T-3450

AIRGUN SIGNATURE ESTIMATION AND WAVELET PROCESSING OF MARINE SEISMIC DATA

by

Yuesheng Li

ProQuest LLC.
789 East Eisenhower Parkway
P.O. Box 1346
Ann Arbor, MI 48106 – 1346

T-3450

A thesis submitted to the Faculty and the Board of Trustees of the Colorado School of Mines in partial fulfillment of the requirements for the degree of Master of Science (Geophysics).

Golden, Colorado
Date August 14, 1987

Signed: Yuesheng Li
Yuesheng Li

Approved: Raymond L. Sengbush
Raymond L. Sengbush
Thesis Advisor

Golden, Colorado
Date 15 August 1987

P R Romig
Phillip R. Romig,
Head, Department of
Geophysics

ABSTRACT

A new technique to estimate the airgun array signature directly from the seismic data and a new wavelet processing method are proposed. The array signature is simulated by a gaussian-damped sine function, and the signature estimate is based on the superposition of arrivals of this function along all possible primary and multiple raypaths, taking into account spherical divergence and reflectivity in each path. Parameters in the simulated signature, frequency, phase and damping, are iterated until an optimal match is obtained between the simulated first arrivals and the actual first arrivals on the seismic data.

The validity of the array signature estimation technique is proven by showing that the simulated signatures match measured far-field signatures very well.

Airgun signatures are never minimum phase, and it is necessary in wavelet processing to obtain a measurement or a valid estimate of the signature in order to phase compensate the seismic data. In this wavelet processing method, phase compensation is accomplished by a compensator derived from the estimated source signature.

Deconvolved data must be bandlimited because the noise outside the seismic band is amplified by Wiener-Robinson

statistical deconvolution. In this wavelet processing method, a matched filter derived from the estimated source signature after passing through the instruments is used as the bandlimiting filter. Because the matched filter has the same amplitude spectrum as the seismic pulse, the high frequency components of the signal are preserved and the noise is suppressed simultaneously. Thus, the new method completely avoids the difficulty in conventional methods of correctly choosing the bandlimiting filter.

Quantitative comparisons on both noise-free and noisy synthetic data show that this new wavelet processing method is superior to conventional methods in the sense of preserving high frequency components of signal and simultaneously suppressing noise.

The new wavelet processing method applied to real marine data confirms the superior performance of the new method over the conventional methods.

TABLE OF CONTENTS

TABLE OF CONTENTS (continued)

LIST OF FIGURES

LIST OF FIGURES (continued)

LIST OF FIGURES (continued)

LIST OF FIGURES (continued)

LIST OF FIGURES (continued)

LIST OF FIGURES (continued)

LIST OF TABLES

ACKNOWLEDGEMENTS

I would like to express my gratitude to Professor Ray L. Sengbush for serving as advisor, suggesting valuable ideas, sacrificing precious time, offering hard-earned experience for this thesis work, and also for his guidance and encouragement throughout my academic program.

Sincere appreciation is extended to Dr. Frank A. Hadsell and Dr. Guy H. Towle for their service as thesis committee members, and for the amicableness shown to me. Special thanks go to Dr. Thomas L. Davis for his guidance and encouragement in the begining of my graduate study.

I am indebted to the China National Offshore Oil Corporation for providing the educational opportunity, to the Atlantic Richfield Company for establishing the fellowship awards, and to The Asia Foundation for participating in this cooperative effort. Without this support, this study would have been impossible.

I also would like to thank GECO Geophysical Company, Inc. and Geophysical Service Inc. for providing the seismic data used in this study.

Finally, I would like to express my deepest thanks to my wife, Ling, for her affection, commitment, and encouragement throughout the course of my graduate study.

INTRODUCTION

Detailed seismic interpretation requires that the seismic sections have high resolution, high signal-to-noise ratio, and correct traveltimes. High resolution is achieved by applying deconvolution. The type used almost universally is Robinson's adaptation of Wiener optimal filter theory to seismic exploration, which is optimal if the data are minimum phase. If the data are not minimum phase, then it is necessary to phase compensate the deconvolved data. Phase-compensated Wiener-Robinson deconvolution is called wavelet processing.

In marine seismic exploration, the airgun signature and the instruments are nonminimum phase. To compensate for nonminimum phase, it is necessary to measure or estimate the far-field signature after it has passed through the instruments.

The impulse response of the instruments can be measured directly, but the source wavelet is impossible to measure during field surveys because most marine seismic exploration is done on shallow continental shelfs, where the depths of water are much less than the necessary depth (at least 250 m) for far-field measurement (Ziolkowski, 1987). One solution to this problem is to perform a

separate experiment in deep water away from the survey area. However, a separate experiment is expensive and time-consuming and, as pointed out by Ziolkowski, suffers from a number of difficulties. Several computer-related techniques have been suggested by geophysicists in recent years. These techniques include a conversion method (Ziolkowski et al, 1982), wavefield extrapolation (Hargreaves, 1984), computer simulation (Dragoset, 1984; Vaage and Ursin, 1987), and scan extraction (Zhang, 1985).

In this study, a scan method, which is similar in principle to Zhang's method, is proposed and used to estimate the airgun array signature. The principle is developed in detail and the validity is proven by using real data.

A common problem in conventional wavelet processing is the selection of the bandlimiting filter. Wiener-Robinson statistical deconvolution amplifies not only signal but also noise. Because noise has broader spectrum than signal, the overall signal-to-noise ratio is reduced after deconvolution. To make the processing results interpretable, a zero-phase bandlimiting filter is applied to deconvolved data. The filtered data depends on the passband of the filter used. Narrow passbands will cause the loss of high frequency components of the signal, while

wide passbands will not suppress the amplified noise very well.

In this study, a matched filter is proposed as the bandlimiting filter. This matched filter is obtained from the bandlimited source signature by means of fast Fourier transform. The advantage of this method is that the difficulty of selecting the passband of the bandlimiting filter in conventional methods is avoided completely.

The validity and superiority of the new wavelet processing method are shown on both real and synthetic data.

BRIEF REVIEW ON WAVELET ESTIMATION

Wavelet estimation must be resorted to when measurements are unreliable or simply not available. Direct measurements of the far-field signature of airgun array must be made in deep water to avoid interfering reflections. Most often, water depths in the survey areas are too shallow to allow direct measurements during the course of normal production shooting. One solution to this problem is to perform separate experiments in deep water away from the survey areas, in hope of obtaining a signature representive of that generated by the array under normal shooting conditions. However, it is probable that the vertical far-field signal emitted by the airgun array is different in the seismic survey from the far-field measurement. Separate experiments are also expensive and time-consuming. To solve this problem, several computer-related approaches have been suggested in recent years. These computer-related approaches include conversion method, wavefield extrapolation, computer simulation, and scan extraction. A brief review on these methods will be given in this Chapter.

Far-field Conversion from Near-field Measurements

Rather than making a direct measurement of the far-

field signature, Ziolkowski et al (1982) have suggested an alternative approach to obtain the far-field signature. Their technique includes two parts: near-field measurement and far-field conversion. In the technique, each gun of the array is equipped with a permanent near-field hydrophone at a distance of one meter. The pressure field near each gun of the array is recorded during the field survey. Then, an iterative technique is employed to calculate a notional array of noninteracting sources from the near-field hydrophone measurements. The notional signatures form the basis for calculating the array signature in any direction and any distance. The far-field signature of the array is then computed by superposition of the notional sources, scaled and delayed relative to each other according to distance and direction. In Ziolkowski's refined technique (1984), the forward motion of the hydrophones and upward motion of the airgun bubbles are taken into account by a linear model.

Ziolkowski's method avoids the difficulties of direct far-field measurement (Ziolkowski, 1987), and the results obtained are quite accurate according to Ziolkowski's papers (1982, 1984, 1987). The technique requires a knowledge of the array geometry precisely. This is a severe limitation since fluctuations in the source depth and

changes in sea state may introduce undetected changes in the array signatures and variations in the array geometry (Hargreaves, 1984).

Wavefield Extrapolation

A technique for far-field signature estimation by wavefield extrapolation was proposed by Hargreaves (1984). Like Ziolkowski's method, Hargreaves's technique also includes two steps: field measurement and computer processing. During the field survey, a mini-streamer consisting of several closely spaced hydrophones is located beneath the source array and records the pressure wavefield at the depth of the streamer. The far-field signature is then obtained by downward continuation, or wavefield extrapolation, of the recorded near-field data. This extrapolation can be performed either by a phase-shift operation or by a summation-type solution of the wave equation. In principle, this technique may be used to obtain far-field signatures in any direction away from the array.

Hargreaves's technique does not require any precise knowledge of the array geometry or the location of the recording sensors, which he considers to be the main advantages over Ziolkowski's method. Because the wavefield extrapolation used is a 2-D rather than 3-D process, the

ghost component of the derived far-field signature is slightly underdeveloped.

Computer Simulation

Several computer simulation methods have been proposed by geophysicists (Ziolkowski, 1982; Dragoset, 1984; Vaage, and Ursin, 1987) to calculate the far-field signature of an airgun array. Each of these methods is based on models and empirical approximations. Vaage's method is only applicable to linear airgun arrays where the distances between the airguns are such that the non-linear interaction among the airguns is negligible. Both Ziolkowski's and Dragoset's methods include airgun interactions and therefore can be used to simulate any airgun arrays. The basic equations in Dragoset's model are similar to those in Ziolkowski's model except that the initial conditions and amplitude decay are treated somewhat differently. Here only Dragoset's method is discussed.

In Dragoset's method, a system of differential equations is used to calculate synthetic signatures for airgun arrays. The system includes three major elements: (1) the behavior of ideal, freely oscillating air bubbles, (2) the behavior of the air in the guns and bubbles while the gun ports are open, and (3) a first-order description of the interactions between the oscillating bubbles and the

pressure waves they create. In addition to the mechanical parameters of the airguns and the array geometry, four adjustable parameters are considered: two damping coefficients, the specific heat ratio of the air, and the air pressure remaining in the gun when the ports close. The parameter values are found by matching synthetic signatures to measured signatures. The final model is a combination of theory and empiricism, and the results shown in Dragoset's paper (1984) indicate that the technique works quite well.

Scan Extraction

A scan extraction method was proposed by Zhang (1985) to extract the source wavelet from real seismic data. Only a brief introduction of the method was given in his paper. The technique is based on stacking theory and similarity analysis. First, a theoretical source wavelet is calculated based on an undisclosed waveform function. Then, a synthetic waveform is computed according to stacking theory, and the similarity between the synthetic and measured waves is measured. A scan method is employed to obtain the synthetic source wavelet corresponding to minimum error between synthetic and measured waves. The results shown in Zhang's paper indicate that the technique works quite well.

BRIEF REVIEW ON CONVENTIONAL WAVELET PROCESSING

Since statistical deconvolution was first invented by Robinson in 1954, wavelet processing has been an attractive topic in seismic exploration. Based on different models, many processing techniques have been proposed. In this Chapter, two conventional methods commonly used in petroleum industry are reviewed.

Conventional Method without Phase-compensation

In conventional seismic data processing, a simple method is often used to compress the seismic wavelet to a wavelet of known phase, which is usually assumed to be zero, as well as to broaden the amplitude spectrum of the wavelet for improving resolution and possibility of detailed interpretation. This simple method is completely based on the Wiener-Levinson model for statistical deconvolution which makes the following assumptions:

1) The effective seismic pulse is time invariant and minimum phase.
2) The additive noise is a stationary random process.
3) The reflectivity has the correlation property of white noise.

The processing procedure includes three steps:

1) Removing or suppressing coherent noise,

2) Statistical deconvolution of the seismic data,

3) Zero-phase bandpass filtering of deconvolved data.

The simple processing method, in most cases, improves the resolution of the seismic data, but phase distortion is introduced by the processing whenever the effective seismic pulse contains nonminimum-phase components, such as source wavelet and impulse response of instruments. Phase distortion degrades the resolution of the data and results in miscalculations of depths due to incorrect traveltimes.

Conventional Method with Phase-compensation

To reduce the problem of phase distortion caused by applying spiking deconvolution to nonminimum phase data, phase compensation is incorporated into the processing procedure reviewed above. To phase compensate, the nonminimum phase components must be known, either by reliable measurements or by accurate estimation. Then, the all-pass phase compensator is calculated from the nonminimum phase components and applied to the seismic data (Connelly, 1985; Sengbush, 1986). Phase compensation may be applied either before or after deconvolution, because all-pass filtering commutes with deconvolution. All other steps in the processing remain the same. This modified conventional method works better provided the nonminimum phase components are accurately measured or estimated.

A Common Problem

A common problem of conventional methods is that it is not easy to select correctly the passband of the zero-phase filter (the d-filter). As pointed out by Hatton et al (1986), the choice of the filter must be based on a balance between resolution improvement and signal to noise (S/N) degradation. As the noise spectrum is broader than the signal spectrum, broadening the signal spectrum amplifies frequencies with poorer S/N and reduces overall S/N ratio. S/N estimates are notoriously difficult to make, and degradation is unpreditable for a specific filter.

NONMINIMUM PHASE COMPONENTS

As mentioned in the introduction, the effective seismic pulse contains two possible nonminimum phase components: the source wavelet and impulse response of instruments. To check the validity of this statement, the Wiener transform is deployed to compute the minimum-phase equivalents of instrument responses and airgun signatures.

The Wiener Transform

As defined by Sengbush (1986), the Wiener transform is a short, concise name for spiking deconvolution (without additive white processing noise) applied to deterministic finite-length functions; there are no statistical implications. So the Wiener transform of a function y, written W(y), is the inverse of the minimum-phase equivalent of y, i.e., $W(y)=y_m^{-1}$, where subscript m means minimum phase, and superscript -1 means inverse.

Having defined the Wiener transform, we may compute the minimum-phase equivalent of any signature by applying the Wiener transform twice, i.e., $W(W(y))=W(y_m^{-1})=y_m$.

Impulse Response of Instruments

The measured inpulse responses of DFS-V^{TM} (Trademark Texas Instruments) seismic instruments with different

filters were obtained from GSI (Geophysical Service Inc.). The Wiener transform was applied twice to each of these impulse responses to obtain their minimum-phase equivalents, and the results are superimposed with their corresponding measured impulse responses in Figures 1 to 3, where the solid curve represents the measured impulse response and the dashed curve represents the minimum-phase equivalent. Figure 1 is for the filter with passband out--128 Hz, 72 dB/octave rejection rate. Figures 2 and 3 are for the filters with passbands out--256/72 and 8/18--256/72, respectively. The correlation coefficients (hereafter called CC value, see Appendix A) at zero shift between measured impulse responses and their minimum-phase equivalents are 0.763, 0.307 and 0.274, respectively. With 2 ms time shift, the CC values become 0.977, 0.976 and 0.958, respectively.

The above results indicate the nonminimum-phase characteristic of the impulse response of DFS-V instruments regardless of the filter passband. The impulse response is delayed beyond the time at which the impulse is applied. Thus, if phase correction is not applied to the data recorded by DFS-V instruments, then the processed seismic data will have a static delay which will result in miscalculations of reflector depths.

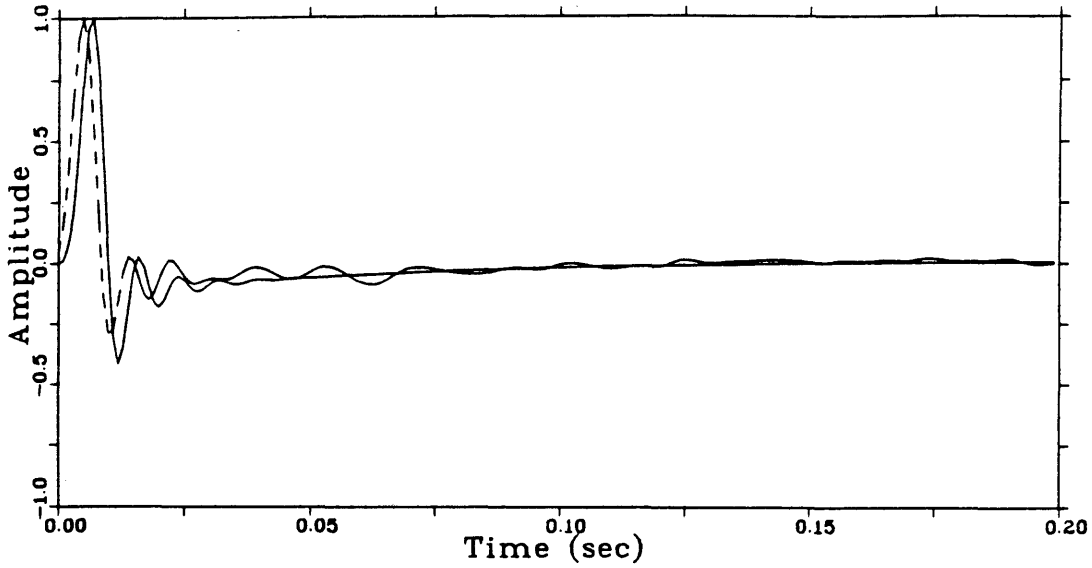

Figure 1. Impulse response of DFS-V with filter out-128/72 (solid) and its minimum-phase equivalent (dashed).

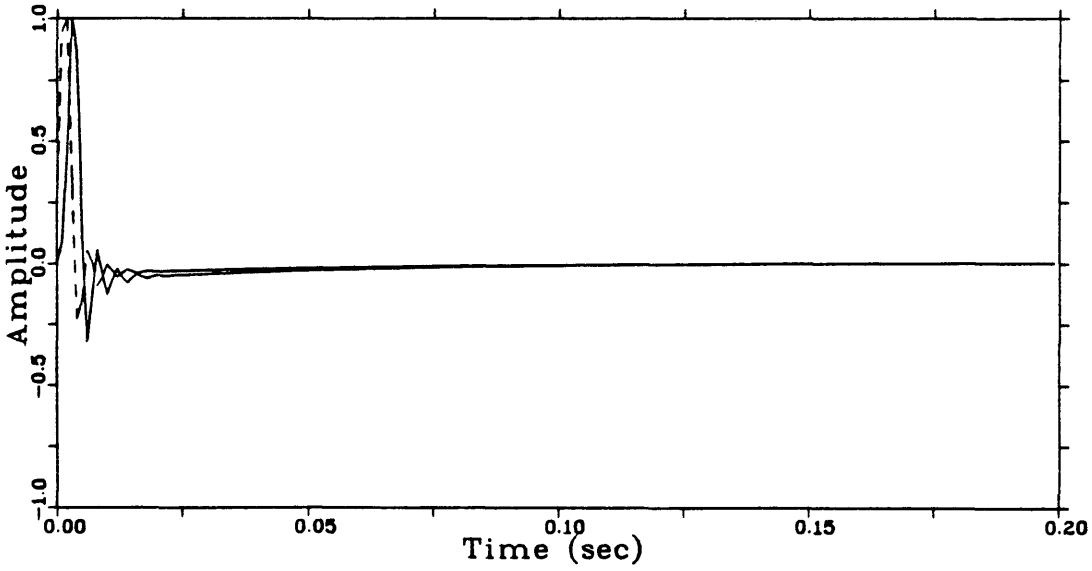

Figure 2. Impulse response of DFS-V with filter out-256/72 (solid) and its minimum-phase equivalent (dashed).

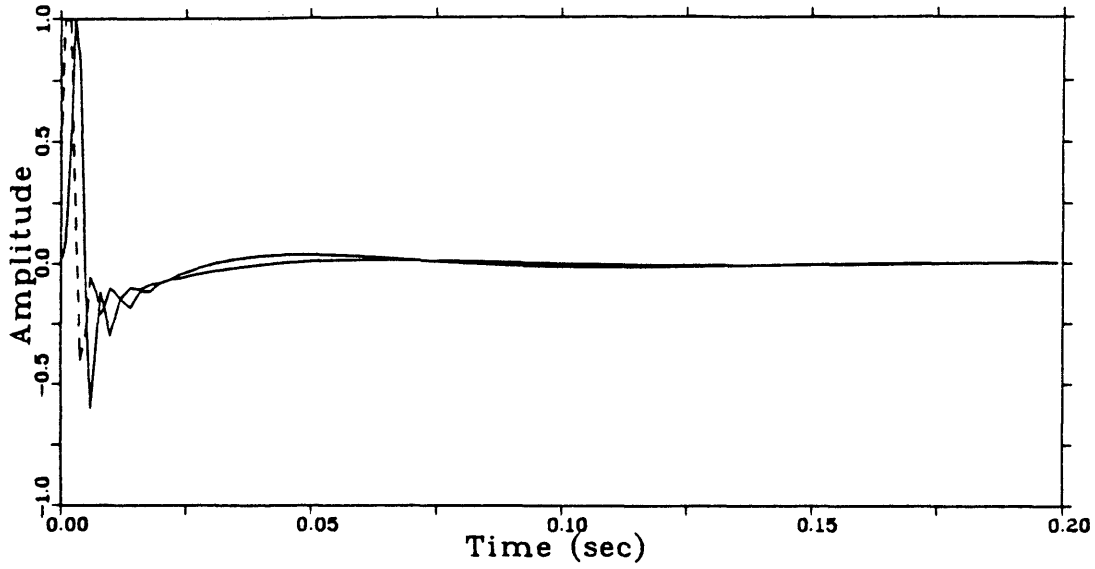

Figure 3. Impulse response of DFS-V with filter 8/18-256/72 (solid) and its minimum-phase equivalent (dashed).

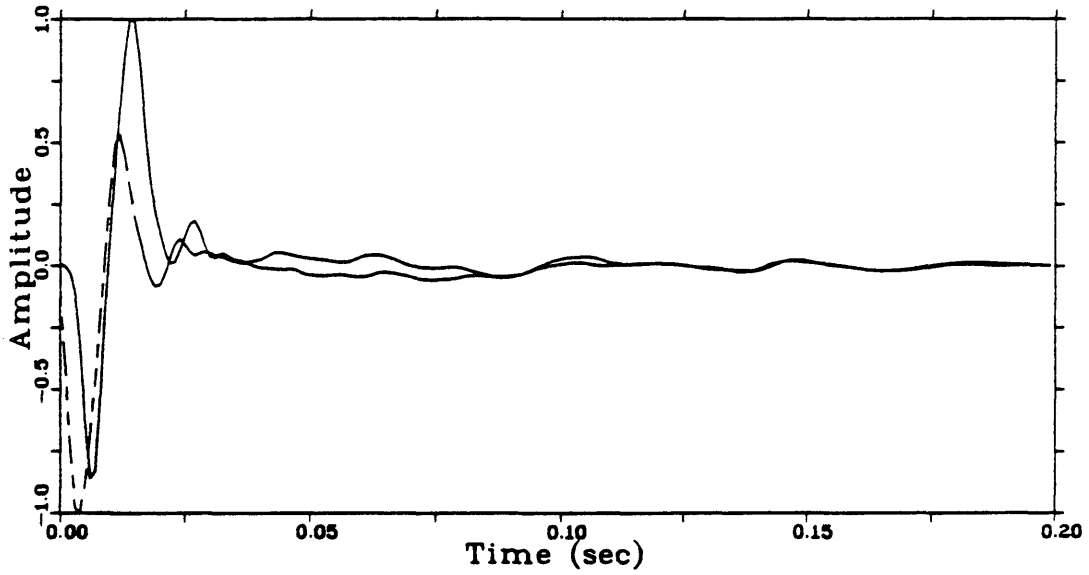

Figure 4. Measured far-field airgun array signature (solid) from GSI and its minimum-phase equivalent (dashed).

Far-field Airgun Array Signature

A measured far-field signature of GSI airgun array is obtained by using a DFS-V system with filter passband out--256/72. The airgun array has 3380 cu.in. volume and is located at depth of 6 meters. The signature (traveltime removed) is plotted in Figure 4 by the solid curve, and its minimum-phase equivalent, found by using the Wiener transform twice, is the superimposed dashed curve in Figure 4. The CC value between two curves is 0.599 with zero shift and 0.825 with 2 ms time shift. The measured far-field signature is definitely nonminimum phase.

AIRGUN SIGNATURE ESTIMATION

In this Chapter, a scan method to estimate airgun array signatures is developed. The method, similar in principle to Zhang's method but different from other methods mentioned in the previous Chapter, is based on raypath theory and stacking theory. The principle of the method is discussed in detail, and real data are used to prove the validity of the method.

Formulation of Airgun Signature

Airgun arrays are the main source used in marine seismic exploration. The waveform of the source is relatively stable and can be approximated by a mathematical formula. A set of far-field airgun array signatures provided by GECO Geophysical Company, Inc. is shown in Figure 5. The signatures are from the same airgun array at the same source depth. After analyzing these signatures, we find that the main part of the signatures can be described by a gaussian-damped sine function, which is expressed by

$$b(t) = \exp(-\alpha t^2) \sin(2\pi ft+\phi), \qquad (1)$$

where b(t) is the signature; α is the damping coefficient; t, f and ϕ are time, frequency and phase, respectively. In order to prove this idea, a set of measured signatures is

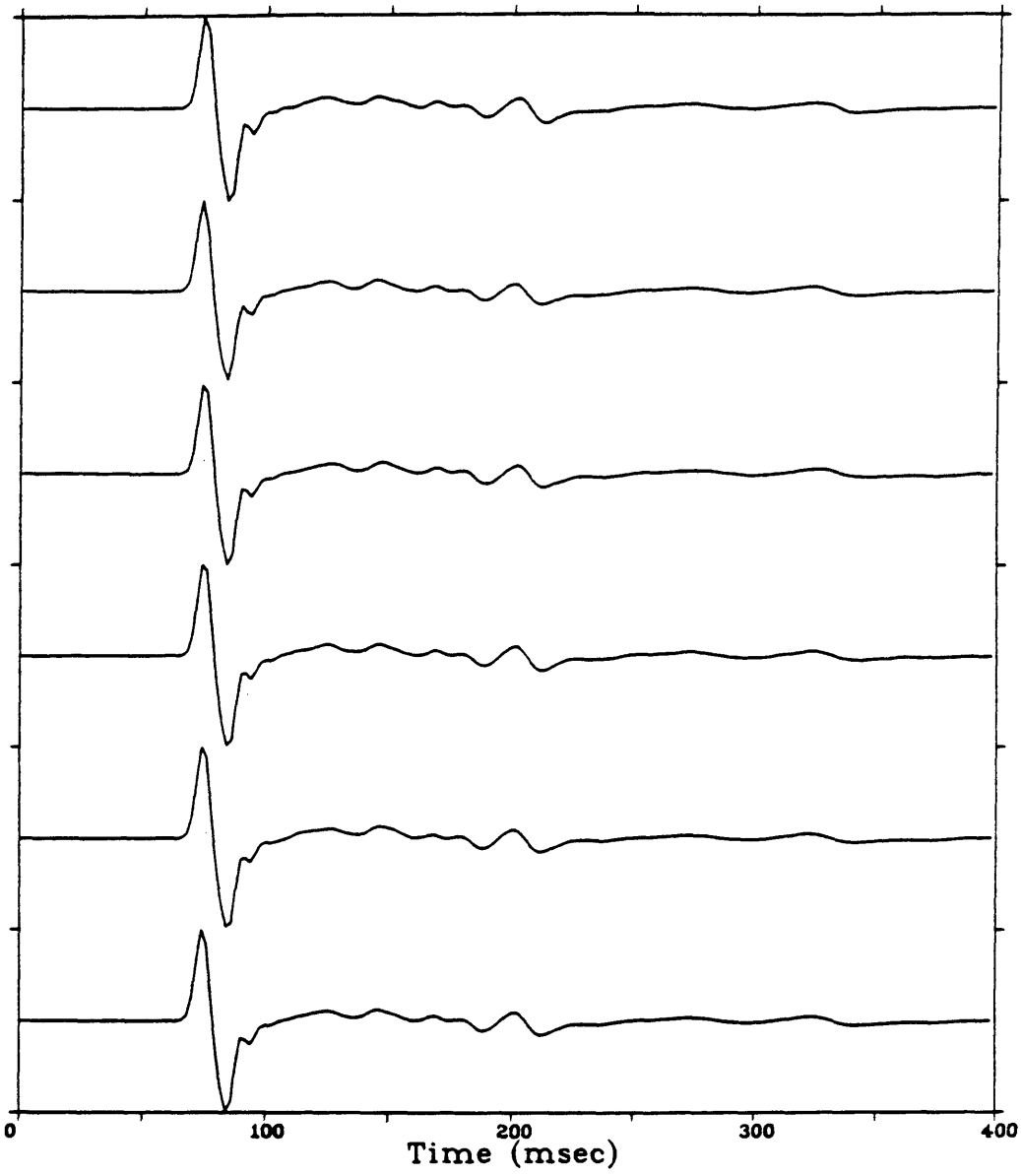

Figure 5. A set of far-field airgun array signatures obtained by using same array at same depth. Signatures provided by GECO.

compared with their corresponding simulated signatures based on equation (1). This will be discussed in next section.

The Validity of Formulation

Important questions which we need to answer before we do any further work are, how well does the exponential damped sine function approach an airgun array signature, or how close is the calculated signature to the measured signature? To answer these questions, measured far-field signatures are compared with their corresponding calculated signatures.

The simulation is based on equation (1). The algorithm uses the correlation coefficient between the measured signature and the calculated signature as the critical parameter to evaluate the accuracy of the result, and a scan method is used to optimize the calculated signature based on error evaluation, where the error equals (1-CC). The procedure is summarized by the following steps:

1) Set up the geometrical model similar to field conditions and give initial values to parameters α, f, ϕ, and err0, where err0 is the threshold value that determines when iteration ceases.
2) Compute the airgun array signature based on equation (1).

3) Compute the far-field signature by adding the shot ghost.
4) Convolve the result of step 3 with the impulse response of instruments.
5) Do similarity analysis between the theoretical signature and measured signature to obtain the CC value.
6) Make a decision whether or not to stop iteration. If the error (err = 1-CC) is less than or equal to the threshold value err0, stop the iteration, normalize the result and write the result into a data file for plotting. If the error is larger than err0, then modify parameters α, f, ϕ and repeat steps 2 to 6.

The above calculation is in time domain. In order to see the difference between calculated and measured signatures in the frequency domain, an FFT algorithm is employed to compute the amplitude spectra of calculated and measured signatures. The measured signature and the corresponding calculated signature are superimposed in both the time domain and frequency domain for comparison purpose.

The measured far-field airgun array signature from GSI and its corresponding calculated signature are shown in

Figure 6. The solid curve represents the measured signature, and the dashed curve represents the calculated signature. This far-field signature is obtained from a 3380 cu.in. airgun array mixture of GSI PUN-CON and sleeve guns at 2000 psi pressure at depth of 6 meters, with the recording hydrophone approximate 120 meters below the center of the gun array in water depth greater than 1000 meters, using a DFS-V recording system with sampling interval 1 ms, filter passband, out--256, and 72 dB/octave rejection.

The error between measured signature and simulated signature is 3.34%. Comparison of signatures in Figure 6 shows that the main part of the wavelet is almost exactly the same, and that there are some small difference in the tail part. This small difference should not be so important for wavelet processing because most energy of the wavelet is concentrated in the main part. In the frequency domain, the amplitude spectra in Figure 7 match closely except at the ghost notch. On simulated data, the ghost operator is given by $\delta(t)-R\delta(t-T)$, where δ is a delta function, and T is the time delay of the ghost. On real data the ghost notch is smeared because impulses don't exist in the real world.

The simulated signatures without source ghost before

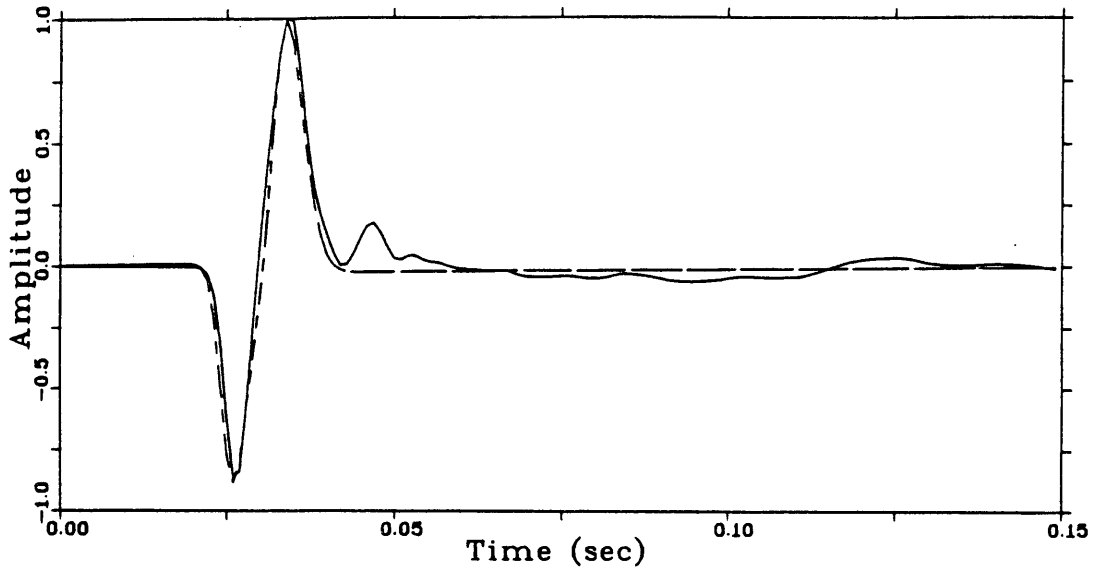

Figure 6. GSI measured far-field signature (solid) and its corresponding simulated signature (dashed).

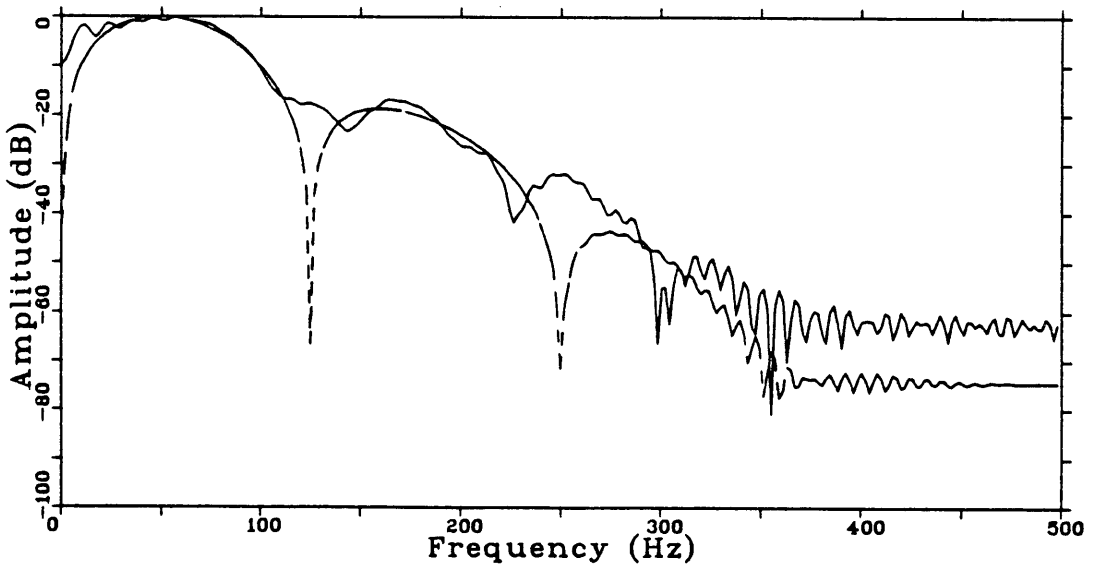

Figure 7. The amplitude spectra of signatures in Figure 6, solid for measured and dashed for simulated.

and after convolving with the DFS-V instrument response are plotted in Figure 8: the solid curve for wavelet without effect of DFS-V and the dashed curve for wavelet with effect of DFS-V. Their amplitude spectra are superimposed in Figure 9.

To show that equation (1) can be used to simulate signatures of different airgun arrays, several measured signatures from GECO Geophysical Company, Inc. are tested. All signatures are recorded by a DFS-V system with sampling interval 2 ms, filter passband, out--128/72. The sources are 959 cu.in. sub-array at depths of 7.3 meters or 9.2 meters, and 1254 cu.in. sub-array at depth of 7.3 meters. The GECO simulation results show very good agreement between measured signatures and their corresponding calculated signatures in both the time and frequency domains. The CC values between measured and calculated signatures are over 95%. The solid curve in Figure 10 represents the measured signature from the 959 cu.in. sub-array at gun depth of 7.3 meters, and the dashed curve is the calculated signature. The error is 4.48%. Their amplitude spectra are displayed in Figure 11 by the same plotting convention, solid for measured and dashed for calculated. The signatures from 1254 cu.in. sub-array are shown in Figures 12 and 13 in the time domain and frequency

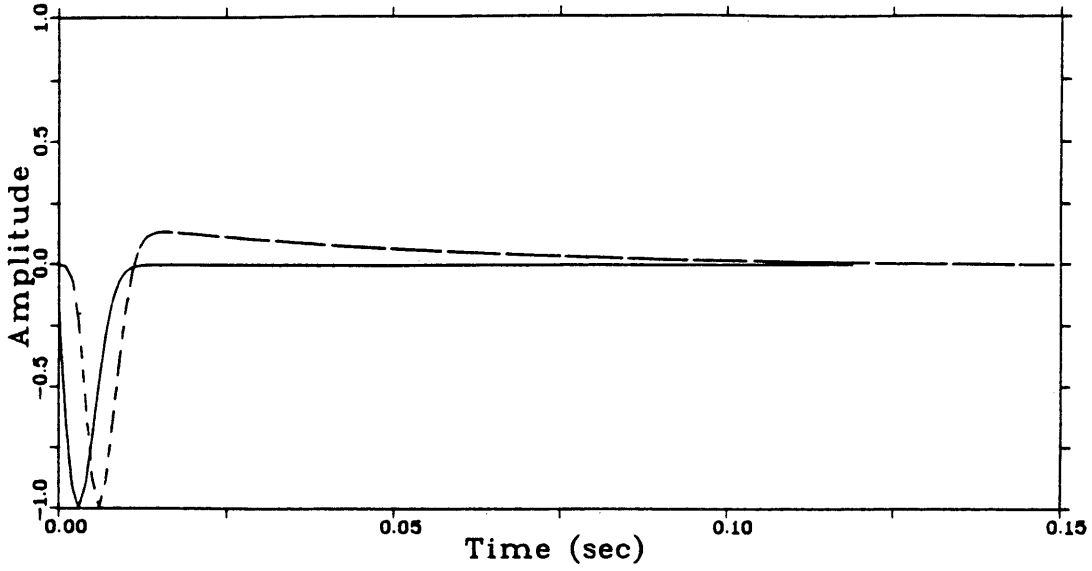

Figure 8. The simulated GSI signatures w/o ghost before (solid) and after (dashed) passing through DFS-V instruments.

