T-2221

STATISTICAL EXAMINATION OF NOISE ON PHASE ESTIMATION IN SPIKING DECONVOLUTION

by

Neal E. Fausset

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 Dec 14, 1979

Signed: ______________________
Neal E. Fausset

Golden, Colorado

Date Dec 14, 1979

Approved: ______________________
Thesis Advisor

Golden, Colorado

Date 12/14/, 1979

Approved: ______________________
Head of Department

ABSTRACT

Deconvolution in the presence of random noise must introduce phase errors damaging to a final seismic section in terms of resolution and amplitude and waveform variations unrelated to geologic impedence changes. This study addresses the quantification of these distortions.

A band-pass minimum phase wavelet, sampled at a two millisecond rate, is convolved with an ensemble of random noise realizations to color the noise spectra with the amplitude response of the wavelet. In practice, this procedure is equivalent to passing ambient background noise through a recording filter.

The filtered noise traces are autocorrelated, scaled in power relative to the autocorrelation of the wavelet such that a prescribed signal to noise ratio is adhered to, and added to the autocorrelation of the wavelet. The Wiener-Levinson spiking deconvolution algorithm is then applied. The resulting deconvolution filters designed from wavelets corrupted by noise with the same average spectrum are analyzed in a statistical sense with regard to phase spectrum error for signal-to-noise ratios of 2.0, 1.0, 0.5, and 0.25. The analysis indicates that for signal and noise of the same average spectral shape, deconvolution is a robust process, and a reasonable estimate of the wavelet can be made, even for signal-to-noise ratios as low as 0.25. The phase estimation error is nearly linear with

frequency and results in a lag and slight asymmetry in the deconvolved pulse.

To provide a more realistic example demonstrating the effect of inelastic attenuation, the wavelet spectrum is tapered such that the amplitude is down 12 dB at one-half the Nyquist frequency. In another model, notches are introduced into the spectrum by convolution with a ghost operator. Deconvolution of these altered signals in the presence of noise shaped with the original wavelet spectrum indicates the phase may be substantially distorted. The phase error for these models is also inversely dependent upon signal-to-noise ratio.

The observed phase error is significant, especially in light of modern stratigraphic interpretation techniques, where a zero-phase condition is often assumed in the final processed section.

TABLE OF CONTENTS

LIST OF FIGURES

ACKNOWLEDGEMENTS

I wish to express my gratitude to the petroleum industry sponsors of the Integrated Geophysics Project at the Colorado School of Mines, for without their financial support, this study would have been impossible.

In particular, I am indebted to my advisor, Dr. William A. Schneider, for his invaluable assistance from the inception through the completion of this study.

I should also like to acknowledge the assistance of my committee members. Dr. J. E. White provided many valuable suggestions. He also devoted long hours to the Integrated Geophysics Project, from which many students, including myself, benefited. Dr. David Butler contributed greatly through his expertise in computer programming.

I would also like to thank my mother and father for their moral and financial support.

Mary Lou Bowers and Elizabeth Williams typed the manuscript; their assistance is greatly appreciated.

Finally, I would like to thank Jan, whose patience and understanding helped me to survive this ordeal.

INTRODUCTION

Current seismic technique is tending more and more toward exploration for subtle stratigraphic hydrocarbon traps, and detailed interpretation and comparison with borehole data is essential. High quality deconvolution results become, therefore, a requirement. An effective deconvolution method should yield broad band, zero-phase wavelets with smooth amplitude spectra.

Schoenberger (1974) and Berkhout (1977) discuss two arguments favoring zero phase wavelets. First, they point out the effective length of a wavelet w_t determines the resolution of the seismic trace and this length will reach a minimum if the phase property of the wavelet is zero. Second, they explain that for any given amplitude spectrum, the maximum amplitude will be produced by the zero phase wavelet at t = 0. Consequently, transformation of minimum or mixed phase reflections into their zero phase correspondents will maximize the detectability of their arrival times.

Deconvolution with a conventional inverse filter in the presence of seismic noise will generally not produce sufficiently high-quality results. Filtering by this method will lead to a delay in the wavelet, with a resulting deviation in phase from the optimum zero lag case.

This result may be qualitatively demonstrated in the frequency domain by use of the Hilbert transform technique

(Sheriff, 1973). Assume the following reflection seismic model:

$$s(t) = w(t) * r(t) + n(t), \qquad (1)$$

where s(t) is the recorded signal, w(t) is the wavelet, r(t) is the primary reflectivity series, and n(t) is a colored realization of seismic noise. The discrete time parameter is t. Transformation to the frequency domain, computation of the respective power spectra, and rearrangement of the terms in equation (1) results in

$$S^2(f) = W^2(f) \bullet R^2(f)\left[1 + \frac{N^2(f)}{W^2(f)R^2(f)}\right]. \qquad (2)$$

Taking the natural logarithm of both sides yields

$$\ln S^2(f) = \ln W^2(f) + \ln R^2(f) + \ln\left[1 + \frac{N^2(f)}{W^2(f)R^2(f)}\right]. \qquad (3)$$

The Hilbert transform technique states

$$\phi(f) = \tfrac{1}{2}\ln P(f) * (1/\pi f), \qquad (4)$$

where $\phi(f)$ is the minimum phase spectrum of some wavelet, and P(f) is its power spectrum. Applying this concept to equation (3) in a qualitative sense leads to

$$\phi_S(f) = \phi_W(f) + \phi_R(f) + \phi_N(f). \qquad (5)$$

Here $\phi_W(f)$ is the minimum phase spectrum of the wavelet, and $\phi_R(f)$ and $\phi_N(f)$ are phase errors associated with the reflectivity sequence and the colored seismic noise. If

R(f) has a white amplitude spectrum, the phase error due to the reflectivity function can and will be assumed to be zero. Then the only component of phase error is imparted by the addition of noise n(t). As a result of $N^2(f)$ being a random variable, the phase error is also random.

Only two authors appear to have treated the subject of deconvolution in the presence of noise to any great extent.

Berkhout (1977) compared Wiener filtering with wavelet deconvolution. He examined conventional deconvolution in the presence of high frequency noise, low frequency noise, and white noise, and concluded that if the noise power spectrum is not known, estimation of the influence of noise on filter coefficients is difficult. His examples indicated that one-sided least-squares inverse filtering tends to time delay the filtered wavelet. In fact, in the presence of low frequency noise, an apparent polarity reversal occurred. Berkhout also investigated two-sided least squares inverse filtering and found it to be not only superior to the one-sided case, but remarkably similar to wavelet deconvolution. Finally, he studied wavelet deconvolution and found the filter phase spectrum to be correct, provided a good estimate of the input wavelet was known. He conceded, however, that the necessity of knowing the wavelet is a serious drawback to wavelet deconvolution, one not encountered with one-sided Wiener filtering.

Berkhout does not discuss bandlimited noise in his models. Bandlimited noise is considered here to be a more realistic model because in practice, random noise passes through a field recording filter and is colored with the filter's spectral characteristics.

DeVoogd (1978) considered a finite realization of a discrete random noise process to be a one-sided energy signal whose phase property could be described by means of a center position. He found for processes described by the convolution of a wavelet with a white noise series, the center position was independent of the phase property of the wavelet. Based on these results, he concluded the homomorphic technique of wavelet estimation through cepstrum stacking gave questionable outcomes and the superposition of a realization of random noise on a minimum phase wavelet in general gave a mixed phase result.

Neither of the authors test statistical variability in their models. It therefore becomes apparent that phase errors for common geophysical models should be assigned numerical values based on a statistical evaluation. Three seismic noise models are examined; one with an unattenuated wavelet, the second with an attenuated wavelet, and the third with a wavelet convolved with a ghost operator. In addition, the industry-wide practice of prewhitening a wavelet before deconvolution is evaluated with regard to phase errors.

METHOD OF INVESTIGATION

The Wavelet

The choice of design parameters for wavelet generation is not particularly critical, except that the amplitude response of the wavelet should resemble as closely as possible the recording filter response of a typical seismic data acquisition system. Consequently, the wavelet is designed in the frequency domain to have a trapezoidal amplitude spectrum and zero phase spectrum. A frequency bandpass defined by a 36 dB/octave low cut filter and an anti-alias high cut filter is not uncommon throughout the industry, and for a sample rate of 2 milliseconds, a 5-10-125-188 bandpass, shown in Figure 1, meets these design constraints quite well. Transformation of this amplitude response into the time domain yields a zero phase Ormsby wavelet (Figure 2).

Computer program WAVLET.FOR performs this transformation digitally using the equation (Schneider, 1979),

$$w(n) = \frac{1}{\pi^2 n^2 \Delta t} \left[\frac{K\sin\{\pi(f_4 + f_3)n\Delta t\} \cdot \sin\{\pi(f_4 - f_3)n\Delta t\}}{(f_4 - f_3)} + \frac{(1 - K)\sin\{\pi(f_3 + f_2)n\Delta t\} \cdot \sin\{\pi(f_3 - f_2)n\Delta t\}}{(f_3 - f_2)} - \frac{\sin\{\pi(f_2 + f_1)n\Delta t\} \cdot \sin\{\pi(f_2 - f_1)n\Delta t\}}{(f_2 - f_1)} \right], \tag{6}$$

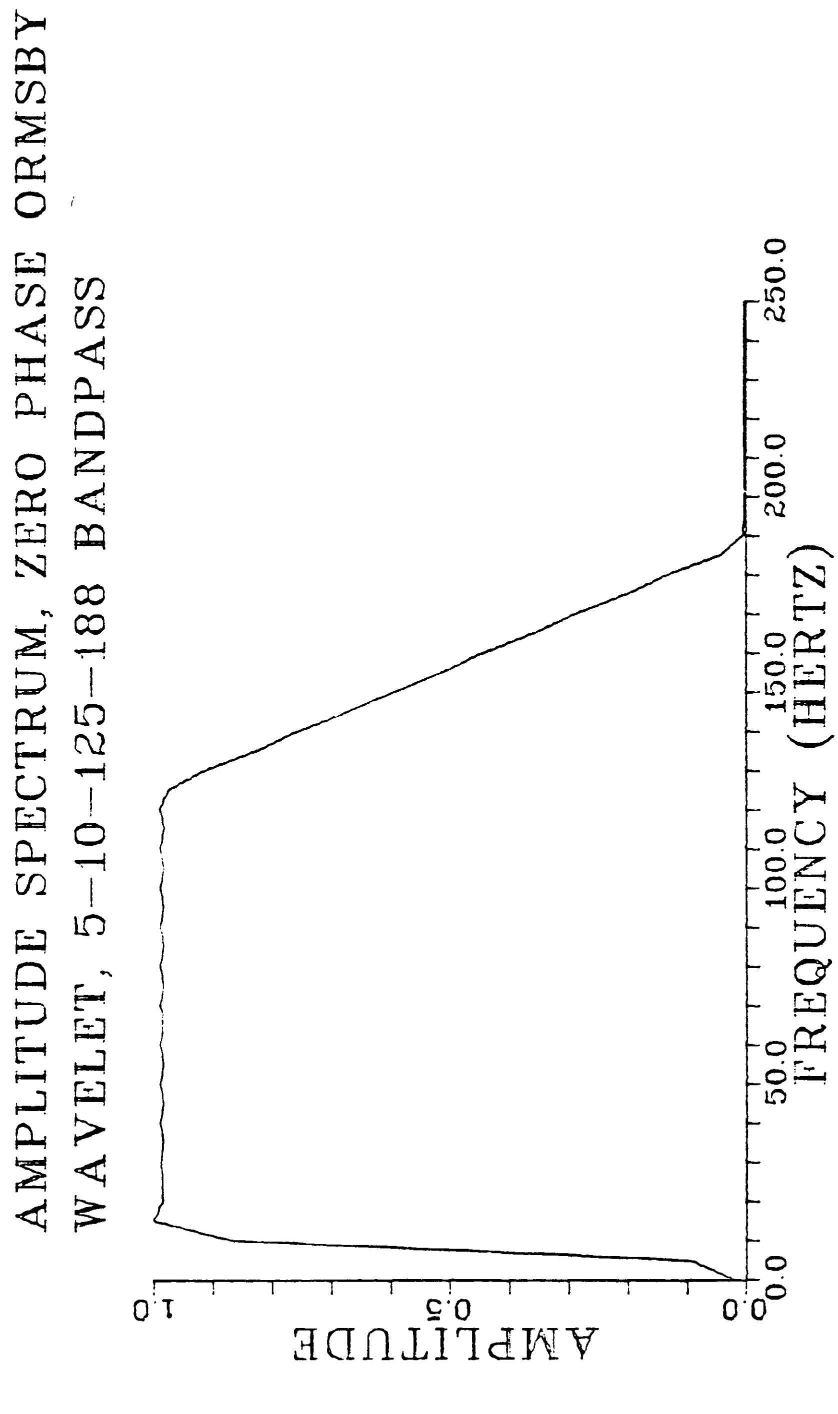

AMPLITUDE SPECTRUM, ZERO PHASE ORMSBY
WAVELET, 5–10–125–188 BANDPASS
AMPLITUDE
0.0
0.5
1.0
FREQUENCY (HERTZ)
0.0
50.0
100.0
150.0
200.0
250.0

Figure 1

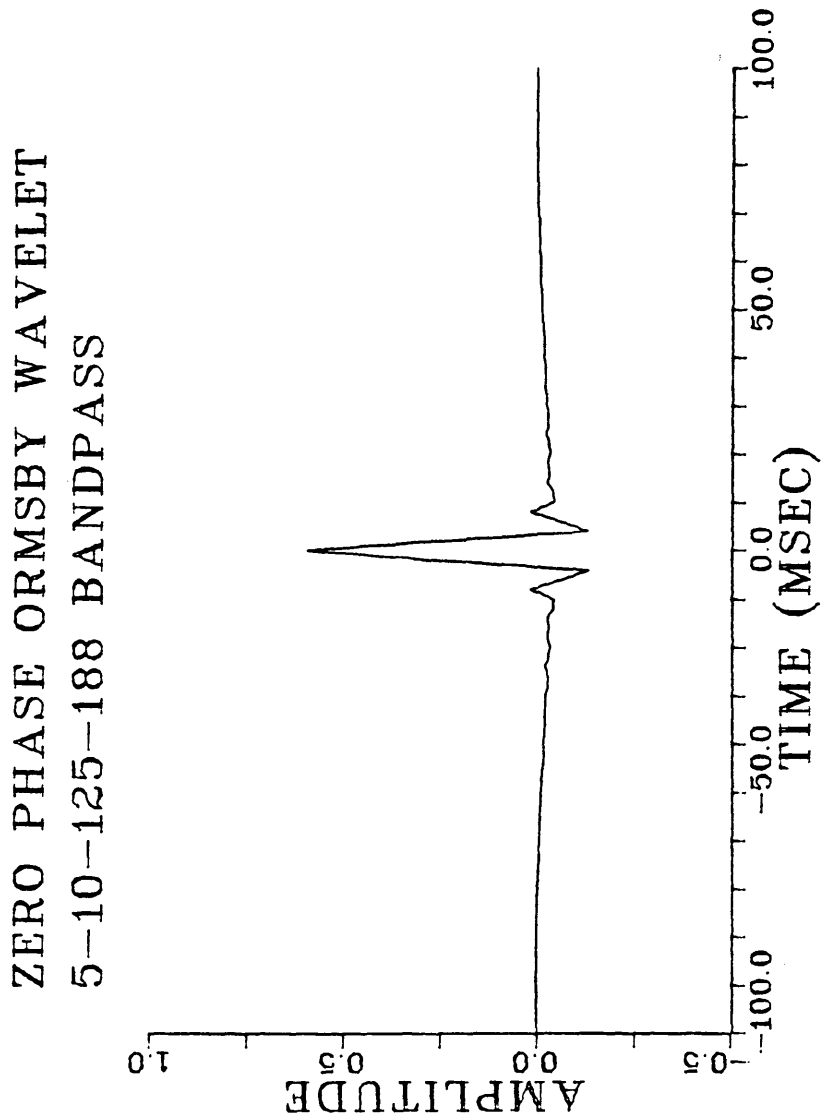

ZERO PHASE ORMSBY WAVELET
5—10—125—188 BANDPASS
AMPLITUDE
1.0
0.5
0.0
-0.5
TIME (MSEC)
-100.0
-50.0
0.0
50.0
100.0

Figure 2

where w(n) are the wavelet coefficients for the trapezoidal response (n being an integer), Δt is the sample rate in milliseconds, and f_1, f_2, f_3, and f_4 are the four corner frequencies defining a trapezoid which represents relative passband amplitudes of 0%, 100%, 100%, and 0%, respectively. K, the relative passband amplitude at f_3, may vary between zero and one, and is incorporated into the program to modify the spectrum, allowing the generation of an attenuated wavelet.

An immediate problem is recognized; that is, evaluation of equation (6) for w(0) results in a zero over zero indeterminant form. This situation is corrected by inserting appropriate values into the McLaurin Series expansion of the sine function, namely

$$\sin\{\pi(f_a \pm f_b)n\Delta t = \pi(f_a \pm f_b)n\Delta t$$

$$- \frac{\{\pi(f_a \pm f_b)n\Delta t\}^3}{6} + \text{H. O. T.}$$

Substitution of this expression into equation (6), cancellation of higher order forms, and simplification yields

$$w(0) = \Delta t\{K(f_4 - f_2) + (f_3 - f_1)\}. \qquad (7)$$

A wavelet length of 101 samples is arbitrarily chosen, representing a sequence in time from -100 milliseconds to +100 milliseconds. This operator is long enough to allow the amplitude of the wavelet to sufficiently diminish, yet is short enough that generation and processing is not unwieldy in terms of computer time.

Implementation of program WAVLET.FOR is further discussed in Appendix A.

The Noise Realization

Program RNOISE.FOR (Appendix B) generates a finite realization of discrete random noise with an essentially white spectrum. These numbers vary between zero and one, and can be regarded as statistically independent.

By convolving a wavelet w_t with a white series n_t, seismic noise may be simulated and represented by the linear relation (Sheriff, 1973),

$$h_s = \Delta t \sum_t w_t n_{s-t}, \qquad (8)$$

where Δt is the sampling increment and s and t are discrete time variables. In further discussions, Δt is assumed to be unity, since no loss in generality will result.

In practice, this is equivalent to coloring a random noise signal with the response of a field recording filter.

DeVoogd (1978) demonstrated that seismic noise realizations should be formed by convolving the wavelet with the white series and subsequently truncating, rather than first truncating the white series and then convolving, in consequence of the latter technique possibly leading to erroneous results. This theoretical result is consistent with seismic practice in that ambient background noise is infinitely long with respect to a seismic pulse.

Accordingly, the 101 point wavelet was convolved with a 1200 sample noise train, using program CONVLV.FOR (Appendix C).

Wavelet Processing In the Absence of Noise

Signal processing is initiated by computing the right-hand side of the autocorrelation function of the wavelet. By employing program AUTCOR.FOR, discussed in Appendix D, the autocorrelation $r_0, r_1, \ldots r_k$ of the signal $x_0, x_1, \ldots x_m$ may be obtained.

AUTCOR.FOR computes autocorrelation coefficients according to the formula (Sheriff, 1973),

$$r_s = \sum_t x_t x_{t-s}. \qquad (9)$$

These coefficients are then input to a digital Wiener filter, whose properties are briefly discussed in the following text.

Digital filtering can be described by the convolutional model

$$y_s = \sum_t x_t f_{s-t}, \qquad (10)$$

where x_t is the input, f_t is the filter, and y_s is the output. Assume f_t is a prediction filter or operator with prediction distance τ. Then the output y_s will be an estimate of the input x_t at some time $t + \tau$ in the future.

That is, (Peacock and Treitel, 1969)

$$y_s = \bar{x}_{t+\tau} \quad , \qquad (11)$$

where $\bar{x}_t + _\tau$ is some estimate of $x_t + _\tau$. It is possible to define an error series as the difference between a true value and some estimated or predicted value, therefore

$$\varepsilon_{t+\tau} = x_{t+\tau} - \bar{x}_{t+\tau} \; , \qquad (12)$$

ε_t being an output series representing the nonpredictable part of x_t.

Combining equations (10) and (11) and substituting the equivalent expression for $\bar{x}_t + \tau$ into equation (12) yields

$$\varepsilon_{t+\tau} = x_{t+\tau} - \sum_t x_t f'_{s-t}. \qquad (13)$$

The z-transform of equation (13) may be written as

$$z^{-\tau}E(z) = z^{-\tau}X(z) - X(z)F(z). \qquad (14)$$

Multiplication throughout equation (14) by z^τ results in

$$E(z) = X(z)\{1 - z^\tau F(z)\}. \qquad (15)$$

The function $1-z^{\tau}F(z)$ is the difference between a zero lag unit spike and the prediction filter F(z) delayed by the prediction distance τ, and is the z-transform of the prediction error operator. Thus, the prediction error filter removes the predictable portion of a seismic trace, and the output of such a filter, to a good approximation, represents an estimate of the reflection coefficient series of a layered subsurface (Peacock and Treitel, 1969).

Note that if the prediction operator is given by the time series

$$f_t = f_o, f_1, \ldots f_{n-1},$$

then the corresponding prediction error operator is given by

$$a_t = 1, 0, 0, \ldots 0, -f_o, -f_1, \ldots -f_{n-1},$$

where the series a_t contains $\tau - 1$ zeros.

The actual design of the prediction operator must now be dealt with. Three signals are involved in the general model of a least squares, or Wiener filter, namely the input x_t, the desired output d_t, and the actual output y_t. If the energy difference between the designed output and the actual output, i.e., the expression

$$I = \sum_t (d_t - y_t)^2 \qquad (16)$$

is minimized, it results in the so-called normal equations, represented in matrix form by

$$\begin{bmatrix} r_o & r_i & \cdots & r_n \\ r_i & r_o & \cdots & r_{n-i} \\ & & \vdots & \\ r_n & r_{n-i} & \cdots & r_o \end{bmatrix} \begin{bmatrix} f_o \\ f_i \\ \vdots \\ f_n \end{bmatrix} = \begin{bmatrix} g_o \\ g_i \\ \vdots \\ g_n \end{bmatrix} . \qquad (17)$$

Here r_t is the autocorrelation of the input, g_t is the crosscorrelation between the input and the desired output, and f_t is the Wiener filter described by Treitel and Robinson (1966).

A recursive algorithm to solve system (17) was first described by Levinson (1947), and further expanded upon by Wiggins and Robinson (1965).

The prediction filter is that filter which operates on an input signal at some time t and estimates the signal amplitude at some future time $t + \tau$; therefore, it is reasonable to define the desired output of the filter as being a time-advanced version of the input x_t. The prediction filter can now be expressed in terms of a particular Wiener filter.

The autocorrelation coefficients of the input and the positive lag coefficients of the crosscorrelation between the input and the desired output must be known to solve equation (17). The autocorrelation coefficients are given by equation (9) while the crosscorrelation coefficients are given by Sheriff (1973), as

$$g_s = \sum_t d_t x_{t-s} . \qquad (18)$$

The desired output is a time-advanced version of the input, that is

$$d_t = x_{t+\tau} ,$$

thus by substitution equation (18) becomes

$$g_s = \sum_t x_{t+\tau} x_{t-s} = \sum x_t x_{t-(s+\tau)} = r_{s+\tau} . \qquad (19)$$

Hence by definition, the crosscorrelation between the desired output and the input is equal to the autocorrelation of the input for time lags τ. The normal equations (17) become (Peacock and Treitel, 1969)

$$\begin{bmatrix} r_o & r_1 & \cdots & r_n \\ r_1 & r_o & \cdots & r_{n-1} \\ & & \vdots & \\ r_n & r_{n-1} & \cdots & r_o \end{bmatrix} \begin{bmatrix} f_o \\ f_1 \\ \vdots \\ f_n \end{bmatrix} = \begin{bmatrix} r_\tau \\ r_{\tau+1} \\ \vdots \\ r_{\tau+n} \end{bmatrix} . \qquad (20)$$

Appendix E discusses LSQINF.FOR, a computer program employing a Levinson recursion scheme to solve systems (17) or (20).

Peacock and Treitel (1969) demonstrate the least-squares inverse filter which transforms an unknown signal to an impulse at zero delay is equivalent to the prediction error operator for which τ is unity. Treitel and Robinson (1966) point out that for the minimum delay case, best filter performance is achieved at the zero lag value. By setting $\tau = 1$ in equation (20), the matrix relation which gives the predictive deconvolution filter for the minimum phase case is obtained, namely

$$\begin{bmatrix} r_o & r_1 & \cdots & r_n \\ r_1 & r_o & \cdots & r_{n-1} \\ & & \vdots & \\ r_n & r_{n-1} & \cdots & r_o \end{bmatrix} \begin{bmatrix} f_o \\ f_1 \\ \vdots \\ f_n \end{bmatrix} = \begin{bmatrix} r_1 \\ r_2 \\ \vdots \\ r_{n+1} \end{bmatrix} . \qquad (21)$$

Because spiking deconvolution represents an effort to compress or "spike" a signal at the zero lag value, predictive deconvolution for the minimum delay case and spiking deconvolution are equivalent operations.

It was pointed out earlier that the zero lag least-squares inverse filter for $\tau = 1$ is equal to the prediction error operator, whose coefficients are

$$a_t = 1, -f_o, -f_1, \ldots -f_n$$

This implies that if the coefficients r_t in the Töplitz matrix of equation (21) represent the autocorrelation of a zero phase wavelet w_t, then

$$m_t = \frac{1}{a_t} \qquad (22)$$

where m_t is the minimum phase estimate of w_t. Both m_t and a_t are polynomials in z.

The polynomial division represented in equation (22) is performed by program DIVIDE.FOR, whose documentation is discussed at length in Appendix F.

This processing flow was utilized on the zero-phase wavelet shown in Figure 2, and the resulting minimum-phase wavelet is depicted in Figure 3. The amplitude and phase spectral characteristics of this minimum phase wavelet are displayed in Figures 4a and 4b.

Convolution of the minimum phase wavelet m_t with its inverse a_t yields the zero lag spike shown in Figure 5. Considering this spike hypothetically represents one reflection coefficient of a series portraying a sedimentary sequence, convolution with the wavelet in Figure 2 produces the desired processing result, i.e., a seismic section with broad-band zero phase wavelets.

The Least Squares Inverse Filter in The Presence of Noise

The deconvolution analysis must now be extended to the case where the signal is a wavelet w_t embedded in a finite realization of seismic noise n_t with a known autocorrelation function q_t, that is,

$$x_t = w_t + n_t .$$

The sum of the squared errors between the desired output and the actual output plus the weighted power of the filtered noise is minimized (Treitel and Robinson, 1966). Presented in symbols, this quantity is

$$I = \sum_t \{(d_t - y_t)^2 + pA\langle n_t^2\rangle\}, \qquad (23)$$

where p is an arbitrary parameter assigning the relative weighting between the input wavelet and noise functions,

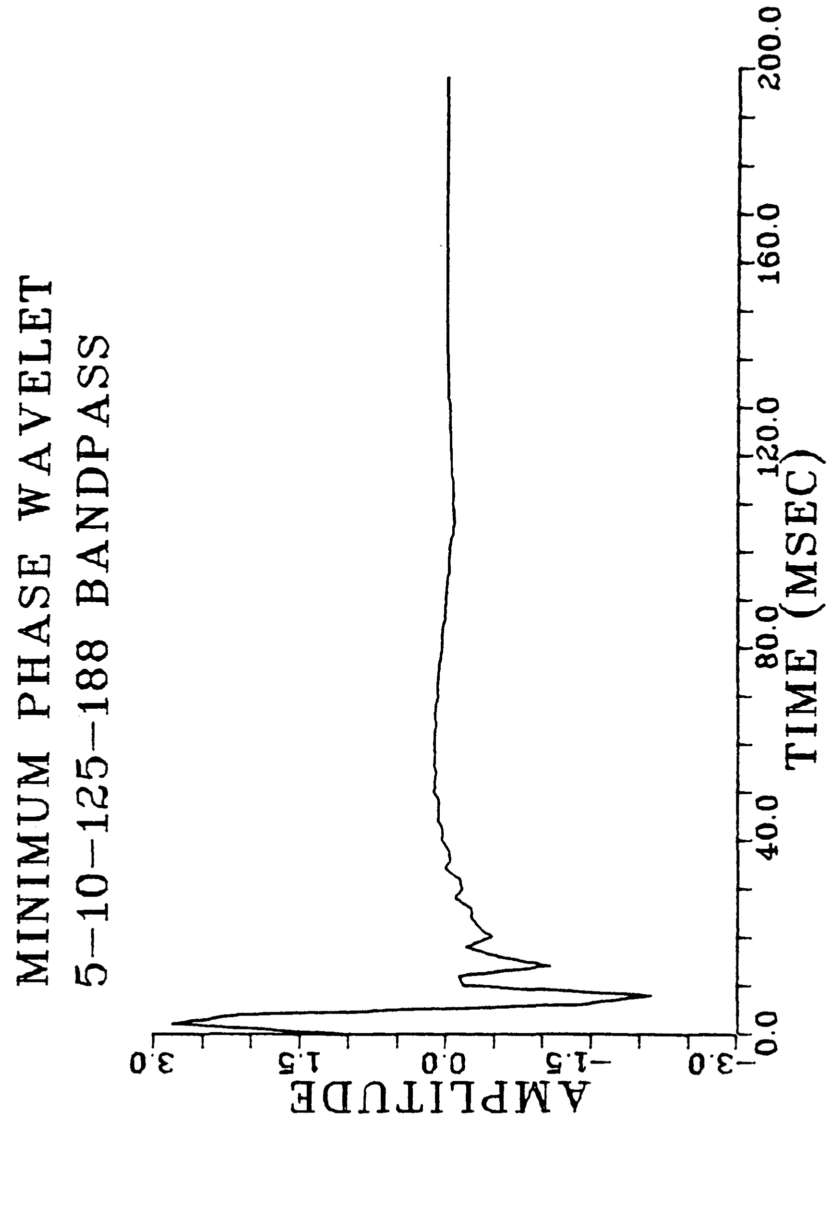
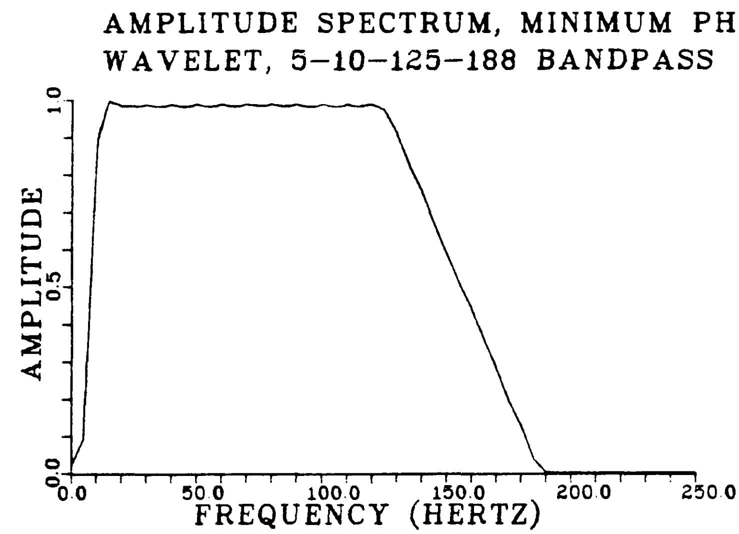
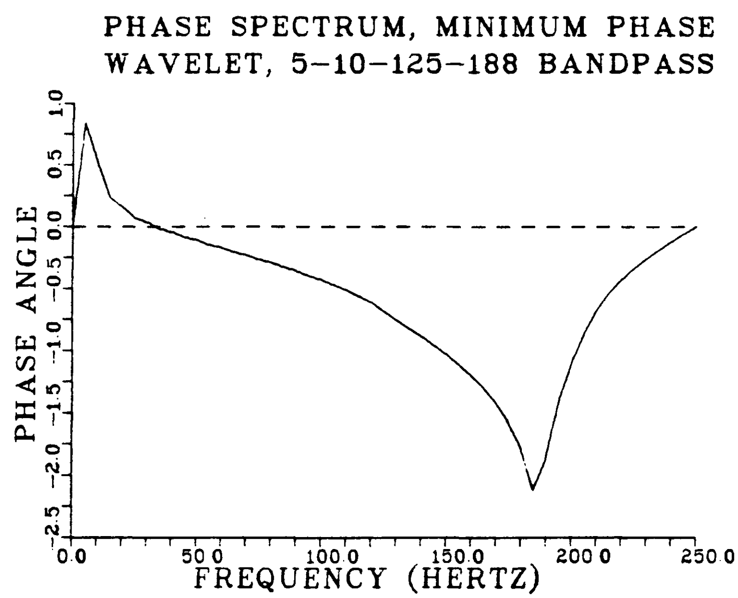

MINIMUM PHASE WAVELET
5—10—125—188 BANDPASS
TIME (MSEC)
AMPLITUDE
0.0
40.0
80.0
120.0
160.0
200.0
3.0
1.5
0.0
-1.5
-3.0

Figure 3


AMPLITUDE SPECTRUM, MINIMUM PHASE
WAVELET, 5-10-125-188 BANDPASS
AMPLITUDE
0.0
0.5
1.0
0.0
50.0
100.0
150.0
200.0
250.0
FREQUENCY (HERTZ)

(a)


PHASE SPECTRUM, MINIMUM PHASE
WAVELET, 5-10-125-188 BANDPASS
PHASE ANGLE
-2.5
-2.0
-1.5
-1.0
-0.5
0.0
0.5
1.0
0.0
50.0
100.0
150.0
200.0
250.0
FREQUENCY (HERTZ)

(b)

Figure 4

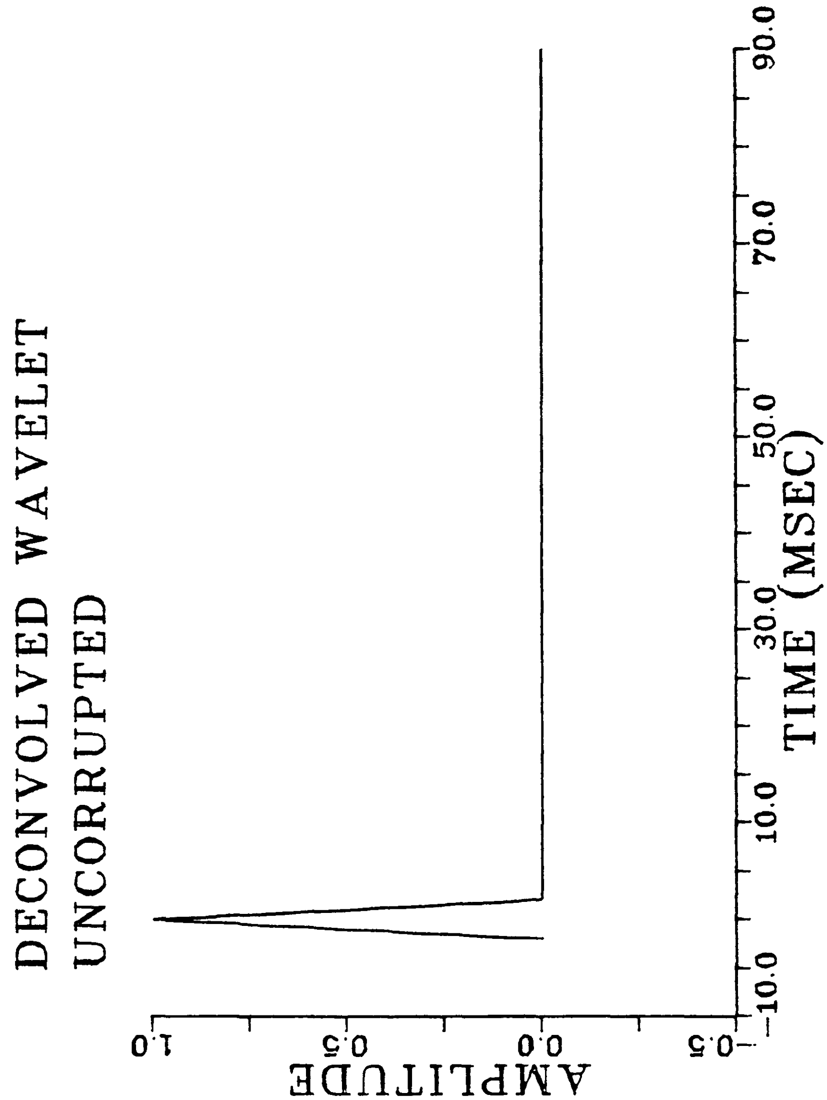

DECONVOLVED WAVELET
UNCORRUPTED
TIME (MSEC)
AMPLITUDE
-10.0
10.0
30.0
50.0
70.0
90.0
1.0
0.5
0.0
-0.5

Figure 5

and the notation A< > indicates the ensemble average. Simplifying equation (23), differentiating I with respect to the filter coefficients, and then setting the derivatives equal to zero, the set of simultaneous equations

$$g_s = \sum_t f_t (r_{s-t} + pq_{s-t}) \qquad (24)$$

is obtained.

Clearly, the replacement of coefficients r_n in equation (20) by coefficients $r_n + pq_n$ will correctly lead to the design of the Wiener filter in the presence of noise.

Attention is now focused on signal to noise ratio as a design criterion for the weighting parameter p.

