functions and the definitions for  $g_{ab}(t)$  and  $C_g$  given in Section 2.1, we can proceed with the proof of (10).

*Proof of resolution of identity theorem:*

$$I = \int_{-\infty}^{\infty} \frac{1}{a^2} \int_{-\infty}^{\infty} \langle f_1, g_{ab} \rangle \langle g_{ab}, f_2 \rangle db da \quad (46)$$

By using Parseval's theorem, i.e.,  $\int_{-\infty}^{\infty} x(t) y^*(t) dt = \int_{-\infty}^{\infty} X(\omega) Y^*(\omega) d\omega$ , the following is obtained:

$$I = \int_{-\infty}^{\infty} \frac{1}{a^2} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} F_1^*(\omega) G_{ab}(\omega) G_{ab}^*(\omega') F_2(\omega') db da d\omega d\omega' \quad (47)$$

Now remembering that  $G_{ab}(\omega) = \mathcal{F}\left(\frac{1}{\sqrt{|a|}} g\left(\frac{t-b}{a}\right)\right) = \sqrt{|a|} \exp(-jb\omega) G(a\omega)$ ,

(using Fourier transform properties), (47) can be rewritten as such:

$$I = \int_{-\infty}^{\infty} \frac{1}{a^2} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} |a| \exp(-jb(\omega - \omega')) F_1^*(\omega) G(a\omega) G^*(a\omega') F_2(\omega') db da d\omega d\omega' \quad (48)$$

Using the fact that  $\int_{-\infty}^{\infty} \exp(-jb(\omega - \omega')) db = \delta(\omega - \omega')$  the following relation is obtained:

$$I = \int_{-\infty}^{\infty} \frac{1}{|a|} \int_{-\infty}^{\infty} G(a\omega') G^*(a\omega') F_1^*(\omega') F_2(\omega') d\omega' da \quad (49)$$

Finally, making the substitution,  $\omega'' = \omega a$   $d\omega'' = \omega da$ , one gets:

$$I = \int_{-\infty}^{\infty} \frac{1}{|\omega''/\omega|} \int_{-\infty}^{\infty} G(\omega'') G^*(\omega'') F_2(\omega') F_1^*(\omega') d\omega' \frac{d\omega''}{\omega} \quad (50)$$

$$I = \langle f_1, f_2 \rangle \int_{-\infty}^{\infty} \frac{|G(\omega'')|^2}{|\omega''|} d\omega'' \quad (51)$$

which equals the right side of (10),  $C_g \langle f_1, f_2 \rangle$ . Q.E.D.

*Derivation of Inverse Wavelet Transform using Resolution of Identity Theorem:*

Having proven the resolution of identity theorem, it can be used to derive the IWT by setting  $f_2(t') = \delta(t' - t)$ . The resolution of identity theorem then becomes:

$$\begin{aligned} C_g \langle f_1, f_2 \rangle &= C_g f_1(t) = \int_{-\infty}^{\infty} \frac{1}{a^2} \int_{-\infty}^{\infty} \langle f_1, g_{ab} \rangle \langle g_{ab}, \delta(t' - t) \rangle db da \\ C_g f_1(t) &= \int_{-\infty}^{\infty} \frac{1}{a^2} \int_{-\infty}^{\infty} \langle f_1(t'), g_{ab}(t') \rangle g_{ab}(t) db da \end{aligned} \quad (52)$$

Using the definition for the wavelet transform (6) and the definition for

$g_{ab}(t) = \frac{1}{\sqrt{|a|}} g\left(\frac{t-b}{a}\right)$ , the above relation can be written as such:

$$C_g f_1(t) = \int_{-\infty}^{\infty} \frac{1}{a^2} \int_{-\infty}^{\infty} W_{f_1}(a, b) \frac{1}{\sqrt{|a|}} g\left(\frac{t-b}{a}\right) db da \quad (53)$$

As can be seen, by dividing (53) by  $C_g$  one obtains the equation for the inverse wavelet transform as presented in (9).

## Appendix 2.B Linear Convolution, Downsampling, Upsampling, etc.

*Linear Convolution:* The convolution of a discrete sequence  $x[n]$  with another sequence  $h[n]$  is defined as such:

$$y[n] = x[n] * h[n] = \sum_{k=-\infty}^{\infty} x[k] h[n-k] \quad (54)$$

Convolution in the time domain is just a product in the z-transform domain, i.e.:

$$x[n] * h[n] \leftrightarrow X(z) H(z) \quad (55)$$

where the z-transform is defined in Section 2.3.1.

*Downsampling:* This term refers to the keeping of every  $M^{th}$  sample of a sequence and removing the rest. Thus, if the original sequence is  $x[n]$  and the downsampled sequence is called  $x_d[n]$  then:

$$x_d[n] = x[nM] \quad (56)$$

In the z-transform domain downsampling becomes:

$$X_d(z) = \frac{1}{2} \{ X(z^{1/M}) + X(-z^{1/M}) \} \quad (57)$$

*Upsampling* - In this operation  $M$  zeros are inserted between every sample of a sequence.

So, if the original sequence is  $x[n]$ , then the upsampled sequence,  $x_u[n]$ , is defined as:

$$x_u[n] = \begin{cases} x[n/M], & n = \text{multiple of } M \\ 0, & \text{otherwise} \end{cases} \quad (58)$$

In the z-transform domain this operation is represented in the following manner:

$$X_u(z) = X(z^M) \quad (59)$$

*Time Reversal* - The time reversed version of  $x[n]$  is represented as  $x[-n]$ . In the z-transform domain time reversal is derived as:

$$X_r(z) = \sum x[-n] z^{-n} = \sum x[n'] (1/z)^{-n'} = X(z^{-1}) \quad (60)$$

*Multiplication of  $x[n]$  by  $(-1)^n$  -*

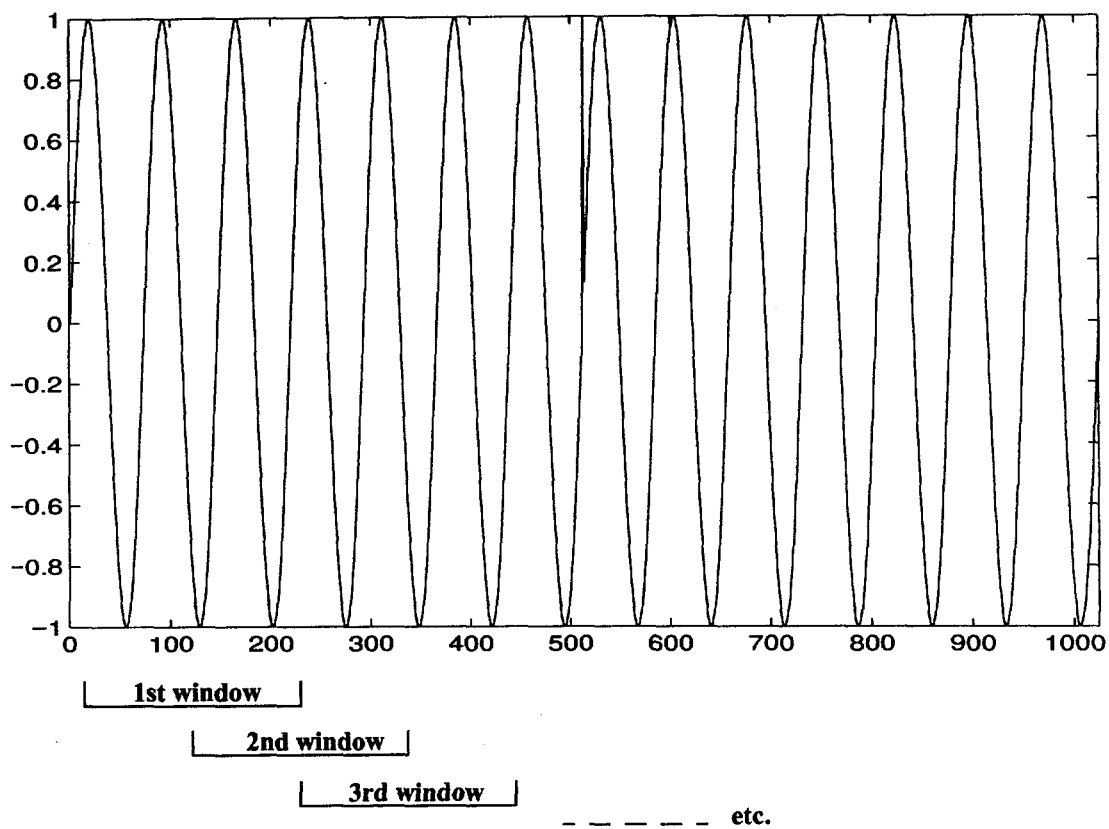
$$X_{zr}(z) = \sum (-1)^n x[n] z^{-n} = \sum x[n] (-z)^{-n} = X(-z) \quad (61)$$

Therefore this operation leads to reversal in the z-transform domain, i.e.,

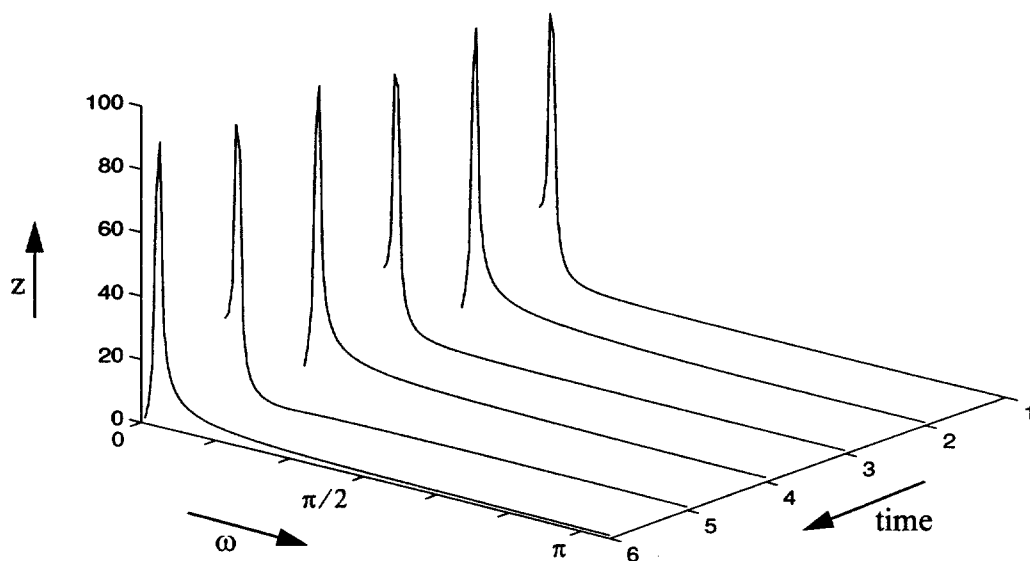
$$X(-z) \leftrightarrow (-1)^n x[n].$$



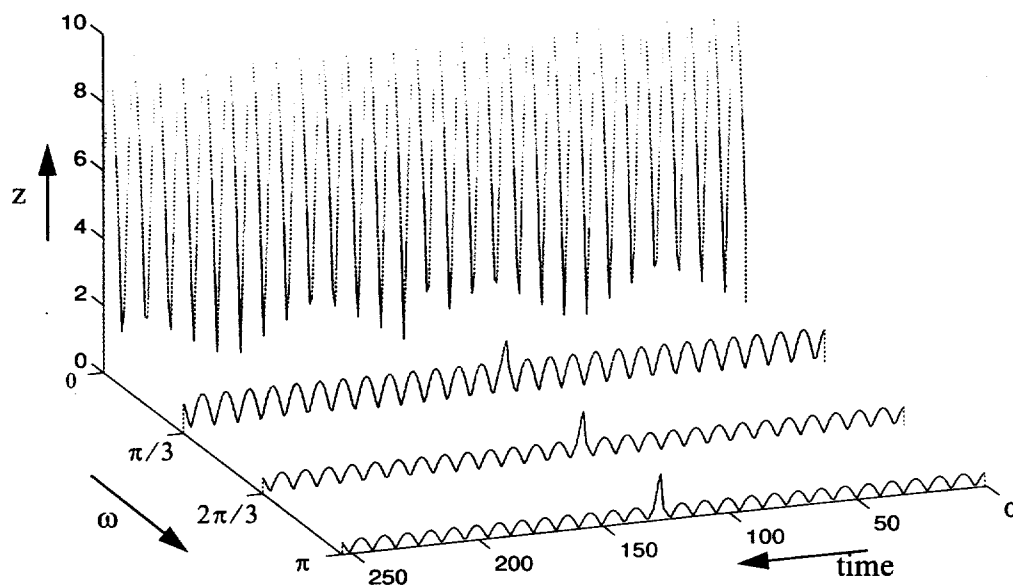
**FIGURE 1. Delta function superimposed on a sinusoid. Below the plot is an illustration of how the overlapping windows are set up for the STFT plots of Figures 2 and 3. For each windowed segment the FT is calculated.**



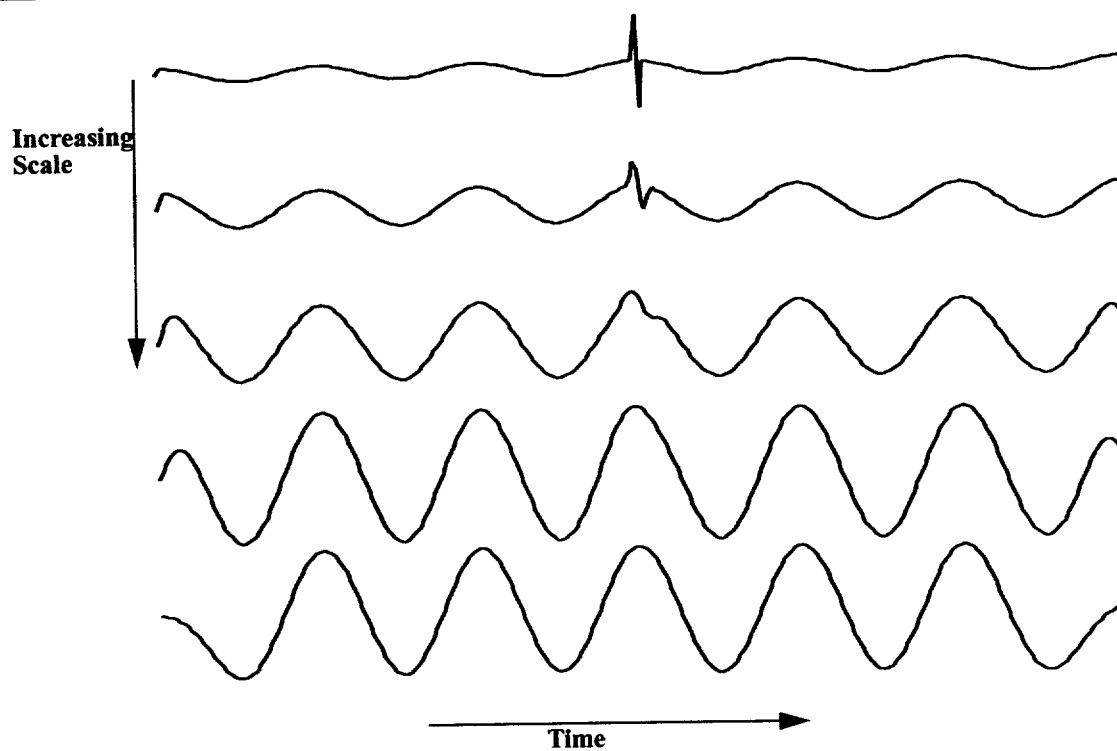
**FIGURE 2.** STFT of signal in Figure 1 using wide windows. In this case the STFT is taken with 6 overlapping windows (x-axis) each of length 256. The z-axis represents the absolute value of the frequency response.



**FIGURE 3.** STFT of signal in Figure 1 using narrow windows. In this case the STFT is taken with 256 overlapping windows (x-axis) each of length 8. The z-axis represents the absolute value of the response.



**FIGURE 4. WT of signal in Figure 1. Notice localization of delta function in low scales and resolution of sinusoid in higher scales**



**FIGURE 5. A wavelet for three different scale values (scale increasing downwards)**

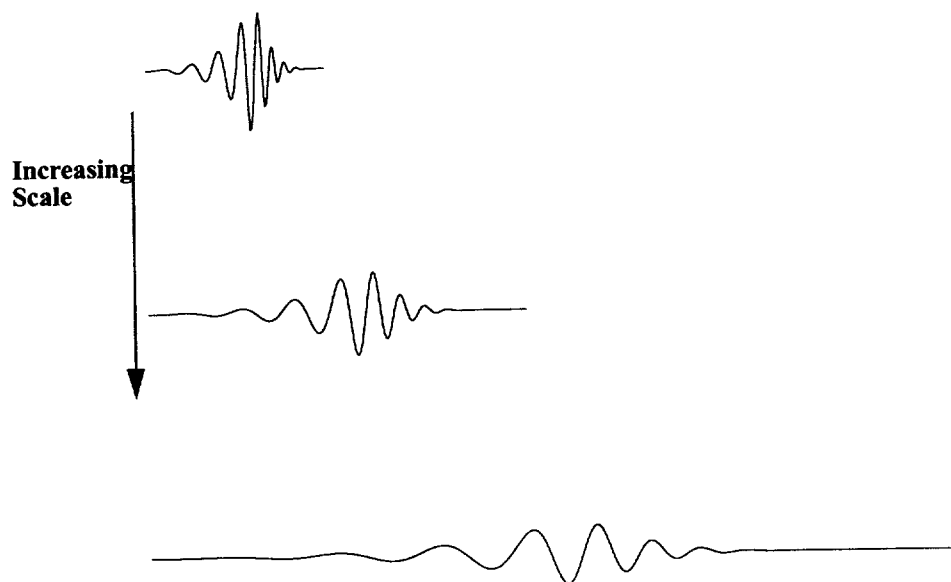


FIGURE 6. Two-band filter bank

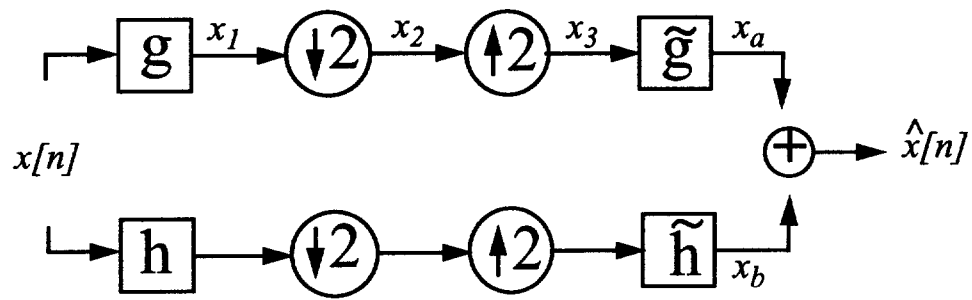
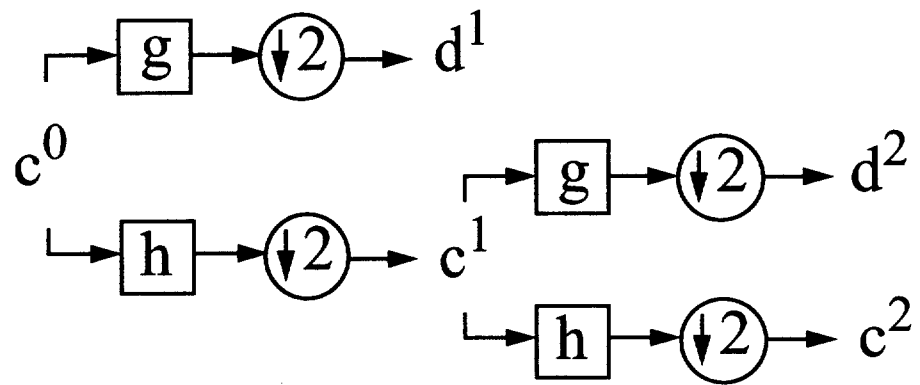
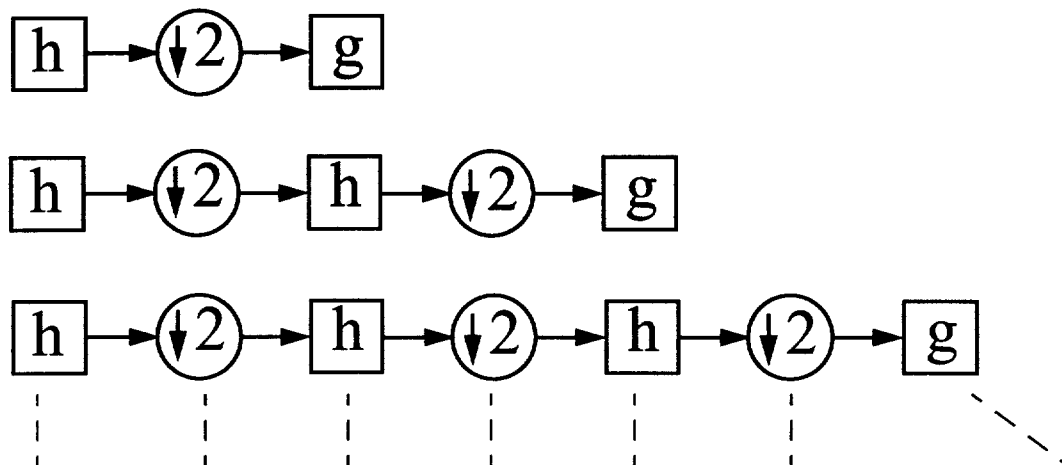


FIGURE 7. Two stages (scales) of wavelet decomposition



**FIGURE 8.** Iterations of filters that lead to wavelets. The iterations below lead to the same wavelet at different scales such as those shown in Figure 5. The dashed lines signify the continuation of the iteration to  $J$  scales.



**CHAPTER 3***Wavelet Transform  
Methods for Phase  
Identification in Three-  
component  
Seismograms***3.0 Summary**

In this chapter the focus is on the use of the wavelet transform in terms of feature identification. The method that is presented helps to locate the arrivals in single station three-component short-period seismograms using polarization and amplitude information contained in the wavelet coefficients of the signals. The main idea is that strong features of the seismic signal will appear in the wavelet coefficients across several scales. Each component of a three-component short period seismogram is decomposed using the discrete-time wavelet transform (DTWT). Polarization and rectilinearity functions are calculated for each scale. A composite rectilinearity function is developed based on information present

across scales. A second composite function based on the relative wavelet coefficient amplitudes of the transverse and radial components is also constructed. These composite functions prove to be very effective at identifying important arrivals. Results are compared with arrival times picked by an analyst.

### 3.1 Introduction

The three-component seismogram represents motion on a ground detector in three orthogonal directions; two in the horizontal plane and one vertical. The seismic signal has several features that represent various types of phase arrivals (Aki and Richards, 1980). The two most significant are the P (compressional) and S (shear) phase arrivals (refer to Figure 6). The arrival of the P phase precedes the arrival of the S phase as is illustrated in Figure 1. Accurately determining the time of these arrivals is important in determining the location of an event, such as an earthquake or explosion. Traditionally this has been done using data from various locations, however recently research has been carried out in trying to determine this information more accurately from just a single station seismogram (Kanasewich, 1981; Roberts et al., 1989; Magotra et al., 1989; Jarpe and Dowla, 1991), especially for small regional events. In this work, I extend the polarization analysis methods of Kanasewich (1981) and others (Park, 1987; Means, 1972; Fowler et al., 1967), to the wavelet coefficients instead of the original signal. The main motivation for this is that significant features in the signal will be retained over several resolutions ("scales" in wavelet transform terminology), while less important features will be present in only one

or two scales. This idea of retention of features over scales was first introduced in the area of vision by Marr (1982) and has played an important role in image processing (Mallat, 1992). Just as the inter-scale behavior of edges is analyzed in images, I look at the inter-scale behavior of transitions in the seismic signal.

The algorithm developed has two main parts. In part 1, three short-period components ( $se$ ,  $sn$ , and  $sz$ ) are processed using the discrete-time wavelet transform (DTWT) producing the wavelet coefficients as output ( $d_e^j$ ,  $d_n^j$ , and  $d_z^j$ ). At each scale,  $j$ , a 3x3 covariance matrix is used to find the rectilinearity function ( $F^j$ ). These rectilinearity functions are used to construct a “composite” rectilinearity function ( $C_F$ ). The location of the maximum for this function is very effective at locating the P arrival. The maximum eigenvector at this location is used to find the back azimuth. In the second part of the algorithm this back azimuth is used to rotate  $se$  and  $sn$  to get the radial and transverse components,  $sr$  and  $st$ . These are then processed using the DTWT resulting in the wavelet coefficients  $d_r^j$  and  $d_t^j$ . At each scale, a *transverse over radial amplitude ratio* is calculated, all of which are combined to produce a second composite function ( $C_t$ ) which is used to locate the S phase arrival. Both  $C_F$  and  $C_t$  are powerful functions in that they highlight features that are present across scales while suppressing those features that are not.

In the following sections details of the algorithm as well as results are presented. Theoretical background as well as a description of the algorithm are given in Section 3.2. The results of an application of the algorithm to a set of events is presented in Section 3.3.



Summary and conclusions are found in Section 3.4. Appendix 3.A contains some mathematical background on how polarization is affected by the wavelet transform.

## **3.2 Theoretical Background**

### *3.2.1 The Wavelet Transform and Feature Identification*

As stated in Section 2.5.1 the wavelet transform has shown potential as a tool for the purposes of feature identification, particularly in the case of transient signals (Kronland-Martinet, 1988; Frisch and Messer, 1992; Szu et al., 1993; Trejo and Shensa, 1993; Li et al., 1995). This is due to the ability of the wavelet transform to resolve features at various scales and the fact that the wavelet used can be matched to the particular feature of interest. Formalization of wavelet theory was actually initiated by work on seismic signals (Goupillaud et al., 1984; Grossmann and Morlet, 1984). Recent work commenting on the potential application of the wavelet transform to seismic data has been carried out by Yomogida (1994), and Chakraborty and Okaya (1995). The work presented in this chapter represents an application of the wavelet transform to seismic data that utilizes the key advantages of wavelet analysis, and helps in further establishing the wavelet transform as a valuable seismic analysis tool.

The two aspects of the wavelet transform that are of use for feature identification have been mentioned in the previous paragraph as well as in Chapter 2. The first is that the analyzing wavelet can be chosen based on the feature that is to be identified. The other is the

multi-scale representation produced by the wavelet transform. The result is that at low scales high frequency behavior is localized, while at high scales (when the wavelet is stretched out) low frequency features are better resolved. This is of significant benefit when one is dealing with signals, such as the seismic signal, containing features with various frequency characteristics. This aspect was clearly demonstrated with the sinusoid plus delta function example of Chapter 2 (refer to Figures 1 and 4 of Chapter 2). More detail about the wavelet transform and feature identification as it relates to the seismogram is given in Section 3.2.B.

Since the seismic data being analyzed is discrete-time, the DTWT is used for the wavelet processing portion of the algorithm. The DTWT is implemented using the subband coding scheme where the wavelet is the highpass filter ( $g$ ), and the lowpass filter ( $h$ ) is the scaling function (Akansu, 1992; Daubechies, 1992; Vetterli and Kovacevic, 1995) as described in Chapter 2. As can be noticed from Figure 7 in Chapter 2, there is a downsampling step which results in the number of points in the wavelet coefficients,  $d^j$ , being scale dependent;  $d^{j+1}$  has half the number of points as  $d^j$ . Since my algorithm involves inter-scale comparisons, this type of decomposition is not the best because features should be at the same location in each scale for ease of analysis. Therefore, the wavelet coefficients,  $d^j$ , are interpolated so that the output at each scale contains the same number of points as the original signal. This interpolation is accomplished using upsampling and convolution with a combination of the filters,  $\tilde{h}$  and  $\tilde{g}$  ( $\tilde{g}[n] = (-1)^n h[n]$  and  $\tilde{h}[n] = (-1)^{n+1} g[n]$ ). Figure 2 shows how this interpolation is implemented for three

scales. The result is a set of coefficients that are interpolated versions of  $d^j$ . Further references in this chapter to  $d^j$  will refer to the *interpolated* coefficients. This type of decomposition is often called *multiresolution analysis* (Mallat, 1989; Vetterli and Kovacevic, 1995; Rioul and Vetterli, 1991). Figure 3 shows the vertical component of a three component short-period seismogram and eight (multiresolution) wavelet scales. (i. e.  $d^j$  for  $j = 1, 2, 3, \dots, 8$ ) The lowest scale is at the top and the highest scale at the bottom.

As mentioned many times already, the wavelet used in the wavelet transform can be chosen based on the application. Certain conditions on the wavelet  $g$  and the scaling function  $h$  have to be met (Daubechies, 1992; Vetterli and Kovacevic, 1995) for the filter bank to be a perfect reconstruction filter bank (it will be discussed later how these conditions might be able to be relaxed for this study). The wavelets (Daubechies, 1992; Vetterli and Herley, 1992) used in this study are shown in Figure 4. This wavelet “library” was chosen based on the expected shape of the seismic phase arrivals. In Figure 5 the shapes of a few P phase arrivals are shown to illustrate how their shapes are similar to the wavelets in Figure 4. Notice how the first two examples in Figure 5 look similar to the Vetterli-22 wavelet, while the third example looks similar to the biorthogonal-18 and Daubechies-12 wavelets. For part 1 of the algorithm the P location accuracy is not affected significantly by which wavelet in Figure 4 is used. However, as will be discussed later, in part 2 of the algorithm wavelet choice proved to be more important.

### 3.2.2 *Characterization of Seismograms by the Wavelet Transform*

Seismic signals are commonly modeled as the convolution of a “source” with a filter representing the transmission response of subsurface media. The “source” refers to a series of vibrations in the Earth that can be due to an earthquake or explosion. These vibrations set off a traveling wave that has longitudinal as well as shear components (refer to Figure 6). The longitudinal or compressional motion is represented on the seismogram by the P phase, while the transverse motion is represented by the S phase (Figure 1). Since the subsurface media cannot be assumed to be homogeneous, the wave encounters discontinuities in the media through which selective reflection and transmission takes place. Ricker (1940) first hypothesized and later Morlet et al. (1982), demonstrated experimentally that this “scattering” is frequency dependent. Low frequencies are transmitted more easily while high frequencies are not. For intermediate frequencies there is frequency dependent dispersion. An effect of this scattering is the broadening of the initially sharp source impulse, resulting in seismograms composed of wavelet-like structures. Experimental demonstrations of this effect were first presented by Ricker (1940), and more extensively in a later report (Ricker, 1953).