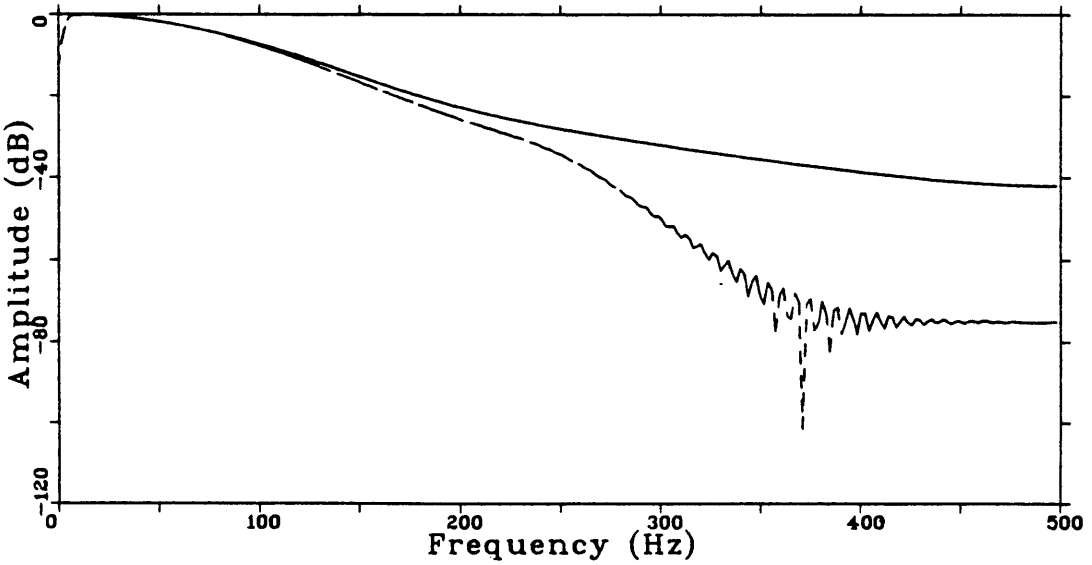

Figure 9. The amplitude spectra of signatures in Figure 8, before (solid) and after (dashed) passing through DFS-V instruments.

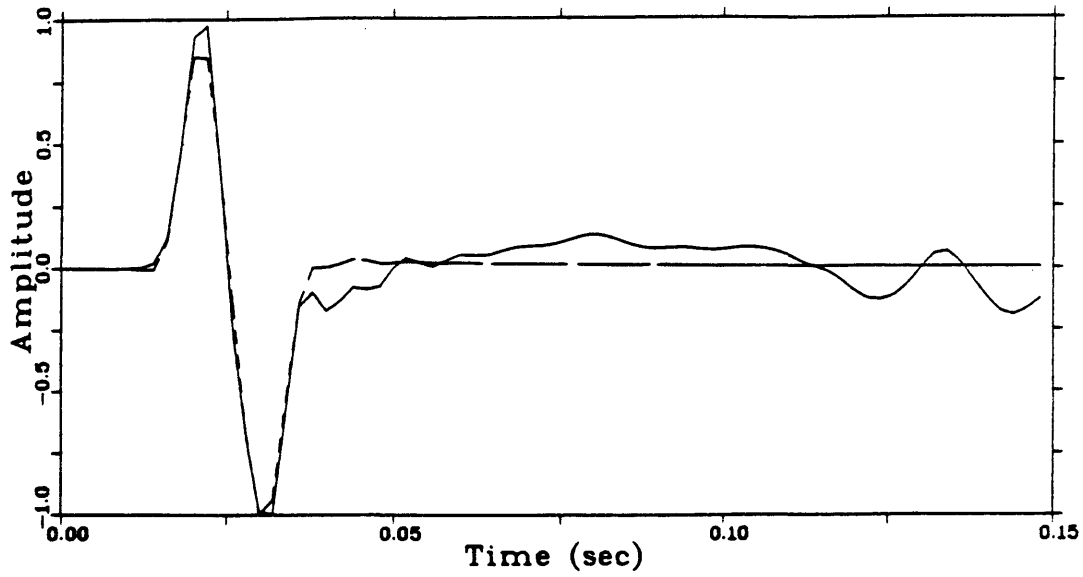

Figure 10. GECO measured far-field signature from a 959 cu.in. array (solid) and its corresponding simulated signature (dashed).

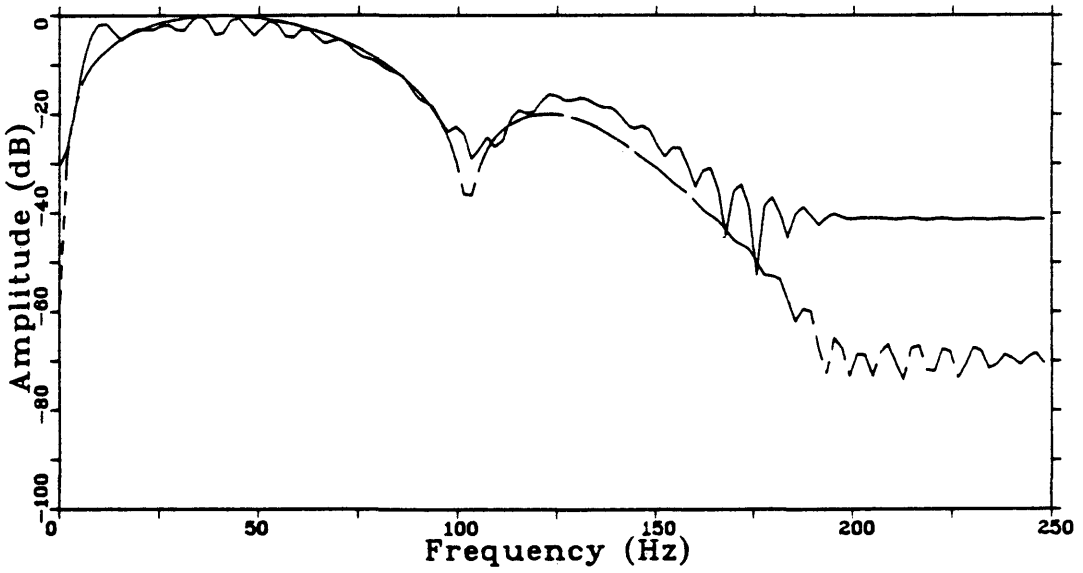

Figure 11. The amplitude spectra of signatures in Figure 10, solid for measured and dashed for simulated.

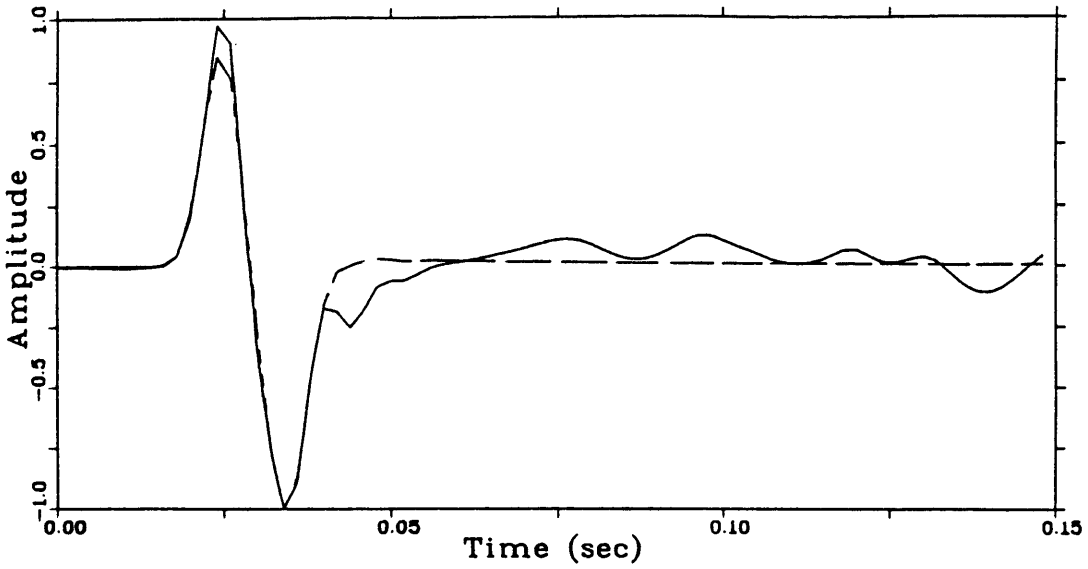

Figure 12. GECO measured far-field signature from a 1254 cu.in. array (solid) and its corresponding simulated signature (dashed).

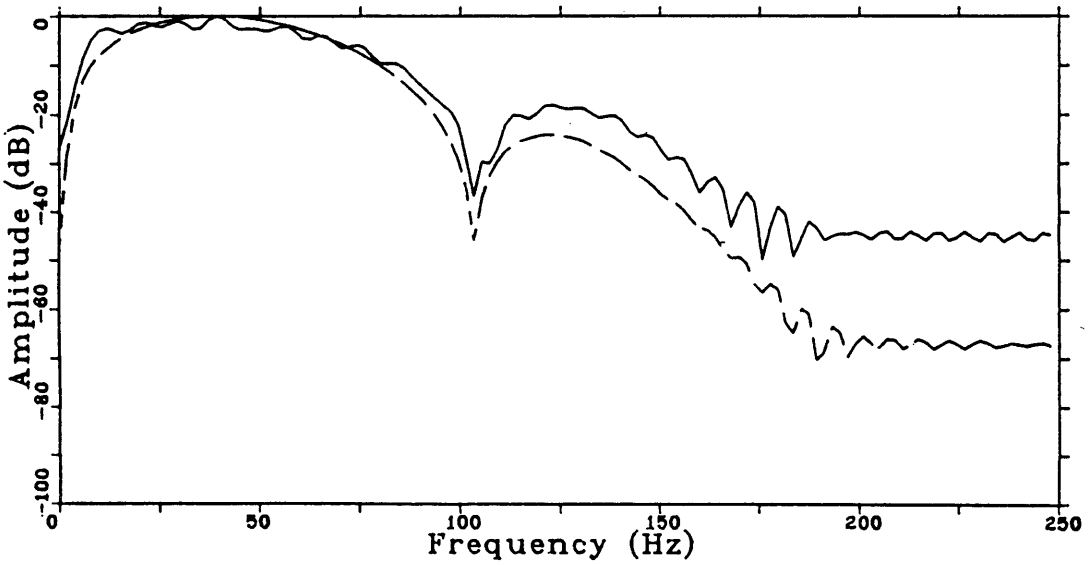

Figure 13. The amplitude spectra of signatures in Figure 12, solid for measured and dashed for simulated.

domain, respectively. The error between the two signatures is 3.47%. In the time domain, the calculated signatures are very similar to their corresponding measured signatures on the main part. In the frequency domain, they are also very close except the calculated signatures have a smoother spectra.

The small difference between measured and calculated signatures is caused by three possible reasons, which are as follows:

1) The bubble pulse oscillations which are not depressed completely by the source array.
2) The impulse response of the DFS-V recording system used in simulation may be different from that of the instrument used in field survey. Though the passbands of the filters are nominally the same, the impulse responses may vary from instrument to instrument.
3) The effect of the hydrophone is not included in computer simulation.

From the above results, we can see that airgun array signature can be approximated quite accurately by equation (1). The result gives us an interesting hint: Can the airgun array signature be estimated from marine seismic data? This question will be answered in next section.

The Principle of Wavelet Estimation

As mentioned in the previous Chapter, most marine seismic exploration is done on the shallow continental shelf where direct measurement of airgun array signature is impossible. To solve this problem, a new method of estimating the wavelet from marine seismic data is developed.

Because of the special hydrological and geological conditions, marine seismic data are more stable than land data. After analysis of marine data, we find that the early part of the seismic trace, which is about 100 ms in length from the first break, has a very stable waveform. This is especially true for near-shot-trace gathers because the early part of near-shot traces may not include subsurface reflections. A near-shot trace gather from GECO in Figure 14 shows that the early part of the waveform (about 85 ms from the first break) is very similar for different shots. This characteristic implies that it might be possible to describe the early part of the near-shot trace by some mathematical formula.

The validity of formulation of the airgun array signature has been proven in last section. According to raypath theory and stacking theory, the early part of near-shot traces may be expressed by the following formula:

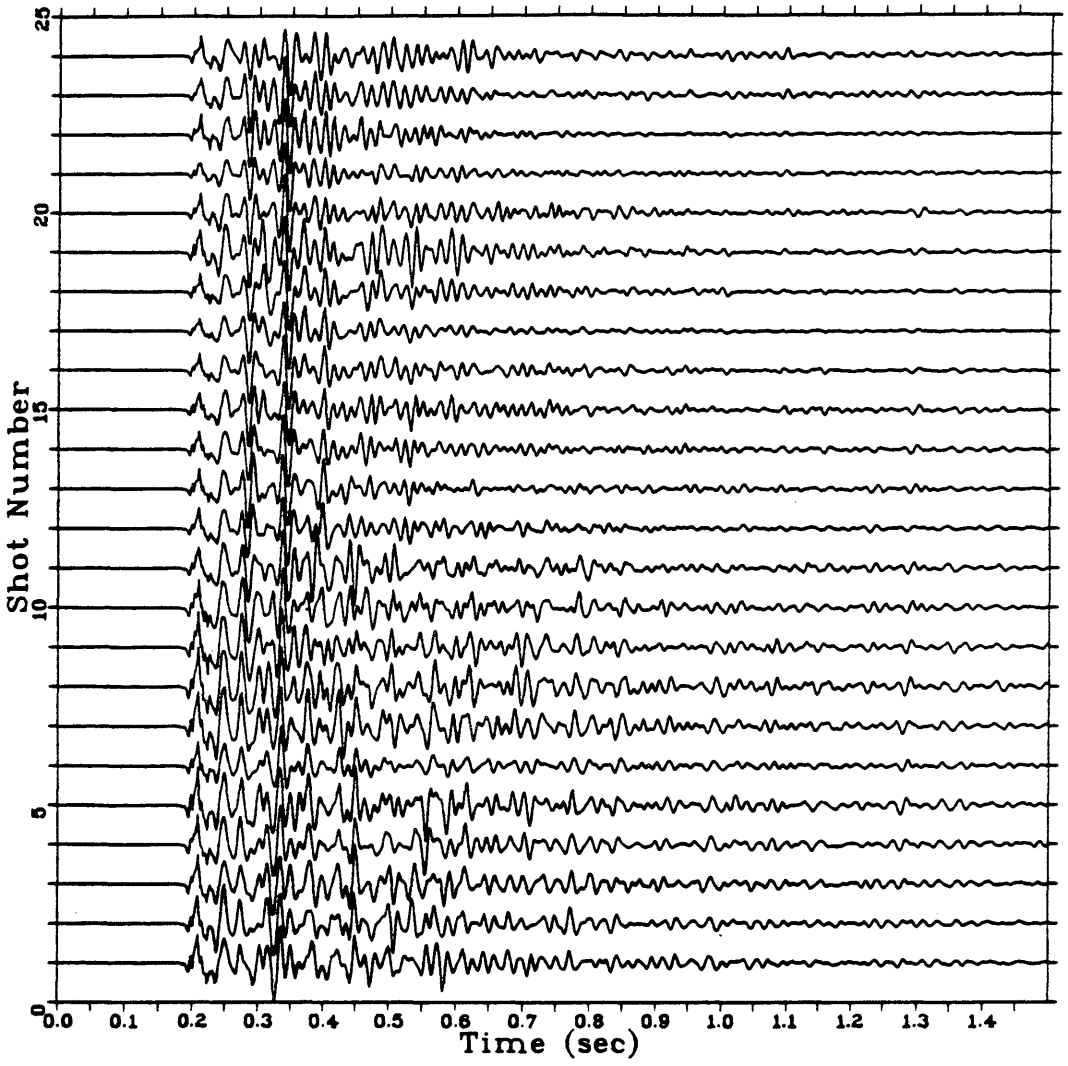

Figure 14. A near-shot trace gather of 24 shots from GECO test 5 marine data, displayed up to 1.5 seconds.

$$W(t,x) = \sum_{k=1}^{n} A_k(x)\ b[t-T_k(v,x)], \qquad (2)$$

where W is the composite wave at time t and offset distance x; b(t) is the source wavelet function given by equation (1); T_k is the travel time for the kth wave determined by its raypath and velocity function. A_k is the amplitude of the kth wave determined by spherical divergence and reflection coefficients in its raypath. The summation sign indicates the stacking of all possible waves which arrive at the receiver at offset distance x at time t.

Figure 15 shows four types of waves which affect the early part of near-shot trace in marine seismic exploration. They are the direct wave, free-surface reflection, sea-bottom reflection and sea-bottom refraction. In addition to these four waves, the possible waves in equation (2) also include the source ghost, receiver ghost, water-layer multiples (reverberations), and in some cases, reflections from the subsurface below the sea bottom.

The amplitude in equation (2) depends on the travel distance of each wave and the reflection coefficients in the path, which are functions of velocity, density, Poisson's ratio and incident angle, and are obtained by solving the Zoeppritz equations. The travel distance

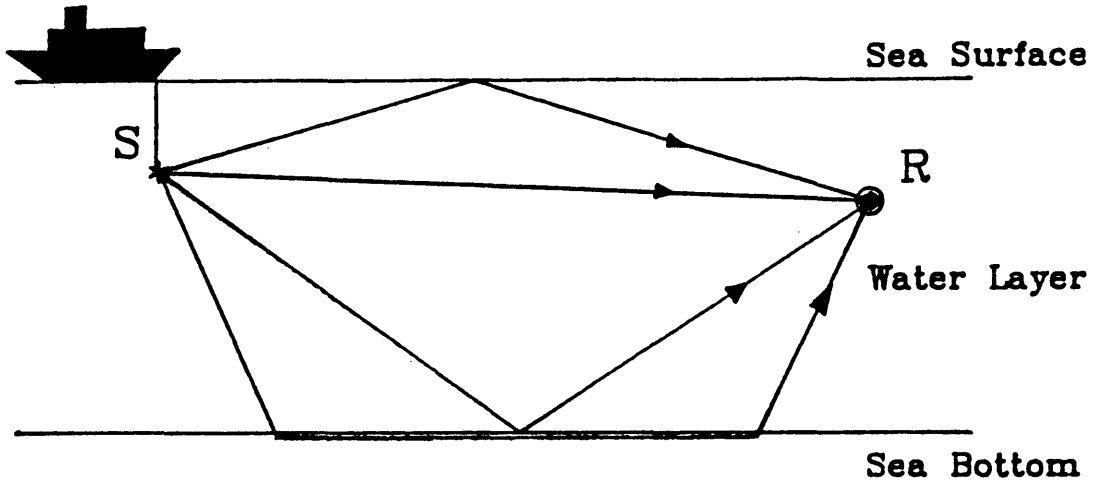

Figure 15. Four types of waves affecting the early part of near-shot trace in marine seismic exploration.

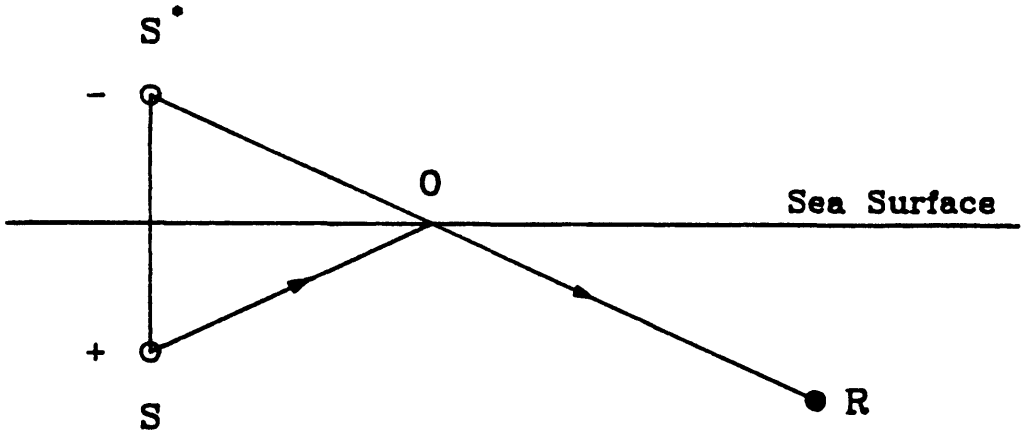

Figure 16. A method of image solution for the sea-surface reflection (shot ghost).

affects the amplitude because of the effect of geometric divergence.

Having introduced equation (2) to describe the early part of near-shot traces of marine data, it now becomes possible to estimate the airgun array signature from marine seismic data if we use the correct model. Before the estimation method is discussed in detail, some techniques relating to the model will be introduced first.

Distance and Incident Angle

Travel distance is a very important parameter for determining the travel time T_k and amplitude A_k in equation (2). It depends on the travel path which in turn depends on the geometric locations of source and receiver, water-layer depth, offset, and wave type. Incident angle varies from wave to wave and is an important parameter in determining amplitude A_k because the reflection coefficient is a function of incident angle. Incident angle is determined by the same parameters as those for travel distance.

To find formulas which can be used to calculate travel distances and incident angles for different waves, image-source theory is employed. Without this technique, the problem would become complicated for ghosts and reverberations.

Figure 16 illustrates the principle of image solution

for the sea-surface reflection; i.e., the source ghost. In the figure, S is the source, and R is the receiver. The wave travels from S to sea surface O and is then reflected at O and received by R. The image theory states that the raypath SOR is equivalent to the raypath S^*OR, where S^* is the image point of S. This technique makes it easy to calculate the travel distance and the incident angle. Now, we can write the travel distance r and incident angle α as follows:

$$r = [x^2 + (d_s + d_r)^2]^{1/2}, \tag{3}$$

$$\alpha = \tan^{-1} [x / (d_s + d_r)], \tag{4}$$

where x is the offset distance; d_s and d_r are source and receiver depths below the sea surface, respectively.

For the reflection from the sea bottom, the travel distance r_1 and incident angle α_1 can be determined by the same method. We can write out the equations easily by a small modification to equations (3) and (4):

$$r_1 = [x^2 + (2d_w - d_s - d_r)^2]^{1/2}, \tag{5}$$

$$\alpha_1 = \tan^{-1} [x / (2d_w - d_s - d_r)], \tag{6}$$

where d_w is the water depth.

For reflections which include ghosts at shot or

receiver or both shot and receiver, the problem seems complicated but still can be solved easily by the method of images. The number of ghost reflections observed at a hydrophone is three. Figure 17(a) shows the raypaths of these reflections from sea bottom, and Figure 17(b) shows the equivalent paths after applying the image technique, where, r_2 is equivalent to path S - surface - bottom - R; r_3 is equivalent to path S - bottom - surface - R; and r_4 is equivalent to path S - surface - bottom - surface - R. They can be expressed by the following equations:

$$r_2 = [x^2 + (2d_w + d_s - d_r)^2]^{1/2}, \quad (7)$$

$$r_3 = [x^2 + (2d_w - d_s + d_r)^2]^{1/2}, \quad (8)$$

$$r_4 = [x^2 + (2d_w + d_s + d_r)^2]^{1/2}. \quad (9)$$

Again, x, d_w, d_s and d_r are offset, water depth, source and receiver depths, respectively. Their corresponding incident angles are α_2, α_3 and α_4, which are given by

$$\alpha_2 = \tan^{-1} [x / (2d_w + d_s - d_r)], \quad (10)$$

$$\alpha_3 = \tan^{-1} [x / (2d_w - d_s + d_r)], \quad (11)$$

$$\alpha_4 = \tan^{-1} [x / (2d_w + d_s + d_r)]. \quad (12)$$

The above equations only work for reflections without

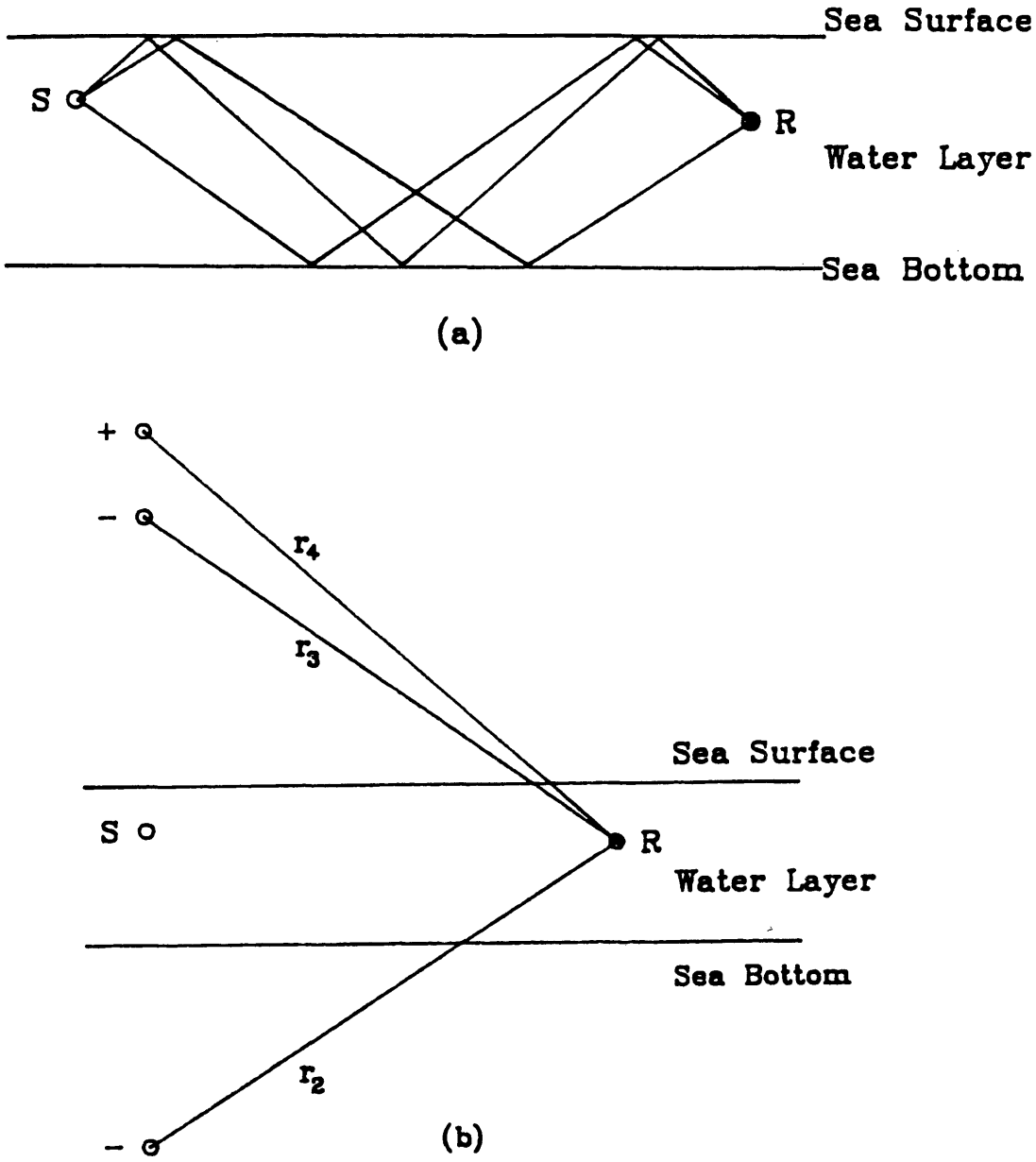

Figure 17. (a) The raypaths of ghost reflections from sea bottom, and (b) the equivalent paths after applying the image technique.

reverberations. In order to include reverberations into the model, the image technique is employed again. The results are obtained by simply modifying the above equations to following equations:

$$r_1 = \{x^2 + [2(n+1)d_w - d_s - d_r]^2\}^{1/2}, \quad (13)$$

$$r_2 = \{x^2 + [2(n+1)d_w + d_s - d_r]^2\}^{1/2}, \quad (14)$$

$$r_3 = \{x^2 + [2(n+1)d_w - d_s + d_r]^2\}^{1/2}, \quad (15)$$

$$r_4 = \{x^2 + [2(n+1)d_w + d_s + d_r]^2\}^{1/2}, \quad (16)$$

$$\alpha_1 = \tan^{-1} \{x / [2(n+1)d_w - d_s - d_r]\}, \quad (17)$$

$$\alpha_2 = \tan^{-1} \{x / [2(n+1)d_w + d_s - d_r]\}, \quad (18)$$

$$\alpha_3 = \tan^{-1} \{x / [2(n+1)d_w - d_s + d_r]\}, \quad (19)$$

$$\alpha_4 = \tan^{-1} \{x / [2(n+1)d_w + d_s + d_r]\}, \quad (20)$$

where n is the order of water-layer multiple. The other parameters have the same meanings as those defined before. Equations (13) to (20) are also suitable to those reflections without water-layer multiples provided n equal zero.

For the direct wave, the travel distance r_0 is easy to obtain. It is given by

$$r_0 = [x^2 + (d_r - d_s)^2]^{1/2}. \quad (21)$$

For the sea-bottom refraction, the travel distance r and incident angle α are as follows:

$$\alpha = \sin^{-1} (v_1/v_2), \quad (22)$$

$$r = x - (2d_w - d_s - d_r)\tan(\alpha) + (2d_w - d_s - d_r)/\cos(\alpha), \quad (23)$$

where v_1 is the velocity of sea water, and v_2 is the velocity of the sediments.

In some cases, the sediments are so thick that the reflection from the first subsurface interface is not within the early part of the near-shot trace, and the reflection from this interface is excluded from the possible waves in equation (2). If the influence from the first subsurface interface needs to be taken into account, the travel distance and the incident angle should be calculated first. Again, the image method is applied to the raypath within the water layer. Let w_1, w_2, w_3 and w_4 represent the total vertical travel distances within water layer for waves without ghosts, with shot ghost, with receiver ghost, and with both shot and receiver ghosts, respectively. The vertical travel distances are given by

$$w_1 = 2(n+1)d_w - d_s - d_r, \quad (24)$$

$$w_2 = 2(n+1)d_w + d_s - d_r, \tag{25}$$

$$w_3 = 2(n+1)d_w - d_s + d_r, \tag{26}$$

$$w_4 = 2(n+1)d_w + d_s + d_r. \tag{27}$$

If the thickness of the sediment is z, then the travel distance are given by

$$r_i = w_i/\cos(\alpha_i) + 2z/\cos(\beta_i), \qquad i=1,2,3,4, \tag{28}$$

where α and β are the incident angle in water and the transmitted compressional angle in the sediments. The angles α and β can be determined by the following equations:

$$x = w_i \tan(\alpha_i) + 2z \tan(\beta_i), \tag{29}$$

$$\sin(\beta_i) = (v_2/v_1) \sin(\alpha_i), \qquad i=1,2,3,4. \tag{30}$$

Equations (29) and (30) are solved numerically to obtain α_i and β_i.

Travel Time and Amplitude

To simulate the early part of near-shot trace based on equation (2), the travel time and amplitude for each possible wave must be known. The travel time is a function of travel path and velocities of the mediums passed by the wave. The amplitude of wave depends on the travel distance

and reflection coefficients. The reflection coefficients are obtained by solving the Zoeppritz equations for different incident angles.

For those waves traveling inside the water layer, the travel time is obtained by simply dividing the total travel distance by the velocity of the water, regardless of how complicated the travel path. For those waves traveling through the water layer into the sediments, the travel time consists of two parts, one for water and the other for sediments. The travel time of sea-bottom refraction is given by

$$T = [x - (2d_w - d_s - d_r)\tan(\alpha)]/v_2 + (2d_w - d_s - d_r)/v_1 \cos(\alpha), \quad (31)$$

where T is the travel time. All other parameters have the same meanings as those in last section. For reflections from the first subsurface, the travel time T_i is given by

$$T_i = w_i/v_1 \cos(\alpha_i) + 2z/v_2 \cos(\beta_i), \quad i=1,2,3,4, \quad (32)$$

where, i=1,2,3,4 represent waves without ghost, with shot ghost, with receiver ghost, and with both shot and receiver ghosts. All parameters have been defined in last section.

In a homogeneous medium (water layer), seismic waves from a point source are spherical, and the amplitude

decreases as the first power of the travel distance according to spherical divergence theory. For waves travelling inside the water layer, the amplitude A is determined by the travel distance r and the reflection coefficient R_0 of the sea-bottom which varies with the incident angle. Amplitude A can be expressed by

$$A = (-1)^i R_0^j / r, \tag{33}$$

where i is the order of reflection from sea surface and j is the order of reflection from sea bottom. Equation (33) works for all waves travelling inside the water layer, provided i, j, R_0 and r are changed according to the raypaths of the waves.

For reflections from the first subsurface interface, the transmission loss at the base of the water should be taken into account. The amplitude is given by

$$A = (-1)^i (1-R_0^2) R_0^j R_1 / r, \tag{34}$$

where i is the order of reflection from sea surface, j is the order of water-layer multiples, R_0 and R_1 are the reflection coefficients of sea bottom and first subsurface, and r is the total travel distance. Equation (34) is applicable to all waves reflected from the first subsurface. The amplitude loss due to attenuation of the

sediments is not included in equation (34).

Wavelet Estimation from Marine Data

In the last few sections, equation (2) was introduced to describe the amplitude and time delay of all waves that create the initial part of the seismic data. Then equations for calculating the travel distance, incident angle, amplitude and travel time are given for all possible waves that are stacked in equation (2).

Now, it is time to introduce a new method of estimating the airgun array signature from marine seismic data. Figure 18 is a flow-chart which summarizes the main steps in wavelet estimation program (Appendix B). The algorithm in the program is similar to that used for simulating the far-field signature, which is introduced in an earlier section of this Chapter. Again, a scan method is employed. In order to make it easier to understand the principle and algorithm of the wavelet estimation program, the calculation steps will be introduced in detail.

1) According to the field hydrological and geological conditions and the field parameters, determine which waves will arrive at the near-shot receiver within a given time beyond the first break. Set up a model; read in the field parameters; give initial values to phase ϕ, frequency f, damping coefficient α and

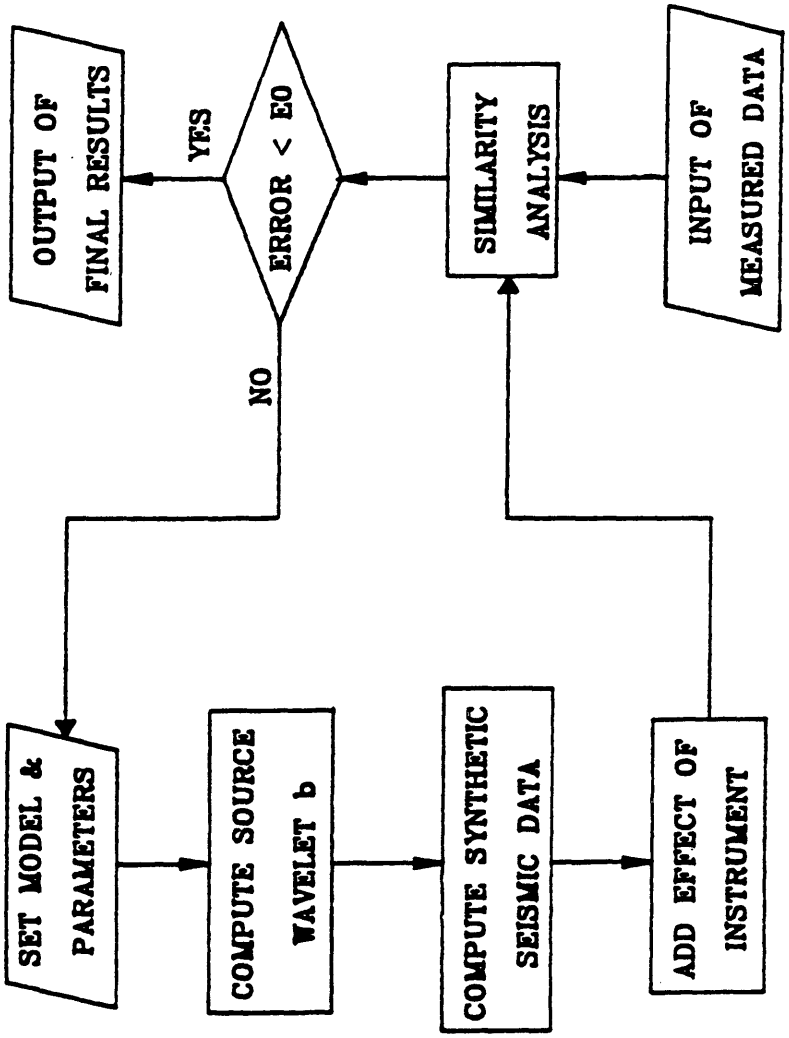

Figure 18. A flow-chart summarizing the main steps in wavelet estimation technique in this study.

error threshold ERR0.

2) Based on equations introduced in last two sections, compute travel distances, incident angles, amplitudes and travel times for all possible waves determined in step 1.
3) Based on equation (1), compute the source wavelet b(t) using a selected frequency, phase, and damping coefficient.
4) According to equation (2), generate synthetic data (composite wave) by stacking together all possible waves which have their own distinct amplitudes and travel times.
5) Take the impulse response of the recording system into account by convolving its impulse response with the synthetic data.
6) Do similarity analysis between synthetic data and real seismic data from a near-shot trace for a given time duration out to about 100 ms beyond the first break; calculate the error between the synthetic and real data using ERR = 1 - CC.
7) Compare the error ERR with its threshold value given by ERR0. If ERR is less than or equal to ERR0, normalize the source wavelet and synthetic data, and write them into data files. If ERR is larger than

ERR0, then modify parameters f, ϕ and α; go to step 3 and repeat all calculations from step 3 to step 7. Repeat this procedure until ERR is less than or equal to ERR0.

The hydrological and geological conditions include water depth, thickness, density, and velocity of sediments; field parameters include the shot depth, receiver depth and offset. The velocity and thickness of the sediments can be obtained by velocity analysis; the density of the sediments may be obtained from well-log data or other methods, and the water depth is determined by a fathometer during the field survey.