By definition, the total power of a signal or a noise realization is equal to the zero lag value of its autocorrelation function, or

$$S^2 = r_o,$$

and

$$N^2 = q_o.$$

Therefore

$$\left(\frac{S}{N}\right)^2 = \frac{r_o}{q_o},$$

and rearranging terms,

$$q_o(\text{desired}) = \frac{r_o}{\left(\frac{S}{N}\right)^2 (\text{desired})}. \qquad (25)$$

The desired signal to noise ratio is merely a scaled version of the actual signal to noise ratio, that is,

$$q_o(\text{desired}) = pq_o(\text{actual}). \qquad (26)$$

Substitution of equation (26) into equation (25), followed by algebraic manipulation, yields

$$p = \frac{r_o}{q_o\left(\frac{S}{N}\right)^2}. \qquad (27)$$

Thus, in the presence of noise, the autocorrelation coefficient inputs to the matrix system (21) become

$$\left[r_n\right] = \left[r_n + \frac{r_o}{q_o\left(\frac{S}{N}\right)^2} q_n\right]. \qquad (28)$$

Program SCALE.FOR (Appendix G) performs this computation.

WAVELET MODEL

Although it might seem difficult to cite numerous examples where divergence, reflection, absorption and scattering do not play significant roles in the seismic method, a deconvolutional model with an unattenuated wavelet is discussed here to provide a "best possible result" to compare with more realistic models.

Autocorrelation functions of unattenuated wavelets with a bandpass of 5-10-125-188 Hertz are added to the autocorrelations of colored seismic noise and processed through the Wiener-Levinson spiking deconvolution algorithm to generate corrupted minimum phase equivalent wavelets. Twelve wavelets each are generated for signal-to-noise ratios of 2.0, 1.0, 0.5, and 0.25. Figures 6a through 6d show one such wavelet from each of the four suites. Note the slight decrease in amplitude of the main lobes of the wavelet and the increase in random reverberations later in the signal (20 to 200 msec) as the signal-to-noise ratio decreases from 2.0 to 0.25. However, also note these corrupted wavelets do not really differ drastically from their noise free minimum phase counterpart (Figure 3).

For a minimum phase wavelet of sufficient length, deconvolution by its inverse yields a zero lag spike of unit amplitude, shown in Figure 5. To determine the effect of deconvolution in the presence of noise on the wavelet only, the aforementioned suites of wavelets were deconvolved

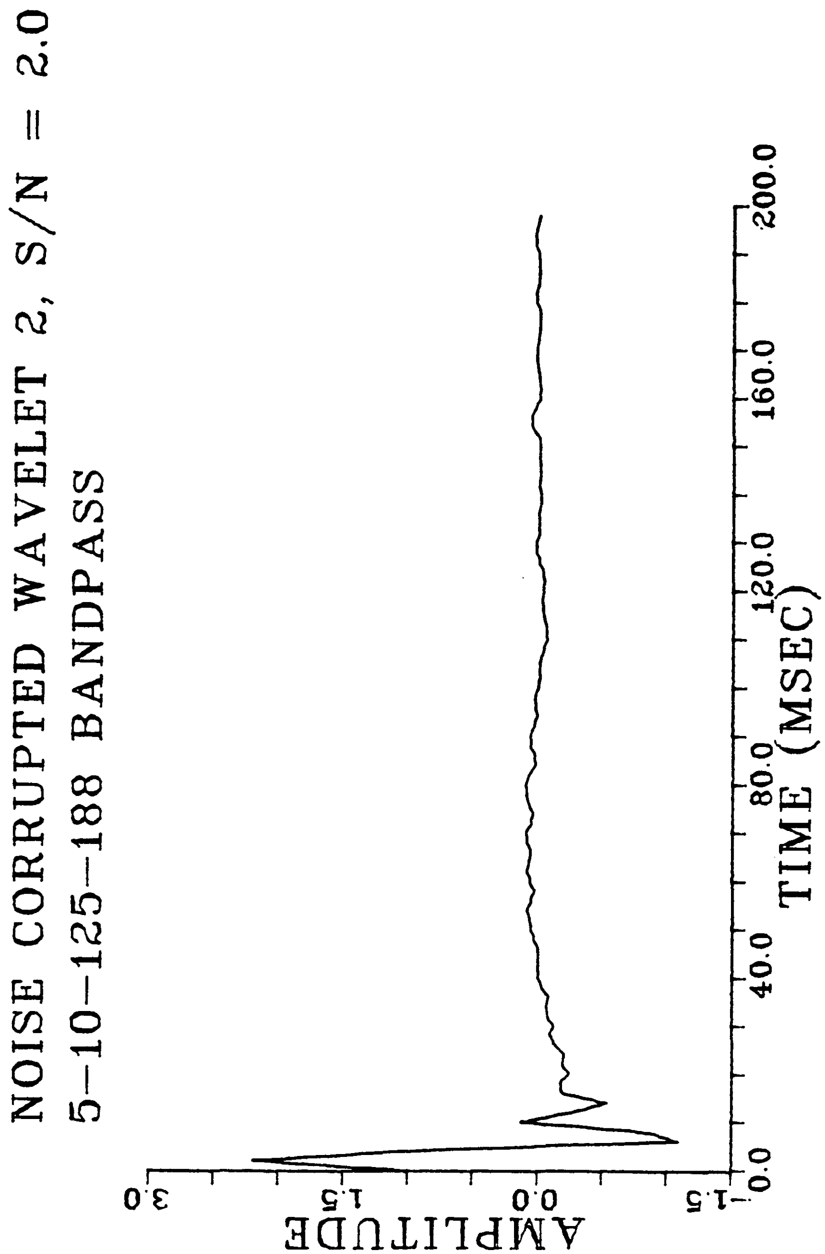

NOISE CORRUPTED WAVELET 2, S/N = 2.0
5—10—125—188 BANDPASS
TIME (MSEC)
AMPLITUDE
0.0
40.0
80.0
120.0
160.0
200.0
-1.5
0.0
1.5
3.0

Figure 6a

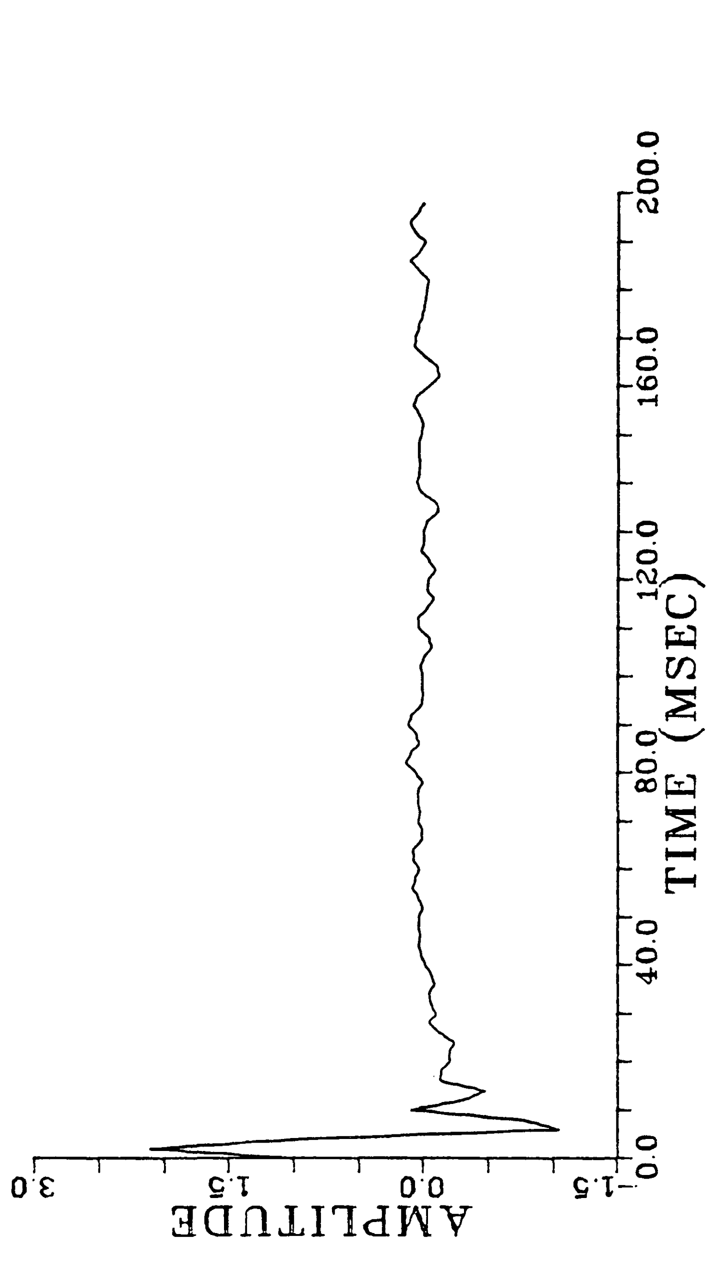

NOISE CORRUPTED WAVELET 1, S/N = 1.0
5-10-125-188 BANDPASS
AMPLITUDE
3.0
1.5
0.0
-1.5
TIME (MSEC)
0.0
40.0
80.0
120.0
160.0
200.0

Figure 6b

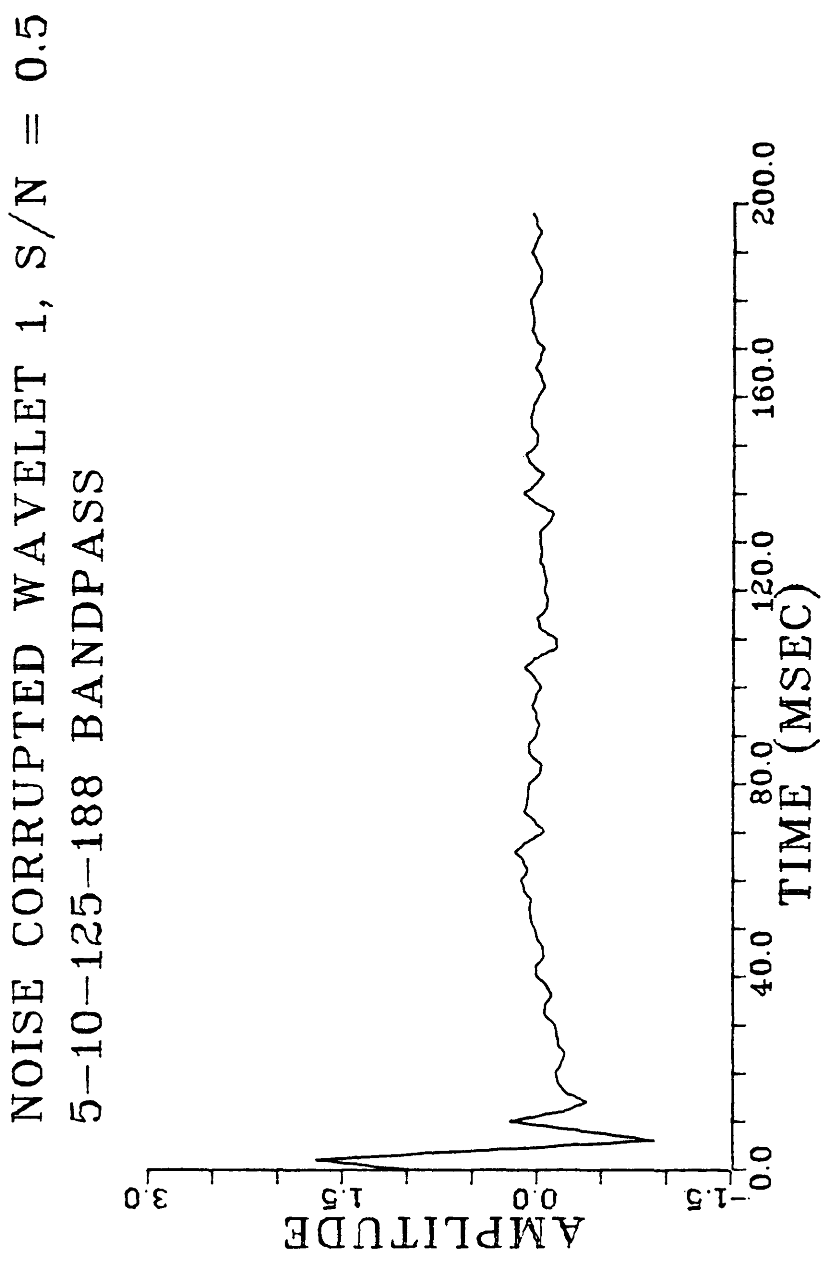

NOISE CORRUPTED WAVELET 1, S/N = 0.5
5–10–125–188 BANDPASS
AMPLITUDE
3.0
1.5
0.0
-1.5
TIME (MSEC)
0.0
40.0
80.0
120.0
160.0
200.0

Figure 6c

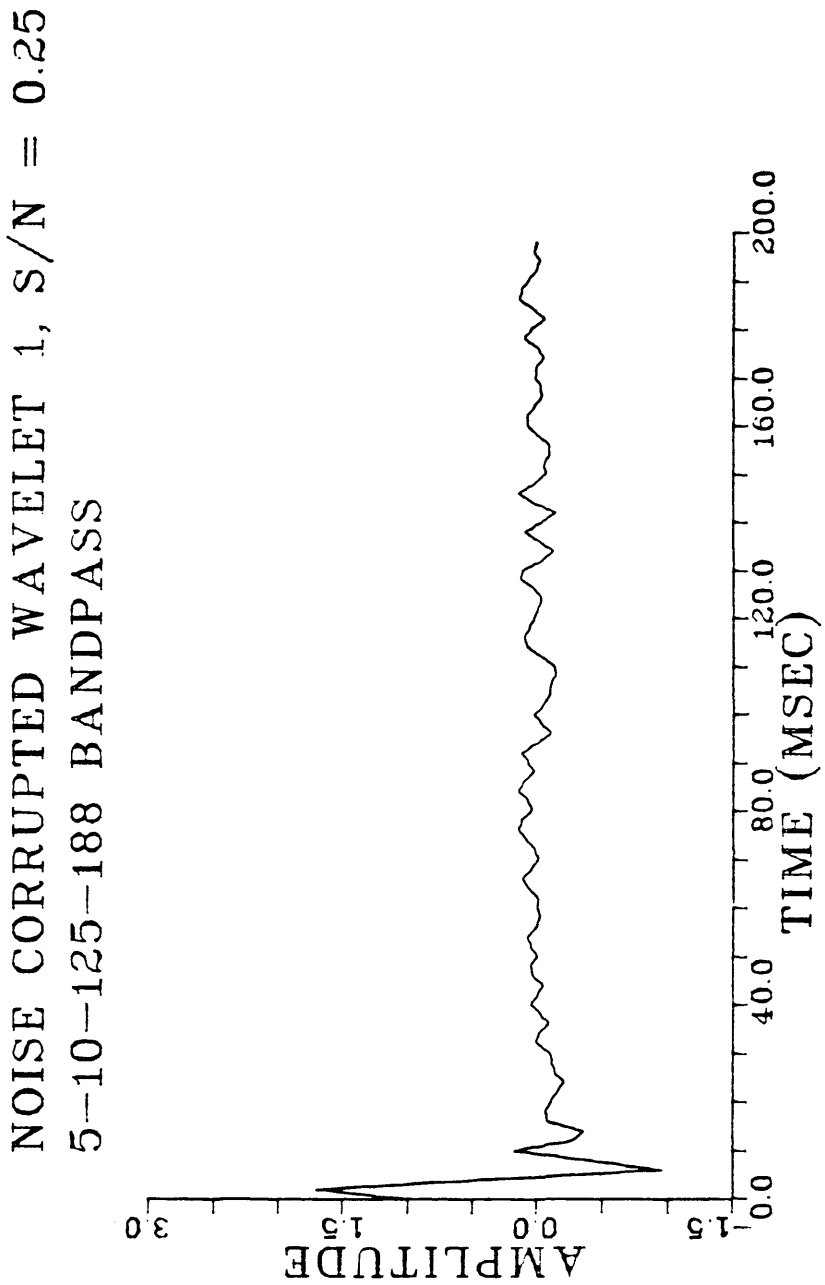

NOISE CORRUPTED WAVELET 1, S/N = 0.25
5—10—125—188 BANDPASS
AMPLITUDE
-1.5
0.0
1.5
3.0
TIME (MSEC)
0.0
40.0
80.0
120.0
160.0
200.0

Figure 6d

by their noise free minimum phase inverse. Deconvolution of a noisy wavelet by its noise free inverse might seem illogical, when common practice is to deconvolve a noise free wavelet by a corrupted inverse. The former technique was used merely for convenience and ease of computation. Clearly, the choice of technique is immaterial; convolution is commutative, and one method will reproduce the result of the other method exactly.

Results from the deconvolution of the wavelets in Figure 6 are depicted in Figures 7a through 7d. The resulting "spikes" vary in amplitude, become broader, and are delayed in time. A significant amount of choppiness follows the positive amplitude spike, and as the signal-to-noise ratio worsens, a rather noticeable negative spike develops. This is best demonstrated at a time of six milliseconds in Figures 7c and 7d.

The filter performance parameter P is obtained by subtracting the least squared error between the desired and actual outputs from one. At P = 1.0, perfect filter performance is achieved. Filter performance parameters are calculated for the deconvolution filters just described, and a slight decrease in performance is observed as the signal-to-noise ratio worsens.

The deconvolved wavelets depicted in Figure 7 can each be considered one coefficient of a reflectivity series. A common final processing step throughout the industry is bandpass filtering; accordingly the deconvolved wavelets

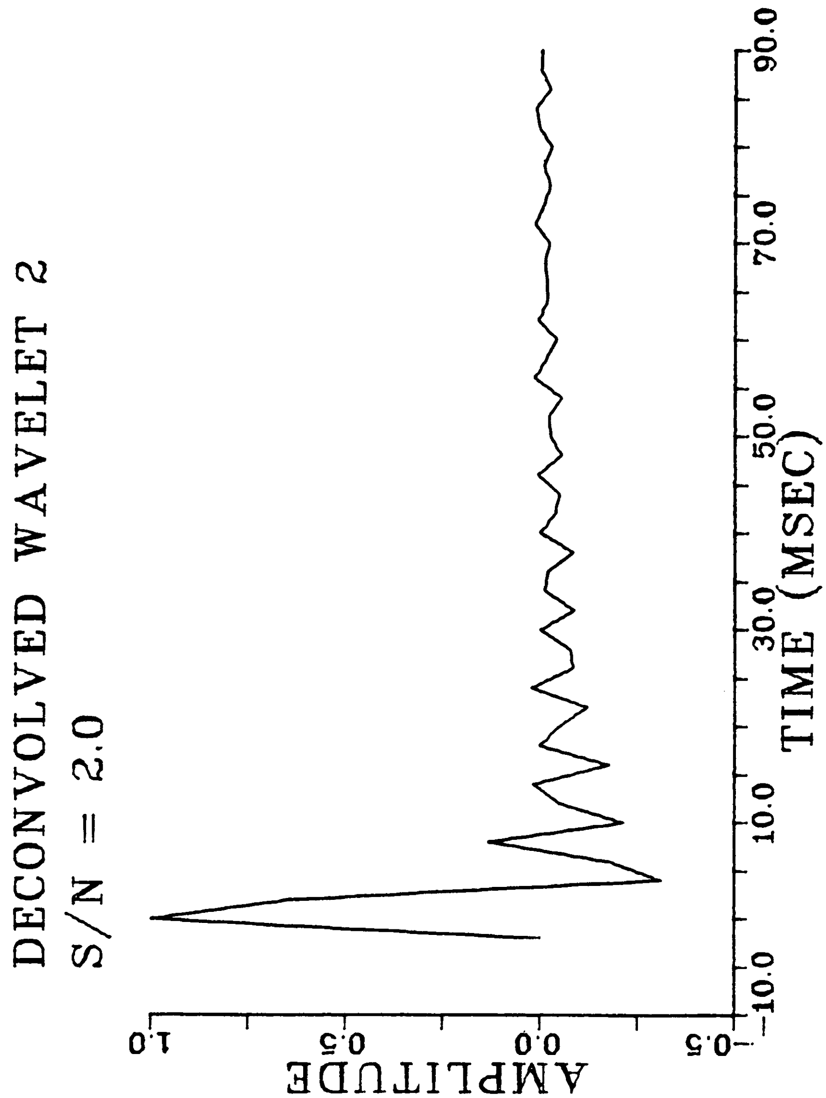

DECONVOLVED WAVELET 2
S/N = 2.0
AMPLITUDE
TIME (MSEC)
1.0
0.5
0.0
-0.5
-10.0
10.0
30.0
50.0
70.0
90.0

Figure 7a

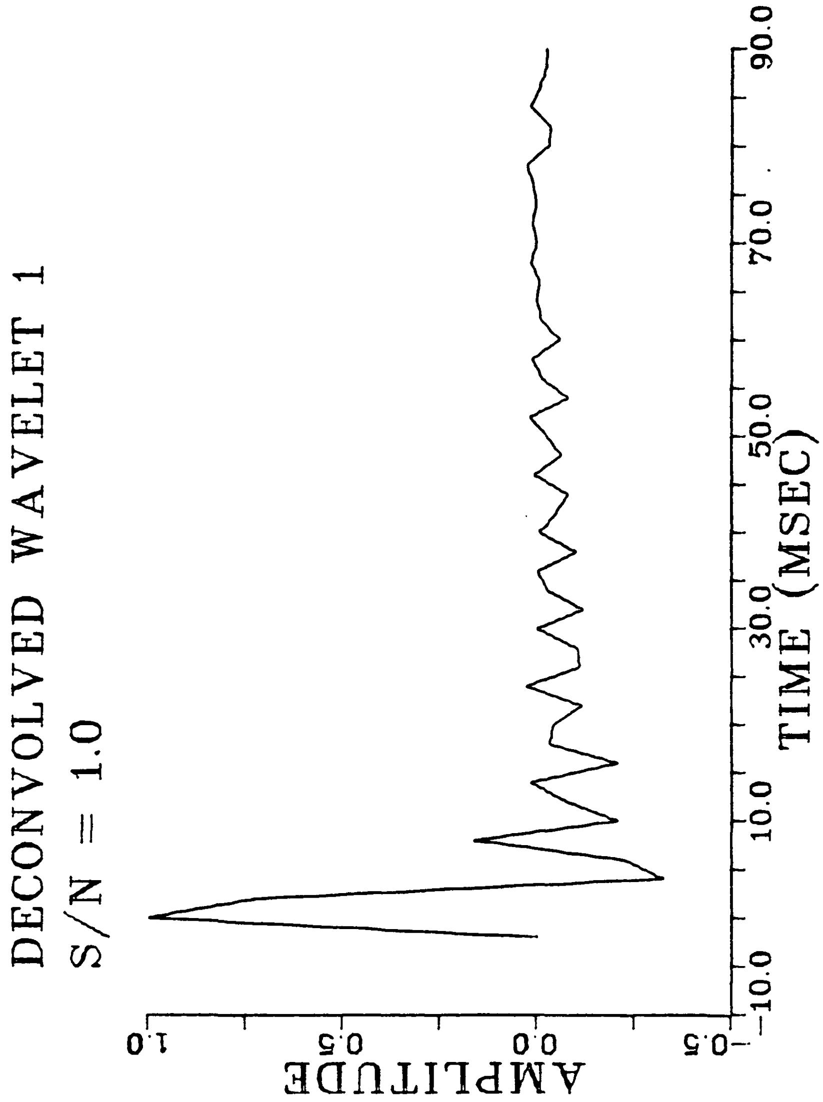

DECONVOLVED WAVELET 1
S/N = 1.0
AMPLITUDE
1.0
0.5
0.0
-0.5
TIME (MSEC)
-10.0
10.0
30.0
50.0
70.0
90.0

Figure 7b

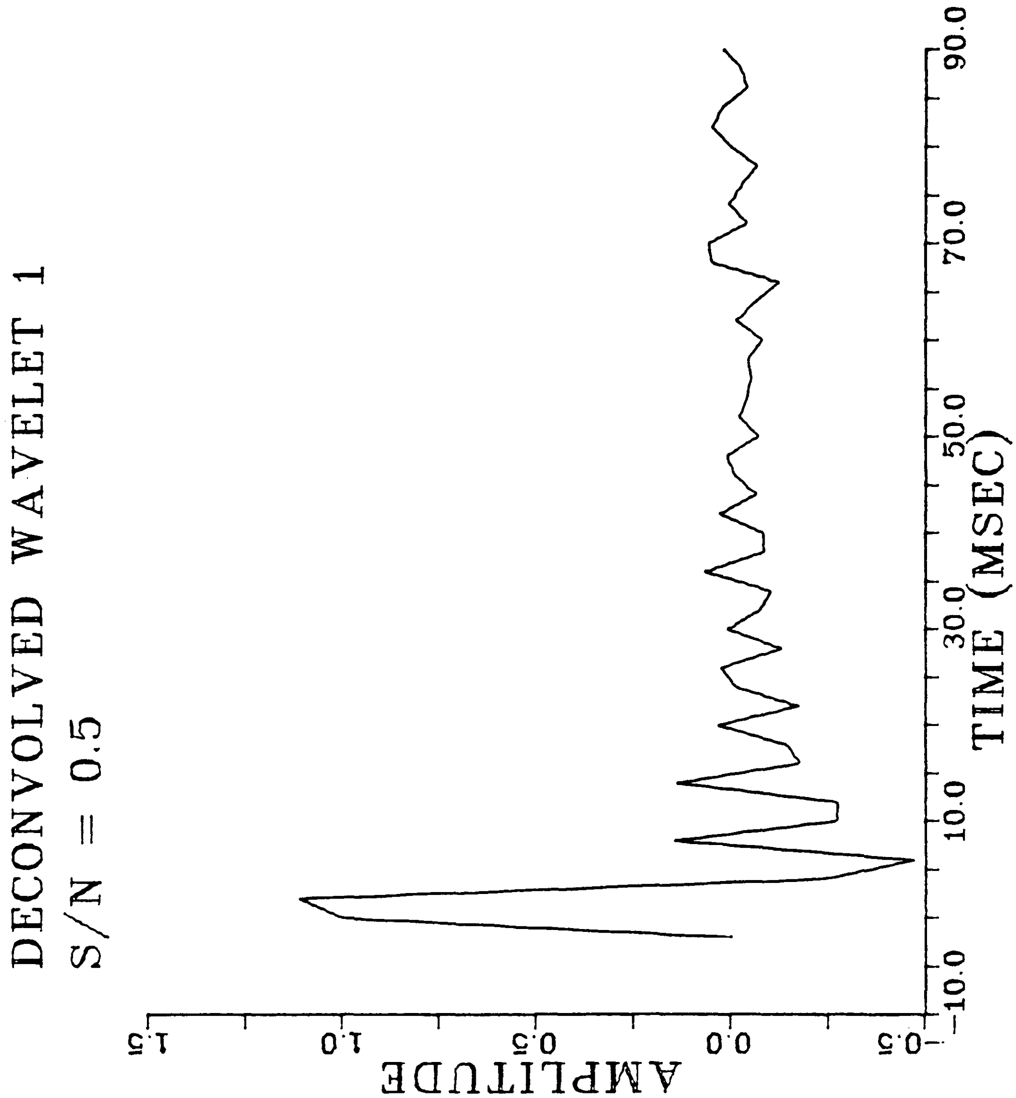

DECONVOLVED WAVELET 1
S/N = 0.5
TIME (MSEC)
-10.0
10.0
30.0
50.0
70.0
90.0
AMPLITUDE
-0.5
0.0
0.5
1.0
1.5

Figure 7c

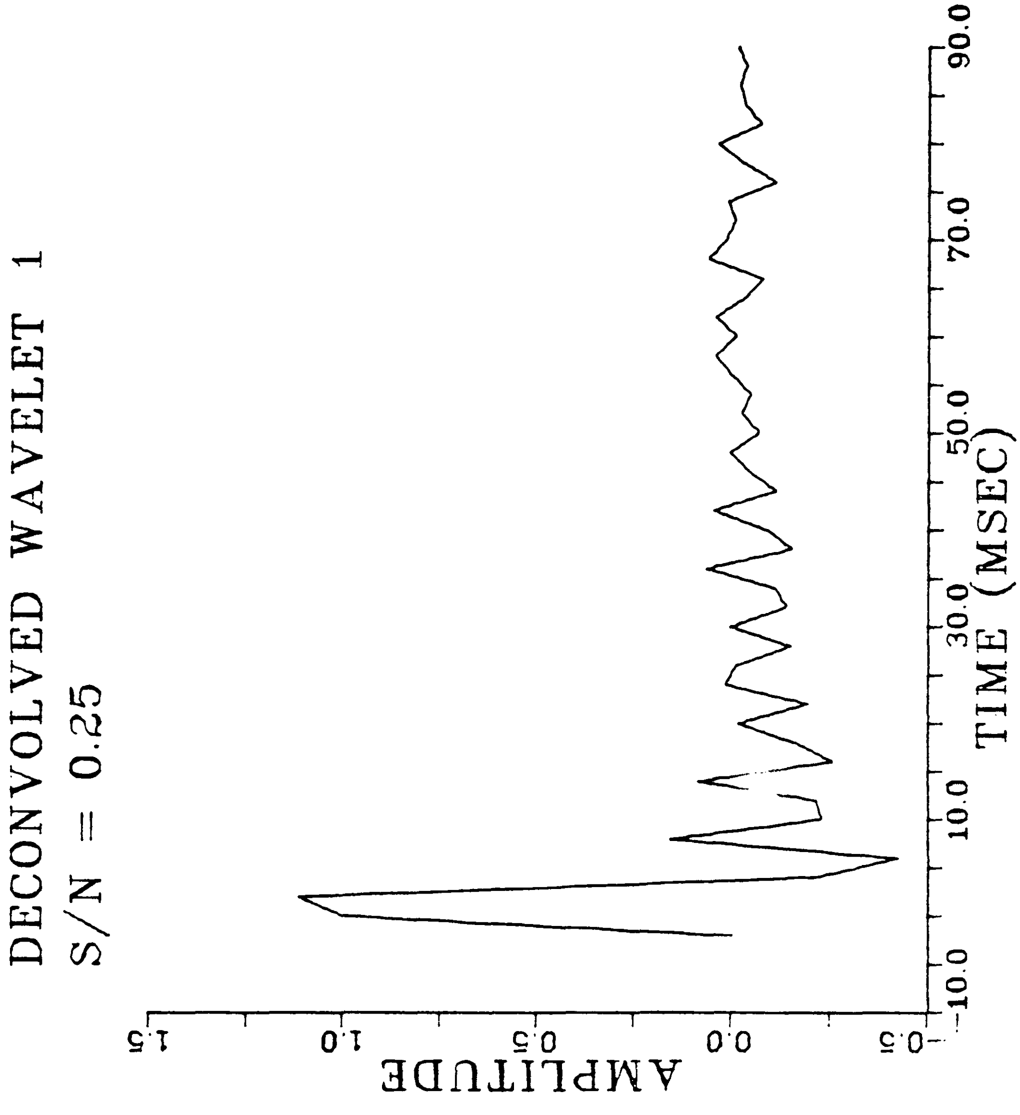

DECONVOLVED WAVELET 1
S/N = 0.25
TIME (MSEC)
-10.0
10.0
30.0
50.0
70.0
90.0
AMPLITUDE
-0.5
0.0
0.5
1.0
1.5

Figure 7d

are convolved with the original zero phase 5-10-125-188 passband wavelet. These results are shown in Figures 8a through 8d. The desired output at this processing stage would be a symmetrical zero phase pulse; however, as shown by Figure 8, the result in the presence of noise is a time delayed wavelet that appears to be minimum phase, followed by a train of noisy reverberations. The results worsen noticeably as the signal-to-noise ratio decreases.

The departure of these wavelets from zero phase is even more marked upon examination of their spectra. Fourier transformation of the wavelet in Figure 8a results in the amplitude and phase spectra shown by Figures 9a and 9b. Whereas the amplitude spectrum shows little variance from its ideal counterpart (Figure 1), the phase spectrum departs significantly from zero phase.

It should be noted here that plus or minus one on the phase angle axis of Figure 9b implies a phase lead or lag of 180 degrees. This rule will also apply to ensuing phase spectral curves.

From DC to about sixty Hertz the wavelet has a phase lead; after sixty Hertz, the phase lags and reaches a minimum value of minus π at the Nyquist frequency. Within the useable bandpass of the wavelet, from about ten to 120 Hertz, there is an approximate linear phase shift of 0.3 degrees/Hertz.

Outside the bandlimits of the filter, where the amplitudes drop to negligible values, the phase values

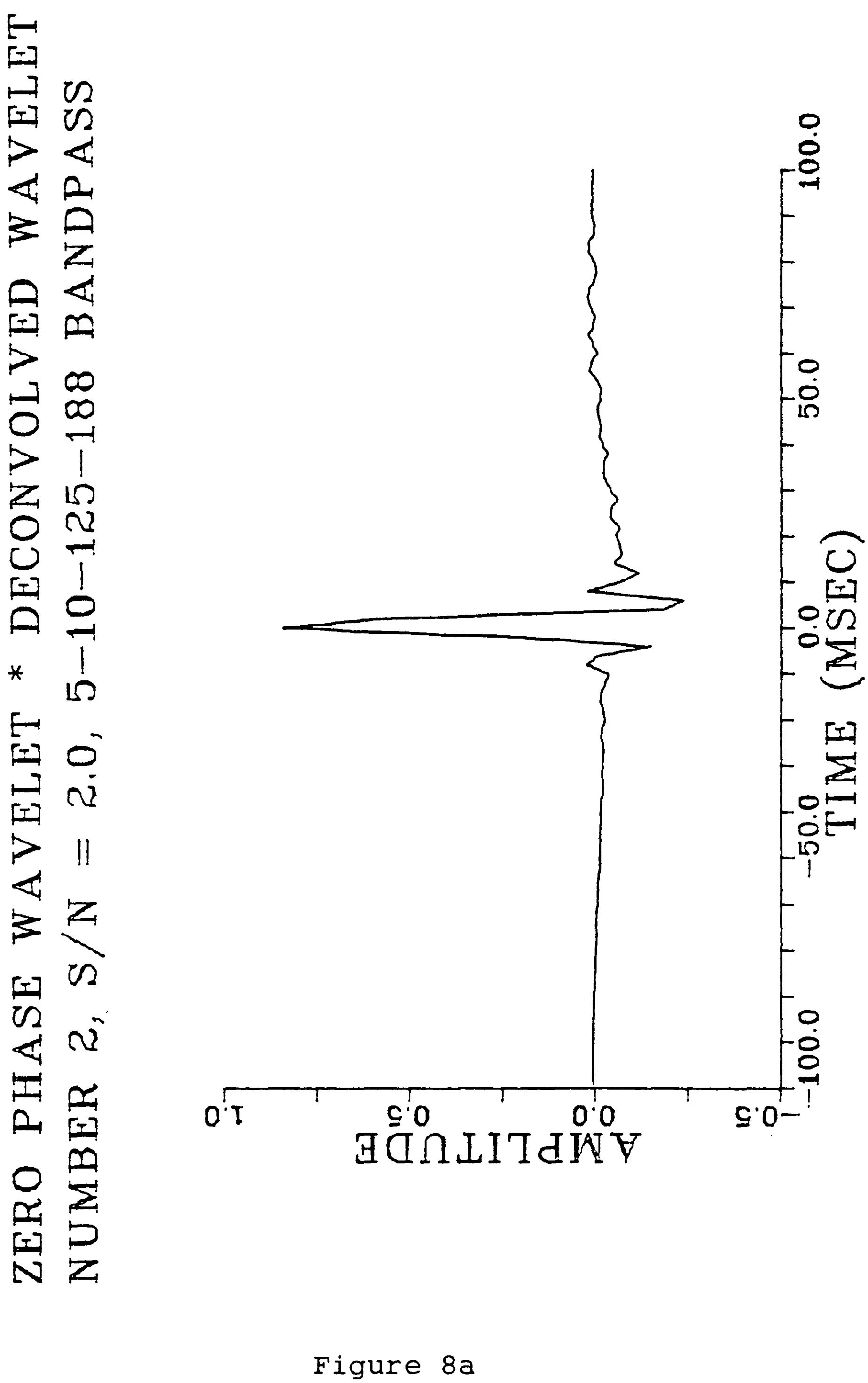

ZERO PHASE WAVELET * DECONVOLVED WAVELET
NUMBER 2, S/N = 2.0, 5–10–125–188 BANDPASS
AMPLITUDE
1.0
0.5
0.0
-0.5
TIME (MSEC)
-100.0
-50.0
0.0
50.0
100.0