The suitability of applying the wavelet transform to seismic signals can thus be viewed in terms of the type of decomposition as well as in terms of pattern matching. Based on the frequency dependent nature of the scattering mentioned above, and on the fact that the features of interest in this study are onset times, the basis functions used to analyze the seismic signal should be well defined in the time domain, as well as in the frequency

domain. The basis functions of the wavelet transform are well defined in the two domains, and in fact have constant  $\Delta t/T$  and  $\Delta f/f$ , where  $\Delta t$  is the duration or breadth of the wavelet,  $T$  is the mean period,  $\Delta f$  is the bandwidth, and  $f$  is the mean frequency. In a wavelet transform decomposition one would expect the scattered arrivals to be present in some scales, but not present consistently across scales. The direct arrivals (P and S), on the other hand, should be present across several scales. Thus, interscalar analysis should isolate the direct arrivals while neglecting the arrivals due to scattering.

The characterization of the seismogram as a combination of wavelet-like structures justifies the use of wavelets to match arrival shapes. The arrival shapes will vary with event as the source wave travels through different media and distances for each event. Thus, the particular wavelet to use in the wavelet decomposition of a seismogram will depend on the event. This proves to be important in S arrival detection (refer to Results section). The wavelet that best approximates the S arrival shape will result in strong correlations in the wavelet scales. Thus, the S arrival detector function,  $C_t$  (described in detail in the Algorithm subsection), will have the highest magnitude, and therefore the highest level of confidence with regards to S arrival onset detection when the S arrival shape and wavelet shape are closest. The algorithm which is presented in the next subsection uses this idea to pick S onset time.

For cases where the P arrival *shape* can be clearly identified, it is observed that the wavelet that leads to the highest magnitude  $C_t$  is nearest in shape to the P arrival. This observation motivates the idea of directly using the P arrival shape to aid in S arrival

detection. This can be achieved by either using a wavelet from a library that best matches the P arrival shape or by designing a wavelet based on the P arrival shape. This approach assumes that there is some relationship between the P arrival shape and the S arrival shape. The fact that for a given event, both arrivals originate from the same source and travel through the same media would indicate that the P and S shapes should be correlated. For the simple case of homogeneous subsurface media the shapes are similar, with the S arrival generally being broader than the P arrival. In a heterogeneous media where the velocity gradients are not large the arrival shapes will still be fairly similar with possible differences in the very high frequencies (very low scales). Even in the situation of a heterogeneous media containing large velocity gradients, the distortions in arrival shapes due to scattering will be frequency dependent and thus isolated in certain scales and will not be consistent across scales.

In summary, the wavelet decomposition employing the “best wavelet” captures (and with appropriate processing, emphasizes) the features of a phase across the various scales. Hence, we are able to develop algorithms for phase identification by observing these features across scales. Since there are reasons to believe that the P and S arrivals are correlated (in energy, spectrum, or shape), the choice of wavelet to use for S arrival detection can be based on the P arrival. Although the problem of choosing the “best wavelet” is the focus of Chapter 5, observations on this issue are discussed in Section 3.2.4.

### 3.2.3 The Algorithm

As mentioned in the introduction, the algorithm is made up of two main parts. The first part results in the detection of the P arrival and the second part results in the detection of the S arrival.

#### *i) Part 1: Locating the P Arrival*

It is known that the P phase is a compressional body wave that is highly linearly polarized (refer to Appendix 3.A for polarization definitions). Thus, a metric that measures the degree of linear polarization of a wave would be very helpful in detecting P arrivals. Such a metric is defined by Kanasewich (1981). It is known as the rectilinearity function and is defined as:

$$F = 1 - (\lambda_2 / \lambda_1)^p \quad (1)$$

where  $\lambda_1$  is the largest eigenvalue and  $\lambda_2$  is the second largest eigenvalue of the 3 by 3 covariance matrix:

$$\mathbf{M} = \begin{bmatrix} Var(\mathbf{x}) & Cov(\mathbf{x}, \mathbf{y}) & Cov(\mathbf{x}, \mathbf{z}) \\ Cov(\mathbf{x}, \mathbf{y}) & Var(\mathbf{y}) & Cov(\mathbf{y}, \mathbf{z}) \\ Cov(\mathbf{x}, \mathbf{z}) & Cov(\mathbf{y}, \mathbf{z}) & Var(\mathbf{z}) \end{bmatrix} \quad (2)$$

where  $\mathbf{x}$ ,  $\mathbf{y}$ , and  $\mathbf{z}$  would be the three components in case of a three component seismic signal.  $Var(\mathbf{x}) = Cov(\mathbf{x}, \mathbf{x})$  and the covariance of  $\mathbf{x}$  and  $\mathbf{y}$  is defined as:

$$Cov(\mathbf{x}, \mathbf{y}) = \frac{1}{N} \sum_{i=1}^N (x(i) - \mu_x) (y(i) - \mu_y) \quad (3)$$

where  $\mu_x$  is the mean value of  $x$ . The function  $F$  is equal to one at points at which a wave is fully linearly polarized and equal to zero at points where there is no polarization.

The first step in the algorithm is the wavelet processing of each of the short-period components ( $se$ ,  $sn$ , and  $sz$ ) of a three component seismogram. After wavelet processing of the three components, the outputs are  $d_e^j$ ,  $d_n^j$ , and  $d_z^j$ . At each scale, a covariance matrix  $M^j[i]$  is found at each point  $i$  over a  $T$  point window (refer to Figure 7) using equation (2) with  $d_e^j$  substituted for  $x$ ,  $d_n^j$  substituted for  $y$ , and  $d_z^j$  substituted for  $z$ . The window length (ranging from 5 seconds to 25 seconds) is determined using a measure called the varimax norm (Wiggins, 1978). Refer to Appendix 3.B for details. The eigenvalues for each  $M^j[i]$  are found and the rectilinearity functions  $F^j$  (using  $p = 1$ ) are constructed. So, the result is a rectilinearity function for each scale. The next step is the most crucial in part 1. The idea that edges in an image are present over several scales has been used in the area of image representation (Mallat, 1992). One of the first presentations of this idea was in the area of vision by Marr (1982). An analogy can be made between edges in an image and transitions in  $F^j$ . The P phase arrival produces a transition (discontinuity) in the rectilinearity function. However, there are other changes of polarization present in any one given scale. Therefore, to capture only the most linearly polarized portion of a wave (which should be the P phase arrival), a composite rectilinearity function is developed by combining the rectilinearity functions of each scale. This function is defined as:



$$C_F = \prod_j F^j \quad (4)$$

For  $j > 8$ ,  $F^j$  is uniformly close to unity (for our data set) which does not contribute to  $C_F$ . Thus, a wavelet decomposition of eight scales is carried out for the calculation of  $C_F$ . This function represents an inter-scale comparison of rectilinearity. The position at which this function is a maximum is chosen as the P arrival time. This proves to be an excellent method for determining P arrival time as can be seen in the case shown in Figure 10. Refer to the results section for more details about the effectiveness of  $C_F$  in locating P arrivals.

The eigenvector associated with the maximum eigenvalue at the estimated P arrival time is used to calculate the back azimuth (Jarpe and Dowla, 1991). This calculation is carried out at each scale as well as on the original signals. Since the original three component seismogram often contains noise it is found that a more accurate back azimuth is found by looking at the third scale and higher. This is due to the fact that high frequency noise shows up in the first one or two scales, and higher scales are “cleaner”.

## *ii) Part 2: Identification of the S Arrival*

The S phase arrival represents a shear wave with particle motion in the transverse direction (Kanasewich, 1981) with a velocity that is approximately 50-60% that of the P phase velocity. Since the S phase motion is in the transverse direction, it generally has a higher amplitude in the transverse component relative to its amplitude in the radial or vertical component. Thus, to locate this phase arrival the amplitude ratio of the transverse to radial

components of the seismogram is analyzed. The first step is to rotate  $se$  and  $sn$  into the radial and transverse directions using the back azimuth ( $\theta$ ) found in part 1 and the following relation for each point  $i$ :

$$\begin{bmatrix} sr[i] \\ st[i] \end{bmatrix} = \begin{bmatrix} \sin\theta & \cos\theta \\ -\cos\theta & \sin\theta \end{bmatrix} \begin{bmatrix} se[i] \\ sn[i] \end{bmatrix} \quad (5)$$

These rotated components are then processed by the wavelet transform resulting in the wavelet coefficients  $d_r^j$  and  $d_t^j$ . Here a ten scale decomposition is taken. It may seem that  $d_r^j$  and  $d_t^j$  could have been more easily obtained by just rotating the coefficients  $d_e^j$  and  $d_n^j$ . However, it turns out that for part 2 various wavelets are tested, thus, it is necessary to reprocess with the wavelet transform. At each scale the following ratio is calculated:

$$ttr^j = \frac{env(d_t^j)}{env(d_t^j) + env(d_r^j)} \quad (6)$$

where  $env(x)$  is called the envelope function and is defined as:

$$env(x) = \sqrt{x^2 + h^2} \quad (7)$$

where  $h$  is the Hilbert transform of  $x$ . The envelope function is used to avoid divide by zero problems. For the data in the test set, the function  $ttr^j$  becomes very “flat” for  $j > 10$ . Therefore, it is not necessary to decompose beyond ten scales, as no features are present in the higher scales. A composite function  $C_t$  is constructed in the same manner as  $C_F$ :

$$C_t = \prod_j ttr^j \quad (8)$$

By using a very simple thresholding method, the function  $C_t$  is used to locate the point of the S phase arrival. The first point after the P arrival time that has a value that is at least one-half the maximum of  $C_t$  is chosen as the S onset time. The peak of  $C_t$  itself is not used, because it locates the time at which the S arrival attains its highest magnitude, which is a few seconds after the onset time. Figure 11 shows two examples of the use of  $C_t$ .

As mentioned earlier, various wavelets are used in the wavelet transform decomposition. The choice of wavelet used in processing is important in how well  $C_t$  locates the S arrival onset time. The wavelets shown in Figure 4 are each used and the resulting  $C_t$  that has the highest amplitude peak is used in locating the S arrival. The idea behind this is that the wavelet that is closest in shape to the S phase will produce the best match in the wavelet scales and thus produce the  $C_t$  with the greatest dynamic range. Refer to the results section for an elaboration of which wavelets were successful for the test events.

### 3.2.4 Possible Refinements of Algorithm

As argued in an earlier subsection, the P arrival and the S arrival should generally be of similar shapes. Thus, the next step would be to design a wavelet directly from the P phase arrival and use it to construct  $C_t$ . A thorough presentation of this idea is given in Chapter 5 which focuses on the wavelet design issue. Here a preliminary investigation of this idea is carried out. For a few cases where the wavelet choice proved especially important, a

pseudo-wavelet is constructed from the P phase arrival (as located by  $C_F$ ) and is used in part 2 of the algorithm to produce  $C_t$ . It is being called a pseudo-wavelet because the perfect reconstruction and regularity conditions have been ignored. The only conditions being imposed are that of finiteness and zero-mean, as well as a tapering to zero. This is partially acceptable since the algorithm being used does not involve any reconstruction. However, because the perfect reconstruction and regularity conditions are not met these pseudo-wavelet will not retain their shape across many scales (refer to Chapter 2 as well as to designed wavelets of Chapter 5), and thus the resulting decomposition will not be of the subband type. In addition the lack of regularity could lead to the introduction of artificial discontinuities (due to the filter and not the signal) in the decomposition domain. Nonetheless, for this preliminary test, the few conditions imposed will be sufficient.

Figure 8 shows a pseudo-wavelet that is constructed from the P arrival. Below it is the biorthogonal-18 wavelet, which is the wavelet that produced the maximum peak in  $C_t$  for the particular event. Figure 9 shows the  $C_t$  produced by the pseudo-wavelet as well as the  $C_t$  resulting from the use of the biorthogonal-18 wavelet. Both functions find the S arrival time to within one second of the analyst pick. In contrast, using the Vetterli-22 wavelet to produce  $C_t$  for this event resulted in an S arrival time that was significantly different than the analyst pick. This is due to the fact that for this event the biorthogonal-18 wavelet is closest in shape to the arrival shape and thus produces the maximum peak in  $C_t$  (and the best S arrival pick time). The main purpose of this illustration is to show that it is possible to locate the S arrival using information about the P arrival shape, and that this can be

done by either using known wavelets to closely match P arrival shape (similar to the expected S arrival shape), or design a wavelet directly from the P arrival shape. These observations are reinforced in Chapter 5 where a wavelet design algorithm is utilized to design *actual* wavelets (perfect reconstruction and regular) based on the P arrival in order to locate the S arrival.

### 3.3. Results

The methods described in Section 3.2 are applied to a set of actual seismic data and the results are presented in this section. The data used is from a group of events recorded from a single station (39.68°N, 3.96°W) in the Sonseca array in Spain and covers a five year period beginning in 1990. The data set consists of east, north, and vertical short period components sampled at 20 Hz. A map of the event locations is shown in Figure 12. The test set contains 23 events, all of which are within 2500 kilometers from the recording station. A bulletin presenting information about each of the events is presented in Table 1.

#### 2.3.1 P Locator Results

As mentioned in Section 3.2,  $C_F$  is extremely effective at locating the P arrival time. A compilation of P locator results is shown in Table 2. The algorithm pick times are compared to analyst pick times. IASPE91 model pick times are also presented in the fourth column of Table 2. An illustration of how  $C_F$  is used as a P locator function is shown in Figure 10. A comparison is made between the performance of this algorithm and the

short-term-average to long-term-average (STA/LTA) method (Earle and Shearer, 1994). This is presented in Figure 13. As can be noticed, the algorithm compared favorably with STA/LTA. There were two cases for which STA/LTA produced a significant error (-89.61 seconds and 82.43 seconds) in P arrival time. The fact that these cases had low signal-to-noise ratios highlights the advantage of providing a pick time based on analysis of information over several wavelet scales. As mentioned earlier, signal noise in the wavelet domain tends to be concentrated in the first few scales. The results show that using  $C_F$  is a consistently accurate method to automatically find the P arrival time.

TABLE 1. Bulletin information for test set cases

| Case | Latitude | Longitude | Depth (km) | Magnitude | Distance (km) |
|------|----------|-----------|------------|-----------|---------------|
| 1    | 35.63    | 11.66     | 22.0       | 4.8       | 1447.3        |
| 2    | 34.16    | 8.33      | 10.0       | 5.3       | 1253.3        |
| 3    | 31.36    | -4.32     | 29.0       | 5.3       | 923.7         |
| 4    | 36.41    | 9.33      | 10.0       | 4.5       | 1220.4        |
| 5    | 35.04    | 12.37     | 10.0       | 5.0       | 1532.6        |
| 6    | 37.43    | 20.62     | 26.0       | 5.3       | 2146.7        |
| 7    | 40.67    | 23.47     | 28.0       | 5.1       | 2329.7        |
| 8    | 40.03    | 21.66     | 14.0       | 5.2       | 2181.7        |
| 9    | 35.33    | 8.25      | 10.0       | 4.7       | 1181.3        |
| 10   | 34.48    | 14.48     | 10.0       | 5.3       | 1734.7        |
| 11   | 31.30    | -4.43     | 25.0       | 5.1       | 931.8         |
| 12   | 34.83    | 12.44     | 10.0       | 4.7       | 1548.4        |
| 13   | 39.16    | 15.18     | 291.0      | 5.9       | 1646.3        |
| 14   | 38.39    | 21.99     | 10.0       | 5.0       | 2243.8        |
| 15   | 35.95    | 21.91     | 33.0       | 5.3       | 2307.0        |
| 16   | 45.38    | 21.05     | 10.0       | 5.0       | 2141.0        |
| 17   | 45.34    | 21.04     | 25.0       | 5.4       | 2139.0        |
| 18   | 51.16    | 5.82      | 20.0       | 5.4       | 1484.0        |
| 19   | 39.82    | 24.39     | 7.0        | 4.9       | 2419.0        |
| 20   | 40.98    | 22.34     | 18.0       | 5.8       | 2228.0        |
| 21   | 37.20    | 15.50     | 10.0       | 5.4       | 1710.0        |
| 22   | 38.58    | 14.95     | 246.0      | 5.3       | 1636.0        |
| 23   | 40.18    | 13.80     | 401.0      | 5.4       | 1519.0        |

TABLE 2.

**P arrival time estimates in seconds**

| Case | Algorithm | Analyst | Model  | Analyst Comments          |
|------|-----------|---------|--------|---------------------------|
| 1    | 182.95    | 186.19  | 185.83 |                           |
| 2    | 165.15    | 164.51  | 162.94 |                           |
| 3    | 128.55    | 120.78  | 119.72 | weak first motion         |
| 4    | 152.90    | 160.42  | 158.83 |                           |
| 5    | 200.20    | 197.56  | 197.85 |                           |
| 6    | 270.50    | 265.37  | 273.19 | weak first motion         |
| 7    | 290.70    | 283.10  | 284.33 | very small and emergent P |
| 8    | 280.90    | 268.51  | 279.05 | poor quality signal       |
| 9    | 159.10    | 155.65  | 153.93 |                           |
| 10   | 226.55    | 222.27  | 223.12 |                           |
| 11   | 121.45    | 120.61  | 121.00 |                           |
| 12   | 211.05    | 202.39  | 199.82 | poor quality signal       |
| 13   | 194.85    | 197.30  | 195.36 |                           |
| 14   | 272.30    | 275.49  | 274.68 |                           |
| 15   | 278.45    | 280.32  | 277.87 |                           |
| 16   | 272.10    | 266.36  | 273.84 | weak signal               |
| 17   | 265.55    | 264.40  | 273.82 |                           |
| 18   | 185.40    | 187.43  | 190.58 |                           |
| 19   | 301.25    | 293.68  | 294.74 |                           |
| 20   | 272.20    | 273.39  | 272.72 |                           |
| 21   | 223.40    | 219.91  | 221.56 |                           |
| 22   | 200.40    | 197.73  | 196.79 |                           |
| 23   | 180.05    | 179.75  | 177.81 |                           |

**3.3.2 S Arrival Locator Results**

The S phase arrival is significantly more difficult to locate than the P phase arrival. In fact, analysts found it much more difficult to pick S arrival times versus P arrival times for the cases in the test set. In cases where analyst pick times are not available, the IASPE91 model is used for comparison. Though the S arrival algorithm estimates are not as accu-

rate as the P arrival estimates, they are quite promising because of the increased difficulty in locating S arrival onset. Figure 11 shows a couple of illustrations of how  $C_t$  is used to locate the secondary arrival time. Table 3 shows a compilation of secondary arrival locator results. Figure 14 is an error plot based on the data in Table 3. Since the STA/LTA method uses long-period data to locate secondary arrivals, a comparison of STA/LTA with our algorithm is not possible with regards to secondary arrival times. The data of interest in this study is the short-period data, and the algorithm presented strictly uses this type of data to locate arrival times.

As stated in Section 3.2, several wavelets are independently used in the wavelet decomposition of part 2, and the resulting  $C_t$  with the greatest dynamic range is chosen to locate the secondary arrival. For 9 (out of 23) of the events the Vetterli-22 wavelet was the best choice, but for the other events one of the other wavelets shown in Figure 4 resulted in a more accurate S arrival time. This is most likely due to the arrival shape in many cases being similar to the first example in Figure 5, but in the other cases the arrival shape is different and may look more like the third example shown in Figure 5.

From Figure 14 it can be seen that there are four cases (6, 13, 17, 23) for which the errors were significantly large (greater than 15 seconds). Cases 6 and 17 have very poorly recorded S arrivals, in fact the analyst could not read an S arrival for case 6 (for this case the IASPE91 model S onset times is used for comparison). Cases 13 and 23 are very deep events (refer to Table 1). For both these events the strongest arrival in the transverse component is a secondary S arrival which is due to a refraction from the upper mantle bound-



ary at 660 km depth and which is approximately 15 seconds after the much weaker primary S arrival. The analyst has stated that this is an uncommon occurrence, and thus these events are aberrations. Therefore, overall, the results are quite favorable, especially when the difficulty in locating S phase arrivals in short period seismograms is taken into account.

**TABLE 3. S arrival time estimates in seconds**

| Case | Algorithm | Analyst | Model  | Analyst Comments              |
|------|-----------|---------|--------|-------------------------------|
| 1    | 329.10    | 324.86  | 331.82 |                               |
| 2    | 302.15    | 302.91  | 290.66 |                               |
| 3    | 221.70    | 220.89  | 213.62 |                               |
| 4    | 293.25    | No pick | 283.31 | Sn time poor                  |
| 5    | 346.85    | 346.33  | 353.15 |                               |
| 6    | 470.75    | No pick | 488.19 | S arrival could not be read   |
| 7    | 513.10    | 509.06  | 513.62 |                               |
| 8    | 479.15    | 486.03  | 498.47 |                               |
| 9    | 281.95    | 272.22  | 274.54 |                               |
| 10   | 390.10    | 387.65  | 398.37 |                               |
| 11   | 223.40    | 221.23  | 215.96 |                               |
| 12   | 361.50    | 353.10  | 356.67 |                               |
| 13   | 380.50    | 357.83  | 373.84 | weak S, strong "triplicate" S |
| 14   | 498.95    | 490.14  | 502.16 |                               |
| 15   | 523.95    | 518.61  | 508.53 |                               |
| 16   | 488.25    | 481.94  | 489.17 |                               |
| 17   | 454.45    | 484.31  | 489.11 | S poorly recorded             |
| 18   | 326.90    | 328.96  | 340.30 |                               |
| 19   | 540.55    | 525.87  | 540.08 |                               |
| 20   | 485.00    | 479.44  | 498.63 |                               |
| 21   | 391.30    | 382.31  | 404.21 |                               |
| 22   | 352.20    | 356.02  | 359.80 |                               |
| 23   | 356.35    | 325.00  | 339.55 | weak S, strong "triplicate" S |

### 3.4 Discussion and Conclusions

There are two important ideas that have been presented in this chapter. The first is the idea that by analyzing information over several resolutions or scales important features in the seismic signal can be identified. This is due to the concept that strong features in a signal will be maintained over several scales, while weaker features will be present in only a few scales or just one scale. In the algorithm presented this type of inter-scale analysis of polarization information and transverse/radial amplitude ratio information has been shown to be an effective means of identifying P and S phase arrivals. The second important idea is that one can match the wavelet to the arrival shape to help locate the arrival in the seismic signal. This proved to be especially important in the problem of locating the S phase arrival, and is confirmed by the effectiveness of using a pseudo-wavelet constructed from the P arrival shape to locate the S arrival. The overall result of using these ideas is the development of a wavelet transform based method that automatically locates primary and secondary arrivals by analyzing just single station three component short-period seismograms. The key contribution is the introduction of an additional tool for the analysis of seismic signals. As the S arrival is more difficult to locate, methods that help locate this phase are of more interest to the seismological community. Further work will concentrate on further developing and refining the methods used to locate the S arrival, with particular emphasis placed on the idea of designing wavelets specially suited to the seismic signal being analyzed (refer to Chapter 5).

### Appendix 3.A Example of Conservation of Polarization Through Scales

Consider the following two signals that make up a wave:

$$x(t) = \exp(j(\omega t + \varphi_1)) \quad (9)$$

$$y(t) = A \exp(j(\omega t + \varphi_2)) \quad (10)$$

In this case there are two phase terms. For  $\varphi_1 = \varphi_2$  the two components above represent a linearly polarized wave, and for  $0 < \varphi_2 - \varphi_1 \leq \pi/2$  they represent an elliptically polarized wave where equality to  $\pi/2$  will lead to a circularly polarized wave. To find the polarization of this example two-component signal, first determine the covariance matrix and then find the eigenvalues and eigenvectors of the matrix. The eigenvector associated with the largest eigenvalue gives the direction of polarization of the wave. The covariance matrix for a two-component signal has the following form:

$$\begin{bmatrix} \text{Var}(x) & \text{Cov}(x, y) \\ \text{Cov}(y, x) & \text{Var}(y) \end{bmatrix} = \mathbf{M} \quad (11)$$

where  $\text{Var}(x) = \text{Cov}(x, x)$  and  $\text{Var}(y)$  are the variances of  $x(t)$  and  $y(t)$  respectively, and

$$\text{Cov}(x, y) = \lim_{T \rightarrow \infty} \frac{1}{T} \int_{-T/2}^{T/2} x(t) y^*(t) dt \quad (12)$$

is the covariance of  $x(t)$  and  $y(t)$ . With the signals in this example this matrix becomes:

$$\mathbf{M} = \begin{bmatrix} 1 & A \exp(j(\varphi_1 - \varphi_2)) \\ A \exp(-j(\varphi_1 - \varphi_2)) & A^2 \end{bmatrix} \quad (13)$$

The eigenvalues are determined by using the following relationship:

$$\det(\mathbf{M} - \lambda \mathbf{I}) = 0 \quad (14)$$

Using the above relation it is found that:

$$\lambda(\lambda - (A^2 + 1)) = 0 \quad (15)$$

Therefore  $\lambda_1 = 0$  and  $\lambda_2 = A^2 + 1$ .

To find the eigenvectors solve for  $\mathbf{u}$  in the following equation:

$$(\mathbf{M} - \lambda \mathbf{I}) \mathbf{u} = 0 \quad (16)$$

Thus, for  $\lambda_1$  we have:

$$\begin{bmatrix} 1 & A \exp(j(\varphi_1 - \varphi_2)) \\ A \exp(-j(\varphi_1 - \varphi_2)) & A^2 \end{bmatrix} \begin{bmatrix} u(1) \\ u(2) \end{bmatrix} = 0 \quad (17)$$

and for  $\lambda_2$  we have:

$$\begin{bmatrix} -A^2 & A \exp(j(\varphi_1 - \varphi_2)) \\ A \exp(-j(\varphi_1 - \varphi_2)) & -1 \end{bmatrix} \begin{bmatrix} u(1) \\ u(2) \end{bmatrix} = 0 \quad (18)$$

The resulting eigenvectors for  $\lambda_1$  and  $\lambda_2$  respectively are:

$$\begin{bmatrix} -\alpha A \exp(j(\varphi_1 - \varphi_2)) \\ \alpha \end{bmatrix} \text{ and } \begin{bmatrix} \alpha \\ \alpha A \exp(-j(\varphi_1 - \varphi_2)) \end{bmatrix} \quad (19)$$

We now determine what happens to the polarization when the components are processed by the wavelet transform. Let us determine the wavelet coefficients of the original components (9) and (10). Plugging  $x(t)$  and  $y(t)$  respectively into the definition of the CWT (Chapter 2) we get:

$$W_x(a, b) = \frac{1}{\sqrt{a}} \int_{-\infty}^{\infty} e^{j\omega_0 t} g\left(\frac{t-b}{a}\right) dt \quad (20)$$

$$W_y(a, b) = \frac{A e^{j\varphi}}{\sqrt{a}} \int_{-\infty}^{\infty} e^{j\omega_0 t} g\left(\frac{t-b}{a}\right) dt \quad (21)$$

Now let us define the Fourier transform of a periodic signal as:

$$F(\omega_0) = \int_{-\infty}^{\infty} f(t) e^{j\omega_0 t} dt \quad (22)$$

Using (22) and Fourier transform properties, (20) and (21) can be written as:

$$W_x(b) = A_1 \exp(j(\omega a b + \varphi_1)) \quad (23)$$

$$W_y(b) = A_2 \exp(j(\omega a b + \varphi_2)) \quad (24)$$

where  $A_1 = \sqrt{a} G(a\omega_0)$  and  $A_2 = A A_1$ , and  $a$  redefined as  $1/a$ . Equations (23) and (24) have been written as functions of just  $b$ , the translation parameter, because we will be evaluating the covariance matrix for each scale,  $a$ , in terms of  $b$ . Now we proceed with finding the covariance matrix for  $W_x(b)$  and  $W_y(b)$  and then determining the eigenvalues and eigenvectors. The covariance matrix is:

$$\mathbf{M}_b = A_1 A_1^* \begin{bmatrix} 1 & A \exp(j(\varphi_1 - \varphi_2)) \\ A \exp(-j(\varphi_1 - \varphi_2)) & A^2 \end{bmatrix} \quad (25)$$

Using (14) it is found that:

$$\lambda \left( \lambda - (A_1 A_1^* A^2 + A_1 A_1^*) \right) = 0 \quad (26)$$

Therefore  $\lambda_1 = 0$  and  $\lambda_2 = A_1 A_1^* A^2 + A_1 A_1^*$ .

Using (16) to find the eigenvectors we have for  $\lambda_1$  and  $\lambda_2$  respectively:

$$A_1 A_1^* \begin{bmatrix} 1 & A \exp(j(\varphi_1 - \varphi_2)) \\ A \exp(-j(\varphi_1 - \varphi_2)) & A^2 \end{bmatrix} \begin{bmatrix} v(1) \\ v(2) \end{bmatrix} = 0 \quad (27)$$

$$A_1 A_1^* \begin{bmatrix} -A^2 & A \exp(j(\varphi_1 - \varphi_2)) \\ A \exp(-j(\varphi_1 - \varphi_2)) & -1 \end{bmatrix} \begin{bmatrix} v(1) \\ v(2) \end{bmatrix} = 0 \quad (28)$$

Since the term in front of the matrix is going to divide out, it can be seen that the eigenvectors that result are the same as those in (19).

Thus, from the example above it can be seen that the polarization of a wave is conserved as it is processed by the wavelet transform. To visually illustrate this, the real parts of equations (9) and (10) were used to synthetically produce a polarized wave (with  $A = 2$  and with  $\varphi_2 - \varphi_1 = \pi/4$ ). The resulting wave was processed using the wavelet transform. The resulting  $x$  versus  $y$  plots for each scale (scale 0 refers to the original

wave) are shown in Figure 15. The shape and orientation of the ellipses are the same, so polarization is conserved.