In order to prove the validity of the method and to check the accuracy of the results, the estimation method is applied to GECO test 5 marine data. The data are obtained by using a 959 cu.in. airgun array, a DFS-V recording system with filter passband 8/18--256/75, with source depth 7.5 meters, cable depth 10 meters, near-shot offset 298 meters, shotpoint interval 50 meters and receiver-group interval 25 meters. The water depth was not provided, so it was estimated using spectral analysis of a window from the data. The estimate was 50 meters. Figure 14 is a near-shot trace gather from 24 shots, and Figure 19 is a 24-trace common-shot gather from shot #2. The waveform of the early

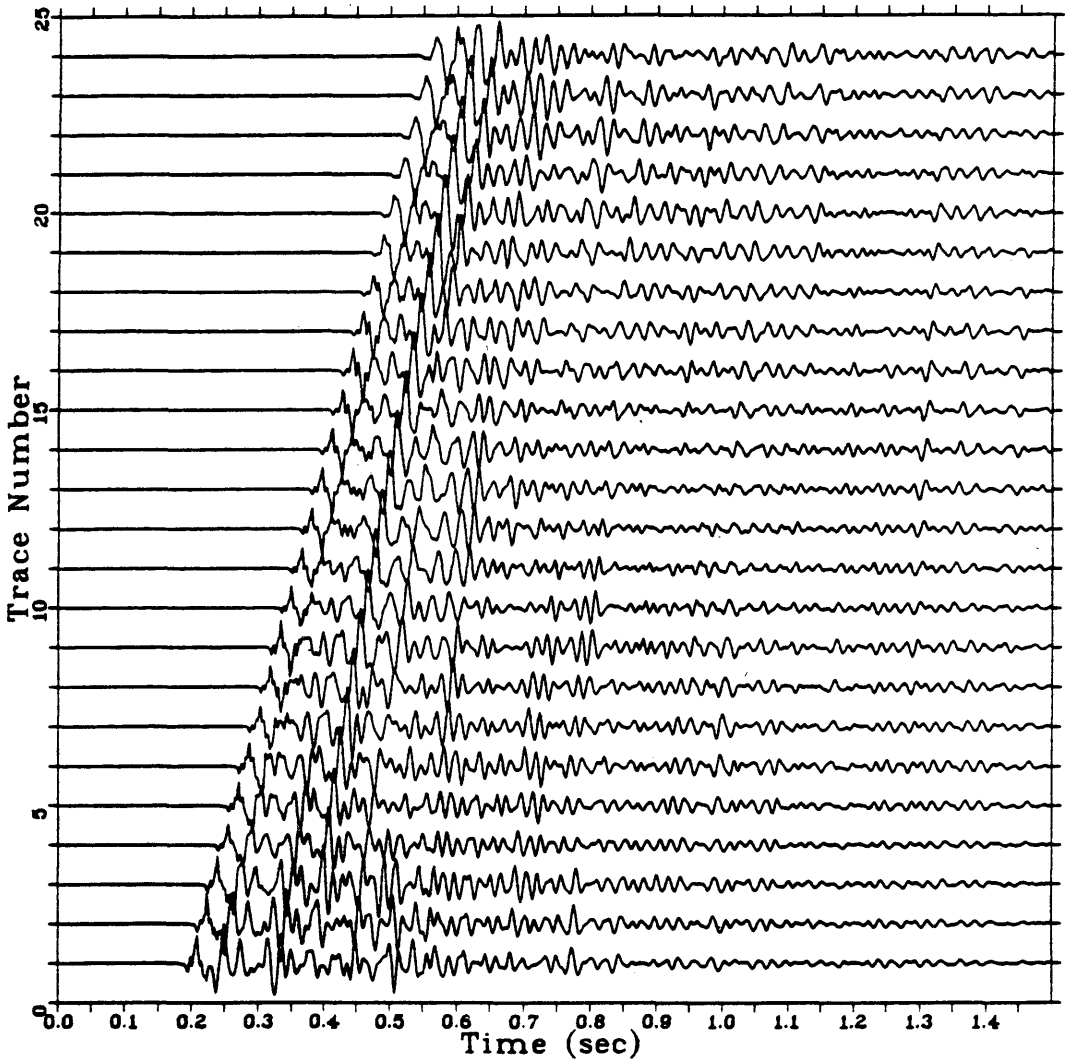

Figure 19. A common-shot gather of 24 traces from GECO test 5 marine data, displayed up to 1.5 seconds.

part varies from trace to trace in the common shot gather, but the waveform of the early part of the near traces is very stable for about 85 ms beyond the first break. The reason for the waveform change in the common-shot gather is that reflections from the subsurface affect the early part of seismogram for large offset. To avoid this effect, the near-shot trace of shot #2 is used for wavelet estimation.

A good result is obtained by applying the estimation method introduced in this section to the near-shot trace of shot #2. The synthetic data (dashed curve in Figure 20) matches closely the measured data (solid curve) for a length of 83 ms from the first break. A length of 186 ms before the first break is removed for plotting purpose. The CC value between the synthetic data and measured data in Figure 20 is 94.4%. The corresponding estimated airgun array signatures are shown in Figure 21, where the solid curve represents the wavelet without the effect of DFS-V instruments, and the dashed curve represents the wavelet including the effect of the instruments.

If the same airgun array is shot at the same depth in a deep water area and a hydrophone is put 100 meters below the center of the array, then the far-field signature is obtained. Figure 22 shows such a far-field signature before (solid curve) and after (dashed curve) passing

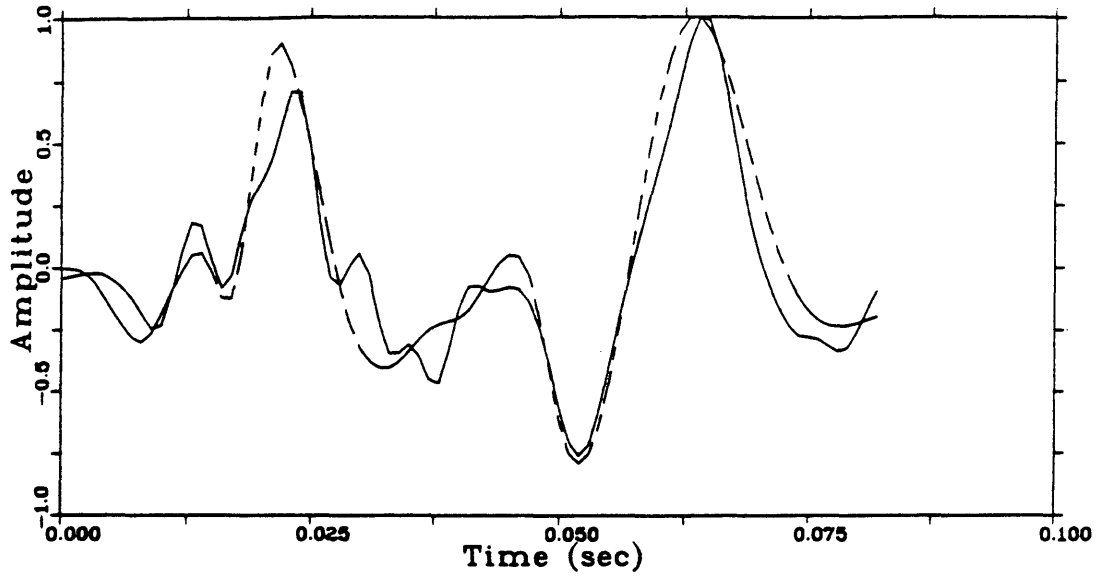

Figure 20. The early part (begining from first break) of near-shot trace for shot #2 of GECO test 5, solid for measured and dashed for synthetic.

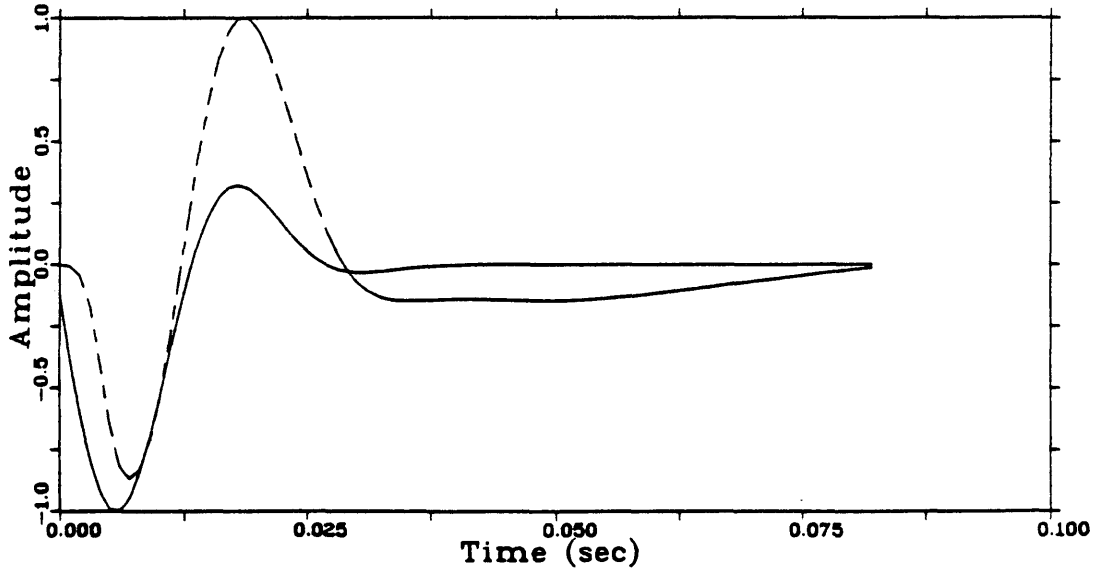

Figure 21. Airgun array signature estimated from shot #2 of GECO data, without (solid) and with (dashed) effect of DFS-V instruments.

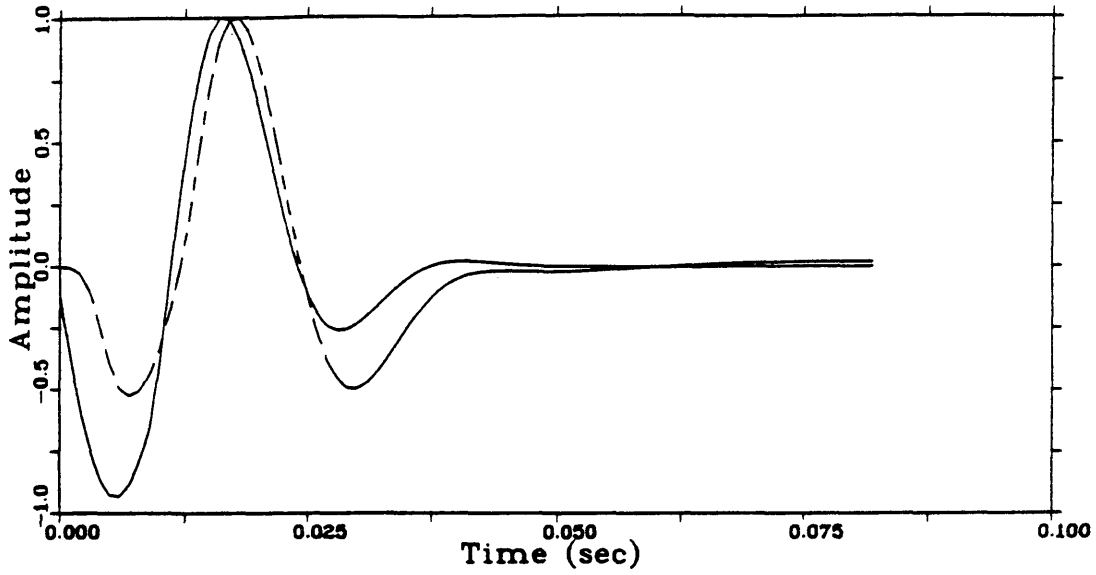

Figure 22. Estimated far-field signature before (solid) and after (dashed) passing through DFS-V instruments.

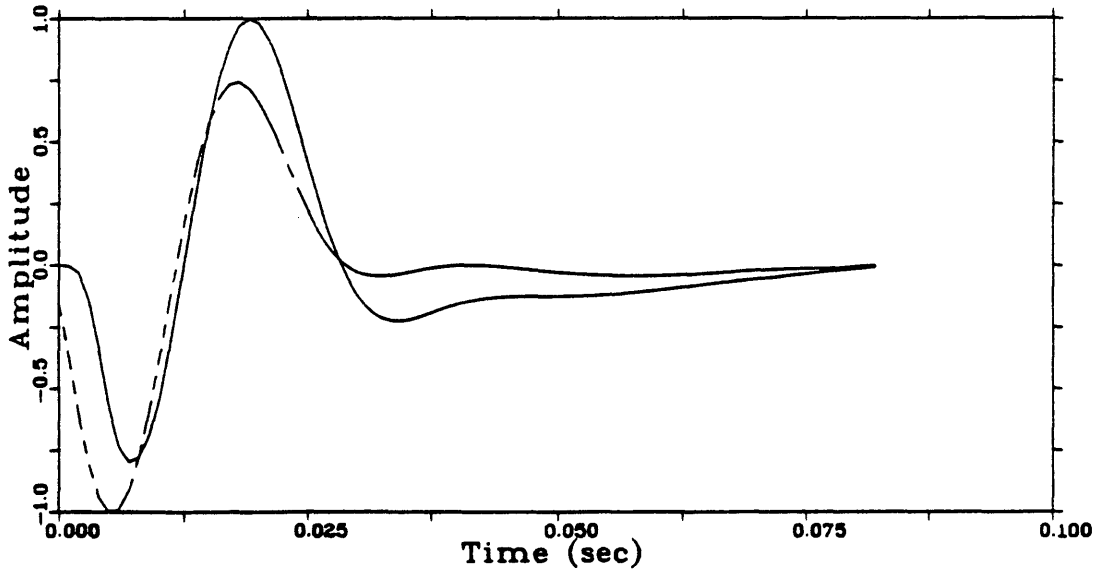

Figure 23. Comparison of estimated signature (solid) and its minimum-phase equivalent (dashed), with DFS-V effect.

through the DFS-V instruments by computer simulation, where the delay time (66 ms) is removed.

The Wiener transform is applied twice to the estimated signature after passing through the DFS-V instruments to get the minimum-phase equivalent, and the result (dashed curve) is superimposed with the estimated signature (solid curve) in Figure 23. The CC value between two curves is 86.4%.

Theoretically speaking, the CC value between the synthetic data and measured data should be equal to 100% provided the model is correct and all factors are taken into account. In practice, some factors, like rough sea surface caused by wind, are not easy to estimate. Therefore, it is impossible to include all affecting factors into the simulation. In this study, the geometrical effects of the source and receiver arrays are not taken into account, and the effect of the residual bubble pulse is not included. Therefore, some small error will exist, but the estimation result is still very accurate.

To show that the estimated wavelet is correct, wavelet processing is applied to GECO marine data, and the estimated wavelet is used for phase compensation. This will be discussed in next Chapter.

A NEW WAVELET PROCESSING METHOD

According to the discussion in the previous Chapter, conventional wavelet processing methods improve the quality of seismic data, but the passband of the d-filter is not easy to choose. In order to solve the problem, a new method which uses a matched filter is proposed. To show the advantage of this method over the conventional methods and to prove the validity of the estimated wavelet, both conventional methods and the new method are applied to GECO Test 5 marine data and to synthetic data.

The Convolution Model

In analyzing the seismic process, we can express the seismic data as a convolutional model given by

$$s = (b * i * e * g * m * p * r)a + n, \qquad (35)$$

where, s is the seismic data; b is the source wavelet; i is the impulse response of the instrument; e is the Earth attenuation; g is the ghost operator; m is the complete set of multiples and reverberations; p is the filtering effect caused by processing; r is the reflectivity function; a is the amplitude loss due to spherical divergence, and n is additive noise. The main purpose of seismic exploration is to obtain the reflectivity function r. To achieve this

purpose, the other components in equation (35) must be removed or suppressed as much as possible. Wavelet processing is a powerful tool to do this work.

A Wavelet Processing Method

Based on the convolution model, equation (35), a new wavelet processing method is proposed. Different from conventional methods, this method avoids the difficulty of choosing the bandlimiting final display filter, called d-filter, to preserve the signal and suppress the noise after deconvolution. A matched filter, which is the zero-phase equivalent of the source wavelet after passing through the instruments, is used here as the d-filter to suppress the noise which is amplified by spiking deconvolution. Theoretically speaking, this new method is superior to the conventional methods in the sense of suppressing noise and preserving high frequency components as much as possible.

The new wavelet processing method includes the following steps:

1) Suppress additive noise by velocity filters or by muting (cut-off method).
2) Compensate for the effects of attenuation and spherical divergence.
3) Estimate the source wavelet by the scan method.

4) Calculate the phase-compensator from the estimated source wavelet after passing through the instruments.
5) Apply the phase-compensator to the seismic data to make them minimum phase.
6) Apply spiking deconvolution to seismic data.
7) Compute the matched filter from source wavelet after passing through the instrument by using FFT algorithm.
8) Convolve deconvolved seismic data with the matched filter.

Steps 5 and 6 are interchangeable in order.

To get a clear idea about the method, some steps are explained in following sections.

Suppression of Additive Noise

Additive noise is split into two components: coherent noise and random noise. In reflection prospecting, direct wave, sea-bottom refraction, cable motion, etc. belong to coherent noise. Microseisms, wind noise, sea wave, instrument noise, etc. are random noise.

The performance of spiking deconvolution is degraded by the presence of noise, so every effort that is economically feasible should be made in acquisition and preprocessing to reduce the noise level before deconvolution. Berkhout

(1977) shows the degradation of performance under a variety of noise conditions in the data (Sengbush et al, 1986).

Because direct waves and sea-bottom refractions appear in the early part around the first breaks, they can be removed by the cut-off method (changing the values at the sampling points to zero), or their effect can be avoided by choosing a window excluding these waves when computing the deconvolution operator. The cable motion noise can be suppressed by velocity filters because the apparent velocity of cable motion is very different from those of primary reflections. Random noise can be suppressed by stacking.

Amplitude Compensation

Spherical divergence and attenuation cause amplitude loss, and attenuation also produces time variance. One of the assumptions for statistical deconvolution is that the effective seismic pulse is time invariant and minimum phase. Therefore, spherical divergence and attenuation degrade the performance of deconvolution. To reduce their effects, amplitude restoration may be incorporated as a data processing step.

Spherical divergence is caused by the spherical spreading of the wavefront. When a medium is disturbed by a point source, the disturbance travels outward with a

spherical wavefront, provided the velocity is constant. Since the total energy on the ever-increasing sphere is constant, the energy density (energy per unit area) must decrease in proportion to the total area of the sphere. Thus, the energy of the pulse at any point on the sphere decreases as the square of the distance from the source. Since energy is proportional to the square of the amplitude, the amplitude decreases as the first power of the distance (Sengbush, 1983, p.38). The above theory is also suitable to a layered earth (Newman,1973), provided the velocity is rms velocity.

Attenuation is a frequency dependent energy loss. When a seismic wave travels through a medium, it causes the relative movement of particles of the medium, and this relative movement produces friction between particles. The friction of particles changes part of the seismic energy into heat energy and causes some energy loss. The energy loss is proportional to vibration velocity of particles, which in turn is proportional to the frequency of the seismic wave. This frequency-dependent energy loss causes amplitude decay and change of waveform with distance traveled. A linear attenuation model matches experimental results very closely (Sengbush, 1983, p.39). In the model, the amplitude loss is given by $\exp(-\gamma x)$, where the

attenuation factor γ is a function of frequency and x is the distance traveled.

In seismic data processing, it is difficult to correct for time variance due to attenuation because the frequency dependent attenuation factor varies from rock to rock and is not easily obtained. To reduce the effect on deconvolution, a window from the seismic trace is used to compute the operator. The amplitude loss due to spherical divergence and attenuation is usually compensated by an exponential compensator $Aexp(\alpha t)$, where t is travel time; A and α are constants determined from seismic data. However, as pointed out by Ziolkowski (1984, p.35), this amplitude restoration introduces distortion into the effective seismic pulse. Up to now, no method has been found to correctly compensate for divergence and attenuation.

Phase Compensation

One of the assumptions for statistical deconvolution is that the effective seismic pulse is minimum phase and time invariant. As we have discussed in the previous Chapter, the source wavelet and the impulse response of the instrument are probably not minimum phase, and this will introduce phase distortion in the deconvolved data. To solve this problem, phase compensation must be applied to seismic data either before or after spiking deconvolution.

To compute the phase compensator, all nonminimum-phase components must be known. The impulse response of the instruments can be obtained by measurement, but the source wavelet must be estimated from seismic data if direct measurement is impossible. A wavelet estimation method has been proposed in the last Chapter.

Phase compensation includes two steps:

1) Compute the allpass phase compensator g, called the Wapco operator (Fourmann, 1974), by convolving y with the Wiener transform of y, i.e., g=y*W(y), where y is the estimated source wavelet after passing through the instrument: y=b*i. The allpass compensator g is stable and its phase spectrum is the same as the phase distortion introduced by spiking deconvolution of the seismic data.
2) Apply phase compensation to the seismic data by correlating the seismic data s with the phase compensator g either before or after spiking deconvolution. This is equivalent to convolving s(t) with g(-t).

After the above processing, the effect caused by nonminimum phase components is removed.

Matched Filtering

After spiking deconvolution and phase compensation, the

seismic pulse has broad and flat amplitude spectrum and correct arrival time, and the resolution of the data is improved, but noise dominates the spectral ranges where the signal-to-noise ratio is poor. The reason is that the Wiener process amplifies not only signal but also noise. To reduce the level of noise amplified by spiking deconvolution, some type of bandlimiting filter must be applied to the deconvolved data.

As the final step of conventional wavelet processing, a zero-phase bandpass filter is used as the d-filter to restrict the bandwidth of the deconvolved data. However, the results will depend on the passband of the d-filter. If the passband of the filter is too narrow, some high frequency components of the signal will be lost. If the passband is too wide, the noise, which has been amplified by spiking deconvolution, will not be suppressed very well.

In this study, a matched filter, which is the zero-phase equivalent of the source wavelet after passing through the instrument, is proposed as the d-filter. Because this filter has the same amplitude spectrum as the seismic pulse, it is superior to other filters in the sense of preserving high frequency components of the signal and suppressing noise. The zero-phase property of the filter will insure that the filtered data will have the correct

arrival time.

The matched filter is easily obtained by using the Fourier transform. First, compute the Fourier transform of the estimated source signature after passing through the instruments; then compute the amplitude spectrum; finally, apply the inverse Fourier transform to the amplitude spectrum to obtain the zero-phase equivalent, which is the matched filter.

As the final step of the new method, the deconvolved and phase-compensated data is filtered by the zero-phase matched filter, instead of a bandpass d-filter used in conventional methods.

Wavelet Processing of Marine Data

To check the effectiveness of the new wavelet processing method and the accuracy of the estimated airgun array signature, both conventional methods and the new method are applied to GECO Test 5 marine data. The field parameters for the data have been given in the last Chapter. The processing results will be discussed in this section.

The near-shot trace gather. Figure 24 shows a near-shot trace gather after amplitude compensation by an exponential compensator. Comparing this figure with Figure

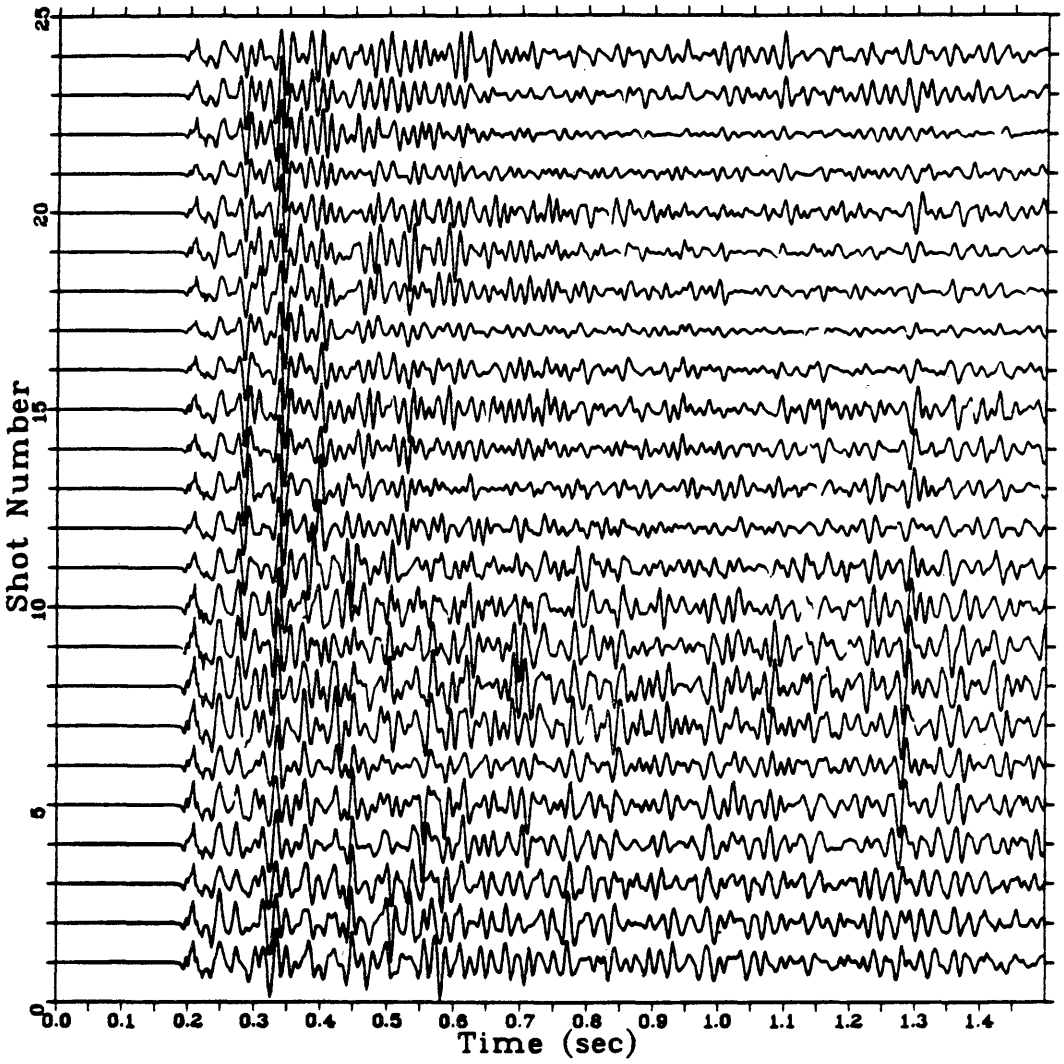

Figure 24. Amplitude compensated near-shot trace gather of GECO test 5 marine data.

14 (the same data before amplitude compensation), we can see that the deep reflections in Figure 24 are much clearer than those in Figure 14. Reverberations dominate the middle part of the data, and this indicates the necessity for wavelet processing.

Spiking deconvolution is applied to the near-shot trace gather, and the results are shown in Figure 25. Deconvolution operators of length 150 ms are calculated trace by trace to take care the variations in the different traces, and a window for one second in length is selected from 0.5 to 1.5 seconds to avoid shallow effects. The operator is then applied to the corresponding trace begining from 0 to 1.5 seconds. From Figure 25, we can see that the spiking deconvolved data have higher resolution but contain more noise, especially high-frequency random noise around the first break. This high-frequency noise has been greatly amplified by spiking deconvolution because the signal-to-noise ratio is very low in the high frequency area. This result indicates the necessity of applying a bandlimiting filter to the deconvolved data.

As one of the conventional methods does, a bandpass filter is applied to the data directly after spiking deconvolution. To see the difference caused by the passband, a narrow passband with (0, 10 - 55, 70) Hz

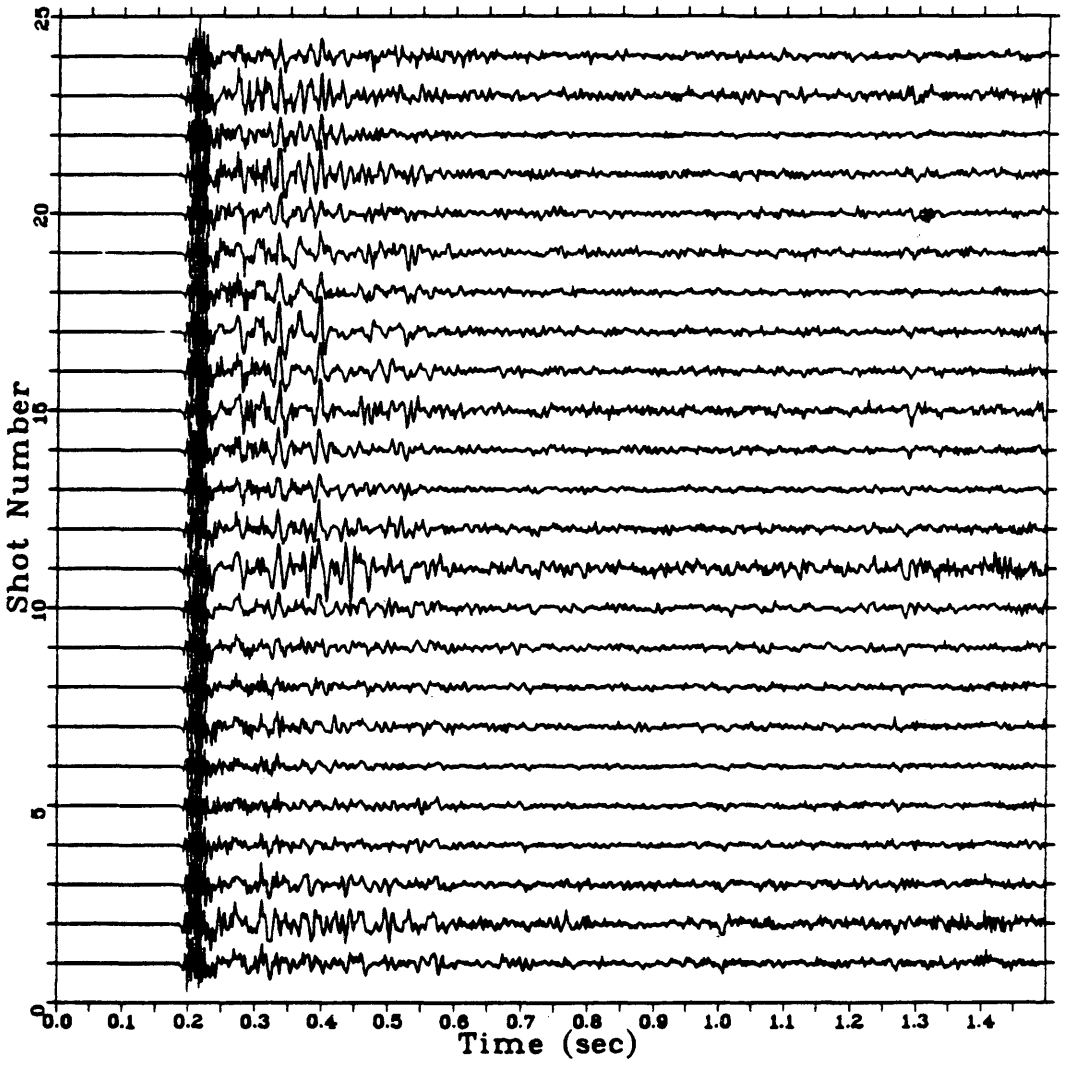

Figure 25. Spiking deconvolution of the near-shot trace gather of Figure 24.

(Figure 26) and a wide passband with (0, 10 - 100, 120) Hz (Figure 27) are used. The filtered results are shown in Figure 28 for narrow passband and Figure 29 for wide passband. The narrow-band filtered data have a clear background and clear reflections in the middle and deep parts of the section. The wide-band filtered data have high resolution but noisy background. Comparing Figures 28 and 29 with Figure 24, we can see that the reverberations in the original data (Figure 24) have been suppressed by the processing. Therefore, this simple method really improves the quality of seismic data.

Because the source signature after passing through the instrument is nonminimum phase (Figure 23), spiking deconvolution introduces phase distortion to the data. Thus, the arrival times of reflections in Figures 28 and 29 are incorrect.

Both the wavelet without shot ghost (dashed curve in Figure 21) and the wavelet with shot ghost (dashed curve in Figure 22) are used for phase compensation. Phase compensation is applied to deconvolved data, followed by bandpass filtering. Again, (0, 10 - 55, 70) Hz and (0, 10 - 100, 120) Hz filters are used when filtering. The results are shown in Figures 30 to 33. Figures 30 and 31 are the data filtered narrow band and phase-compensated based on

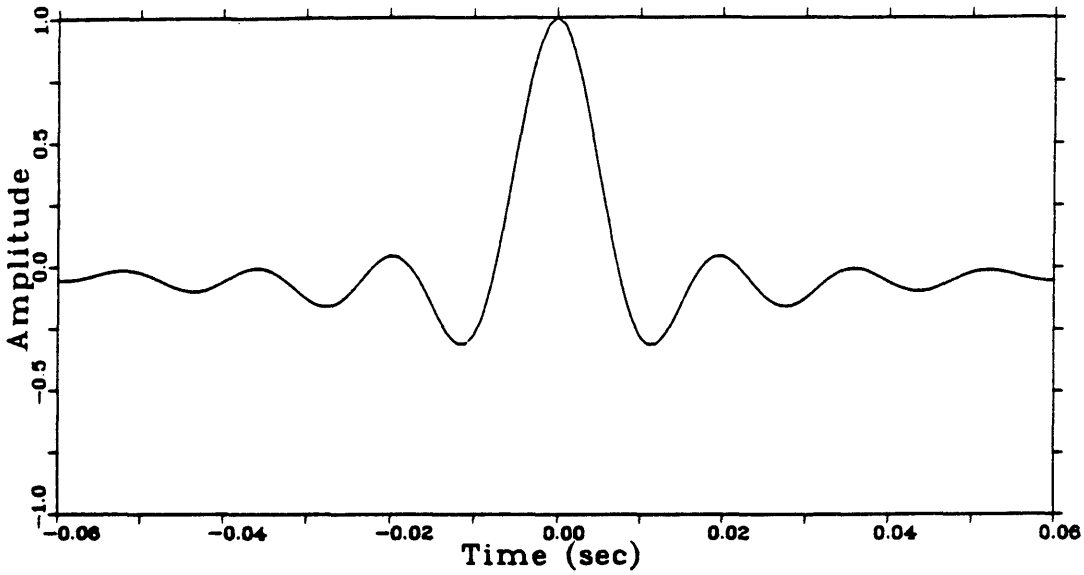

Figure 26. Impulse response of a narrow-band filter with (0,10-55,70) Hz passband.

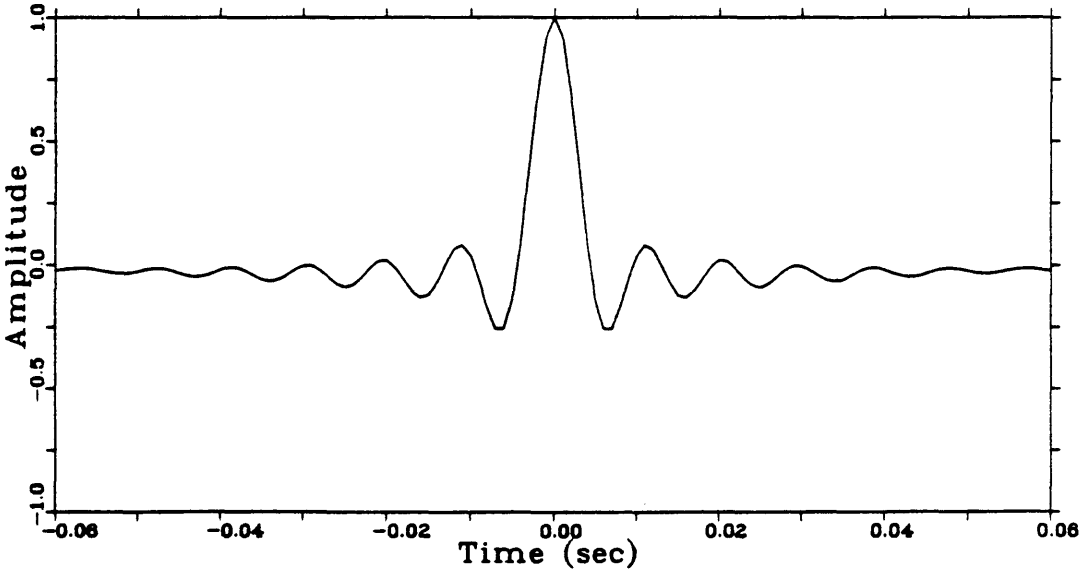

Figure 27. Impulse response of a wide-band filter with (0,10-100,120) Hz passband.

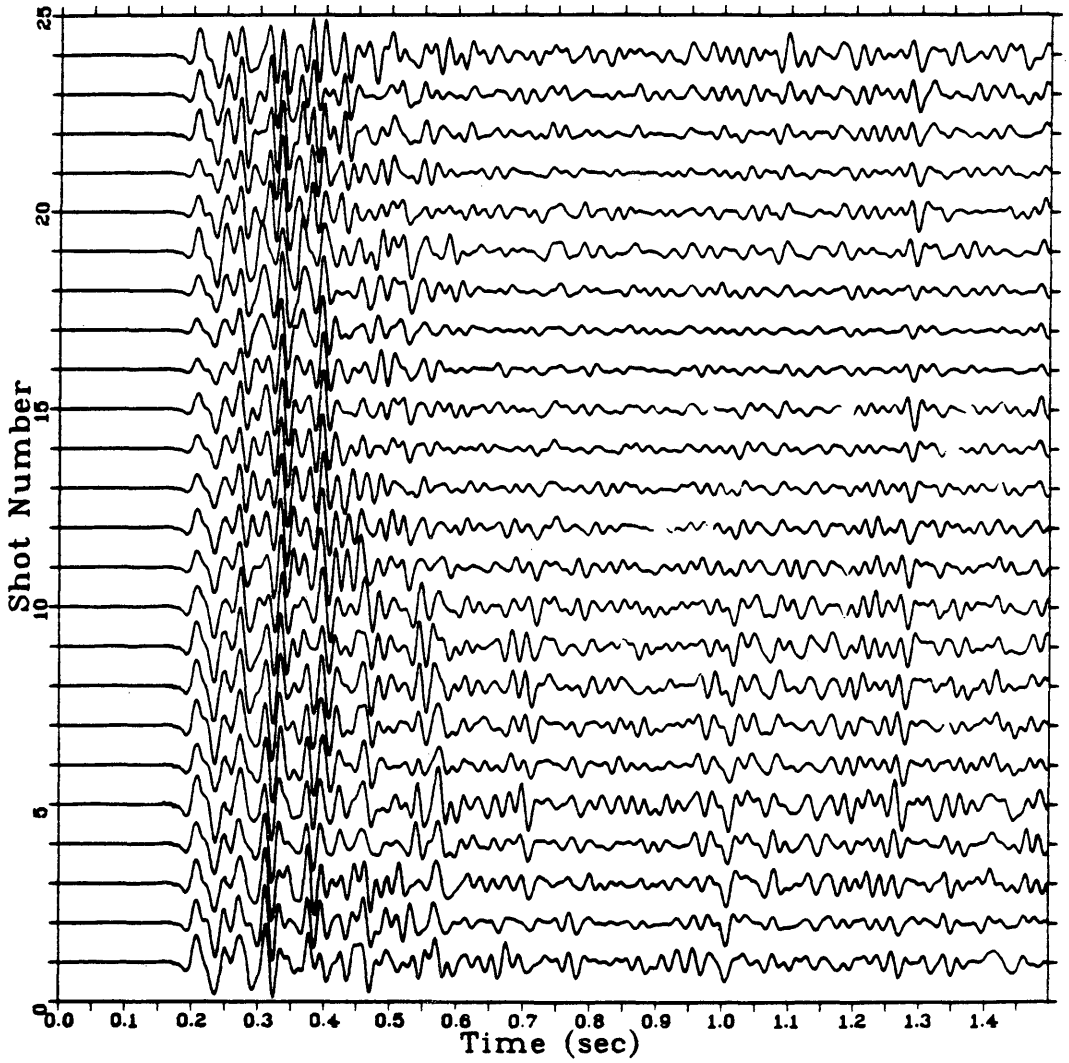

Figure 28. Deconvolved near-shot trace gather after narrow-band filtering.