Figure 8a

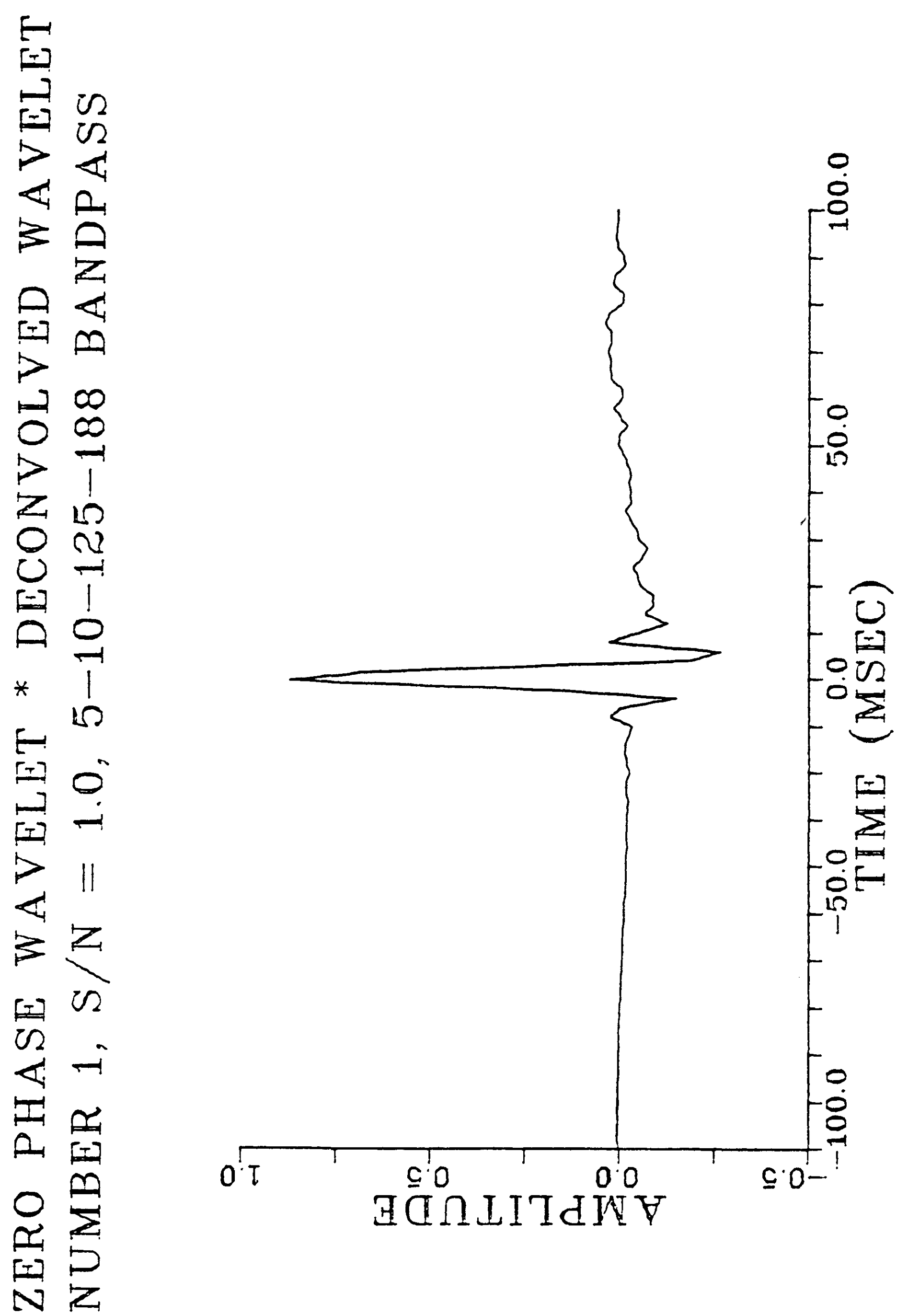

ZERO PHASE WAVELET * DECONVOLVED WAVELET
NUMBER 1, S/N = 1.0, 5-10-125-188 BANDPASS
AMPLITUDE
1.0
0.5
0.0
-0.5
TIME (MSEC)
-100.0
-50.0
0.0
50.0
100.0

Figure 8b

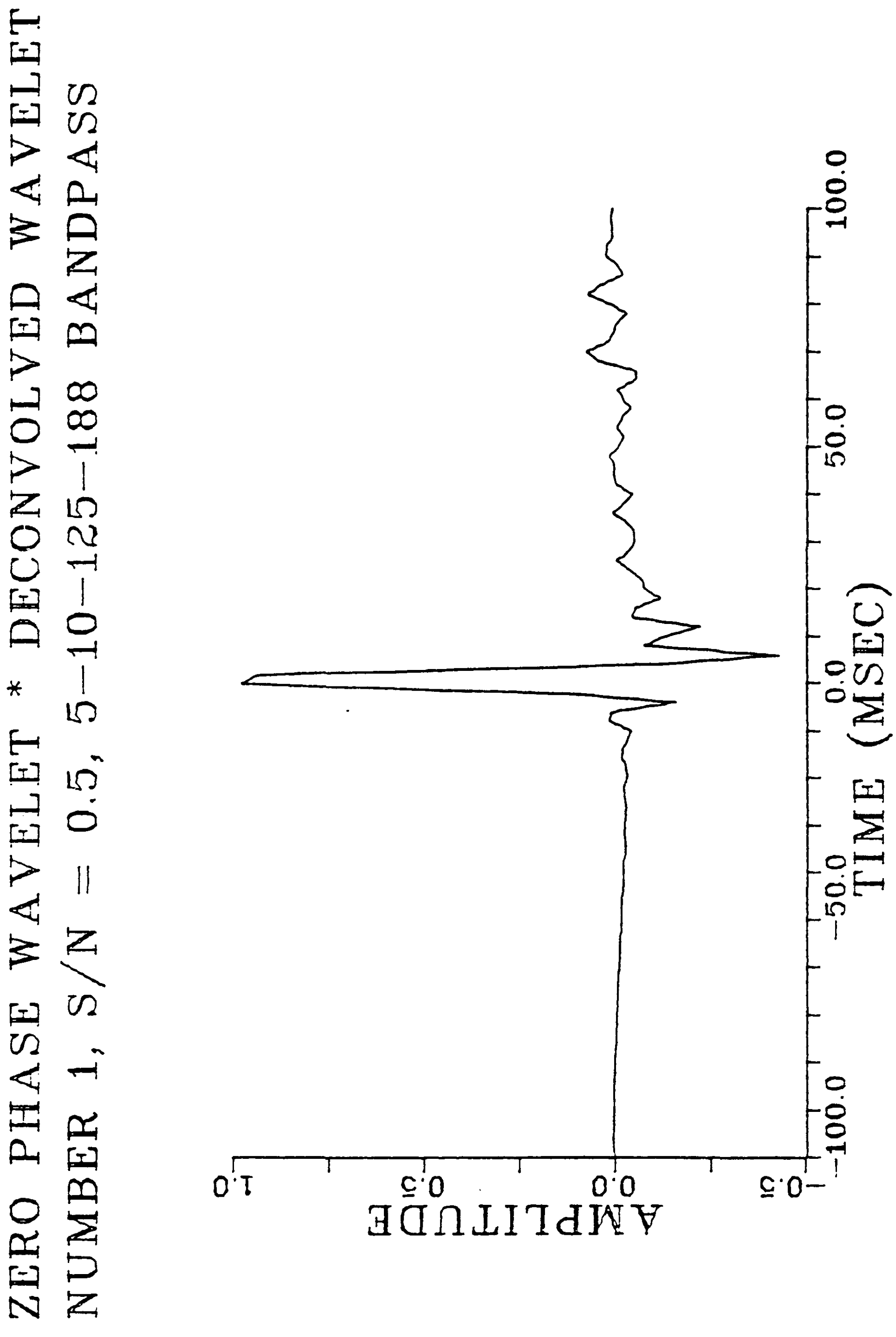

ZERO PHASE WAVELET * DECONVOLVED WAVELET
NUMBER 1, S/N = 0.5, 5—10—125—188 BANDPASS
AMPLITUDE
1.0
0.5
0.0
-0.5
TIME (MSEC)
-100.0
-50.0
0.0
50.0
100.0

Figure 8c

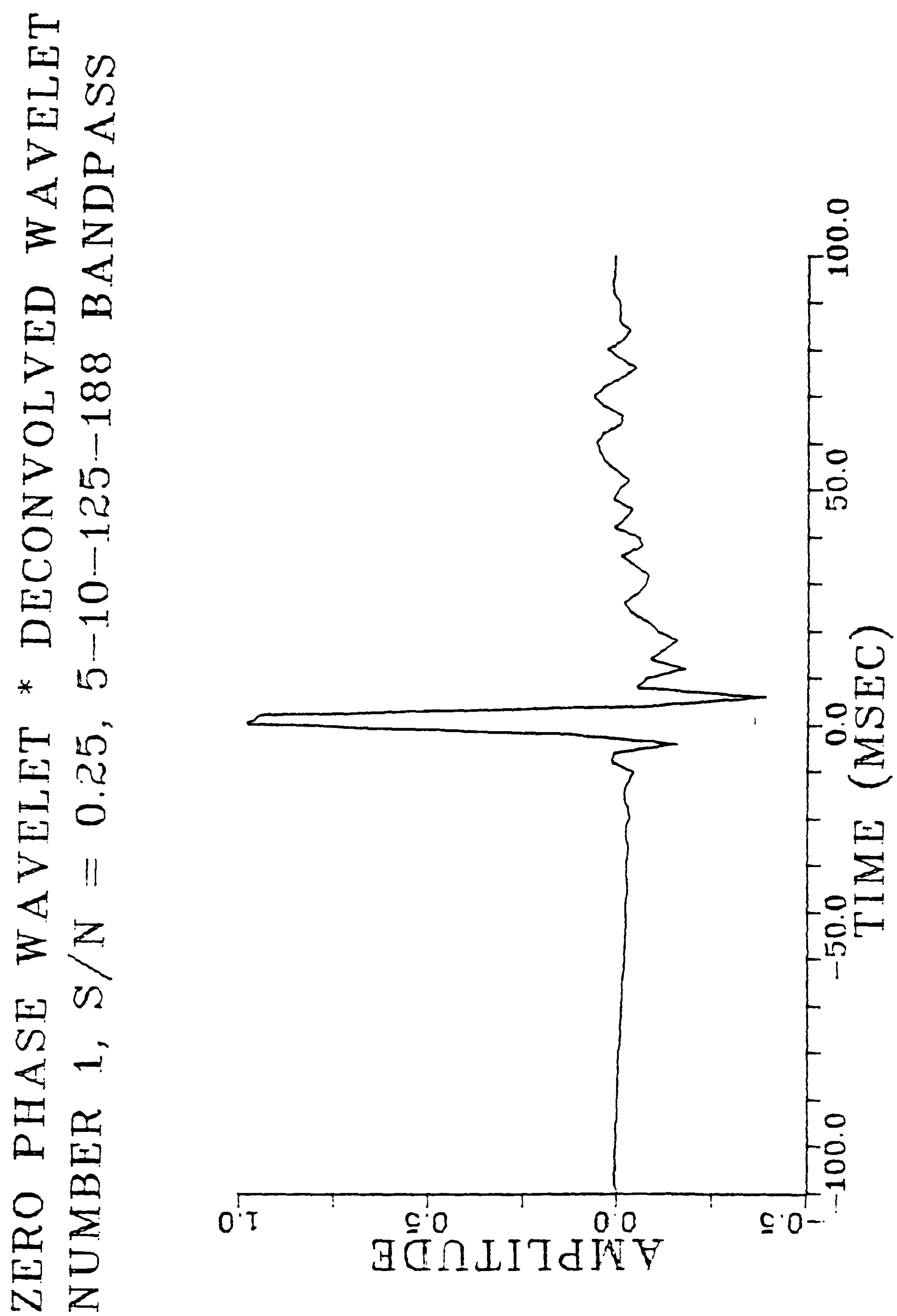
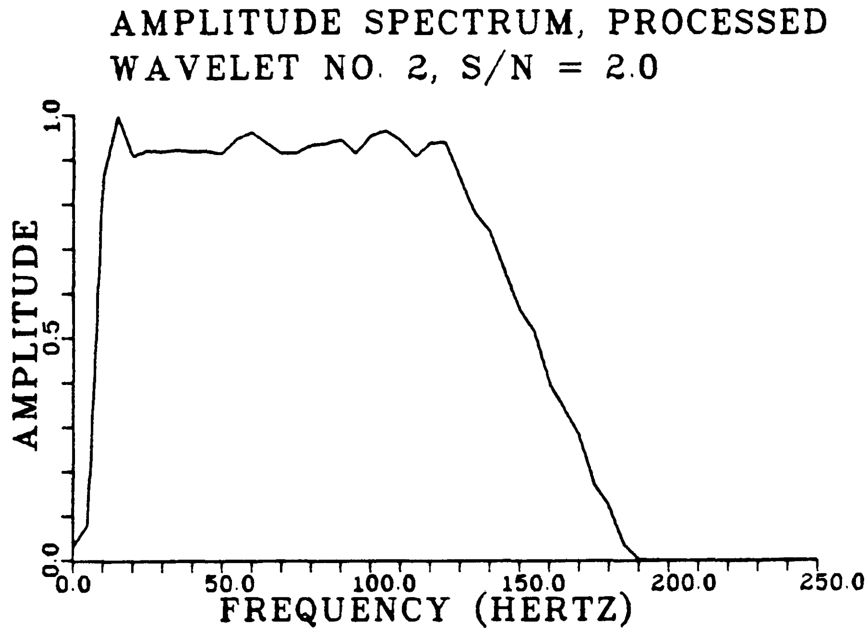
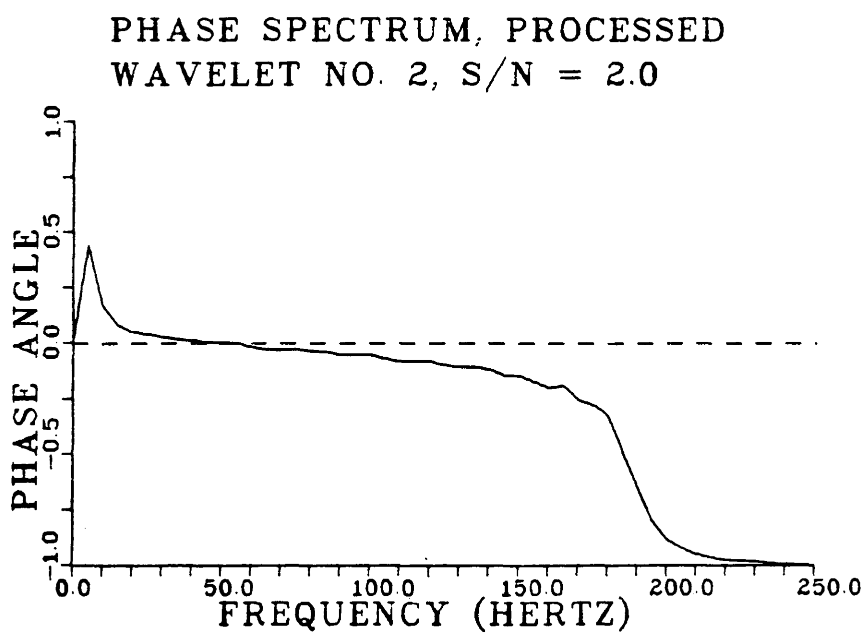

ZERO PHASE WAVELET * DECONVOLVED WAVELET
NUMBER 1, S/N = 0.25, 5-10-125-188 BANDPASS
AMPLITUDE
1.0
0.5
0.0
-0.5
TIME (MSEC)
-100.0
-50.0
0.0
50.0
100.0

Figure 8d


AMPLITUDE SPECTRUM, PROCESSED
WAVELET NO. 2, S/N = 2.0
AMPLITUDE
1.0
0.5
0.0
0.0
50.0
100.0
150.0
200.0
250.0
FREQUENCY (HERTZ)

(a)


PHASE SPECTRUM, PROCESSED
WAVELET NO. 2, S/N = 2.0
PHASE ANGLE
1.0
0.5
0.0
-0.5
-1.0
0.0
50.0
100.0
150.0
200.0
250.0
FREQUENCY (HERTZ)

(b)

Figure 9

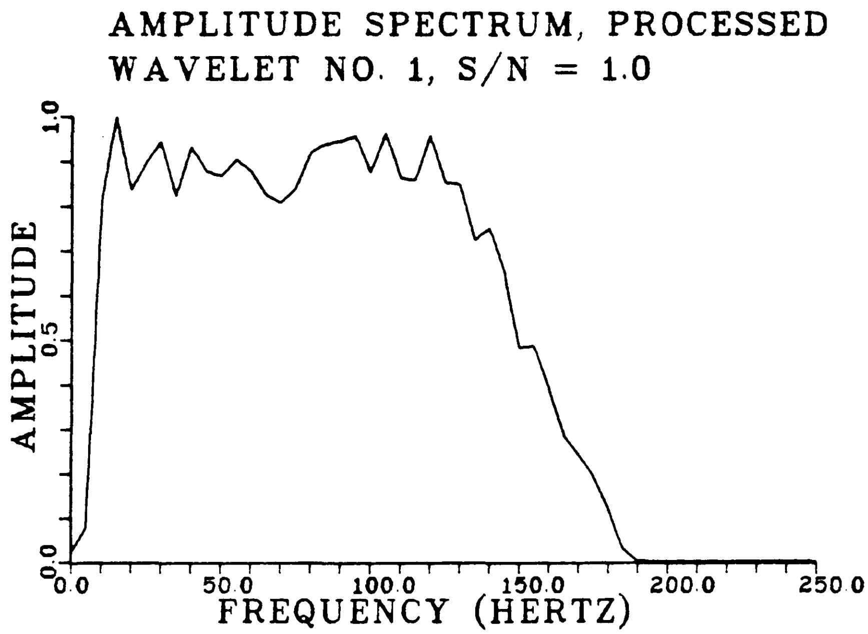
are unreliable and should be discounted. Inability of the computer to correctly compute the arctangent for small real values causes this phenomenon. The actual output values display many discontinuities and are made relatively smooth and continuous for purely aesthetic reasons. This smoothing also results in the curves not returning to zero at the Nyquist frequency, which is theoretically the outcome for minimum phase spectra.

The computer does, however, give correct phase values within the bandpass of the wavelet, and results from the discussion of the preceding paragraph have no influence upon the conclusions reached in this study.

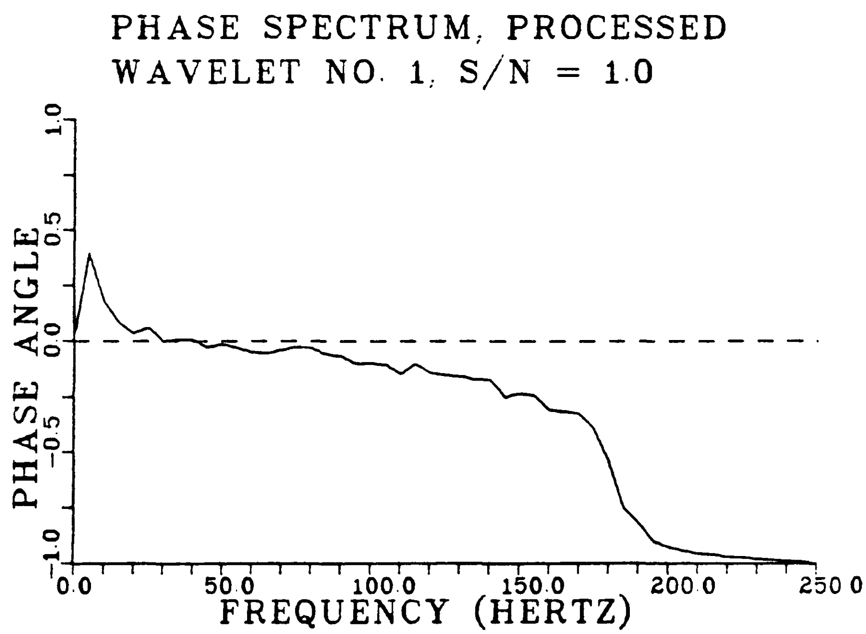
When the signal-to-noise ratio decreases, the general shape of the spectra do not worsen to any great extent, nevertheless, the curves do become increasingly choppy and irregular due to the ever increasing presence of noise. Figures 10, 11, and 12 demonstrate this result. These three figures depict, as did Figure 9b, the -0.3 degrees/Hertz linear phase shift throughout the frequency bandwidth of the wavelet. Interpolation of the observed linear phase shift back to the ordinate axis yields a value slightly less than one-eighth π, translating to a phase shift of approximately twenty degrees.

The time shift of the preceding sequence of wavelets is given by

$$\frac{d\phi}{dt} = -2\pi\Delta t \qquad (29)$$

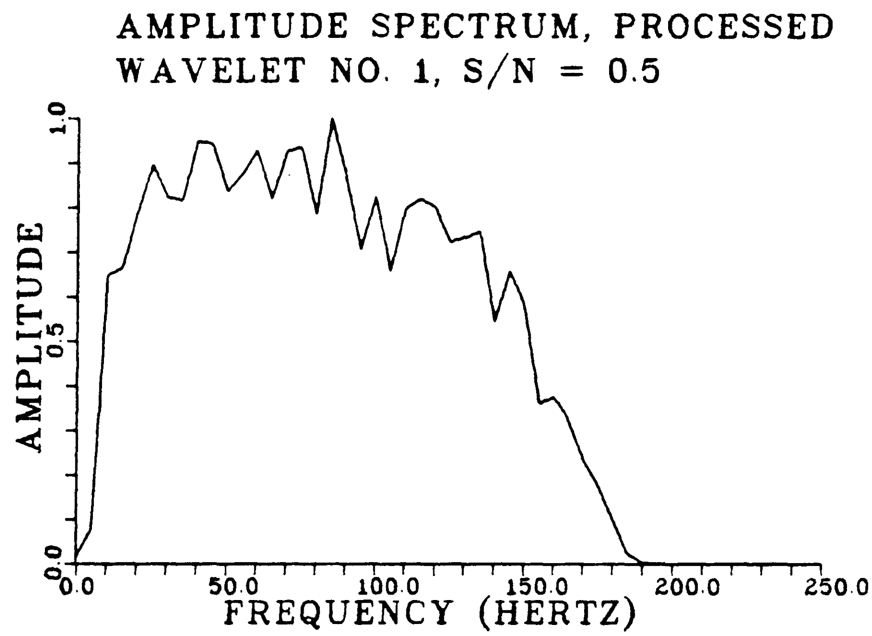

AMPLITUDE SPECTRUM, PROCESSED
WAVELET NO. 1, S/N = 1.0
AMPLITUDE
0.0
0.5
1.0
0.0
50.0
100.0
150.0
200.0
250.0
FREQUENCY (HERTZ)

(a)

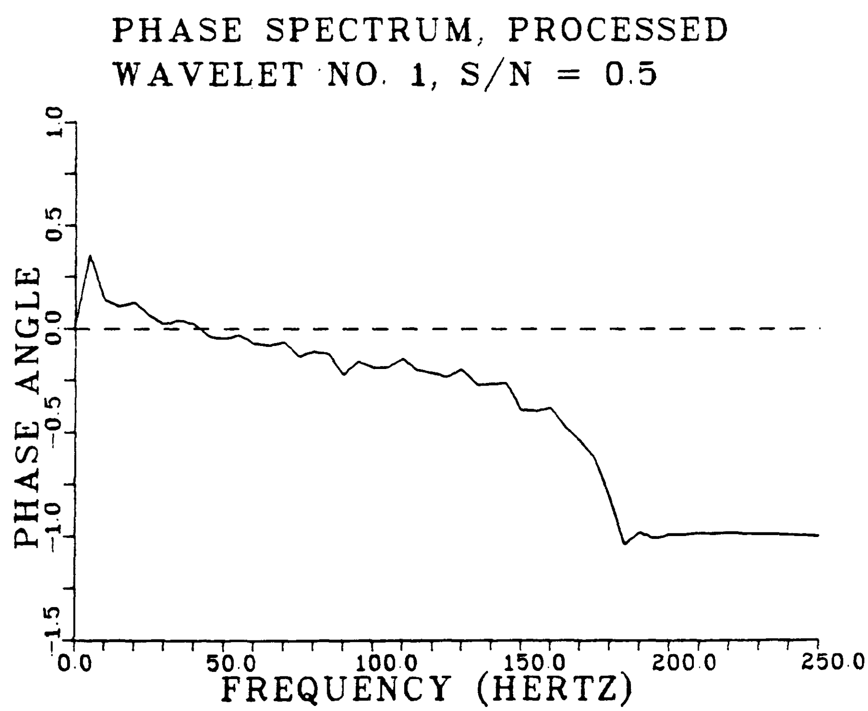

PHASE SPECTRUM, PROCESSED
WAVELET NO. 1, S/N = 1.0
PHASE ANGLE
-1.0
-0.5
0.0
0.5
1.0
0.0
50.0
100.0
150.0
200.0
250.0
FREQUENCY (HERTZ)

(b)

Figure 10

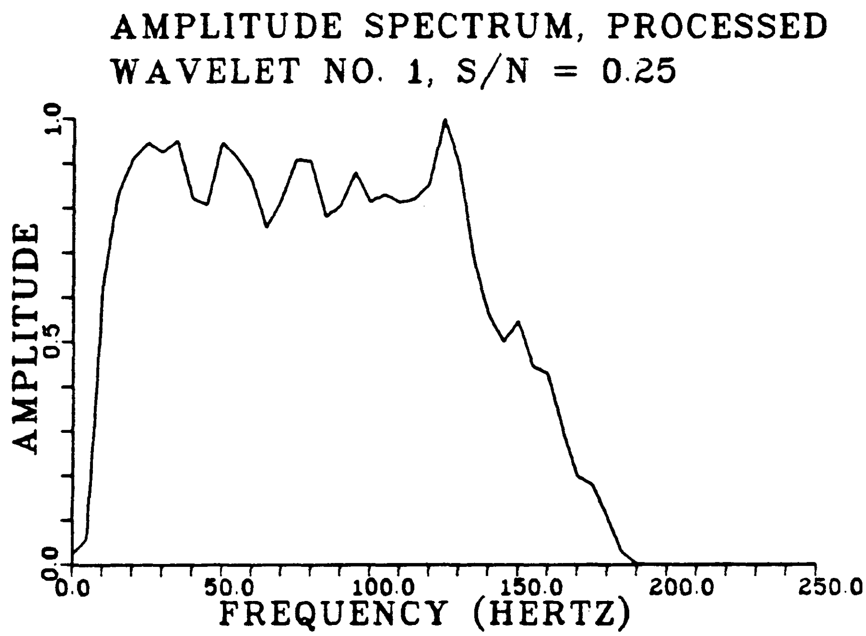

AMPLITUDE SPECTRUM, PROCESSED
WAVELET NO. 1, S/N = 0.5
AMPLITUDE
0.0
0.5
1.0
0.0
50.0
100.0
150.0
200.0
250.0
FREQUENCY (HERTZ)

(a)

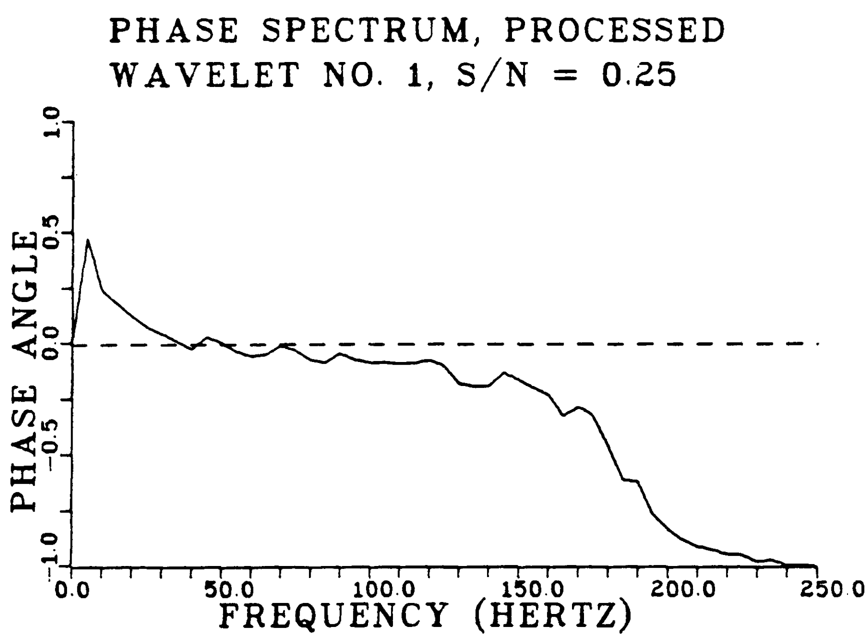

PHASE SPECTRUM, PROCESSED
WAVELET NO. 1, S/N = 0.5
PHASE ANGLE
-1.5
-1.0
-0.5
0.0
0.5
1.0
0.0
50.0
100.0
150.0
200.0
250.0
FREQUENCY (HERTZ)

(b)

Figure 11


AMPLITUDE SPECTRUM, PROCESSED
WAVELET NO. 1, S/N = 0.25
AMPLITUDE
0.0
0.5
1.0
0.0
50.0
100.0
150.0
200.0
250.0
FREQUENCY (HERTZ)

(a)


PHASE SPECTRUM, PROCESSED
WAVELET NO. 1, S/N = 0.25
PHASE ANGLE
-1.0
-0.5
0.0
0.5
1.0
0.0
50.0
100.0
150.0
200.0
250.0
FREQUENCY (HERTZ)

(b)

Figure 12

where ϕ is the phase angle of the wavelet, and Δt is the time shift. Given the time derivative of the phase to be -0.3 degrees/Hertz, a time shift of .0008 seconds may be calculated by use of equation (29). Hence, the wavelets depicted in Figures 8a through 8d possess a phase shift of twenty degrees and a time lag of approximately one millisecond.

Few geophysicists today are required to interpret one hundred percent data. Consequently, to attain more realistic results, each ensemble of a dozen wavelets recorded at a particular signal-to-noise ratio are averaged to simulate a twelve fold common depth point stack. Depicted in Figure 13 is an ensemble of twelve wavelets, their CDP stacked equivalent, and an uncorrupted wavelet. The uncorrupted wavelet, from Figure 2, is the desired processing result and is included for comparison purposes. Figure 14 is a magnified version of the stacked wavelet from Figure 13. With the exclusion of the noisy reverberations following the main pulse, which have been reduced by the stacking procedure, the waveform looks similar to its respective unstacked components (Figures 8a, 13).

Assuming the stacked wavelet represents the seismic response of some continuous geologic horizon, an interpreter would have little difficulty mapping that horizon, despite the wavelet being assymetrical and slightly delayed in time. Yet seismic integration and inversion schemes that produce acoustic impedance traces are becoming popular

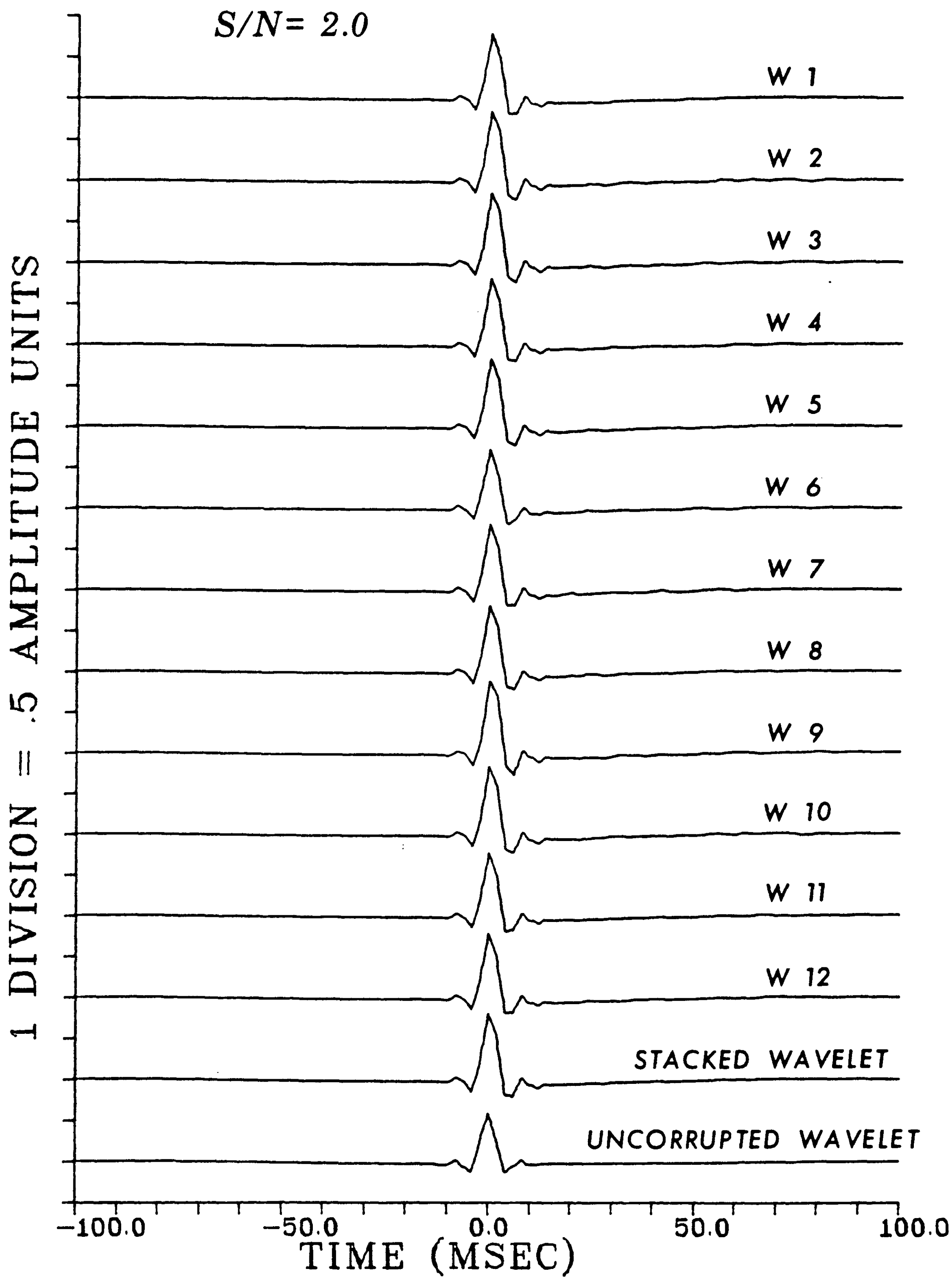


Figure 13 - Comparison of ideal vs. noise corrupted results, wavelet model

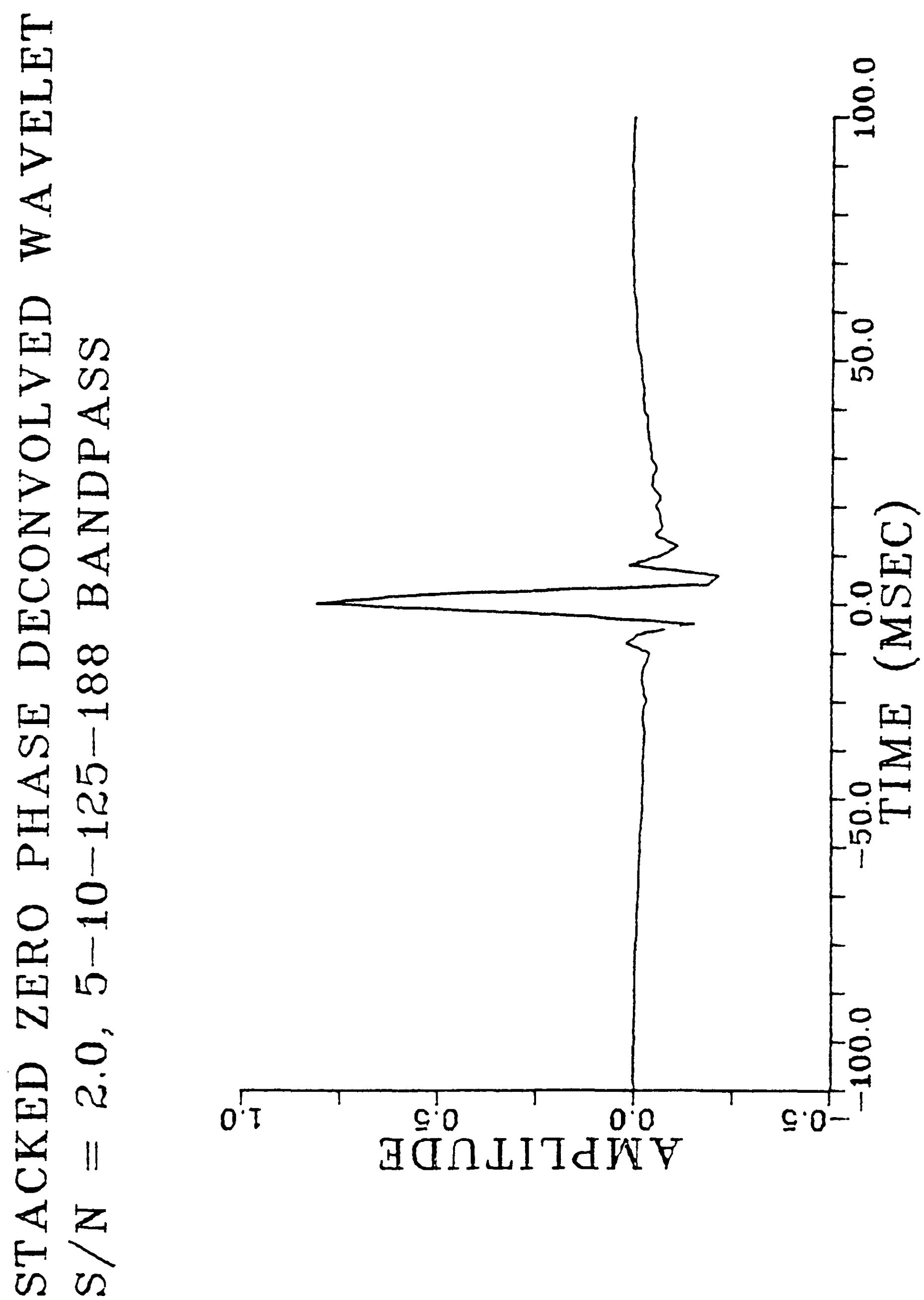

STACKED ZERO PHASE DECONVOLVED WAVELET
S/N = 2.0, 5-10-125-188 BANDPASS
AMPLITUDE
1.0
0.5
0.0
-0.5
TIME (MSEC)
-100.0
-50.0
0.0
50.0
100.0

Figure 14

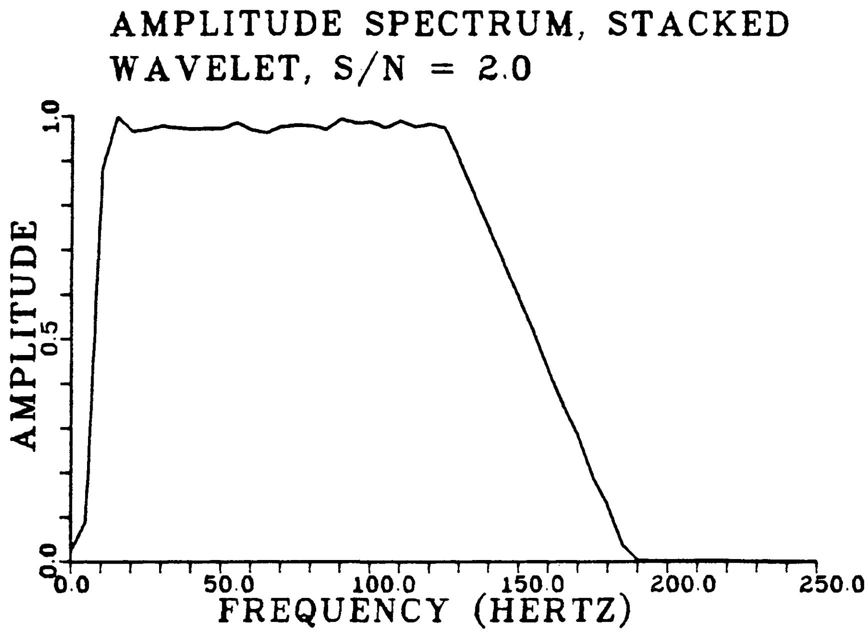
for stratigraphic interpretation throughout the industry, requiring more precise deconvolution results. Indeed, the critical process for successful seismic data inversion is a deconvolution method which yields the best approximation to a zero phase spike signal representing each reflection coefficient (Schoenberger, 1974; Lindseth, 1979). Examination of the amplitude spectrum (Figure 15a) and phase spectrum (Figure 15b) of the stacked wavelet reveals a result with the desired bandwidth, but of minimum, not zero phase.