### Appendix 3.B Determination of Window for Polarization Calculation

As stated in Section 3.2, a covariance matrix  $\mathbf{M}^j[i]$  is calculated at each point  $i$  of  $\mathbf{d}_e^j$ ,  $\mathbf{d}_n^j$ , and  $\mathbf{d}_z^j$  over a  $T$  point window. The size of the window affects the ultimate output,  $\mathbf{C}_F$ . The goal is to obtain a  $\mathbf{C}_F$  that has only one prominent spike with a minimum of secondary spikes. This can be seen as trying to minimize the entropy of  $\mathbf{C}_F$ . There is a measure called the varimax norm (Wiggins, 1978) which can be used for this purpose. The higher the varimax norm of a signal, the fewer spikes the signal has. The varimax norm of  $\mathbf{C}_F$  is calculated as:

$$V_c = \frac{\left( \sum_i C_F^4[i] \right)}{\left( \sum_i C_F^2[i] \right)^2} \quad (29)$$

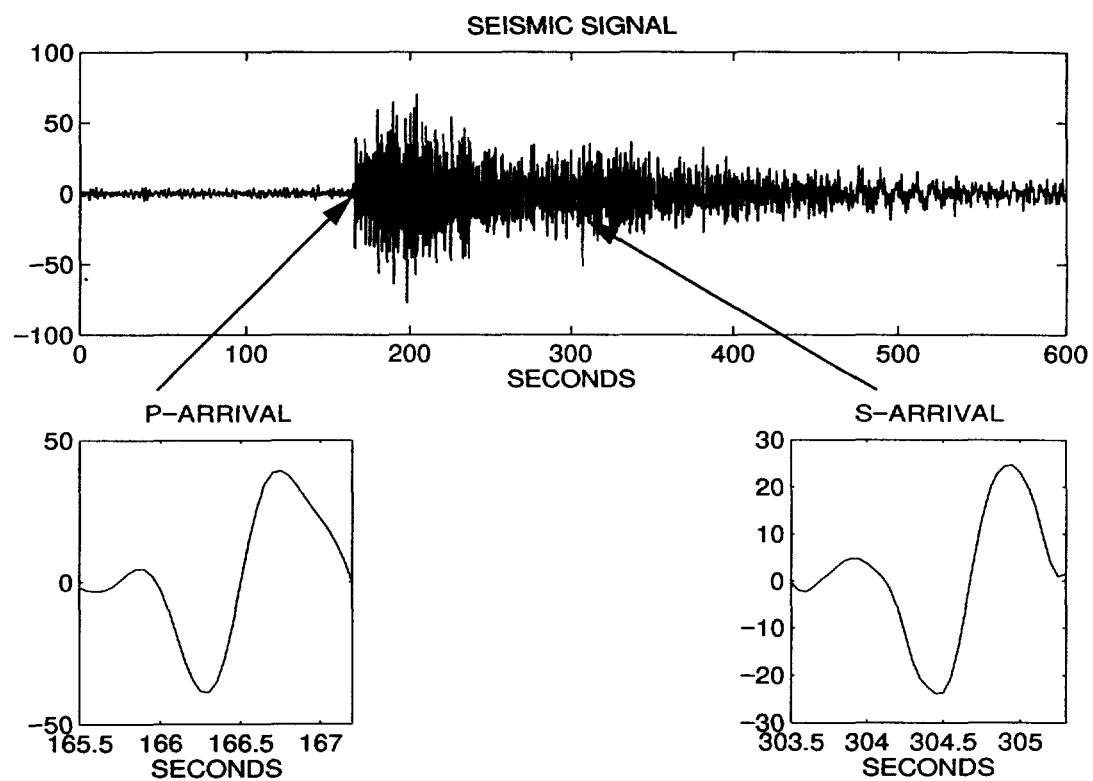
Therefore, before choosing the size of the  $T$  point window, several size windows are tested in order to maximize  $V_c$ . The window resulting in a  $\mathbf{C}_F$  with the highest varimax norm is chosen as the window for the particular event. Windows ranging from 2.5 seconds to 25 seconds are used with a step of 2.5 seconds. Figure 16 is a plot of the varimax norm of  $\mathbf{C}_F$  versus the window size used to construct the covariance matrix. For this particular event 12.5 seconds is chosen as the window length.

### **Appendix 3.C Use of the Wavelet Transform as a Glitch Detector**

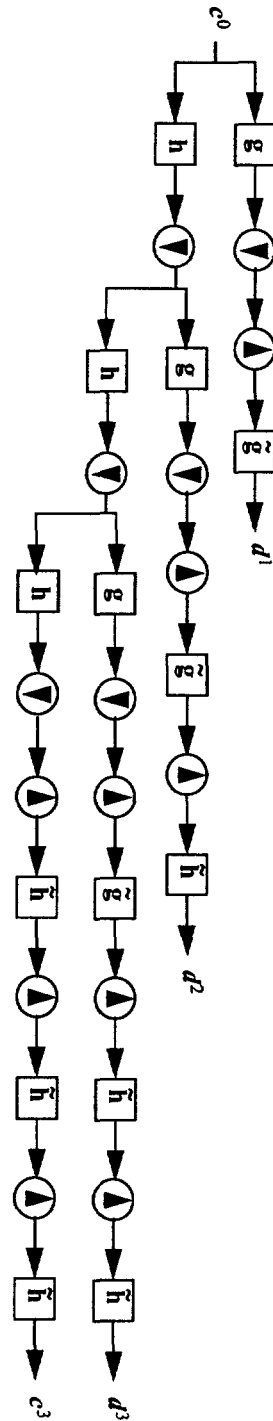
In going through the data it has been found that the wavelet transform can be used as a glitch detector to eliminate bad data. Coincidentally, several researchers have expressed an interest in a method to easily determine the location of glitches in seismic signals. It turns out that glitches (unnatural discontinuities) in the original components show up as very prominent spikes in the first one or two wavelet transform scales. This makes sense, since such very high frequency behavior will manifest itself strongly in the lowest scales (which represent high frequency information). The position of these spikes is the exact position of the glitches in the components. This was very useful in eliminating bad data. Figure 17 shows an example of how glitches show up in the low scale wavelet transform coefficients. Cases with glitches were not used in the test set (an area of future research could be to reconstruct the original data by modifying the wavelet coefficients at the spike locations).



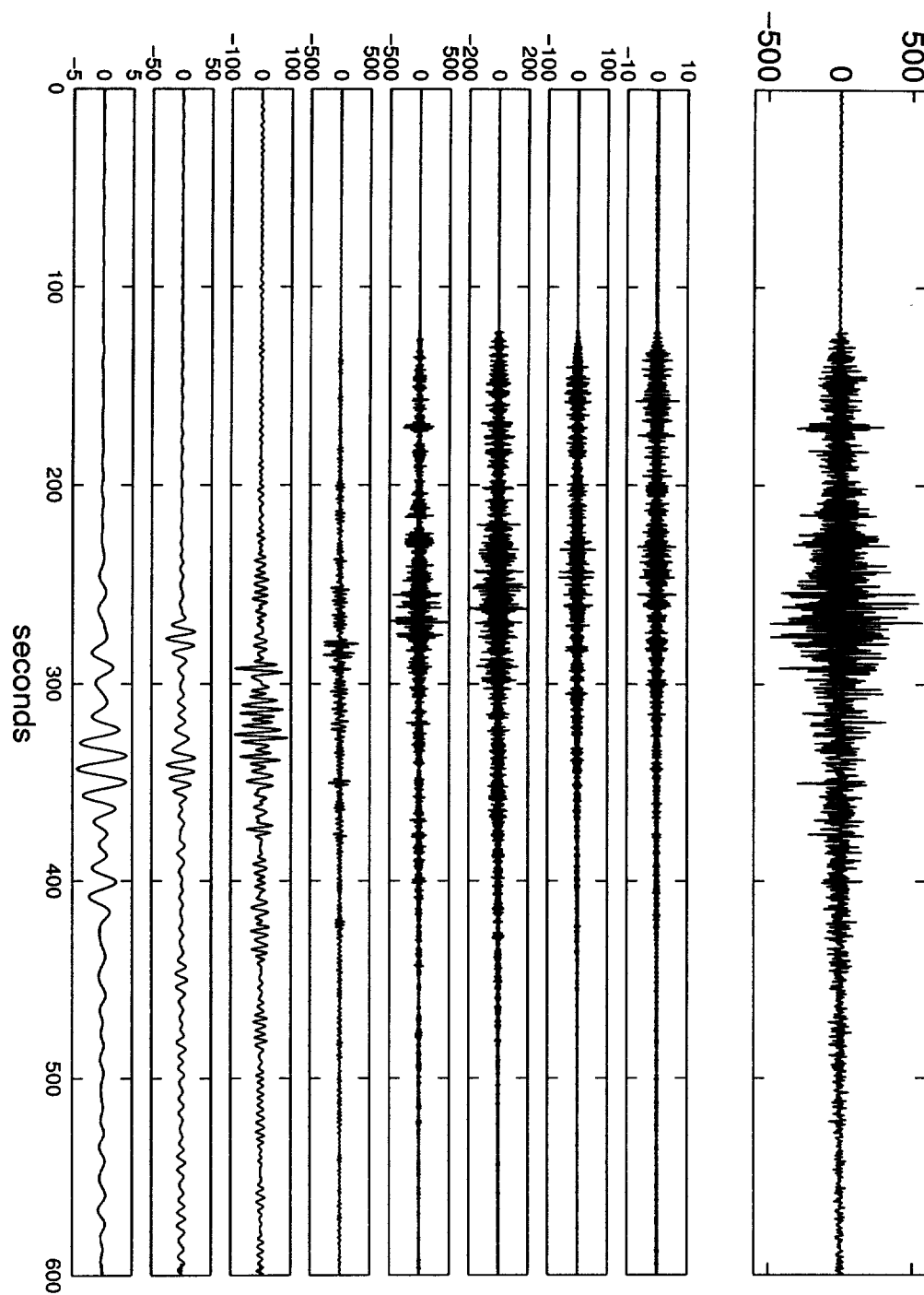
**FIGURE 1. One component of a seismogram showing P and S arrival locations**



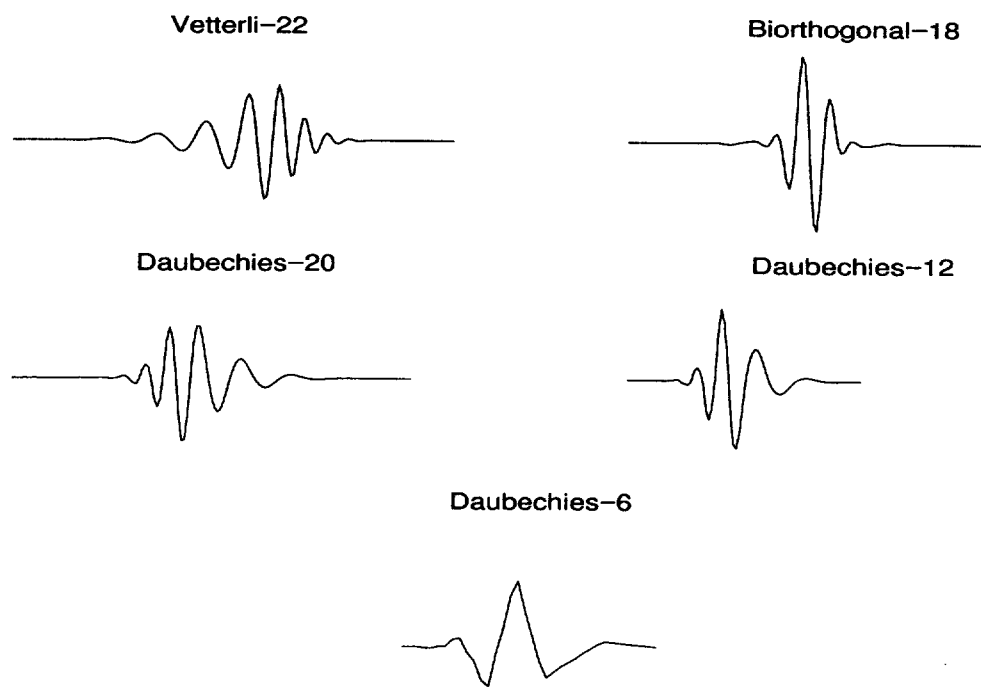
**FIGURE 2. Multiresolution tree used to obtain wavelet coefficients with equal number of samples (i.e.  $\text{length}(d^{j+1}) = \text{length}(d^j)$ ). Downward arrows indicate downsampling by two and upward arrows indicate upsampling by two. The outputs ( $d^j$ ) are the interpolated coefficients.**



**FIGURE 3. One component of a three component short-period seismogram and the interpolated wavelet coefficients for 8 scales.**



**FIGURE 4. Wavelets used in this study**



**FIGURE 5. P phase arrivals cut from three different vertical components**

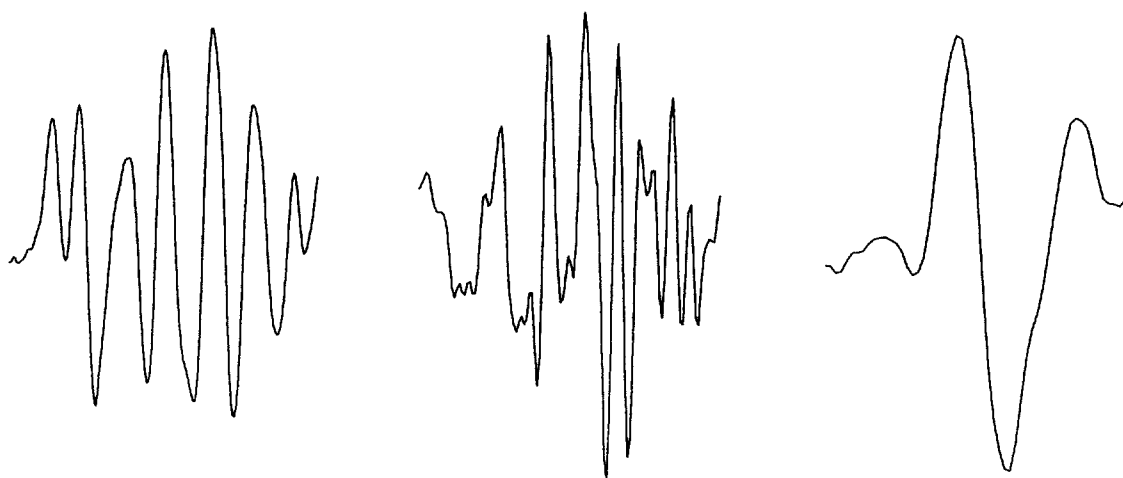


FIGURE 6. P and S wave motions

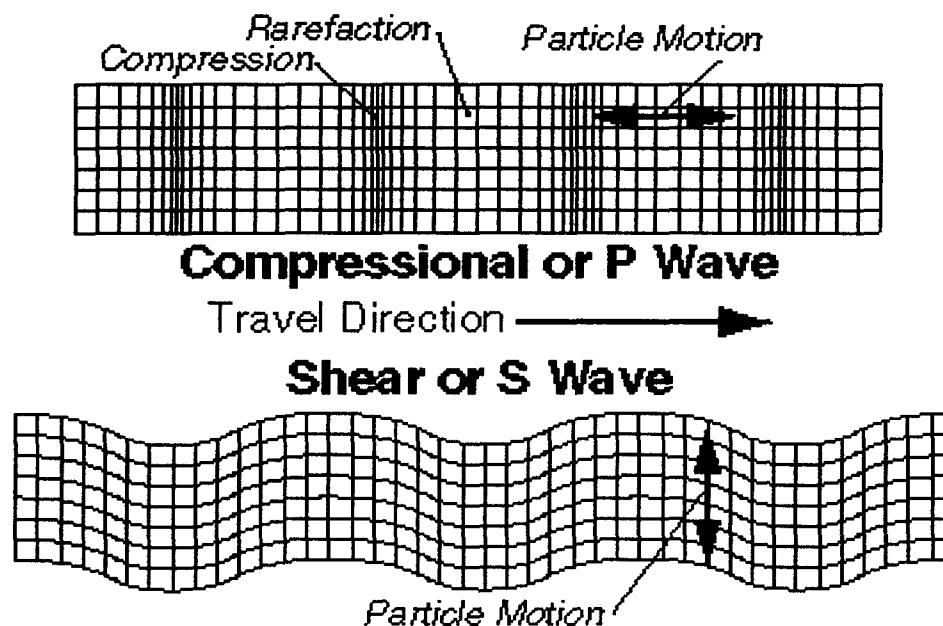
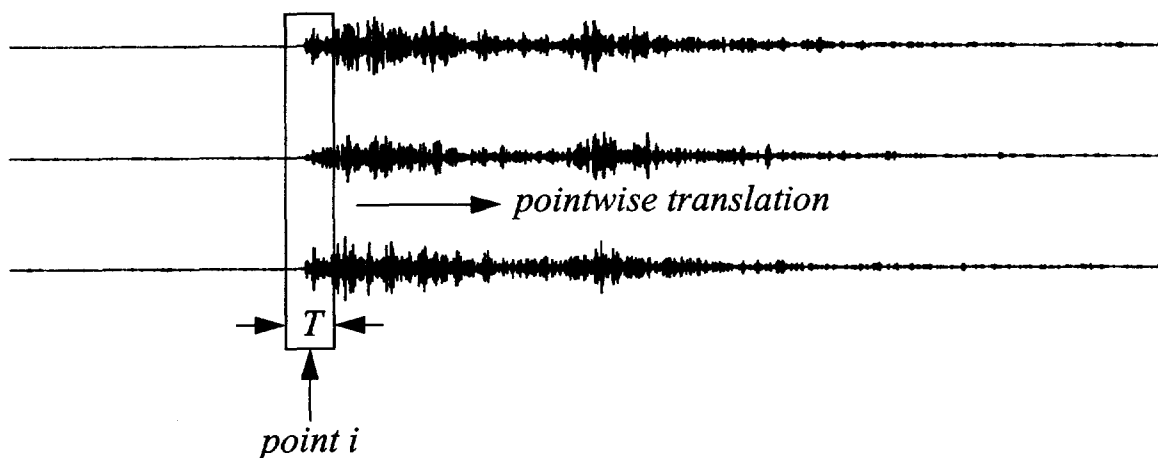


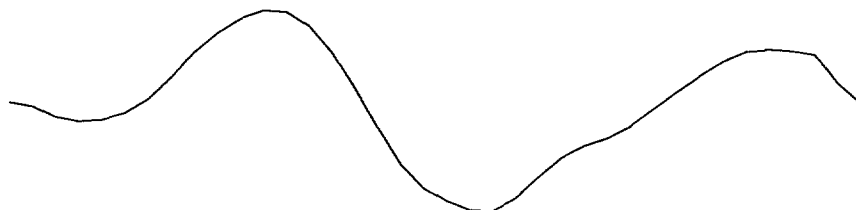
FIGURE 7. Illustration of windowing used to calculate covariance matrices  $M^j[i]$ . Each matrix is calculated over a  $T$  point window. The point  $i$  is the midpoint of the window. The window is translated pointwise.



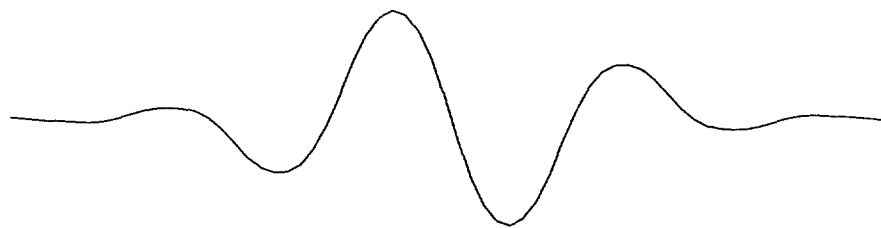
**FIGURE 8. Pseudo-wavelet function obtained by cutting P phase arrival from a vertical short-period component, removing the mean, and tapering to zero at the ends. Also, the biorthogonal-18 wavelet function.**

---

Portion of P arrival used as pseudo-wavelet function



Biorthogonal-18 wavelet function



**FIGURE 9.** Three component short-period seismogram, secondary arrival locator function,  $C_t$ , resulting from processing with pseudo-wavelet shown in previous figure, and  $C_t$  resulting from processing with biorthogonal-18 wavelet.

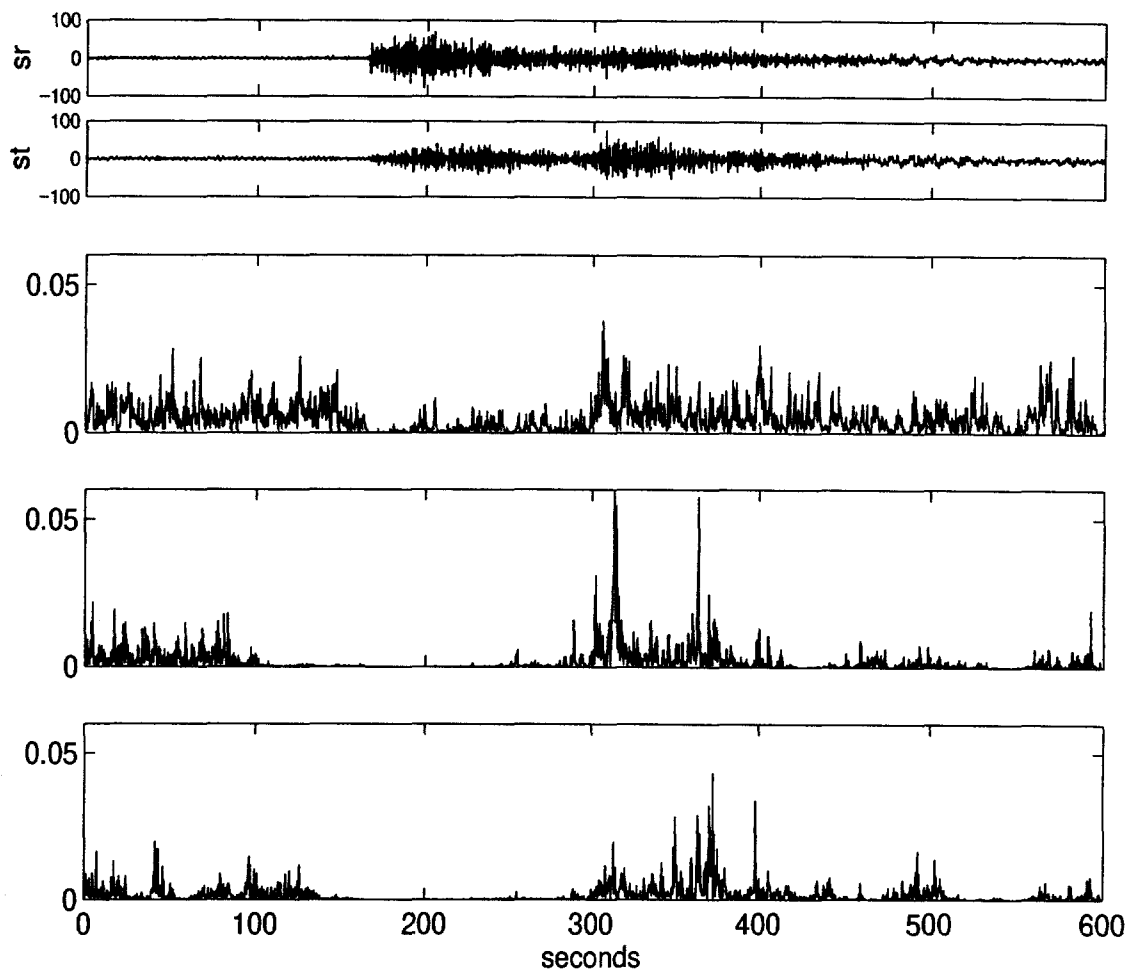


FIGURE 10.

Three-component short-period seismogram (east, north, and vertical components) and the function  $C_F$ . The peak of this function is used to locate the P phase arrival.

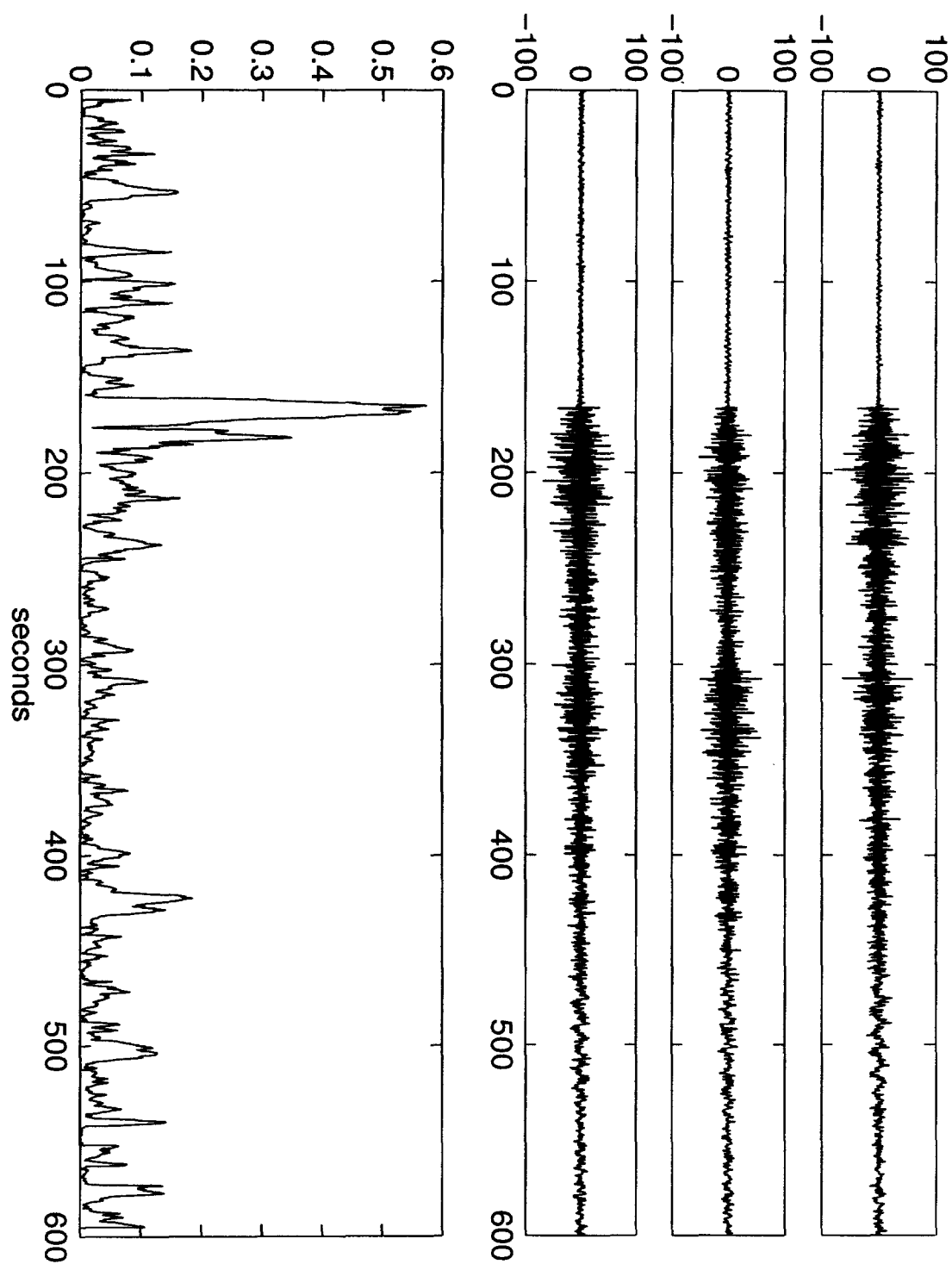
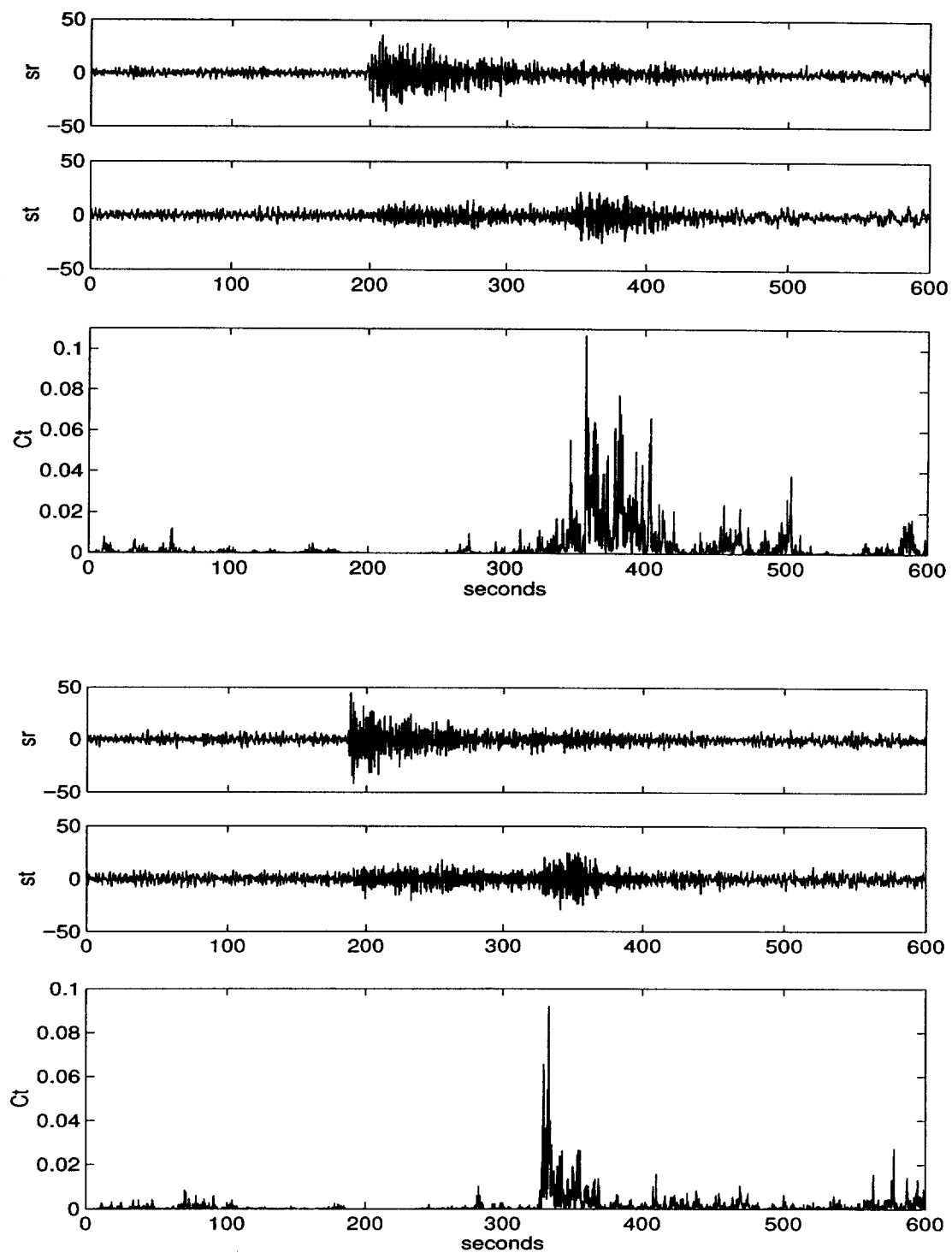




FIGURE 11.

Two examples of three component short-period seismograms and the secondary arrival locator,  $C_t$ . This locator function is used to locate the secondary phase arrival.