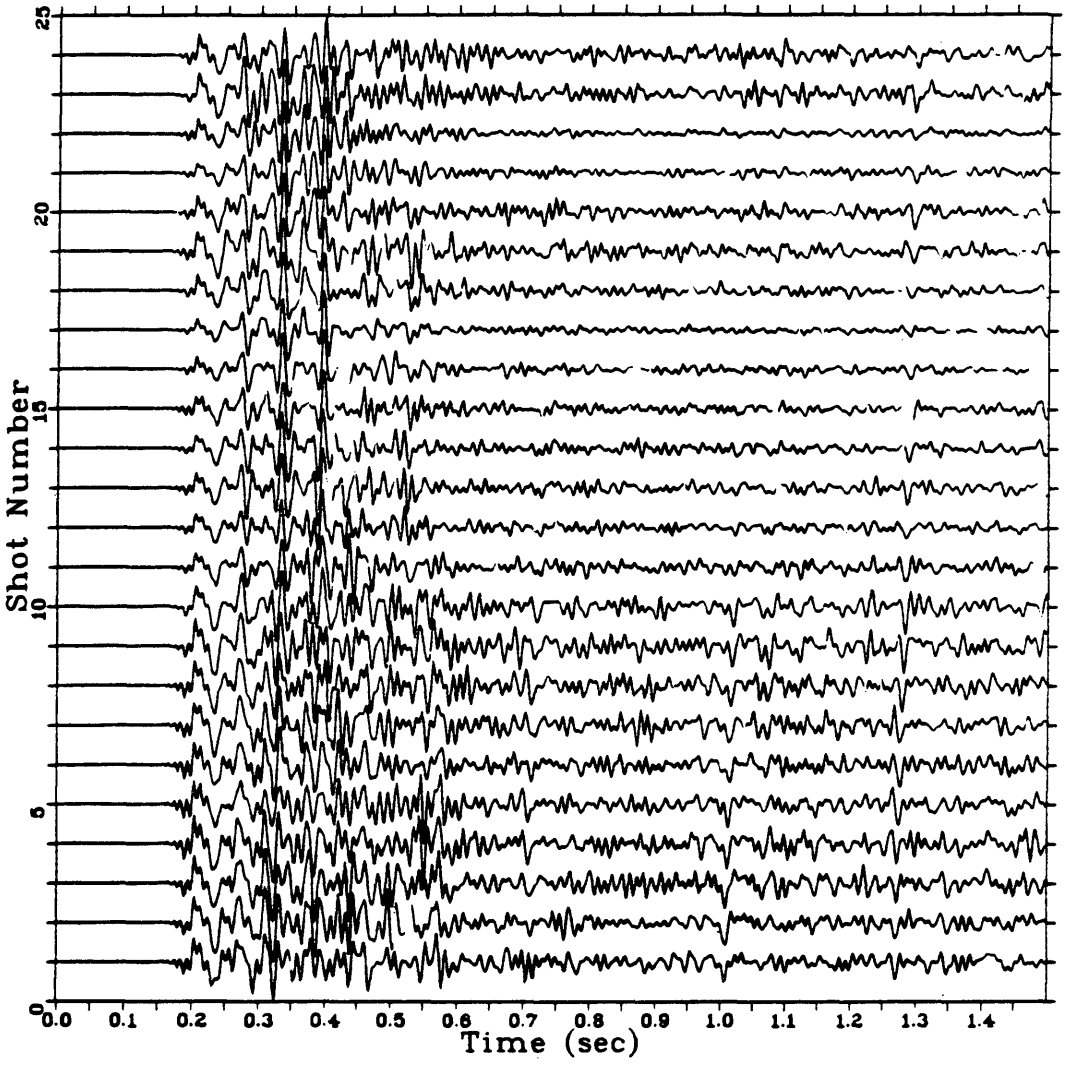

Figure 29. Deconvolved near-shot trace gather after wide-band filtering.

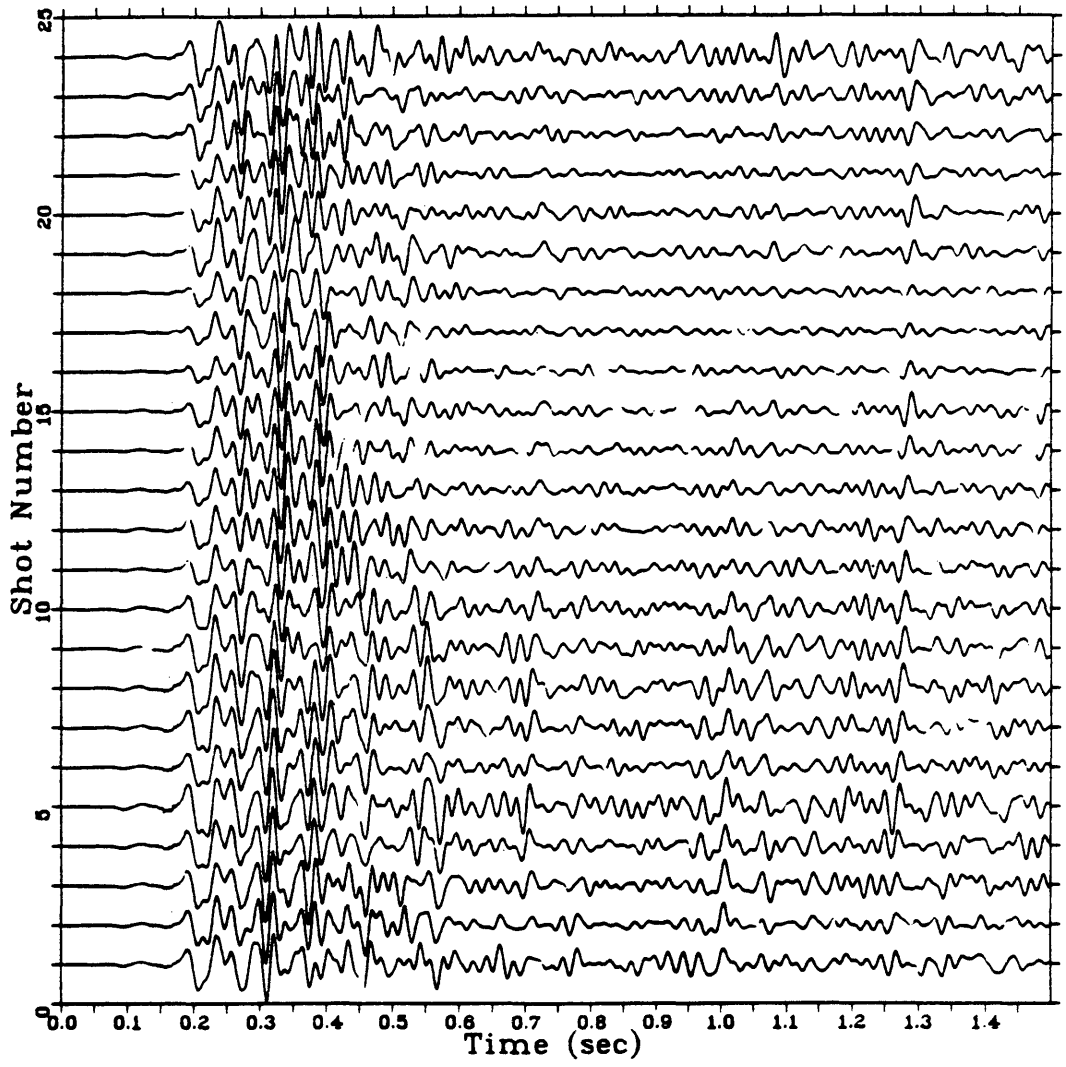

Figure 30. Deconvolved near-shot trace gather after narrow-band filtering and phase-compensation based on wavelet without ghost.

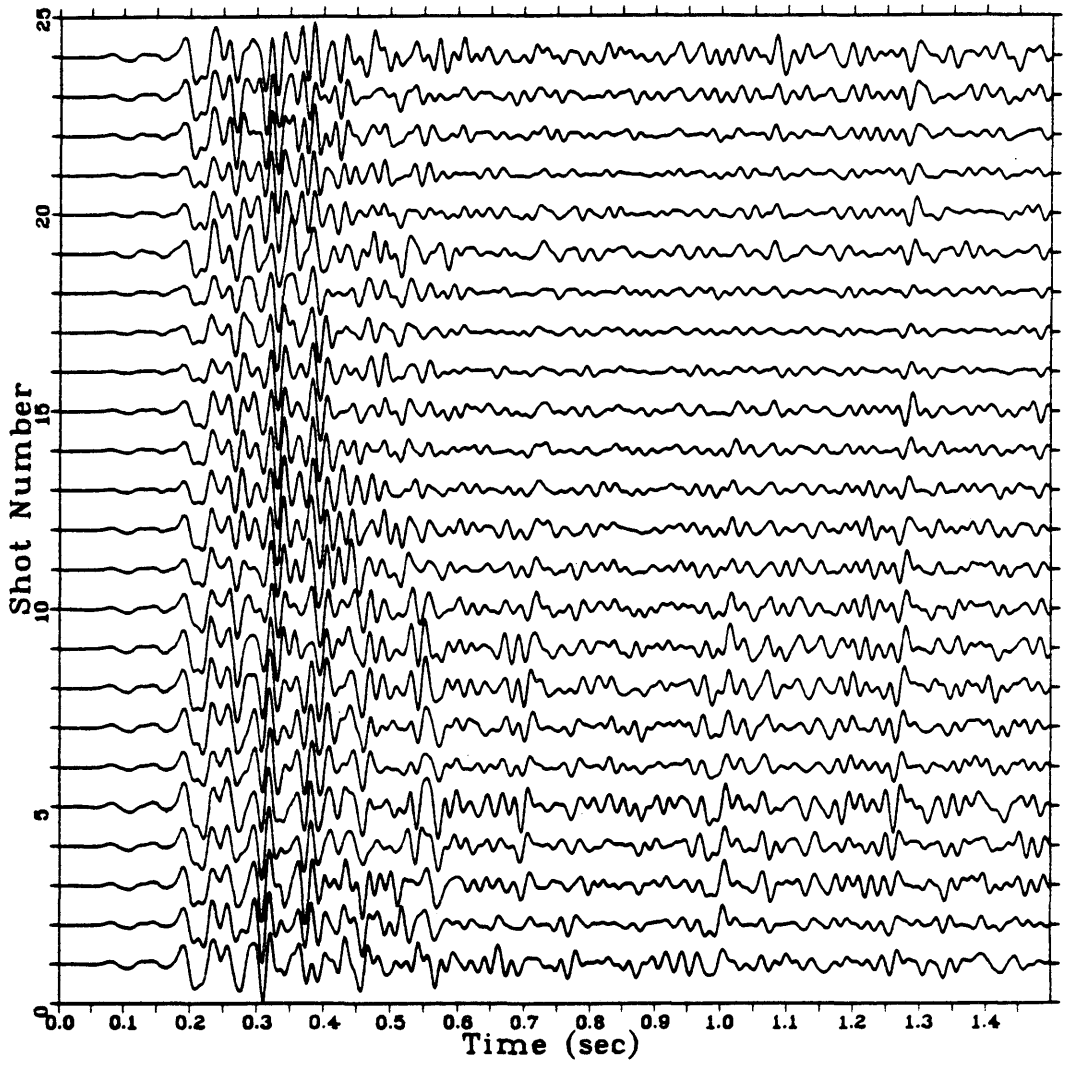

Figure 31. Deconvolved near-shot trace gather after narrow-band filtering and phase-compensation based on wavelet with shot ghost.

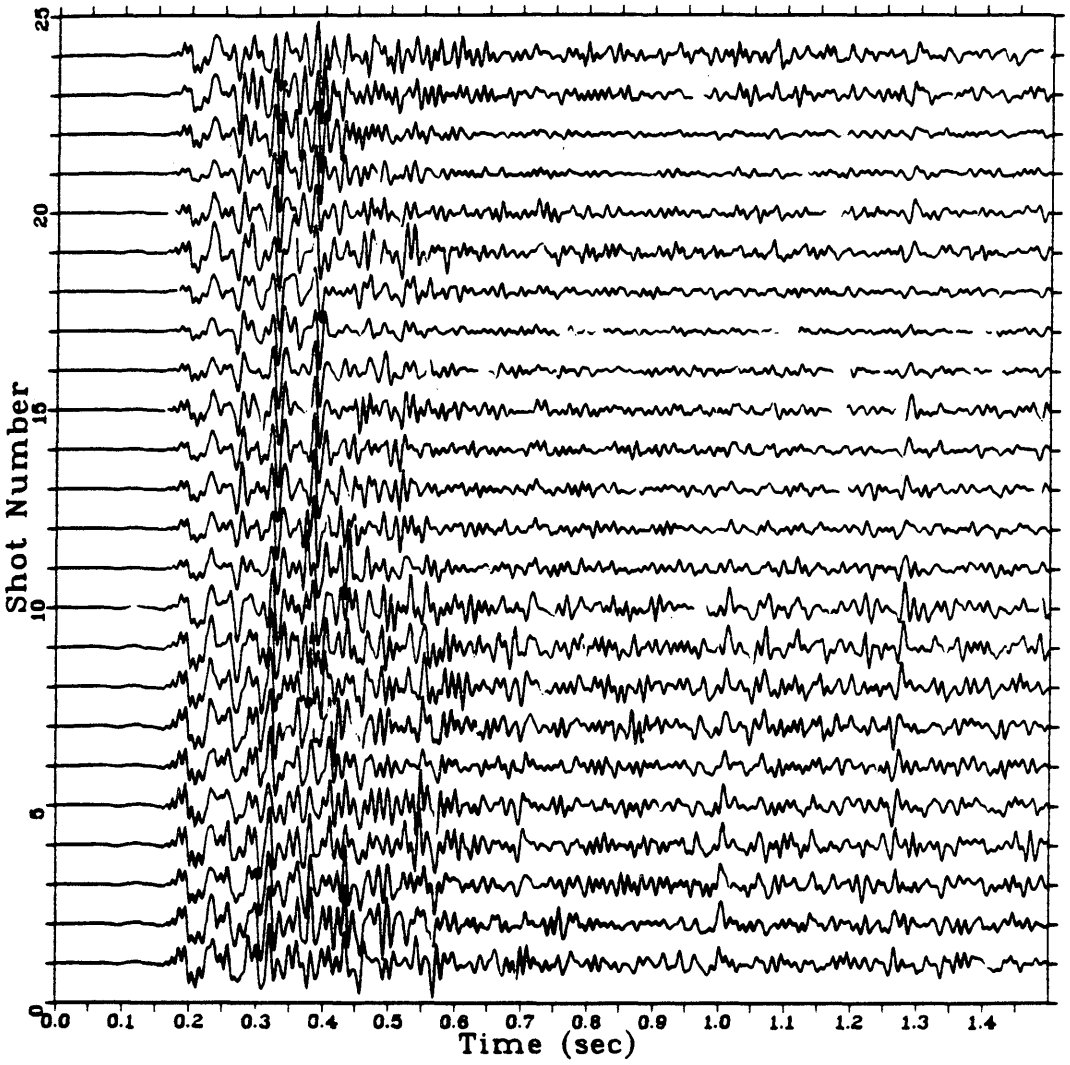

Figure 32. Deconvolved near-shot trace gather after wide-band filtering and phase-compensation based on wavelet without ghost.

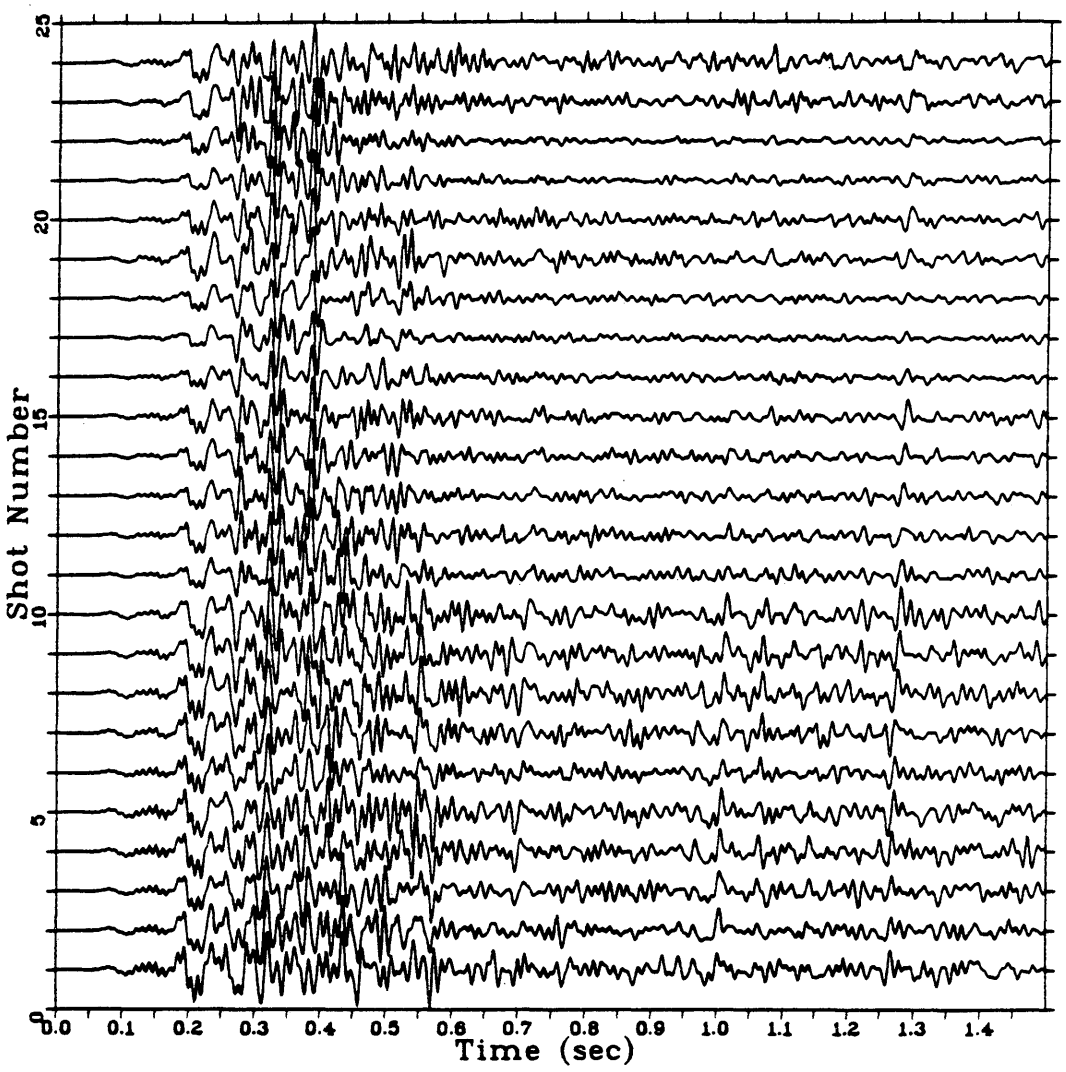

Figure 33. Deconvolved near-shot trace gather after wide-band filtering and phase-compensation based on wavelet with shot ghost.

wavelets without ghost and with shot ghost, respectively. Figures 32 and 33 show the results filtered wide band and phase-corrected based on wavelets without ghost and with shot ghost, respectively. The data phase-compensated based on the wavelet without ghost are almost identical to those based on the wavelet with shot ghost, provided the filters are the same. Again, the narrow passband produces a clear background and more continuous reflections in the deep part of the section, and the wide passband produces high resolution data with noisy background. By comparing Figures 30 and 31 with Figure 28 and Figures 32 and 33 with Figure 29, we can see that phase-compensation reduces the arrival time of all reflections for about 20 ms, and the seismic data after phase compensation have more continuous deep reflections. These results indicate the importance of phase-compensation and prove the validity and accuracy of the estimated airgun signature.

Figure 34 shows the matched filter calculated from the wavelet without ghost, and Figure 35 shows the matched filter calculated from wavelet with shot ghost. These filters are very similar in period, and different only in the amplitude of the sidelobes. Comparing the matched filters with the bandpass filters (Figures 26 and 27), we find that the waveform of the matched filters is preferable

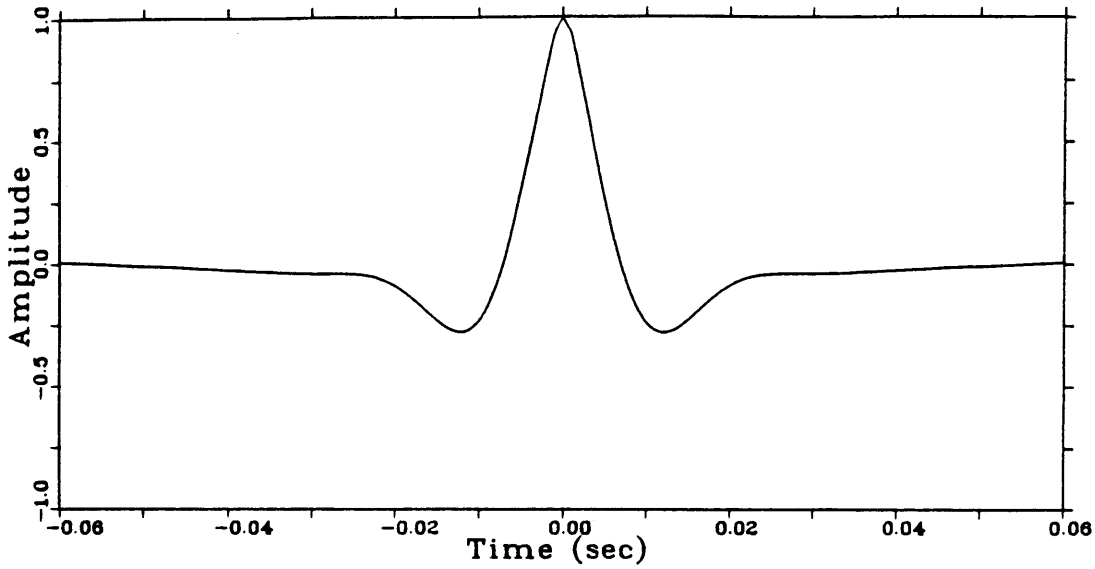

Figure 34. A matched filter derived from the estimated wavelet without ghost.

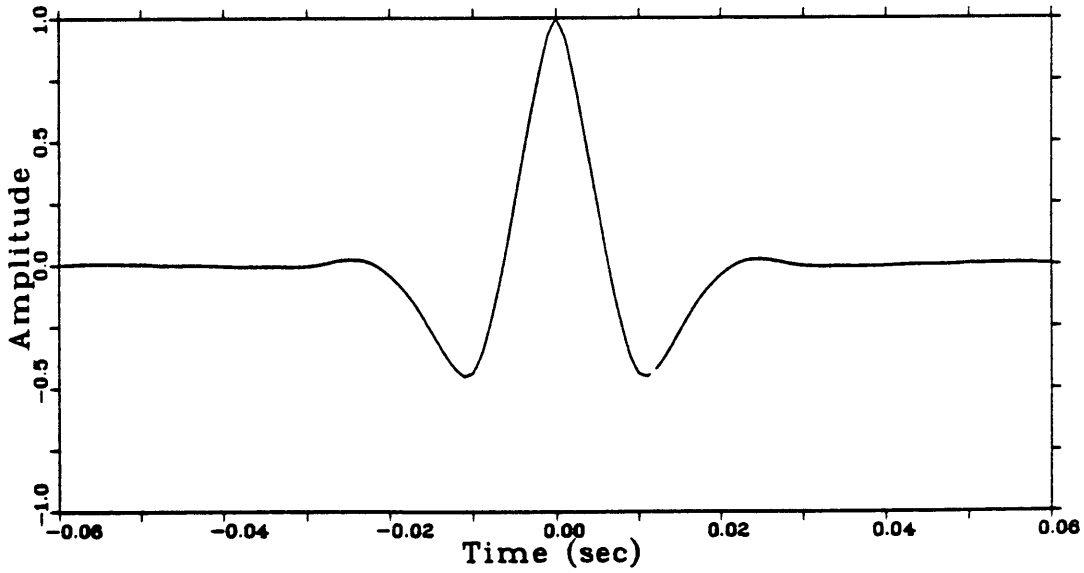

Figure 35. A matched filter derived from the estimated wavelet with shot ghost.

because the fluctuations of the matched filters die out much faster than that of the bandpass filters.

The matched filter in Figure 34 is applied to the data phase-compensated based on the wavelet without ghost, and the result is shown in Figure 36. Likewise, the matched filter in Figure 35 is applied to the data phase-compensated based on the wavelet with shot ghost, and the result is shown in Figure 37. These two results are almost identical because the two matched filters are very similar. The data processed by the new method have correct reflection arrival times, continuous reflections and clear background. The results are very similar to those filtered by narrow passband but much better than those filtered by wide passband. This is expected because the main pulse of the matched filters is very close to that of the narrow-band filter.

The common-shot gather. A common-shot gather (shot #2 of GECO Test 5 marine data) for 24 channels is used to test the wavelet processing methods. The gather has been shown in Figure 19 in the last Chapter. The data after exponential amplitude compensation, plotted in Figure 38, is dominated by reverberations. Deconvolution operators are computed trace-by-trace to take the variation with trace into account. Operators of 150 ms length are obtained from

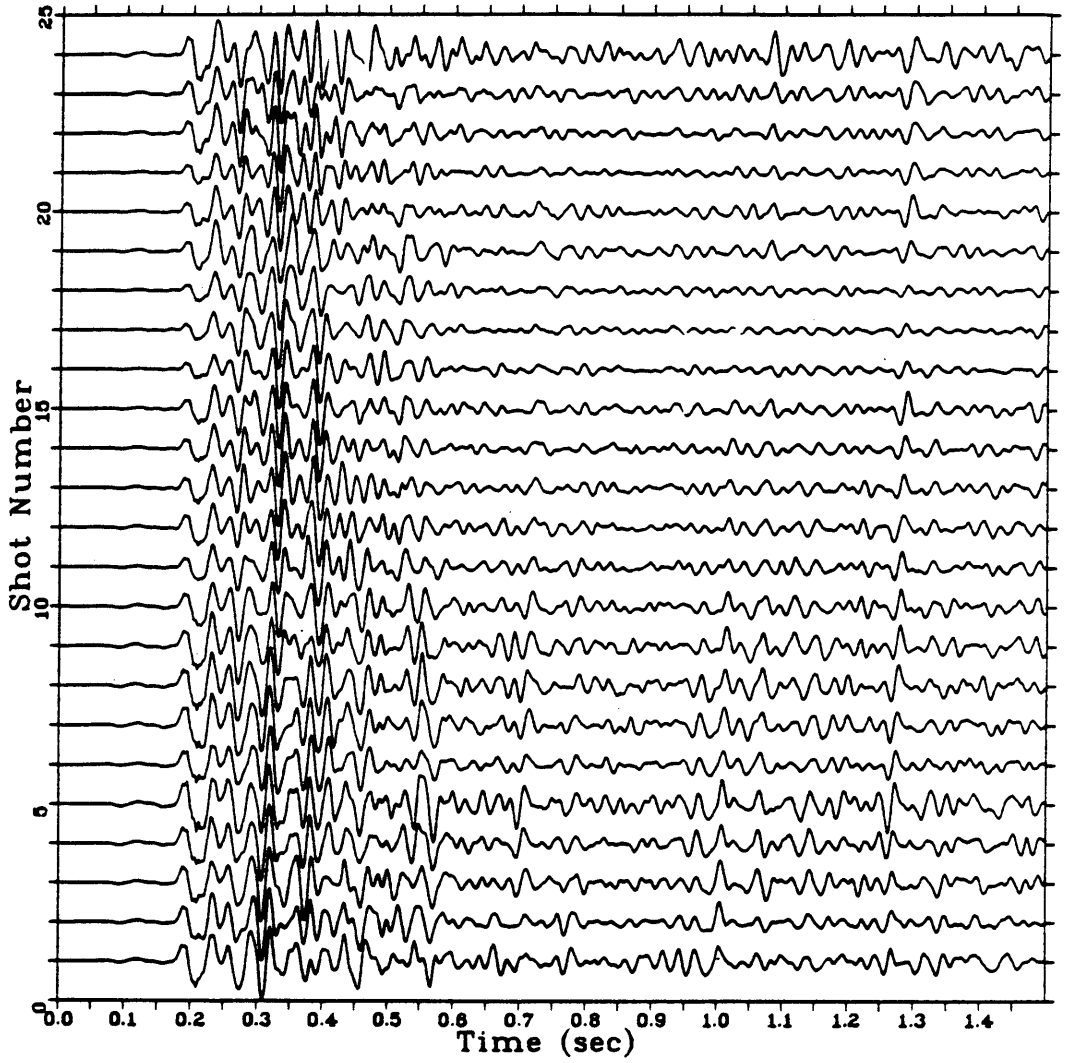

Figure 36. Wavelet processing of the near-shot trace gather using the matched filter derived from signature without ghost.

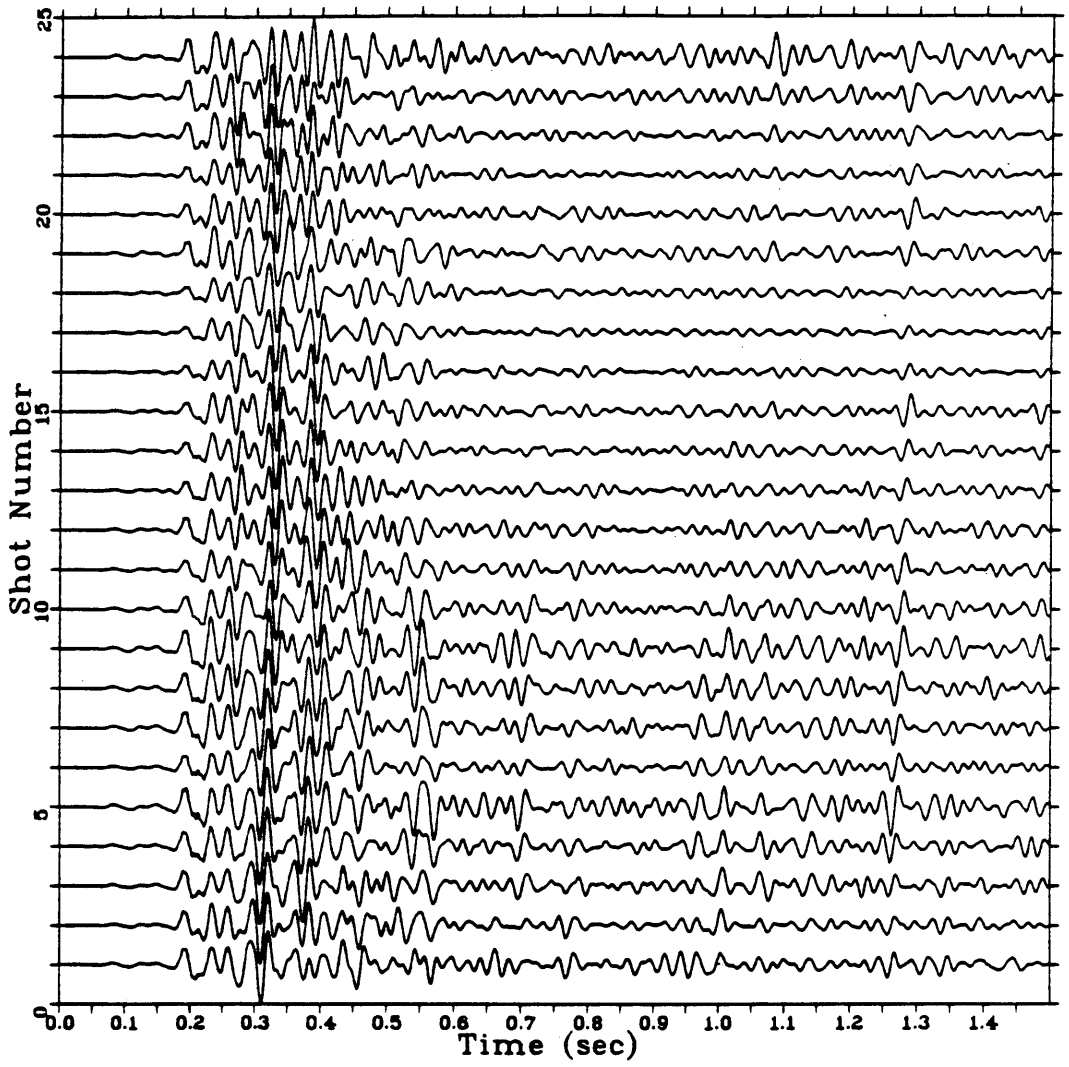

Figure 37. Wavelet processing of the near-shot trace gather using the matched filter derived from signature with shot ghost.

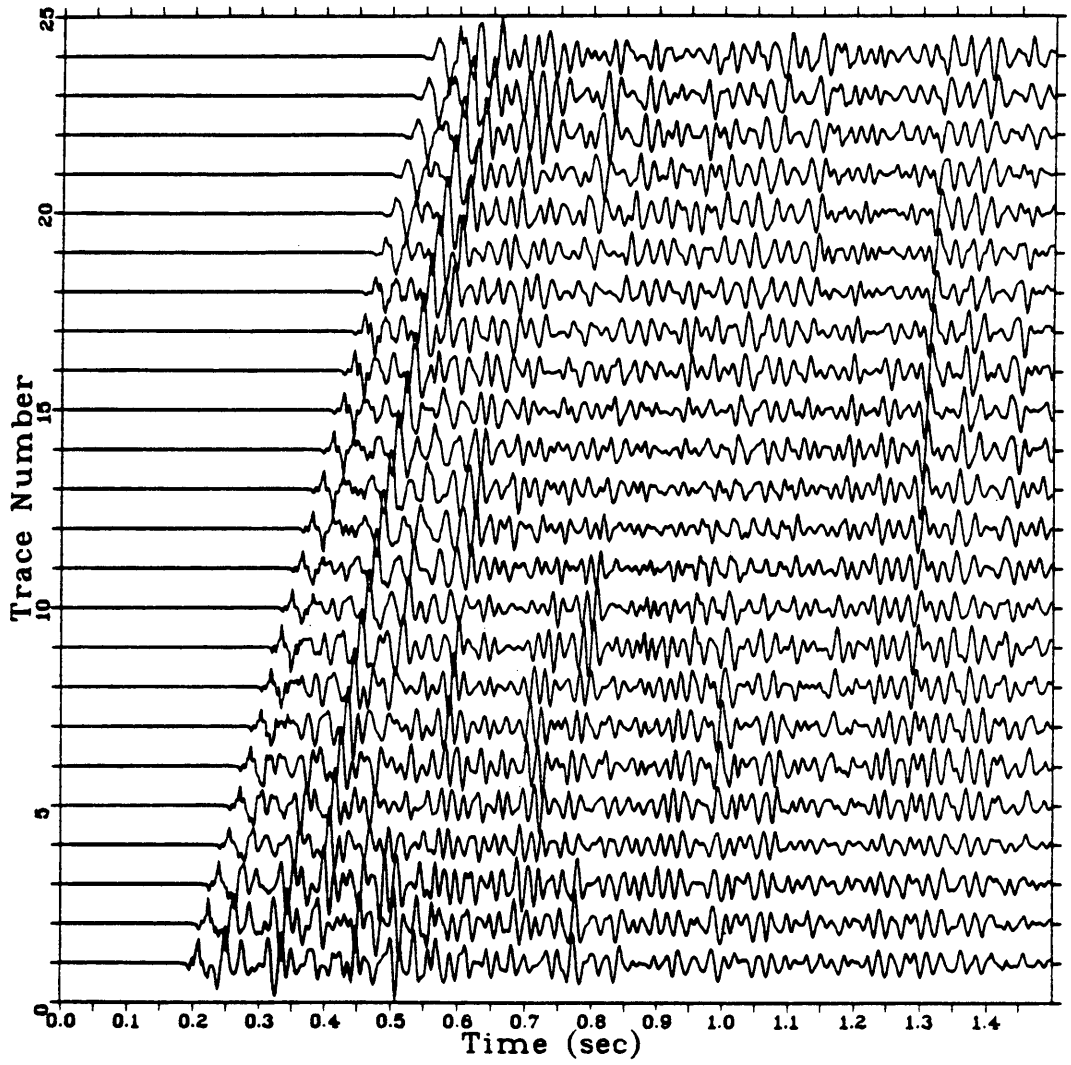

Figure 38. Amplitude compensated shot gather of GECO test 5 marine data, shot #2.

windows of one second length. To avoid the shallow effect, the window is chosen begining from 0.5 second for the near-shot trace with a increment of 12 ms/trace, i.e., the window for second trace begins from 512 ms, etc., and the operator is then applied to the corresponding trace from the begining.

The deconvolved data are shown in Figure 39. Then, both narrow and wide-band filters are applied to the deconvolved data, and the results shown in Figure 40 for narrow band and Figure 41 for wide band, respectively. The estimated signature (without ghost) after passing through the instrument is used to calculate the phase compensator and matched filter. Phase compensation is applied to the deconvolved data, followed by either narrow band or wide band filtering or matched filtering of the data. The results are shown in Figures 42 to 44, respectively.

Comparing the results mentioned above, we find that all processing methods improve the quality of the data. The reverberations are greatly suppressed by spiking deconvolution. The deconvolved data without filtering have high resolution but contain amplified noise, especially high frequency noise around the first break. The simplest processing method, which uses a bandpass filter directly after spiking deconvolution, makes the data interpretable.

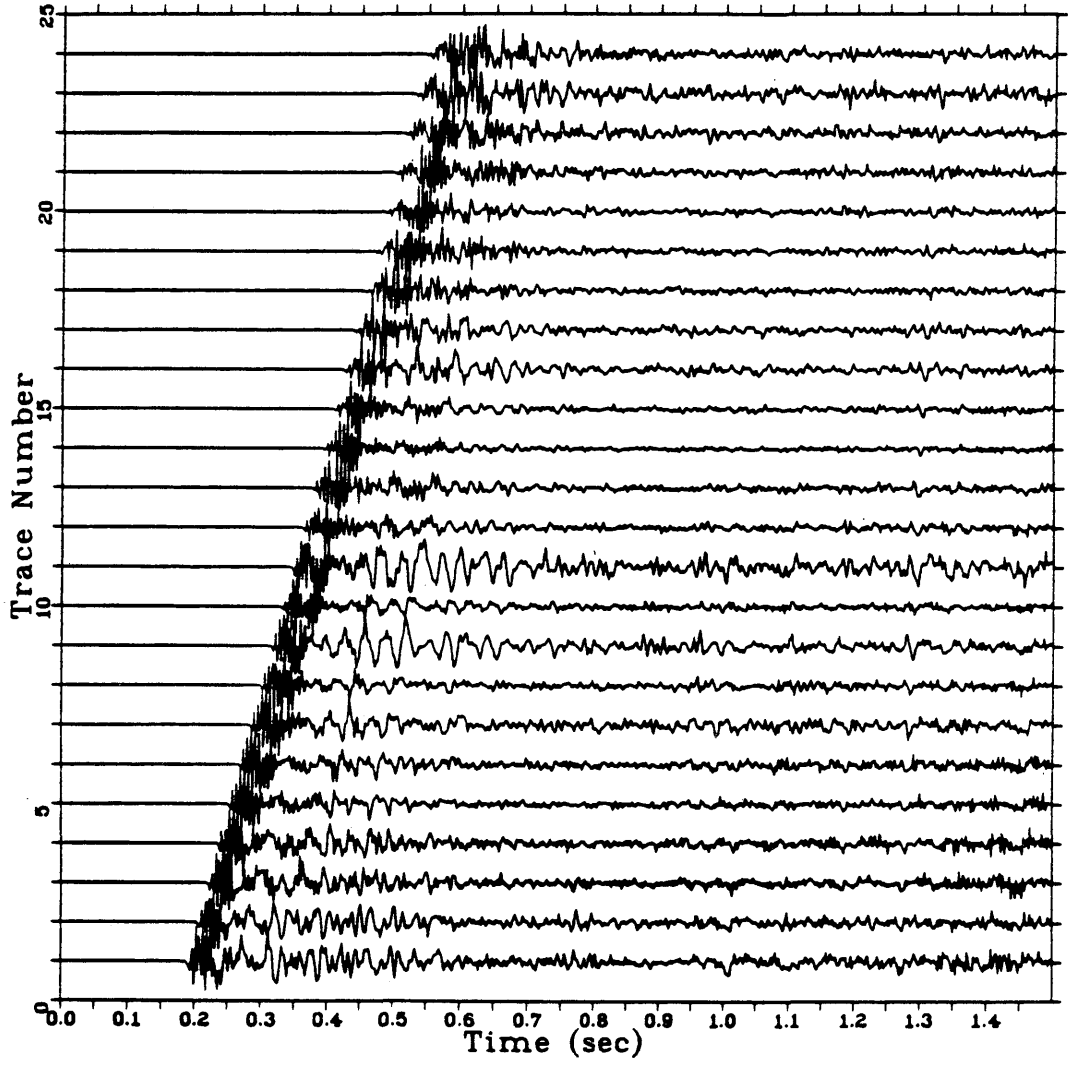

Figure 39. Spiking deconvolution of the shot gather of GECO test 5, shot #2.