The phase spectrum resembles that of the individual wavelet described earlier in Figure 9, although it is much smoother due to noise suppression via stacking.

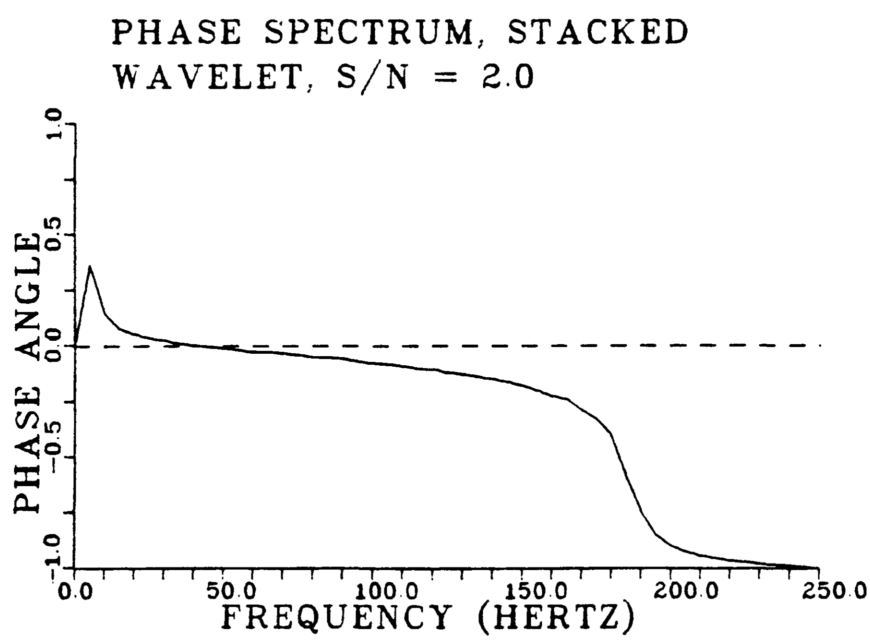
Figures 16, 17 and 18 depict the stacked wavelets and the desired result versus the twelve component wavelets of the simulated stack for signal-to-noise ratios of 1.0, 0.5, and 0.25, respectively. As the signal-to-noise ratio worsens, the bandlimited stacked results differ almost insignificantly from the result demonstrated for a signal-to-noise ratio of two.

The enlarged stacked wavelets (Figures 19, 20, and 21) also show little visible variation. Accordingly, their respective amplitude and phase spectra (Figures 22, 23, and 24) demonstrate few dissimilarities.

Slope y-intercept analyses of the stacked wavelet phase spectra yield results identical to that of the


AMPLITUDE SPECTRUM, STACKED
WAVELET, S/N = 2.0
AMPLITUDE
0.0
0.5
1.0
0.0
50.0
100.0
150.0
200.0
250.0
FREQUENCY (HERTZ)

(a)


PHASE SPECTRUM, STACKED
WAVELET, S/N = 2.0
PHASE ANGLE
-1.0
-0.5
0.0
0.5
1.0
0.0
50.0
100.0
150.0
200.0
250.0
FREQUENCY (HERTZ)

(b)

Figure 15

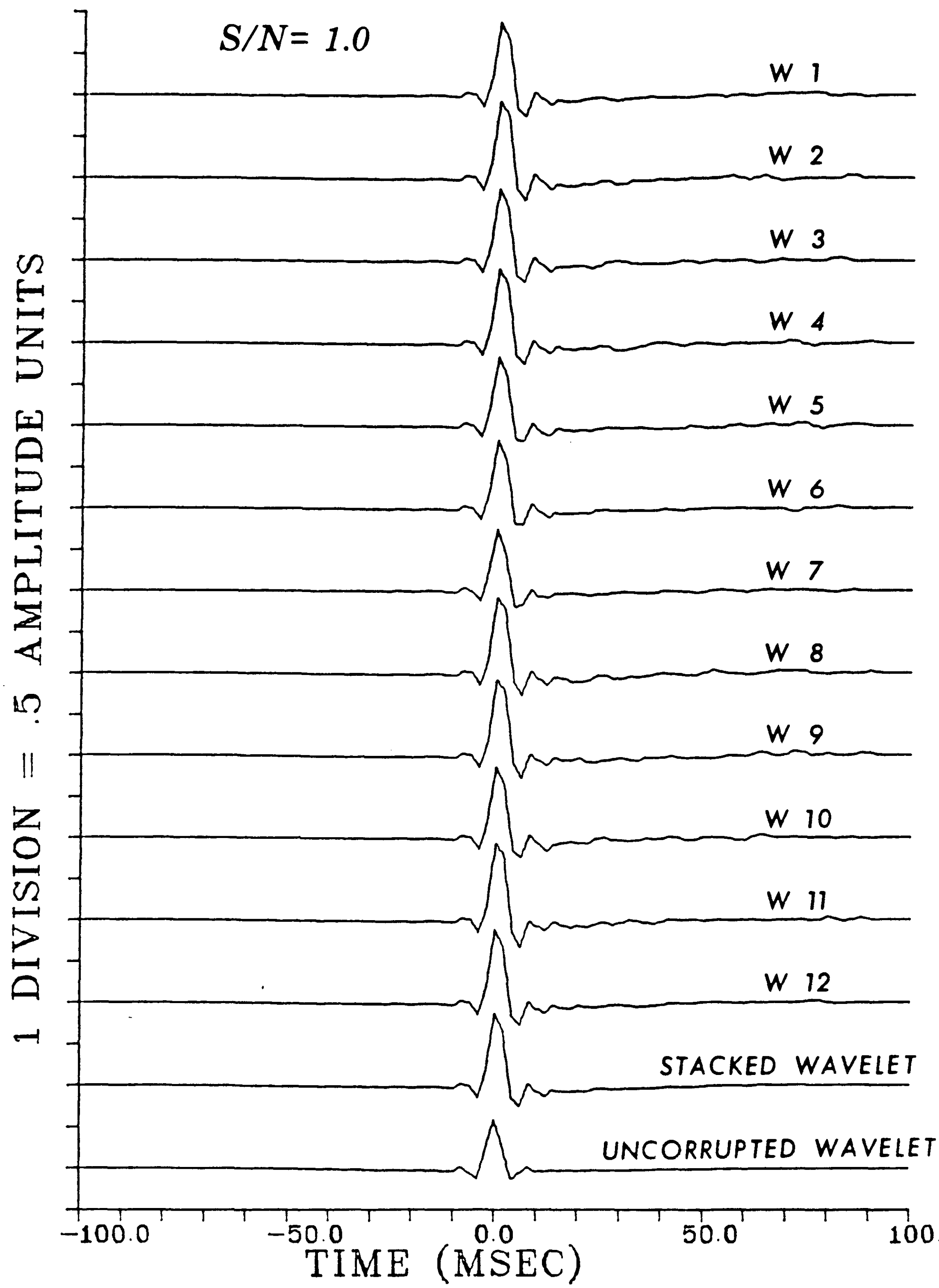


Figure 16 - Comparison of ideal vs. noise corrupted results, wavelet model

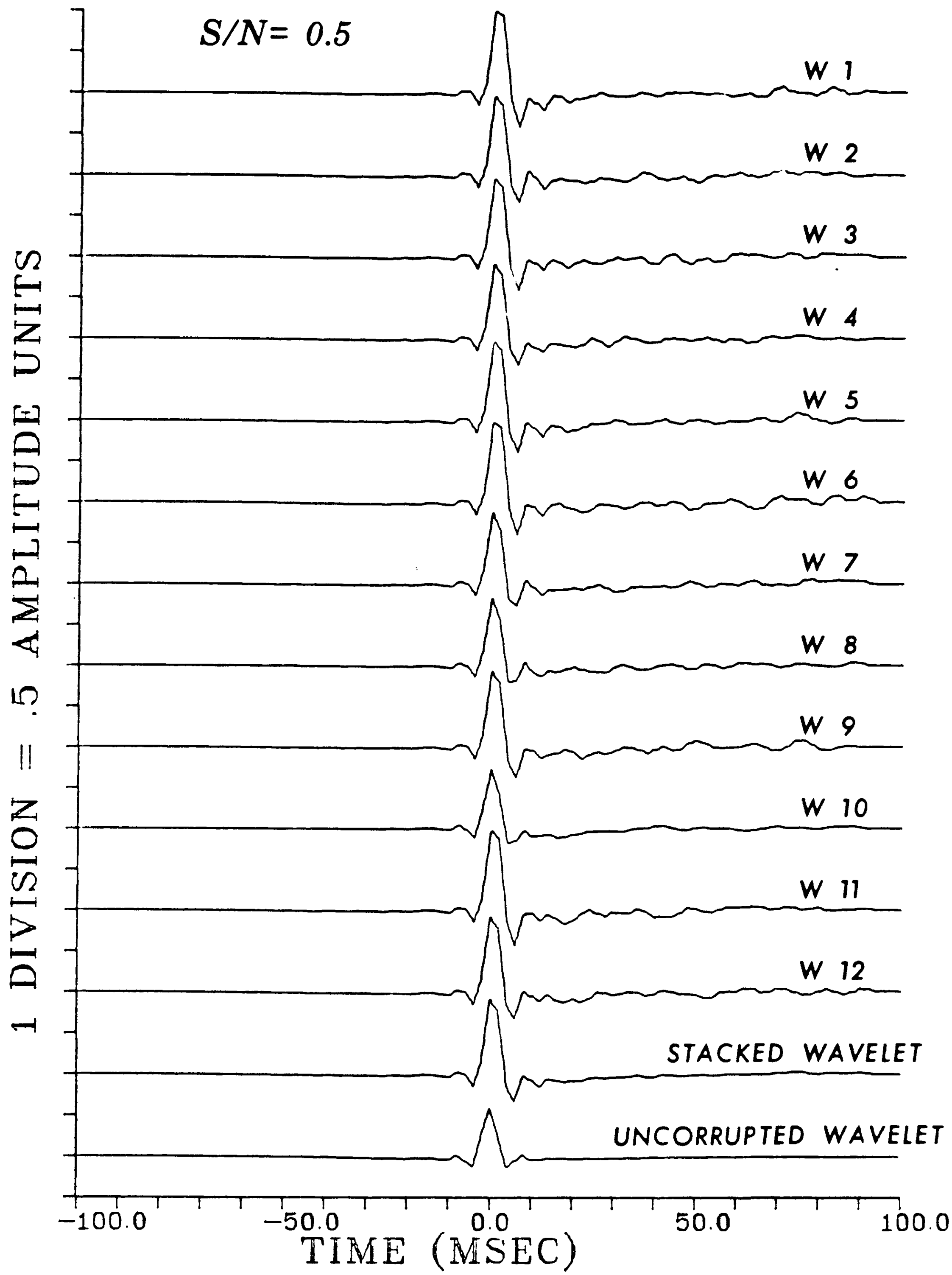


Figure 17 - Comparison of ideal vs. noise corrupted results, wavelet model

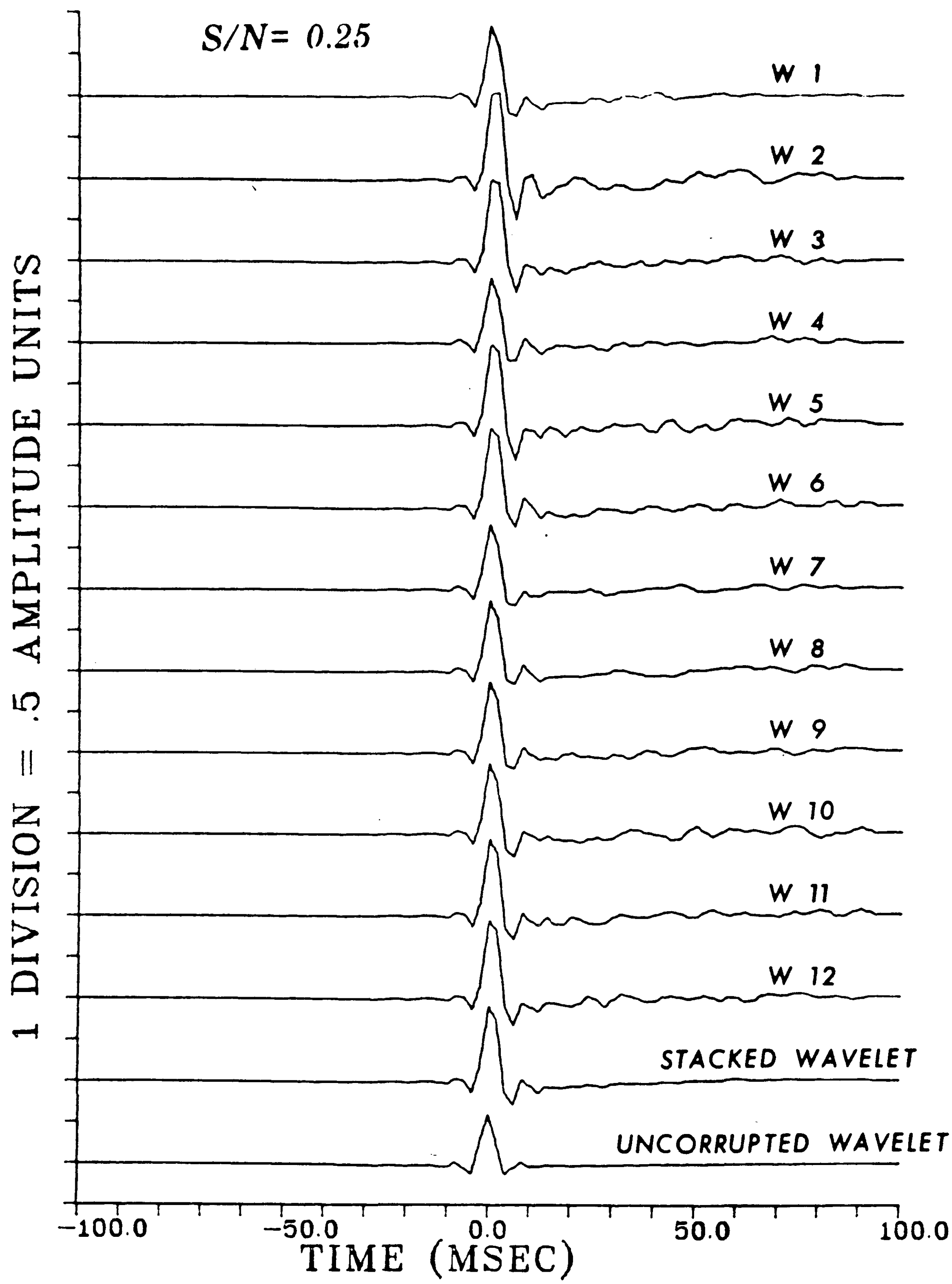


Figure 18 - Comparison of ideal vs. noise corrupted results, wavelet model

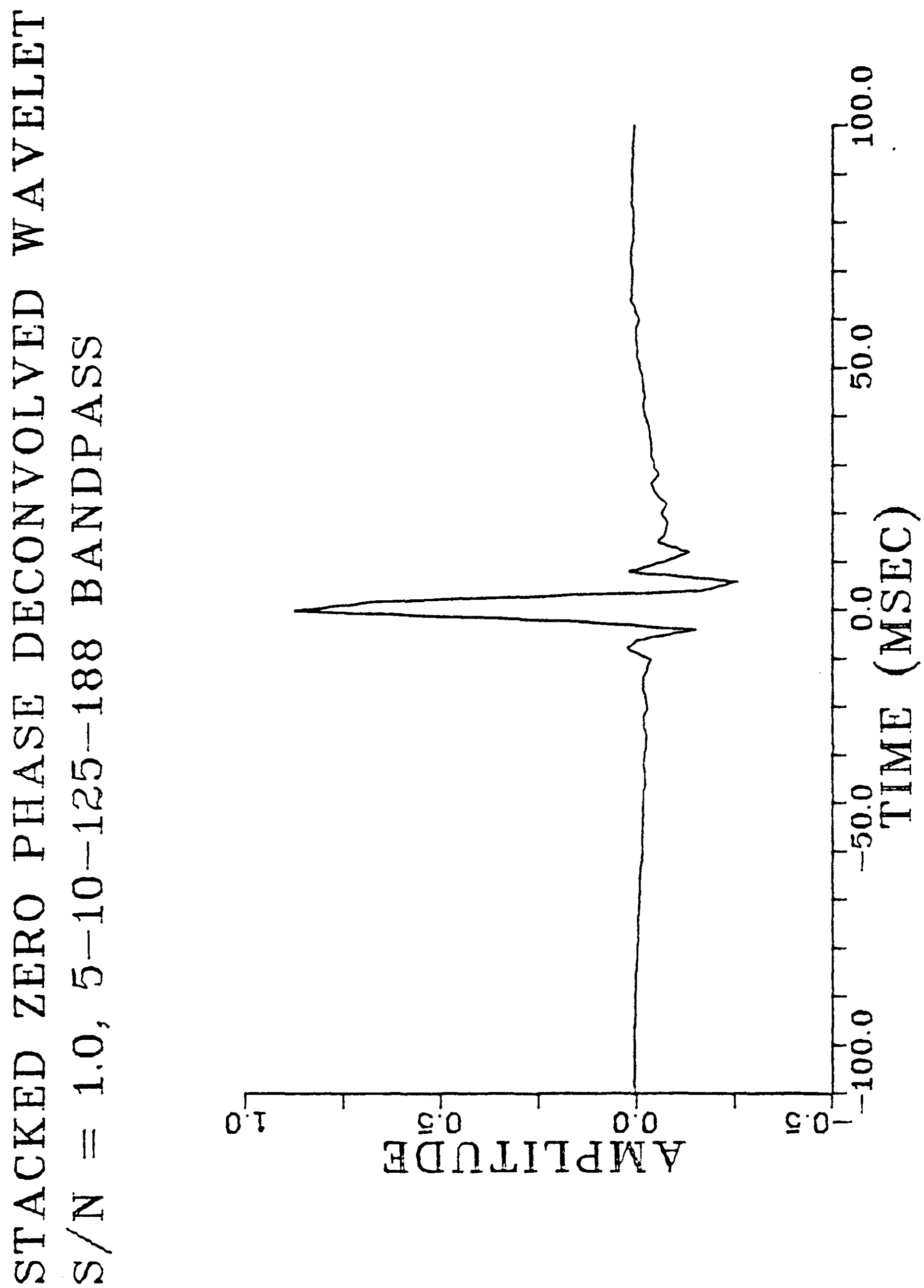

STACKED ZERO PHASE DECONVOLVED WAVELET
S/N = 1.0, 5-10-125-188 BANDPASS
AMPLITUDE
1.0
0.5
0.0
-0.5
TIME (MSEC)
-100.0
-50.0
0.0
50.0
100.0

Figure 19

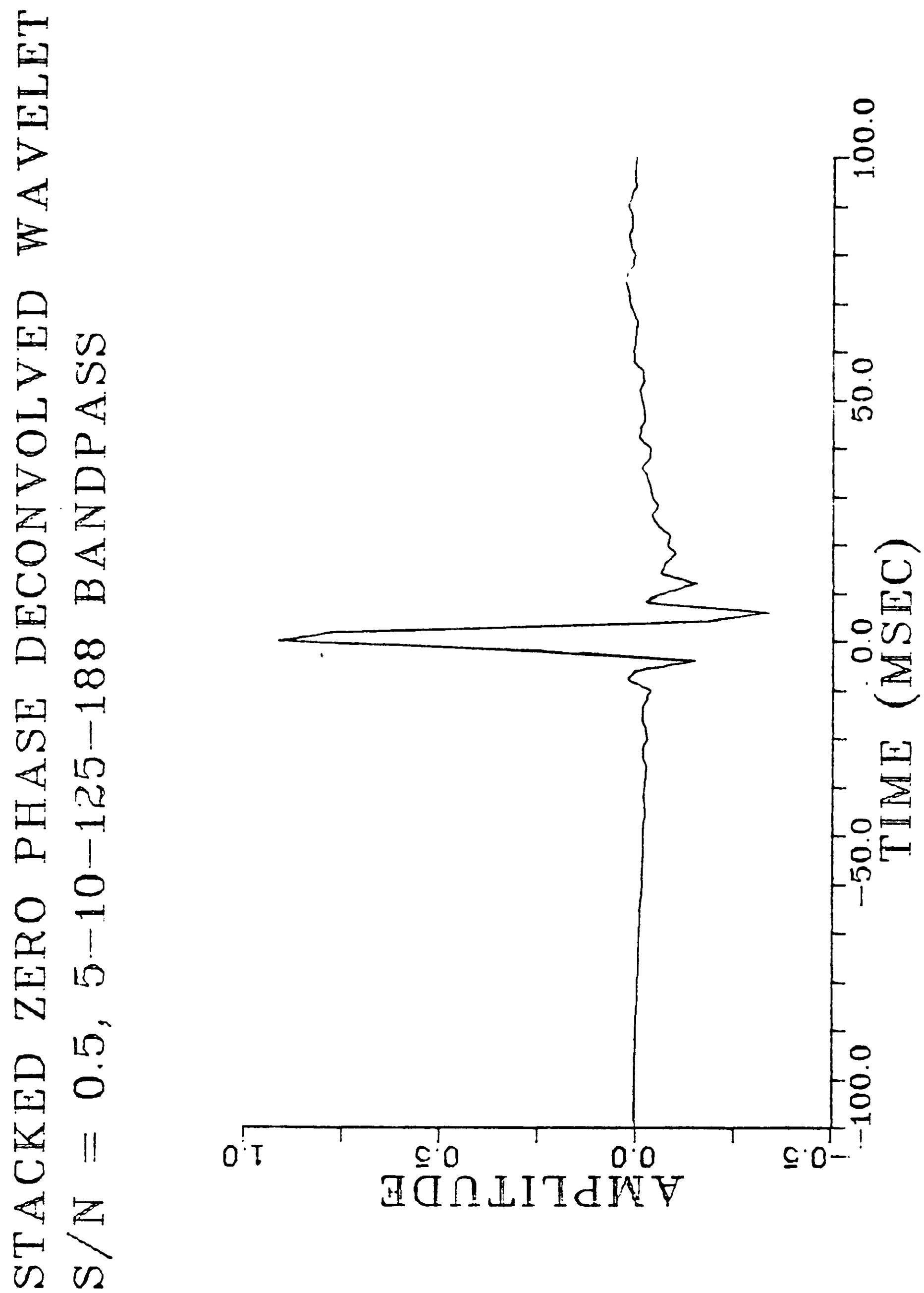

STACKED ZERO PHASE DECONVOLVED WAVELET
S/N = 0.5, 5-10-125-188 BANDPASS
AMPLITUDE
1.0
0.5
0.0
-0.5
-100.0
-50.0
0.0
50.0
100.0
TIME (MSEC)

Figure 20

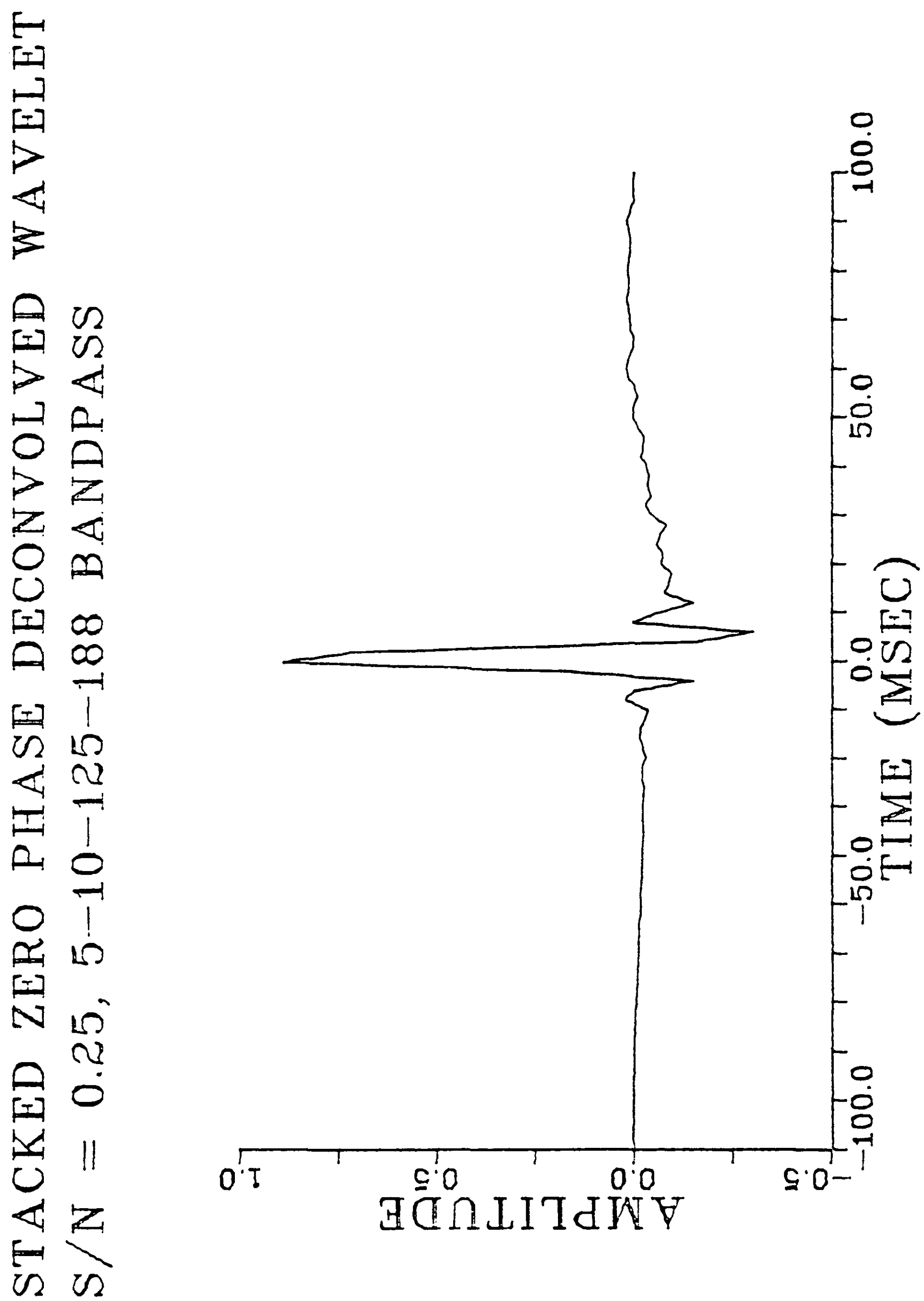
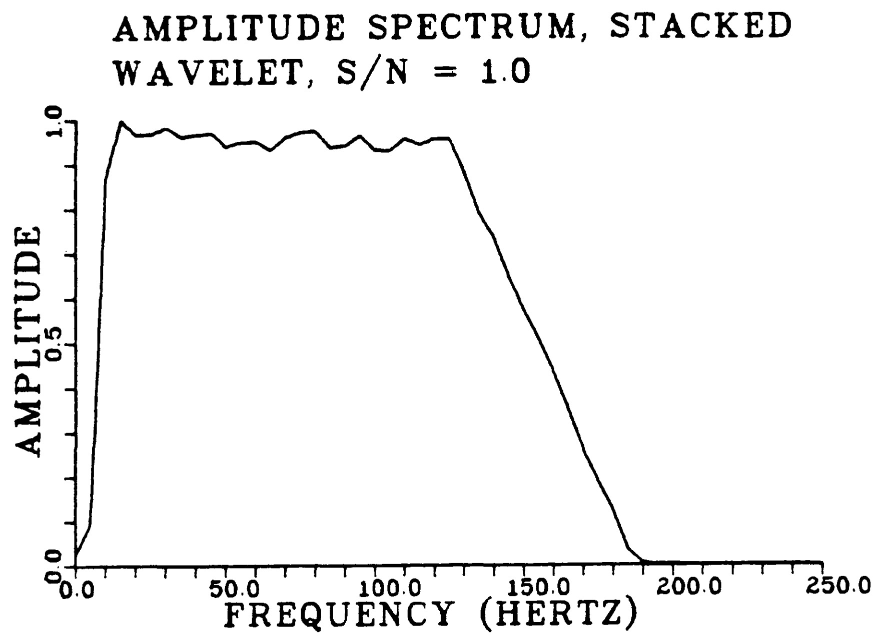
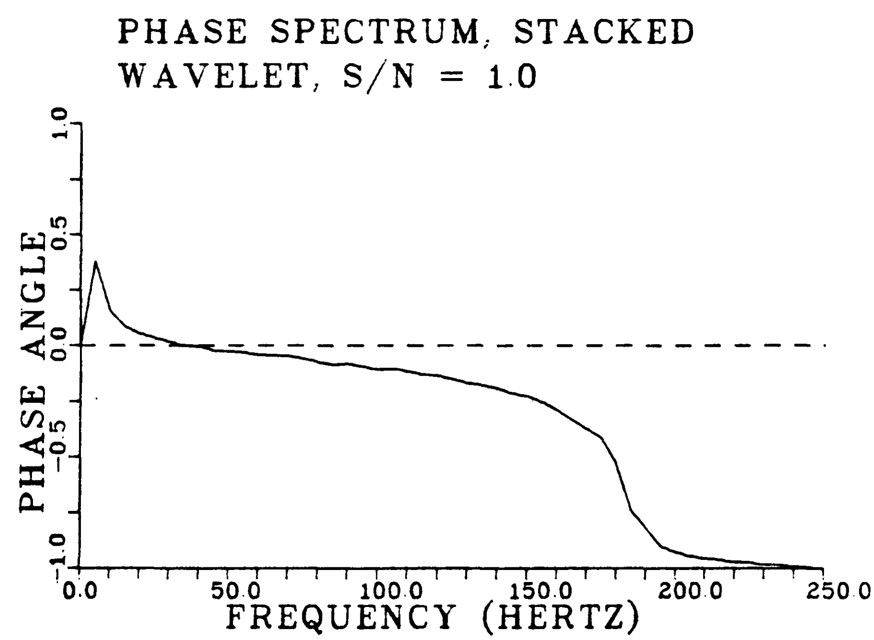

STACKED ZERO PHASE DECONVOLVED WAVELET
S/N = 0.25, 5-10-125-188 BANDPASS
AMPLITUDE
1.0
0.5
0.0
-0.5
TIME (MSEC)
-100.0
-50.0
0.0
50.0
100.0

Figure 21

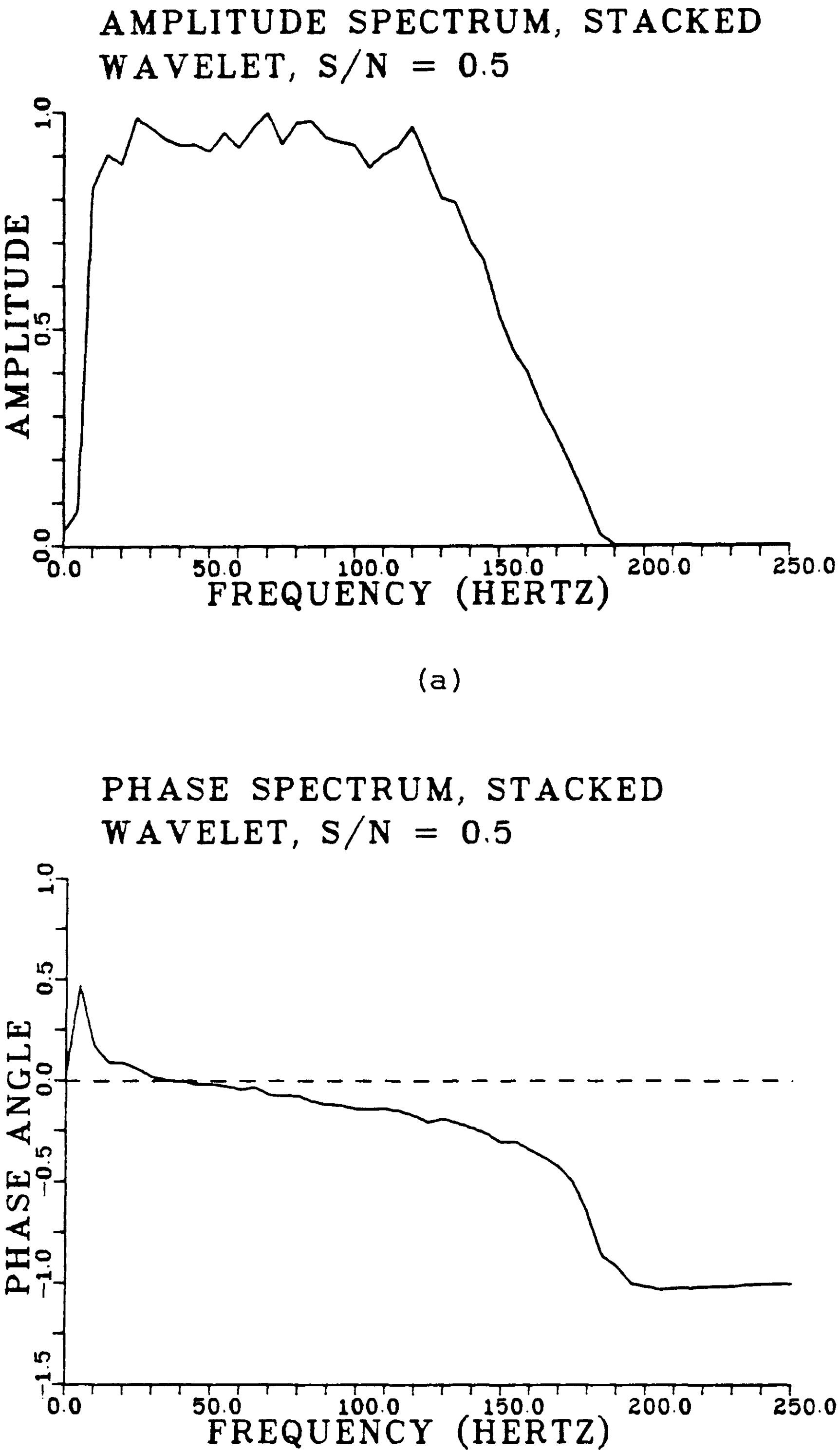

AMPLITUDE SPECTRUM, STACKED
WAVELET, S/N = 1.0
AMPLITUDE
0.0
0.5
1.0
0.0
50.0
100.0
150.0
200.0
250.0
FREQUENCY (HERTZ)

(a)


PHASE SPECTRUM, STACKED
WAVELET, S/N = 1.0
PHASE ANGLE
-1.0
-0.5
0.0
0.5
1.0
0.0
50.0
100.0
150.0
200.0
250.0
FREQUENCY (HERTZ)

(b)

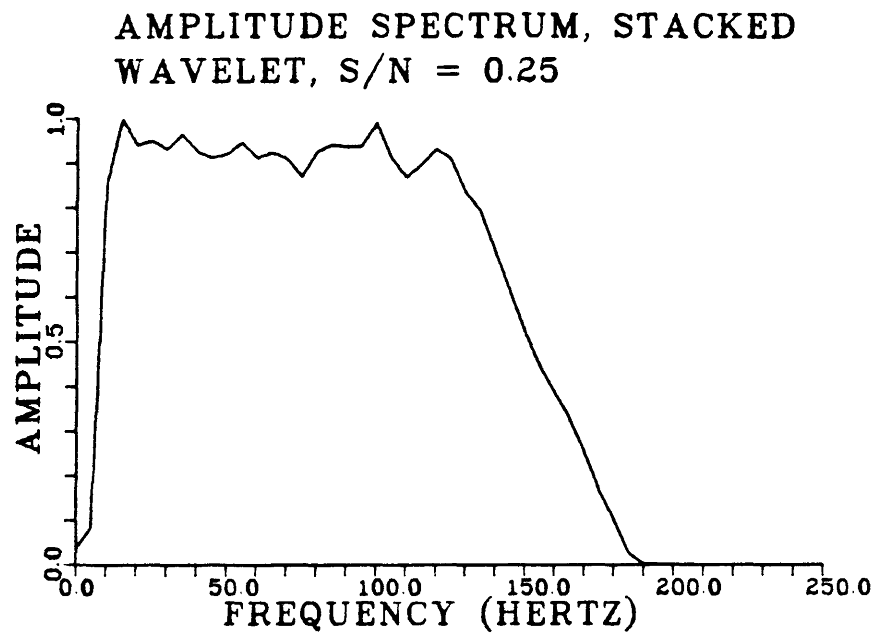
Figure 22


AMPLITUDE SPECTRUM, STACKED WAVELET, S/N = 0.5
AMPLITUDE
0.0
0.5
1.0
0.0
50.0
100.0
150.0
200.0
250.0
FREQUENCY (HERTZ)
PHASE SPECTRUM, STACKED WAVELET, S/N = 0.5
PHASE ANGLE
-1.5
-1.0
-0.5
0.0
0.5
1.0
FREQUENCY (HERTZ)

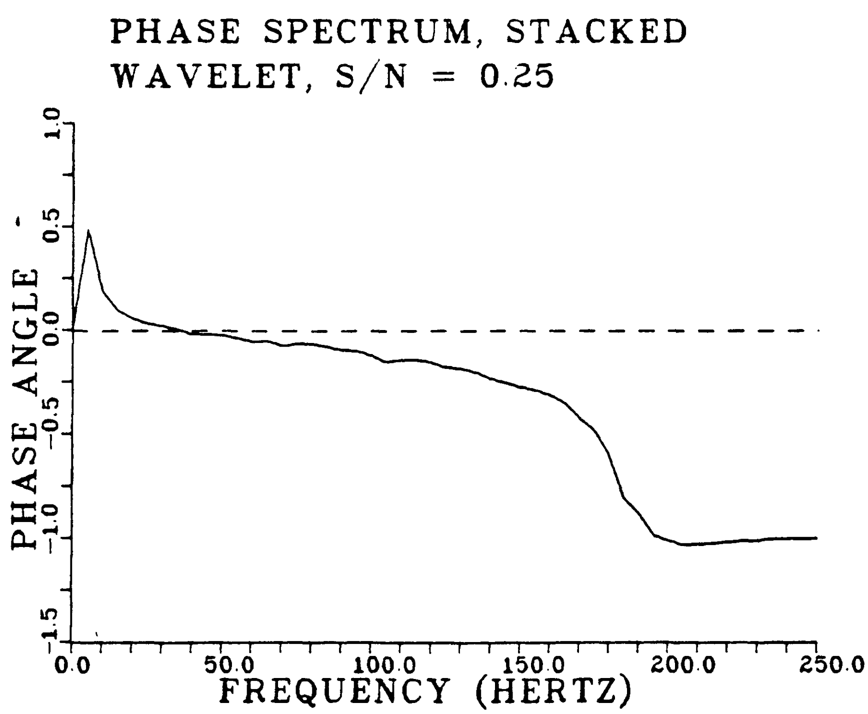
(a)

(b)

Figure 23


AMPLITUDE SPECTRUM, STACKED
WAVELET, S/N = 0.25
AMPLITUDE
1.0
0.5
0.0
0.0
50.0
100.0
150.0
200.0
250.0
FREQUENCY (HERTZ)

(a)


PHASE SPECTRUM, STACKED
WAVELET, S/N = 0.25
PHASE ANGLE -
1.0
0.5
0.0
-0.5
-1.0
-1.5
0.0
50.0
100.0
150.0
200.0
250.0
FREQUENCY (HERTZ)

(b)

Figure 24

respective unstacked components; that is, twenty degree wavelets delayed one millisecond in time.