**FIGURE 12. Event and recording station locations (station is marked by star in Spain). Some of the event locations overlap.**

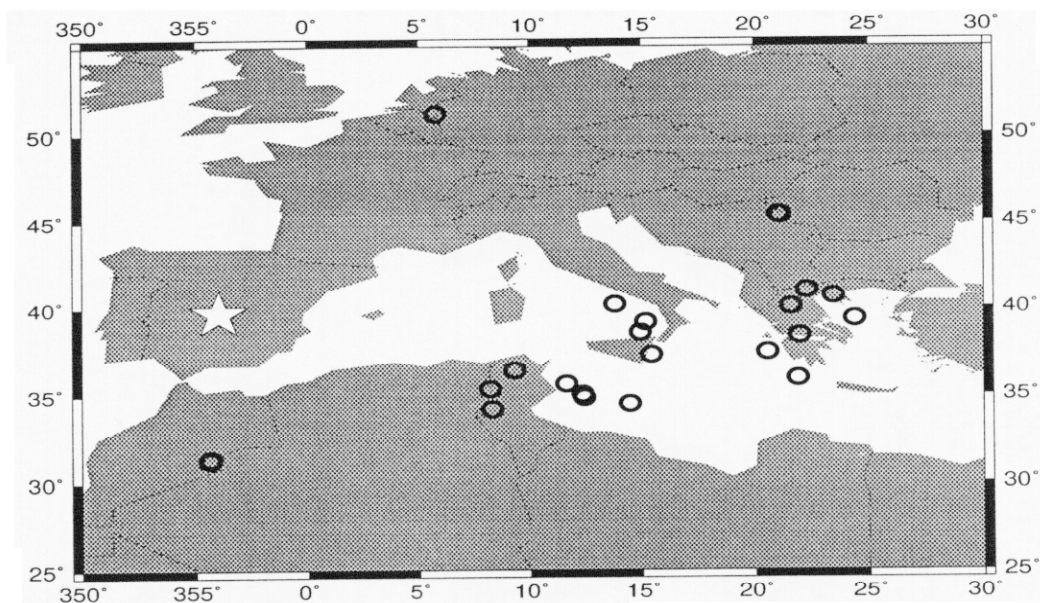


FIGURE 13.

**P phase arrival residual plot. The circles represent the error between algorithm pick time and the analyst pick time. The asterisks represent the error between the STA/LTA pick and the analyst pick.**

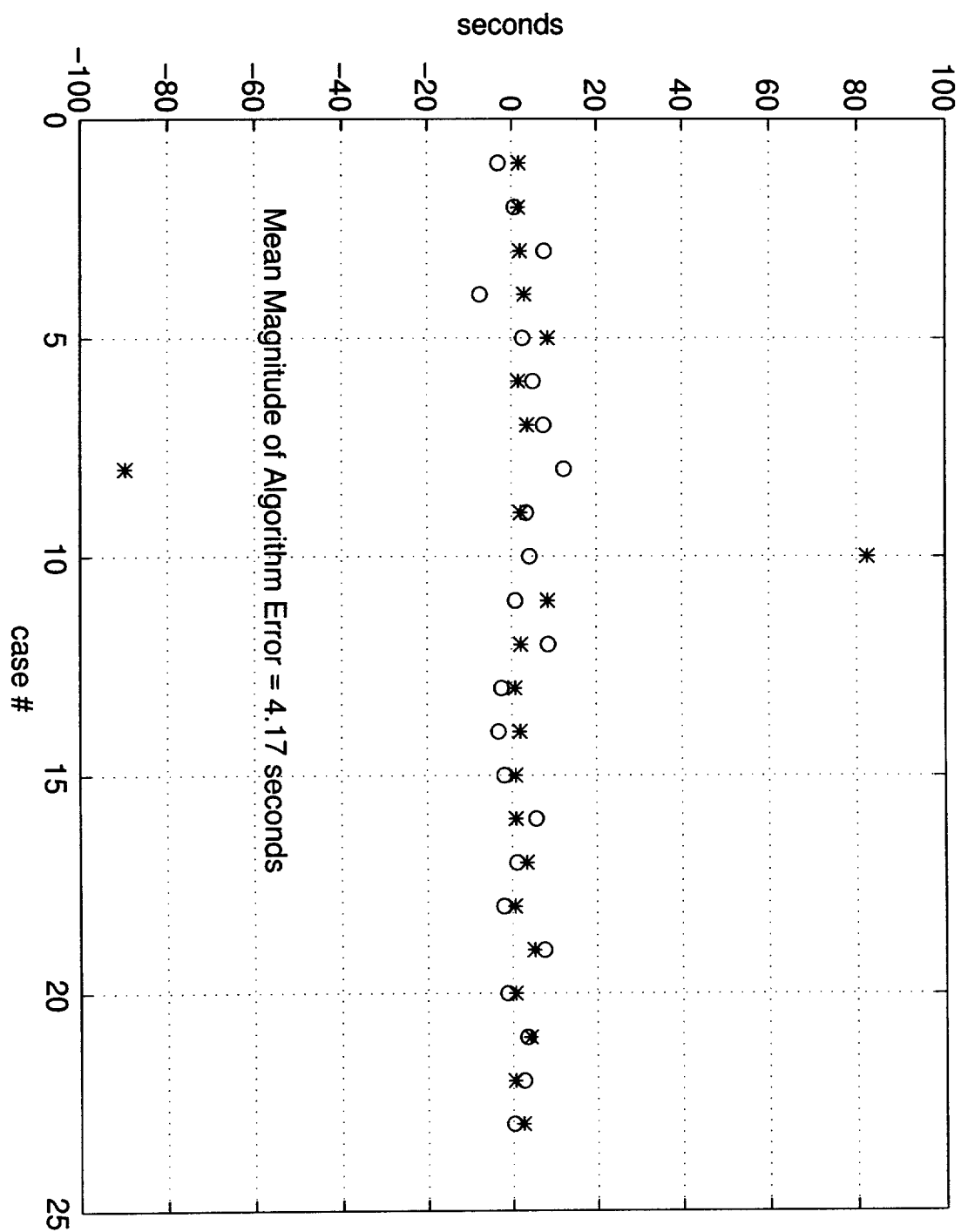


FIGURE 14. S phase arrival residual plot. The circles represent the error between the algorithm pick time and the analyst/model pick time.

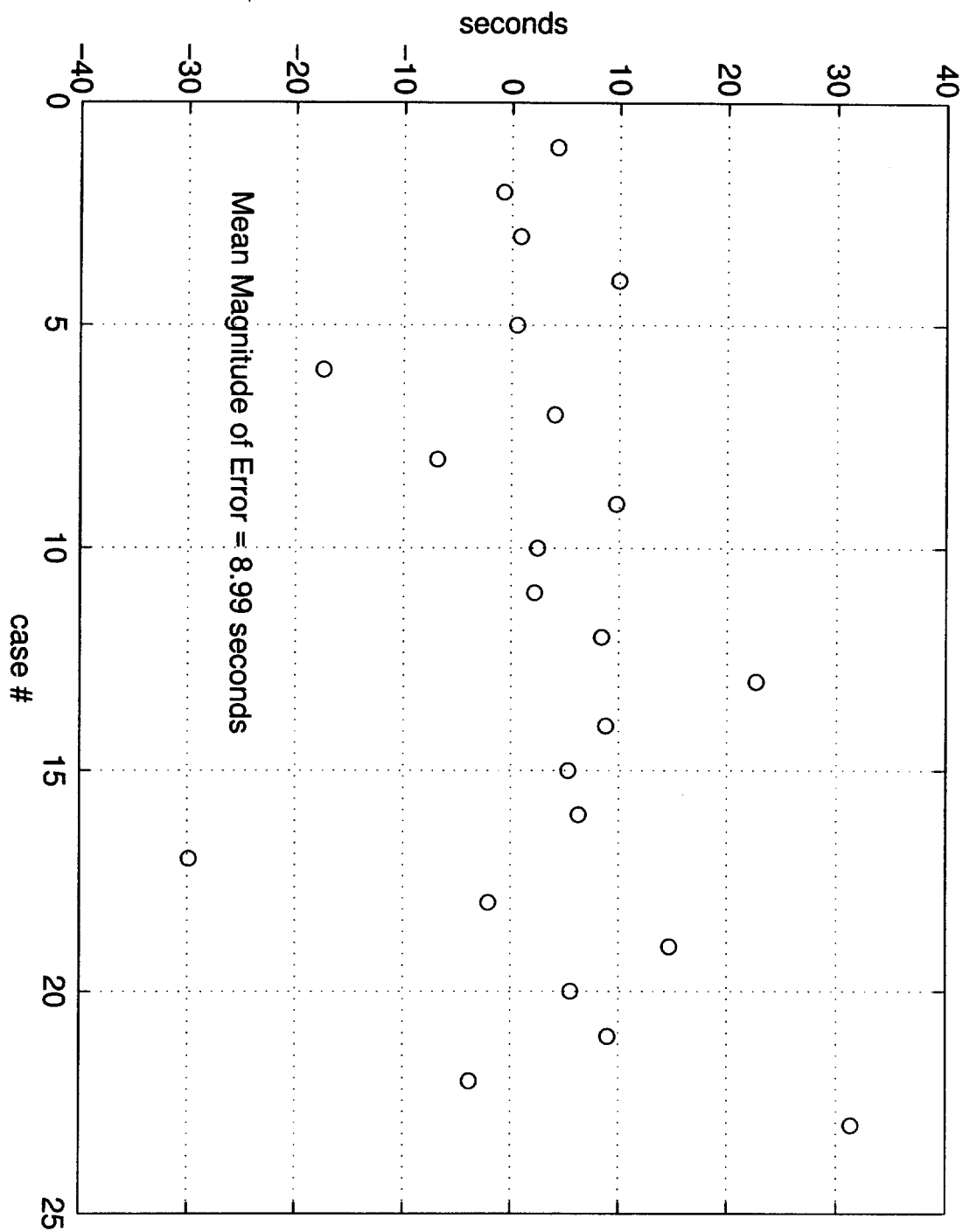


FIGURE 15.

Polarization curves for synthesized signal and five scales

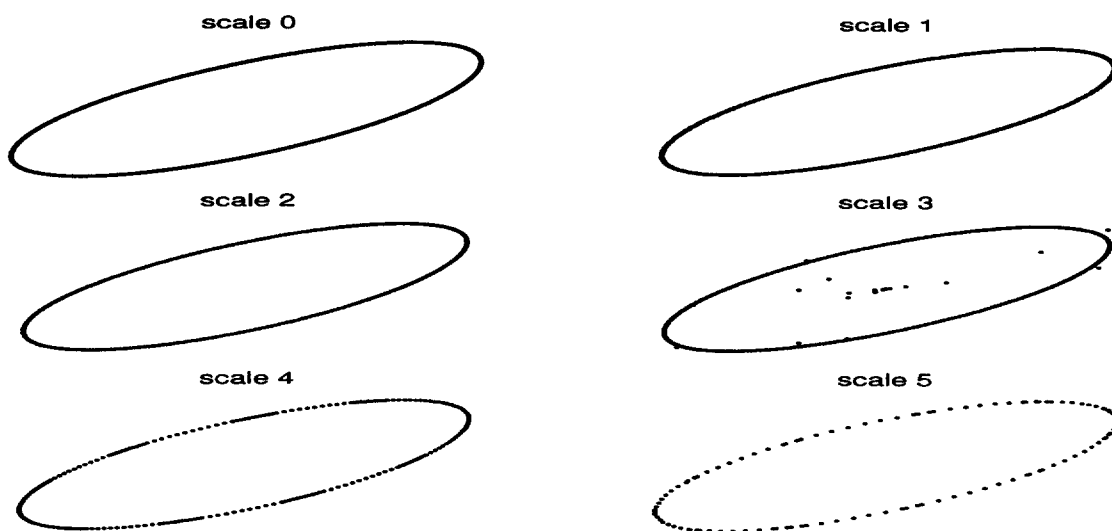
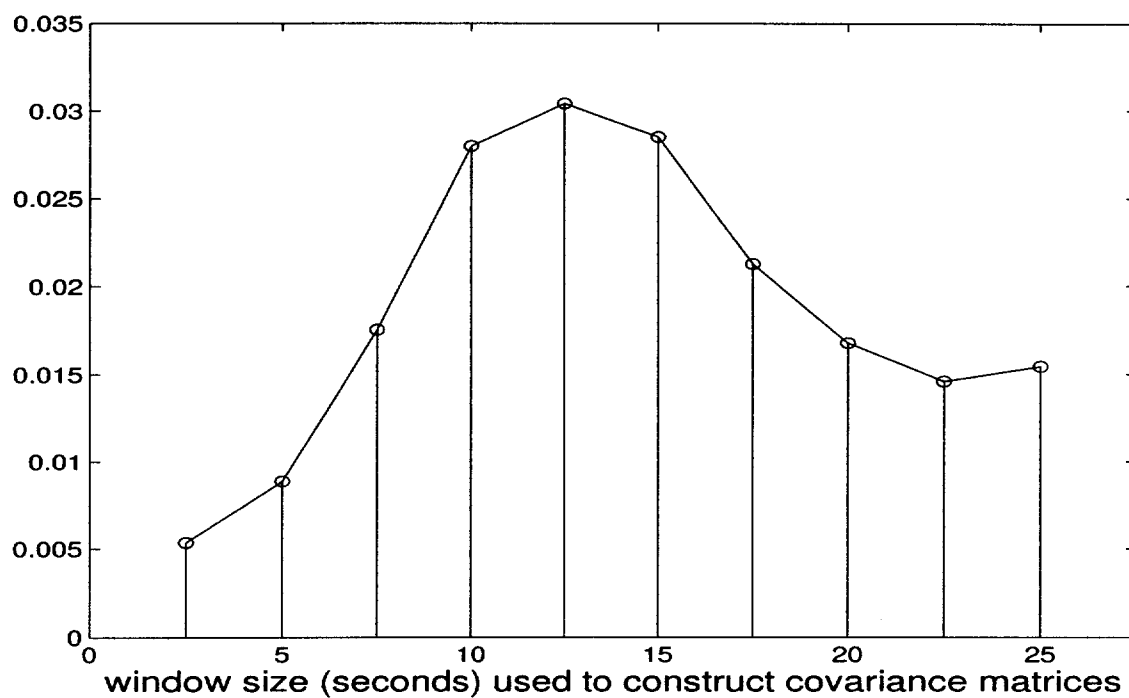
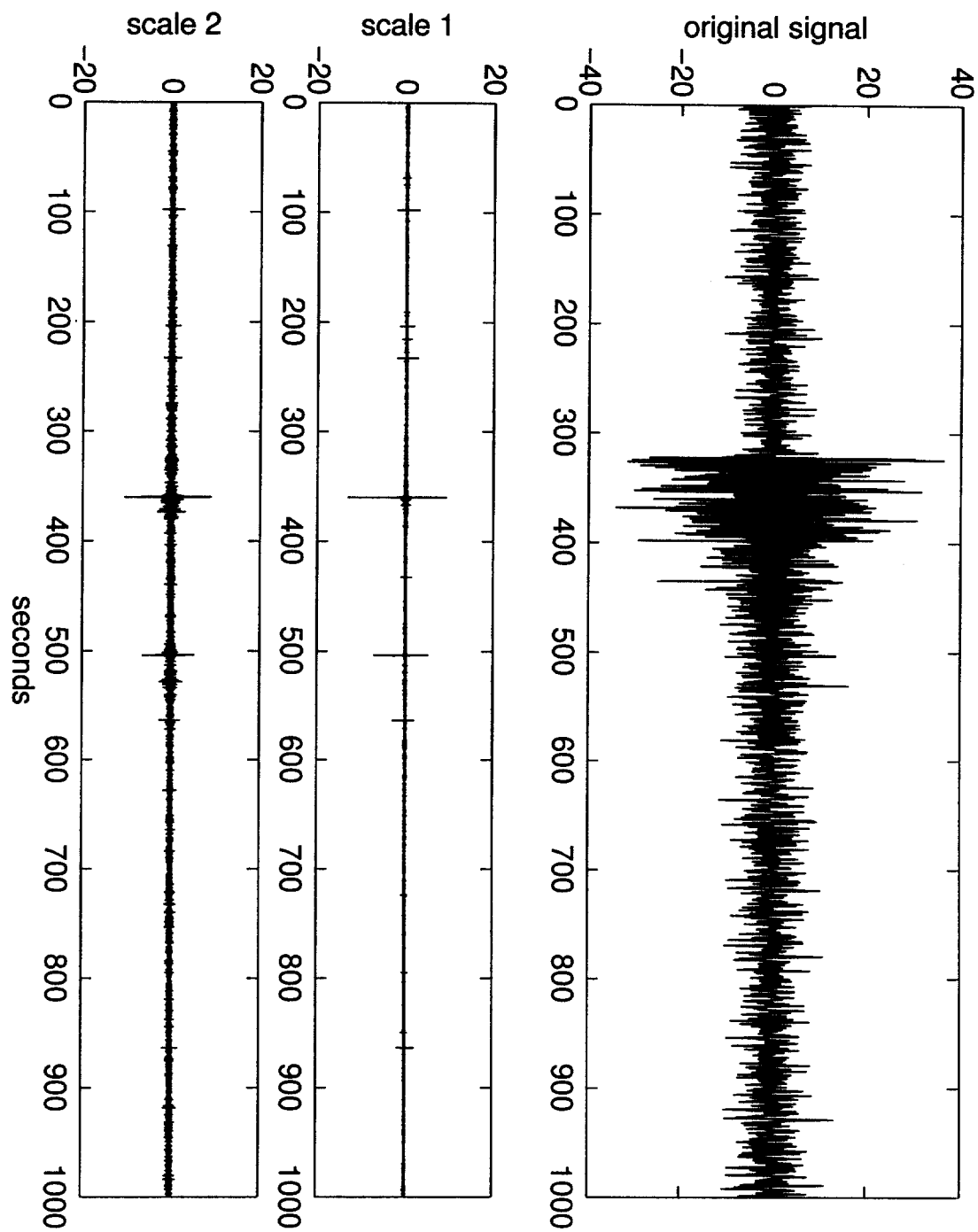


FIGURE 16. Plot demonstrating the effect of window size used to calculate the covariance matrices,  $M^j[i]$  on the varimax norm of the composite rectilinearity function,  $C_F$ . For this particular event a window length of 12.5 seconds is chosen.



**FIGURE 17.** One component of a three component short-period seismogram and the wavelet coefficients for the first two scales. The sharp peaks mark the exact locations of glitches in the original signal.



**CHAPTER 4***Compression of the  
ECG using the Wavelet  
Transform and Vector  
Quantization***4.0 Summary**

The previous chapter described a method of using the wavelet transform to detect and identify phase arrivals in the seismogram. That was a feature identification application of the wavelet transform. This chapter shows how the wavelet transform can be used for efficient signal *representation*, better known as signal *compression*. The signal of interest in this case is the electrocardiogram (ECG). The ECG represents the electrical activity of the heart and will be described in some detail in Section 4.3. The focus of this chapter will be the presentation of an improved wavelet compression algorithm for ECG signals which I

developed by using *vector quantization* on wavelet coefficients. The results indicate that the proposed method excels over standard techniques for high fidelity compression, with a low bit-per-sample rate and the retention of important ECG features.

The organization of this chapter is as follows. The first two sections (4.1 and 4.2) discuss transform based compression methods in general. Sections 4.3-4.5 introduce the ECG and ECG compression techniques. Section 4.6 describes the ECG compression method that I have developed which shows the advantage of using the wavelet transform (in combination with vector quantization) to code the ECG. One of the important results (Section 4.7) is the dependence of coding gain on the wavelet used. This is discussed in the last section of the chapter.

#### 4.1. Compression Methods using Linear Transforms

As mentioned in Chapter 2, the type of compression method that is of concern in this thesis is lossy. In other words, only information that is defined to be significant is retained, while other information is not, for the purpose of maximum compression. The information that is not kept is unrecoverable. The basic procedure used in transformation based compression is shown in the block diagram of Figure 1. A signal  $x$  is operated on by a linear transform,  $T$ . The output,  $y = Tx$ , is processed using methods such as thresholding and quantization. After this processing, the reconstructed signal,  $\hat{x}$ , is obtained by using the inverse transform,  $T^{-1}$ . It is standard procedure to add a lossless compression step such as entropy encoding before the inversion step (Vetterli and Kovacevic, 1995; Welch, 1984).



Let us first discuss the techniques implemented in the processing of the transform coefficients (the second box in Figure 1).

*Thresholding:* There are many thresholding techniques, with the simplest being hard thresholding:

$$y[n] = \begin{cases} x[n], & |x[n]| \geq T \\ 0, & |x[n]| < T \end{cases} \quad (1)$$

$$n = 0, 1, 2, \dots, N-1$$

where  $y$  is the thresholded output and  $T$  is some positive value based on the statistics of  $x$ . For example,  $T$  can be chosen to be some fraction of the maximum value of  $x$ . A more sophisticated thresholding method that has proven effective, particularly in conjunction with the wavelet transform is called *soft thresholding*:

$$y[n] = \text{sgn}(x[n]) (|x[n]| - T)_+ \quad (2)$$

where the plus sign here means that the argument in brackets is non-zero only when it is positive. The effect of the relation above is to set to zero data that is below  $T$  in absolute value, while data that is greater than  $T$  in absolute value is shifted towards the origin by an amount  $T$ . Soft thresholding of wavelet coefficients using  $T = \sigma \sqrt{2 \log(N^j) / N^j}$  (where  $\sigma$  is a measure of noise level, and  $N^j$  equals the number of points in each scale  $j$ ) has been used for the purposes of noise reduction, where the goal is to recover an unknown signal from noisy data. Using arguments based on statistics, Donoho and Johnstone (Donoho and Johnstone, 1994; Donoho, 1995; Donoho and Johnstone, 1995) have shown that soft thresholding is “near optimal” for this purpose. In addition, their work provides a good overall review of thresholding method selection from a statistical point of view.

*Quantization:* The mapping of real numbers to discrete values is termed quantization. When samples are quantized individually the term used is *scalar quantization*. A common type of scalar quantization is the representation of a real number by the nearest integer. The range and resolution of discrete values used to represent a real number determines the bit-per-sample rate. For example, in the case of a binary representation where a choice of  $N$  discrete values are used to represent any sample  $x[n]$ , the bit-per-sample rate (number of bits needed to represent each sample) will be  $\log_2 N$ .

Scalar quantization can be seen as a subset of *vector quantization* (VQ) (Gersho and Cuperman, 1983; Gray, 1984; Gersho and Gray, 1992). Instead of quantizing each individual sample, in vector quantization a block of samples is represented by a single discrete value. When the size of this block or vector is equal to one we have the situation represented by scalar quantization (vector quantization of order one). Refer to Figure 2 for a simple illustration of the difference between scalar and vector quantization. One drawback of VQ is that the construction of the library of vectors or *codevectors* is a more complicated task. With scalar quantization the set of codevectors (*codebook*) consists simply of a set of discrete values, for example the integers from 0 to  $(2^N - 1)$ . With VQ the codebook has to be constructed using information about the data and generally some type of clustering algorithm. For example, in this work a training set of ECG data and *k-means clustering* is used to construct the codebooks (refer to Section 4.5 for details). The higher the order of the vectors used, the greater the complexity of the codebook. Given a codebook, each block of a signal is quantized by matching it with the codevector (which is represented by a discrete pointer) that is closest based on a distance measure.

Shannon (1959) showed that the limit on data compression performance (minimization of distortion for a given bit-per-sample rate) can be approached arbitrarily closely by quantizing vectors as in VQ instead of quantizing scalars. In particular the main advantage

of VQ is that by quantizing blocks of data, instead of individual samples, fractional bit-per-sample rates can be achieved. For example, if a function  $x$  is represented by  $N$  vectors of length  $k$ , then the bit-per-sample rate is  $(\log_2 N)/k$ . Therefore, at least theoretically, bit-per-sample rates below one (the Holy Grail of scalar quantization) can be achieved. Achieving the minimum bit-per-sample rate while retaining signal features that are deemed important is the goal of any compression scheme. Vector quantization has proven to be successful in the areas of speech, image, and video compression (Gersho and Cuperman, 1983; Lookabaugh et al., 1993), and in this work it will be shown to be a promising method, in conjunction with the wavelet transform, for ECG compression.

#### 4.2. The Wavelet Transform and Compression

We have not yet discussed why the compression procedures mentioned are applied in the transform domain instead of directly on the raw signal. One reason is that usually the data as a function of time in its original form has a varying degree of correlation. However, it is possible to make the coefficients in the transform domain uncorrelated. This can be understood in terms of the diagonalization of the autocorrelation matrix (as defined in Appendix 5.B). Defining the autocorrelation matrix of  $x$  to be  $R_x$ , the autocorrelation matrix of  $y = Tx$  is:

$$R_y = TR_x T^t \quad (3)$$

where  $T^t$  is the transpose of the linear transform matrix ( $T$  is unitary so,  $T^t = T^{-1}$ ). For maximum decorrelation  $T$  should be chosen so that  $R_y$  is diagonal. This is accomplished by having the rows of  $T$  equal to the eigenvectors of  $R_x$ . The transform that realizes this optimally is called the Karhunen-Loeve transform (KLT) (Gersho and Gray, 1992). How-

ever, the calculation of the KLT is computationally expensive, requiring  $N^2$  operations (where  $N$  here is the length of  $\mathbf{x}$ ). Approximations to the KLT such as the discrete-time cosine transform (DCT) (Ahmed et al., 1974) can be implemented using fast algorithms, but the DCT is a Fourier based decomposition and the disadvantages of such decompositions when compared to wavelet decompositions have been discussed in Chapter 2 and is discussed further in this section.

The decorrelation property of transforms leads to a more efficient representation of the data and thus improved compression can be obtained. Though there is no formal result that guarantees improved compression, in practice the quantization of decorrelated transform coefficients is more efficient than the quantization of the original data. Another advantage is that often information in the transform domain is easier to deal with in terms of coding operations. An example of this is the representation by scales provided by the wavelet transform. The different scales represent different frequency characteristics of the original data. The scales in which features of interest are not present can be neglected. This type of task would be much more difficult, if not impossible, if one was only operating on the domain of original data.

The advantage of the WT for compression over other transforms is that it can be implemented via fast algorithms, and most importantly it has the power to decorrelate more efficiently than Fourier based transforms, producing a more efficient representation of the original data. To understand why this is so let us again look at the relations for the inverse FT and the inverse WT from Chapter 2. The inverse FT is expressed as:

$$x(t) = \frac{1}{2\pi} \int_{-\infty}^{\infty} X(\omega) \exp(j\omega t) d\omega \quad (4)$$

and the inverse WT is:

$$x(t) = \frac{1}{C} \int_{-\infty}^{\infty} \frac{1}{2} \int_{-\infty}^{\infty} W_x(a, b) \frac{1}{\sqrt{|a|}} g\left(\frac{t-b}{a}\right) db da \quad (5)$$

Looking at (4) it can be seen that  $x(t)$  is essentially expressed as a summation of sinusoids. As stated earlier, for transient signals this is not an efficient representation, since transients will require many weighted sinusoids in order to be described by (4). On the other hand, (5) is a representation of  $x(t)$  based on wavelets. Since wavelets are finite and zero mean, and thus more representative of actual signal features than sinusoids, it can be seen that (5) will lead to a representation where the wavelet transform coefficients will be more “concentrated” than the Fourier coefficients of (4). In other words, significant information will be contained in only a few of the wavelet coefficients. The closer the wavelet is to the actual signal features, the more concentrated the representation. This is often referred to as a sparse representation. This “sparseness” can be maximized by using a wavelet that closely approximates the features of the signal being analysed. For example, in the ideal situation where a signal  $x$  is made up of one feature  $f$  at different scalings, then if the wavelet  $g = f$  the number of nonzero wavelet coefficients  $W_x$  needed to represent  $x$  in the wavelet domain is a minimum.

Several researchers have shown (mathematically and experimentally) that the wavelet transform leads to greater compression than traditional methods (DeVore et al., 1992; Desarte et al., 1992; Wickerhauser, 1992). The conclusions made in these works follow some of the same arguments that have been made in this section, namely that sharp discontinuities or transients, which are well localized in time can be represented more compactly with the WT than with Fourier-based transforms (such as the DCT).

Due to the benefits that result from using the wavelet transform for compression it has found many successful uses in compression applications, particularly speech compression

(Evangelista, 1993), image compression (Yang, 1995), and video compression (Zafar, 1993), and as will be shown below, ECG compression.

#### **4.3. The ECG as a Representation of Heart Function**

In this work the focus is on the compression of the electrocardiogram (ECG), and, therefore, in this section a description of the ECG signal is given as well as what it represents.

The ECG is a recording of the electrical activity of the heart recorded via electrodes placed on the body surface, and thus is a plot of the changing potential differences between the electrodes. The primary features of the ECG are the P-wave, the QRS complex, and the T-wave (refer to Figure 4). Analysis of these features gives information about the functioning of the heart. The heart is composed of four chambers. The upper chambers are known as the left and right atria, while the lower, larger chambers are known as the left and right ventricles (refer to Figure 3). Each of these chambers contracts when an electrical impulse travels across it, due to the depolarization (followed by repolarization) of muscle cells.

The electrical impulse originates in a small group of specialized cells located in the right atrium known as the *sinoatrial (SA) node*. This impulse first travels through the muscle cells of the right and left atria causing these chambers to contract, and force blood into the ventricles. The P-wave on the ECG represents this electrical activation of the atria.

The impulse then reaches the *atrioventricular (AV) node*, where conduction is extremely slow, thereby delaying the impulse to give time for the ventricles to fill with blood. The short period of inactivity between the end of the P-wave and the beginning of the QRS complex on the ECG represents this AV node conduction time. The electrical impulse is then passed to the *bundle of His* (refer to Figure 3), which is a strand of specialized muscle tissue providing the only normal conduction pathway between the atria and ventricles. The impulse moves down the bundle of His and then along the strands of the right and left bundle branches. At the end of each of these branches is a system of fine electrically conducting fibers known as the *Purkinje fibers* which spread the impulse evenly through the muscle cells of the ventricles, causing them to contract and force blood out of the heart and into the body. On the ECG, the QRS complex represents this electrical activation of the ventricles (Katz, 1992).

The QRS complex (composed of the Q, R, and S waves) is the most identifiable feature of the ECG (refer to Figure 4), and has a much greater amplitude than the P-wave because of the larger mass of the ventricles compared to the atria. It is of about the same duration as the P-wave because the bundle branches and Purkinje fibers facilitate the rapid movement of an electrical impulse through the ventricles. The T-wave represents the repolarization of the ventricle muscle cells (Passman, 1976; Plonsey and Barr, 1988). There is also a wave representing atrial repolarization, however it is masked by the QRS complex. The amplitude, duration, and rhythm of the ECG features (particularly the P-wave and the QRS complex) are used to aid in identifying abnormalities.

Individual ECG records (containing multiple ECG cycles) used in clinical diagnosis can range from a few seconds to many hours. There can be varying degrees of noise present due to machine noise and/or patient movement. The analog ECG is typically sampled at a sampling rate that can vary from 250 Hz to 500 Hz. This range in the sampling rate is sufficient as most of the important information in an ECG is confined to frequencies below 100 Hz (Riggs et al., 1979).

The ECG data used in this work is from the MIT-BIH Arrhythmia Database (MIT-BIH, 1992), which contains data recorded using 24 hour Holter monitors. The digitized data in this database is a result of the sampling the analog recording at 360 Hz. The values of this digitized ECG data range from 0 to 2047, and thus the bit-per-sample rate is 11.