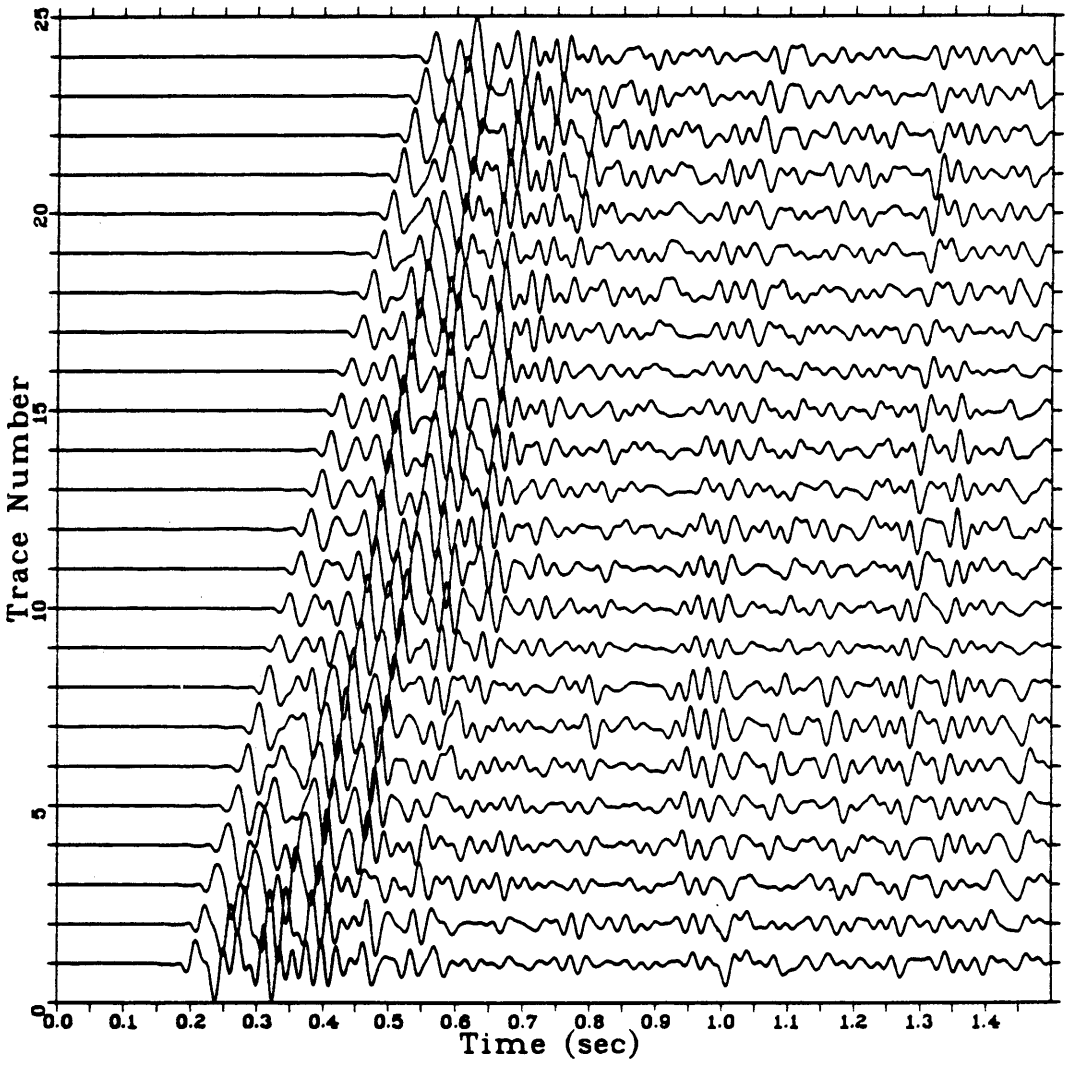

Figure 40. Deconvolved shot gather after narrow-band filtering.

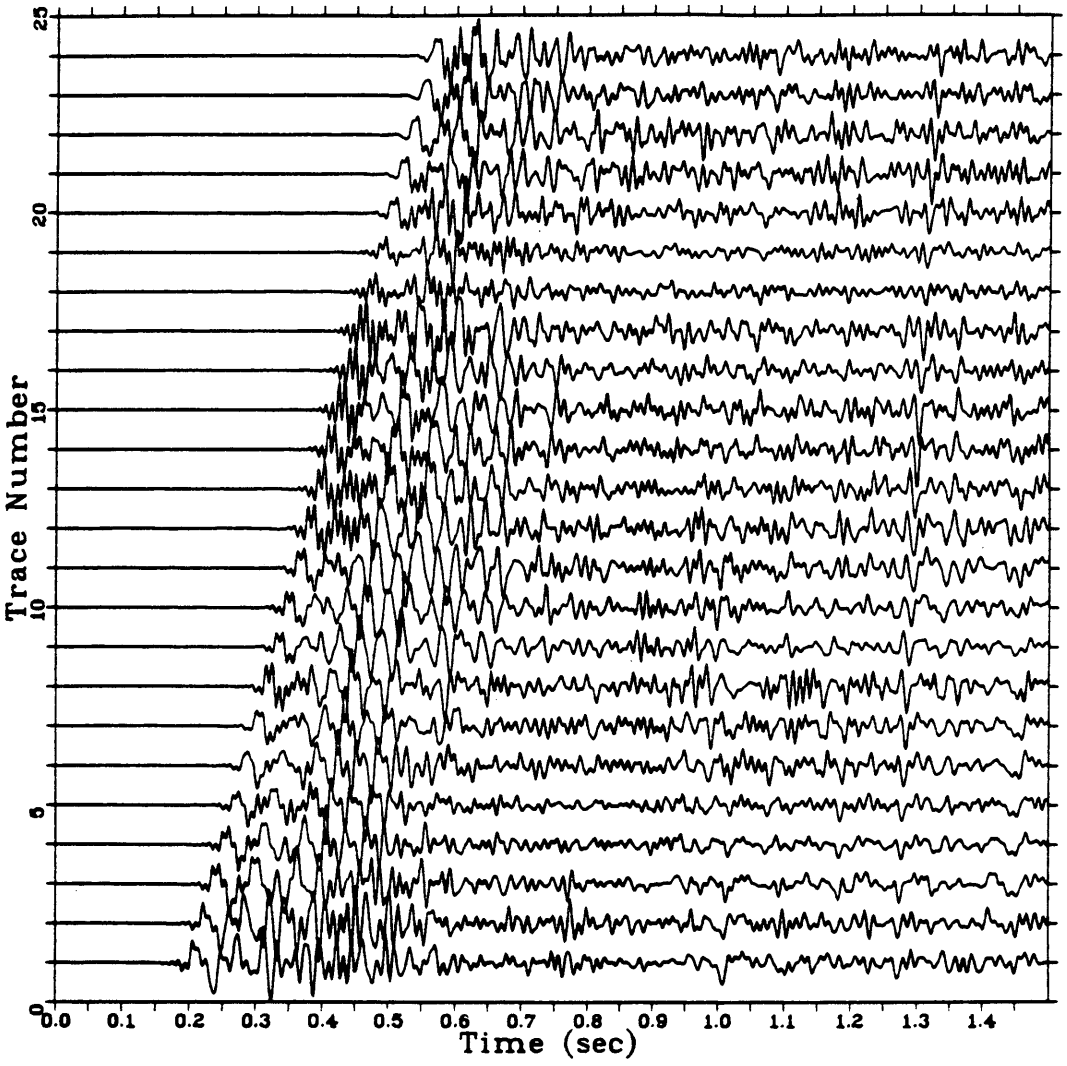

Figure 41. Deconvolved shot gather after wide-band filtering.

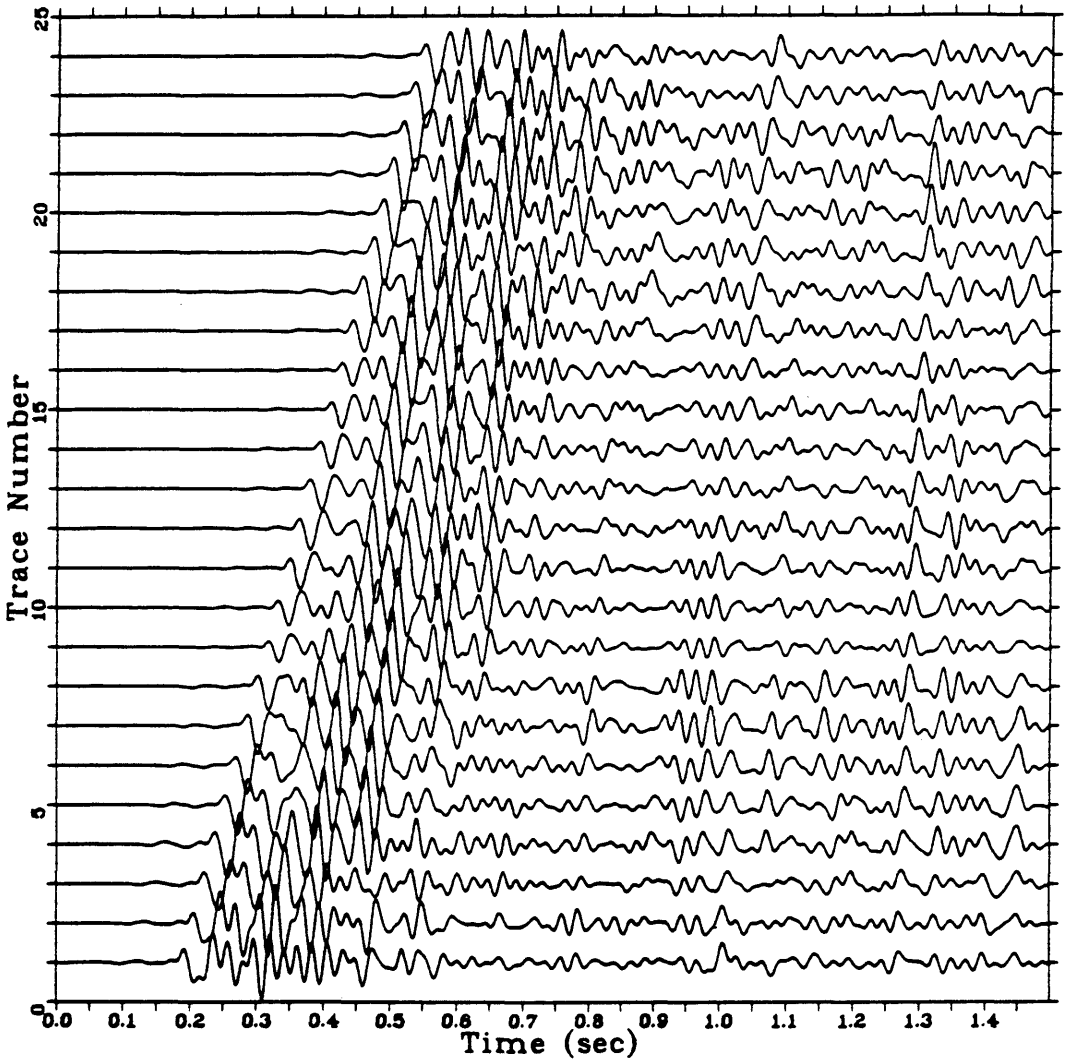

Figure 42. Deconvolved, phase-compensated, and narrow-band filtered shot gather.

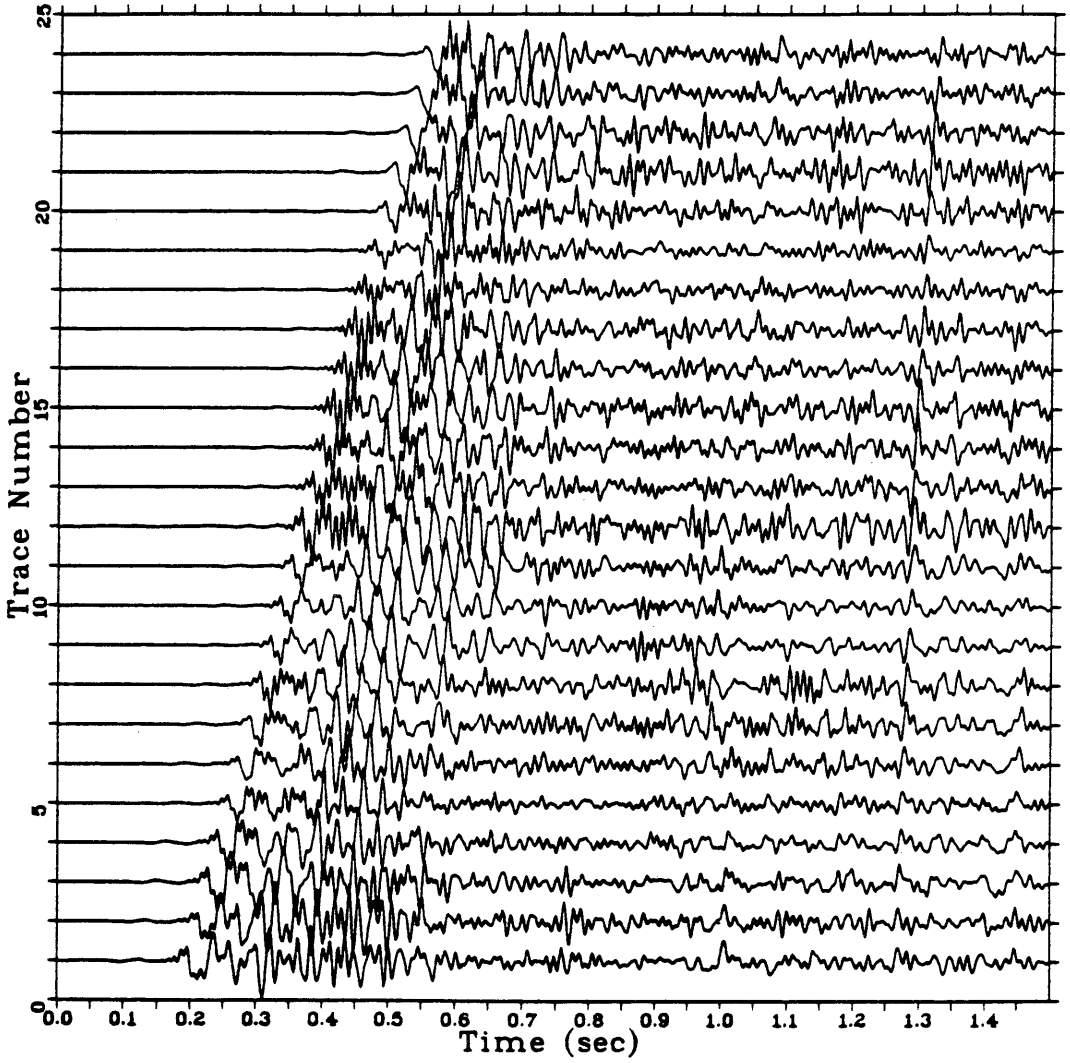

Figure 43. Deconvolved, phase-compensated, and wide-band filtered shot gather.

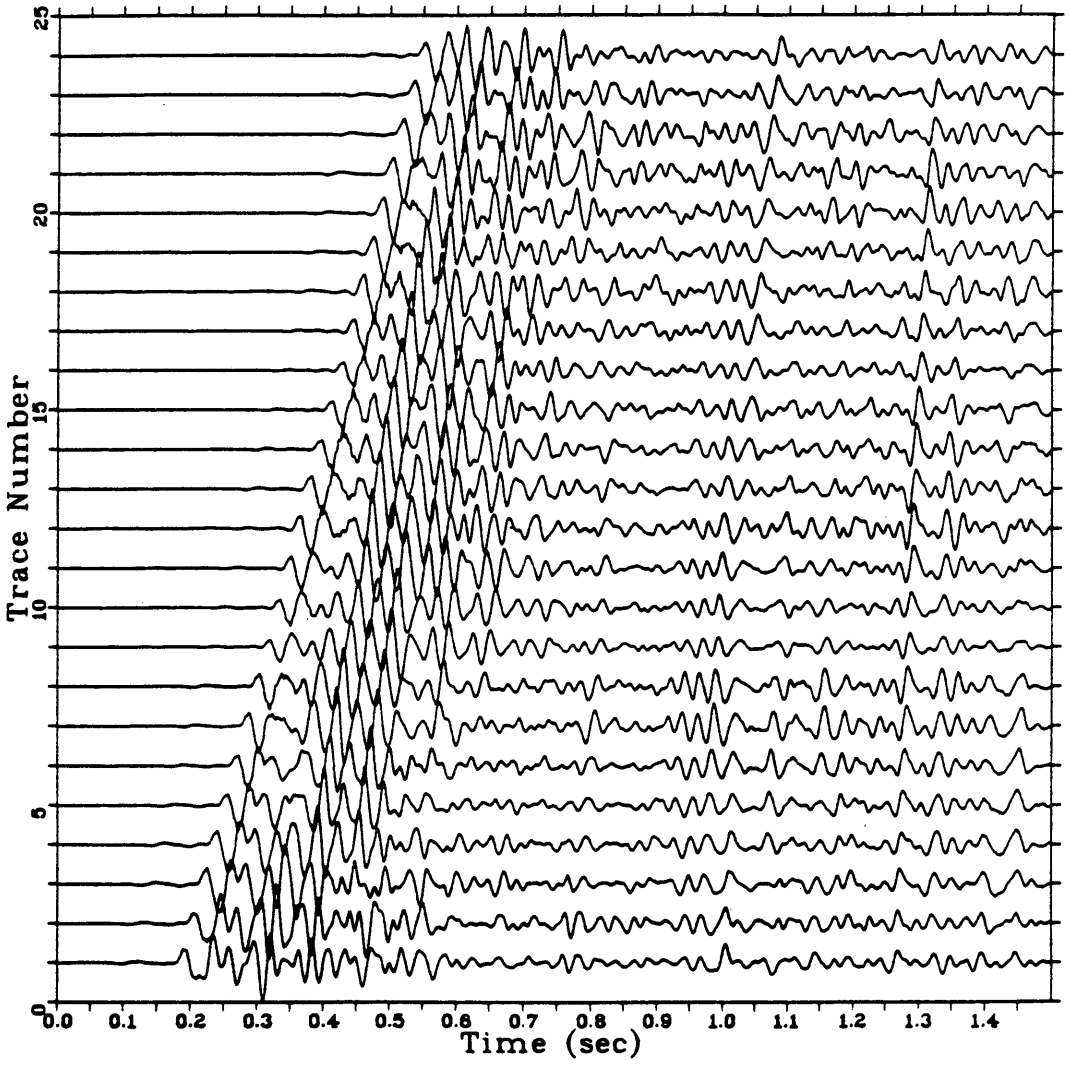

Figure 44. Wavelet processing of shot gather using matched filter.

The more conventional method, which employs phase compensation before bandpass filter, makes the deep reflections more continuous and all reflections have correct arrival times (about 20 ms earlier after phase-compensation). However, the data processed by conventional methods depend on the passband of the filter used. In this example, the data filtered by the narrow-band filter have clear reflections under the clear background, and the data filtered by the wide-band filter have high resolution but noisy background because the noise amplified by spiking deconvolution is not suppressed very well. The new processing method, which uses the matched filter as the d-filter, performs very well. The data processed by the new method have correct reflection arrival times, continuous reflections and clear background (high signal-to-noise ratio). The new method avoids the difficulty of choosing the d-filter passband correctly. The above results also prove that the estimated signature is correct and may be used for wavelet processing.

From the discussion in this section, we can see the importance of wavelet processing, the accuracy of the estimated signature, and the effectiveness of the new wavelet processing method. However, because the real geological model in the survey area is unknown, we can not

give a quantitative evalution to the processing results. To do this, a model study is employed, and this will be discussed in next section.

Wavelet Processing of Synthetic Data

To compare different wavelet processing methods quantitatively, synthetic seismic data without and with additive noise are generated and then processed by different methods, and the results are compared based on the CC values between the processing results and the desired outputs.

The solid curve in Figure 45 is an airgun array signature resampled with 4 ms interval from a measured signature recorded by DFS-V instruments with sampling interval 2 ms and filter passband, out--128/72 dB/octave rejection rate. The dashed curve in Figure 45 is the corresponding simulated signature. The delay time is removed when plotting. Their amplitude spectra are plotted in Figure 46 with solid for measured and dashed for simulated. The measured signature is used as the source signature to generate synthetic seismic data.

When processing the data, both measured and simulated signatures are used to compute phase-compensators and matched filters. The phase-compensators are shown in Figure 47, where the solid curve is based on measured signature

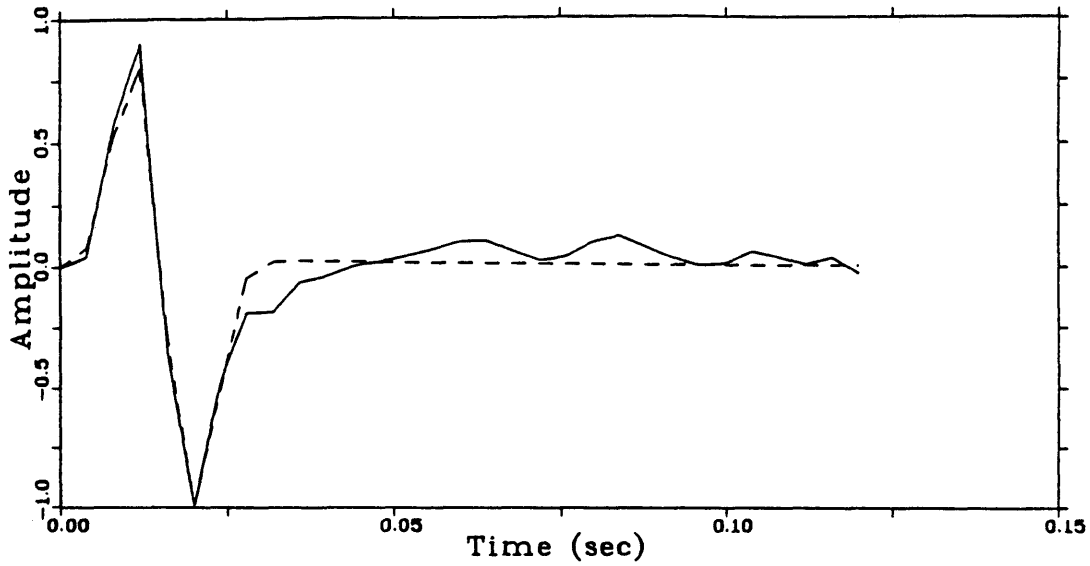

Figure 45. Airgun array signature resampled with 4 ms interval (solid) and the corresponding simulated signature (dashed).

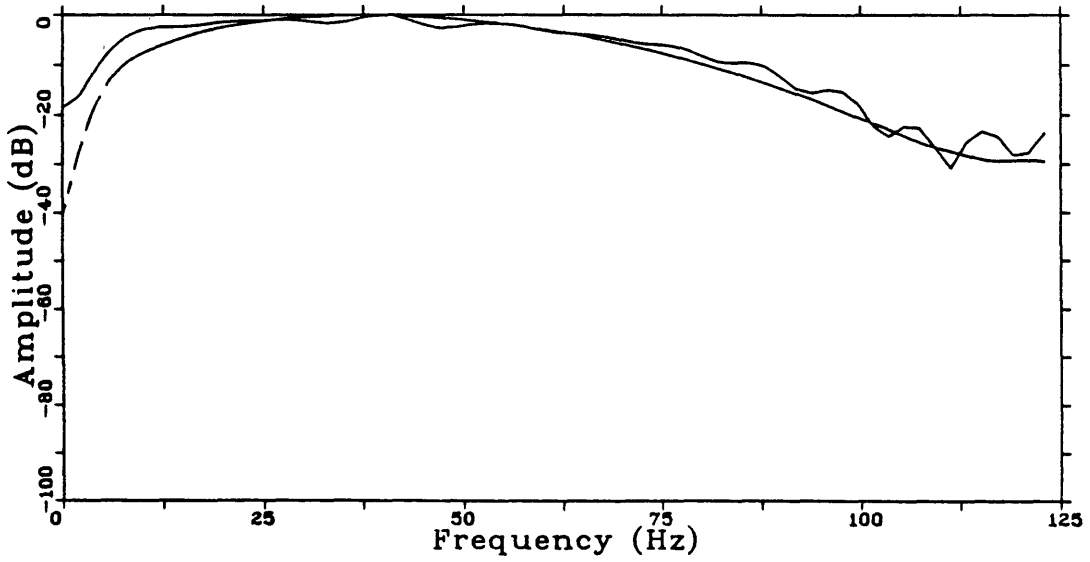

Figure 46. The amplitude spectra of signatures in Figure 45, solid for measured and dashed for calculated.

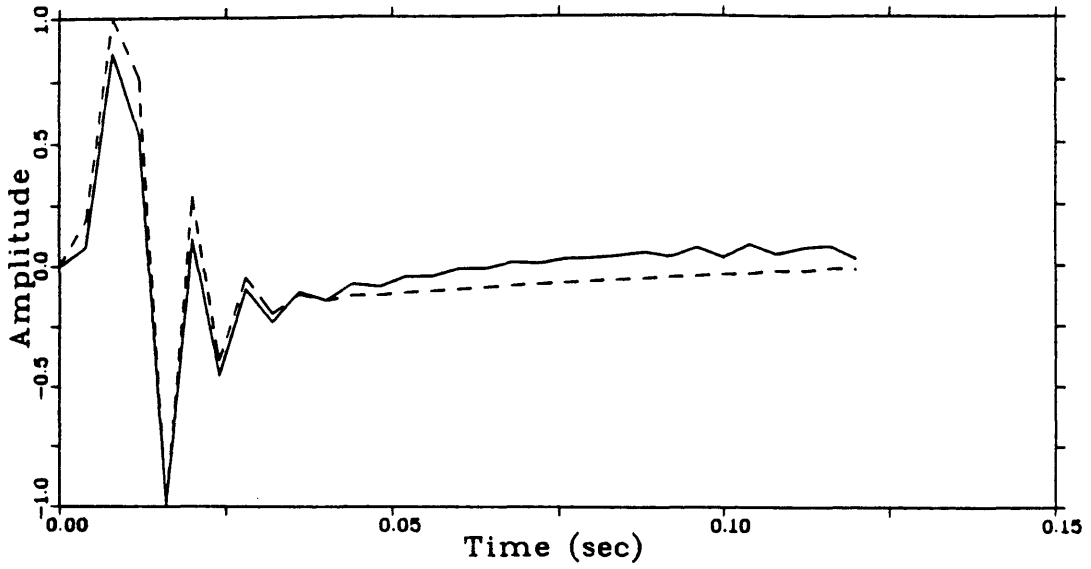

Figure 47. Phase-compensators derived from measured (solid) and simulated (dashed) signatures in Figure 45.

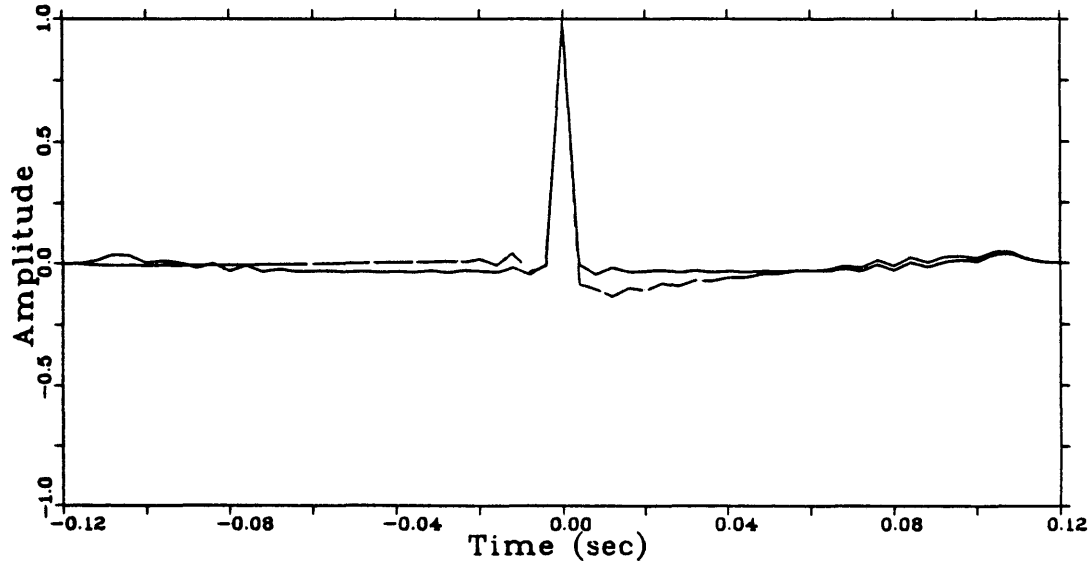

Figure 48. Autocorrelation (solid) of measured compensator and cross-correlation (dashed) of measured and calculated compensators in Figure 47.

and the dashed curve is based on calculated signature. Autocorrelation of the phase-compensator based on measured signature and cross-correlation of the measured and calculated phase-compensators are plotted in Figure 48 by solid and dashed curves. From the above figures, we can see that the small difference between the measured and calculated signatures is not important in phase compensation.

Figure 49 shows the reflectivity function of a marine model, which has source depth 7.5 meters (same as that of the measured signature), receiver depth 9 meters, water depth 30 meters, reflection coefficient 0.5 at sea bottom, and zero offset. The reflectivity function is calculated from well-log data with sampling interval 4 ms. The zero offset synthetic data including ghosts and reverberations are generated using the reflectivity function and measured airgun signature shown above, and the synthetic data are plotted in Figure 50. Because the noise level is not zero even in the best real data, a normally distributed random process (Figure 51) is produced and added to the synthetic data as additive natural noise. The data including 5% noise are shown in Figure 52.

The synthetic data both without noise and with 5% noise are processed by the conventional and new methods. Four

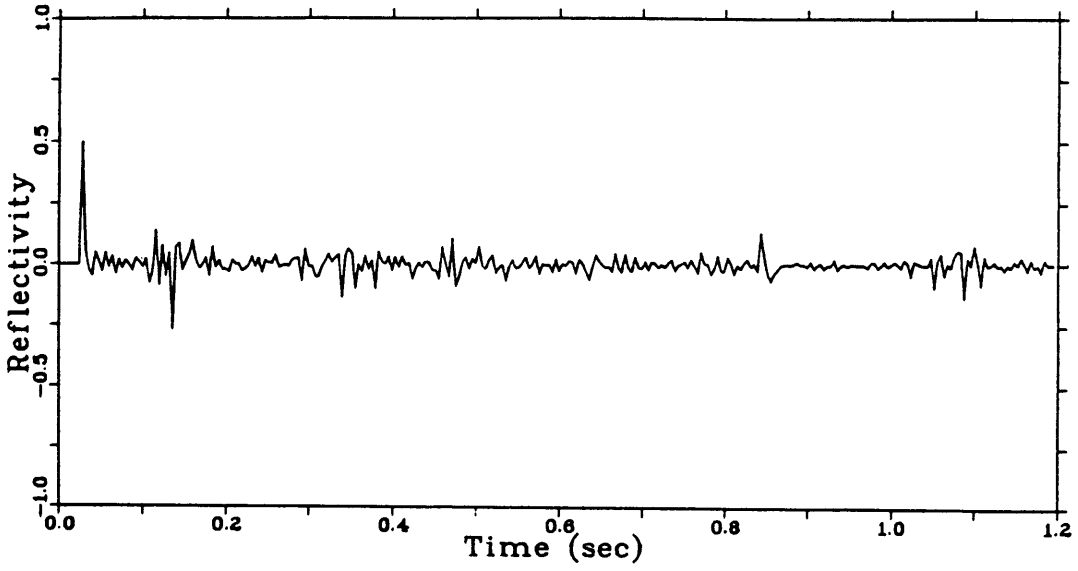

Figure 49. Reflectivity function for a marine model.

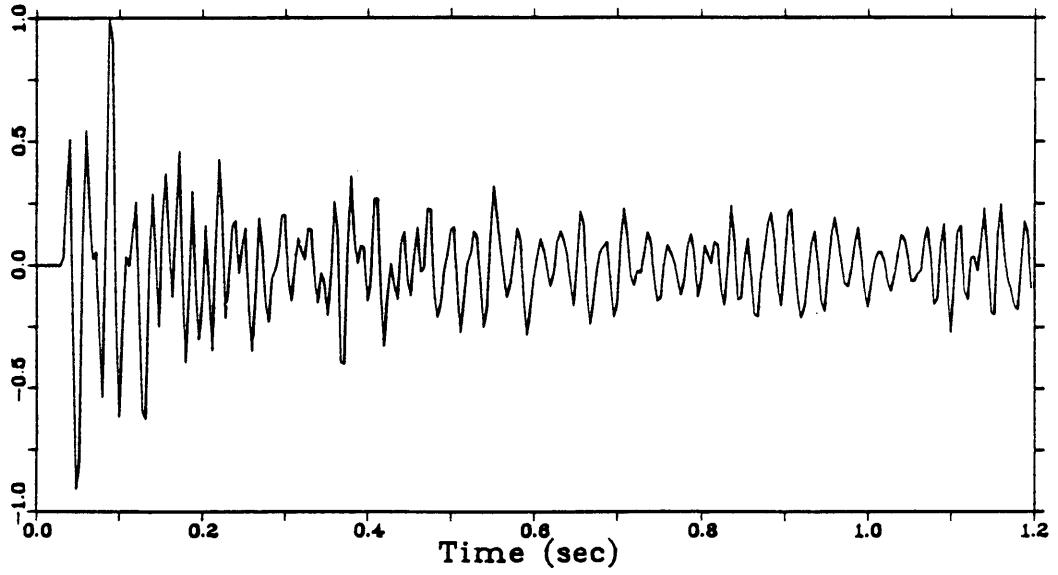

Figure 50. Synthetic marine seismic data without noise.

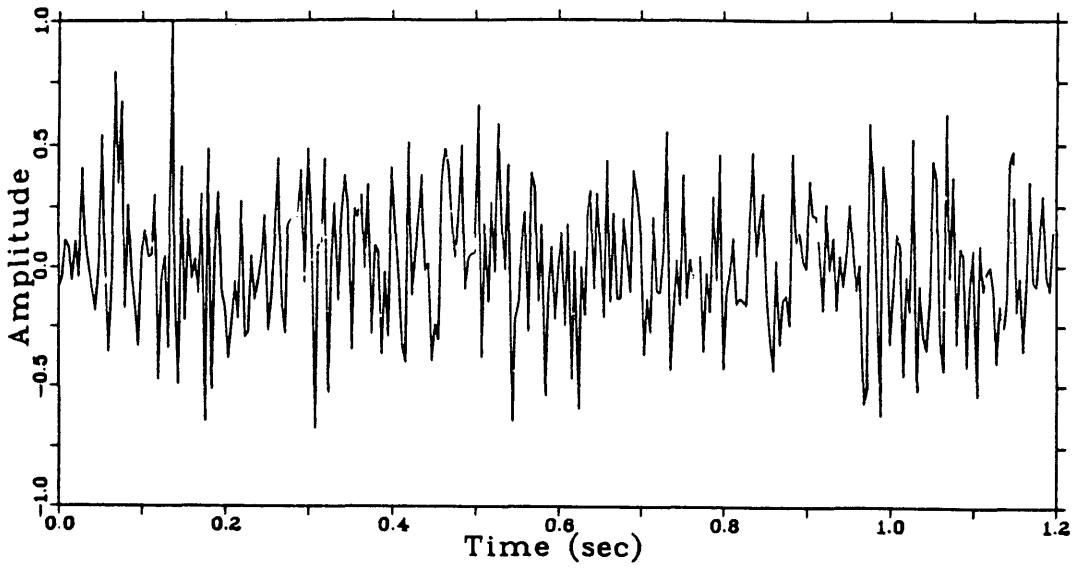

Figure 51. Normally distributed random process.

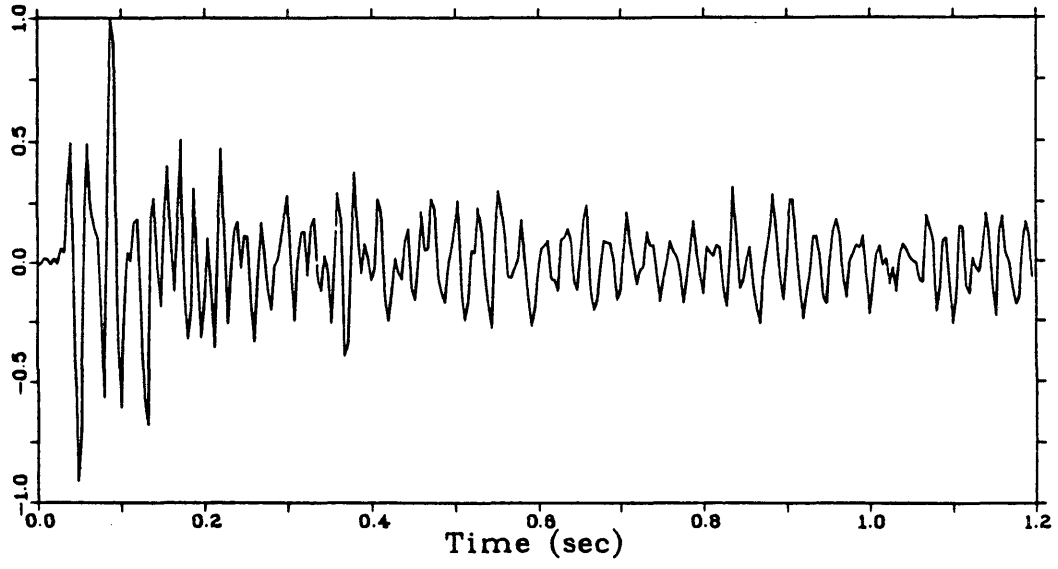

Figure 52. Synthetic marine seismic data with 5% additive random noise.

different filters, which include a narrow-band filter with (0, 10 - 60, 80) Hz passband, a wide-band filter with (0, 10 - 100, 120) Hz passband, a matched filter from the measured source signature, and a matched filter from the calculated signature, are used as d-filters. Thus, four different desired outputs are obtained by convolving the reflectivity with four different d-filters respectively. The desired outputs are shown in Figures 53 to 56. These desired outputs are used to evaluate the processing results.