These results tend to suggest that for two signals with similar spectral characteristics, phase error, while a function of frequency, is not strongly signal-to-noise ratio dependent, if at all.

This outcome is perhaps best depicted in Figure 25, where the log of the signal to noise ratio is plotted versus phase error in degrees for a family of curves. Each curve represents a particular frequency. Within the bandpass of the wavelet, the curves again demonstrate little dependence upon signal-to-noise ratio at higher frequencies (70-130 Hz) and no dependence at lower frequencies (30-50 HZ). The 110 Hertz curve, for example, shows only a ten degree change in phase error through a signal-to-noise ratio improvement of eighteen dB. The 150 and 170 Hz curves are well outside the bandlimits, and should not be considered as part of the general result implied by the figure.

To compare the variance in phase error from trace to trace, the standard deviation of phase error is computed and plotted versus frequency for each twelve-fold ensemble. Figures 26, 27, 28, and 29 exhibit the results. A general increase in standard deviation with increasing frequency is observed throughout the wavelet bandpass. Comparison

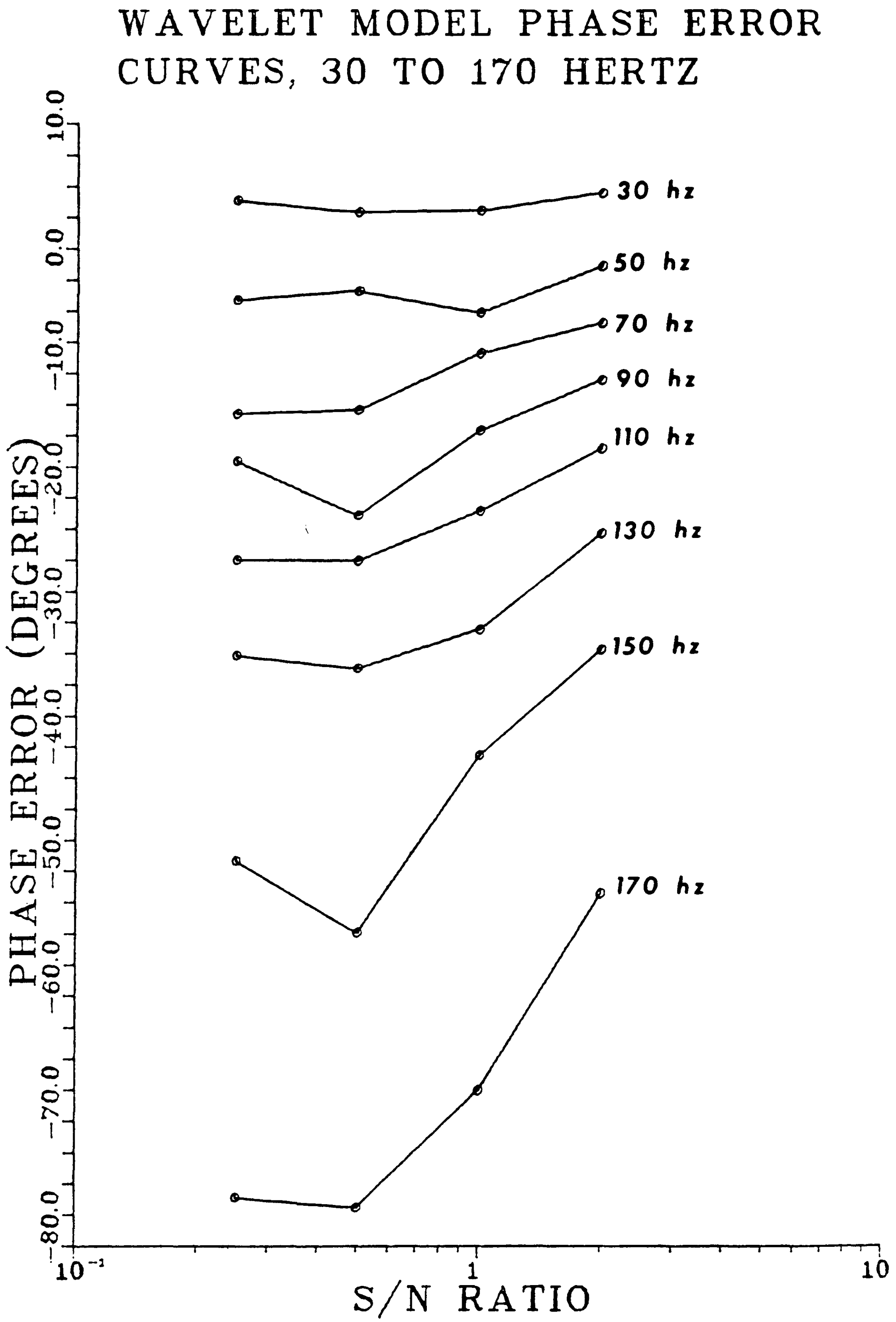

WAVELET MODEL PHASE ERROR
CURVES, 30 TO 170 HERTZ
PHASE ERROR (DEGREES)
10.0
0.0
-10.0
-20.0
-30.0
-40.0
-50.0
-60.0
-70.0
-80.0
30 hz
50 hz
70 hz
90 hz
110 hz
130 hz
150 hz
170 hz
10^{-1}
1
10
S/N RATIO

Figure 25

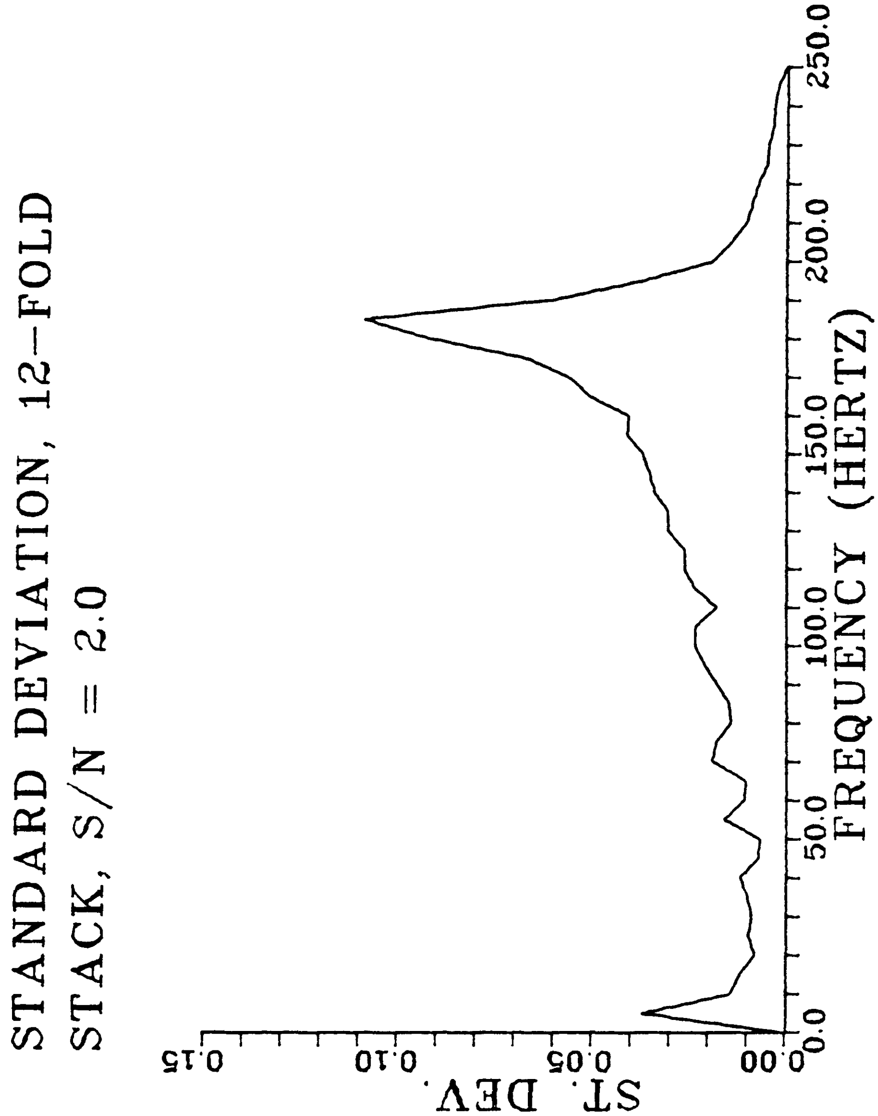

STANDARD DEVIATION, 12-FOLD
STACK, S/N = 2.0
FREQUENCY (HERTZ)
ST. DEV.
0.0
50.0
100.0
150.0
200.0
250.0
0.00
0.05
0.10
0.15

Figure 26

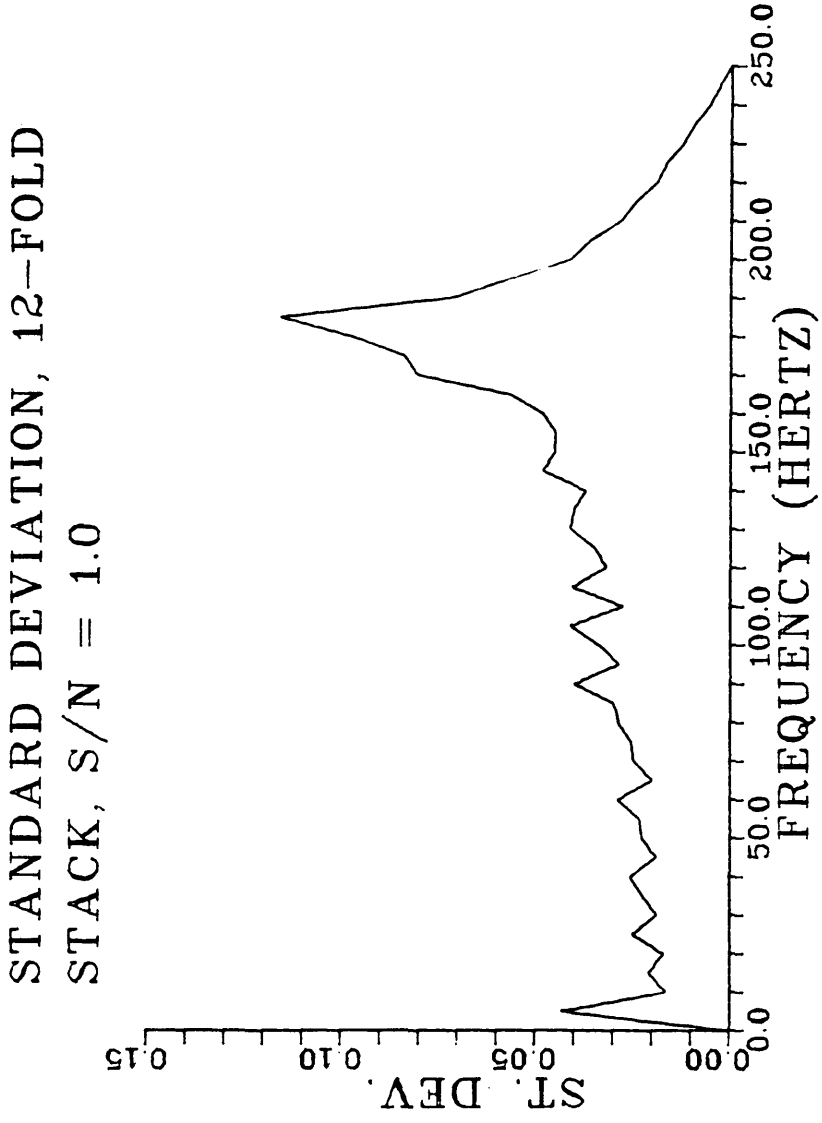

STANDARD DEVIATION, 12—FOLD
STACK, S/N = 1.0
FREQUENCY (HERTZ)
0.0
50.0
100.0
150.0
200.0
250.0
ST. DEV.
0.00
0.05
0.10
0.15

Figure 27

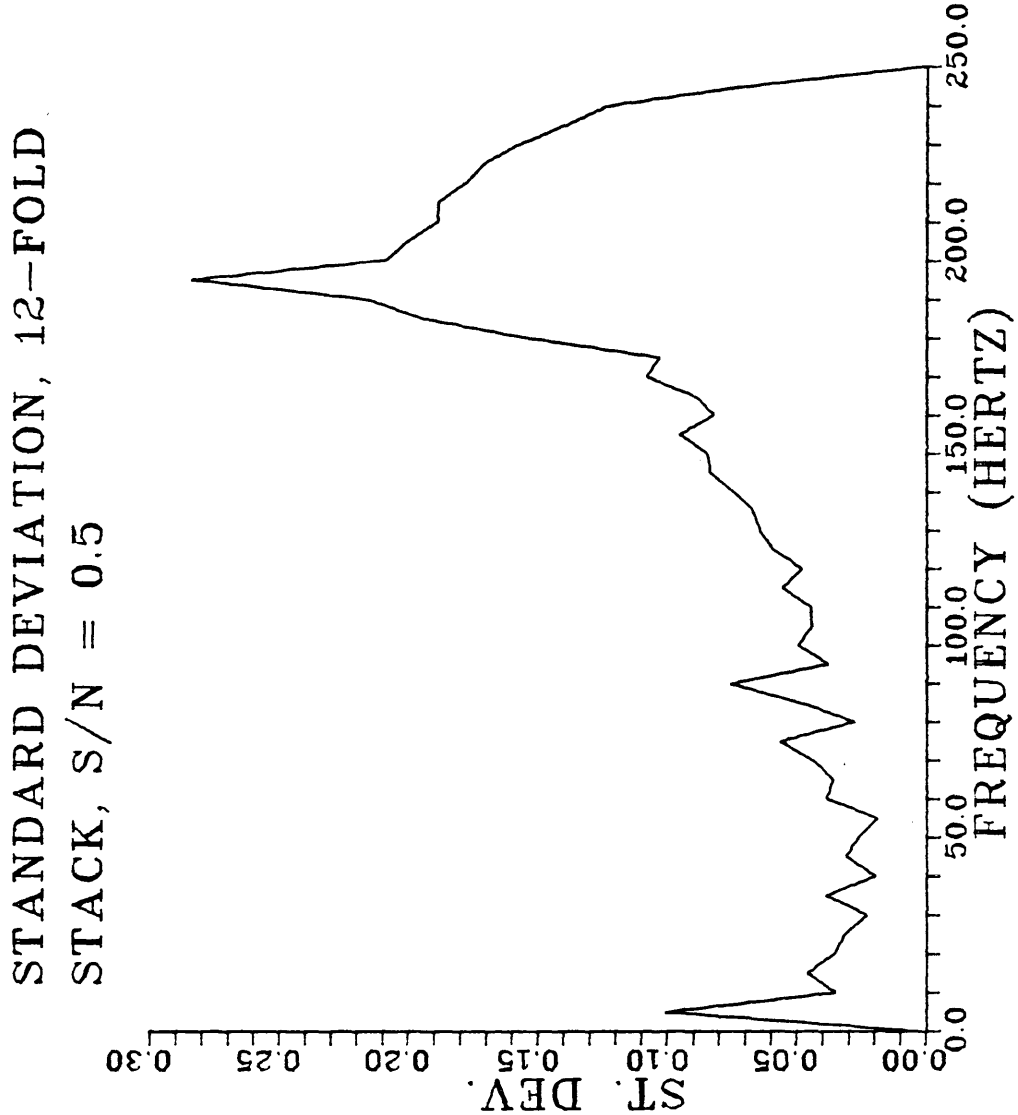

STANDARD DEVIATION, 12—FOLD
STACK, S/N = 0.5
FREQUENCY (HERTZ)
ST. DEV.
0.0
50.0
100.0
150.0
200.0
250.0
0.00
0.05
0.10
0.15
0.20
0.25
0.30

Figure 28

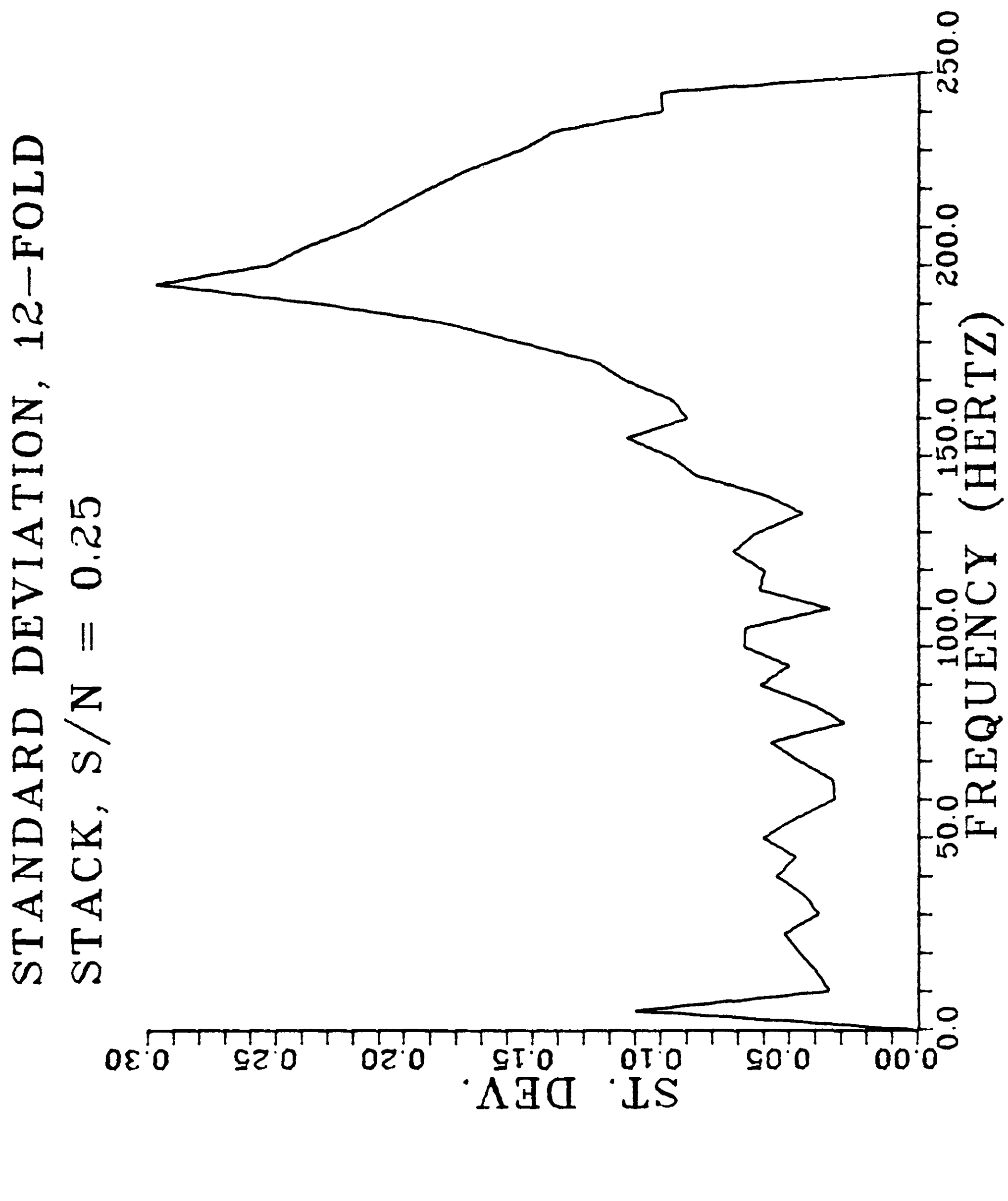

STANDARD DEVIATION, 12—FOLD
STACK, S/N = 0.25
FREQUENCY (HERTZ)
ST. DEV.
0.0
50.0
100.0
150.0
200.0
250.0
0.00
0.05
0.10
0.15
0.20
0.25
0.30

Figure 29

in the level of the curves as the signal-to-noise ratio decreases. This tendency is only slight, however.

The standard deviation in degrees is plotted versus the log of signal-to-noise ratio in Figure 30. The general relationship of increasing standard deviation with increasing frequency and decreasing signal-to-noise is seen. As mentioned before, the dependence is only slight; a maximum deviation of ten degrees arises through a signal-to-noise ratio difference of eighteen dB. Ten degrees of deviation is also the maximum value observed throughout the frequency bandlimits of the model.

This observation implies a loss in amplitude is to be expected when CDP stacking is employed at high frequencies or low signal-to-noise ratios. The peaks of the component wavelets do not align exactly, consequently, the peak of the stacked wavelet is of lesser amplitude. This conclusion is evident to some extent in graphs of the amplitude spectra of the stacked wavelets (Figures 15a, 22a, 23a, and 24a). Although the amplitude throughout the bandpass is fairly uniform at a value of one for the case where the signal to noise ratio equals two (Figure 15a), the amplitude drops slightly at higher frequencies for lesser signal to noise ratios (Figures 22a, 23a, and 24a). Unfortunately, comparison of the four spectra with regard to relative amplitude is meaningless because the computer algorithm used to compute the spectral values normalizes the maximum absolute amplitude to one.

WAVELET MODEL STANDARD DEVIATION CURVES, 30 TO 170 HERTZ

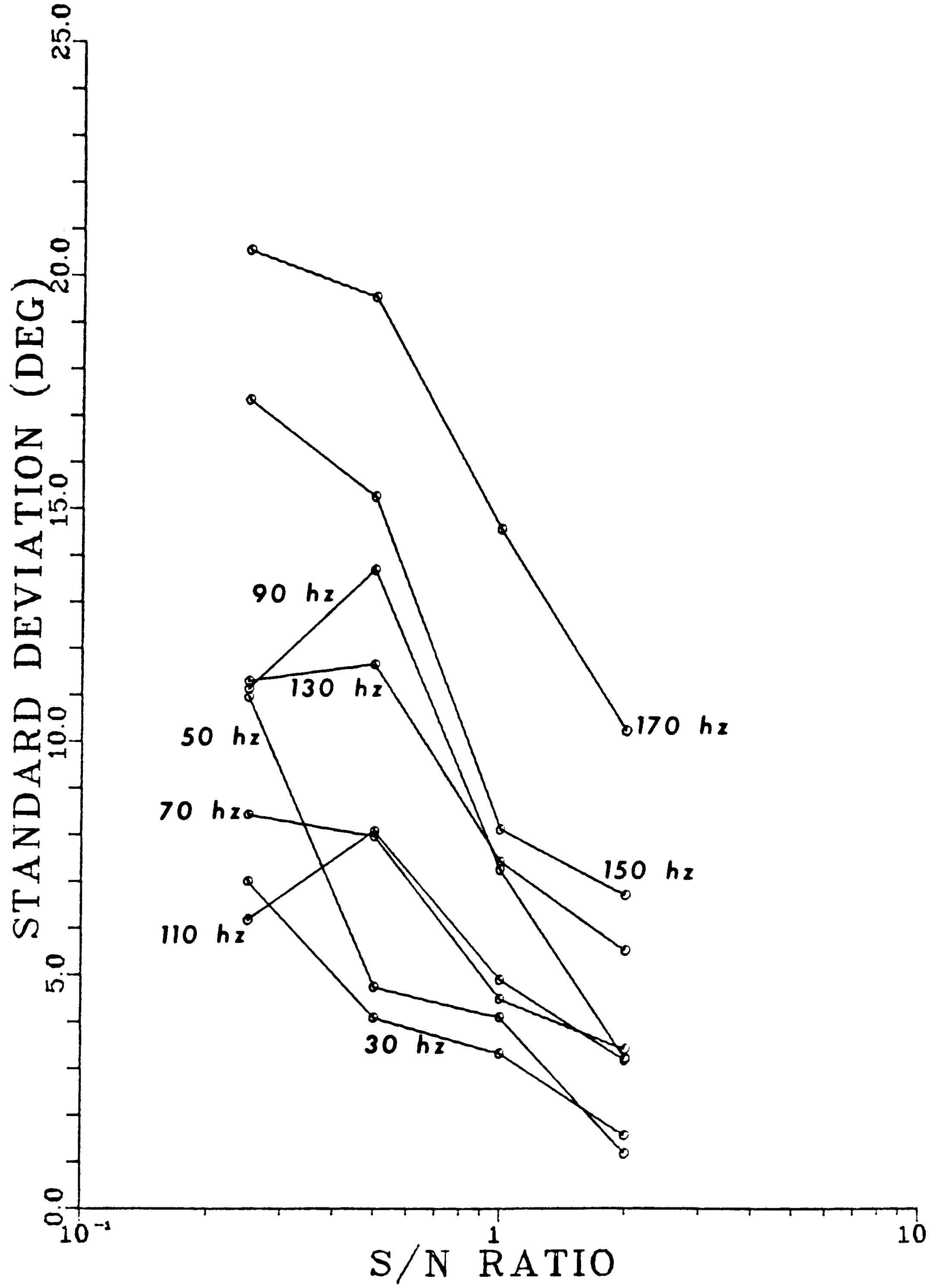


Figure 30

The lack of distortion of the wavelet as a function of signal-to-noise ratio is clearly pointed out in a final example, where a bandlimited noise realization is deconvolved in the presence of no signal. An artificial estimate of a minimum phase wavelet is computed from the colored noise and is deconvolved by the minimum phase inverse of the uncorrupted wavelet. Subsequent zero phase bandpass filtering yields the result shown in Figure 31. Although the signal is very choppy and distorted, the approximation to a zero phase wavelet is as reasonable as for much higher signal-to-noise ratios.

Spiking deconvolution appears to be quite a robust process when the signal and noise have approximately the same spectral characteristics, as exemplified by similar results ranging from signal-to-noise ratios of zero to two. This may in part explain why in extremely noisy situations, the results of one-sided Wiener filtering may be improved if zero phase bandpass filtering is applied before deconvolution (Berkhout, 1977).

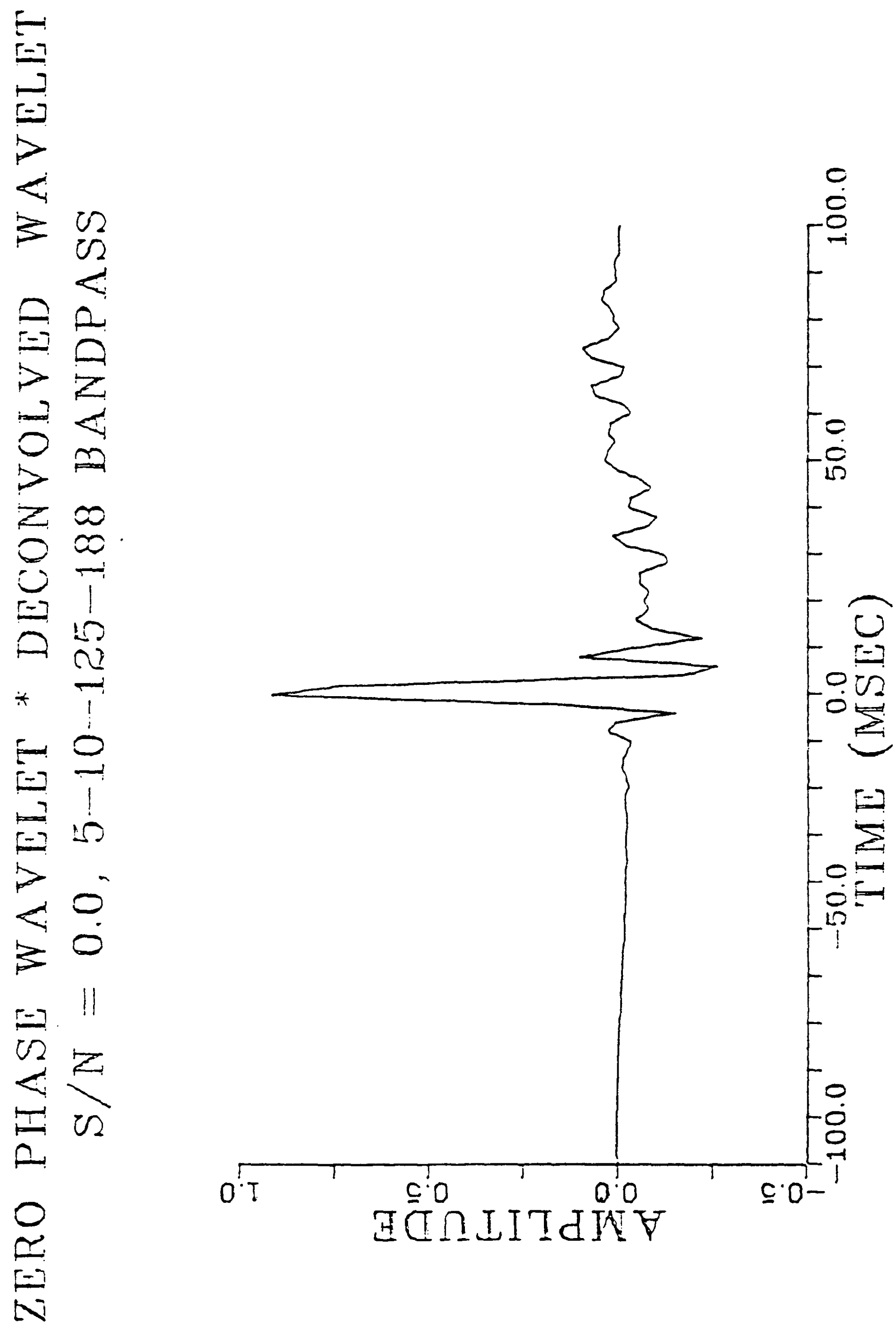

ZERO PHASE WAVELET * DECONVOLVED WAVELET
S/N = 0.0, 5-10-125-188 BANDPASS
AMPLITUDE
1.0
0.5
0.0
-0.5
TIME (MSEC)
-100.0
-50.0
0.0
50.0
100.0

Figure 31

ATTENUATED WAVELET MODEL

A second, more realistic model is studied utilizing an attenuated wavelet whose amplitude is decreased by twelve dB at half the Nyquist frequency, as demonstrated by the spectrum shown in Figure 32. A relative amplitude of 0.25 can be seen at 125 Hertz. The zero phase, attenuated time domain wavelet corresponding to this spectrum is depicted in Figure 33. The sidelobes of the wavelet are greatly decreased, as is the peak amplitude. The peak amplitude of the attenuated wavelet is slightly over 0.3, whereas the relative peak amplitude of the original zero phase wavelet (Figure 2) is about 0.7.

As with the first model, the wavelet is autocorrelated and input to a Wiener-Levinson algorithm to compute both a minimum phase estimate of the wavelet (Figure 34), whose spectral characteristics are shown in Figure 35, and a minimum phase inverse. Deconvolution of the wavelet by its inverse yields, as before, a zero lag spike (Figure 5). Subsequent Ormsby filtering results in the zero phase wavelet pictured originally in Figure 2.