#### **4.4. Background on ECG Compression Techniques**

Efficient coding of the ECG is an important problem in biomedical signal processing because of storage and transmission limitations. This is especially the case for long duration ECGs (such as those resulting from Holter monitors), and for ECGs that need to be transmitted using low bandwidth telephone lines. The goal of ECG compression is to reduce the bit-rate significantly while keeping signal distortion at a clinically acceptable level. For the ECG, this means minimizing the distortion in the P-wave, QRS complex, and the T-wave, the features of most interest in terms of clinical diagnosis. Traditional methods used to code the ECG include DPCM, entropy encoding, and transform methods



such as Fourier and Walsh. An excellent overview of the application of these methods to the ECG is given by Jalaieddine et al. (1990). The conclusions that can be drawn about the effectiveness of these methods for the ECG are as follows:

*DPCM (Differential Pulse Code Modulation)* - Instead of coding the raw signal  $x$ , an error signal  $e = \hat{x} - x$  is used, where  $\hat{x}$  is an estimate of  $x$  using the polynomial predictor:

$$\hat{x}[n] = x[n-1] + \Delta x[n-1] + \Delta^2 x[n-1] + \dots + \Delta^k x[n-1] \quad (6)$$

where  $\Delta x[n-1] = x[n-1] - x[n-2]$  and for  $k > 1$  (where  $k$  is the order of the predictor),  $\Delta^k x[n-1] = \Delta^{k-1} x[n-1] - \Delta^{k-1} x[n-2]$ . A simple implementation of this method (first order) is to set  $\hat{x}[n] = x[n-1]$ . In such an implementation the error signal represents the difference between successive samples of  $x$ . Higher order predictors can be used such as the difference equation,  $\hat{x}[n] = 2x[n-1] - x[n-2]$ , however using higher order difference equations have been shown not to result in any significant increase in compression of the ECG (Jalaieddine et al., 1990). DPCM with thresholding does not result in significant compression for the ECG and can lead to significant distortion in the P-wave (Jalaieddine et al., 1990; Stewart et al., 1973). One interesting thing to note is that the error signal can be viewed as the first detail scale of a wavelet transform, as this is often known as a difference signal.

*Entropy coding* - This lossless coding method is typically used as a follow up to transform based coding (and is applied in this study as well). In this work adaptive entropy

coding (Ziv and Lempel, 1977; Welch, 1984) has been applied as the last step of the algorithm (refer to the Methods section).

*Fourier based transform methods* - Based on the comparison of Fourier based techniques and wavelet transform based techniques with regards to transient signals, and on Section 4.2, these methods do not offer the potential for compression that is possible from wavelet based methods. Previous work has shown that FT based methods result in decreased performance when compared to direct compression techniques such as DPCM. (Shridhar and Stevens, 1979).

#### **4.5. The Wavelet Transform, Vector Quantization, and ECG Compression**

Due to its transient nature, the ECG is a good candidate for study using the wavelet transform. The idea to use the wavelet transform to process ECG signals was initially put forth in the early 1990's (Morlet et al., 1991; Crowe et al., 1992; Michaelis et al., 1993; Anant et al., 1994). Recently, some preliminary studies into the use of wavelet transform based methods for ECG compression have been carried out (Thakor et al., 1993; Chen et al., 1993; Crowe et al., 1992). These preliminary works demonstrated the potential that the wavelet transform has in terms of ECG compression.

The work presented in this chapter links modern transform methods (the wavelet transform) with advanced quantization methods (vector quantization) in developing an ECG compression method. The advantages of these methods were discussed in Sections 4.1 and 4.2. This work extends previous wavelet-based ECG compression techniques by using VQ

on the wavelet coefficients.

In developing the method I have found that the best reconstruction results are obtained with VQ on scales with long duration and low dynamic range (low scales), and scalar quantization (VQ of order one) on scales of short duration and high dynamic range (high scales). This is reasonable since the highest scales have the fewest points yet high dynamic range means that these scales contain transitions of greater magnitude than in the lower scales. These types of transitions prove to be difficult to consistently model using vectors. Though theoretically VQ should give superior performance over scalar quantization for even such situations (follows from the basic Shannon source coding theorems and is demonstrated in Gersho and Gray (1992)), in this application the potential benefit was not worth the cost of VQ coding for the highest scales (refer to Section 4.6.2).

To study the effect of wavelet choice, three different wavelets are used as basis functions and the results are compared. The three wavelets are labeled as Daubechies-12, Bior-orthogonal-18, and Vetterli-22, and have been obtained from recent literature (Daubechies, 1992; Vetterli and Herley, 1992). As the biorthogonal wavelet produces a redundant representation, it is expected that its use will not lead to the “best” compression when compared to the orthogonal wavelets (Daubechies-12 and Vetterli-22). In terms of the orthogonal wavelets, the wavelet that has the highest degree of *regularity* (“smoothest”) should lead to better results. There are two reasons to expect this. First of all, as has been argued by Rioul (1993), the less regular a wavelet is, the greater the chance that quantization errors will be magnified in the reconstruction. In addition, a wavelet with a higher degree of regularity will lead to less correlation between the resulting wavelet coefficients (Tewfik and

Kim, 1992), and the decomposition will more closely approximate the optimal decorrelation of the KLT expansion.

The theory discussed in the previous sections led to an ECG compression method (presented in the following section) which results in a low bit-per-sample rate compared to conventional techniques, with minimal increase in cost for initial coding.

## 4.6. Method

### 4.6.1. Some Notation and Nomenclature

In order to simplify the remaining presentation, a few definitions from Chapter 2 will be restated, and a few new definitions will be introduced. The raw signal (i.e. before wavelet decomposition) will be viewed in terms of the discrete-time sequence:  $c_i^0$  ( $i = 0, \dots, N-1$ ), where  $N$  is the number of samples in the ECG. The discrete-time wavelet transform (DTWT) is implemented using the subband coding scheme that was presented in Chapter 2; where the wavelet is the high-pass filter ( $g$ ), and the low-pass filter ( $h$ ) is the scaling function. As a reminder, there are two outputs at each stage of this procedure:

*Wavelet Coefficients:*

$$d_n^{j+1} = \sum_m g(2n-m) c_m^j \quad (7)$$

*Scaling Coefficients:*

$$c_n^{j+1} = \sum_m h(2n-m) c_m^j \quad (8)$$

where  $j$  represents scale number. The output scaling coefficients become the input to the next stage in the DTWT. (Refer to Figure 5). The reconstruction is accomplished using the reconstruction filters,  $\tilde{g}$  and  $\tilde{h}$ , (as defined in Chapter 2) and the following formula:

$$c_n^{j-1} = \sum_m \tilde{h}(n-2m) c_m^j + \sum_m \tilde{g}(n-2m) d_m^j \quad (9)$$

Therefore, the DTWT decomposes the initial sequence,  $c_i^0$ , into a set of subsequences:  $d_n^1$  ( $n = 0, \dots, (N/2) - 1$ ),  $d_n^2$  ( $n = 0, \dots, (N/4) - 1$ ), ...,  $d_n^M$  ( $n = 0, \dots, (N/2^M) - 1$ ), and  $c_n^M$  ( $n = 0, \dots, (N/2^M) - 1$ ), where the decomposition order (total number of scales) is  $M$ .

#### 4.6.2. Development of Method

The first issue in the development of the method was the construction of the set of code-book vectors. A total of 100 ECG segments, each containing 1024 samples and three to four ECG cycles (i.e. P-wave, QRS complex, and T-wave) were used to develop the training set. Each of these ECG segments was decomposed into five wavelet scales,  $d^j$  ( $j = 1, \dots, 5$ ) and  $c^5$ . It should be clear from the formula in (9) that these are the only coefficients needed for reconstruction. These coefficients are used to develop the training set. Each  $d^j$  (and  $c^5$ ) is divided into vectors of length  $k$ . These training vectors are then grouped using *k-means clustering* (which is also known as the Generalized Lloyd Algorithm (Gersho and Gray, 1992)). K-means clustering groups the initial set of vectors based

on nearest neighbor partitioning (Duda and Hart, 1973; Selim and Ismail, 1984). Nearest neighbor based encoders have been shown by Gersho and Gray (1992) to be optimal in terms of minimizing average distortion. This is the case for both vector quantization as well as scalar quantization. In k-means clustering the initial  $N$  training vectors are initialized as the  $N$  cluster centers (where the number of cluster centers desired is pre-determined). The remaining training vectors are then grouped in the clusters based on their distance from the cluster centers. The cluster centers are then recomputed and then the vectors are again clustered. This process continues until the cluster centers converge. A more detailed description as well as an example of this process is provided in Appendix 4.A.

The number of training vectors used in the clustering was more than sufficiently large, ranging from 1,700 to 13,000, depending on the scale number. (Since at higher scales the number of samples are fewer than at lower scales, the number of training vectors decreases with scale). The cluster center vectors are chosen as the *codebook* vectors, each of which has length  $k$ . A codebook,  $B^j$ , was developed for each scale,  $j$ . Each codebook contained  $N/k2^j$  vectors with each vector represented by a pointer  $q_n^j$  ( $n = 0, \dots, (N/k2^j) - 1$ ), where  $N$  is the length of the original ECG.

One of the key issues regarding the use of VQ is the choice of the vector length,  $k$ . Since VQ is being applied to the wavelet coefficients,  $d^j$ , at several scales,  $j$ , and since each  $d^j$  will have different characteristics (number of points, dynamic range, etc.), it is to be expected that the value of  $k$  will be dependent on scale. To determine the values of  $k$  a preliminary analysis using one test ECG segment was required. This segment was pro-

cessed by the wavelet transform and each scale was coded (the coding procedure will be described in detail in the next subsection) using codebooks based on the same value of  $k = 6$  for each scale. It was initially decided to have 256 vectors in each codebook (meaning 8 bits are required to point to any vector in a given codebook). Upon decoding and reconstruction of each  $d^j$  (and  $c^5$ ), it was found that there was significant error in the reconstruction of the higher scales (low frequencies). Increasing the number of vectors in each codebook to 512 (9 bits) did not change the error in the high scales significantly. However, by using a lower value of  $k$  for the higher scales, the error in the reconstruction of these scales was reduced drastically. The conclusion drawn from this analysis is that the value of  $k$  to be used in VQ coding is dependent on scale number, and the higher the scale the lower the necessary value of  $k$  to achieve minimal error in reconstruction of the coefficients. As was noted previously this variation of the value of  $k$  to use by scale is due to the fact that the dynamic range of the coefficients increases with scale. In the next subsection it will be shown how these observations were used in the coding of the test set.

#### *4.6.3. Application of Method to Test Set*

The test set consisted of 100 ECGs similar, yet completely separate, to those in the training set. The first step was to decompose each ECG into five wavelet scales. The wavelet decomposition of a test ECG using the Vetterli-22 wavelet is shown in Figure 6. Three different wavelet filters were tested and the performance between them was compared. As mentioned in Section 4.5, the wavelets used were the Vetterli-22, Biorthogonal-

18 (Vetterli and Herley, 1992), and the Daubechies-12 (Daubechies, 1992) wavelets.

These wavelets are shown in Figure 4 of Chapter 3.

The subsequences  $d^j$  and  $c^5$  resulting from the wavelet decomposition were each divided into vectors of length  $k$ . These vectors were compared to the appropriate set of codebook vectors (utilizing the codebooks,  $B^j$ ), and a codebook vector was chosen based on the squared Euclidean distance measure. The codebook vector closest to the vector being coded was chosen, and, the coded vector was represented by the appropriate pointer,  $q_n^j$ . Using the observations of Section 4.6.2, the codebooks for the low scales contained different length vectors than the codebooks for the two highest scales. In addition it was found that only 64 (6 bit descriptor) vectors were needed for each  $B^j$ . Using fewer vectors resulted in significant increases in reconstruction PRDs while using more vectors per codebook increased the bit-per-sample rate and did not result in noticeable decreases in PRD.

To encode  $d^j$  ( $j = 1, 2, 3$ ), three different values of  $k$  (4, 6 and 8) were tested and the results compared, while for  $d^j$  ( $j = 4, 5$ ), as well as for  $c^5$ , a value of  $k = 1$  was chosen. All the 100 test ECGs were coded in this manner, and in the end the only data stored was the set of pointers,  $q_n^j$ , for each ECG. The final step was to use apply entropy coding to the set of  $q_n^j$  to minimize any redundancies. As mentioned earlier, this is a standard step in any transformation based coding procedure (Vetterli and Kovacevic, 1995). The entropy coding was achieved using a Lempel-Ziv-Welch compress utility (Welch, 1984).

For reconstruction of the each test ECG, the compressed data was first uncompressed resulting in the appropriate set of  $q_n^j$  values which were used in conjunction with the



codebooks to reconstruct the wavelet and scaling coefficients. These coefficients,  $\hat{d}^j$  (and  $\hat{c}^5$ ), were then processed by the inverse DWT. A block diagram representation of the method described in this section is shown in Figure 7.

#### 4.7. Results

Two performance criteria were used to evaluate the effectiveness of the method described in the previous section. One was the bit-per-sample rate and the other the percent root mean square difference (PRD). The bit-per-sample rate measures the degree of compression while the PRD measures the error between the reconstructed ECG and the original ECG. To calculate the bit-per-sample rate, the storage size in bits of the compressed data was divided by the number of samples in the original data.

$$\text{bit-per-sample rate} = \frac{\text{Size of compressed data in bits}}{\text{Number of samples in original data}} \quad (10)$$

The PRD was calculated in the usual manner (Lynch, 1985):

$$PRD = \frac{\sqrt{\sum_{i=0}^{N-1} (\hat{x}(i) - x(i))^2}}{\sqrt{\sum_{i=0}^{N-1} x^2(i)}} \times 100 \quad (11)$$

where  $x$  represents the original sequence and  $\hat{x}$ , the reconstructed sequence. This is a

standard error measure used in ECG analysis (Jalaleddine et al., 1990; Cetin et al., 1993; Shahein and Abbas, 1994). As stated by Shahein and Abbas (1994), the PRD is a fairly good error measure for the ECG, but it may sometimes be misleading, particularly when significant noise is present in the original data. This will be seen in an illustration below, where a good reconstruction from a clinical point of view has a higher than average PRD.

By using a value of  $k = 4$  to encode  $d^j$  ( $j = 1, 2, 3$ ), the resulting bit-per-sample rate was 1.5, for  $k = 6$  the bit-per-sample rate was 1.2, while for  $k = 8$  the rate achieved was down to 1.1. Even with a rate as low as 1.1 there were several test set cases with PRDs below 5.5% with one case having a PRD of 4.8%. Another variable of interest was the wavelet used to decompose the ECG, and the results are, thus, arranged according to wavelet in the tables below. Tables 1 and 3 show the overall results for the 100 case test set, while Table 2 shows the PRD results for one particular case in the test set. Table 1 shows the average PRD for the test set versus bit-per-sample rate achieved and wavelet function used in the decomposition step. Plots of the data in Table 1 are presented in Figures 10-12. Table 3 shows how the three different wavelets performed on a case by case basis.

For each case, three different PRD results were available based on wavelet used for the decomposition. The wavelet that produced the lowest PRD for each case was noted and the result of this compilation is what is presented in the table. For example, the second row shows that for 65 of the 100 test cases the Vetterli-22 wavelet led to the lowest PRD (for an encoded bit-per-sample rate of 1.2). In Figures 8 and 9 are shown original ECGs and their respective reconstructed ECGs, i. e.,  $c^0$  and  $\hat{c}^0$  (an example of this can also be found

in Figure 7). Figure 8 is a presentation of a case where a below average PRD was achieved using the Vetterli-22 wavelet and a value of  $k = 8$  in the VQ step. Figure 9 is again a presentation of a case that used the Vetterli-22 wavelet and  $k = 8$ , but the resulting PRD (17.07%) was well above average. In fact, based on the PRD this case produced one of the worst results. However, as can be seen from the plot, the reconstruction looks quite good, particularly when viewed from the perspective of a clinician. There does not seem to be any possibility that  $\hat{c}^0$  in Figure 9 would lead to different diagnostic conclusions than  $c^0$ . Even for such a noisy ECG the key features (P-wave, QRS complex) retain their integrity after processing. Thus, for a case such as this just using PRD is not the best way to measure the success of the method.

From the results as presented in the tables it can be seen that, overall, the Vetterli-22 wavelet produced significantly better PRD results than the other two wavelets that were used. This is particularly highlighted in Table 3. The Biorthogonal-18 wavelet produced the worst results of the three wavelets. This is the result predicted in Section 4.5 and will be discussed further in the next section.

**TABLE 1. Average PRD (%) for 100 case test set (standard deviation appears in brackets). Each column represents the PRDs for a different wavelet used in the decomposition step.**

| BITS/<br>SAMPLE | AVERAGE PRD(%) FOR TEST SET USING X WAVELET |             |             |
|-----------------|---|-------------|-------------|
|                 | X=VET-22                                    | X=BIORTH-18 | X=DAUB-12   |
| 1.1             | 9.44 (3.76)                                 | 9.85 (3.89) | 9.50 (3.49) |
| 1.2             | 8.26 (2.97)                                 | 9.00 (3.11) | 8.67 (3.12) |
| 1.5             | 6.27 (1.79)                                 | 6.71 (1.73) | 6.72 (1.78) |

**TABLE 2. PRD (%) data for one particular ECG segment.**

| BITS/<br>SAMPLE | PRD(%) FOR SIG115 USING X WAVELET |             |           |
|-----------------|-----------------------------------|-------------|-----------|
|                 | X=VET-22                          | X=BIORTH-18 | X=DAUB-12 |
| 1.1             | 5.12                              | 5.64        | 5.19      |
| 1.2             | 4.78                              | 5.50        | 4.82      |
| 1.5             | 4.08                              | 5.15        | 4.46      |

**TABLE 3. Comparison of PRD values for 100 case test set. Table is divided based on wavelet used in decomposition step and bit-per-sample rate.**

| BITS/<br>SAMPLE | % OF CASES WHERE USE OF X WAVELET RESULTED IN LOWEST PRD |             |           |
|-----------------|--|-------------|-----------|
|                 | X=VET-22   | X=BIORTH-18 | X=DAUB-12 |
| 1.1             | 44   | 21          | 35        |
| 1.2             | 65   | 13          | 22        |
| 1.5             | 61   | 19          | 20        |

#### 4.8. Discussion and Conclusions

The results above show that using VQ on the wavelet coefficients of ECG signals leads to very low bit-per-sample rates while retaining feature integrity in the processed signal.

Even with relatively high resultant PRDs, the important signal features are retained as is demonstrated in Figure 9. This happens to be the goal of ECG compression techniques as stated in Section 4.4. The algorithm described produces results that can be viewed favorably when compared to results from other techniques (Jalaleddine et al., 1990). Both the use of VQ and the wavelet transform have contributed to the success of this method. Vector quantization made possible the very low bit-per-sample rates, since this rate is governed by  $(\log_2 N) / k$  (refer to Section 4.1). The use of the wavelet transform was helpful because of its power to decorrelate data and produce a resulting compact representation (refer to Section 4.2).

The choice of wavelet was shown to be important, as was expected. As stated in Section 4.7, the Biorthogonal-18 wavelet produced the worst PRD results when compared to the results produced from the use of the other two wavelets. This is not surprising, since the decomposition resulting from the use of *biorthogonal* wavelets is redundant, while *orthogonal* wavelets such as Vetterli-22 and Daubechies-12 produce non-redundant, efficient representations in the wavelet domain (refer to Chapter 2). Though the results in Table 1 and 2 show that the performance of the Vetterli-22 and Daubechies-12 wavelets is similar, when the information from Table 3 is taken into account, it is clear that the best performance was achieved with the use of the Vetterli-22 wavelet. This is also an expected result as the Vetterli-22 wavelet is “smoother” than the Daubechies-12 wavelet and thus has a higher *degree of regularity*, which is an important factor in terms of compression (refer to Section 4.5). In addition, the fact that the Vetterli-22 wavelet is the smoother wavelet means that it produces a set of bandpass filters that are better defined in the fre-

quency domain (refer to Figure 13), which also implies a more efficient decomposition. Thus, the importance of regularity has been confirmed in terms of wavelet choice for ECG compression. This can be expressed either in the time domain (“smooth” wavelet functions) or in the frequency domain (well defined frequency selection of the bandpass filters).

#### Appendix 4.A K-means clustering

K-means clustering is a nearest neighbor based clustering algorithm, which means that the distance between vectors:

$$\|x - y\| = [(x(1) - y(1))^2 + (x(2) - y(2))^2 + \dots + (x(k) - y(k))^2]^{1/2} \quad (12)$$

is used to group the vectors. Given the set of training vectors  $x_1, x_2, x_3, \dots, x_L$  where  $L$  is the size of the training set, the initial cluster centers vectors,  $R_j^0$ , are chosen as:

$$R_j^0 = x_j \quad j = 1, 2, 3, \dots, N \quad (13)$$

where  $N$  equals the number of desired clusters. The remaining training vectors,  $x_i : i = N + 1, N + 2, \dots, L$  are grouped into the appropriate cluster using the following measure:

$$x_i \text{ belongs to } R_j^n \text{ for the } j \text{ such that } \|x_i - R_j^n\| \text{ is a minimum} \quad (14)$$

remembering that  $n = 0$  for the initial set of cluster centers. The cluster center vectors are then recalculated using the following relation:

$$\mathbf{R}_j^n = \frac{1}{N_j} \sum_{i=1}^{N_j} (x_{j,i}(1), x_{j,i}(2), x_{j,i}(3), \dots, x_{j,i}(k)) \quad (15)$$

where  $N_j$  is the number of vectors in cluster  $j$ ,  $k$  is the length of the vectors, and  $x_{j,i}$  is the  $i^{th}$  vector in cluster  $j$ . This can be seen as finding the centroid of the cluster. The entire training set is then read again and the vectors are regrouped using (14) and new cluster center vectors are computed using (15). This process is iterated until the cluster center vectors converge.

As an example of the above procedure consider the following set of vectors:

$$\begin{aligned} \mathbf{x}_1 &= (0.1, 0.2) \\ \mathbf{x}_2 &= (0.3, 1.0) \\ \mathbf{x}_3 &= (0.1, 0.0) \\ \mathbf{x}_4 &= (0.9, 0.8) \\ \mathbf{x}_5 &= (1.0, 0.1) \end{aligned}$$

Let's set the desired number of clusters to  $N = 2$ . So, using (13) we have:  $\mathbf{R}_1^0 = \mathbf{x}_1$  and  $\mathbf{R}_2^0 = \mathbf{x}_2$ . Using (14) it is found that cluster 1 consists of  $\mathbf{x}_1, \mathbf{x}_3, \mathbf{x}_5$  and cluster 2 contains the vectors,  $\mathbf{x}_2, \mathbf{x}_4$ . The cluster center vectors are then recalculated using (15):

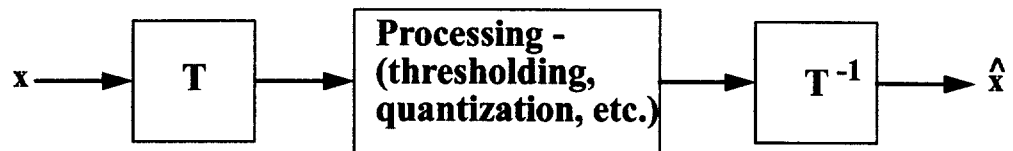
$$\mathbf{R}_1^1 = \frac{1}{3} \{ (0.1 + 0.1 + 1.0), (0.2 + 0.0 + 0.1) \} = (0.4, 0.1) \quad (16)$$

$$\mathbf{R}_2^1 = \frac{1}{2} \{ (0.3 + 0.9), (1.0 + 0.8) \} = (0.6, 0.9) \quad (17)$$

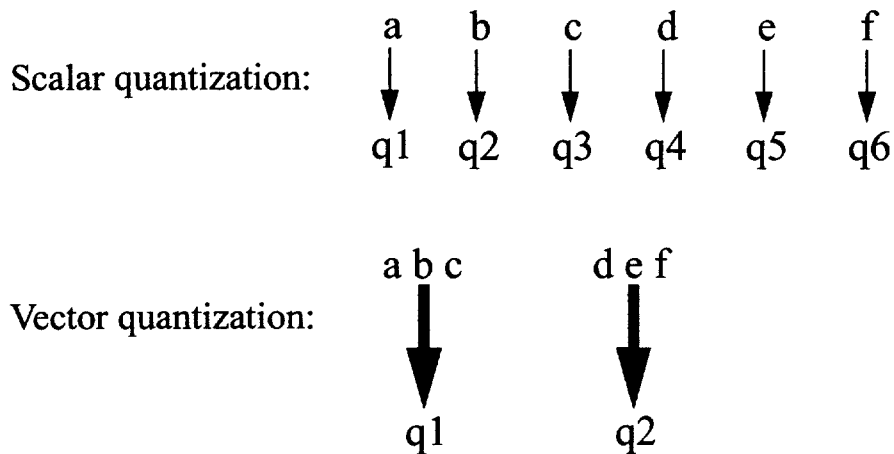
The vectors,  $\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3, \mathbf{x}_4, \mathbf{x}_5$ , are then reclustered using (14) and the procedure is iterated until  $\mathbf{R}_1^n$  and  $\mathbf{R}_2^n$  converge. In this particularly simple example the cluster centers have already converged, since by using the new cluster centers computed in (16) and (17) it can be seen that cluster 1 still contains  $\mathbf{x}_1, \mathbf{x}_3, \mathbf{x}_5$  and cluster 2 contains  $\mathbf{x}_2, \mathbf{x}_4$ . Thus, the cluster centers for the second iteration will be the same, i.e.,  $\mathbf{R}_1^2 = \mathbf{R}_1^1$  and  $\mathbf{R}_2^2 = \mathbf{R}_2^1$ . For more complicated cases (greater number of vectors, higher dimensions) more iterations would be required.



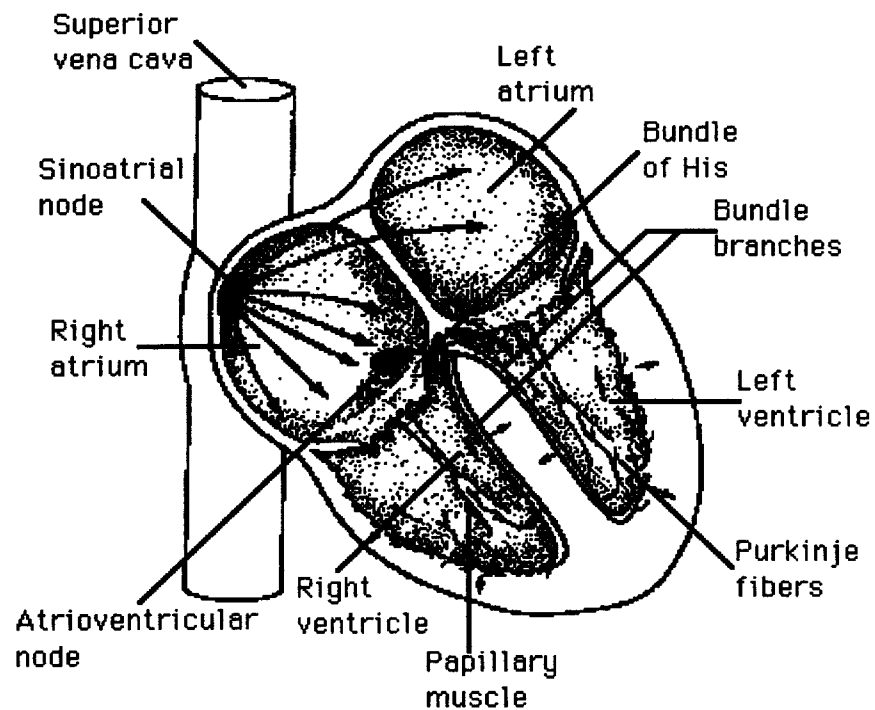
**FIGURE 1. Basic transformation based compression procedure**



**FIGURE 2. Scalar quantization and vector quantization; scalar quantization maps a single sample to a discrete value, while vector quantization maps a block or vector of samples to a discrete value**



**FIGURE 3. A cross section of the human heart (Gruener, 1996)**



**FIGURE 4. The ECG and its important features (P wave, QRS complex, and T wave)**

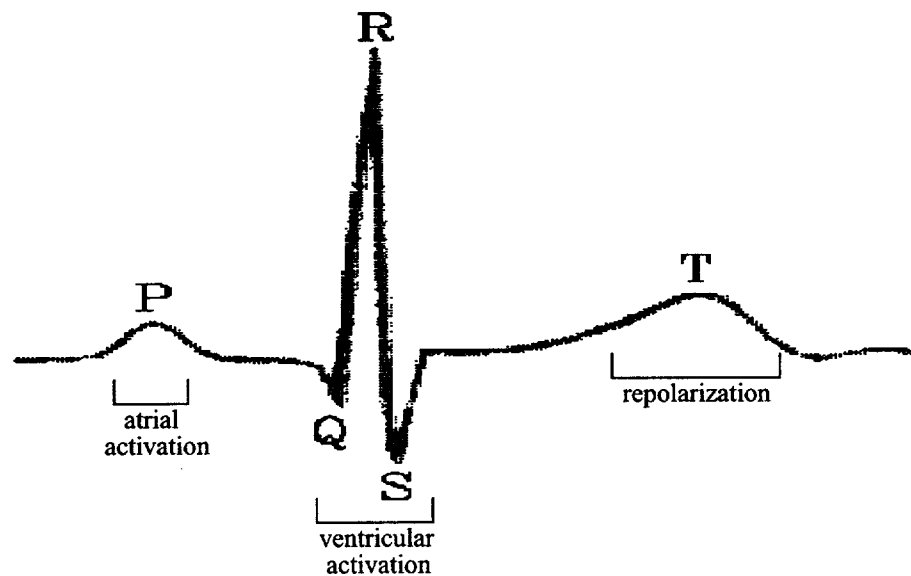


FIGURE 5. Two scales of wavelet decomposition of ECG,  $c^0$ .