The synthetic data without additive noise are processed by the conventional and new methods. The deconvolution operator and phase compensator are computed without adding any processing noise. The CC values between the processing results and desired outputs are calculated and tabulated in Table 1, where r(t) is the reflectivity function; d_1 is the narrow-band filter with (0, 10 - 60, 80) Hz; d_2 is the wide-band filter with (0, 10 - 100, 120) Hz; d_3 is the matched filter from measured signature; d_4 is the matched filter from calculated signature; b_m and b_e are the measured signature and estimated signature; SP is spiking deconvolution; PC is phase compensation. The value inside the parenthesis below the CC value indicates the relative time shift (in ms) of the resulting output to the desired

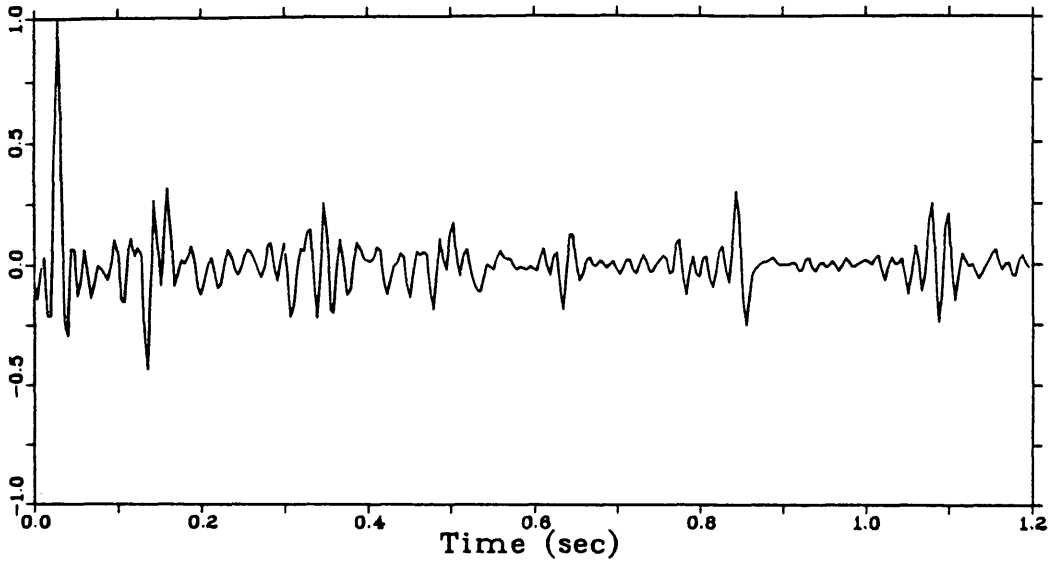

Figure 53. Desired output: reflectivity convolved with narrow-band filter.

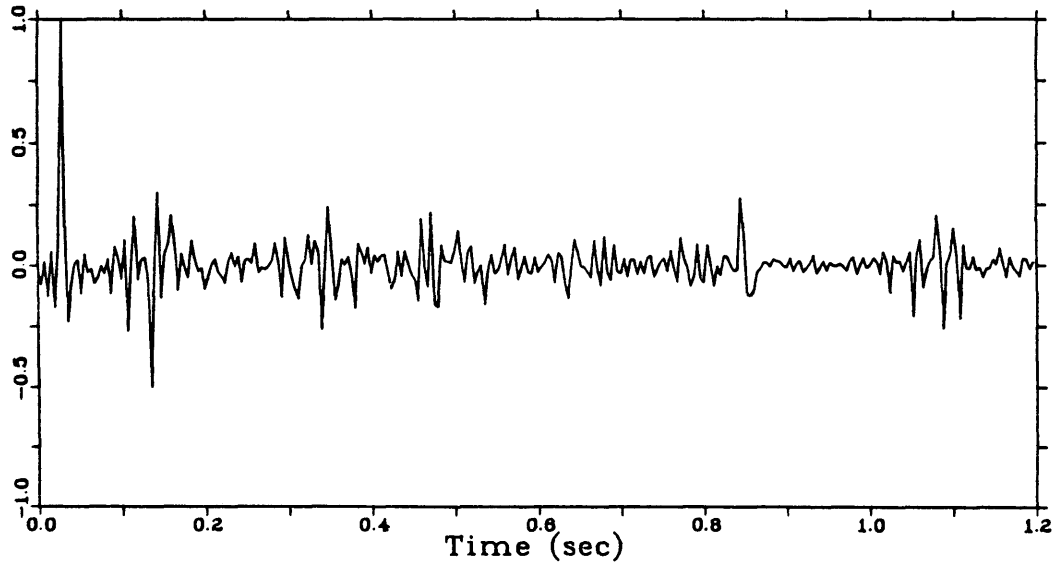

Figure 54. Desired output: reflectivity convolved with wide-band filter.

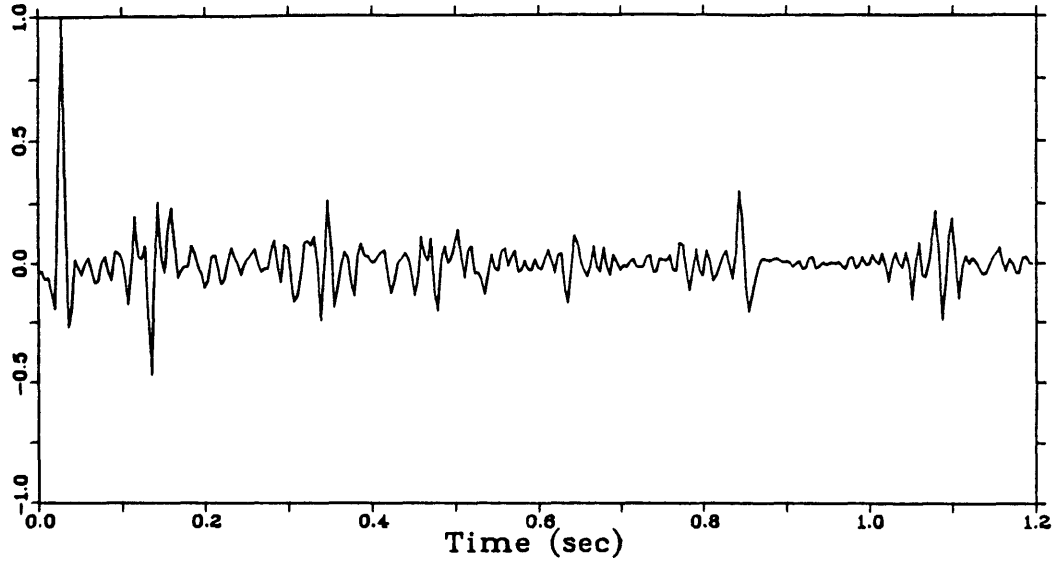

Figure 55. Desired output: reflectivity convolved with matched filter from measured source.

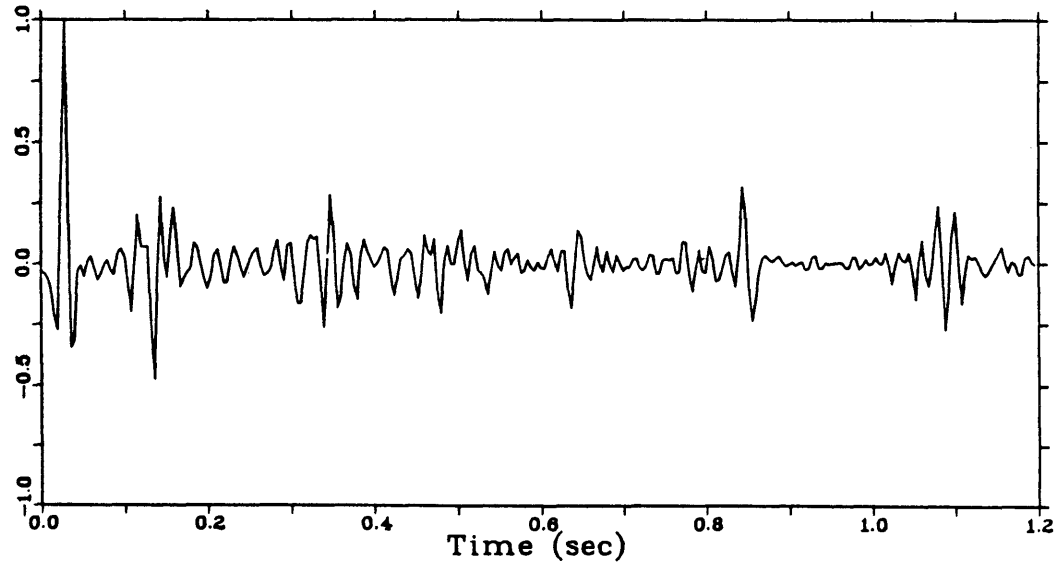

Figure 56. Desired output: reflectivity convolved with matched filter from simulated source.

Table 1. Correlation coefficients between desired outputs and wavelet processing results for synthetic data w/o additive noise.

Desired Outputs	$r(t)*d_1$	$r(t)*d_2$	$r(t)*d_3$	$r(t)*d_4$
SP-Decon $*d_1$	0.793 (-8)	0.623 (-8)	0.765 (-8)	0.792 (-8)
SP-Decon $*d_2$	0.675 (-8)	0.672 (-8)	0.715 (-8)	0.721 (-8)
PC by b_m $*d_1$	0.914	0.719	0.871	0.868
PC by b_m $*d_2$	0.779	0.897	0.850	0.823
PC by b_m $*d_3$	0.889	0.801	0.908	0.895
PC by b_e $*d_1$	0.856	0.675	0.821	0.834
PC by b_e $*d_2$	0.729	0.862	0.809	0.794
PC by b_e $*d_4$	0.875	0.771	0.890	0.906

SP - spiking deconvolution
PC - phase compensation
b_m - measured airgun signature
b_e - estimated signature
d_1 - 0,10-60,80 Hz filter
d_2 - 0,10-100,120 Hz filter
d_3 - matched filter from b_m
d_4 - matched filter from b_e
r - reflectivity function
* - convolution

output, with minus sign indicating delay. Only the non-zero values of time shift appear in the table.

From Table 1, we find that all results processed by the simplest method (i.e. without phase compensation) have 8 ms time delay, regardless of what filter is used. This indicates phase distortion in the deconvolved data and the necessity of phase correction. All of the phase-compensated results have correct arrival time, and the CC values with phase compensation are higher than the corresponding values without phase compensation. This indicates the effective performance of the phase compensators from both measured and calculated signatures.

For phase compensation based on the measured signature, the CC value using the matched filter is 8.2% higher than that using the narrow-band filter when the desired output is the reflectivity filtered by the wide-band filter; the CC value using the matched filter is 11.0% higher than that using the wide-band filter when the desired output is the convolution of reflectivity with the narrow-band filter.

For phase compensation based on the calculated signature, the CC value using the matched filter is 9.6% higher than that using the narrow-band filter when the desired output is the reflectivity convolved with the wide-band filter; the CC value using the matched filter is

14.6% higher than that using the wide-band filter when the desired output is the convolution of the reflectivity with the narrow-band filter. From these results, we can see that the narrow passband causes loss of high frequency components while the wide passband does not suppress noise very well. Only the matched filter can both preserve high frequency components and suppress noise.

Table 1 also indicates that the calculated signature is accurate enough for wavelet processing. The processing results are shown in Figures 57 to 64 in the same order as Table 1.

The synthetic data with 5% additive noise (Figure 52) are processed by the same methods as those for the synthetic data without noise, and the processing results are shown in Figures 65 to 72. The CC values between these results and the desired outputs are tabulated in Table 2. All symbols in Table 2 have the same meanings as those in Table 1 except that P=-1 in Table 2 means inverse polarity. By comparing Figures 65 to 72 with Figures 57 to 64 and Table 2 with Table 1, it is apparent that additive noise in the data greatly degrades the performance of wavelet processing. The deconvolved data without phase-compensation have inverse polarity and 20 ms time delay (except one has positive polarity and 8 ms delay) relative to the desired

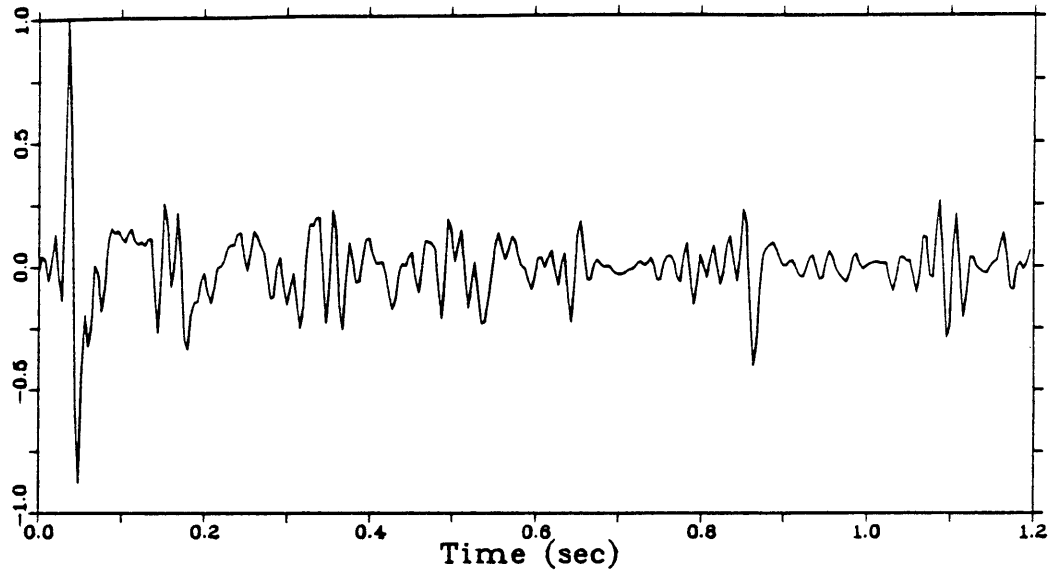

Figure 57. Deconvolved and narrow bandlimited synthetic data (without noise).

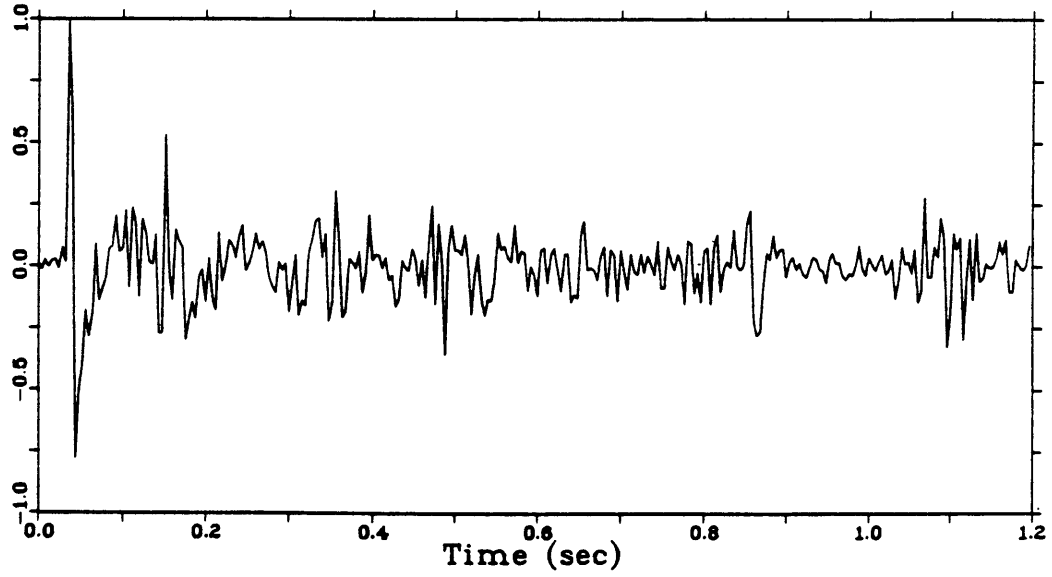

Figure 58. Deconvolved and wide bandlimited synthetic data (without noise).

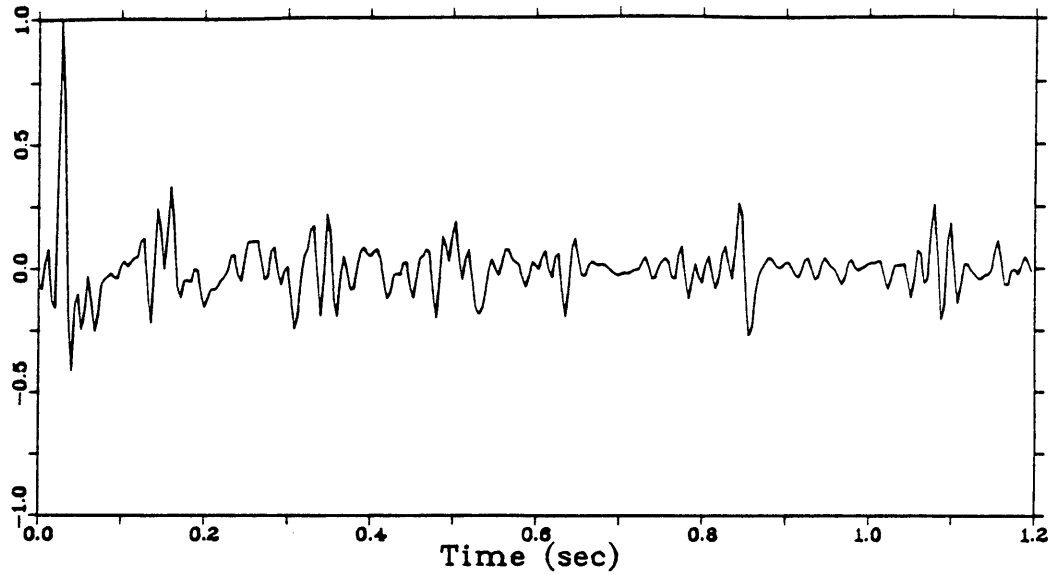

Figure 59. Deconvolved, phase-compensated, and narrow-band filtered data (without noise). Phase-compensation based on measured signature.

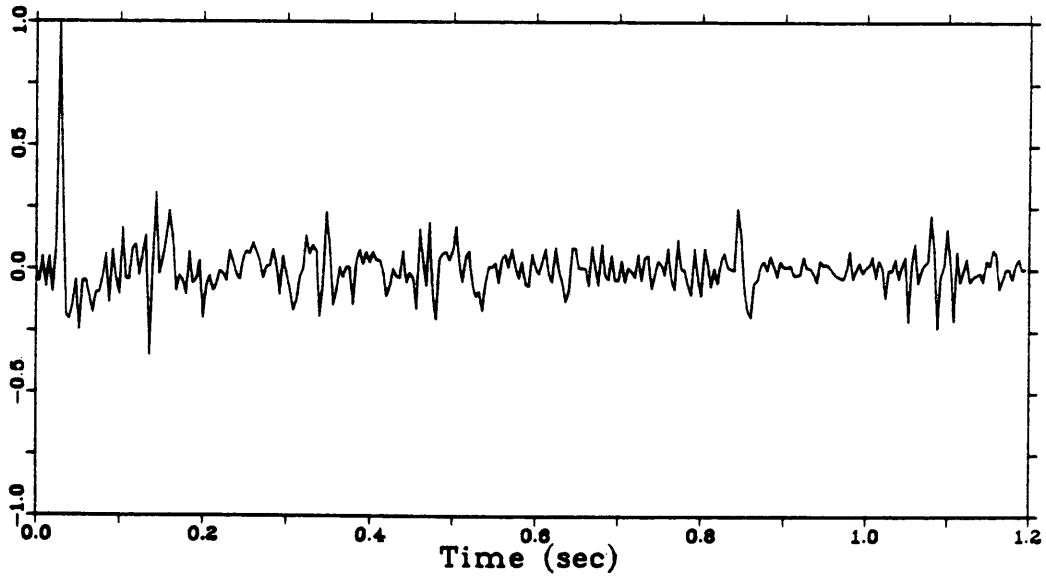

Figure 60. Deconvolved, phase-compensated, and wide-band filtered data (without noise). Phase-compensation based on measured signature.

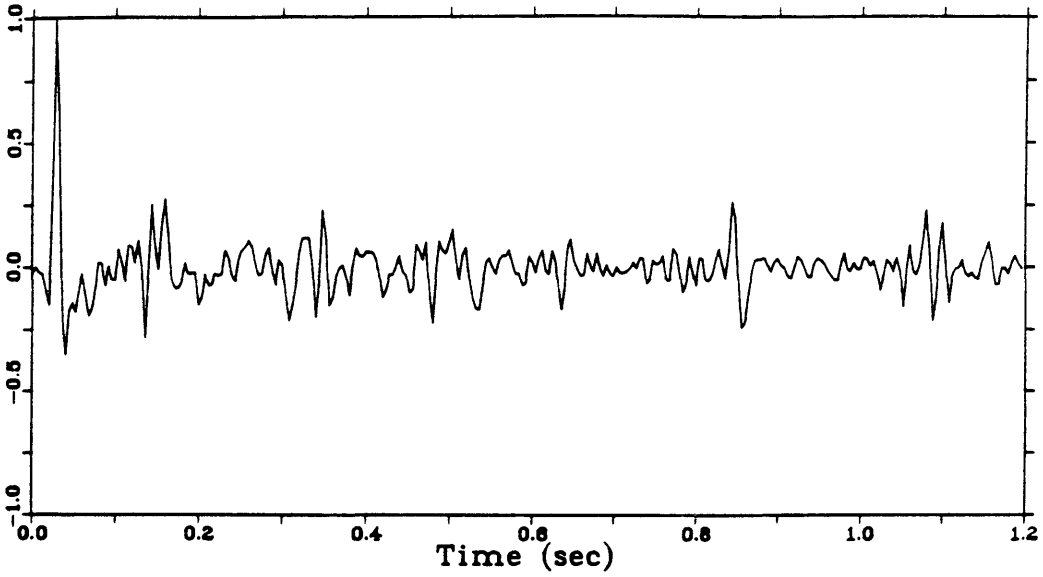

Figure 61. Wavelet processing of synthetic data (without noise) using matched filter from measured signature.

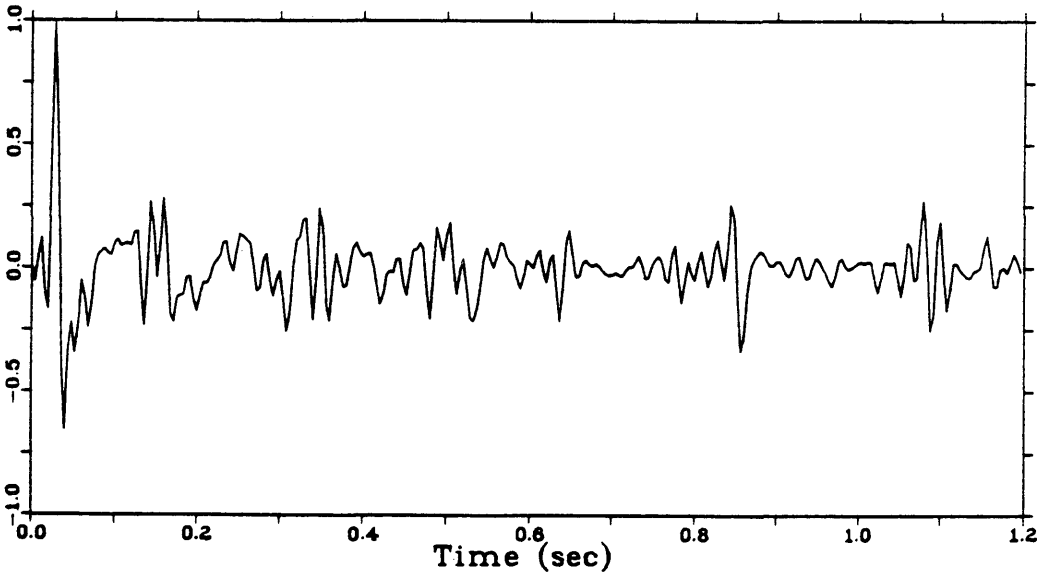

Figure 62. Deconvolved, phase-compensated, and narrow-band filtered data (without noise). Phase compensation based on simulated signature.

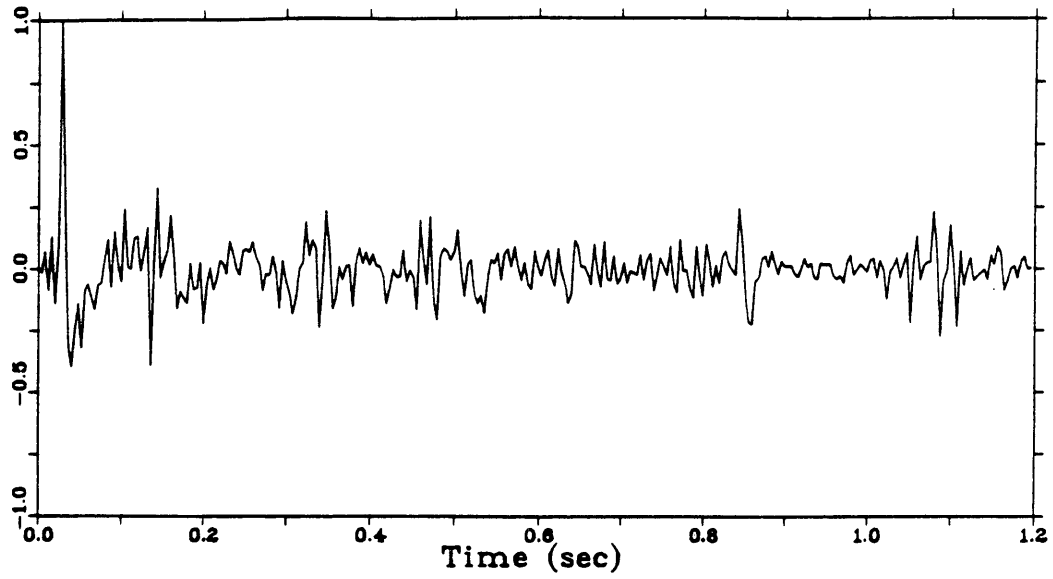

Figure 63. Deconvolved, phase-compensated, and wide-band filtered data (without noise). Phase compensation based on simulated signature.

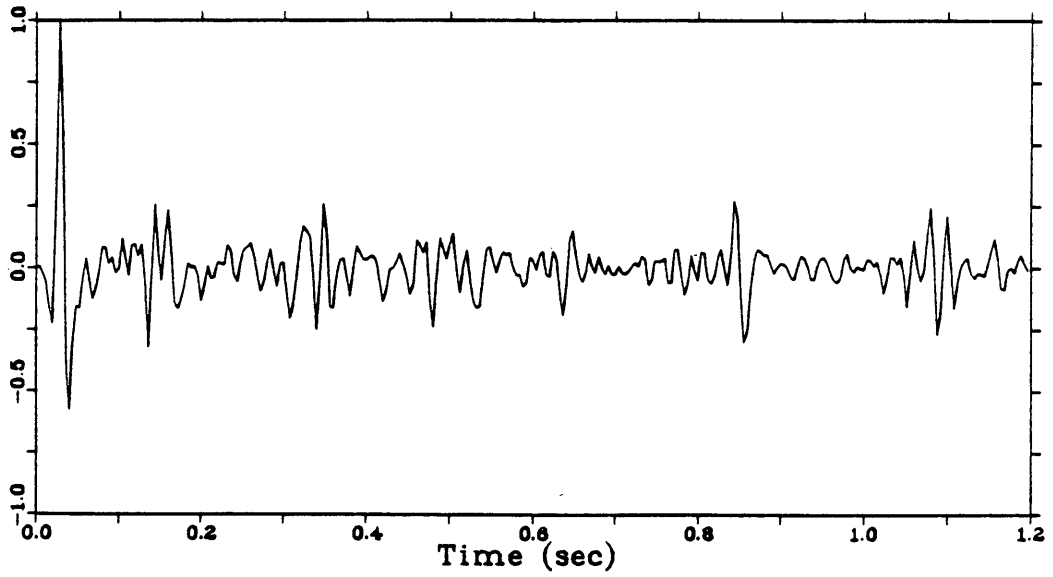

Figure 64. Wavelet processing of synthetic data (without noise) using matched filter from simulated signature.

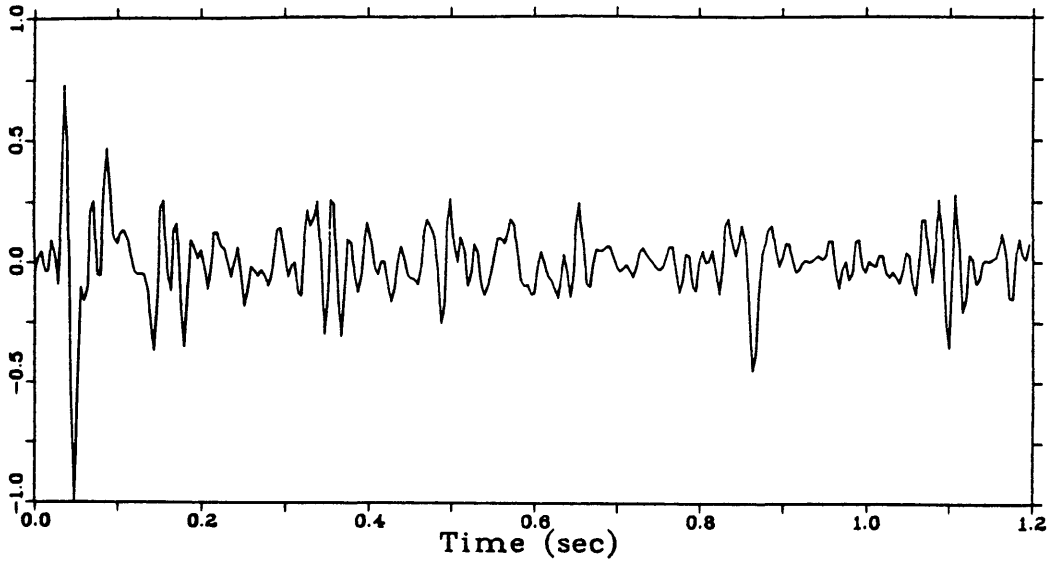

Figure 65. Deconvolved and narrow-band filtered synthetic data (with 5% additive noise).

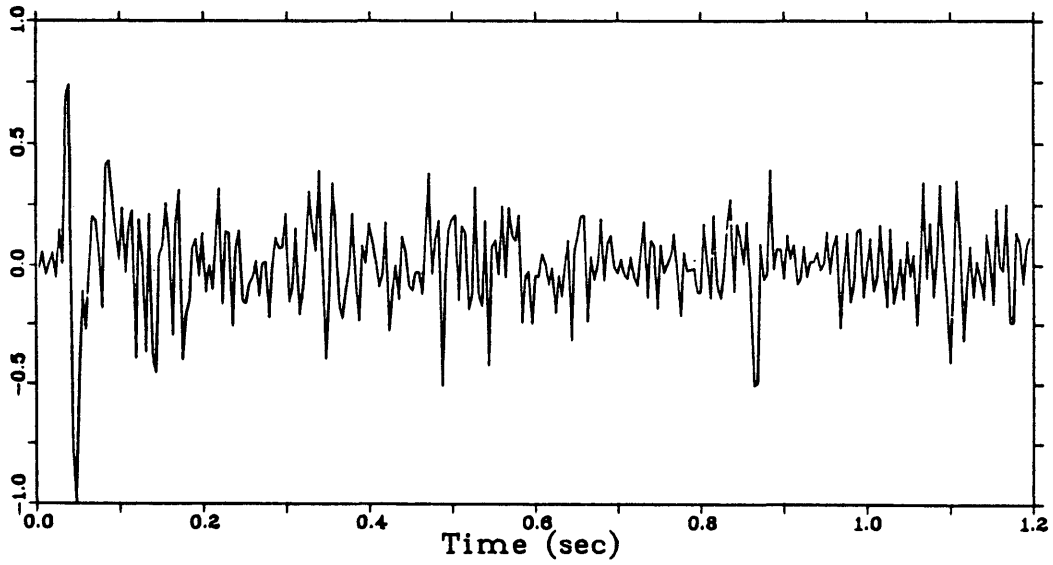

Figure 66. Deconvolved and wide-band filtered synthetic data (with 5% additive noise).

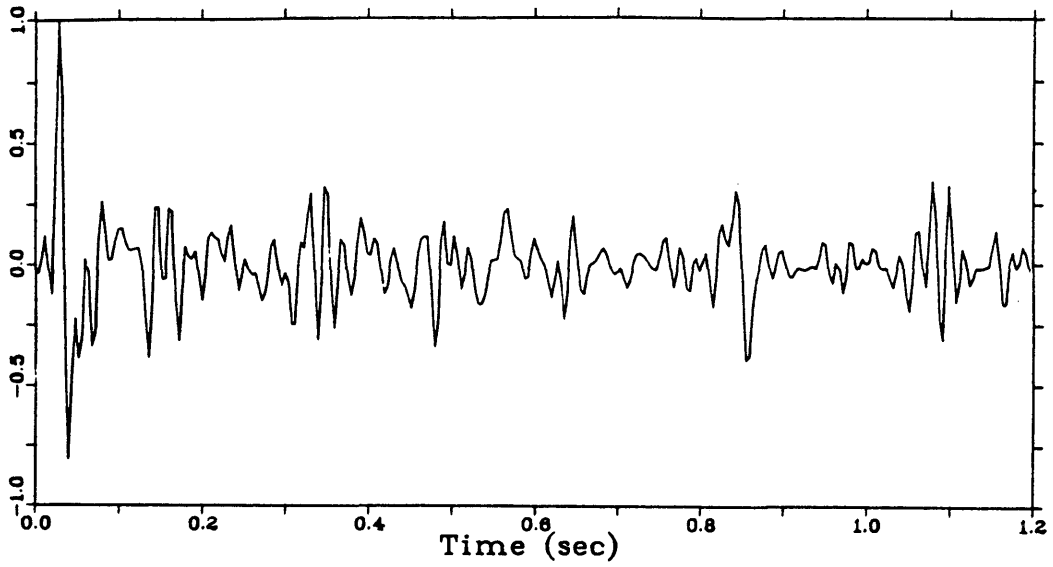

Figure 67. Deconvolved, phase-compensated, and narrow-band filtered data (with 5% noise). Phase-compensation based on measured signature.

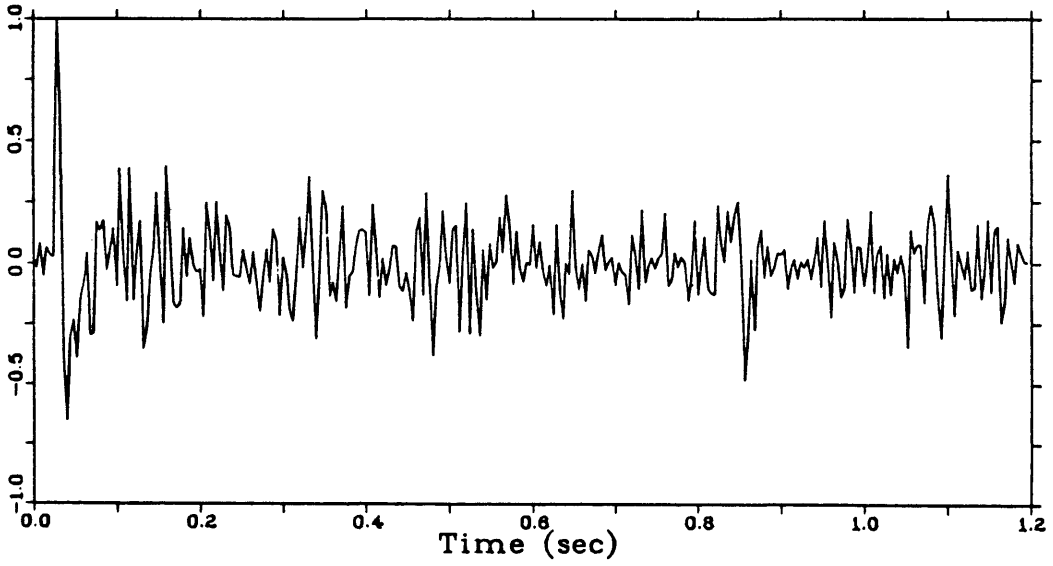

Figure 68. Deconvolved, phase-compensated, and wide-band filtered data (with 5% noise). Phase-compensation based on measured signature.

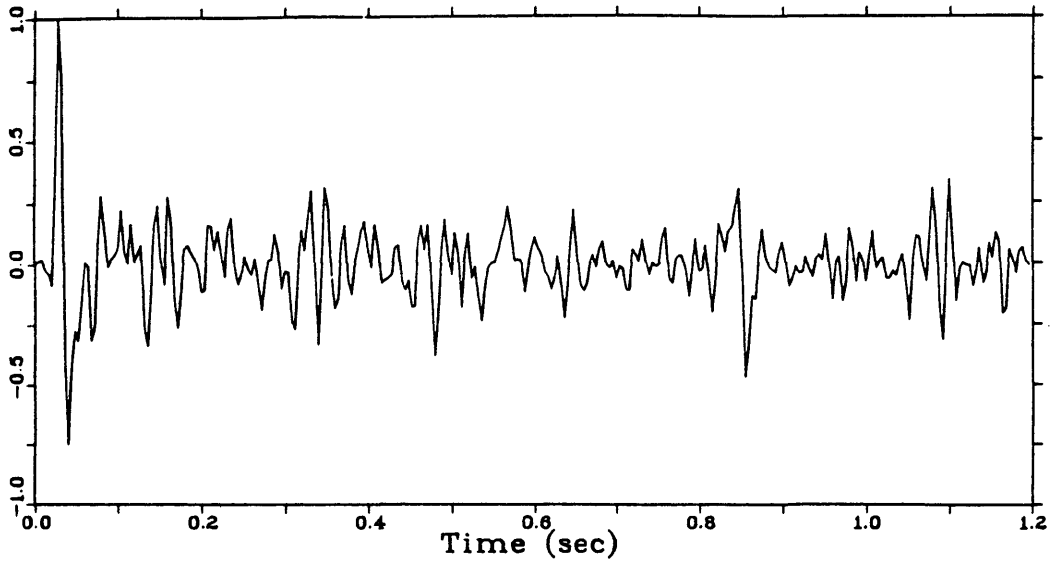

Figure 69. Wavelet processing of synthetic data (with 5% noise) using matched filter from measured signature.

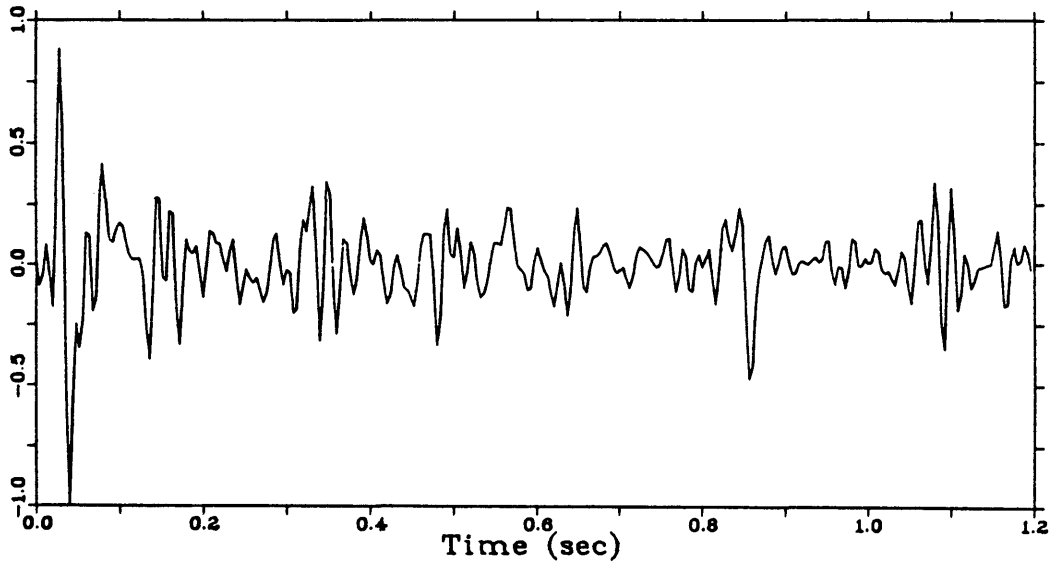

Figure 70. Deconvolved, phase-compensated, and narrow-band filtered data (with 5% noise). Phase compensation based on simulated signature.