Noise corrupted attenuated minimum phase wavelets are designed for signal-to-noise ratios of 2.0, 1.0, 0.5, and 0.25. The wavelets are deconvolved by the attenuated, uncorrupted minimum phase inverse, and finally bandpass filtered by the spectrum of the unattenuated zero phase wavelet. An ensemble of twelve wavelets are generated

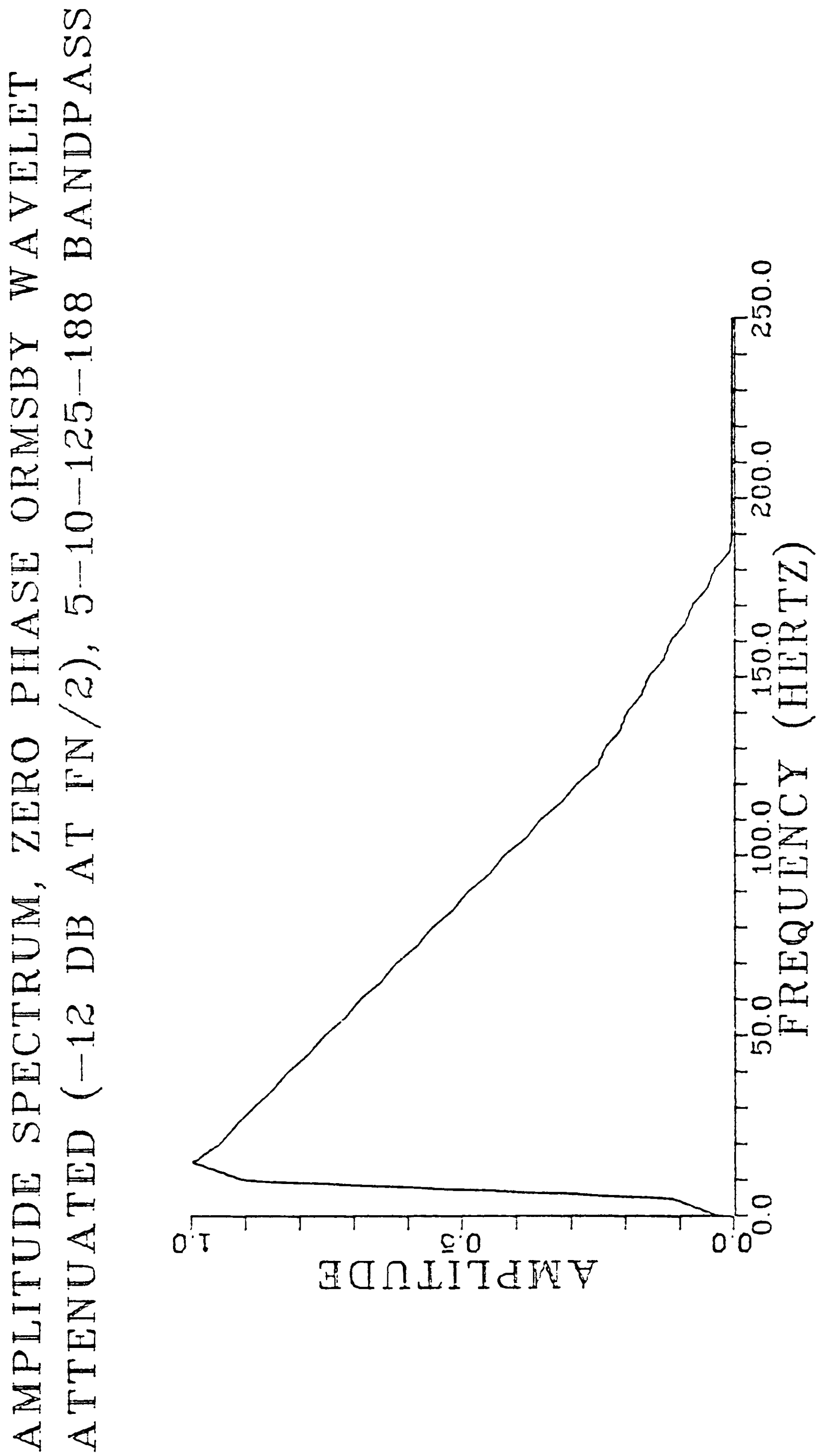

AMPLITUDE SPECTRUM, ZERO PHASE ORMSBY WAVELET
ATTENUATED (−12 DB AT FN/2), 5−10−125−188 BANDPASS
AMPLITUDE
0.0
0.5
1.0
FREQUENCY (HERTZ)
0.0
50.0
100.0
150.0
200.0
250.0

Figure 32

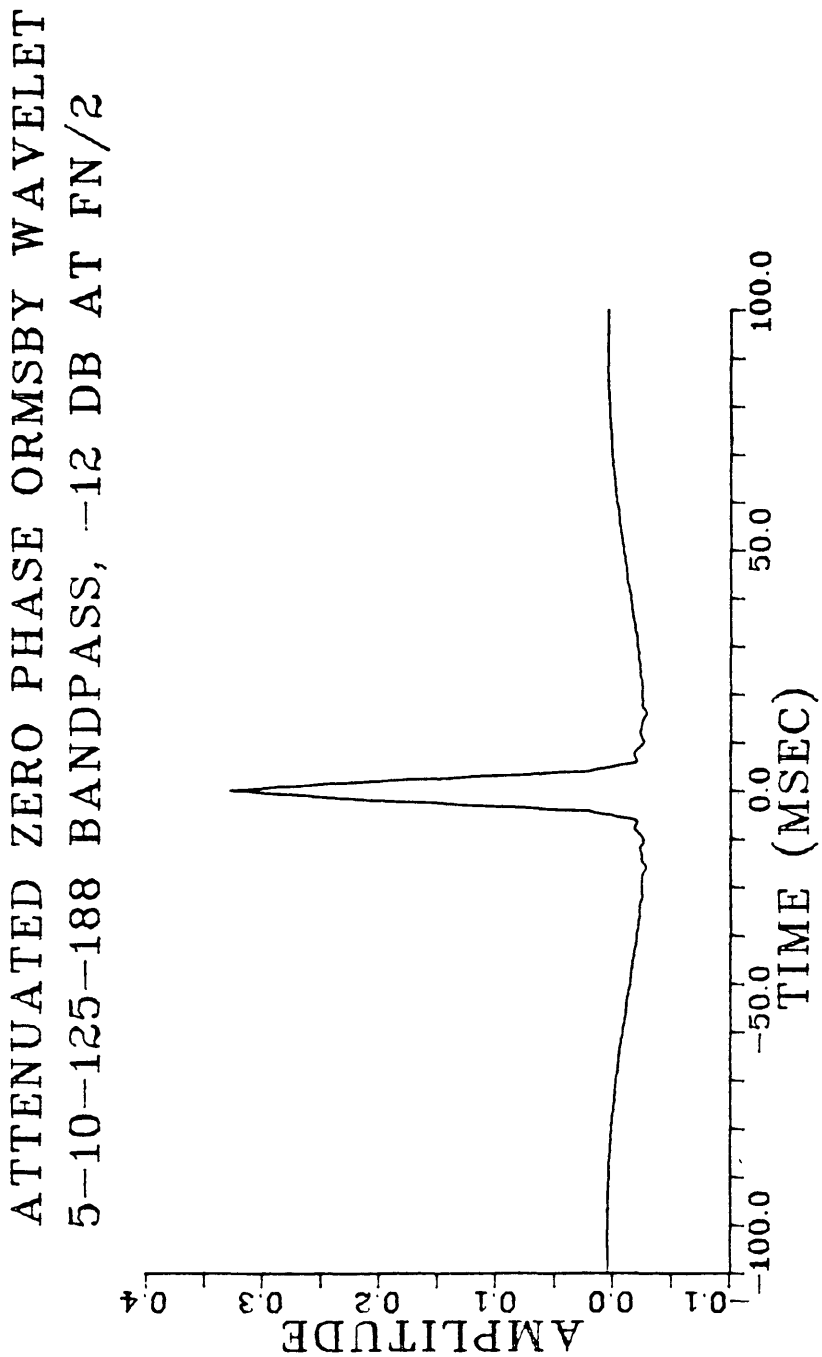

ATTENUATED ZERO PHASE ORMSBY WAVELET
5-10-125-188 BANDPASS, -12 DB AT FN/2
AMPLITUDE
-0.1
0.0
0.1
0.2
0.3
0.4
TIME (MSEC)
-100.0
-50.0
0.0
50.0
100.0

Figure 33

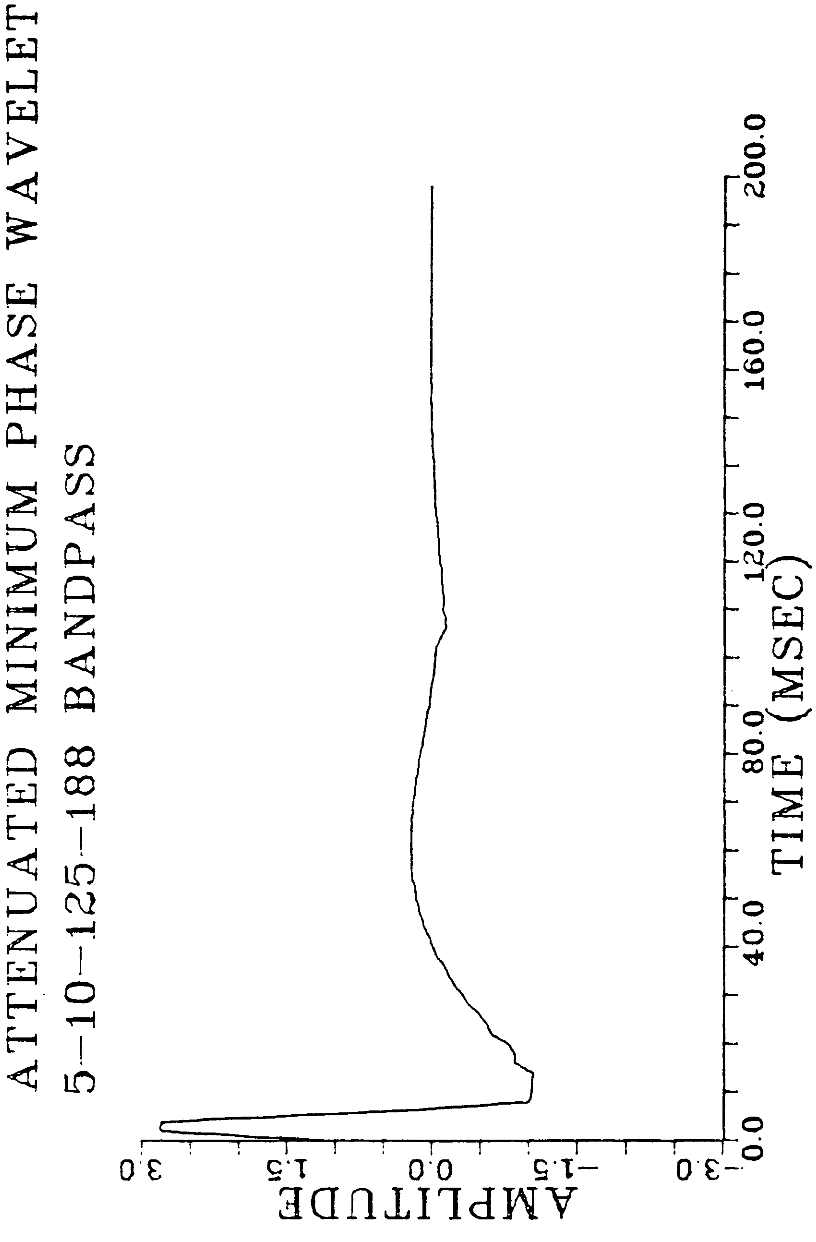
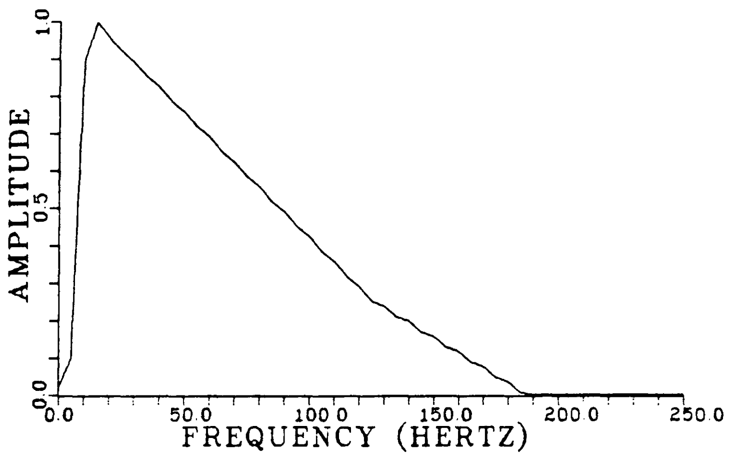

ATTENUATED MINIMUM PHASE WAVELET
5--10--125--188 BANDPASS
TIME (MSEC)
0.0
40.0
80.0
120.0
160.0
200.0
AMPLITUDE
3.0
1.5
0.0
-1.5
-3.0

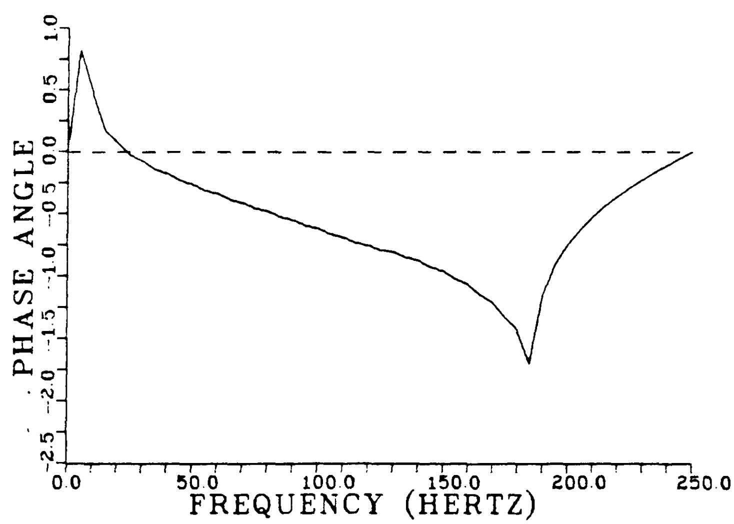
Figure 34

AMPLITUDE SPECTRUM, MINIMUM PHASE WAVELET
ATTENUATED (−12 DB AT FN/2), 5−10−125−188 BANDPASS



(a)

PHASE SPECTRUM, MINIMUM PHASE WAVELET
ATTENUATED (−12 DB AT FN/2), 5−10−125−188 BANDPASS



(b)

Figure 35

for each of the four signal to noise ratios; consistent with the method used in the study of the original wavelet.

Four individually processed component wavelets from their respective twelve-fold ensembles are plotted in Figures 36 through 39. The deconvolution process becomes more reliable as the signal-to-noise ratio increases, as evidenced by comparison of the four figures. Examination of Figure 36 reveals, for a signal-to-noise ratio of two, a wavelet that closely approximates the ideal, uncorrupted result. However, the outcome starts to resemble the attenuated wavelet as the signal-to-noise ratio decreases (Figures 37, 38, and 39). The wavelets also become increasingly choppy, noisy, and erratic.

Computations of filter performance parameters indicate that as the signal-to-noise ratio increases, filter performance also increases. In addition, filter performance dependance upon signal-to-noise ratio appears to be more marked for the attenuated wavelet model than for the wavelet model.

The amplitude spectra of the component wavelets are presented in Figures 40a through 43a; inspection of the curves reinforces the aforementioned conclusion. For a signal-to-noise ratio of two, the spectrum is relatively smooth and white, approaching that of the ideal wavelet. Nevertheless, the spectra becomes choppier and approaches the spectrum of the attenuated wavelet as the signal-to-noise ratio decreases to 0.25, as shown by the sequence of Figures 41a, 42a, and 43a.

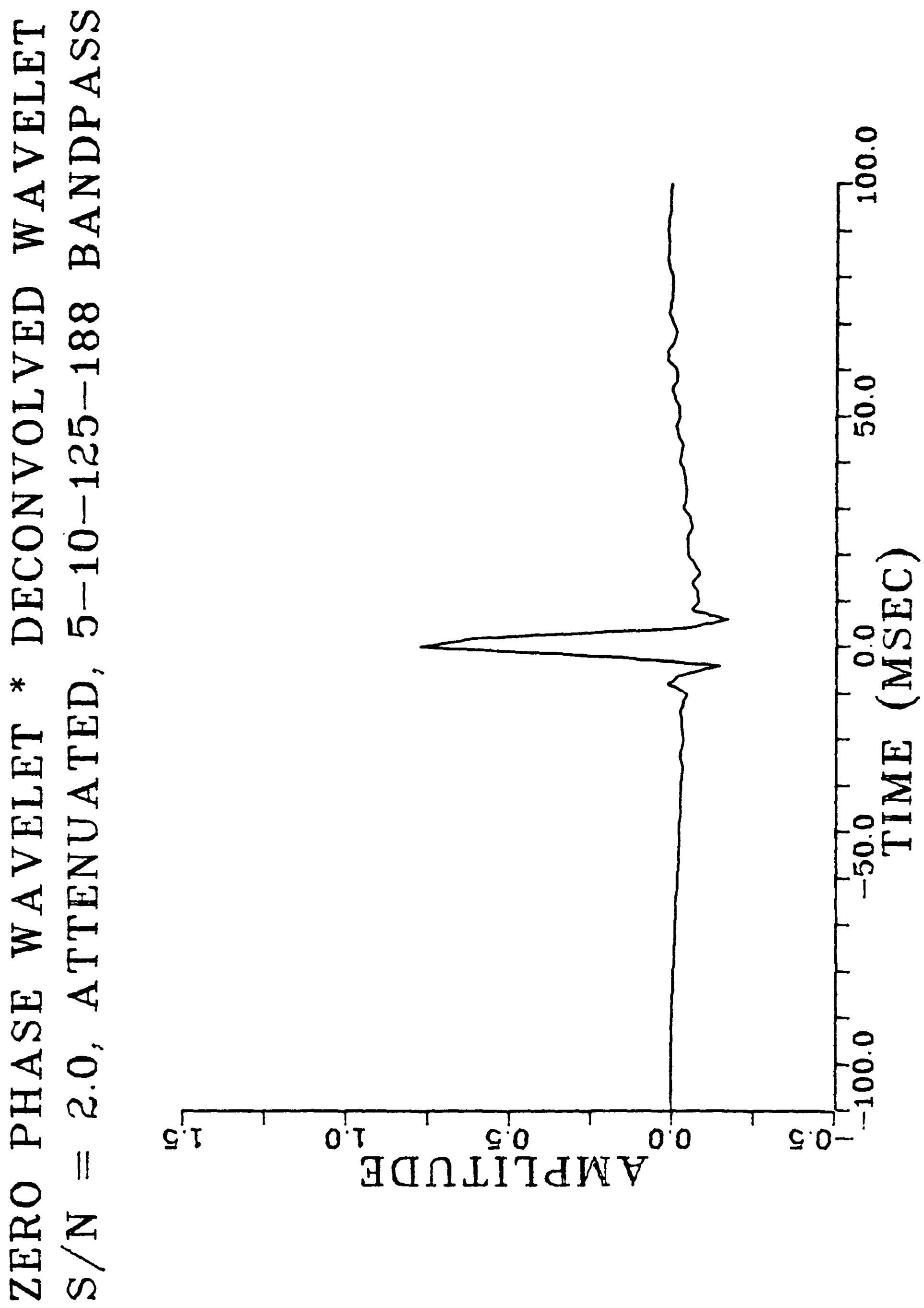

ZERO PHASE WAVELET * DECONVOLVED WAVELET
S/N = 2.0, ATTENUATED, 5-10-125-188 BANDPASS
AMPLITUDE
-0.5
0.0
0.5
1.0
1.5
TIME (MSEC)
-100.0
-50.0
0.0
50.0
100.0

Figure 36

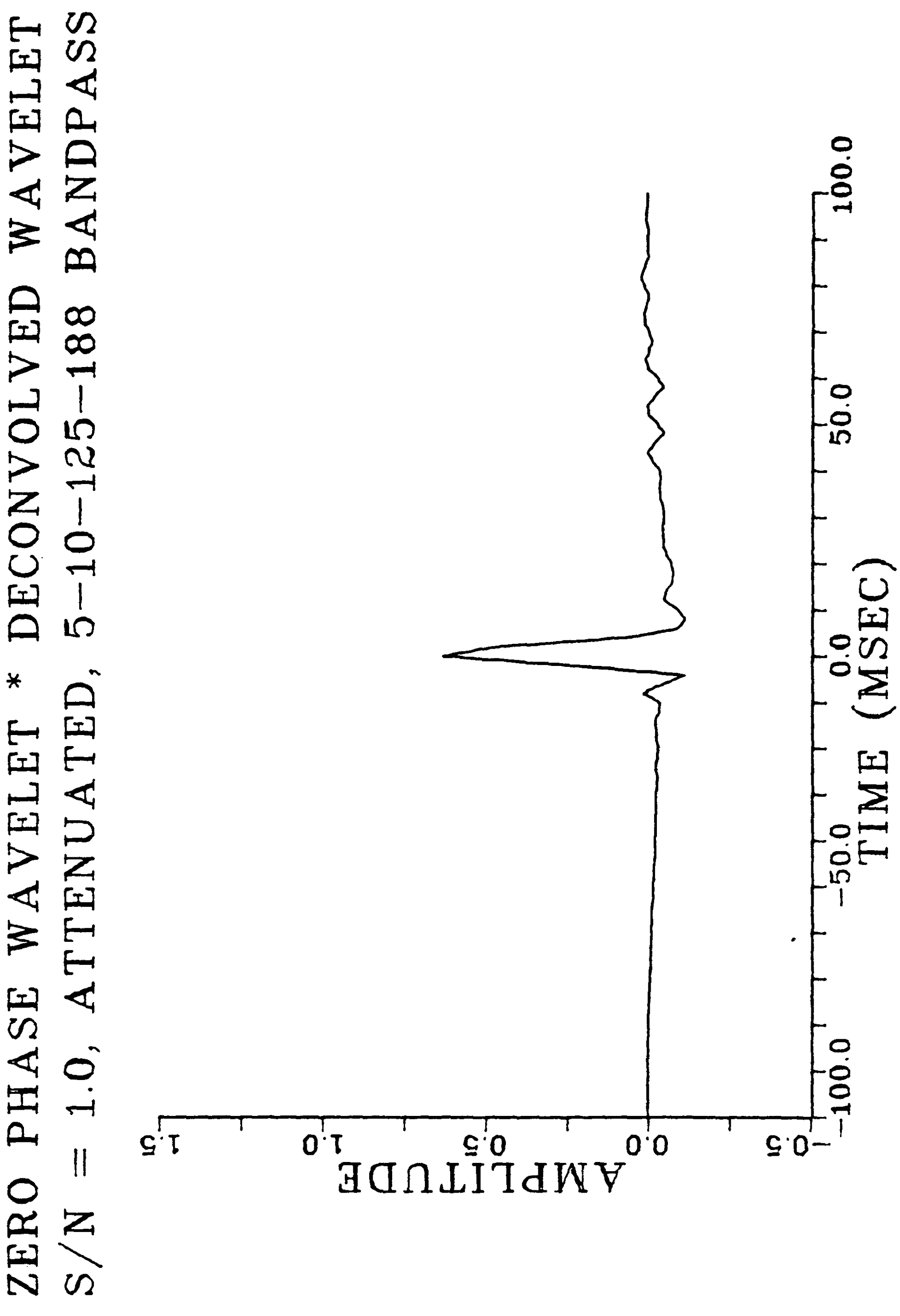

ZERO PHASE WAVELET * DECONVOLVED WAVELET
S/N = 1.0, ATTENUATED, 5–10–125–188 BANDPASS
AMPLITUDE
TIME (MSEC)
-0.5
0.0
0.5
1.0
1.5
-100.0
-50.0
0.0
50.0
100.0

Figure 37

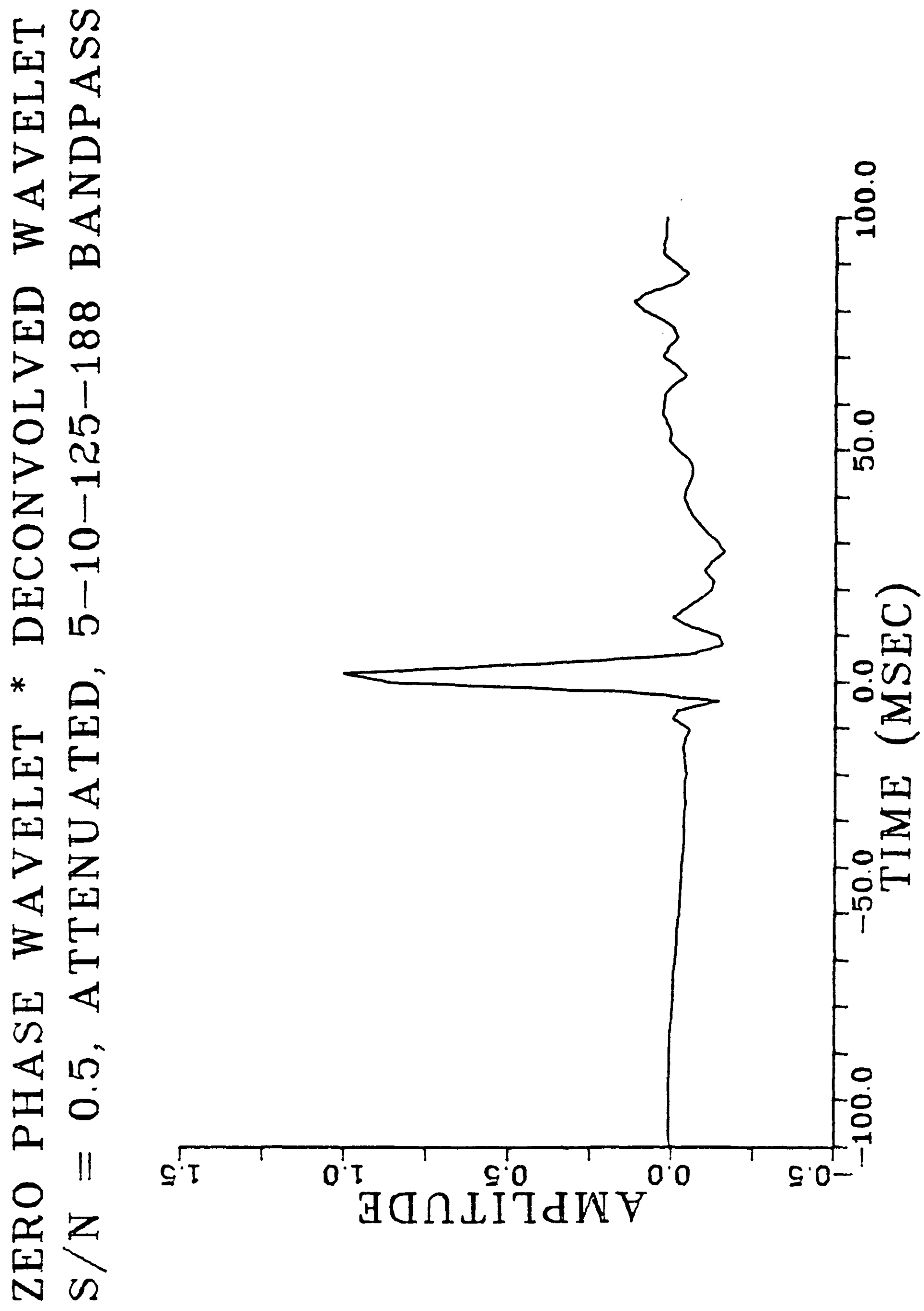

ZERO PHASE WAVELET * DECONVOLVED WAVELET
S/N = 0.5, ATTENUATED, 5-10-125-188 BANDPASS
AMPLITUDE
TIME (MSEC)
-100.0
-50.0
0.0
50.0
100.0
-0.5
0.0
0.5
1.0
1.5

Figure 38

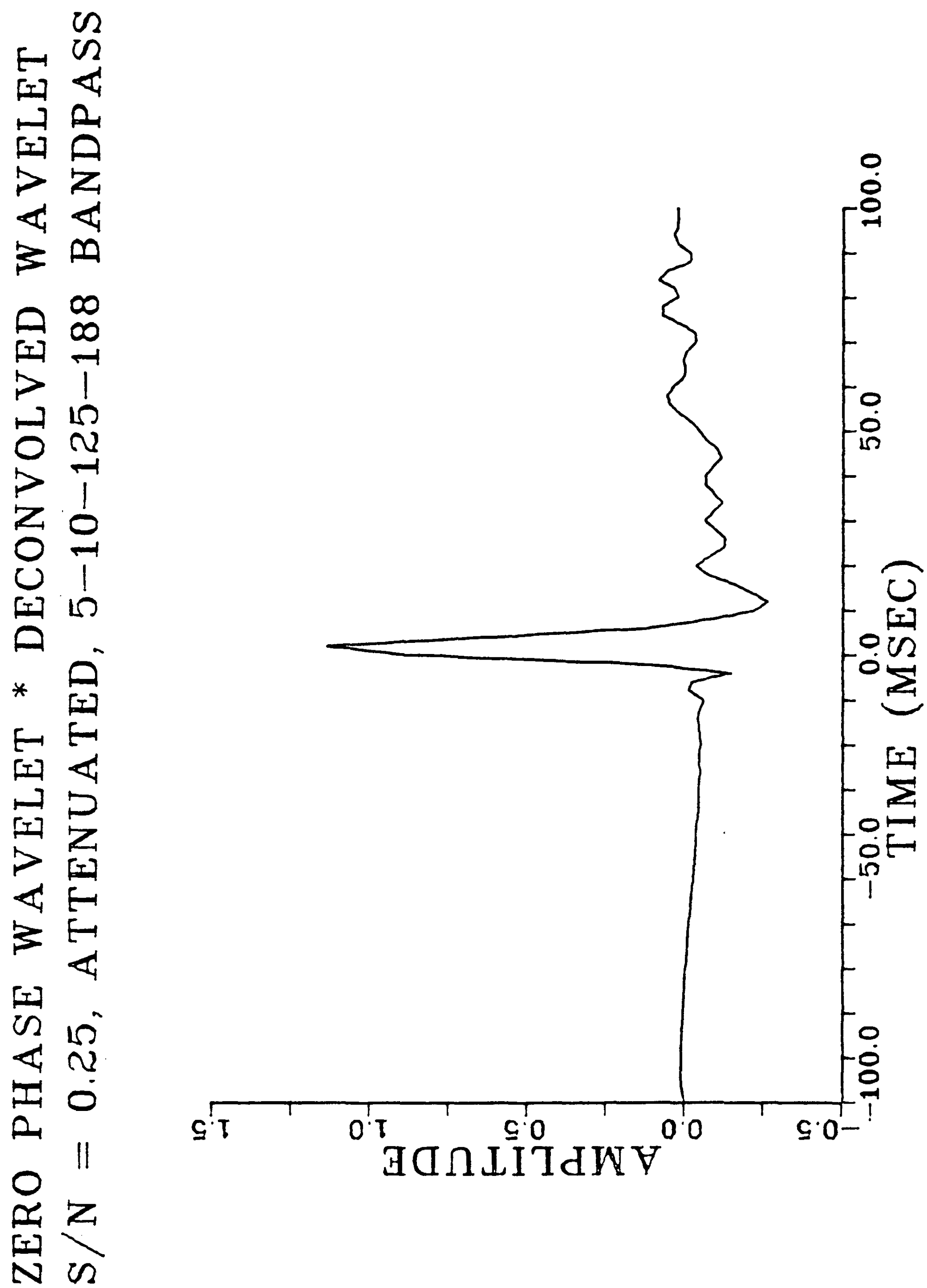
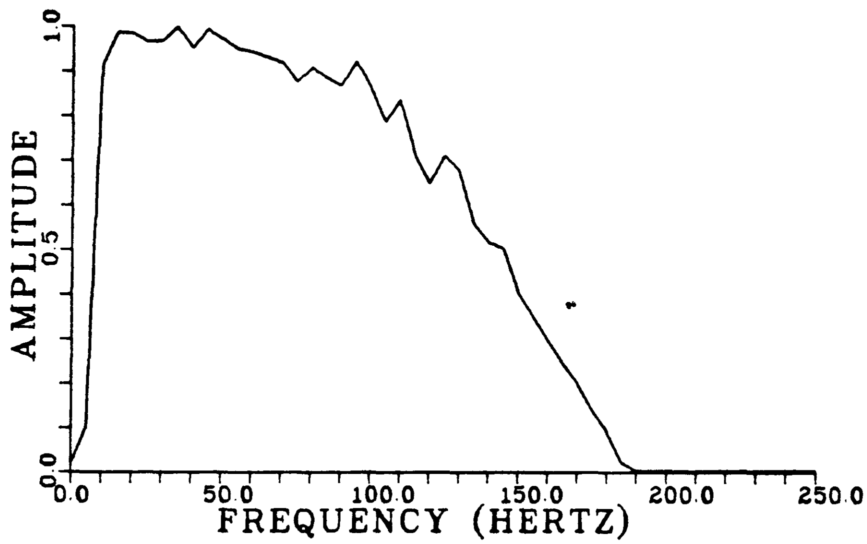

ZERO PHASE WAVELET * DECONVOLVED WAVELET
S/N = 0.25, ATTENUATED, 5–10–125–188 BANDPASS
AMPLITUDE
-0.5
0.0
0.5
1.0
1.5
TIME (MSEC)
-100.0
-50.0
0.0
50.0
100.0

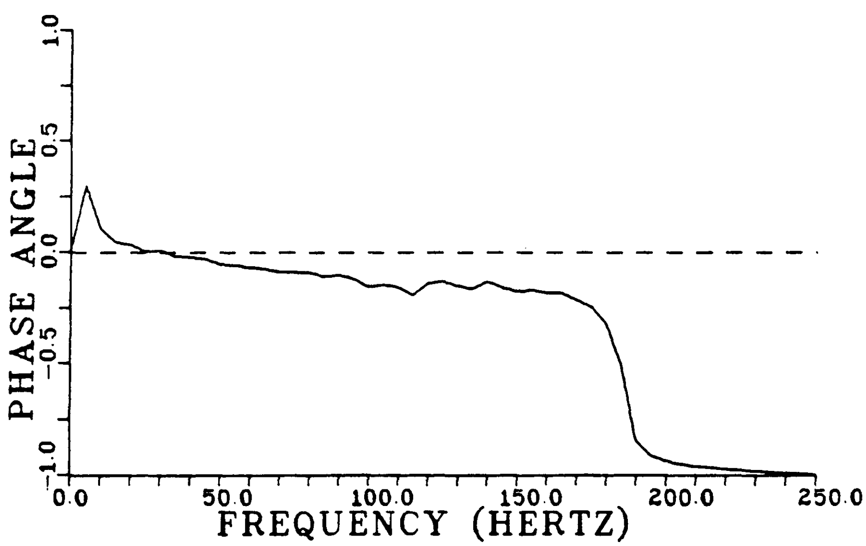
Figure 39

AMPLITUDE SPECTRUM, PROCESSED ATTENUATED WAVELET NO. 1, S/N = 2.0



(a)

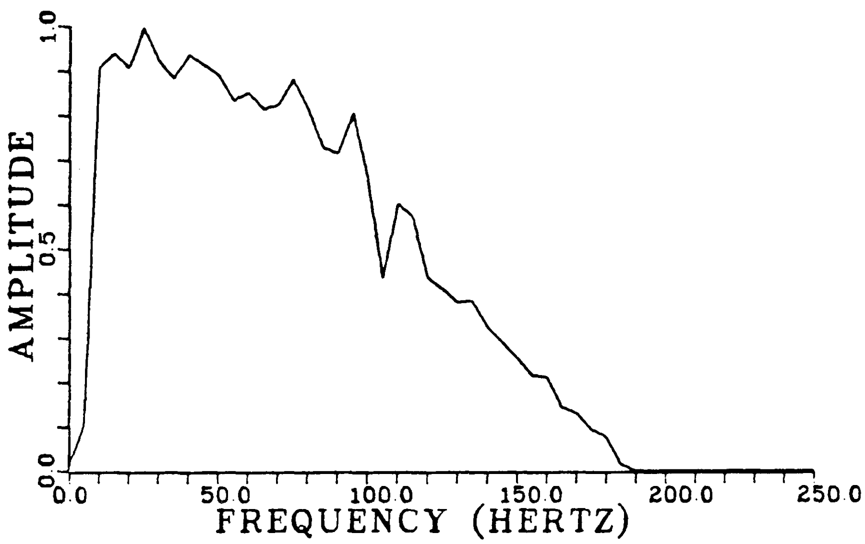
PHASE SPECTRUM, PROCESSED ATTENUATED WAVELET NO. 1, S/N = 2.0



(b)

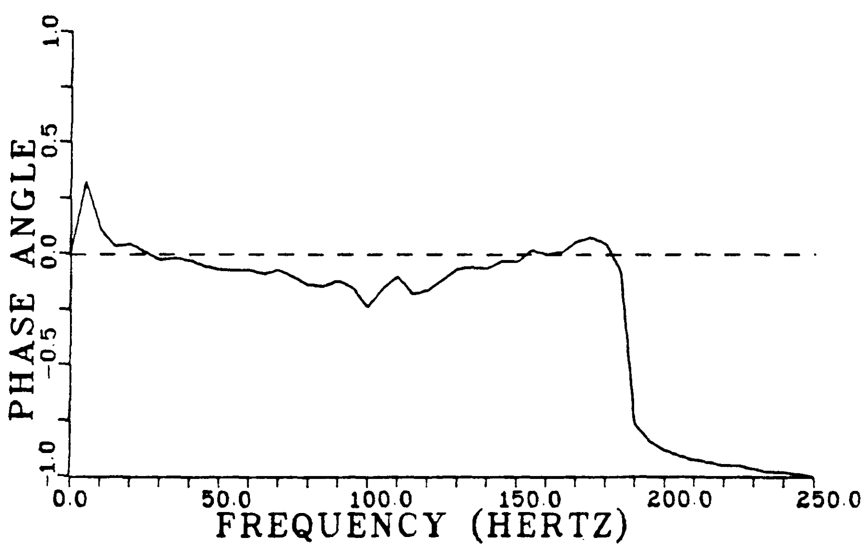
Figure 40

AMPLITUDE SPECTRUM, PROCESSED ATTENUATED WAVELET NO. 1, S/N = 1.0



(a)

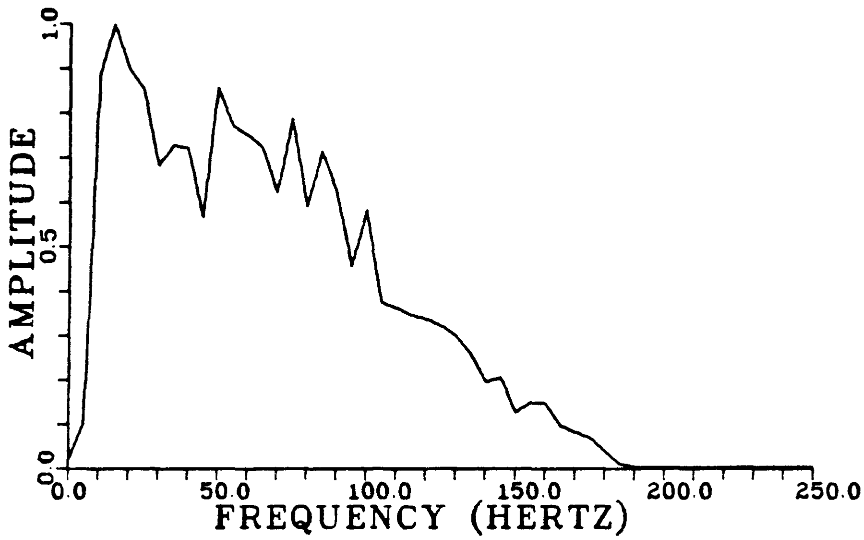
PHASE SPECTRUM, PROCESSED ATTENUATED WAVELET NO. 1, S/N = 1.0



(b)

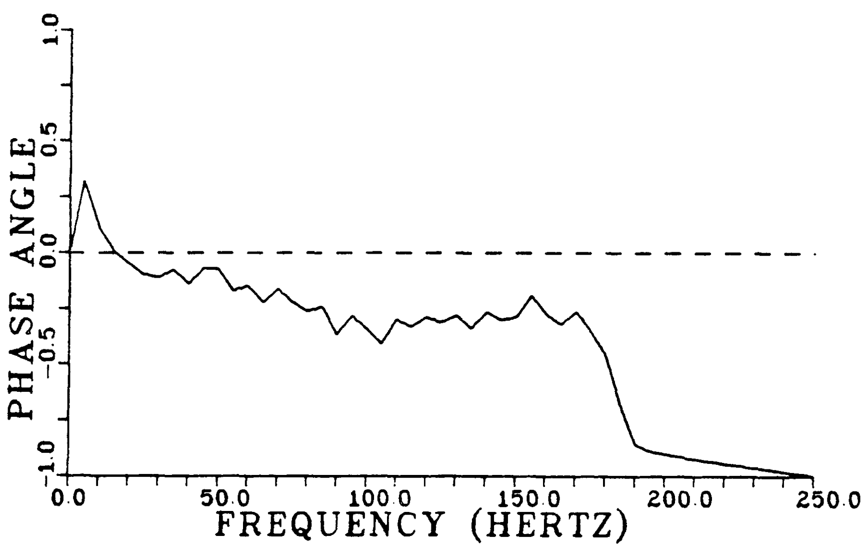
Figure 41

AMPLITUDE SPECTRUM, PROCESSED ATTENUATED WAVELET NO. 1, S/N = 0.5



(a)

PHASE SPECTRUM, PROCESSED ATTENUATED WAVELET NO. 1, S/N = 0.5



(b)

Figure 42

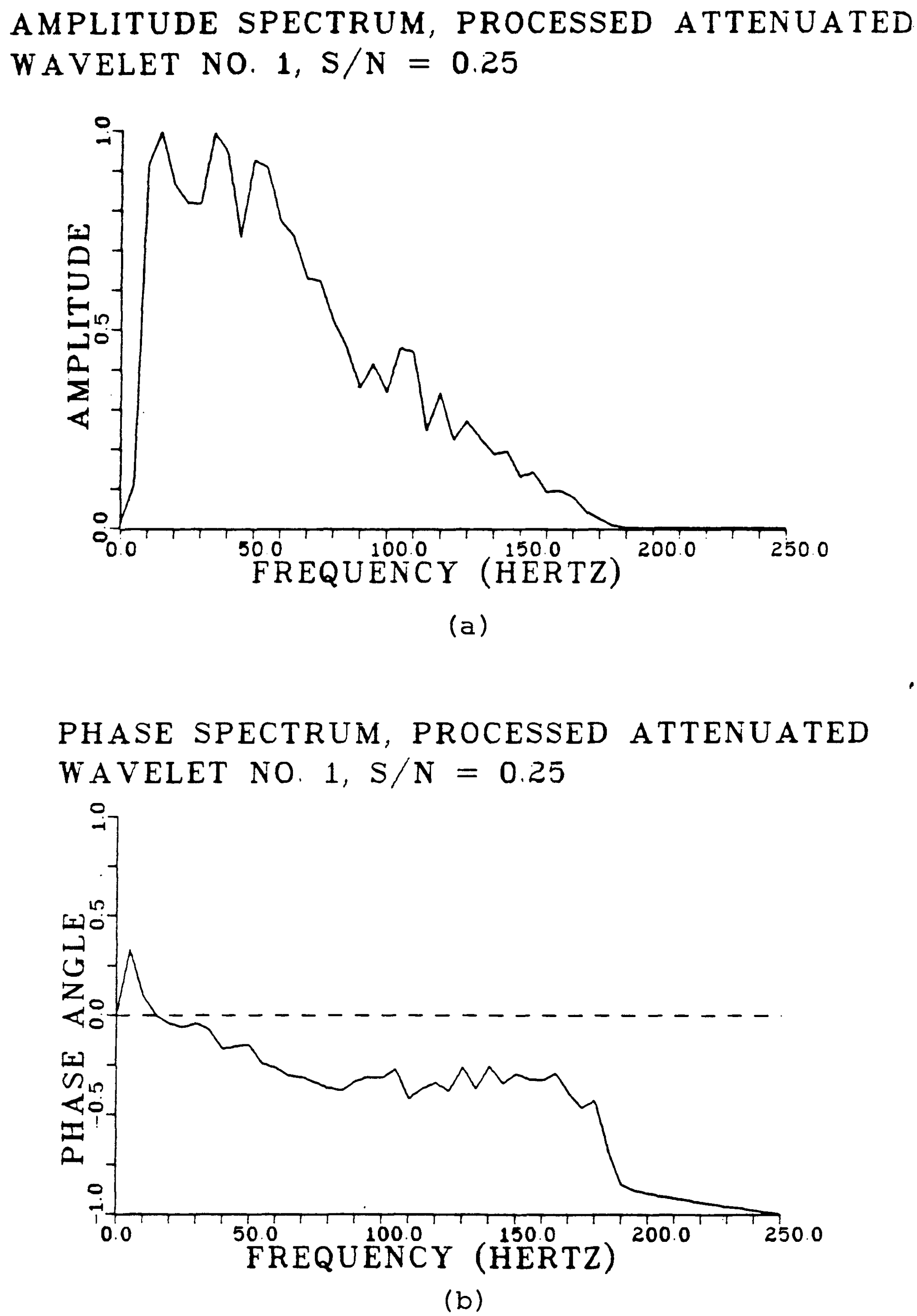

AMPLITUDE SPECTRUM, PROCESSED ATTENUATED
WAVELET NO. 1, S/N = 0.25
AMPLITUDE
0.0
0.5
1.0
0.0
50.0
100.0
150.0
200.0
250.0
FREQUENCY (HERTZ)
(a)
PHASE SPECTRUM, PROCESSED ATTENUATED
WAVELET NO. 1, S/N = 0.25
PHASE ANGLE
-1.0
-0.5
0.0
0.5
1.0
0.0
50.0
100.0
150.0
200.0
250.0
FREQUENCY (HERTZ)
(b)

Figure 43

Figures 40b through 43b picture the phase spectra of the wavelets. The spectrum of the first example (Figure 40b) is reasonably smooth, minimum phase, and shows a linear phase shift of -0.35 degrees/Hertz throughout the bandpass. Slope y-intercept analysis of the curve results in a fifteen degree wavelet delayed one millisecond. When the signal-to-noise ratio decreases to one, the spectrum becomes more erratic, and the linear phase shift increases to -0.55 degrees/Hertz, corresponding to a twenty-two degree wavelet with a two millisecond delay (Figure 41b). The phase spectrum for a signal-to-noise ratio of 0.5 (Figure 42b) exhibits a phase shift of -0.6 degrees/Hertz, implying once again an approximate phase shift of twenty-three degrees, with a time lag of two milliseconds. The worst case, at a signal-to-noise ratio of 0.25, is exemplified by a phase spectrum (Figure 43b) with a linear phase shift of -0.8 degrees/Hertz. This translates to a twenty-five degree wavelet delayed approximately two milliseconds in time.