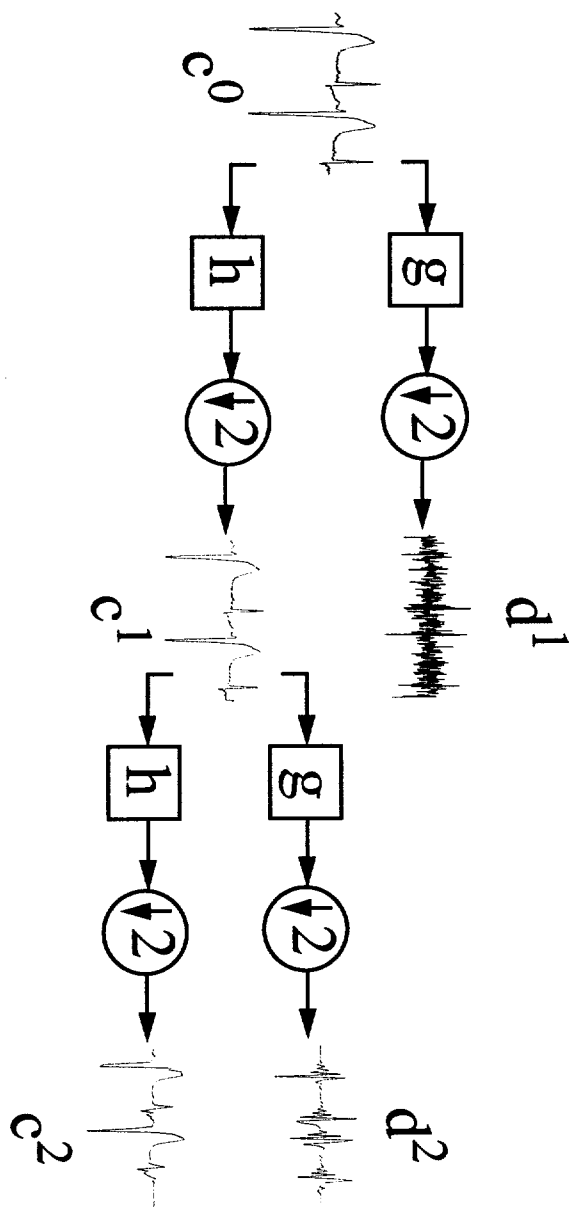
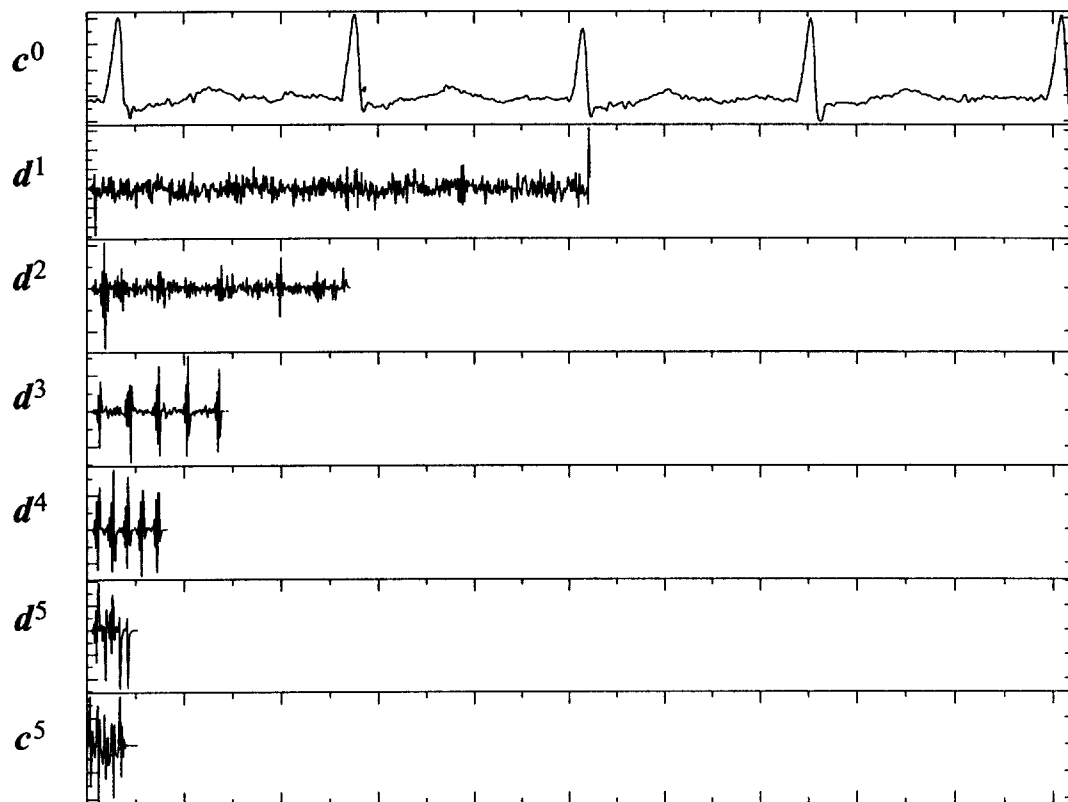
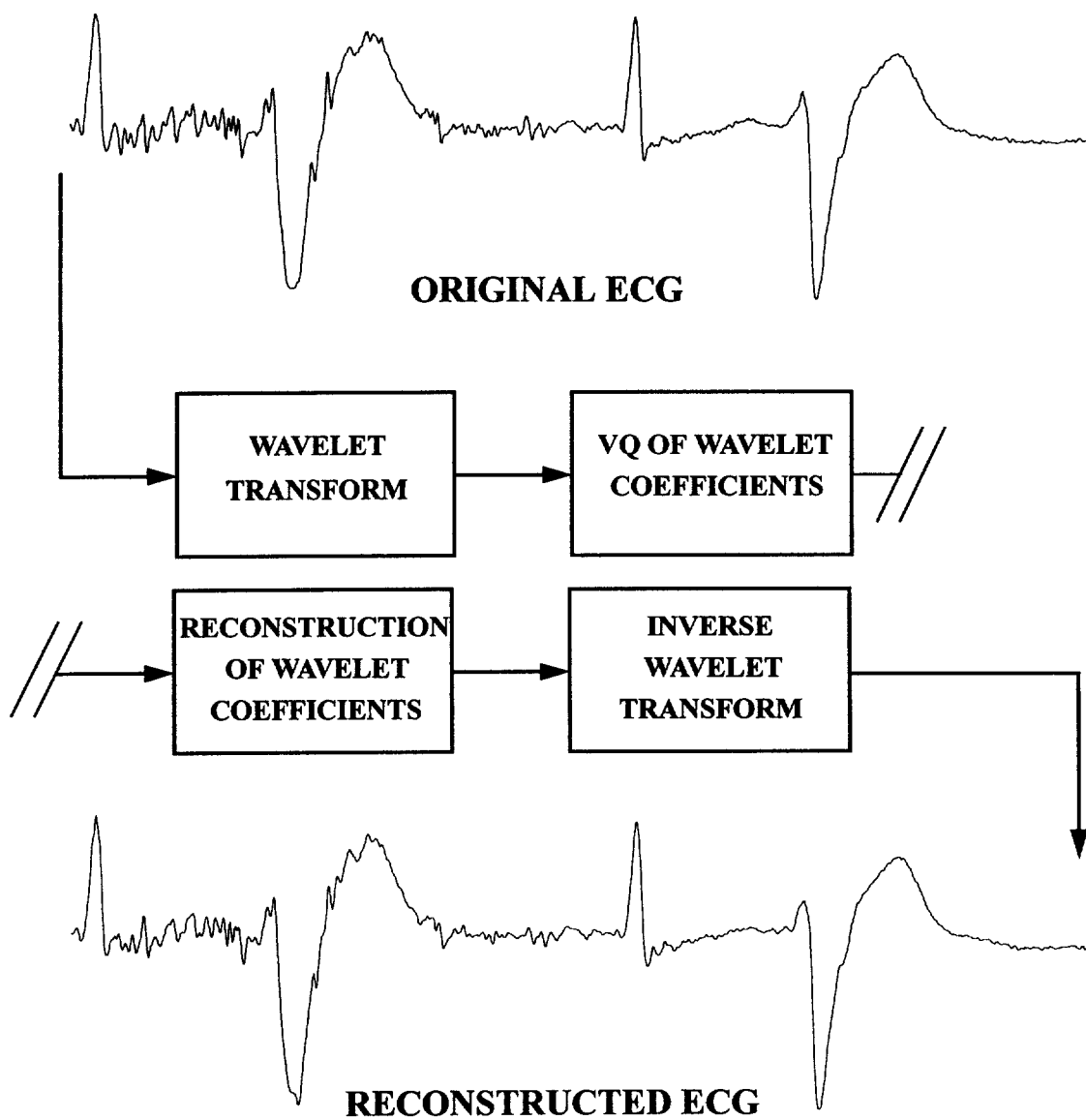
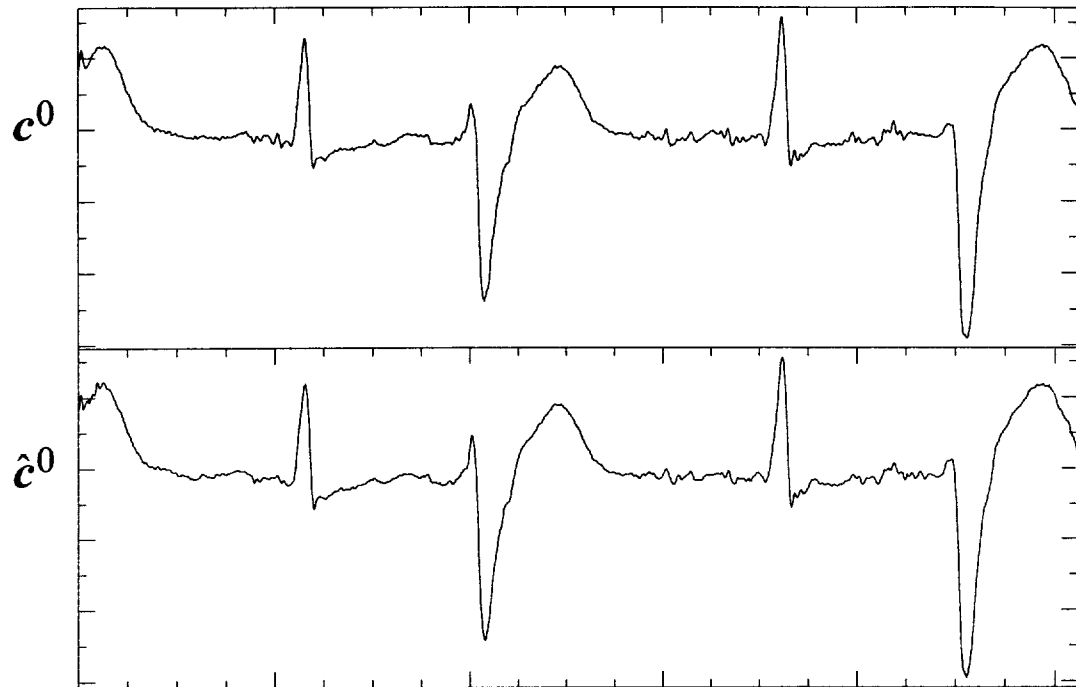


FIGURE 6. A test ECG and its wavelet decomposition of five scales. The original signal is labeled  $c^0$ , the wavelet coefficients are labeled  $d^j$  ( $j = 1, 2, 3, 4, 5$ ), and the scaling coefficients at the final scale are labeled  $c^5$ .

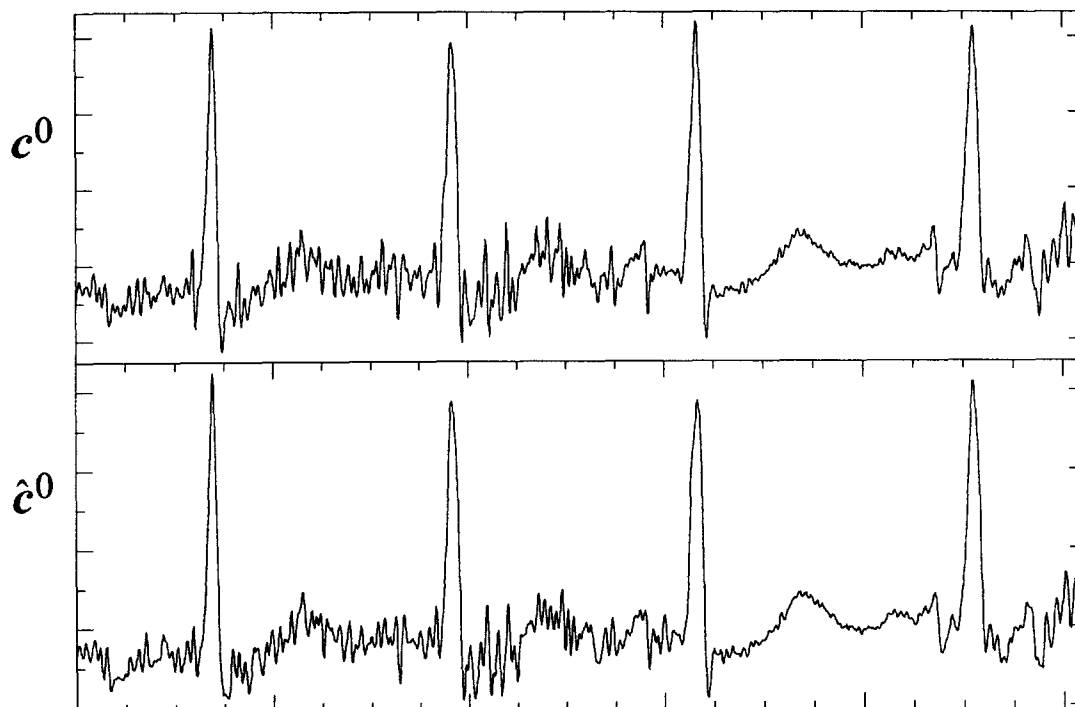


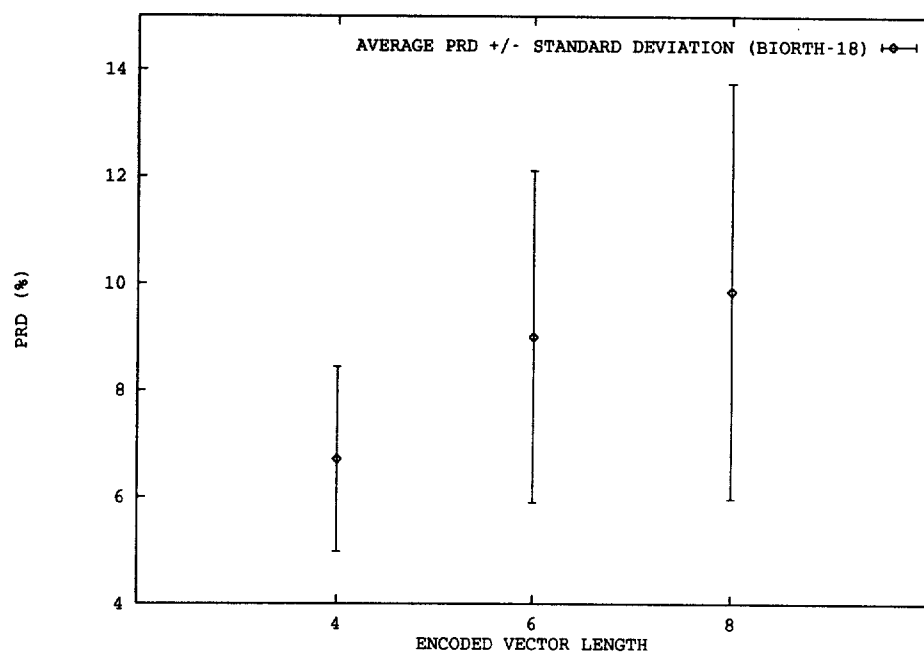
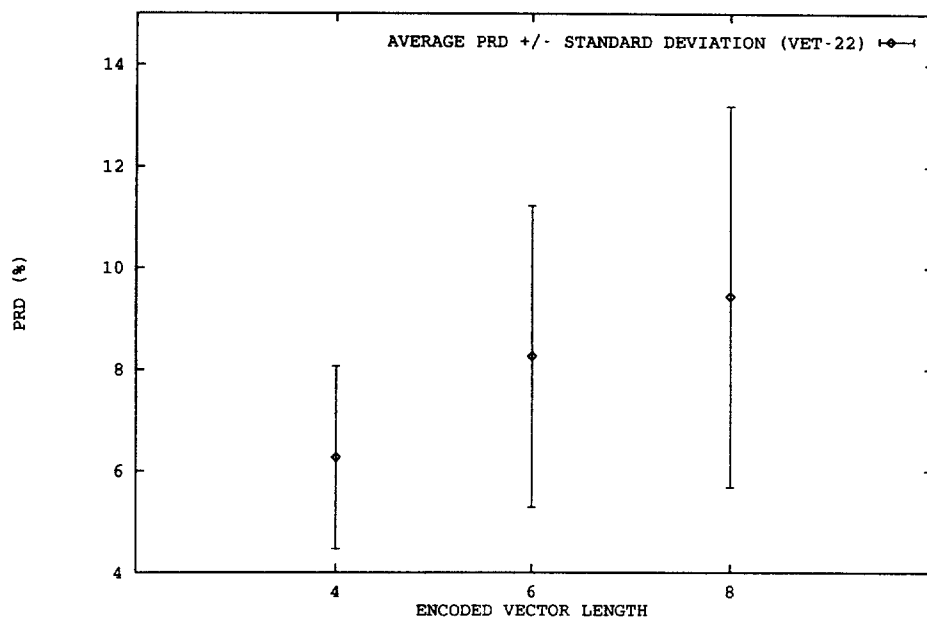
**FIGURE 7. Block diagram of method**

**FIGURE 8. Original ECG and reconstructed ECG from a case resulting in a below average PRD (5.43%).**

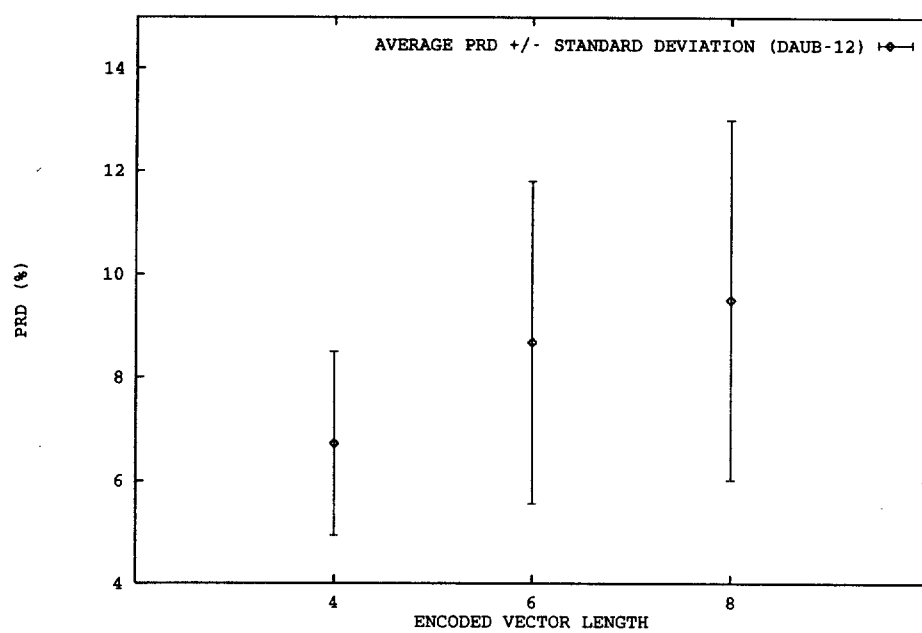


**FIGURE 9. Original ECG and reconstructed ECG for case with above average PRD(17.07%). Notice how  $c^0$  and  $\hat{c}^0$  still look very similar and in particular the key features of clinical importance are retained.**

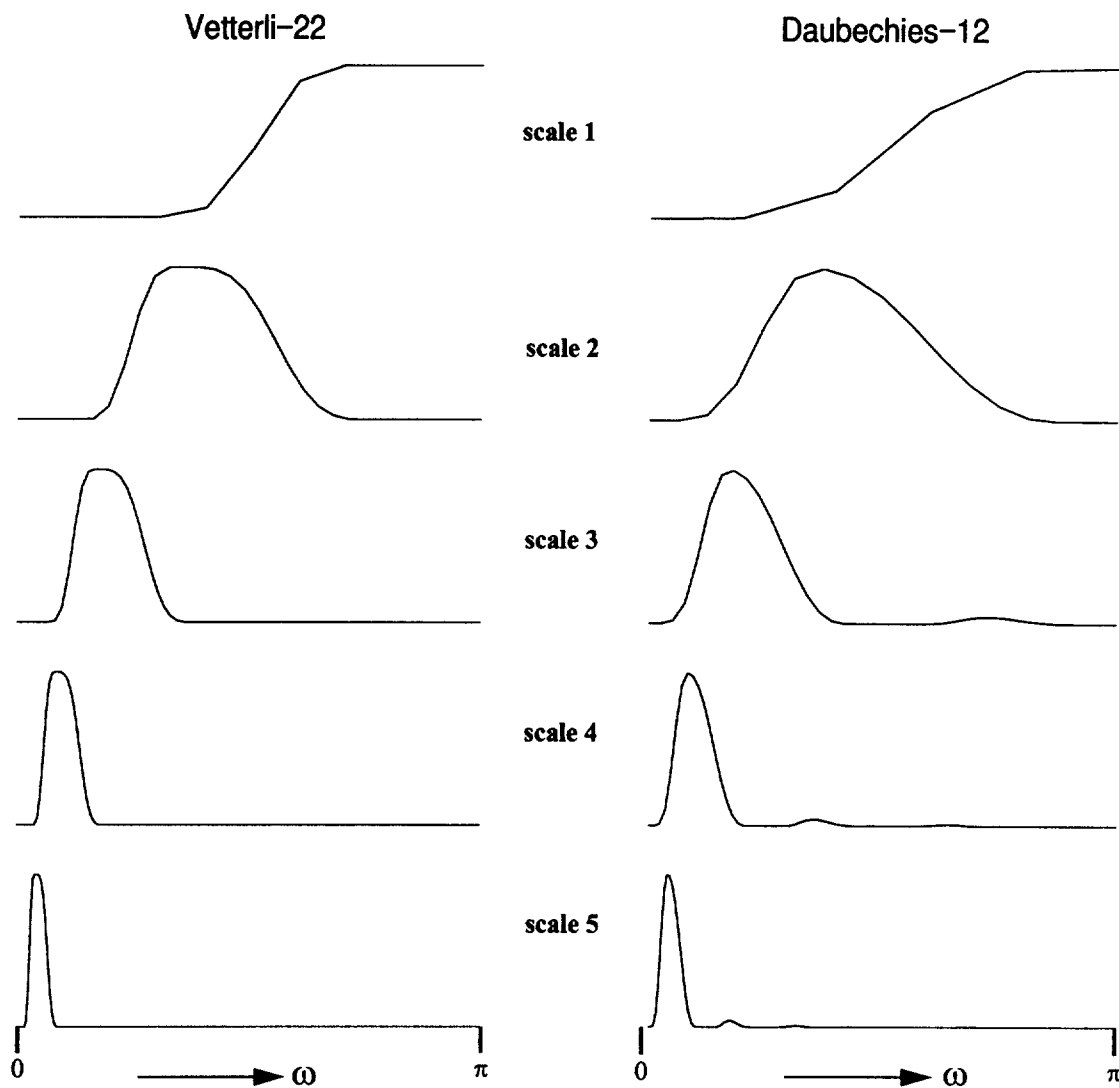


**FIGURE 10. PRD as a function of vector length used for biorthogonal-18 wavelet****FIGURE 11. PRD as a function of vector length used for Vetterli-22 wavelet**



**FIGURE 12. PRD as a function of vector length used for Daubechies-12 wavelet**

**FIGURE 13. Frequency domain decomposition of Vetterli-22 and Daubechies-12 wavelets by scale. Notice how the bands for the Vetterli-22 wavelet are “sharper”.**



**CHAPTER 5***Wavelet Design by Linear Prediction Lifting***5.0 Summary**

As should have been noticed from Chapters 3 and 4, the choice of wavelet in applications of the wavelet transform can significantly influence results. Therefore it is not surprising that an ongoing area of interest in wavelet research is wavelet design using information about the signal being analyzed. Many of the design methods involve matching the Fourier spectrum of the wavelet with the signal. Due to the constraints in these design methods, the resulting type of wavelets are limited and usually do not offer any advantage over existing “library” wavelets. A strictly time domain wavelet design method has recently been introduced by Sweldens (1995). In this chapter, a modification of this method is developed by introducing a linear prediction step into the design algorithm. The resulting filters are perfect reconstruction and regular and thus qualify as wavelet filters as defined in Chapter 2.

## 5.1 Introduction

Throughout this thesis a recurring issue has been the choice of wavelet basis. As mentioned in the previous chapters, analyzing wavelets can be chosen from a library of existing wavelet functions (which are constructed mathematically and are not based on signal or feature matching) or can be designed independently (using signal or feature information). In Chapter 4 it was found that promising results were obtained by using existing wavelets. Orthogonal wavelets provided the best compression results. In Chapter 3, library wavelets provided good results for most cases, but evidence was provided that *pseudo-wavelets* designed directly from the P arrival of the seismic signal could provide improved results particularly for some of the problem cases. These pseudo-wavelets lacked perfect reconstruction properties and thus were only used to test the hypothesis. In this chapter, I have taken this idea of wavelet design one step further by introducing an algorithm that allows for the design of perfect reconstruction wavelets based on the feature of interest by using linear prediction. These wavelets are applied to the problem cases of Chapter 3, with positive results. The algorithm that I present is based on a modification of a recently developed time-domain based wavelet design method called the lifting scheme (refer to Section 5.3). In Section 5.2 and 5.3 existing wavelet design methods are reviewed. Sections 5.4 and 5.5 contain the theory and application respectively of a wavelet design method using the lifting scheme and linear prediction.

## 5.2 Wavelet Design Methods: Fourier Transform Based

Most current wavelet design methods are Fourier transform based and involve the use of popular filter design methods such as that introduced by Parks and McClellan (1972). Such a wavelet design method is presented in detail by Vaidyanathan (1993). It is based on the following relation that was presented in Chapter 2:

$$H(z)H(z^{-1}) + H(-z)H(-z^{-1}) = 2 \quad (1)$$

which can also be written as:

$$P(z) + P(-z) = 2 \quad (2)$$

where  $P(z) = H(z)H(z^{-1})$ . The goal is to design  $P(z)$  and then use *spectral factorization* to get  $H(z)$ . Here the term *spectral factorization* refers to the method by which the real coefficient minimum-phase spectral factor,  $H(z)$ , is separated from  $P(z)$ . This is accomplished by finding the zeros of  $P(z)$  and then retaining all the zeros that are in the unit circle (in the  $z$ -plane) and one out of every double zero on the unit circle. The representation in the time-domain,  $h[n]$ , is found by simply using the inverse transform. The remaining analysis filter,  $g[n]$ , as well as the reconstruction filters,  $\tilde{g}[n]$  and  $\tilde{h}[n]$ , are easily obtainable using relations (27) and (38) from Chapter 2. The algorithm and code that implements the above described design method is presented in Appendix 5.A.

Theoretically, by using information about the frequency response of the signal or feature being analyzed, the above described method can be used to design wavelets that have

similar frequency responses. In practice, this is very difficult, as it is rare for the frequency response of a real signal to be well defined. In addition the constraint that  $\omega_p + \omega_s = \pi$  (refer to Appendix 5.A) leaves little room to develop a great variation of wavelets. In addition, given the design algorithm of Appendix 5.A, for a given order of desired filter, designing more than a few *perfect reconstruction* filters is difficult. Figure 1 illustrates three filters designed using  $N = 10$ ; and  $\omega_p = 0.2\pi$ ,  $\omega_p = 0.25\pi$ , and  $\omega_p = 0.3\pi$ . Varying the passband frequency,  $\omega_p$ , further resulted in filters that were not perfect reconstruction. As can be seen, the filters in Figure 1 are not that different, which is due to the design constraints mentioned above. Also, as has been stated previously, the Fourier-domain is a rather abstract domain, and thus a time-domain based design method (the topic of the next sections) would be preferable to a Fourier based one.

### 5.3 Wavelet Design Methods: Lifting Scheme

Recently a new method, known as the lifting scheme, has been developed for constructing wavelet transforms as well as designing wavelets strictly in the time-domain (Sweldens, 1995; Sweldens, 1996a; Sweldens, 1996b). The lifting scheme is a way to execute the wavelet transform involving three steps: *split*, *predict*, and *update*. The wavelet in this procedure is the transfer function resulting from the combination of these three operations. A description of these steps is given below. The notation used is consistent with that used in Chapter 2, while some new terminology is introduced from the references listed above.

1. *Split* - In this step the function,  $c^j$  (where  $c^0$  is the original signal), is split into even and odd samples. The even-indexed samples are kept as the scaling coefficients while the odd-indexed samples are kept as the wavelet coefficients.

$$c^{j+1}[n] = c^j[2n] \quad (3)$$

$$d^{j+1}[n] = c^j[2n+1] \quad (4)$$

Both these sets of coefficients will be modified in the remaining steps of the lifting scheme. The implementation of the split phase as represented by (3) and (4) is termed the *Lazy wavelet* implementation. This phase can also be implemented by using other methods such as average and difference operations (Haar wavelet) to accomplish the splitting. The split phase can be seen as analogous to the conventional implementation of the wavelet transform.

2. *Predict* - To decorrelate the data further and to decrease the energy in the wavelet coefficients ( $d^{j+1}$ ), a predictor is used to predict the odd-indexed samples of  $c^j$ , and then to encode  $d^{j+1}$  as the difference between the predicted samples and actual samples. Thus, in this step the scaling coefficients are used to obtain the wavelet coefficients, i.e.:

$$d^{j+1}[n] = c^j[2n+1] - \mathcal{P}(c^j[2n+1]) \quad (5)$$

where  $\mathcal{P}$  is the prediction operator. Refer to Figure 2 for an illustration of how the above equation is applied. In the lifting scheme implemented by Sweldens (1995) the prediction operator is a polynomial predictor whose order can be chosen.  $N$  samples are used to con-

struct polynomials of degree  $N-1$ . For example, the simplest predictor is implemented by using  $N = 2$ :

$$d^{j+1}[n] = c^j[2n+1] - \frac{1}{2}(c^{j+1}[n] + c^{j+1}[n+1]) \quad (6)$$

Higher order polynomial predictors simply use more of the samples from  $c^{j+1}$  (with the appropriate weight factors) for the prediction. In this implementation the prediction is always symmetric. With a good predictor a compact representation of the original data can be realized.

3. *Update* - To maintain some global properties of the original data (such as the mean) in the scaling coefficients,  $c^j$ , these coefficients are *lifted* using the wavelet coefficients,  $d^j$ , i.e.:

$$c^j = c^j + \mathcal{U}(d^j) \quad (7)$$

where  $\mathcal{U}$  is the update operator. The result of this step is to preserve the moments of  $c^j$  at every scale. For example to preserve the first order moment (the mean) it is required that:

$$\sum_n c^{j+1}[n] = \frac{1}{2} \sum_n c^j[n] \quad (8)$$

The update equation (7) then becomes:

$$c^j[n] = c^j[n] + \frac{1}{4}(d^j[n-1] + d^j[n]) \quad (9)$$



Again, just as in prediction, higher order moments can be preserved by using more wavelet coefficients with the relevant weightings. Also just as in the prediction, the *lifting* is applied in a symmetric manner.

Figure 2 illustrates, in block diagram form, the steps outlined above. These steps are repeated for the number of scales,  $j$ , that are desired. Also in Figure 2 are examples which illustrate the application of each of the lifting method steps. The reconstruction is accomplished by reversing the order of the above operations (with opposite signs in (5) and (7)). Of course, in the reconstruction, the split step becomes a *merge* step.

As mentioned earlier, the wavelet in the lifting scheme is the transfer function that results from the combination of the operations above. The variables are the order of prediction, and order of update. In practice the order of update is best left as in (9). The wavelet is obtained by executing the inverse with only a delta function in the last  $d^j$ . Figure 3 shows two wavelets resulting from using different degree polynomials in the prediction step.