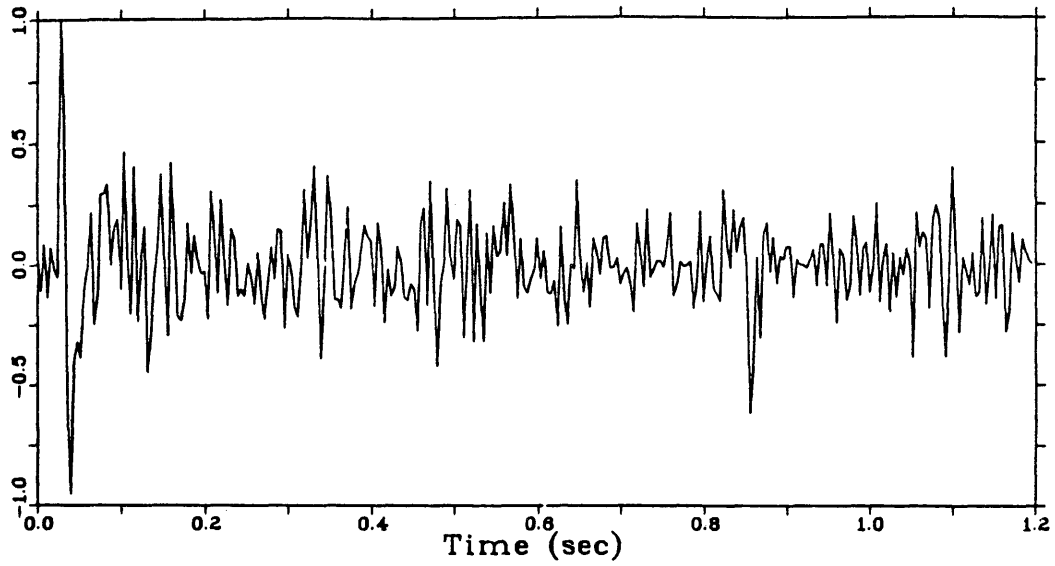

Figure 71. Deconvolved, phase-compensated, and wide-band filtered data (with 5% noise). Phase-compensation based on simulated signature.

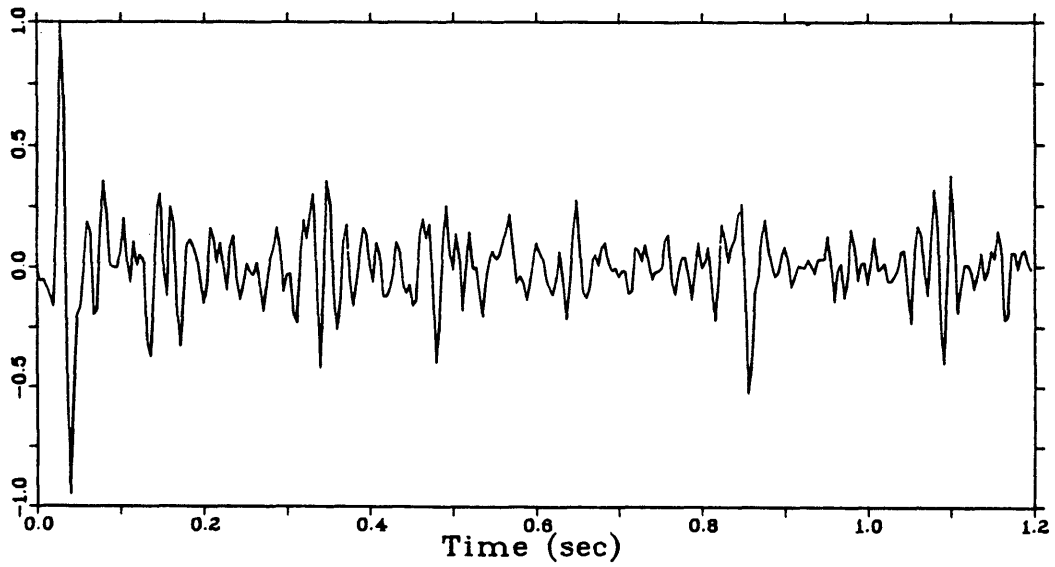

Figure 72. Wavelet processing of synthetic data (with 5% noise) using matched filter from simulated signature.

Table 2. Correlation coefficients between desired outputs and processing results for data with 5% additive noise.

Desired Outputs	$r(t)*d_1$	$r(t)*d_2$	$r(t)*d_3$	$r(t)*d_4$
SP-Decon $*d_1$	0.734 (-20) p=-1	0.570 (-20) p=-1	0.699 (-20) p=-1	0.712 (-20) p=-1
SP-Decon $*d_2$	0.588 (-20) p=-1	0.441 (-8)	0.530 (-20) p=-1	0.547 (-20) p=-1
PC by b_m $*d_1$	0.772	0.592	0.737	0.751
PC by b_m $*d_2$	0.611	0.579	0.633	0.631
PC by b_m $*d_3$	0.742	0.618	0.742	0.751
PC by b_e $*d_1$	0.716	0.549	0.688	0.715
PC by b_e $*d_2$	0.566	0.547	0.593	0.601
PC by b_e $*d_4$	0.736	0.600	0.733	0.759

SP - spiking deconvolution
b_m - measured signature
d_1 - 0,10-60,80 Hz filter
d_3 - matched filter from b_m
r - reflectivity function
PC - phase compensation
b_e - estimated signature
d_2 - 0,10-100,120 Hz filter
d_4 - matched filter from b_e
* - convolution

outputs. These problems are the result of phase distortion introduced by spiking deconvolution of noisy nonminimum phase data.

After phase compensation, the problems of inverse polarity and time delay are removed. In spite of the degradation caused by additive noise, it is still apparent from Table 2 that the matched filter suppresses noise better than the wide-band filter and preserves more high frequency components than the narrow-band filter. These results again show the superiority of the new method over conventional methods in the sense of preserving high frequency components and suppressing noise. These results also show the usability of the similated signature in wavelet processing.

According to the phase-compensation theory developed by Sengbush (1986), the level of white noise added to the autocorrelation of the source signature when computing the phase compensator should be the same as the level of natural noise in the data. When 5% processing noise is added in computing the phase compensator, which is then applied to the deconvolved data which contains 5% natural noise, the CC values between the processing results and the desired outputs are obtained and tabulated in Table 3. Comparing Table 3 with Table 2, we can see that all

Table 3. Correlation coefficients between desired outputs and processing results for data with 5% additive noise. (Phase-compensator is obtained by using 5% white noise)

Desired Outputs	$r(t)*d_1$	$r(t)*d_2$	$r(t)*d_3$	$r(t)*d_4$
PC by b_m $*d_1$	0.811	0.626	0.774	0.788
PC by b_m $*d_2$	0.685	0.623	0.701	0.700
PC by b_m $*d_3$	0.783	0.648	0.779	0.788
PC by b_e $*d_1$	0.785	0.608	0.753	0.775
PC by b_e $*d_2$	0.673	0.609	0.689	0.696
PC by b_e $*d_4$	0.795	0.644	0.786	0.812

PC - phase compensation
b_m - measured signature
d_1 - 0,10-60,80 Hz filter
d_3 - matched filter from b_m
* - convolution
r - reflectivity function
b_e - estimated signature
d_2 - 0,10-100,120 Hz filter
d_4 - match filter from b_e

processing results are improved by adding 5% white processing noise when computing the phase compensator.

CONCLUSIONS

Several conclusions can be drawn from the results of this study. They are summarized as follows:

1) The far-field signature of an airgun array can be simulated quite well by a gaussian-damped sine function. The calculated and measured signatures are very similar except for small differences in the tail part due to several possible reasons, including the residual bubble pulses, the effect of hydrophone and the difference of instrument responses in simulation and measurement.

2) It is possible to estimate the airgun array signature quite accurately from marine seismic data by using a scan method based on raypath and stacking theories. The successful estimation example proves the validity of the method.

3) It is apparent that the estimation method proposed in this study is much cheaper than most other methods because no special equipment is required in field survey and only few calculations are needed in estimation process. This advantage makes the method very useful when direct far-field measurements are not available or unreliable.

4) The zero-phase matched filter, derived from the source wavelet after passing through the instruments,

preserves high frequency components of the signal and suppresses noise amplified by deconvolution as much as possible. The processing method using such a matched filter avoids the difficulty of choosing a suitable d-filter, which is a common problem in conventional methods.

5) The estimated signature is accurate enough for wavelet processing purpose. The new wavelet processing method greatly improves the quality of seismic data for both real and synthetic cases. The advantage of the method is clearly shown by its very good performance.

SELECTED BIBLIOGRAPHY

Berkhout, A. J., 1977, Least-squares inverse filtering and wavelet deconvolution, Geophysics, v. 42, p. 1369.

Berteussen, K. A., and Alstad, O. J., 1985, Estimation of seabottom reflection characteristics and source signature from seabed reflected waves, Geophysics, v. 50, p. 1049.

Connelly, D., and Hart, D., 1985, Model-based wavelet processing of deconvolved seismic data, presented at 55th Annual International SEG Meeting, Washington D.C.

Dragoset, B., Hargreaves, N., and Larner, K., 1985, Air gun source instabilities and shot-by-shot signature deconvolution, presented at 55th Annual International SEG Meeting, Washington D.C.

Dragoset, W., 1984, A comprehensive method of evaluating the design of airguns and airgun arrays, The Leading Edge, v. 3, n. 10, p. 52.

Fourmann, J. M., 1974, Deconvolution of a recorded signature (Vapco and Wapco process), CGG Publication.

Gibson, B., and Larner, K., 1984, Predictive deconvolution and the zero-phase source, Geophysics, v. 49, p. 379.

Godfrey, B., and Montalbetti, J., 1984, A comparison between spiking and signature deconvolution, presented

at 54th Annual International SEG Meeting, Atlanta, Georgia.

Hargreaves, N. D., 1984, Far-field signatures by wavefield extrapolation, presented at 46th Annual EAEG Meeting, London, England.

Hatton, L., Worthington, M. H., and Makin, J., 1986, Seismic data processing: theory and practice, London, Blackwell Scientific Publications.

Jovanovich, D. B., Sumner, R. D., and Akins-Easterlin, S. L., 1983, Ghosting and marine signature deconvolution: a prerequisite for detailed seismic interpretation, Geophysics, v. 48, p. 1468.

Larner, K., Chambers, R., Yang, M., Lynn, W., and Wai, W., 1983, Coherent noise in marine seismic data, Geophysics, v. 48, p. 854.

Larner, K., Hale, D., Zinkham, S. M., and Hewlitt, C., 1982, Desired seismic characteristics of an airgun source, Geophysics, v. 47, p. 1273.

Liu, C. Y., 1985, Deconvolution study of P-wave land seismic source signature from a downhole array, presented at 55th Annual International SEG Meeting, Washington D.C.

Newman, B. J., 1985, Deconvolution of noisy seismic data: A method for prestack wavelet extraction, presented at

55th Annual International SEG Meeting, Washington D.C.

Newman, P., 1973, Divergence effects in a layered earth, Geophysics, v. 38, p. 481.

Parkes, G. E., Ziolkowski, A., Hatton, L., and Haugland, T., 1984, The signature of an air gun array: computation from near-field measurements including interactions - practical considerations, Geophysics, v. 48, p. 105

Robinson, E. A., 1954, Predictive decomposition of seismic traces with applications to seismic exploration, Ph. D. Thesis, M.I.T., Cambridge, MA; reprinted in Geophysics, v. 32, p. 418.

Safar, M. H., 1976, Efficient design of air-gun arrays, Geophysical Prospecting, v. 24, p. 773.

Safar, M. H., and Haskey, P., 1983, On the quality control of Datagun arrays, First Break, v. 1, n. 11, p. 13

Sengbush, R. L., 1983, Seismic exploration methods, Boston, IHRDC Press.

Sengbush, R. L., Hato, M., and Chang, H., 1986, Optimal compensation of time variance and nonminimum phase in Wiener-Robinson deconvolution, presented at Research Workshop on Deconvolution and Inversion, Rome, Italy.

Sengbush, R. L., and Hu, S. T., 1986, Wiener-Levinson deconvolution of nonminimum-phase seismic data, Paper

OTC 5160, Proceedings of the 18th Annual Offshore Technology Conference, Houston, Texas.

Shtivelman, V., and Loewenthal, D., 1986, Source wavelet estimation by upward extrapolation, presented at 56th Annual International SEG Meeting, Houston, Texas.

Vaage, S., and Ursin, B., 1987, Computation of signatures of linear airgun arrays, Geophysical Prospecting, v. 35, p. 281.

Voodg, N., DE, 1974, Wavelet shaping and noise reduction, Geophysical Prospecting, v. 22, p. 354.

Zhang, Y., 1985, Wavelet processing for marine seismic data, Oil Geophysical Prospecting, v. 20, p. 555, China.

Zhang, Y., 1986, Wavelet Analysis, Geophysical Prospecting for Petroleum, v. 25, p. 71, China.

Ziolkowski, A., 1982, An air gun model which includes heat transfer and bubble interactions, presented at 52th Annual International SEG Meeting, Dallas, Texas.

Ziolkowski, A., 1984, Deconvolution, Boston, IHRDC Press.

Ziolkowski, A., 1987, The determination of the far-field signature of an interacting array of marine seismic sources from the near-field measurements - results from the Delft Air Gun Experiment, First Break, v. 5, n. 1, p. 15.

Ziolkowski, A., Parkes, G., Hatton, L., and Haugland, T., 1982, The signature of an air gun array: computation from near-field measurements including interactions, Geophysics, v. 47, p. 1413.

APPENDIX A

THE CORRELATION COEFFICIENT

A measure of waveform similarity between two functions x and y is the correlation coefficient (CC value), which is defined by

$$CC = \frac{\max|\Phi_{xy}(\tau)|}{\sqrt{\Phi_x(0)\ \Phi_y(0)}},$$

where $\Phi_{xy}(\tau)$ is the cross-correlation of x and y; τ is the relative time shift; $\Phi_x(0)$ and $\Phi_y(0)$ are autocorrelation functions evaluated at the origin, which equal the total energies in x and y, respectively.

CC ranges from 0 to 1; CC=1 indicates identical waveforms independent of amplitude, polarity, and time shift, while CC=0 indicates uncorrelated waveforms. The shift τ_0 corresponding to maximum absolute value of Φ_{xy} is a measure of linear phase lag of x with respect to y. If $\Phi_{xy}(\tau_0)$ is negative, then polarity of y is inverse to that of x. Therefore, the CC value, shift τ_0 and relative polarity measure the waveform similarity of x and y.

APPENDIX B

PROGRAM FOR SIGNATURE ESTIMATION AND SYNTHETIC DATA GENERATION

```
C     ******************************************************
C
C      This program is written for following processings:
C
C               AIRGUN ARRAY SIGNATURE ESTIMATION
C
C               SYNTHETIC SEISMIC DATA GENERATION
C
C                     ( IN MARINE CASE )
C
C     ......................................................
C                  Written by:  Yuesheng Li
C
C                  Date:    August 10, 1987
C     ......................................................

      CHARACTER  ANS*1
      INTEGER    UN,UN1,UN2,UN3,UN4
      PARAMETER  (N=85,N1=85,UN=6,UN1=1,UN2=2,UN3=3,UN4=4,
     +           AMP=1.0,T=0.001,V=1520.0,V2=1850.0,D2=1.4,
     +           PR2=0.30,DS=7.5,DR=10.0,DB=50.0,X0=298.0,
     +           V3=1750.0,D3=1.4,PR3=0.30,DE=30.0,
     +           NN=300,ID0=0,R0=0.5)
      DIMENSION  ANOISE(NN),BR(NN),DIR(NN),ER(N),G(N),
     +           GS(N),H(N),R(NN),REFR(N),REV(NN),
     +           REV2(NN),RR(NN),S(N),SR(N),W(NN),WW(N),
     +           X(NN),XX(NN),Y(NN),YY(NN),Z(NN)
C
C --- Select processing type
C
      WRITE (UN,*)'This program can do two types of jobs:'
      WRITE (UN,*)'1. Airgun signature estimation.'
      WRITE (UN,*)'2. Synthetic seismic data generation.'
      WRITE (UN,*)'Please enter your choice:'
50    READ  (UN,*) IANS
      IF (IANS.EQ.2) GO TO 1200
      IF (IANS.NE.1) THEN
         WRITE (UN,*)'Your choice is wrong.'
```

```
         WRITE (UN,*)'Please enter your choice correctly:'
         GO TO 50
      END IF
C     ------------------------------------------------------
C                  Airgun Signature Estimation
C     ------------------------------------------------------
      ERRMIN=1.0
C
C --- Input of initial parameters
C
      WRITE (UN,*)'Enter error, damping, frequency, phase:'
      READ (UN,*) ERR0,C0,F0,PHI0
      WRITE (UN,*)'Impulse response of instrument is read.'
      CALL INPUT (H,N,UN,UN1)
      WRITE (UN,*)'Based on far-field signature?(Y/N)'
      READ (UN,'(A)') ANS
      IF (ANS.EQ.'Y'.OR.ANS.EQ.'y') GO TO 600
C     ....................................
C     Estimation is based on seismic data
C     ....................................
      WRITE (UN,*)'The marine data will be read.'
      CALL INPUT (W,NN,UN,UN1)
C
C --- Travel times and travel distances
C
      CALL DIRECT (DS,DR,X0,V,T,IDD,DIS1,T01)
      CALL SURREF (DS,DR,X0,V,T,IDS,DIS2,T02)
      ALPHA=ASIN (V/V2)
      XS=(DB-DS)*TAN(ALPHA)
      XR=(DB-DR)*TAN(ALPHA)
      X0XSXR=X0-XS-XR
C
C --- Scan entry
C
      DO 500 I=1,2
         PHI=PHI0+(I-1)
         DO 400 J=1,30
            F=F0+(J-1)*0.5
            DO 300 IJ=1,15
               C=C0+(IJ-1)*0.5
               IF (X0XSXR.LE.0.0) THEN
                  MINID=IDD
               ELSE
C
C --- Sea-bottom refraction
C
                  CALL REFRAC (DS,DR,DB,X0,V,V2,D2,PR2,
     +                         T,F,PHI,C,N,Y,DIS4,IDR)
```

```
                  MINID=MIN0 (IDD,IDR)
                  IDR1=IDR-MINID
                  CALL MOVE (IDR1,Y,N,REFR,N)
               END IF
C
C --- Sea-bottom reflections
C
               CALL REFLEC (DS,DR,DB,X0,V,V2,D2,PR2,T,
     +                     F,PHI,C,X,N1,XX,N,YY,DIS3,IDB)
C
C --- First subsurface reflections
C
               CALL REFLCT (DS,DR,DB,DE,X0,V,V2,V3,D2,D3,
     +              PR2,PR3,T,F,PHI,C,X,N1,XX,N,RR,DIS5,IDE)
C
C --- Relative time delay
C
               IDD1=IDD-MINID
               IDS1=IDS-MINID
               IDB1=IDB-MINID
               IDE1=IDE-MINID
C
C --- Direct wave & free-surface reflection
C
               CALL SOURCE (T01,X,N1,T,1.0,F,PHI,C)
               CALL SOURCE (T02,XX,N1,T,1.0,F,PHI,C)
C
C --- Stacking of all possible arrivals
C
               CALL MOVE (IDD1,X,N1,DIR,N)
               CALL MOVE (IDS1,XX,N1,SR,N)
               CALL MOVE (IDB1,YY,N,BR,N)
               CALL MOVE (IDE1,RR,N,ER,N)
               AMP2=DIS1/DIS2
               AMP3=DIS1/DIS3
               AMP5=DIS1/DIS5
               IF (X0XSXR.GT.0.0) THEN
                  AMP4=DIS1/DIS4
               END IF
               DO 100 JJ=1,N
                  S(JJ)=DIR(JJ)-AMP2*SR(JJ)+AMP3*BR(JJ)
     +                      +AMP5*ER(JJ)
                  IF (X0XSXR.GT.0.0) THEN
                     S(JJ)=S(JJ)+AMP4*REFR(JJ)
                  END IF
100            CONTINUE
C
C --- Add effect of instruments
```

```
C
                CALL CONVO1 (S,N,H,N,Y,N)
                CALL NORMAL (Y,N)
C
C --- Similarity analysis
C
                DO 200 K=185,190
                   CALL MOVE  (-K+1,W,NN,WW,N)
                   CALL NORMAL (WW,N)
                   CALL ERROR (WW,Y,N,ERR)
                   IF (ERR.LE.ERR0) THEN
C
C --- Output of results
C
                      TIME=T*(K-1)
                      WRITE (UN,*)'ERR=',ERR,' F=',F
                      WRITE (UN,* 'PHI=',PHI,' C=',C
                      WRITE (UN,*)'TIME=',TIME
                      CALL SOURCE (0.0,X,N1,T,1.0,F,PHI,C)
                      CALL GHOST(X,N1,GS,N,DS,V,T,G,F,PHI,C)
                      CALL CONVO1 (GS,N,H,N,G,N)
                      CALL CONVO1 (X,N1,H,N,Z,N)
                      CALL FACTOR (X,X,N1,AMP)
                      CALL OUTPUT (X,N1,T,UN2,1)
                      CALL FACTOR (Z,Z,N,AMP)
                      CALL OUTPUT (Z,N,T,7,1)
                      CALL FACTOR (G,G,N,AMP)
                      CALL OUTPUT (G,N,T,8,1)
                      CALL FACTOR (Y,Y,N,AMP)
                      CALL OUTPUT (Y,N,T,UN3,1)
                      CALL FACTOR (WW,WW,N,AMP)
                      CALL OUTPUT (WW,N,T,UN4,1)
                      GO TO 1300
                   ELSE IF (ERR.LT.ERRMIN) THEN
                      ERRMIN=ERR
                   END IF
200             CONTINUE
300          CONTINUE
400       CONTINUE
500    CONTINUE
       GO TO 1100
C      ........................................
C      Estimation is based on far-field signature
C      ........................................
600    WRITE (UN,*)'Far-field signature will be read now.'
       CALL INPUT (W,N,UN,UN1)
C
C --- Scan entry
```

```
C
      DO 1000 I=1,2
         PHI=PHI0+(I-1)
         DO 900 J=1,30
            F=F0+(J-1)*0.5
            DO 800 IJ=1,15
               C=C0+(IJ-1)*1.0
C
C --- Simulate airgun signature & ghost
C
               CALL SOURCE (0.0,X,N1,T,AMP,F,PHI,C)
               CALL GHOST (X,N1,XX,N,DS,V,T,G,F,PHI,C)
C
C --- Add effect of instruments
C
               CALL CONVO1(XX,N,H,N,Y,N)
C
C --- Simularity analysis
C
               DO 700 K=15,N/3
                  CALL MOVE  (0,W,N,WW,N)
                  CALL MOVE  (0,Y,N,YY,N)
                  CALL NORMAL (WW(K),(N-K+1))
                  CALL NORMAL (YY,(N-K+1))
                  CALL ERROR (WW(K),YY,(N-K+1),ERR)
                  IF (ERR.LE.ERR0) THEN
C
C --- Output of final result
C
                     TIME=T*(K-1)
                     WRITE (UN,*)'ERR=',ERR,' F=',F
                     WRITE (UN,*)'PHI=',PHI,' C=',C
                     WRITE (UN,*)'TIME=',TIME
                     CALL FACTOR (X,X,N1,AMP)
                     CALL OUTPUT (X,N1,T,UN2,1)
                     CALL MOVE ((K-1),Y,N,Z,N)
                     CALL FACTOR (Z,Z,N,AMP)
                     CALL OUTPUT (Z,N,T,UN3,1)
                     CALL FACTOR (W,W,N,AMP)
                     CALL OUTPUT (W,N,T,UN4,1)
                     GO TO 1300
                  ELSE IF (ERR.LT.ERRMIN) THEN
                     ERRMIN=ERR
                  END IF
700            CONTINUE
800         CONTINUE
900      CONTINUE
1000  CONTINUE
```

```
C
C --- Unable to obtain satisfactory result
C
1100  WRITE (UN,*) 'The minimum error is',ERRMIN
      STOP 'The model is incorrect.'
C     ------------------------------------------------
C            Synthetic Marine Seismic Data Generation
C     ------------------------------------------------
1200  CONTINUE
      WRITE (UN,*)'Far-field signature will be read now.'
      CALL INPUT (WW,N,UN,UN1)
C
C --- Remove delay time
C
      CALL MOVE (-ID0,WW,N,W,N)
C
      CALL DIRECT (DS,DR,X0,V,T,IDD,DIS,T0)
      CALL BOTREF (DB,DS,DR,X0,V,T,IDB,DIS,T0)
C
C --- Calculate receiver ghost operator
C
      CALL GHOST1 (DR,T,V,G,N)
      WRITE (UN,*)'Reflectivity function will be read.'
      CALL INPUT (R,NN,UN,UN1)
      DELAY=2.0*DB/V
      ID=IFIX (DELAY/T+0.5)
      ID=IDB-ID
      CALL MOVE (ID,R,NN,RR,NN)
      WRITE (UN,*)'Include reverberation ? (Y/N)'
      READ (UN,'(A)') ANS
      IF (ANS.EQ.'Y'.OR.ANS.EQ.'y') THEN
C
C --- Calculate reverberation operator
C
         CALL REVERB (DB,T,V,R0,REV,NN)
         CALL CONVO1 (REV,NN,REV,NN,REV2,NN)
         CALL CONVO1 (RR,NN,REV2,NN,R,NN)
      ELSE
         CALL MOVE (0,RR,NN,R,NN)
      END IF
C
      WRITE (UN,*)'Include ghost ? (Y/N)'
      READ (UN,'(A)') ANS
      IF (ANS.EQ.'Y'.OR.ANS.EQ.'y') THEN
         CALL CONVO1 (G,N,R,NN,RR,NN)
         CALL MOVE (0,RR,NN,R,NN)
      END IF
C
```

```
C --- Generate synthetic data
C
      CALL CONVO1 (W,N,R,NN,BR,NN)
      CALL MOVE (IDD,W,N,DIR,NN)
      DO I=1,NN
         Y(I)=DIR(I)+BR(I)
      END DO
      WRITE (UN,*)'Add natural noise? (Y/N)'
      READ (UN,'(A)') ANS
      IF (ANS.EQ.'Y'.OR.ANS.EQ.'y') THEN
         WRITE (UN,*)'Enter noise level:'
         READ (UN,*) A0
C
C --- Generate random process
C
         CALL RANDOM (ANOISE,NN,0.0,1.0)
         CALL NORMAL (ANOISE,NN)
         CALL NORMAL (Y,NN)
         DO I=1,NN
            Y(I)=Y(I)+(SQRT(A0))*ANOISE(I)
         END DO
      END IF
      CALL FACTOR (Y,Y,NN,AMP)
      CALL OUTPUT (Y,NN,T,UN3,1)
1300  STOP 'Finish all calculation.'
      END
C     ******************************************************
      SUBROUTINE INPUT (A,LA,IUN,IUN1)
C     ......................................................
C       INPUT reads data from data file on disk.
C     ......................................................

      CHARACTER  INFILE*12
      DIMENSION  A(LA)

      CALL ZERO (A,LA)
      WRITE (IUN,*) 'Enter the input filename:'
      READ  (5,'(A)') INFILE
      OPEN  (UNIT=IUN1,FILE=INFILE,STATUS='OLD',ERR=99)
      DO I=1,LA
         READ (IUN1,*,END=95) TT,A(I)
      END DO
95    CLOSE (IUN1)
      RETURN
99    STOP 'The input filename is wrong.'
      END
C     ******************************************************
      SUBROUTINE SOURCE (T0,B,LB,TSR,AMP,F,PHI,CONST)
```

```
C     ....................................................
C       SOURCE simulates airgun array signature.
C     ....................................................

      DIMENSION  B(LB)

      ALPHA = CONST
      PI=180.0
      DO I=1,LB
         TIME=TSR*(I-1)+T0
         B(I)=AMP*EXP(-ALPHA*TIME*TIME)
     +         *SIND(2.0*PI*F*TIME+PHI)
      END DO
      RETURN
      END
C     ******************************************************
      SUBROUTINE FACTOR (A,B,L,AMP)
C     ....................................................
C       FACTOR normalizes data by amplitude.
C     ....................................................

      REAL MAX
      DIMENSION  A(L), B(L)

      MAX=ABS(A(1))
      DO 30 I=2,L
         IF (ABS(A(I)).LE.MAX) GO TO 30
         MAX=ABS(A(I))
30    CONTINUE
      DO I=1,L
         B(I)=AMP*A(I)/MAX
      END DO
      RETURN
      END
C     ******************************************************
      SUBROUTINE ZERO (A,LA)
C     ....................................................
C       ZERO puts zero value in each element of array A.
C     ....................................................

      DIMENSION  A(LA)

      DO I=1,LA
         A(I)=0.0
      END DO
      RETURN
      END
C     ******************************************************
```

```
      SUBROUTINE CONVO1 (A,LA,B,LB,C,LC)
C     ..................................................
C       CONVOL1 computes the convolution of A with B.
C     ..................................................

      DIMENSION  A(LA),B(LB),C(LC)

      CALL ZERO (C,LC)
      DO 20 I=1,LA
         DO 10 J=1,LB
            K=I+J-1
            IF (K.GT.LC) GO TO 20
            C(K)=C(K)+A(I)*B(J)
10       CONTINUE
20    CONTINUE
      RETURN
      END
C     ******************************************************
      SUBROUTINE GHOST (A,LA,B,LB,DEEP,VELO,T,AA,F,PHI,C)
C     ..................................................
C       GHOST computes ghost and adds it to direct wave.
C     ..................................................

      DIMENSION  A(LA),AA(LA),B(LB)

      AMPL=-1.0
      DELAY=2.0*DEEP/VELO
      CALL WAVE (DELAY,T,F,PHI,C,AA,LA,ID)
      DO 50 I=1,LA
         IF (I.LE.ID) THEN
            B(I)=A(I)
         ELSE
            B(I)=A(I)+AMPL*AA(I-ID)
         END IF
50    CONTINUE
      IF (LA.EQ.LB) GO TO 77
      IF ((LA+ID).LT.LB) THEN
         DO 60 I=LA+1,LA+ID
            B(I)=AMPL*AA(I-ID)
60       CONTINUE
         DO 70 I=LA+ID+1,LB
            B(I)=0.0
70       CONTINUE
      ELSE
         DO 75 I=LA+1,LB
            B(I)=AMPL*AA(I-ID)
75       CONTINUE
      END IF
```

```
77     RETURN
       END
C      ******************************************************
       SUBROUTINE GHOST1 (DEEP,TSR,VELO,A,LA)
C      ......................................................
C        GHOST1 computes ghost operator.
C      ......................................................

       DIMENSION  A(LA)

       AMPL=-1.0
       DELAY=2.0*DEEP/VELO
       ID=IFIX (DELAY/TSR+0.5)
       CALL ZERO (A,LA)
       A(1)=1.0
       IF (ID.LT.LA) THEN
          A(1+ID)=AMPL
       END IF
       RETURN
       END
C      ******************************************************
       SUBROUTINE REVERB (DEEP,TSR,VELO,RC,A,LA)
C      ......................................................
C        REVERB computes reverberation operator.
C      ......................................................

       DIMENSION  A(LA)

       DELAY=2.0*DEEP/VELO
       NUM=IFIX (TSR*(LA-1)/DELAY)
       CALL ZERO (A,LA)
       A(1)=1.0
       IF (NUM.EQ.0) GO TO 10
       DO I=1,NUM
          CI=FLOAT (I)
          ID=IFIX (DELAY*CI/TSR+0.5)
          A(1+ID)=(-RC)**I
       END DO
10     RETURN
       END
C      ******************************************************
       SUBROUTINE DIRECT (DS,DR,X0,V,T,ID,DIS,T0)
C      ......................................................
C        DIRECT computes travel distance and travel time
C               of direct arrival.
C      ......................................................

       DIS=SQRT (X0**2+(DR-DS)**2)
```

```
      TIME=DIS/V
      CALL TDELAY (TIME,T,ID,T0)
      RETURN
      END
C     ******************************************************
      SUBROUTINE SURREF (DS,DR,X0,V,T,ID,DIS,T0)
C     ......................................................
C       SURREF computes travel distance and travel time
C             of free-surface reflection.
C     ......................................................

      DIS=SQRT (X0**2+(DR+DS)**2)
      TIME=DIS/V
      CALL TDELAY (TIME,T,ID,T0)
      RETURN
      END
C     ******************************************************
      SUBROUTINE BOTREF (DB,DS,DR,X0,V,T,ID,DIS,T0)
C     ......................................................
C       BOTREF computes travel distance and travel time
C             of sea-bottom reflection.
C     ......................................................

      DIS=SQRT (X0**2+(2*DB-DS-DR)**2)
      TIME=DIS/V
      CALL TDELAY (TIME,T,ID,T0)
      RETURN
      END
C     ******************************************************
      SUBROUTINE TDELAY (TIME,T,ID,T0)
C     ......................................................
C       TDELAY converts traveltime to sampling number.
C     ......................................................

      ID=IFIX (TIME/T)
      TID=T*ID
      ERR=ABS (TIME-TID)
      IF (ERR.GE.0.0002) THEN
         ID=ID+1
         T0=T*ID-TIME
      ELSE
         T0=0.0
      END IF
      RETURN
      END
C     ******************************************************
      SUBROUTINE REFLEC (DS,DR,DB,X0,V1,V2,D2,PR2,T,
     +                   F,PHI,C,A,LA,B,L,REFL,DIS,ID)
```

```
C     ..................................................
C       REFLEC computes reflections from sea bottom.
C     ..................................................

      DIMENSION  A(LA),B(L),REFL(L),D(4),ANG(4),R(4)

      CALL ZERO (REFL,L)
      D1=1.02
      PR1=0.5
C
C --- NR is the order of reverberation
C
      DO NR=0,2
         CALL DISTAN (NR,DS,DR,DB,X0,V1,D,ANG)
         CALL AI (ANG(1),V1,V2,D1,D2,PR1,PR2,R(1),S1,P2,S2)
         CALL AI (ANG(2),V1,V2,D1,D2,PR1,PR2,R(2),S1,P2,S2)
         CALL AI (ANG(3),V1,V2,D1,D2,PR1,PR2,R(3),S1,P2,S2)
         CALL AI (ANG(4),V1,V2,D1,D2,PR1,PR2,R(4),S1,P2,S2)
         IF (NR.EQ.0) THEN
            DIS=D(1)
            TIME=DIS/V1
            CALL WAVE (TIME,T,F,PHI,C,REFL,LA,ID)
            DO I=1,LA
               REFL(I)=REFL(I)*R(1)
            END DO
            J0=2
         ELSE
            J0=1
         END IF
         DO J=J0,4
            TT=D(J)/V1
            CALL WAVE (TT,T,F,PHI,C,A,LA,ID1)
            I0=ID1-ID
            CALL MOVE (I0,A,LA,B,L)
            IF (J.EQ.2.OR.J.EQ.3) THEN
               AMP=-1.0
            ELSE
               AMP=1.0
            END IF
            AMP1=AMP*((-R(J))**NR)*(DIS/D(J))*R(J)
            DO I=1,L
               REFL(I)=REFL(I)+AMP1*B(I)
            END DO
         END DO
      END DO
      RETURN
      END
C     ****************************************************
```

```
      SUBROUTINE DISTAN (NR,DS,DR,DB,X0,V,D,ANG)
C     ..................................................
C       DISTAN computes travel distances and incident
C              angles of reflections from sea-bottom.
C     ..................................................

      DIMENSION D(4),ANG(4)

      N=NR+1
      CALL ZERO (D,4)
      CALL ZERO (ANG,4)
      XX=X0*X0
      Z1=2.0*N*DB-DS-DR
      Z2=2.0*N*DB+DS-DR
      Z3=2.0*N*DB-DS+DR
      Z4=2.0*N*DB+DS+DR
      D(1)=SQRT (XX+Z1**2)
      D(2)=SQRT (XX+Z2**2)
      D(3)=SQRT (XX+Z3**2)
      D(4)=SQRT (XX+Z4**2)
      ANG(1)=ATAND (X0/Z1)
      ANG(2)=ATAND (X0/Z2)
      ANG(3)=ATAND (X0/Z3)
      ANG(4)=ATAND (X0/Z4)
      RETURN
      END
C     ******************************************************
      SUBROUTINE REFLCT (DS,DR,DB,DE,X0,V1,V2,V3,D2,D3,PR2,
     +                   PR3,T,F,PHI,C,A,LA,B,L,REFL,DIS,ID)
C     ..................................................
C       REFLCT computes reflections from first subsurface.
C     ..................................................

      DIMENSION  A(LA),B(L),REFL(L),DW(4),AG0(4),
     +           AG1(4),AG2(4),R1(4),R2(4)

      CALL ZERO (REFL,L)
      D1=1.02
      PR1=0.5
      ZERR=V3*D3-V2*D2
      IF (ZERR.LT.0.0) THEN
         SIGN=-1.0
      ELSE
         SIGN=1.0
      END IF
C
C --- NR is the order of reverberation
C
```

```
DO NR=0,2
   DW(1)=2*(NR+1)*DB-DS-DR
   DW(2)=2*(NR+1)*DB+DS-DR
   DW(3)=2*(NR+1)*DB-DS+DR
   DW(4)=2*(NR+1)*DB+DS+DR
   AG0(1)=ATAND (X0/(DW(1)+2*DE))
   AG0(2)=ATAND (X0/(DW(2)+2*DE))
   AG0(3)=ATAND (X0/(DW(3)+2*DE))
   AG0(4)=ATAND (X0/(DW(4)+2*DE))
   CALL ANGUL(DW(1),DE,X0,V1,V2,AG0(1),AG1(1),AG2(1))
   CALL ANGUL(DW(2),DE,X0,V1,V2,AG0(2),AG1(2),AG2(2))
   CALL ANGUL(DW(3),DE,X0,V1,V2,AG0(3),AG1(3),AG2(3))
   CALL ANGUL(DW(4),DE,X0,V1,V2,AG0(4),AG1(4),AG2(4))
   CALL AI(AG1(1),V1,V2,D1,D2,PR1,PR2,R1(1),S1,P2,S2)
   CALL AI(AG1(2),V1,V2,D1,D2,PR1,PR2,R1(2),S1,P2,S2)
   CALL AI(AG1(3),V1,V2,D1,D2,PR1,PR2,R1(3),S1,P2,S2)
   CALL AI(AG1(4),V1,V2,D1,D2,PR1,PR2,R1(4),S1,P2,S2)
   CALL AI(AG2(1),V2,V3,D2,D3,PR2,PR3,R2(1),S1,P2,S2)
   CALL AI(AG2(2),V2,V3,D2,D3,PR2,PR3,R2(2),S1,P2,S2)
   CALL AI(AG2(3),V2,V3,D2,D3,PR2,PR3,R2(3),S1,P2,S2)
   CALL AI(AG2(4),V2,V3,D2,D3,PR2,PR3,R2(4),S1,P2,S2)
   IF (NR.EQ.0) THEN
      DIS1=DW(1)/COSD(AG1(1))
      DIS2=2*DE/COSD(AG2(1))
      DIS=DIS1+DIS2
      TIME=DIS1/V1+DIS2/V2
      CALL WAVE (TIME,T,F,PHI,C,REFL,LA,ID)
      AMP=(1-R1(1))*(1+R1(1))*R2(1)*SIGN
      DO I=1,LA
         REFL(I)=REFL(I)*AMP
      END DO
      J0=2
   ELSE
      J0=1
   END IF
   DO J=J0,4
      DIST1=DW(J)/COSD(AG1(J))
      DIST2=2*DE/COSD(AG2(J))
      DIST=DIST1+DIST2
      TT=DIST1/V1+DIST2/V2
      CALL WAVE (TT,T,F,PHI,C,A,LA,ID1)
      I0=ID1-ID
      CALL MOVE (I0,A,LA,B,L)
      IF (J.EQ.2.OR.J.EQ.3) THEN
         AMP=-1.0
      ELSE
         AMP=1.0
      END IF
```

```
            AMP1=AMP*((-R1(J))**NR)*(DIS/DIST)
            AMP2=SIGN*(1-R1(J))*(1+R1(J))*R2(J)*AMP1
            DO I=1,L
               REFL(I)=REFL(I)+AMP2*B(I)
            END DO
         END DO
      END DO
      RETURN
      END
C     ******************************************************
      SUBROUTINE  ANGUL (DW,DE,X0,V1,V2,ALP0,ALP1,ALP2)
C     ......................................................
C       ANGUL computes incident angle and transmitted
C             compressional angle numerically.
C     ......................................................

      DIMENSION   ERR(150)

      DO I=1,150
         ALP2=ALP0+(I-1)*0.1
         ALP1=ASIND ((V1/V2)*SIND(ALP2))
         X=DW*TAND(ALP1)+2*DE*TAND(ALP2)
         ERR(I)=X0-X
      END DO
      ERRMIN=ABS (ERR(1))
      J=1
      DO I=2,150
         C=ABS (ERR(I))
         IF (ERRMIN.GT.C) THEN
            ERRMIN=C
            J=I
         END IF
      END DO
      ALP2=ALP0+(J-1)*0.1
      ALP1=ASIND ((V1/V2)*SIND(ALP2))
      RETURN
      END
C     ******************************************************
      SUBROUTINE REFRAC (DS,DR,DB,X0,V1,V2,D2,PR2,
     +                   T,F,PHI,C,L,REFR,DIS,ID)
C     ......................................................
C       REFRAC computes refraction from sea bottom.
C     ......................................................

      DIMENSION  REFR(L)

      D1=1.02
      PR1=0.5
```

```
      CALL ZERO (REFR,L)
      ALPHA=ASIN (V1/V2)
      ALPHA1=ASIND (V1/V2)
      CALL AI (ALPHA1,V1,V2,D1,D2,PR1,PR2,R0,S1,P2,S2)
      XS=(DB-DS)*TAN(ALPHA)
      ZS=(DB-DS)/COS(ALPHA)
      XR=(DB-DR)*TAN(ALPHA)
      ZR=(DB-DR)/COS(ALPHA)
      TIME=(X0-XS-XR)/V2 + (ZS+ZR)/V1
      DIS=X0-XS-XR+ZS+ZR
      CALL WAVE (TIME,T,F,PHI,C,REFR,L,ID)
      DO I=1,L
         REFR(I)=REFR(I)*R0
      END DO
      RETURN
      END
C     ******************************************************
      SUBROUTINE WAVE (DELAY,T,F,PHI,C,A,LA,ID)
C     ......................................................
C       WAVE computes signature according to travel
C            time and sampling interval.
C     ......................................................

      DIMENSION  A(LA)

      CALL ZERO (A,LA)
      CALL TDELAY (DELAY,T,ID,T0)
      CALL SOURCE (T0,A,LA,T,1.0,F,PHI,C)
      RETURN
      END
C     ******************************************************
      SUBROUTINE AI (ANG,VP1,VP2,D1,D2,PR1,PR2,P1,S1,P2,S2)
C     ......................................................
C       AI computes reflection coefficient by solving
C          Zoeppritz equations.
C     ......................................................

      REAL VP1,VP2,D1,D2,PR1,PR2,ANG,P1,P2,S1,S2,EP1,
     +     EP2,ES1,ES2,TW(4),CN(4),SW(4),VW(4)
      COMPLEX ZOP(4,5),X(5),RI,PV1,PV2,CW(4)
      CHARACTER TSP*2

      N=4
      TSP='P'
      IF (PR1.EQ.0.5) THEN
         VS1=0.0
      ELSE
         VS1=VP1/(2.0*(PR1-1.0)/(2.0*PR1-1.0))**.5
```

```
      END IF
      IF (PR2.EQ.0.5) THEN
         VS2=0.0
      ELSE
         VS2=VP2/(2.0*(PR2-1.0)/(2.0*PR2-1.0))**.5
      END IF
      VW(1)=VP1
      VW(2)=VS1
      VW(3)=VP2
      VW(4)=VS2
      IF (TSP.EQ.'P'.OR.TSP.EQ.'p') THEN
         V=VP1
      ELSE
         V=VW(2)
      END IF
      RI=(0.0,-1.0)
      AP=ANG
      SAP=SIND(AP)
      IF (AP.LT.90.0) TAP=TAND(AP)
      DO 10 J=1,4
         SW(J)=VW(J)/V*SAP
         CW(J)=(ABS(1.0-SW(J)*SW(J)))**0.5
10    CONTINUE
      IF (AP.EQ.0.0) THEN
         DO 15 J=1,4
            IF (VW(J).EQ.0.0) THEN
               TW(J)=0.0
            ELSE
               TW(J)=V/VW(J)
            END IF
            CW(J)=1.0
            CN(J)=1.0
15       CONTINUE
         GO TO 25
      END IF
      DO 20 J=1,4
         IF (SW(J).GT.1.0) THEN
            CW(J)=CW(J)*RI
            TW(J)=0.0
            CN(J)=0.0
         ELSE IF (VW(J).EQ.0.0) THEN
            TW(J)=0.0
            CN(J)=0.0
         ELSE
            TW(J)=TAP/SW(J)*(1.0-SW(J)*SW(J))**0.5
            CN(J)=1.0
         END IF
20    CONTINUE
```

```
      IF (AP.EQ.90.0) THEN
         IF (TSP.EQ.'P'.OR.TSP.EQ.'p') THEN
            TW(1)=1.0
         ELSE
            TW(2)=1.0
         END IF
      END IF
25    IF (TSP.EQ.'SH'.OR.TSP.EQ.'sh') THEN
         IF (AP.EQ.90.0) THEN
            IF (VW(2).EQ.VW(4).AND.VW(2).NE.0.0) THEN
               CW(2)=1.0
               CW(4)=1.0
            ELSE IF (VW(4).EQ.0.0) THEN
               CW(2)=1.0
            END IF
         END IF
         PV1=D1*VW(2)*CW(2)
         PV2=D2*VW(4)*CW(4)
         X(1)=0.0
         X(2)=(PV1-PV2)/(PV1+PV2)
         X(3)=0.0
         X(4)=2.0*PV1/(PV1+PV2)
         GO TO 60
      END IF
      ZOP(1,1)=-CW(1)
      ZOP(2,1)=SW(1)
      ZOP(3,1)=-(CW(2)*CW(2)-SW(2)*SW(2))
      ZOP(4,1)=VW(2)**2*2.0*SW(1)*CW(1)
      ZOP(1,2)=SW(2)
      ZOP(2,2)=CW(2)
      ZOP(3,2)=VW(2)/VW(1)*2.0*SW(2)*CW(2)
      ZOP(4,2)=VW(1)*VW(2)*(CW(2)*CW(2)-SW(2)*SW(2))
      ZOP(1,3)=-CW(3)
      ZOP(2,3)=-SW(3)
      ZOP(3,3)=D2/D1*VP2/VW(1)*(CW(4)*CW(4)-SW(4)*SW(4))
      ZOP(4,3)=D2/D1*VW(4)**2*VW(1)/VW(3)*2.0*SW(3)*CW(3)
      ZOP(1,4)=-SW(4)
      ZOP(2,4)=CW(4)
      ZOP(3,4)=D2/D1*VW(4)/VW(1)*2.0*SW(4)*CW(4)
      ZOP(4,4)=-D2/D1*VW(4)*VW(1)*(CW(4)*CW(4)-SW(4)*SW(4))
      IF (TSP.EQ.'P'.OR.TSP.EQ.'p') THEN
         ZOP(1,5)=-CW(1)
         ZOP(2,5)=-SW(1)
         ZOP(3,5)=CW(2)*CW(2)-SW(2)*SW(2)
         ZOP(4,5)=VW(2)**2*2.0*SW(1)*CW(1)
      ELSE
         ZOP(1,5)=SW(2)
         ZOP(2,5)=-CW(2)
```

```
         ZOP(3,5)=-VW(2)/VW(1)*2.0*SW(2)*CW(2)
         ZOP(4,5)=(CW(2)*CW(2)-SW(2)*SW(2))*VW(1)*VW(2)
      END IF
      IF (VW(2).EQ.0.0.AND.VW(4).EQ.0.0) THEN
         DO 30 J=1,5
            ZOP(2,J)=ZOP(3,J)
30       CONTINUE
         ZOP(1,2)=ZOP(1,3)
         ZOP(2,2)=ZOP(2,3)
         ZOP(1,3)=ZOP(1,5)
         ZOP(2,3)=ZOP(2,5)
         CALL SOL(N,2,ZOP,X)
         X(3)=X(2)
         X(2)=0.0
         X(4)=0.0
         GO TO 60
      ELSE IF (VW(2).EQ.0.0.AND.VW(4).NE.0.0) THEN
         DO 35 J=1,5
            ZOP(2,J)=ZOP(3,J)
            ZOP(3,J)=ZOP(4,J)
35       CONTINUE
         DO 40 J=1,3
            ZOP(J,2)=ZOP(J,3)
            ZOP(J,3)=ZOP(J,4)
            ZOP(J,4)=ZOP(J,5)
40       CONTINUE
         CALL SOL(N,3,ZOP,X)
         X(4)=X(3)
         X(3)=X(2)
         X(2)=0.0
         GO TO 60
      ELSE IF (VW(2).NE.0.0.AND.VW(4).EQ.0.0) THEN
         DO 45 J=1,5
            ZOP(2,J)=ZOP(3,J)
            ZOP(3,J)=ZOP(4,J)
45       CONTINUE
         DO 50 J=1,3
            ZOP(J,4)=ZOP(J,5)
50       CONTINUE
         CALL SOL(N,3,ZOP,X)
         X(4)=0.0
         GO TO 60
      ELSE
         CALL SOL(N,N,ZOP,X)
      END IF
60    P1=X(1)*CONJG(X(1))
      S1=X(2)*CONJG(X(2))
      P2=X(3)*CONJG(X(3))
```

```
      S2=X(4)*CONJG(X(4))
      EP1=P1*TW(1)*(VW(1)/V)**2
      ES1=S1*TW(2)*(VW(2)/V)**2
      EP2=D2/D1*P2*TW(3)*(VW(3)/V)**2
      ES2=D2/D1*S2*TW(4)*(VW(4)/V)**2
      P1=P1**.5*CN(1)
      S1=S1**.5*CN(2)
      P2=P2**.5*CN(3)
      S2=S2**.5*CN(4)
      RETURN
      END
C     ********************************************************
      SUBROUTINE SOL (MN,MK,A,X)
C     ........................................................
C       SOL solves high order equations by using
C           Gauss's elimination method.
C     ........................................................
      INTEGER MN,MK,K1,NN1
      COMPLEX A(MN,MN+1),X(MN),XA,T,C,Q,W
      REAL E,R

      N=MK
      DO 60 K=1,N-1
         XA=A(K,K)
         DO 10 K1=K,N
            E=A(K1,K)*CONJG(A(K1,K))
            R=CONJG(XA)*XA
            IF (E.GE.R) THEN
               XA=A(K1,K)
               II=K1
            END IF
10       CONTINUE
         IF (II.NE.K) THEN
            DO 20 N1=K,N+1
               T=A(K,N1)
               A(K,N1)=A(II,N1)
               A(II,N1)=T
20          CONTINUE
         END IF
         DO 30 NN1=K+1,N
            X(NN1)=A(NN1,K)
30       CONTINUE
         DO 50 J=K,N+1
            DO 40 I=K+1,N
               A(I,J)=A(I,J)-A(K,J)*X(I)/A(K,K)
40          CONTINUE
50       CONTINUE
60    CONTINUE
```

```
      X(N)=A(N,N+1)/A(N,N)
      DO 80 IS=N-1,1,-1
         C=0.0
         DO 70 IT=IS+1,N
            C=C-A(IS,IT)*X(IT)
70       CONTINUE
         X(IS)=(A(IS,N+1)+C)/A(IS,IS)
80    CONTINUE
      RETURN
      END
C     ******************************************************
      SUBROUTINE NORMAL (A,LA)
C     ......................................................
C       NORMAL normalizes data by energy.
C     ......................................................

      DIMENSION  A(LA)
C
      SUM=0.0
      DO I=1,LA
         SUM=SUM+A(I)*A(I)
      END DO
      SUM=SQRT(SUM)
      DO I=1,LA
         A(I)=A(I)/SUM
      END DO
      RETURN
      END
C     ******************************************************
      SUBROUTINE  MOVE (ID,A,LA,B,LB)
C     ......................................................
C       MOVE moves data from array A to array B with
C            shift ID.
C     ......................................................

      DIMENSION   A(LA),B(LB)

      CALL ZERO (B,LB)
      IF (ID.GE.LB) RETURN
      LMIN=MIN0 ((LA+ID),LB)
      IF (ID.GE.0) THEN
         DO 10 I=ID+1,LMIN
            B(I)=A(I-ID)
10       CONTINUE
      ELSE
         DO 20 I=1,LMIN
            B(I)=A(I-ID)
20       CONTINUE
```

```
      END IF
      RETURN
      END
C     ******************************************************
      SUBROUTINE  ERROR (A,B,L,ERR)
C     ......................................................
C       ERROR computes error between two signatures.
C     ......................................................

      DIMENSION   A(L),B(L)
C
      SUM1=0.0
      SUM2=0.0
      SUM3=0.0
      DO 10 I=1,L
         SUM1=SUM1+A(I)*B(I)
         SUM2=SUM2+A(I)*A(I)
         SUM3=SUM3+B(I)*B(I)
10    CONTINUE
      AABB=SQRT(SUM2*SUM3)
      ERR=1.0-SUM1/AABB
      RETURN
      END
C     ******************************************************
      SUBROUTINE OUTPUT (A,LA,TSR,IUN,IFLAG)
C     ......................................................
C       OUTPUT writes data into a file.
C     ......................................................

      DIMENSION  A(LA)

      IF (IFLAG.NE.1) GO TO 20
      DO 10 I=1,LA
         TIME=TSR*(I-1)
         WRITE (IUN,*) TIME,A(I)
10    CONTINUE
      GO TO 40
20    AZERO=0.0
      DO 30 I=1,LA
         TIME=TSR*(I-1)
         IF (A(I).EQ.0.0) THEN
            WRITE (IUN,*) TIME,AZERO
         ELSE
            WRITE (IUN,*) TIME,AZERO
            WRITE (IUN,*) TIME,A(I)
            WRITE (IUN,*) TIME,AZERO
         END IF
30    CONTINUE
```

```
40     CLOSE (IUN)
       RETURN
       END
C      ******************************************************
       SUBROUTINE RANDOM (NRAN,L,MEAN,STDEV)
C      ......................................................
C        RANDOM generates normally distributed random
C              process.
C      ......................................................

       REAL NRAN(L), MEAN

       NSEED=100000
       DO I=1,L
          SUM=0.0
          DO J=1,12
             U=RAN(NSEED)
             SUM=SUM+U
          END DO
          NRAN(I)=MEAN+(SUM-6)*STDEV
       END DO
       RETURN
       END
C      ******************************************************
```