Initial inspection of deconvolution results from this model indicate the phase error is a function of not only frequency, as with the first model, but also of signal-to-noise ratio. As the signal-to-noise ratio decreases, the phase error increases.

To further study the phase error, each ensemble of twelve wavelets is averaged, as before, to simulate common depth point stacking. The twelve component wavelets, their stacked equivalent, and an ideal, uncorrupted wavelet are graphed for each signal-to-noise ratio. Figures 44 through 47 depict the results.

Figures 48 and 49 illustrate an enhanced stacked wavelet and its associated spectra for a pre-stack signal-to-noise ratio of two. Neither the wavelet nor its spectral characteristics differ drastically from the result demonstrated by the pre-stack component (Figures 36 and 40). The wavelet, amplitude spectral, and phase spectral curves are smoother than their respective unstacked equivalents, although the averaging procedure retains the overall shape of all three curves. Also, the phase angle values from the stacked wavelet (Figure 49b) are in general slightly less than the phase angle values from the unstacked wavelet (Figure 40b).

This observation can be generalized to wavelet and spectral diagrams calculated at lesser signal-to-noise ratios, in particular 1.0, 0.5, and 0.25. Respective comparison of Figures 50 and 51 with Figures 37 and 41 demonstrate that CDP stacking suppresses choppy, irregular noise while preserving overall curve shape, for a signal-to-noise ratio of one. For signal-to-noise ratios of 0.5 and 0.25, examination of Figures 52 and 53 versus Figures 38 and 42 and Figures 54 and 55 versus Figures 39 and 43, respectively, implies the same result. Also, at lower signal-to-noise ratios, observation of a slight overall decrease in phase error in the stacked wavelets versus their unstacked counterparts is consistent with the result described for a signal-to-noise ratio of two.

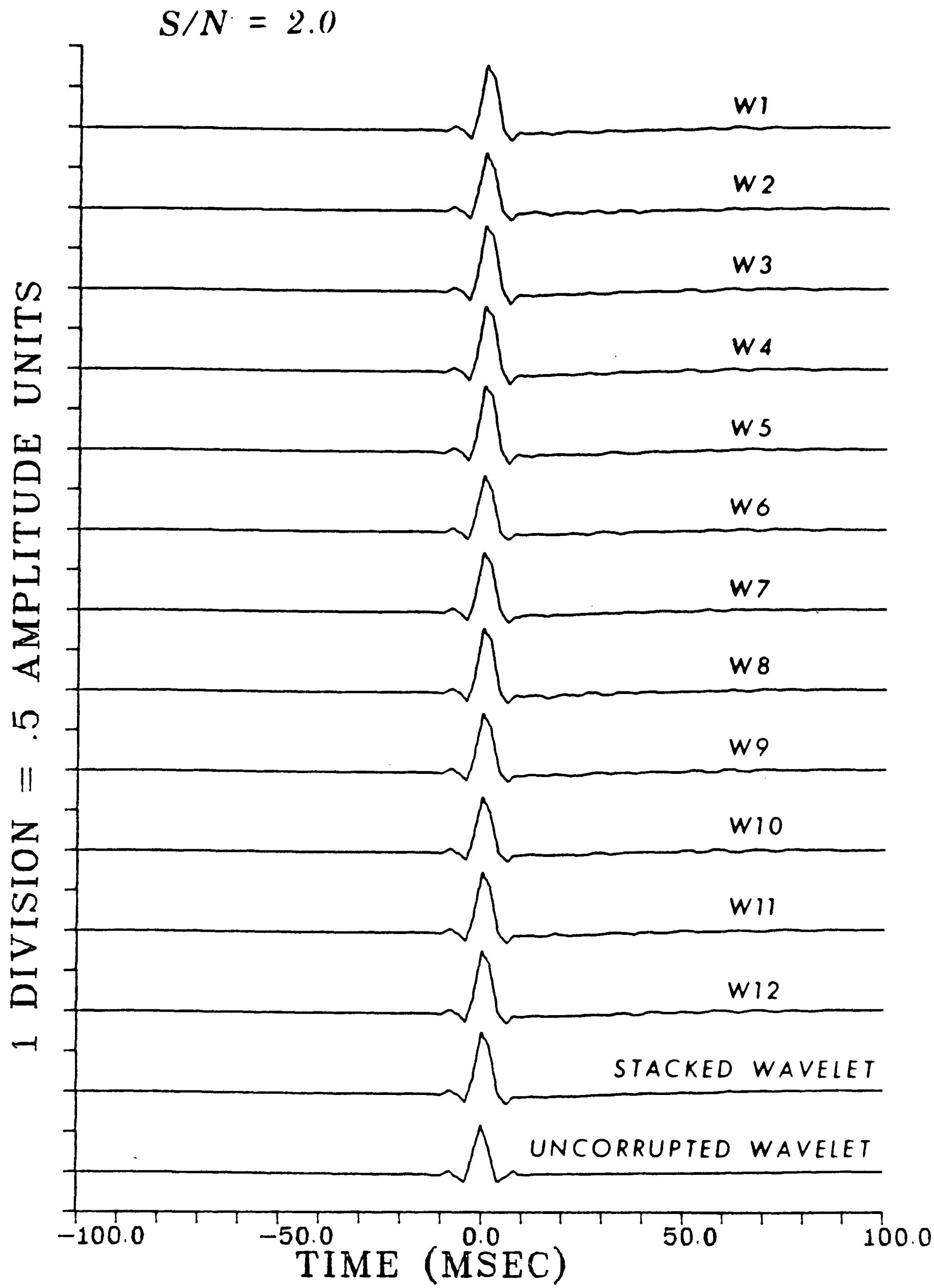


Figure 44 - Comparison of ideal vs. noise corrupted results, attenuated wavelet model

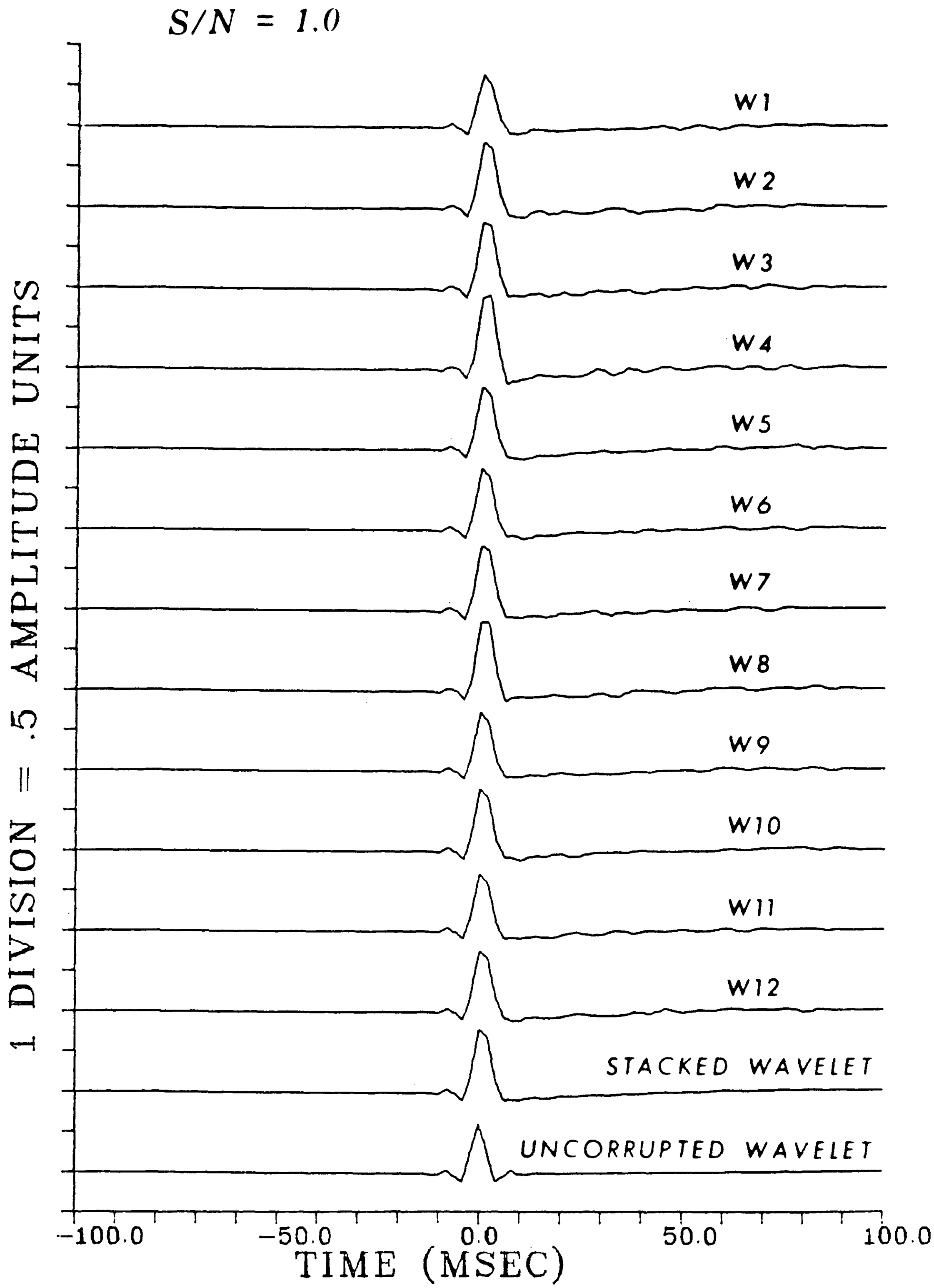


Figure 45 - Comparison of ideal vs. noise corrupted results, attenuated wavelet model

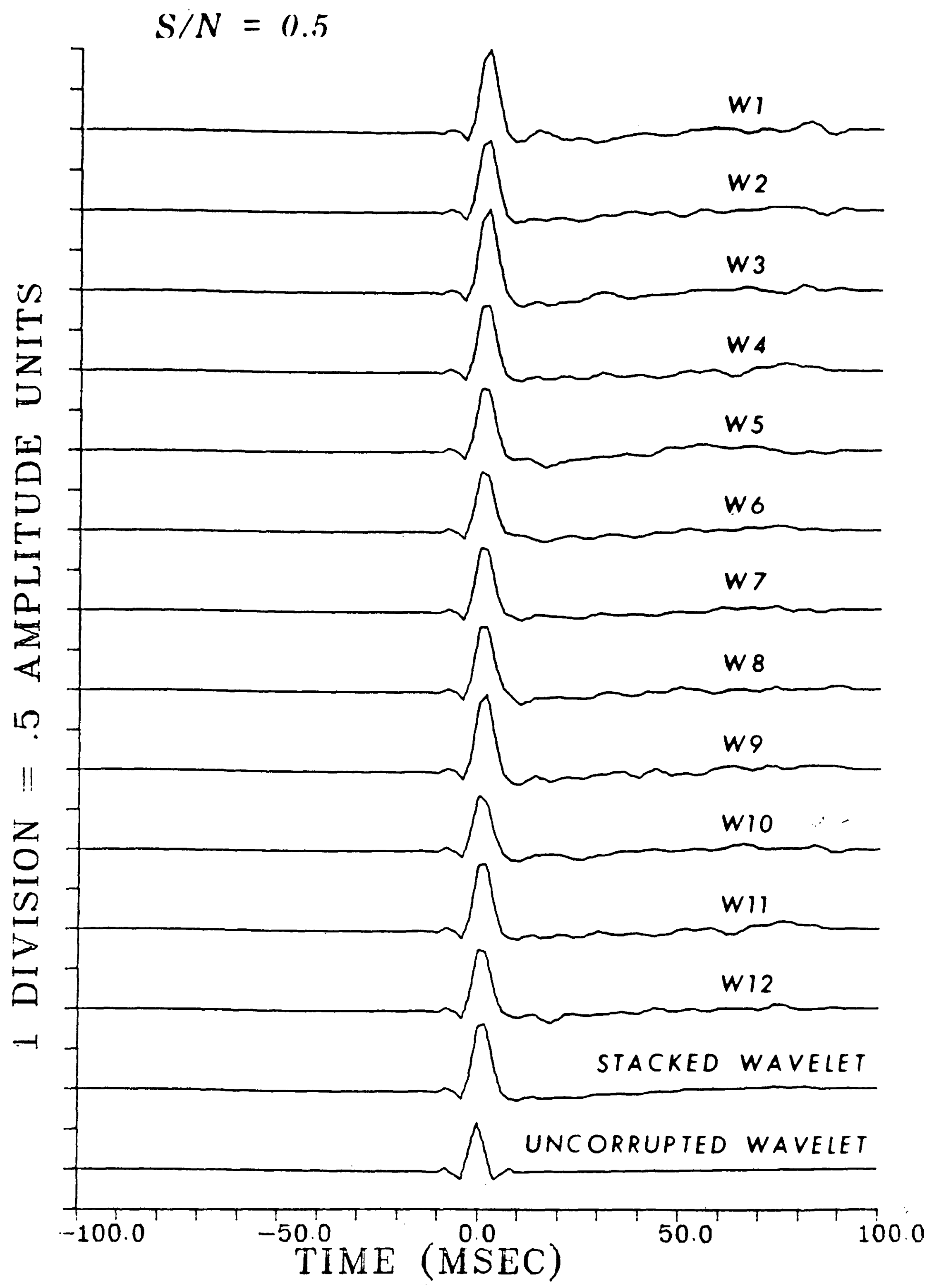


Figure 46 - Comparison of ideal vs. noise corrupted results, attenuated wavelet model

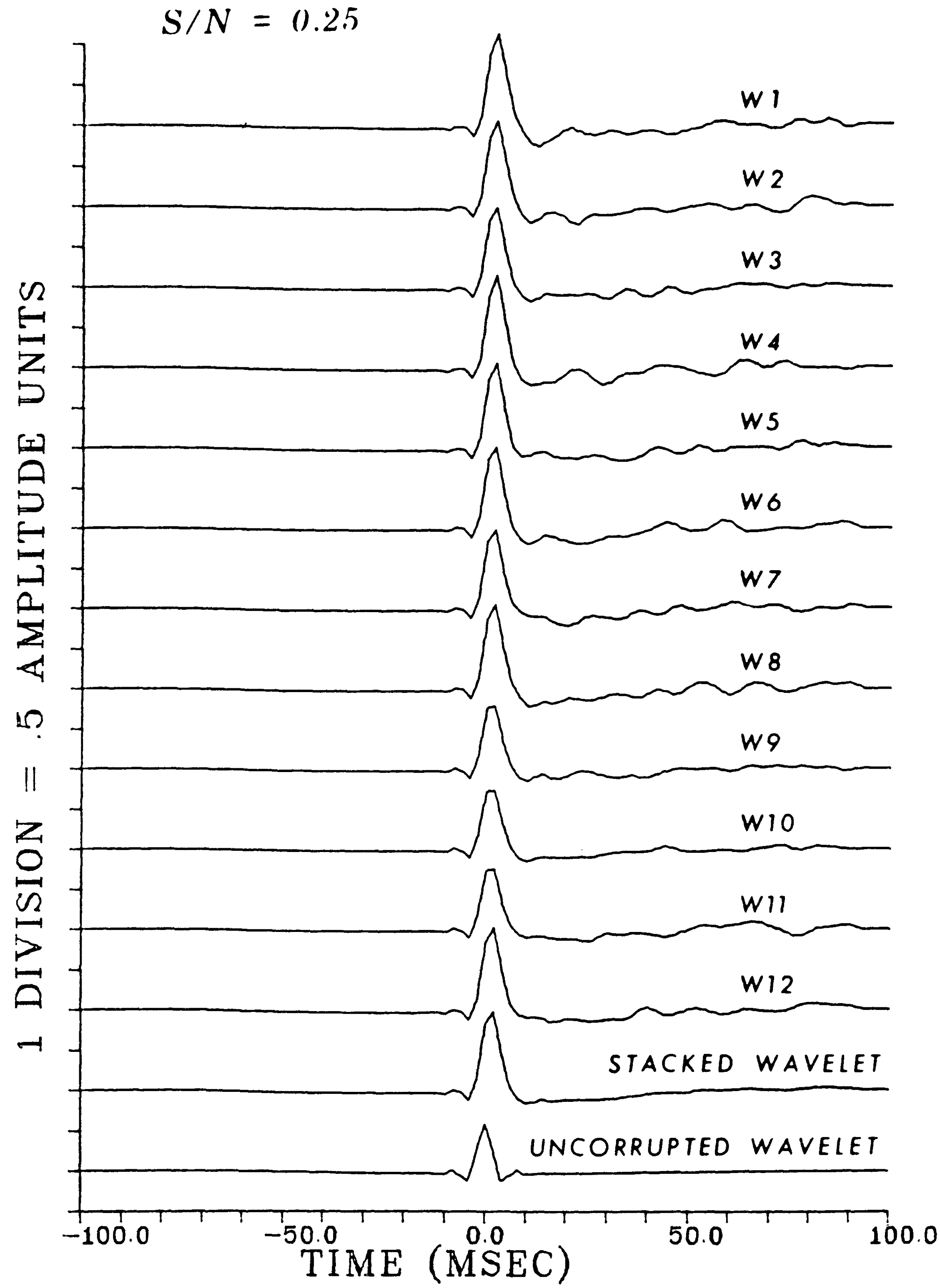


Figure 47 - Comparison of ideal vs. noise corrupted results, attenuated wavelet model

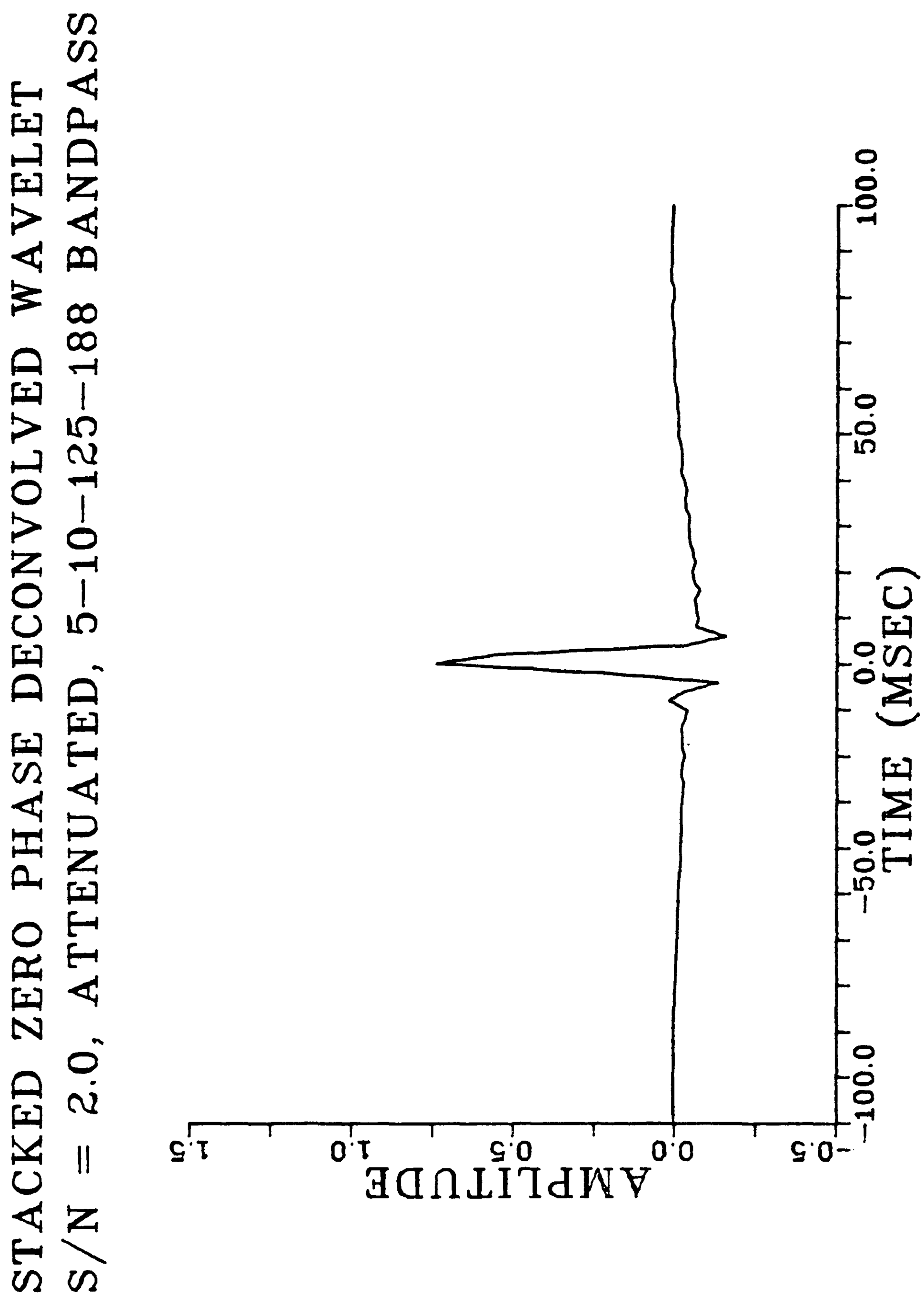
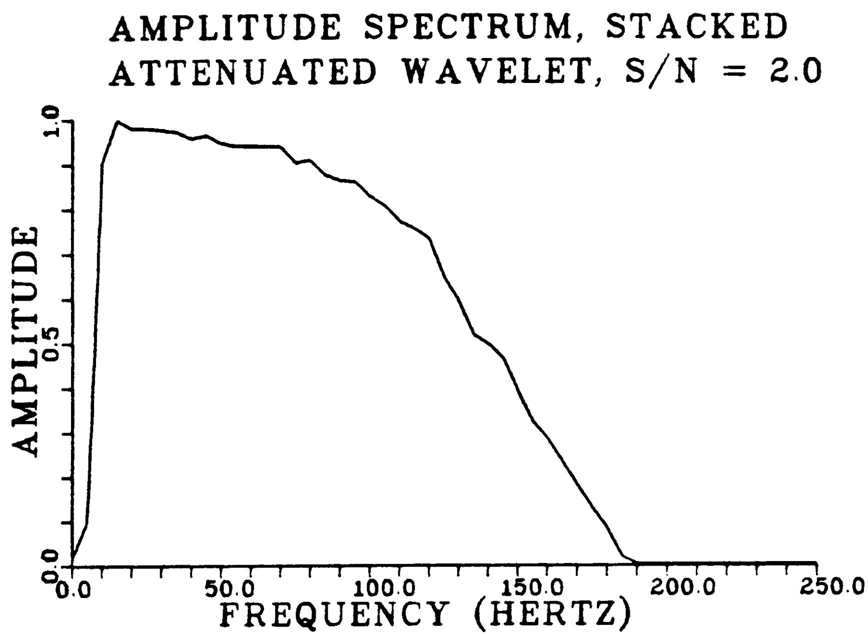
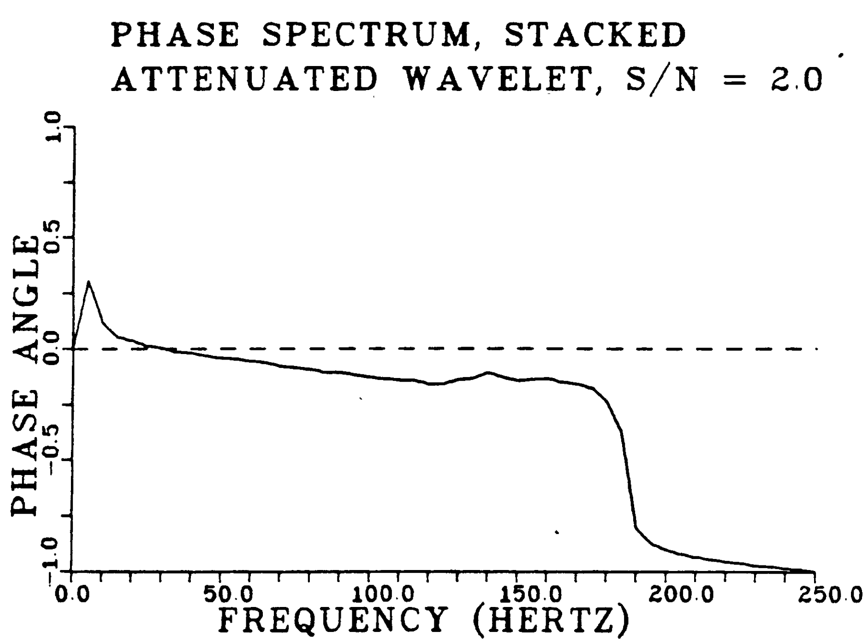

STACKED ZERO PHASE DECONVOLVED WAVELET
S/N = 2.0, ATTENUATED, 5—10—125—188 BANDPASS
AMPLITUDE
-0.5
0.0
0.5
1.0
1.5
TIME (MSEC)
-100.0
-50.0
0.0
50.0
100.0

Figure 48


AMPLITUDE SPECTRUM, STACKED
ATTENUATED WAVELET, S/N = 2.0
AMPLITUDE
0.0
0.5
1.0
0.0
50.0
100.0
150.0
200.0
250.0
FREQUENCY (HERTZ)

(a)


PHASE SPECTRUM, STACKED
ATTENUATED WAVELET, S/N = 2.0
PHASE ANGLE
-1.0
-0.5
0.0
0.5
1.0
0.0
50.0
100.0
150.0
200.0
250.0
FREQUENCY (HERTZ)

(b)

Figure 49

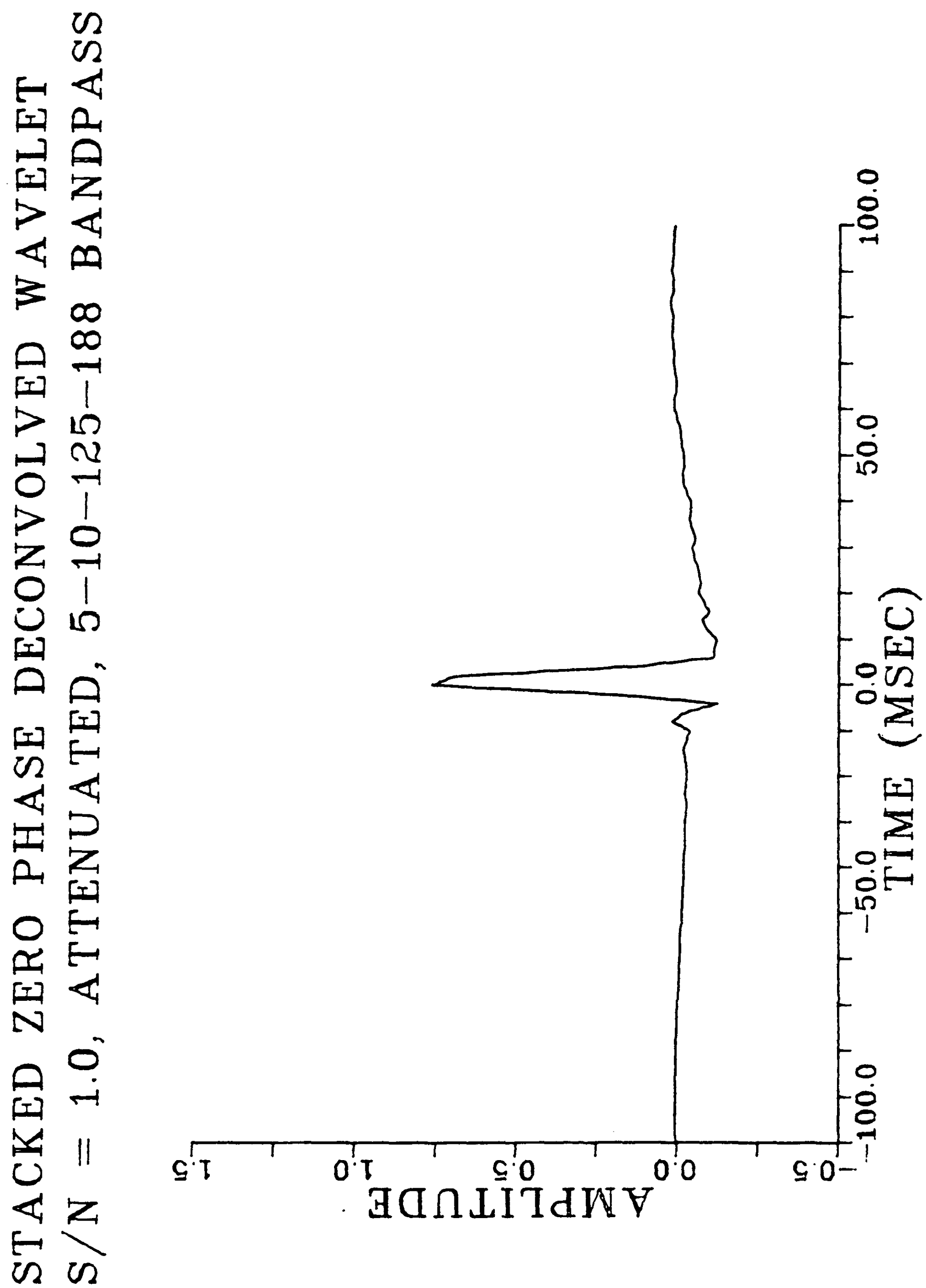
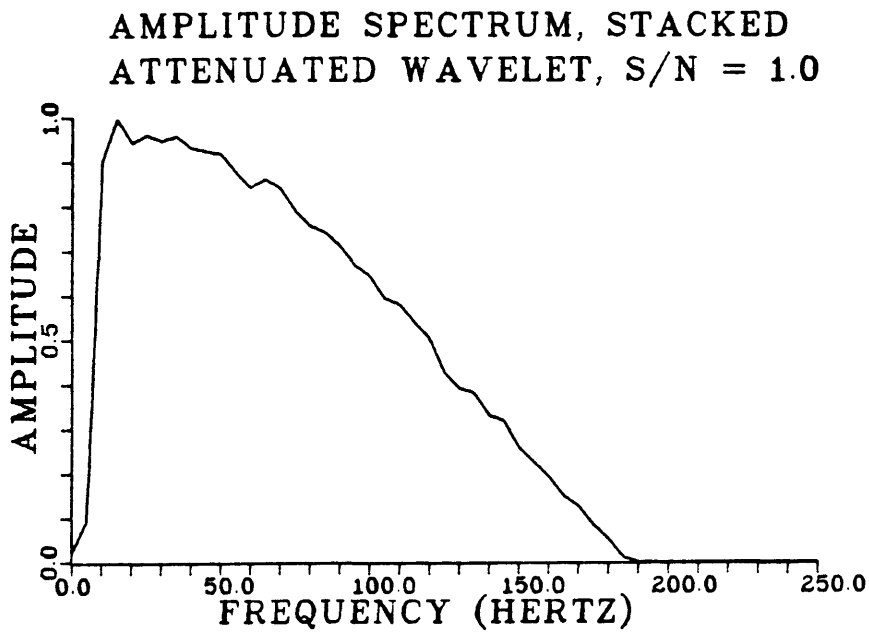
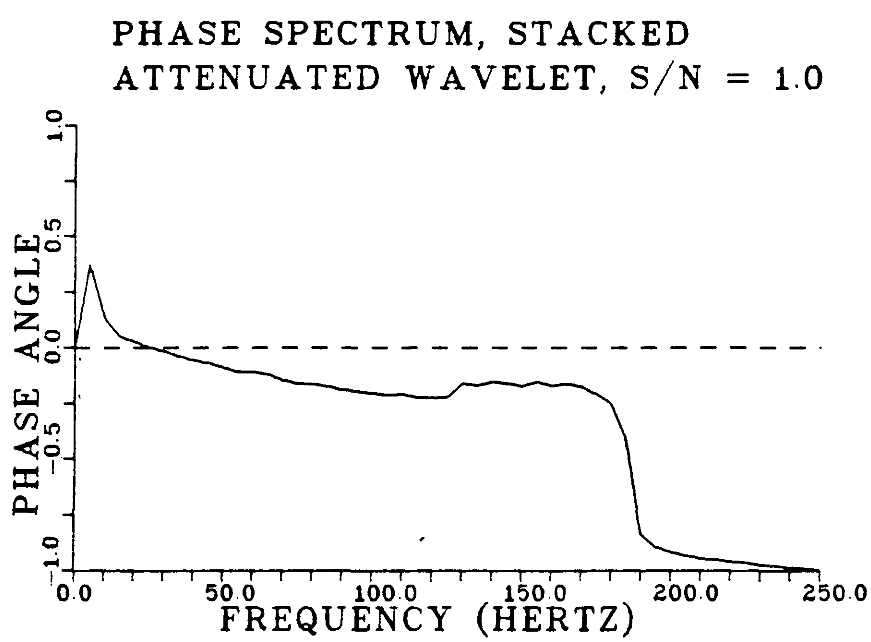