Implementation details of the lifting scheme, including detailed code and examples, are given in a detailed report by Fernandez et al. (1996).

## 5.4 Linear Prediction and Lifting for Wavelet Design

### 5.4.1. Linear Prediction for Predict Step of Lifting Scheme

One advantage of the lifting scheme is that the predictor in the predict step can be modified so that it reflects the characteristics of the data being analyzed. The goal is for the resulting wavelet to thus be geared to the data. In this section a modification of the predict step is introduced that realizes this goal. The prediction method chosen is known as *linear prediction*, and is defined as the prediction of a data point using a linear combination of the  $m$  past values (Makhoul, 1975; Strobach, 1990; Gersho and Gray, 1992), i.e.:

$$\hat{x}[n] = - \sum_{k=1}^m a[k] x[n-k] \quad (10)$$

where the above represents *forward prediction* and  $a[k]$  are known as the *forward prediction coefficients*. Similarly, *backward prediction* is represented as such:

$$\hat{x}[n-m-1] = - \sum_{k=1}^m b[k] x[n-k] \quad (11)$$

where  $b[k]$  are the *backward prediction coefficients*. Under certain stationary assumptions the backward prediction coefficients are just the forward coefficients reversed in time, thus it is necessary to only obtain  $a[k]$ . Numerically, these coefficients are found by solving the normal equation,  $\mathbf{R}\mathbf{a} = -\mathbf{r}$ , where  $\mathbf{R}$  is the autocorrelation matrix,  $\mathbf{a}$  is the vector of prediction coefficients, and  $\mathbf{r}$  is the correlation vector of the first  $m$  lags. A more

detailed description of how the prediction coefficients are determined is given in Appendix 5.B.

I have incorporated linear prediction into the lifting scheme using two separate implementations. One implementation which shall be called the *asymmetric* implementation uses only forward prediction (10), while the other implementation is *symmetric* and uses both forward and backward prediction, (10) and (11). Thus, for the asymmetric implementation,  $\mathcal{P}(c^{j+1}[n])$ , in (5) becomes:

$$\mathcal{P}_{lpa}(c^{j+1}[n]) = - \sum_{k=1}^m a[k] c^{j+1}[n-k+1] \quad (12)$$

and for the symmetric implementation we have:

$$\mathcal{P}_{lps}(c^{j+1}[n]) = - \sum_{k=1}^m a[k] c^{j+1}[n-k+1] - \sum_{k=1}^m a[k] c^{j+1}[n+k] \quad (13)$$

The asymmetric implementation results in wavelets that are asymmetrical, while symmetrical wavelets result from the symmetric implementation.

#### 5.4.2 Using Linear Prediction and Lifting to Construct Filter Bank Filters

The wavelet (highpass) filters are obtained by setting:

$$\begin{aligned} d^j &= \vec{0} && \text{for all } j \\ c^J &= \vec{0} \\ d^J[(\text{length}(d^J))/2] &= 1 \end{aligned} \quad (14)$$

and then executing the lifting scheme in the reverse direction (i.e., update, predict, merge). The lowpass filters are obtained in a similar manner except that the last line in (14) is replaced by  $c^J[(length(c^J))/2] = 1$ . Using standard notation let us call these filters  $g$  and  $h$ , respectively. These filters can then be used in the conventional filter bank implementation of the wavelet transform (refer to Chapter 2) and are perfect reconstruction in that:

$$g^* \tilde{g} + h^* \tilde{h} = \delta \quad (15)$$

where the  $*$  represents convolution and  $\delta$  is the delta function, and  $g, \tilde{g}, h, \tilde{h}$  are the components of a two-band filter bank. As stated in Chapter 2, in addition to perfect reconstruction the filter bank filters must also meet the *regularity* condition in order to qualify as wavelet filters. This condition states that the iteration of the filter bank filters as represented by  $g_j[n]$  should converge to a continuous function as  $j \rightarrow \infty$ , where

$g_{j+1}[n] = \sum_k g_j[k] h[2k-n]$  and  $g_1[n] = g[n]$  (refer to Chapter 2, Section 2.3.2). Rioul (1992; 1993) has expressed this condition in the following manner:

$$\lim_{j \rightarrow \infty} \max |g_j[n+1] - g_j[n]| = 0 \quad (16)$$

Thus, as the iteration stage,  $j$ , increases, the difference between adjacent points in the resulting function should approach zero. The filters resulting from the design method described in this section are indeed regular and thus do lead to wavelet functions as is illustrated below.

Figure 4 shows an asymmetric wavelet at several scalings along with the corresponding spectrums resulting from using the asymmetric implementation of the design method with  $x$  being a sample seismic segment (P arrival) and  $m = 6$ . Figure 5 shows the same information resulting from the symmetrical implementation using the same seismic segment and  $m = 4$ . As can be noticed, the shape of the asymmetrical wavelet deteriorates somewhat at higher scalings, while the symmetrical wavelet shows little deterioration with scale. Also, the spectrum of the symmetrical wavelet at each scale is “sharper” than that of the asymmetrical wavelet. This is due to the fact that the *degree of regularity* (how fast the iterated filters converge to continuous functions) is higher for the filters of the symmetric implementation. This is demonstrated in Figure 6 which shows a plot of  $\max |g_j[n+1] - g_j[n]|$  versus scale. As can be seen the slope of the symmetrical case curve indicates a faster rate of convergence. It should be noted that in both the asymmetrical and symmetrical cases the filters are part of a *biorthogonal* filter bank, and not an orthogonal one.

The strength of this design method is that since a model of the feature in question (using linear prediction) is being used to design the wavelets, these wavelets are such that they ought to match the particular feature,  $f$  (which is the seismic P-arrival in the examples of Figures 4 and 5). Thus, the application of these wavelets to some signal  $s$  should lead to a high degree of correlation at the point where  $f$  is present, and this will be reflected in the wavelet coefficients of  $s$ .

### 5.5 Test Application: Problem Cases from Chapter 3

In this section wavelets designed using the methods outlined in the previous section (instead of “library” wavelets) are applied to three problem cases of Chapter 3. The cases were chosen based on the degree of error in S arrival time estimates and the fact that there was nothing unusual or atypical about the location or magnitude of the events that the cases represent. Information from Chapter 3 about the S arrival time estimates (using “library” wavelets) as well as wavelet used is given below in Table 1.

**TABLE 1. S arrival estimate information (in seconds) for three test cases (Algorithm refers to that used in Chapter 3 to estimate S arrival time, Analyst refers to analyst pick, and Model refers to IASPE91 model pick). The case numbers are those from Chapter 3.**

| Case | Algorithm<br>(using “library”<br>wavelet) | Analyst | Model  | Error:<br>Algorithm<br>-Analyst | Wavelet<br>used in<br>Algorithm | Analyst<br>Comments |
|------|---|---------|--------|---------------------------------|---------------------------------|---------------------|
| 9    | 281.95                                    | 272.22  | 274.54 | 9.73                            | Daub-12                         |                     |
| 17   | 454.45                                    | 484.31  | 489.11 | -29.86                          | Vett-22                         | S poorly recorded   |
| 19   | 540.55                                    | 525.87  | 540.08 | 14.68                           | Daub-20                         |                     |

For each of these cases, the P arrival is used to construct a set of wavelet filters using the methods described in the previous sections. These wavelets are then used in the Algorithm (Part 2) of Chapter 3, the result of which is a S locator function,  $C_t$ . A block diagram of these steps is presented in Figure 7. Remember from Chapter 3 that  $C_t$  is used to locate the S arrival. Figures 8, 9, and 10 show the results of this application. In each figure is the P segment that is modeled to find the linear prediction coefficients, the resulting filters, and  $C_t$ . Also, in Table 2 are the quantitative results of this application.

**TABLE 2. S arrival estimate information (in seconds) for three test cases using wavelets designed by methods of Sections 5.3 and 5.4 (refer to Figures 8, 9, and 10).**

| Case | Algorithm<br>(using designed<br>wavelet) | Analyst | Error:<br>Algorithm<br>-Analyst |
|------|--|---------|---------------------------------|
| 9    | 276.70                                   | 272.22  | 4.48                            |
| 17   | 490.50                                   | 484.31  | 6.19                            |
| 19   | 540.75                                   | 525.87  | 14.88                           |

It can be seen from the results in Table 2 that for two of the cases (9 and 17) the S arrival estimate accuracy showed a very significant improvement, while for the third case the accuracy remained about the same. Case 17 should be particularly noted, in that the use of a “library” wavelet resulted in an unacceptable pick error, while the use of a designed wavelet resulted in a good pick. Overall, it can be stated that for this application the use of designed wavelets (based on the P-arrival information) had a positive impact on the feature (S arrival) location results.

## 5.6 Discussion and Conclusions

In this chapter a new wavelet design method has been introduced that uses *linear prediction* in combination with *lifting*. The important advantage of the method is that information about the data being analyzed is used to design wavelets adapted to the data. These *adaptive* wavelets are constructed purely in the time domain, which leads to a more robust design method than conventional Fourier based methods. This design method was applied to a few problem cases from Chapter 3 with very promising results (major reductions in S location estimate error for two out of three cases). These seismic results not only show

that the wavelet design method is effective, but also further reinforce the idea brought up in Chapter 3 regarding the use of P arrival information to help locate the S arrival.

Of course, the design method introduced here can be applied to seismic signals, as well as to other types of signals. The steps in the design method that has been presented do not depend on the *type* of signal. In fact linear prediction has had applications in a wide variety of areas other than geophysics including neurophysics, speech processing, and even economics. Thus a *linear prediction based wavelet design method* should find applications in these areas as well.

## Appendix 5.A Fourier Based Wavelet Design Method

### *Design Recipe*

1. Using the Parks-McClellan algorithm, construct a finite length linear phase half-band filter,  $f[n]$ , with length  $2N$  where  $N$  is the desired length of the analysis lowpass filter,  $h[n]$ . The term half-band signifies that the amplitude response,  $F(\omega)$ , has a symmetry around the half-band frequency,  $\omega = \pi/2$ . This means that the following constraint must be met:  $\omega_p + \omega_s = \pi$ , where  $\omega_p$  is called the passband frequency and  $\omega_s$  is called the stopband frequency.



2. The next step is to shift the amplitude response  $F(\omega)$  by a factor  $\delta$  so that the resulting response,  $P(\omega)$ , is greater than zero for all  $\omega$ . Thus:

$$P(\omega) = F(\omega) + \delta \quad (17)$$

3. The last step is to find the spectral factor,  $H$ , of  $P$ . This is the most complicated part of the method and is implemented in the following manner:

i). Multiply  $p[n]$  by a decay factor,  $\beta$ , i.e.:

$$a[n] = \beta^{-n} p[n] \quad (18)$$

where  $n = 0, 1, 2, \dots, 2N$ . Vaidyanathan (1993) has found experimentally that 1.02 is a good value to use for the decay factor.

ii). Take the *cepstrum* of  $a[n]$  to get  $\hat{a}[n]$ . Cepstrum analysis is defined and discussed thoroughly in Chapter 12 of Oppenheim and Schaffer (1989), and a program that is commonly used to calculate the cepstrum of a function is written by Tribolet and Quatieri (1979).

iii). Add the causal part of  $\hat{a}[n]$  to the folded anti-causal part of  $\hat{a}[n]$  and divide by two, with the result being  $\hat{h}[n]$ .

iv). The following two relations are used to get the desired analysis lowpass filter,  $h[n]$ :

$$\begin{aligned}
 h[0] &= \exp(\hat{h}[0]) \\
 h[n] &= \sum_{k=1}^n \frac{k}{n} \hat{h}[k] h[n-k]
 \end{aligned}
 \tag{19}$$

The code used to execute the above steps is given below.

*Code based on design recipe*

The code below is the pre-cepstrum part of the Fourier based design code. This program takes an initial symmetric real coefficient filter produced by Parks-McClellan and then 1) computes the frequency response of the filter, 2) shifts the frequency response so that it is strictly greater than zero, 3) finds the impulse response corresponding to this new frequency response and then 4) multiplies this new impulse response by a decay factor.

```

/* Program input: filt
Program output: H_omega (shifted freq. response)
               h_n   (new impulse response)
               g_n   (h_n multiplied by decay factor)
Subroutines called: fft.c */

#include <stdio.h>
#include <math.h>
#define LENFFT 1024

main()
{
FILE *fopen(), *fp, *fp2;
float min,max;
int i,j,Nx,Nxdiv,base2Nx;
float x[LENFFT],imx[LENFFT],xtemp[100],h[100],g[100],negi,rho;

/* initializing */
min = 9999999.;
max = -9999999.;
for(j=0;j<LENFFT;j++) x[j] = 0.0;

/* read filter resulting from Parks-McClellan*/
fp = fopen("filt", "r");
Nx = 0;
while (fscanf(fp, "%f", &xtemp[Nx])!= EOF) Nx++;
fclose(fp);
fprintf(stderr, "PRECEPS: Read%d input data points from ascii file \n", Nx);

fp = fopen("optlen", "w");

```

```

fprintf(fp, "%d\n", Nx);
fclose(fp);

/* zero pad input filter */
for (i=0; i<((Nx+1)/2); i++) x[i] = xtemp[i+((Nx-1)/2)];
for (i=LENFFT-((Nx-1)/2); i<LENFFT; i++) x[i] = xtemp[i-(LENFFT-((Nx-1)/2))];

for (i=0; i<LENFFT; i++) imx[i] = 0.0;
/* call forward fft */
Nxdiv = LENFFT;
base2Nx = 0;
while (Nxdiv!= 1){
    Nxdiv = Nxdiv/2;
    base2Nx += 1;
}
fft(x,imx,LENFFT,base2Nx,0);

/* find min of freq. response and shift response
   by 1.1 times min */
for (i=0; i<LENFFT; i++){
    if (x[i] < min) min = x[i];
}

printf("The min equals:%f\n", min);

for (i=0; i<LENFFT; i++) x[i] += fabs(min)*1.1;
fp = fopen("H_omega", "w");
for(i=0; i<LENFFT; i++) fprintf(fp, "%f\n", x[i]);
fclose(fp);

/* call inverse fft to find new impulse response */
fft(x,imx,LENFFT,base2Nx,1);

for (i=0; i<((Nx+1)/2); i++) h[i+((Nx-1)/2)] = x[i];
for (i=LENFFT-((Nx-1)/2); i<LENFFT; i++) h[i-(LENFFT-((Nx-1)/2))] = x[i];

/* multiply new impulse response by decay factor */
rho = 1.02;
fp = fopen("h_n", "w");
fp2 = fopen("g_n", "w");
for (i=0; i<Nx; i++){
    fprintf(fp, "%f\n", h[i]);
    negi = -i;
    g[i] = pow(rho,negi)*h[i];
    fprintf(fp2, "%f\n", g[i]);
}
fclose(fp);
fclose(fp2);
}

```

The results of the above code are input into the cepstrum code of Tribolet and Quatieri, (1979). The code below is the post-cepstrum part of the Fourier based design code. This

program takes the output of the cepstrum program and produces a minimum phase filter,

$h[n]$  .

```

/*Program input: ghat_n
  Program output: h0_n
  Subroutines called: none */

#include <stdio.h>
#include <math.h>
#define LEN 10000

main()
{
  FILE *fopen(), *fp;
  int i,j,k,n,Nx,Nghat;
  float ghat[LEN],xhat[LEN],hnot[18],realk,realn,sum;

  /* read output of cepstrum program */
  fp = fopen("ghat_n", "r");
  Nghat = 0;
  while (fscanf(fp, "%f", &ghat[Nghat])!= EOF) Nghat++;
  fclose(fp);
  fprintf(stderr, "POSTCEPS: Read%d input data points from ascii file \n", Nghat);

  /* fold anticausal part of ghat and add to causal part */
  for (i=0; i<512; i++) xhat[i] = 0.5*(ghat[i] + ghat[Nghat-i]);

  fp = fopen("optlen", "r");
  fscanf(fp, "%d", &Nx);
  fclose(fp);

  /* finally, find spectral factor hnot using xhat */
  hnot[0] = exp(xhat[0]);
  for (n=1; n<((Nx+1)/2); n++){
    for (k=1; k<=n; k++){
      realn = n;
      realk = k;
      hnot[n] += (realk/realn)*xhat[k]*hnot[n-k];
    }
  }

  /* normalize filter */
  sum = 0.0;
  for (k=0; k<((Nx+1)/2); k++) sum += hnot[k]*hnot[k];
  for (k=0; k<((Nx+1)/2); k++) hnot[k] = hnot[k]/sqrt(sum);

  fp = fopen("h0_n", "w");
  for(i=0; i<((Nx+1)/2); i++) fprintf(fp, "%f\n", hnot[i]);
  fclose(fp);
}

```

## Appendix 5.B Calculation of the Linear Prediction Coefficients

(Makhoul, 1975)

The prediction coefficients,  $a[k]$ , are found by using the method of least squares to model  $x$ . The coefficients are obtained by minimizing the total squared error between  $x$  and  $\hat{x}$  with respect to each of the coefficients. Using (10), the error between the actual and predicted values is:

$$e[n] = x[n] - \hat{x}[n] = x[n] + \sum_{k=1}^m a[k] x[n-k] \quad (20)$$

Therefore, the total squared error is:

$$E = \sum_n e^2[n] = \sum_n \left( x[n] + \sum_{k=1}^m a[k] x[n-k] \right)^2 \quad (21)$$

This error can be minimized by setting:

$$\frac{\partial E}{\partial a[i]} = 0, \quad 1 \leq i \leq m \quad (22)$$

By applying (22) to (21), the following set of equations are found:

$$\sum_{k=1}^m a[k] \sum_n x[n-k] x[n-i] = - \sum_n x[n] x[n-i], \quad 1 \leq i \leq m \quad (23)$$

The above relation represents a set of  $m$  equations with  $m$  unknowns and can be solved for the coefficients,  $a[k]$ . An elegant way to solve this system is to use what is termed the *autocorrelation* method, where the autocorrelation function of  $x[n]$  is defined as:

$$R[i] = \sum_n x[n] x[n+i] \quad (24)$$

which is an even function of  $i$ , i.e.  $R[-i] = R[i]$ . Therefore, (23) can be reduced to:

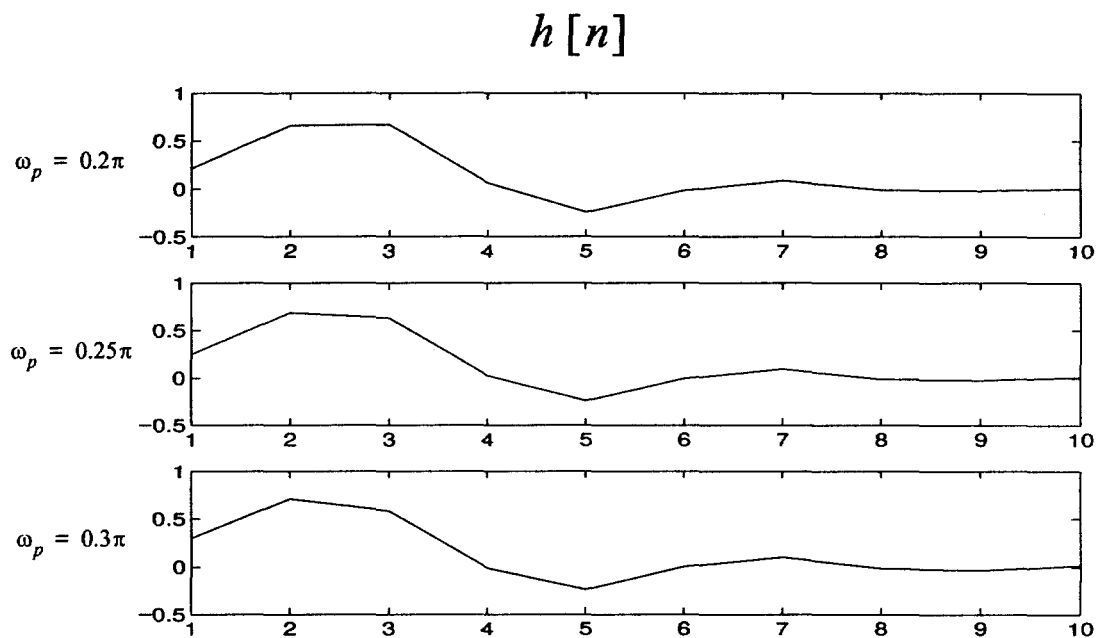
$$\sum_{k=1}^m a[k] R[i-k] = -R[i], \quad 1 \leq i \leq m \quad (25)$$

In matrix form (25) can be represented as:

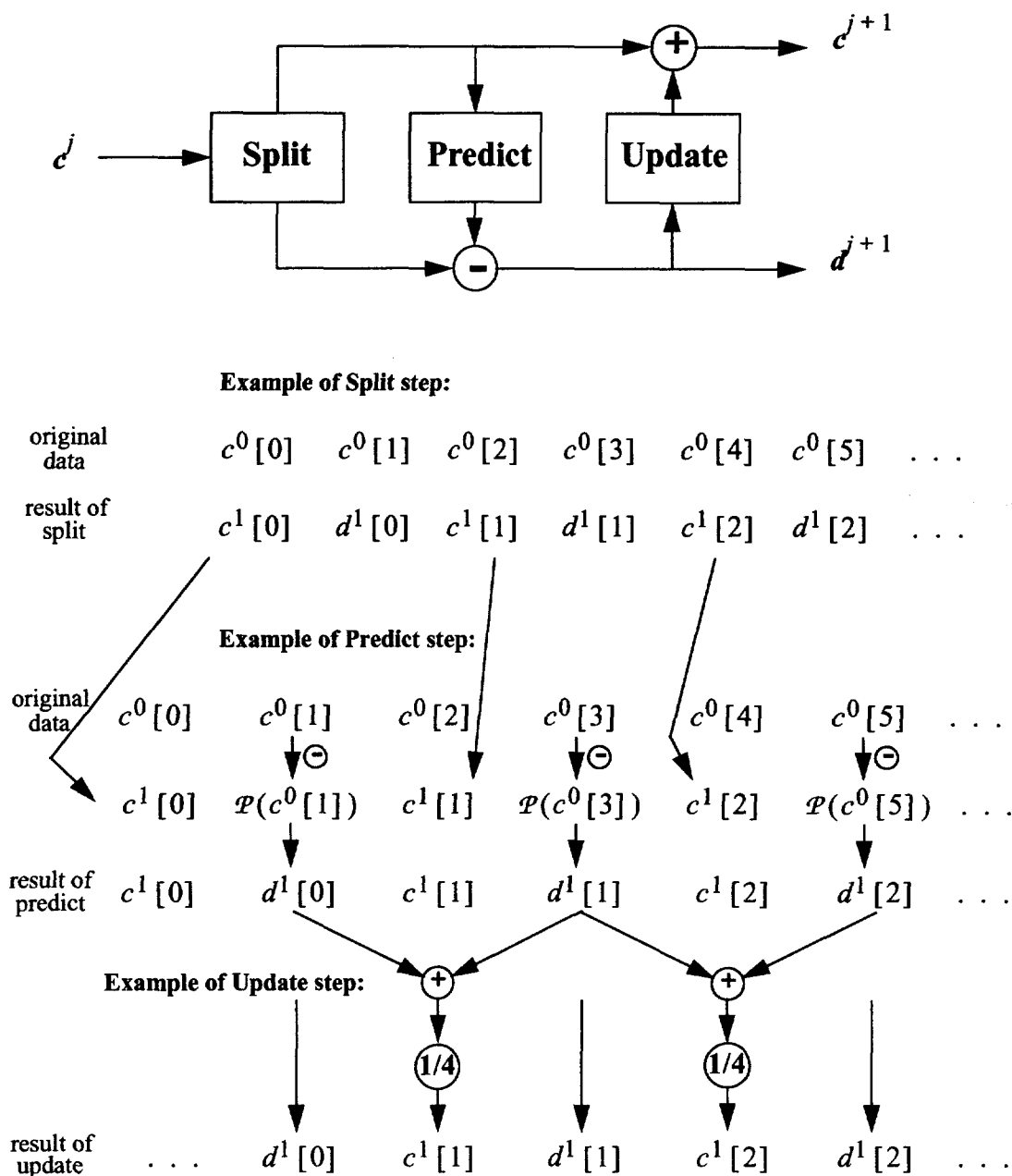
$$\begin{bmatrix} R[0] & R[1] & R[2] & \dots & R[m-1] \\ R[1] & R[0] & R[1] & \dots & R[m-2] \\ R[2] & R[1] & R[0] & \dots & R[m-3] \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ R[m-1] & R[m-2] & R[m-3] & \dots & R[0] \end{bmatrix} \begin{bmatrix} a[1] \\ a[2] \\ a[3] \\ \vdots \\ a[m] \end{bmatrix} = - \begin{bmatrix} R[1] \\ R[2] \\ R[3] \\ \vdots \\ R[m] \end{bmatrix} \quad (26)$$

The matrix on the left is called the *autocorrelation matrix*. The values for  $a[k]$  can be found by using methods such as Gaussian elimination, matrix inversion, etc.

**FIGURE 1. Three filters designed using the Fourier based design method outlined in Section 5.2. All three use a value of  $N = 10$ . The values for the passband frequencies are varied and are  $\omega_p = 0.2\pi$ ,  $\omega_p = 0.25\pi$ , and  $\omega_p = 0.3\pi$ .**

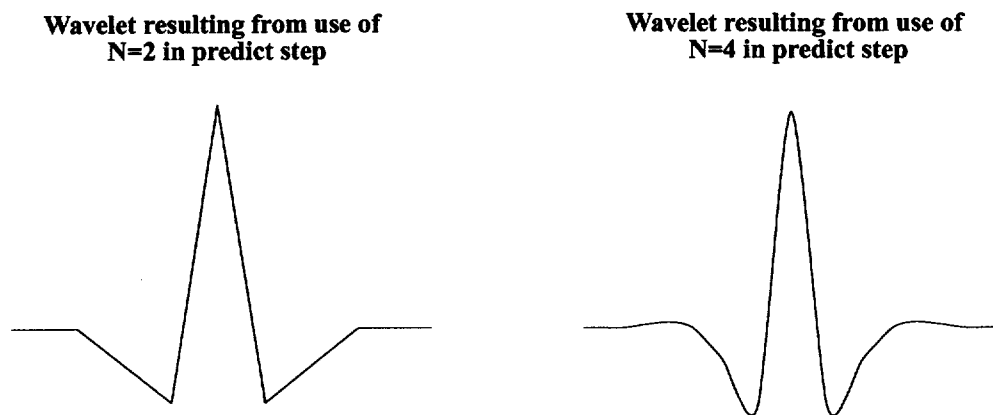


**FIGURE 2.** Diagram of lifting scheme with  $c^j$  and  $d^j$  being the scaling coefficients and wavelet coefficients respectively, just as in Chapter 2. Below the block diagram are examples of each of the steps. The scaling coefficients of the split step are passed on to the predict step and are used to calculate “further decorrelated” wavelet coefficients. The wavelet coefficients of the predict step are then passed on to the update step and are used to “lift” the scaling coefficients.

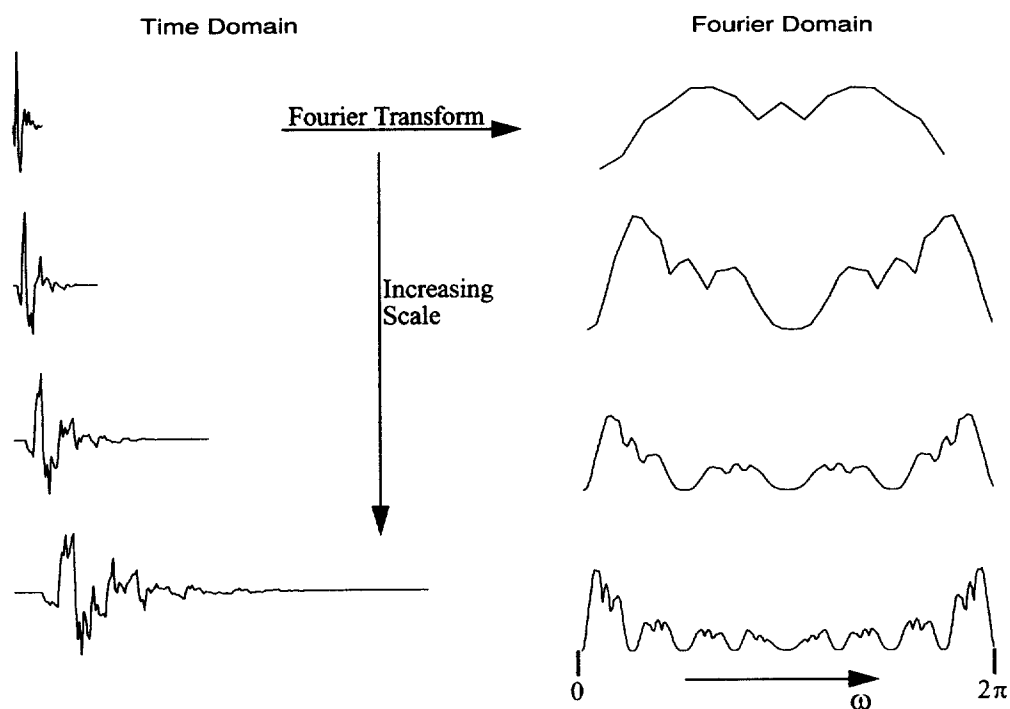




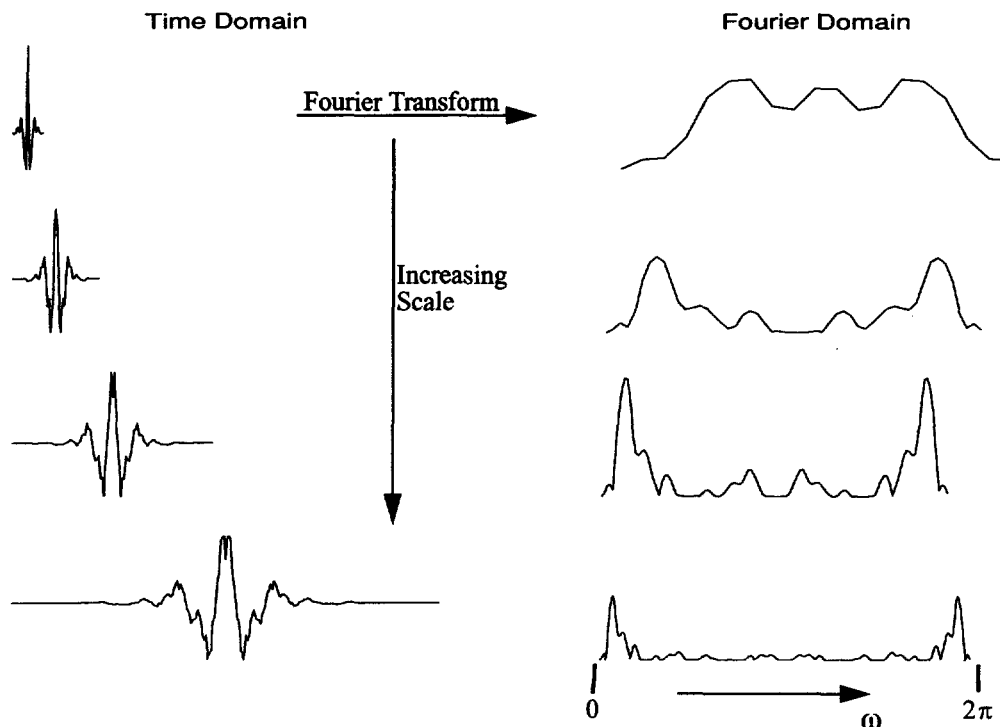
**FIGURE 3.** Two wavelets resulting from lifting scheme using  $N = 2$  and  $N = 4$  respectively, in the prediction step.