APPENDIX C

PROGRAM FOR WAVELET PROCESSING

```
C     ***************************************************
C
C      This program is written for following processings:
C
C                  DOUBLE   WIENER   TRANSFORM
C
C                  SPIKING       DECONVOLUTION
C
C                  MINIMUM-PHASE    CONVERSION
C
C                  MATCHED OR BANDPASS  FILTER
C
C     .....................................................
C                   Written by:  Yuesheng Li
C
C                   Date:    August 10, 1987
C     .....................................................
      CHARACTER  ANS*1,INFILE*12,TITLE*50
      INTEGER    UN,UN1,UN2,UN3,UN4
      PARAMETER (UN=6,UN1=1,UN2=2,UN3=3,UN4=4,N=83,NN=1500,
     +           T=0.001,WN=0.001,AMP=1.0,LCX=128,LS=41,
     +           N1=500,N2=0)
      DIMENSION  A(N,N),B(NN),BB(NN),BBB(NN),C(N),D(N),
     +           DD(N),F(N),G(2*N),H(N),S(N),SS(LCX),
     +           SSS(-LS:LS)
      COMPLEX    CX(LCX)
C
C --- Select processing type
C
      WRITE (UN,*)'This program can do the following jobs:'
      WRITE (UN,*)'1. Double Wiener transform.'
      WRITE (UN,*)'2. Spiking deconvolution.'
      WRITE (UN,*)'3. Minimum phase conversion.'
      WRITE (UN,*)'4. Matched or bandpass filtering.'
      WRITE (UN,*)'Please enter your choice:'
100   READ  (UN,*) IANS
      IF (IANS.EQ.2) GO TO 200
      IF (IANS.EQ.3) GO TO 300
      IF (IANS.EQ.4) GO TO 400
      IF (IANS.NE.1) THEN
```

```
         WRITE (UN,*)'Your choice is wrong.'
         WRITE (UN,*)'Please enter your choice again:'
         GO TO 100
      END IF
C     ------------------------------------------------
C              Double Wiener Transform
C     ------------------------------------------------
      WRITE (UN,*)'Is the input a seismic trace? (Y/N)'
      READ (UN,'(A)') ANS
      IF (ANS.EQ.'Y'.OR.ANS.EQ.'y') THEN
         NP=NN
         CALL INPUT (T,B,NP,UN,UN1)
      ELSE
         NP=N
         CALL INPUT (T,BB,N,UN,UN1)
         CALL MOVE (0,BB,N,B,N)
      END IF
C
C --- Calculate Wiener operator
C
      CALL CORREL (B,NP,B,NP,C,0,N-1)
      CALL FACTOR (C,C,N,1.0)
      CALL MATRIX (A,D,C,N,WN)
      CALL GAUSS (N,A,D)
C
C --- 2nd time Wiener transform
C
      CALL CORREL (D,N,D,N,C,0,N-1)
      CALL FACTOR (C,C,N,1.0)
      CALL MATRIX (A,D,C,N,WN)
      CALL GAUSS (N,A,D)
C
C --- Output of result
C
      CALL FACTOR (D,D,N,AMP)
      CALL OUTPUT (D,N,T,UN2)
      IF (ANS.EQ.'Y'.OR.ANS.EQ.'y') GO TO 1000
      CALL FACTOR (BB,BB,N,AMP)
      CALL OUTPUT (BB,N,T,UN1)
      CALL NORMAL (BB,BB,N)
      CALL NORMAL (D,D,N)
      CALL SIMIL  (BB,D,N,COEF)
      WRITE (UN,*)'The correlation coefficient is', COEF
      GO TO 1000
C     ------------------------------------------------
C              Spiking Deconvolution
C     ------------------------------------------------
200   WRITE (UN,*)'SP deconvolution to seismic data? (Y/N)'
```

```
      READ (UN,'(A)') ANS
      IF (ANS.EQ.'Y'.OR.ANS.EQ.'y') THEN
         NP=NN
         WRITE (UN,*)'How many traces?'
         READ (UN,*) NTRACE
         IF (NTRACE.NE.1) THEN
            WRITE (UN,*)'Enter filename of seismic data:'
            READ (UN,'(A)') INFILE
            OPEN (UNIT=UN1,FILE=INFILE,STATUS='OLD')
         END IF
      ELSE
         NP=N
         NTRACE=1
      END IF
      PN=NP
      BELTA=ALOG(10.0)/PN
C
C --- Input of seismic data
C
      DO I=1,NTRACE
         IF (NTRACE.NE.1) THEN
            READ (UN1,*) (B(J),J=1,NP)
            READ (UN1,'(A)') TITLE
            N0=N1+N2*(I-1)
         ELSE
            CALL INPUT (T,B,NP,UN,UN1)
            N0=0
         END IF
C
C --- Amplitude compensation
C
         DO KK=1,NP
            B(KK)=B(KK)*EXP(BELTA*KK)
         END DO
C
C --- Calculate Wiener operator
C
         CALL MOVE (-N0,B,NP,BBB,NP)
         CALL CORREL (BBB,NP,BBB,NP,C,0,N-1)
         CALL FACTOR (C,C,N,1.0)
         CALL MATRIX (A,D,C,N,WN)
         CALL GAUSS (N,A,D)
C
C --- Deconvolution of data & output
C
         CALL CONVO (B,0,NP-1,D,0,N-1,BB,NP-1)
         CALL FACTOR (BB,BB,NP,AMP)
         IF (NTRACE.NE.1) THEN
```

```
            WRITE (UN2,*) (BB(J),J=1,NP)
            WRITE (UN2,'(A)') TITLE
         ELSE
            CALL OUTPUT (BB,NP,T,UN2)
         END IF
      END DO
      GO TO 1000
C     ------------------------------------------------
C           Minimum Phase Conversion of Seismic Data
C     ------------------------------------------------
300   WRITE (UN,*)'The source wavelet will be read.'
      CALL INPUT (T,S,N,UN,UN1)
C
C --- Calculate phase compensator
C
      CALL CORREL (S,N,S,N,C,0,N-1)
      CALL FACTOR (C,C,N,1.0)
      CALL MATRIX (A,D,C,N,WN)
      CALL GAUSS (N,A,D)
      CALL CONVO (S,0,N-1,D,0,N-1,G,2*N-2)
C
C --- Input of seismic data
C
      WRITE (UN,*)'The seismic data will be read in.'
      WRITE (UN,*)'How many traces?'
      READ (UN,*) NTRACE
      IF (NTRACE.NE.1) THEN
         WRITE (UN,*)'Enter filename of seismic data:'
         READ (UN,'(A)') INFILE
         OPEN (UNIT=UN1,FILE=INFILE,STATUS='OLD')
      END IF
      DO I=1,NTRACE
         IF (NTRACE.NE.1) THEN
            READ (UN1,*) (B(J),J=1,NN)
            READ (UN1,'(A)') TITLE
         ELSE
            CALL INPUT (T,B,NN,UN,UN1)
         END IF
C
C --- Phase compensation & output
C
         CALL CORREL (B,NN,G,2*N-1,BB,0,NN-1)
         CALL FACTOR (BB,BB,NN,AMP)
         IF (NTRACE.NE.1) THEN
            WRITE (UN2,*) (BB(J),J=1,NN)
            WRITE (UN2,'(A)') TITLE
         ELSE
            CALL OUTPUT (BB,NN,T,UN2)
```

```
         END IF
      END DO
      GO TO 1000
C     ------------------------------------------------
C              Matched or Bandpass Filtering
C     ------------------------------------------------
400   WRITE (UN,*)'Matched filtering? (Y/N)'
      READ (UN,'(A)') ANS
      IF (ANS.EQ.'Y'.OR.ANS.EQ.'y') THEN
C
C --- Calculate matched filter
C
         WRITE (UN,*)'The source wavelet will be read.'
         CALL INPUT (T,S,N,UN,UN1)
         CALL PREFFT (S,N,CX,LCX)
         CALL FFT (CX,LCX,1.0)
         CALL SPECTR (CX,LCX,SS,1)
         CALL PREFFT (SS,LCX,CX,LCX)
         CALL FFT (CX,LCX,-1.0)
         SSS(0)=REAL (CX(1))
         DO I=1,LS
            SSS(I)=REAL (CX(I+1))
            SSS(-I)=SSS(I)
         END DO
      ELSE
C
C --- Calculate bandpass filter
C
         WRITE (UN,*)'Enter the filter passband:'
         READ (UN,*) F1,F2,F3,F4
         CALL BPASS (SSS,LS,F1,F2,F3,F4,T)
      END IF
C
C --- Output of impulse response of filter
C
      CALL FACTOR (SSS(-LS),SSS(-LS),2*LS+1,1.0)
      DO I=-LS,LS
         TT=T*I
         WRITE (UN2,*)TT,SSS(I)
      END DO
C
C --- Input of seismic data
C
      WRITE (UN,*)'The seismic data will be read.'
      WRITE (UN,*)'How many traces?'
      READ (UN,*) NTRACE
      IF (NTRACE.NE.1) THEN
         WRITE (UN,*)'Enter filename of seismic data:'
```

```
         READ (UN,'(A)') INFILE
         OPEN (UNIT=UN1,FILE=INFILE,STATUS='OLD')
      END IF
      DO I=1,NTRACE
         IF (NTRACE.NE.1) THEN
            READ (UN1,*) (B(J),J=1,NN)
            READ (UN1,'(a)') TITLE
         ELSE
            CALL INPUT (T,B,NN,UN,UN1)
         END IF
C
C --- Filter seismic data trace by trace
C
         CALL CONVO (B,0,NN-1,SSS,-LS,LS,BB,NN-1)
C
C --- Output of filtered data
C
         CALL FACTOR (BB,BB,NN,AMP)
         IF (NTRACE.NE.1) THEN
            WRITE (UN3,*) (BB(J),J=1,NN)
            WRITE (UN3,'(A)') TITLE
         ELSE
            CALL OUTPUT (BB,NN,T,UN3)
         END IF
      END DO
C
1000  STOP  ' Finish all calculation.'
      END
C     ******************************************************
C
C              Subroutine INPUT reads data from a
C
C                   data file on the disk.
C     ......................................................
C       T      is the sampling interval.
C       N      is the total points.
C       B(N)   contains the input data.
C       IUN    is the unit number of the terminal.
C       IUN1   is the unit nunber for opening a data file.
C     ......................................................

      SUBROUTINE    INPUT (T,B,N,IUN,IUN1)

      CHARACTER   INFILE*12
      DIMENSION   B(N)

      CALL ZERO (B,N)
      WRITE (IUN,10)
```

```
10    FORMAT(1X,'Enter the input filename :')
      READ  (IUN,20) INFILE
20    FORMAT(A12)
      OPEN (UNIT=IUN1,FILE=INFILE,ERR=99,STATUS='OLD')
      DO I=1,N
         READ  (IUN1,*,END=30) TIME,B(I)
      END DO
30    CLOSE (IUN1)
      RETURN
99    STOP 'The input file is not on the disk.'
      END
C     ******************************************************
C
C       Subroutine CORREL correlates x(t) with y(t) and
C
C                  puts the results in z(t).
C       ...................................................
C       LX     is the length of array X.
C       X(LX)  contains data x(t).
C       LY     is the length of array Y.
C       Y(LY)  contains data y(t).
C       LZ1 & LZ2 are lower & upper limits of Z.
C       Z(I)   contains the result of correlation.
C     ...................................................

      SUBROUTINE  CORREL (X,LX,Y,LY,Z,LZ1,LZ2)

      DIMENSION   X(LX), Y(LY), Z(LZ1:LZ2)

      CALL ZERO (Z(LZ1),(LZ2-LZ1+1))
      I1=MAX0 ((-LY+1),LZ1)
      I2=MIN0 ((LX-1),LZ2)
      DO 20 I=I1,I2
         SUM=0.0
         DO 10 J=1,LX
            IF ((J-I).LT.1.OR.(J-I).GT.LY) GO TO 10
            SUM=SUM+X(J)*Y(J-I)
10       CONTINUE
         Z(I)=SUM
20    CONTINUE
      RETURN
      END
C     ******************************************************
C
C       Subroutine NORMAL normalizes input data by energy.
C
C       The total energy of output data equals to 1.
C     ...................................................
```

```
C       L      is the length of arrays A & B.
C       A(L)   contains the input data.
C       B(L)   contains the output data .
C     ..................................................

      SUBROUTINE  NORMAL (A,B,L)

      DIMENSION   A(L), B(L)

      SUM=0.0
      DO I=1,L
         SUM=SUM+A(I)*A(I)
      END DO
      SUM=SQRT(SUM)
      DO I=1,L
         B(I)=A(I)/SUM
      END DO
      RETURN
      END
C     ******************************************************
C
C       Subroutine MATRIX creats a Toplitz matrix and a
C
C       N X 1  matrix by using a set of given data.
C     ..................................................
C       N      is the size of arrays.
C       X(N)   contains a set of given data.
C       B(N)   is a N X 1 matrix.
C       A(N,N) is a N X N Toplitz matrix.
C       WN     is white noise.
C     ..................................................

      SUBROUTINE  MATRIX (A,B,X,N,WN)

      DIMENSION  A(N,N), B(N), X(N)

C
C --- Add white noise
C
      X(1)=(1+WN)*X(1)
C
C --- Create a Toplitz matrix A(N,N)
C
      DO I=1,N
         K=1
         DO J=I,N
            A(I,J)=X(K)
            K=K+1
```

```
         END DO
      END DO
      DO I=2,N
         K=2
         DO J=I-1,1,-1
            A(I,J)=X(K)
            K=K+1
         END DO
      END DO
C
C --- Create matrix B(N)=[1,0,0,......0]
C
      B(1)=1
      DO I=2,N
         B(I)=0.0
      END DO
      RETURN
      END
C     ******************************************************
C
C       Subroutine GAUSS uses the Gauss method to solve
C
C              a set of simultaneous equations.
C     ......................................................
C       N      is the number of equations.
C       A(N,N) contains the coefficients of equations.
C       B(N)   contains the constants of equations
C              at the begining and contains the solution
C              of equations when return.
C     ......................................................

      SUBROUTINE  GAUSS (N,A,B)

      DIMENSION   A(N,N), B(N)

      NM1=N-1
      DO 30 K=1,NM1
         C=0.0
C
C --- Find the main element in the Kth column and
C --- put it in C, put its row number in I0
C
         DO 2 I=K,N
            IF (ABS(A(I,K)).LE.ABS(C)) GO TO 2
            C=A(I,K)
            I0=I
2        CONTINUE
         EPS=1E-10
```

```
         IF (ABS(C).LT.EPS) THEN
            STOP 'Main element is too small.'
         END IF
C
C --- The main element is on the Kth row,
C --- no need to change positions
C
         IF  (I0.EQ.K) GO TO 6
C
C --- Change the positions of the Kth row with
C --- the row where the main element is
C
         DO 4 J=K,N
            TT=A(K,J)
            A(K,J)=A(I0,J)
            A(I0,J)=TT
4        CONTINUE
         TT=B(K)
         B(K)=B(I0)
         B(I0)=TT
C
C --- The Kth time to calculate the matrix.
C
6        KP1=K+1
         C=1.0/C
         B(K)=B(K)*C
         DO 20 J=KP1,N
            A(K,J)=A(K,J)*C
            DO 10 I=KP1,N
               A(I,J)=A(I,J)-A(I,K)*A(K,J)
10          CONTINUE
            B(J)=B(J)-A(J,K)*B(K)
20       CONTINUE
30    CONTINUE
C
C --- Finish matrix calculation.
C --- Solve the equations and get the final answer.
C
      B(N)=B(N)/A(N,N)
      DO 50 K=1,NM1
         I=N-K
         C=0.0
         IP1=I+1
         DO 40 J=IP1,N
            C=C+A(I,J)*B(J)
40       CONTINUE
         B(I)=B(I)-C
50    CONTINUE
```

```
      RETURN
      END
C     ******************************************************
C
C       Subroutine CONVO convolves x(t) with y(t) and
C
C                  puts the result in z(t)
C     ......................................................
C       X(N) & Y(N) contain input functions x(t) & y(t).
C       Z(I)  contains the result z(t)=x(t)*y(t).
C       LX2-LX1  is the size of matrix X.
C       LY2-LY1  is the size of matrix Y.
C     ......................................................

      SUBROUTINE  CONVO (X,LX1,LX2,Y,LY1,LY2,Z,LZ)

      DIMENSION  X(LX1:LX2), Y(LY1:LY2), Z(0:LZ)

C
C --- LX1 & LY1 must be less than or equal to 0.
C
      CALL ZERO (Z(0),LZ+1)
      DO 20 I=0,LZ
         IF (I.GT.(LX2+LY2)) GO TO 30
         SUM=0.0
         DO 10 J=LX1,I-LY1
            IF ((I-J).LT.LY1.OR.(I-J).GT.LY2) GO TO 10
            IF (J.GT.LX2) GO TO 10
            SUM=SUM+X(J)*Y(I-J)
10       CONTINUE
         Z(I)=SUM
20    CONTINUE
30    RETURN
      END
C     ******************************************************
C
C       Subroutine ZERO puts value 0.0 in every element
C
C                  of array A.
C
C     ......................................................

      SUBROUTINE ZERO (A,LA)

      DIMENSION  A(LA)

      DO 10 I=1,LA
         A(LA)=0.0
```

```
10    CONTINUE
      RETURN
      END
C     ******************************************************
C
C       Subroutine FACTOR normalizes the amplitude of a
C
C             function to a given value: AMP.
C     ......................................................
C       AMP     is the desired amplitude.
C       L       is the length of X & Y.
C       X(L)    contains the input data.
C       Y(L)    contains the output date.
C     ......................................................

      SUBROUTINE  FACTOR (X,Y,L,AMP)

      REAL  MAX
      DIMENSION   X(L), Y(L)

      MAX=ABS(X(1))
      DO 10 I=2,L
         IF (ABS(X(I)).LE.MAX) GO TO 10
         MAX=ABS (X(I))
10    CONTINUE
      DO 20 I=1,L
         Y(I)=AMP*X(I)/MAX
20    CONTINUE
      RETURN
      END
C     ******************************************************
C
C       Subroutine OUTPUT writes result into a data file.
C     ......................................................
C       T    is the sampling interval.
C       IUN  is the unit number for output.
C       LX   is the total points of output data.
C       X(LX) contains the output data.
C     ......................................................

      SUBROUTINE   OUTPUT  (X,LX,T,IUN)

      DIMENSION   X(LX)

      DO I=1,LX
         TIME=T*(I-1)
         WRITE (IUN,*) TIME,X(I)
      END DO
```

```
      CLOSE (IUN)
      RETURN
      END
C     ********************************************************
C
C       Subroutine SIMIL computes the CC value between
C
C                     x(t) and y(t).
C     ........................................................
C       X & Y contain functions x(t) & y(t).
C       COEF  is the correlation coefficient.
C     ........................................................

      SUBROUTINE SIMIL (X,Y,L,COEF)

      DIMENSION  X(L),Y(L)

      SUM1=0.0
      SUM2=0.0
      SUM3=0.0
      DO 10 I=1,L
         SUM1=SUM1+X(I)*Y(I)
         SUM2=SUM2+X(I)*X(I)
         SUM3=SUM3+Y(I)*Y(I)
10    CONTINUE
      AABB=SQRT (SUM2*SUM3)
      COEF=SUM1/AABB
      RETURN
      END
C     ********************************************************
C
C       Subroutine MAXMUM finds the maximum value from
C
C               X(LX) and puts the result in MAX.
C     ........................................................

      SUBROUTINE MAXMUM (X,LX,MAX)

      REAL MAX
      DIMENSION  X(LX)

      MAX=X(1)
      DO 10 I=2,LX
         IF (X(I).LE.MAX) GO TO 10
         MAX=X(I)
10    CONTINUE
      RETURN
      END
```

```
C     ****************************************************
C
C       Subroutine MOVE moves data from X(LX) to Y(LY)
C
C               with relative shift ID.
C     ....................................................

      SUBROUTINE MOVE (ID,X,LX,Y,LY)

      DIMENSION  X(LX),Y(LY)

      CALL ZERO (Y,LY)
      LMIN=MIN0 ((LX+ID),LY)
      IF (ID.GE.0) THEN
         DO I=ID+1,LMIN
            Y(I)=X(I-ID)
         END DO
      ELSE
         DO I=1,LMIN
            Y(I)=X(I-ID)
         END DO
      END IF
      RETURN
      END
C     ****************************************************
C
C       Subroutine PREFFT moves data from real array A
C
C               to complex array CB.
C     ....................................................

      SUBROUTINE PREFFT (A,LA,CB,LB)

      COMPLEX  CB(LB)
      DIMENSION A(LA)

      DO I=1,LB
         CB(I)=(0.0,0.0)
      END DO
      LMIN=MIN0 (LA,LB)
      DO I=1,LMIN
         CB(I)=CMPLX(A(I),0.0)
      END DO
      RETURN
      END
C     ****************************************************
C
C       Subroutine FFT computes the Fourier transform
```

```
C
C                using fast algorithm.
C     ..................................................
C       SIGNI (1.0 or -1.0) is the indicator,
C       -1.0 indicates inverse transform.
C     ..................................................

      SUBROUTINE FFT (CX,LX,SIGNI)

      COMPLEX  CX(LX),CARG,CEXP,CW,CTEMP

      J=1
      SC=SQRT(1./LX)
      DO 30 I=1,LX
         IF (I.GT.J) GO TO 10
         CTEMP=CX(J)*SC
         CX(J)=CX(I)*SC
         CX(I)=CTEMP
10       M=LX/2
20       IF (J.LE.M) GO TO 25
         J=J-M
         M=M/2
         IF (M.GE.1) GO TO 20
25       J=J+M
30    CONTINUE
      L=1
40    ISTEP=2*L
      DO 60 M=1,L
         CARG=(0.,1.)*(3.14159265*SIGNI*(M-1))/L
         CW=CEXP(CARG)
         DO 50 I=M,LX,ISTEP
            CTEMP=CW*CX(I+L)
            CX(I+L)=CX(I)-CTEMP
            CX(I)=CX(I)+CTEMP
50       CONTINUE
60    CONTINUE
      L=ISTEP
      IF (L.LT.LX) GO TO 40
      RETURN
      END
C     ***************************************************
C
C       Subroutine SPECTR computes the amplitude or
C
C       phase spectra from a set of complex data.
C     ..................................................
C       IFLAG is the indicator: 1 for amplitude
C     ..................................................
```

```
      SUBROUTINE SPECTR (CB,L,A,IFLAG)

      COMPLEX  CB(L)
      DIMENSION A(L)

      IF (IFLAG.EQ.1) THEN
         DO 10 I=1,L
            A(I)=CABS(CB(I))
10       CONTINUE
      ELSE
         PI=3.14159265
         DO 120 I=1,L
            AR=REAL(CB(I))
            AI=AIMAG(CB(I))
            IF (AI) 20,30,40
20          IF (AR) 50,60,70
30          IF (AR) 80,90,70
40          IF (AR) 100,110,70
50          A(I)=ATAN(AI/AR)-PI
            GO TO 120
60          A(I)=-PI/2.0
            GO TO 120
70          A(I)=ATAN(AI/AR)
            GO TO 120
80          A(I)=-PI
            GO TO 120
90          A(I)=0.0
            GO TO 120
100         A(I)=ATAN(AI/AR)+PI
            GO TO 120
110         A(I)=PI/2.0
120      CONTINUE
      END IF
      RETURN
      END
C     ***************************************************
C
C       Subroutine BPASS computes the impulse response
C
C            of a zero-phase bandpass filter.
C     ...................................................
C       H(LH) contains the impulse response.
C       F1,F2,F3,F4 define the filter passband.
C       TSR is the sampling rate in time domain.
C     ...................................................

      SUBROUTINE BPASS (H,LH,F1,F2,F3,F4,TSR)
```

```
      DIMENSION  H(-LH:LH)

      PI=3.1415926
      A=(F2-F1)/2.0
      B=F1+F2
      C=1.0/(A*A)
      AA=(F4-F3)/2.0
      BB=F3+F4
      CC=1.0/(AA*AA)
      H(0)=BB-B
      DO I=1,LH
         TIME=TSR*I
         PIT=PI*TIME
         H1=C*SIN(B*PIT)/PIT
         H2=(SIN(A*PIT)/PIT)**2
         H3=CC*SIN(BB*PIT)/PIT
         H4=(SIN(AA*PIT)/PIT)**2
         H(I)=H3*H4-H1*H2
         J=-I
         H(J)=H(I)
      END DO
      RETURN
      END
C     ************************************************************
```