**FIGURE 4.** Wavelets at different scalings resulting from asymmetrical implementation of method outlined in Section 5.4. Next to each wavelet is its Fourier spectrum. Notice how the wavelet shape degrades somewhat with scale.



**FIGURE 5.** Wavelets at different scalings resulting from symmetrical implementation of method outlined in Section 5.4. Next to each wavelet is its Fourier spectrum. Notice how the wavelets do not deteriorate as much as the wavelets in Figure 4 and how the spectrum is “sharper”.



**FIGURE 6.** Plots of  $\max |g_j[n+1] - g_j[n]|$  versus scale (or iteration) for asymmetric and symmetric cases. The plots indicate that the symmetric wavelets have a higher degree of regularity.

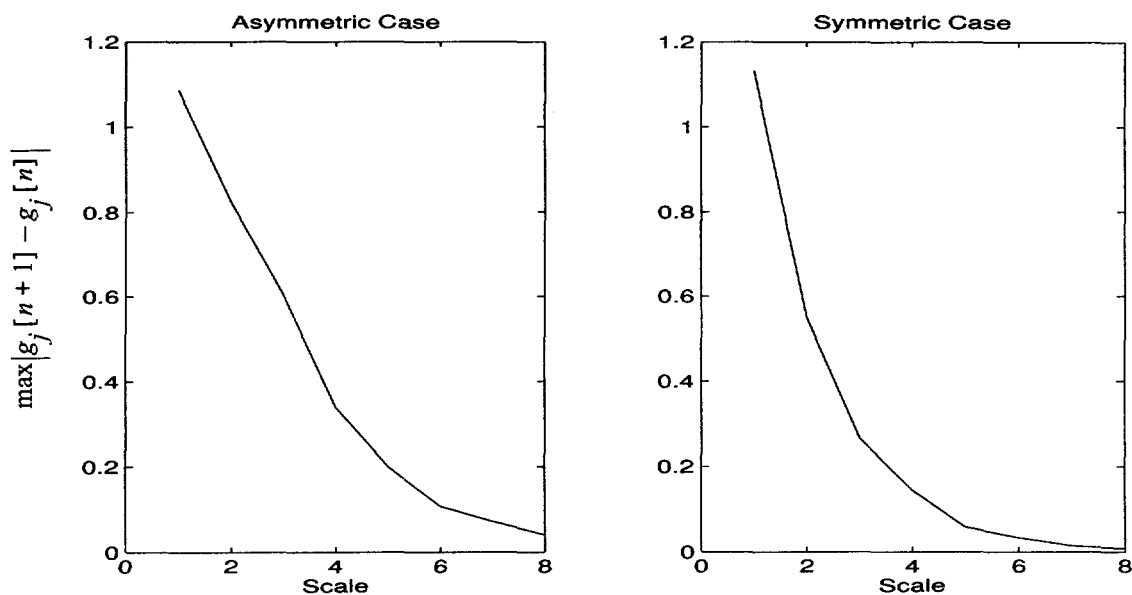


FIGURE 7. Block diagram of seismic test application. The input,  $s$ , is the seismic signal. The filters resulting from the design method are  $h, \tilde{h}, g, \tilde{g}$ , and the output is the S arrival locator function,  $C_t$ .

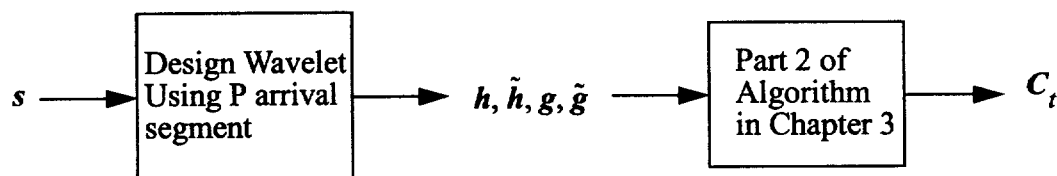
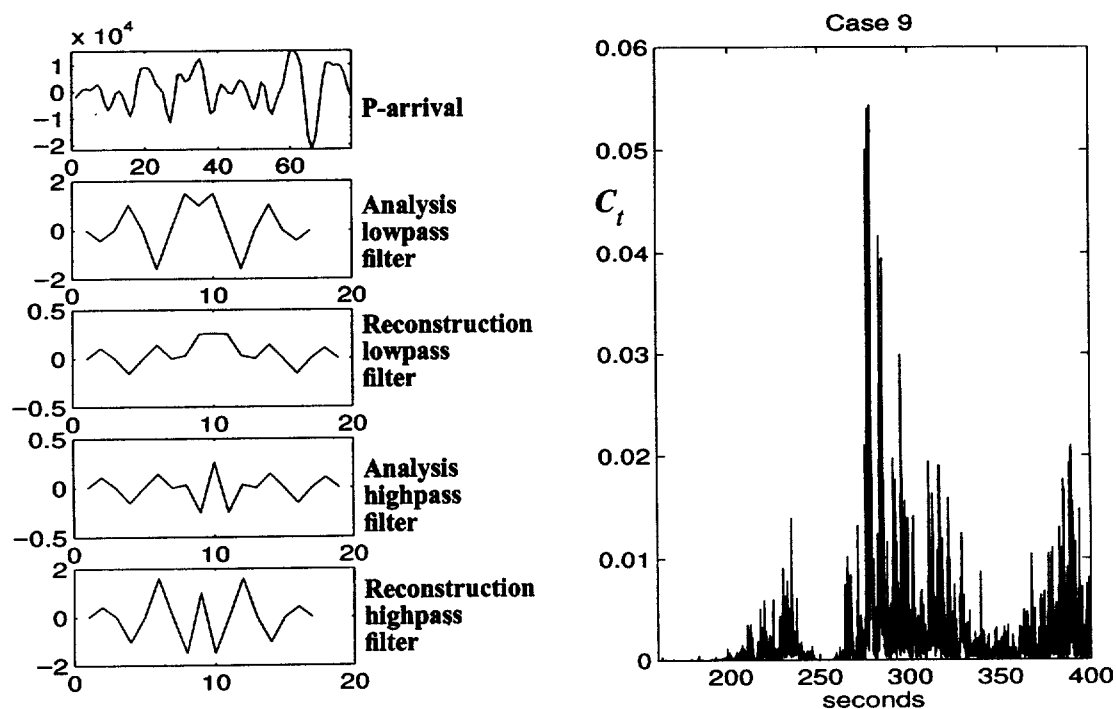
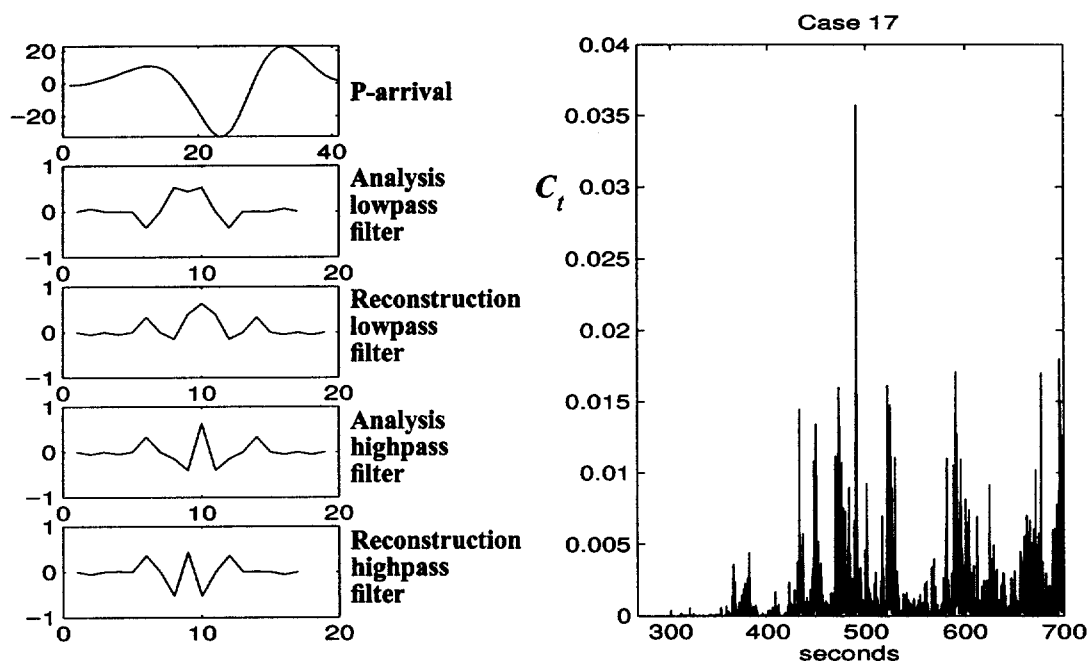


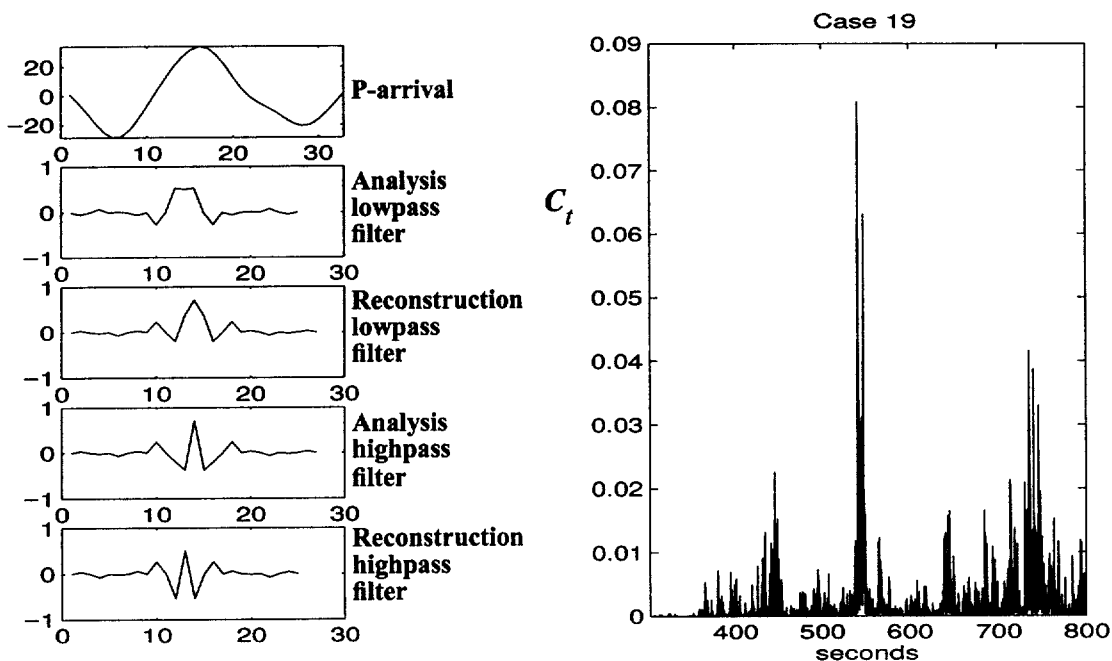
FIGURE 8. Case 9 results. Shown below are the P arrival used in the design method, the resulting filters, and  $C_t$  resulting from using the designed filters in Part 2 of the Algorithm of Chapter 3.



**FIGURE 9. Case 17 results.** Shown below are the P arrival used in the design method, the resulting filters, and  $C_t$  resulting from using the designed filters in Part 2 of the Algorithm of Chapter 3.



**FIGURE 10. Case 19 results.** Shown below are the P arrival used in the design method, the resulting filters, and  $C_t$  resulting from using the designed filters in Part 2 of the Algorithm of Chapter 3.



### 6.1 Conclusions

In this dissertation I have studied the feature identification and compression properties of the wavelet transform with the application signals being the seismogram and the electrocardiogram. The common issues among both of these applications included how to optimally process the information present in the wavelet scales, and how to choose the basis function (wavelet). In Chapter 3, it was shown that by analyzing information *across* scales important seismic arrivals (P and S phase) can be identified with good accuracy. This is due to the fact that significant features in a signal show up strongly in several wavelet scales while weaker (or random) features are present in only one or two scales. It was also shown that wavelet choice was an important parameter in terms of S phase arrival detection. For many of the seismic test cases using one of the wavelets from a library of wavelets led to an accurate pick. The wavelet used was based on the amplitude of the resulting S locator function (refer to Chapter 3). In addition it was shown that there

was a correlation between the “best” wavelet and the P arrival shape. This makes sense as there should be a degree of correlation (depending on the particular event) between the P arrival shape and the S arrival shape. This idea was taken advantage of further by first using the P arrival shape constructing *pseudo*-wavelets (no perfect reconstruction properties) and then in Chapter 5 by designing actual wavelets based on the P arrival shape. For events where the S phase arrival was difficult to locate it was shown that by using this design method an accurate S arrival pick resulted.

In Chapter 4, I presented a compression method using the wavelet transform and vector quantization. Vector quantization was applied to the scales instead of the raw signal. As predicted theoretically, higher scales required the use of smaller coding vectors due to the decreased amount of correlation among the coefficients in these scales. Also it was shown that higher order *orthogonal* wavelets led to the best compression results (low bit-per-sample rate with accurate reconstruction). This is due to the fact that these wavelets provide a nonredundant representation of the signal.

I have found that the main advantage of the wavelet transform is that the wavelet decomposition essentially breaks up a signal so that it can be viewed at different scalings. This decomposition is controlled by the wavelet used. The effect is to provide a different perspective and thus lead to new insights. These new insights have aided me in the development of the methods that I have presented in this dissertation.

## 6.2 Future Direction

In research it is rare for there to be complete closure. There are almost always more questions after one phase of research than before. Some of the questions may get answered, but many more inevitably pop up. This has definitely been the case for this thesis research. There are several directions in which future work can proceed. Some of these directions involve the refinement of the methods presented here (some of which have already been mentioned in the chapters). Though the test data set of Chapter 3 was sufficient (it took many months to put together the data set that was used), more representative quantitative results could be obtained if a larger set of events were available. In addition, the polarization based methods used to locate the P arrival could be extended to *long period* seismograms as another way to locate the S phase arrival. As for the compression method of Chapter 4, there is room for improvement with regards to the application of vector quantization, which include improved clustering and search routines. Many of the other refinements center on the issue of wavelet choice which has been partly addressed in Chapter 5 (wavelet design). For the seismic feature identification problem several questions remain regarding how to best define the P arrival shape when designing a wavelet that is based on it. The methods of Chapter 5 can be refined by trying different predictors (instead of *linear prediction*) in the predict step of the design scheme and possibly altering the update step.

There are also *new* directions in which this research can proceed. These include applications of the methods in this thesis to other classes of signals such as speech signals and

images, and to the development of other types of applications, such as noise removal (a type of compression) and pattern recognition (feature identification). The wavelet design methods of Chapter 5 offer a great deal of potential application to other types of signals because of the use of linear prediction. Given that there is so much potential for future work, I wrap up with a refrain from a poem by Robert Frost<sup>1</sup>:

The woods are lovely dark and deep,  
But I have promises to keep,  
*And miles to go before I sleep,*  
*And miles to go before I sleep...*

---

1. From *Stopping by Woods on a Snowy Evening*



# Bibliography

Ahmed, N., T. Natarajan, and K. Rao (1974). "Discrete cosine transform," *IEEE Transactions on Computers* **23**, 88-93.

Akansu, A. (1992). *Multiresolution Signal Decomposition: Transforms, Subbands, and Wavelets*. Boston, MA: Academic Press.

Aki, K., and P. Richards (1980). *Quantitative Seismology - Theory and Methods*, W. H. Freeman and Company, San Francisco.

Allen, J., and L. Rabiner (1977). "A unified approach to short-time fourier analysis and synthesis," *Proceedings of the IEEE* **65**, 1558-1564.

Al-Nashash, H. (1994). "ECG data compression using adaptive Fourier coefficients estimation," *Medical Engineering and Physics* **16**, 62-66.

Anant, K., F. Dowla, and G. Rodrigue (1994). "Detection of the electrocardiogram P-wave using wavelet analysis," *Proceedings of the SPIE (Wavelet Applications, Orlando, FL, USA, 5-8 April 1994)* **2242**, 744-749.

Cetin, A., H. Koymen, and M. Aydin (1993). "Multichannel ECG data compression by multirate signal processing and transform domain coding techniques," *IEEE Transactions on Biomedical Engineering* **40**, 495-499.

Chakraborty, A., and D. Okaya (1995). "Frequency-time decomposition of seismic data using wavelet based methods," *Geophysics* **60**, 1906-1916.

Chen, J., S. Itoh, and T. Hashimoto (1993). "ECG data compression by using wavelet transform," *IEICE Transactions on Information and Systems* **E76-D**, 1454-1461.

Cohen, A., I. Daubechies, and J.-C. Feauveau (1992). "Biorthogonal bases of compactly supported wavelets," *Communications on Pure and Applied Mathematics* **45**, 485-560.

Crowe, J., N. Gibson, M. Woolfson, and M. Somekh (1992). "Wavelet transform as a potential tool for ECG analysis and compression," *Journal of Biomedical Engineering* **14**, 268-272.

Daubechies, I. (1992). *Ten Lectures on Wavelets*. Philadelphia, PA: SIAM.

Desarte, P., B. Marc, and D. Slock (1992). "Signal-adapted multiresolution transform for image coding," *IEEE Transactions on Information Theory* **38**, 897-904.

DeVore, R., B. Jawerth, and B. Lucier (1992). "Image compression through wavelet transform coding," *IEEE Transactions on Information Theory* **38**, 719-746.

Donoho, D., and I. Johnstone (1994). "Ideal spatial adaptation via wavelet shrinkage," *Biometrika* **81**, 424-544.

Donoho, D., and I. Johnstone (1995). "Adapting to unknown smoothness via wavelet shrinkage," *Journal of the American Statistical Association* **90**, 1200-1224.

Donoho, D. (1995). "De-noising by soft-thresholding," *IEEE Transactions on Information Theory* **41**, 613-627.

Duda, R., and P. Hart (1973). *Pattern Classification and Scene Analysis*. New York: Wiley.

Earle, P. S., and P. M. Shearer (1994). "Characterization of global seismograms using an automatic-picking algorithm," *Bulletin of the Seismological Society of America* **84**, 366-376.

Evangelista, G. (1993). "Pitch-synchronous wavelet representations of speech and music signals," *IEEE Transactions on Signal Processing* **41**, 3313-3330.

Fernandez, G., S. Periaswamy, and W. Sweldens (1996). "Liftpack: a software package for wavelet transforms using lifting," *Proceedings of the SPIE (Wavelet Applications in Signal and Image processing IV)* **2825**, 396-408.

Fowler, R. A., B. J. Kotick, and R. D. Elliott (1967). "Polarization analysis of natural and artificially induced geomagnetic micropulsations," *Journal of Geophysical Research* **72**, 2871-2883.

Frisch, M., and H. Messer (1992). "The use of the wavelet transform in the detection of an unknown transient signal," *IEEE Transactions on Information Theory* **38**, 892-897.

Gabor, D. (1946). "Theory of communication," *Journal IEE* **93**, 429-441.

Gersho, A., and V. Cuperman (1983) "Vector quantization: a pattern-matching technique for speech coding," *IEEE Communications Magazine* **21**, 15-21.

Gersho, A., and R. Gray (1992). *Vector Quantization and Signal Compression*. Boston, MA: Kluwer Academic Publishers.

Goupillaud, P., A. Grossmann, and J. Morlet (1984). "Cycle-octave and related transforms in seismic signal analysis," *Geoexploration* **23**, 85-102.

Gray, R. (1984) "Vector quantization," *IEEE ASSP Magazine* **1**, 4-28.

Grossmann, A., and J. Morlet (1984). "Decomposition of Hardy functions into square integrable wavelets of constant shape," *SIAM Journal of Math Analysis* **15**, 723-736.

Gruener, R. (1996). "Body electricity: a first year colloquium (part II: the electrocardiogram) (195A course notes)," University of Arizona, College of Medicine, Department of Physiology, <http://www.physiol.arizona.edu/CELL/Instruct/BodyElect/EKG.html>.

Jalaleddine, J., C. Hutchens, R. Strattan, and W. Coberly (1990). "ECG data compression techniques-a unified approach," *IEEE Transactions on Biomedical Engineering* **37**, 329-343.

Jarpe, S., and F. Dowla (1991). "Performance of high-frequency three-component stations for azimuth estimation from regional seismic phases," *Bulletin of the Seismological Society of America* **81**, 987-999.

Kanasewich, E. (1981). *Time Sequence Analysis in Geophysics*. Edmonton, Alberta, Canada: The University of Alberta Press.

Katz, A. (1992). *Physiology of the Heart*, New York: Raven Press.

Kroland-Martinet, R. (1988). "The wavelet transform for analysis, synthesis, and processing of speech and music sounds," *Computer Music Journal* **12**, 11-20.

Li, C, C. Zheng, and C. Tai (1995). "Detection of ECG characteristic points using wavelet transforms," *IEEE Transactions on Biomedical Engineering* **42**, 21-28.

Lookabaugh, T., E. Riskin, P. Chou, and R. Gray (1993). "Variable rate vector quantization for speech, image, and video compression," *IEEE Transactions on Communications* **41**, 186-199.

Lynch, T. (1985). *Data Compression - Techniques and Applications*, New York: Van Nostrand Reinhold.

Magotra, N., N. Ahmed, and E. Chael (1989) "Single-station seismic event detection

and location," *IEEE Transactions on Geoscience and Remote Sensing* **27**, 15-23.

Makhoul, J. (1975). "Linear prediction: a tutorial review," *Proceedings of the IEEE* **63**, 561-580.

Mallat, S. (1989). "A theory for multiresolution signal decomposition: the wavelet representation," *IEEE Transactions on Pattern Recognition and Machine Intelligence* **11**, 674-693.

Mallat, S., and S. Zhong (1992). "Characterization of signals from multiscale edges," *IEEE Transactions on Pattern Analysis and Machine Intelligence* **14**, 710-732.

Marr, D. (1982). *Vision*, Freeman, San Francisco.

Means, J. D. (1972). "Use of the three-dimensional covariance matrix in analyzing the polarization properties of plane waves," *Journal of Geophysical Research* **77**, 5551-5559.

Michaelis, M., S. Perz, C. Black, and G. Sommer (1993). "Detection and classification of P waves using Gabor wavelets," *Proceedings: Computers in Cardiology Conference*, 531-534, IEEE Computer Society Press, Los Alamitos, California.

MIT-BIH, (1992). *MIT-BIH Arrhythmia Database Directory, Third Edition*, Harvard University - Massachusetts Institute of Technology Division of Health Sciences and Technology.

Morlet, J., G. Arens, E. Fourgeau, and D. Giard (1982). "Wave propagation and sampling theory: complex signal and scattering in multilayered media," *Geophysics* **47**, 203-221.

Morlet, D., F. Peyrin, P. Desseigne, P. Touboul, and P. Rubel (1991). "Time-scale anal-

ysis of high-resolution signal-averaged surface ECG using wavelet transformation,” *Proceedings: Computers in Cardiology Conference*, 393-396, IEEE Computer Society Press, Los Alamitos, California.

Oppenheim, A., and R. Schaffer (1989). *Discrete-time Signal Processing*. Englewood Cliffs, NJ: Prentice-Hall.

Park, J. (1987). “Frequency dependent polarization analysis of high-frequency seismograms,” *Journal of Geophysical Research* **92**, 12664-12674.

Parks, T., and J. McClellan (1972). “A program for the design of linear phase finite impulse response filters,” *IEEE Transactions on Audio Electroacoustics* **AU-20**, 195-199.

Passman, J. (1976). *The EKG - basic techniques for interpretation: a practical guide for interpreting and analyzing the electrocardiogram*, McGraw-Hill, New York.

Plonsey R., and R. Barr (1988). *Bioelectricity: A Quantitative Approach*, Ch. 9, Plenum Press, New York.

Ramchandran, K., and M. Vetterli (1993). “Best wavelet packet bases in a rate-distortion sense,” *IEEE Transactions on Image Processing* **2**, 160-175.

Ricker, N. (1940). “The form and nature of seismic waves and the structure of seismograms,” *Geophysics* **5**, 348-366.

Ricker, N. (1953). “The form and laws of propagation of seismic wavelets,” *Geophysics* **18**, 10-40.

Riggs, T., B. Isenstein, and C. Thomas (1979). “Spectral analysis of the normal electrocardiogram in children and adults,” *Journal of Electrocardiology* **12**, 377-379.

Rioul, O. (1992). "Simple regularity criteria for subdivision schemes," *SIAM Journal of Mathematical Analysis* **23**, 1544-1576.

Rioul, O. (1993). "Regular wavelets: a discrete-time approach," *IEEE Transactions on Signal Processing* **41**, 3572-3579.

Rioul, O., and P. Duhamel (1992). "Fast algorithms for discrete and continuous wavelet transforms," *IEEE Transactions on Information Theory* **38**, 569-586.

Rioul O., and M. Vetterli (1991). "Wavelets and signal processing," *IEEE Signal Processing Magazine*, October, **8**, 14-38.

Roberts, R., A. Christoffersson, and F. Cassidy (1989). "Real-time event detection, phase identification and source location estimation using single station three-component seismic data," *Geophysical Journal* **97**, 471-480.

Selim, S., and M. Ismail (1984). "K-means-type algorithms: a generalized convergence theorem and characterization of local optimality," *IEEE Transactions on Pattern Analysis and Machine Intelligence* **6**, 81-87.

Shahein, H., and H. Abbas (1994). "ECG data compression via cubic-splines and scan-along polygonal approximation," *Signal Processing*, **35**, 269-283.

Shannon, C. (1959). "Coding theorems for a discrete source with a fidelity criterion," *IRE National Convention Records* **4**, 142-163.

Shridhar M., and M. Stevens (1979) "Analysis of ECG data, for data compression," *International Journal of Bio-Medical Computing*, **10**, 113-128.

Starck, J-L., and A. Bijaoui (1994). "Filtering and deconvolution by the wavelet transform," *Signal Processing*, **35**, 195-211.

Strobach, P. (1990). *Linear Prediction Theory: A Mathematical Basis for Adaptive Systems*. Berlin: Springer-Verlag.

Sweldens, W. (1995). "The lifting scheme: a new philosophy in biorthogonal wavelet constructions," *Proceedings of the SPIE (Wavelet Applications in Signal and Image processing III, San Diego, CA, USA, 12-14 July 1995)* **2569**, 68-79.

Sweldens, W. (1996a). "The lifting scheme: a custom-design construction of biorthogonal wavelets," *Journal of Applied and Computational Harmonic Analysis* **3**, 186-200.

Sweldens, W. (1996b). "Wavelets and the lifting scheme: a 5 minute tour," *Zeitschrift Fur Angewandte Mathematik und Mechanik* **76**, S2: 41-44.

Szu, H, X-Y. Yang, B. Telfer, and Y. Sheng (1993). "Neural network and wavelet transform for scale-invariant data classification," *Physical Review E*, **48**, 1497-1501.

Tewfik, A., and M. Kim (1992). "Correlation structure of the discrete wavelet coefficients of fractional brownian motion," *IEEE Transactions on Information Theory* **38**, 904-909.

Tewfik, A., D. Sinha, and P. Jorgensen (1992). "On the optimal choice of a wavelet for signal representation," *IEEE Transactions on Information Theory* **38**, 747-765.

Thakor, N., Y. Sun, H. Rix, and P. Caminal (1993). "Multiwave: A wavelet-based ECG data compression algorithm," *IEICE Transactions on Information and Systems* **E76-D**, 1462-1469.

Trejo, L., and M. Shensa (1993). "Linear and neural network models for predicting human signal detection performance from event-related potentials: a comparison of the wavelet transform with other feature extraction methods," *Proceedings of the Fifth Work-*



*shop on Neural Networks: Academic/Industrial/NASA/Defense, SPIE, 2204*, 153-161.

Tribolet, J., and T. Quatieri (1979). "Computation of the complex cepstrum," In *Programs for Digital Signal Processing*, IEEE Press.

Vaidyanathan, P. P. (1987). "Quadrature mirror filter banks, m-band extensions and perfect-reconstruction techniques," *IEEE ASSP Magazine* **4**, 4-20.

Vaidyanathan, P. P. (1993). *Multirate Systems and Filter Banks*. Englewood Cliffs, NJ: Prentice-Hall.

Vetterli, M., and C. Herley (1992). "Wavelets and filter banks: theory and design," *IEEE Transactions on Signal Processing* **40**, 2207-2232.

Vetterli, M., and J. Kovacevic (1995). *Wavelets and Subband Coding*. Englewood Cliffs, NJ: Prentice Hall.

Welch, T. (1984). "A technique for high-performance data compression," *Computer*, **17**, 8-19.

Wiggins, R. (1977) "Minimum entropy deconvolution," *Proceedings International Symposium on Computer Aided Seismic Analysis and Discrimination, Falmouth, Mass. IEEE Computer Society*, 7A-14.

Wiggins, R. (1978). "Minimum Entropy Deconvolution," *Geoexploration* **16**, 21:35.

Yang, Z., M. Kallergi, R. DeVore, B. Lucier, and others (1995). "Effect of wavelet bases on compressing digital mammograms." *IEEE Engineering in Medicine and Biology Magazine*, **14**, 570-577.

Yomogida, K. (1994). "Detection of anomalous seismic phases by the wavelet transform," *Geophysical Journal International* **116**, 119-130.

Young, R. (1993). *Wavelet Theory and Its Applications*. Boston, MA: Kluwer Academic Publishers.

Zafar, S. (1993). "Multiscale video representation using multiresolution motion compensation and wavelet decomposition," *IEEE Journal on Selected Areas in Communications* **11**, 24-35.

Zhuang, Y., and J. Baras (1994). "Optimal wavelet basis selection for signal representation," *Proceedings of the SPIE (Wavelet Applications, Orlando, FL, USA, 5-8 April 1994)*, **2242**, 200-211.

Ziv J., and A. Lempel (1977). "A universal algorithm for sequential data compression," *IEEE Transactions on Information Theory*, **23**, 337-343.

*Technical Information Department • Lawrence Livermore National Laboratory*  
University of California • Livermore, California 94551