STACKED ZERO PHASE DECONVOLVED WAVELET
S/N = 1.0, ATTENUATED, 5—10—125—188 BANDPASS
AMPLITUDE
-0.5
0.0
0.5
1.0
1.5
TIME (MSEC)
-100.0
-50.0
0.0
50.0
100.0

Figure 50


AMPLITUDE SPECTRUM, STACKED
ATTENUATED WAVELET, S/N = 1.0
AMPLITUDE
0.0
0.5
1.0
0.0
50.0
100.0
150.0
200.0
250.0
FREQUENCY (HERTZ)

(a)


PHASE SPECTRUM, STACKED
ATTENUATED WAVELET, S/N = 1.0
PHASE ANGLE
-1.0
-0.5
0.0
0.5
1.0
0.0
50.0
100.0
150.0
200.0
250.0
FREQUENCY (HERTZ)

(b)

Figure 51

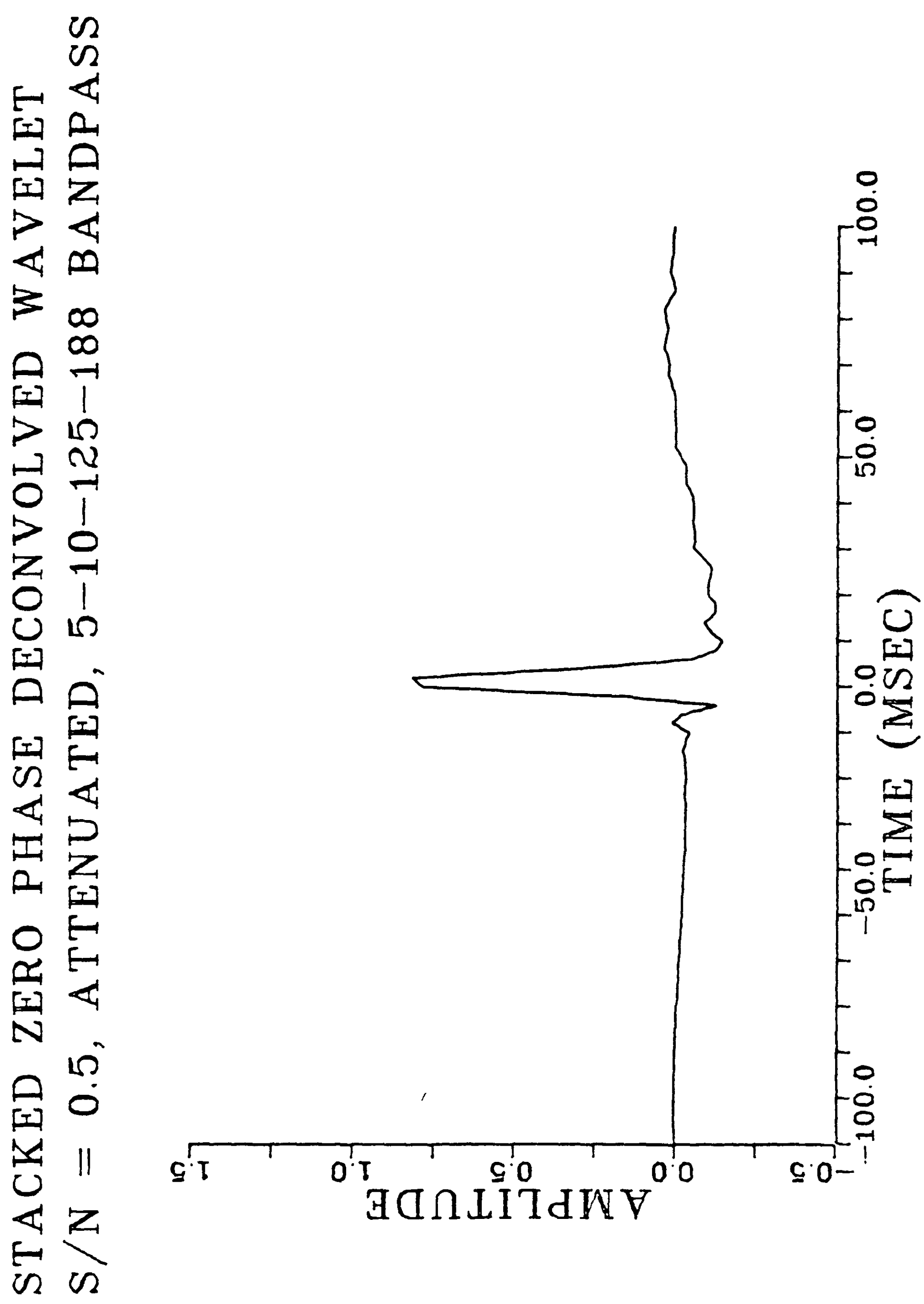

STACKED ZERO PHASE DECONVOLVED WAVELET
S/N = 0.5, ATTENUATED, 5–10–125–188 BANDPASS
AMPLITUDE
-0.5
0.0
0.5
1.0
1.5
TIME (MSEC)
-100.0
-50.0
0.0
50.0
100.0

Figure 52

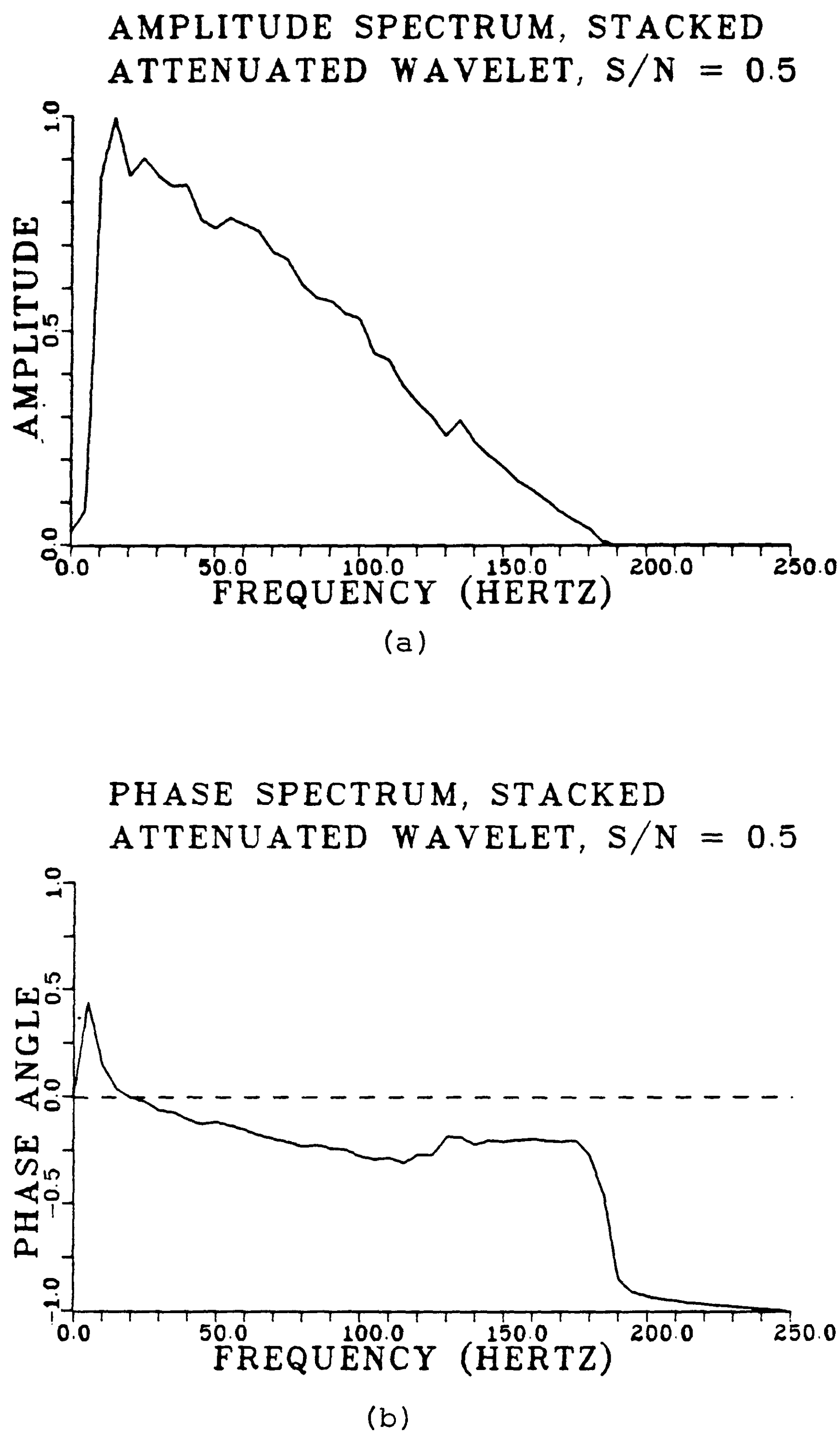

AMPLITUDE SPECTRUM, STACKED
ATTENUATED WAVELET, S/N = 0.5
AMPLITUDE
1.0
0.5
0.0
0.0
50.0
100.0
150.0
200.0
250.0
FREQUENCY (HERTZ)
(a)
PHASE SPECTRUM, STACKED
ATTENUATED WAVELET, S/N = 0.5
PHASE ANGLE
1.0
0.5
0.0
-0.5
-1.0
0.0
50.0
100.0
150.0
200.0
250.0
FREQUENCY (HERTZ)
(b)

Figure 53

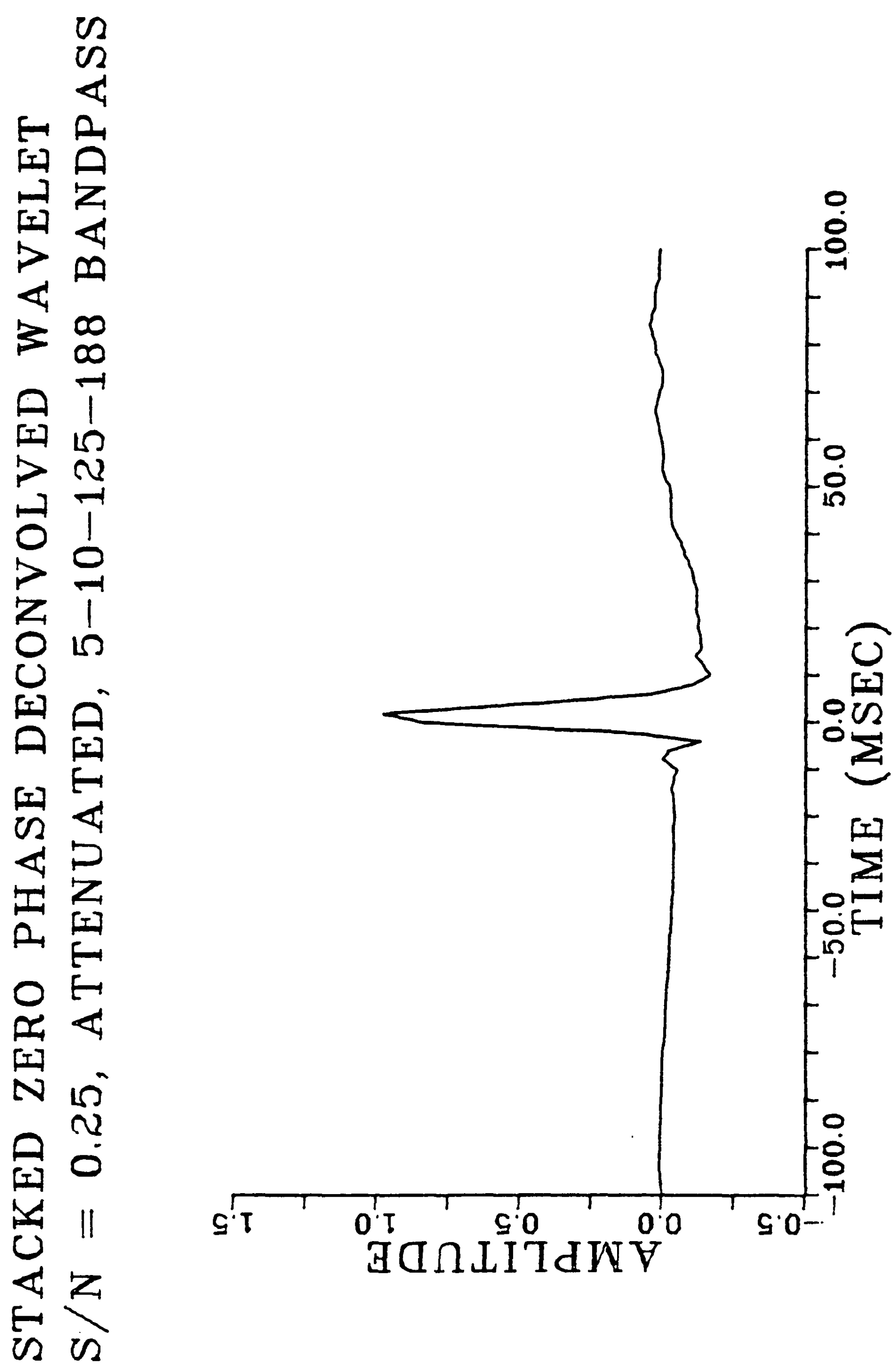
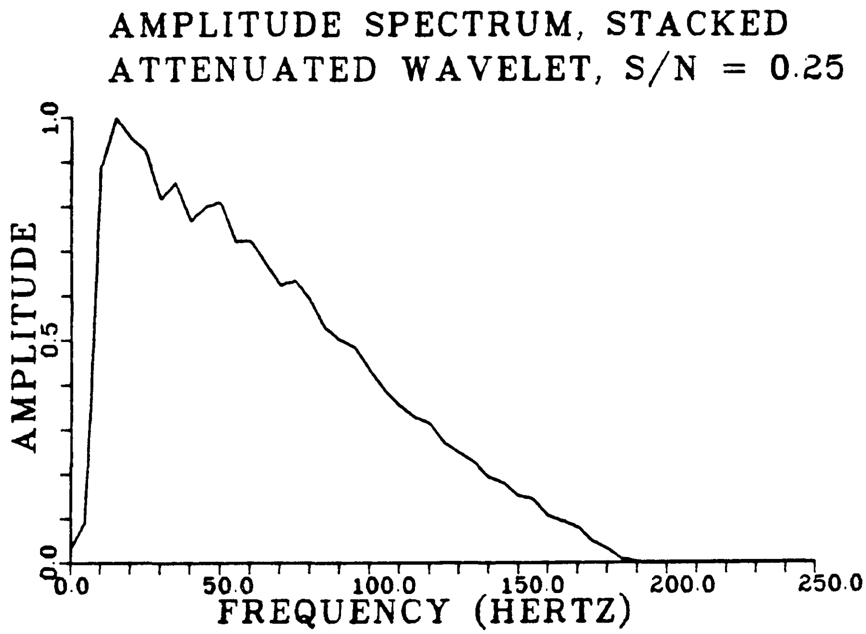
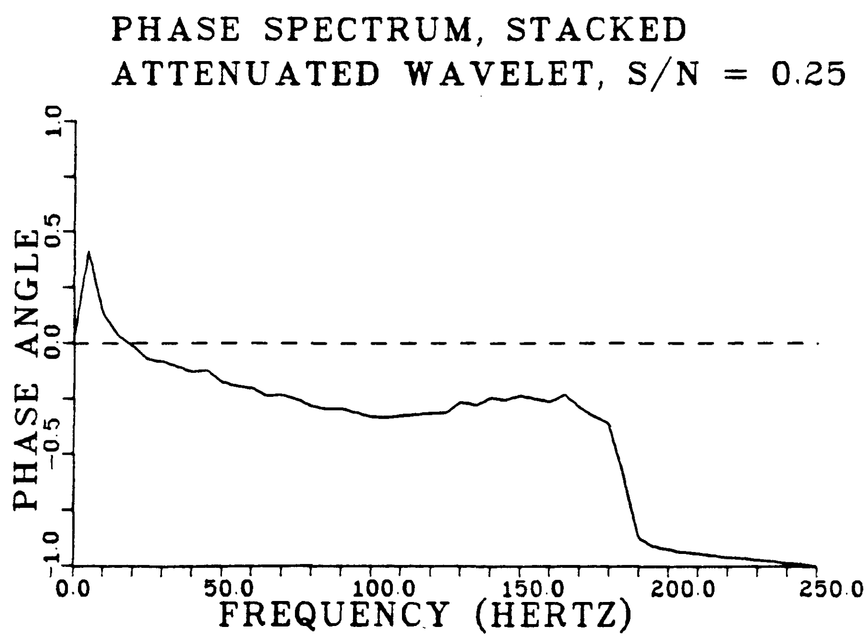

STACKED ZERO PHASE DECONVOLVED WAVELET
S/N = 0.25, ATTENUATED, 5-10-125-188 BANDPASS
AMPLITUDE
-0.5
0.0
0.5
1.0
1.5
TIME (MSEC)
-100.0
-50.0
0.0
50.0
100.0

Figure 54


AMPLITUDE SPECTRUM, STACKED
ATTENUATED WAVELET, S/N = 0.25
AMPLITUDE
1.0
0.5
0.0
0.0
50.0
100.0
150.0
200.0
250.0
FREQUENCY (HERTZ)

(a)


PHASE SPECTRUM, STACKED
ATTENUATED WAVELET, S/N = 0.25
PHASE ANGLE
1.0
0.5
0.0
-0.5
-1.0
0.0
50.0
100.0
150.0
200.0
250.0
FREQUENCY (HERTZ)

(b)

Figure 55

Examination of the attenuated wavelet model indicates that unlike the wavelet model, the phase error is strongly dependent upon signal-to-noise ratio. This observation is demonstrated quite graphically in Figure 56, where phase error in degrees is plotted versus the log of signal-to-noise ratio for the attenuated wavelet model. Within the bandlimits under investigation, phase error differences of up to thirty-five degrees throughout a signal-to-noise improvement of eighteen dB are recognized on the 90 Hertz and 110 Hertz curves.

The contrasting phase error results between the wavelet model and the attenuated wavelet model appear to be a function of the spectral characteristics of both the signal and noise functions. The amplitude spectra of the input wavelet and the noise realization are very similar in the wavelet model, hence the deconvolution process has little difficulty discerning which function to act upon, and the resultant phase error is not signal to noise ratio dependent. But for the attenuated wavelet model, the amplitude spectrum of the noise function remains roughly the same as in the first model, while the input wavelet is attenuated and deficient in high frequency components. This difference in spectral characteristics confuses the process of deconvolution as to which function to act upon, therefore, the resulting phase error is strongly signal-to-noise ratio dependent.

ATTENUATED WAVELET MODEL PHASE ERROR CURVES, 30 TO 170 HERTZ

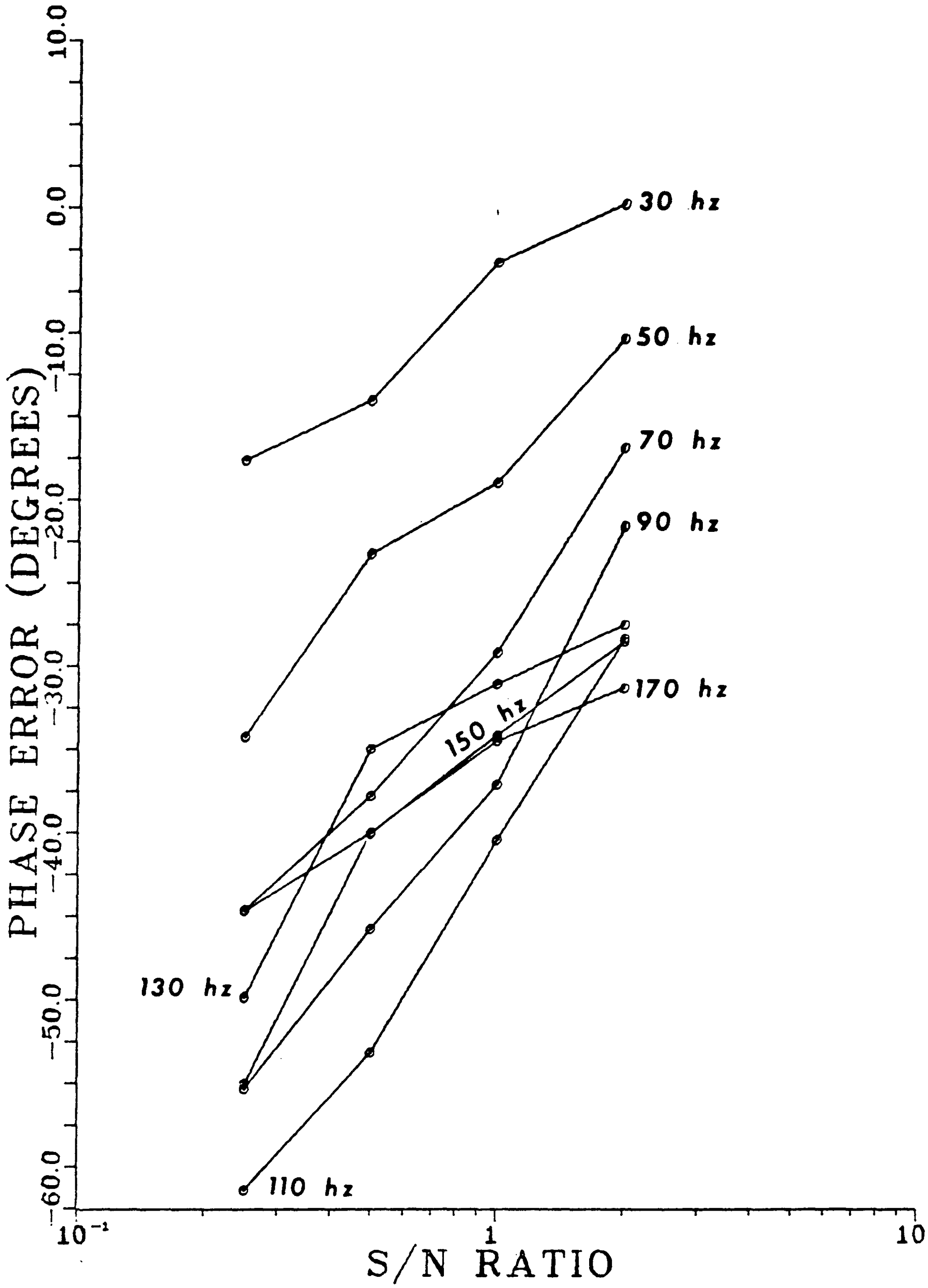


Figure 56

Inelastic attenuation is certainly a problem that must be universally contended with in seismic practice. It is realistic to assume that on a typical seismic trace, the signal and noise functions will posses different spectral characteristics. It seems reasonable, therefore, to assume a seismic model where phase error is signal-to-noise ratio dependent. Analysis of such a model indicates the desirability of improving the signal-to-noise ratio prior to deconvolution. It is suggested that processing schemes applying deconvolution before stack should be closely evaluated as to the reliability of the output, in view of the data here indicating deconvolution after stack to be a superior method.

Finally, the variance from trace to trace, or standard deviation, was examined for this model. Figures 57 through 60 plot standard deviation in fractions of π versus frequency in Hertz for the four signal-to-noise ratio models. While the curves are more erratic than their wavelet model counterparts, and become more erratic as the signal-to-noise ratio decreases, they do not differ significantly in overall shape or relative amplitude from the aforementioned wavelet model equivalents. The standard deviation increases, however, with decreasing signal-to-noise ratio, as comparison of Figures 57 through 60 indicates. This relationship is even more evident upon examination of Figure 61, where standard deviation in degrees is plotted versus signal-to-noise ratio. As with the results described

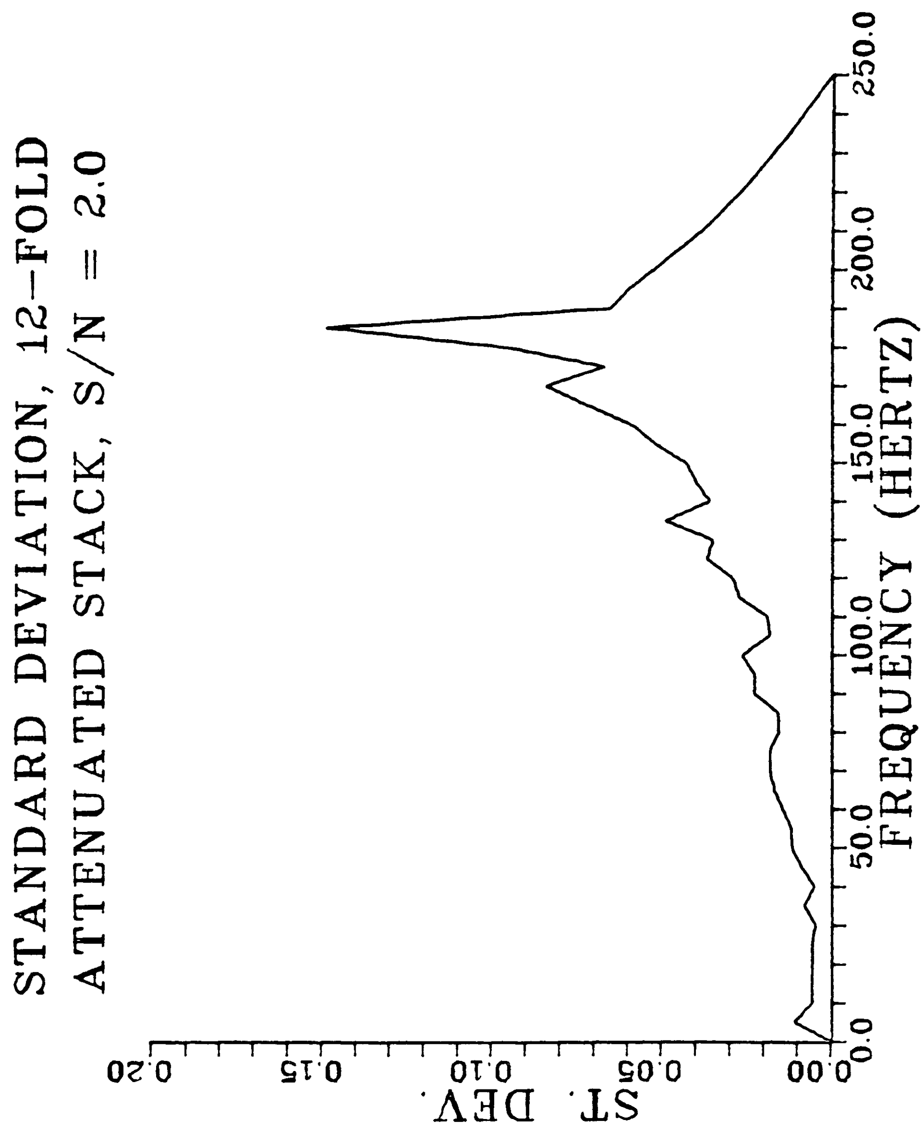

STANDARD DEVIATION, 12—FOLD
ATTENUATED STACK, S/N = 2.0
FREQUENCY (HERTZ)
0.0
50.0
100.0
150.0
200.0
250.0
ST. DEV.
0.00
0.05
0.10
0.15
0.20

Figure 57

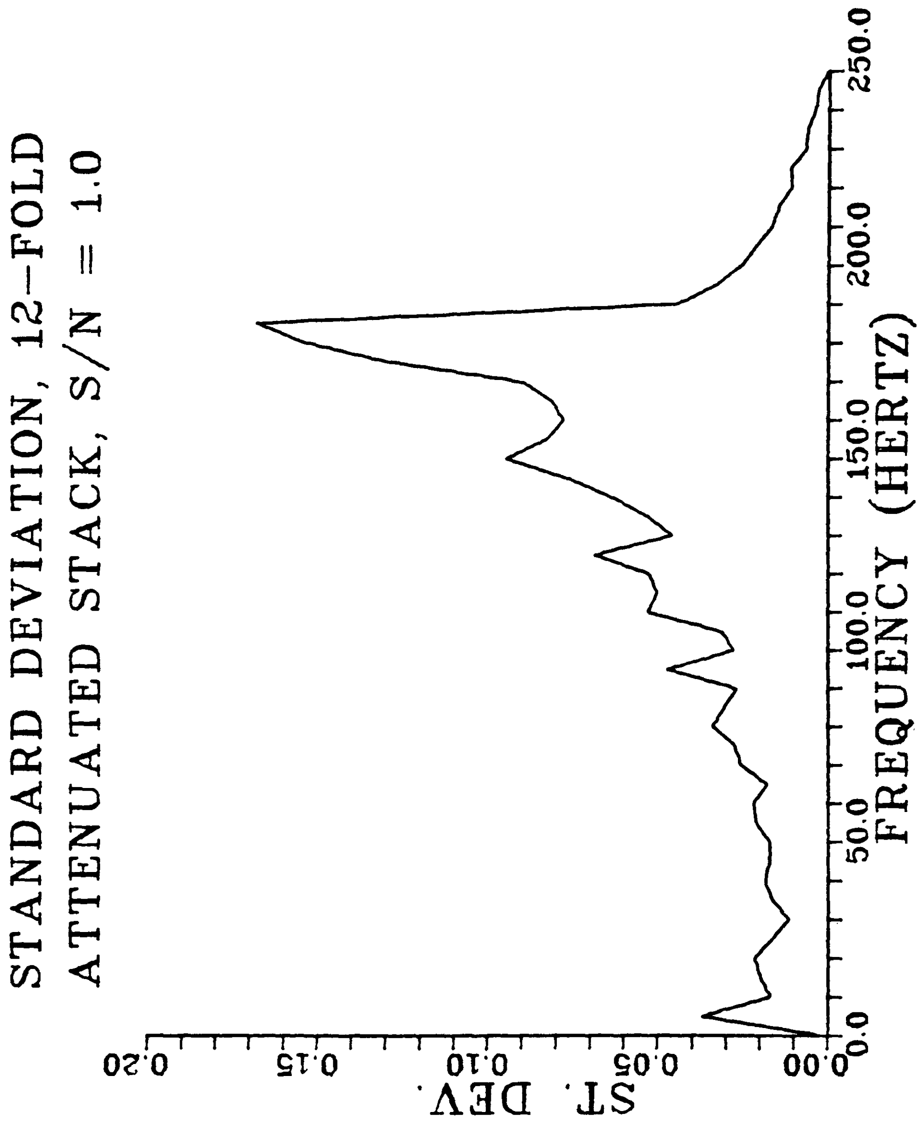

STANDARD DEVIATION, 12–FOLD
ATTENUATED STACK, S/N = 1.0
FREQUENCY (HERTZ)
ST. DEV.
0.0
50.0
100.0
150.0
200.0
250.0
0.00
0.05
0.10
0.15
0.20

Figure 58

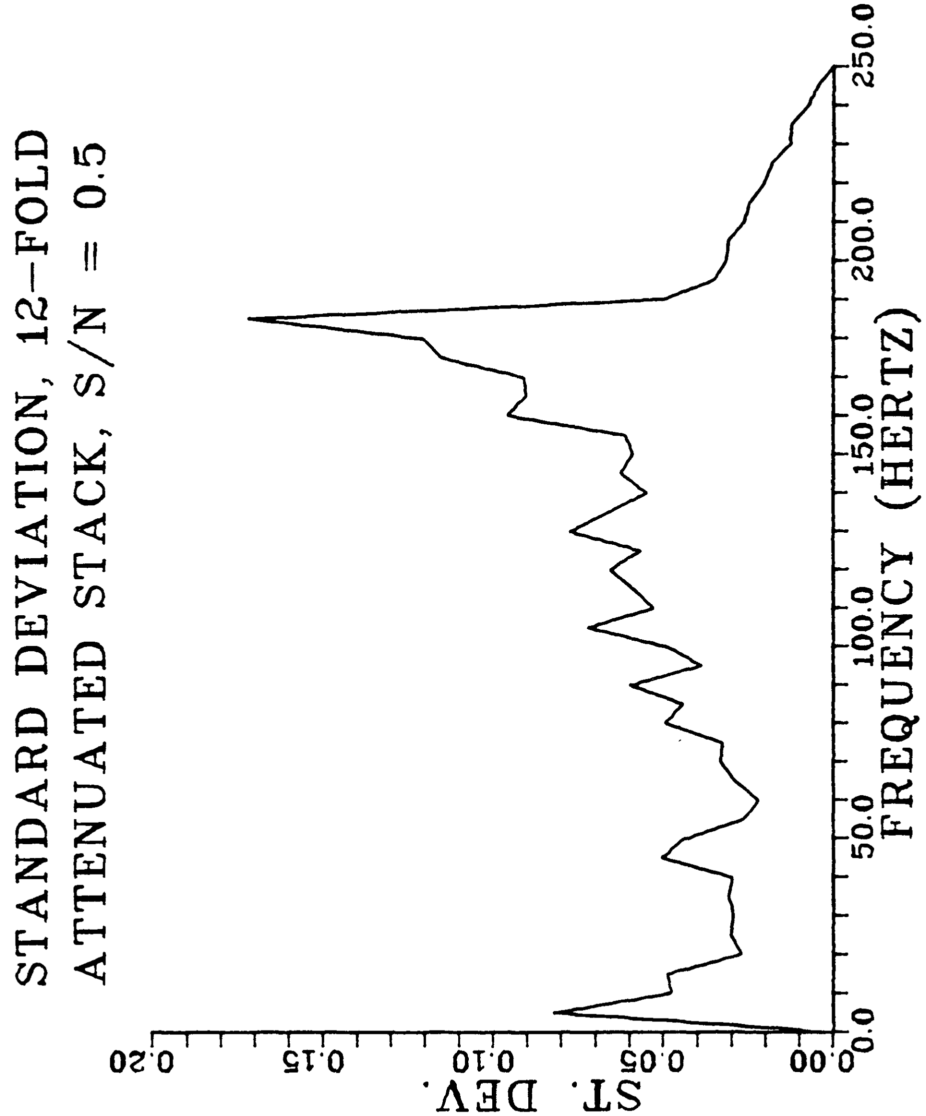

STANDARD DEVIATION, 12—FOLD
ATTENUATED STACK, S/N = 0.5
FREQUENCY (HERTZ)
0.0
50.0
100.0
150.0
200.0
250.0
ST. DEV.
0.00
0.05
0.10
0.15
0.20

Figure 59

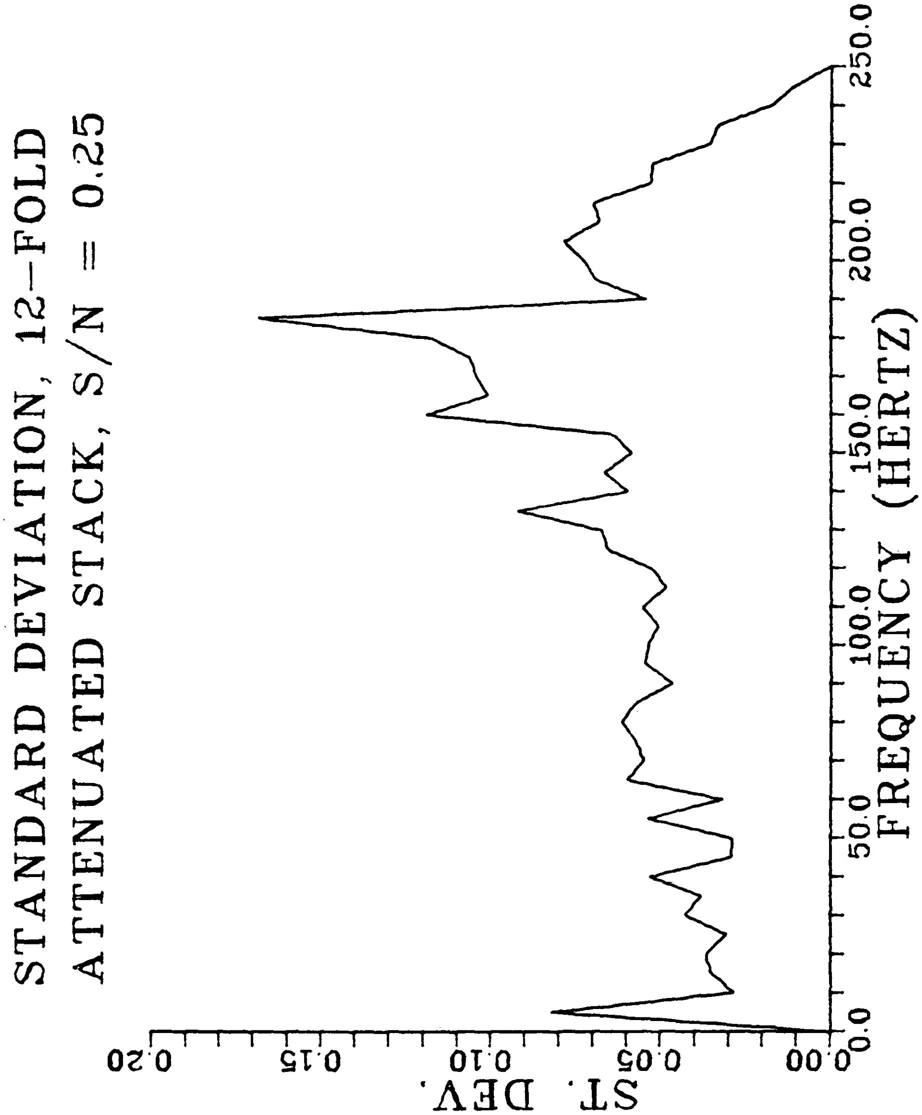

STANDARD DEVIATION, 12—FOLD
ATTENUATED STACK, S/N = 0.25
FREQUENCY (HERTZ)
0.0
50.0
100.0
150.0
200.0
250.0
ST. DEV.
0.00
0.05
0.10
0.15
0.20

Figure 60

ATTENUATED WAVELET MODEL STANDARD DEVIATION CURVES, 30 TO 170 HERTZ

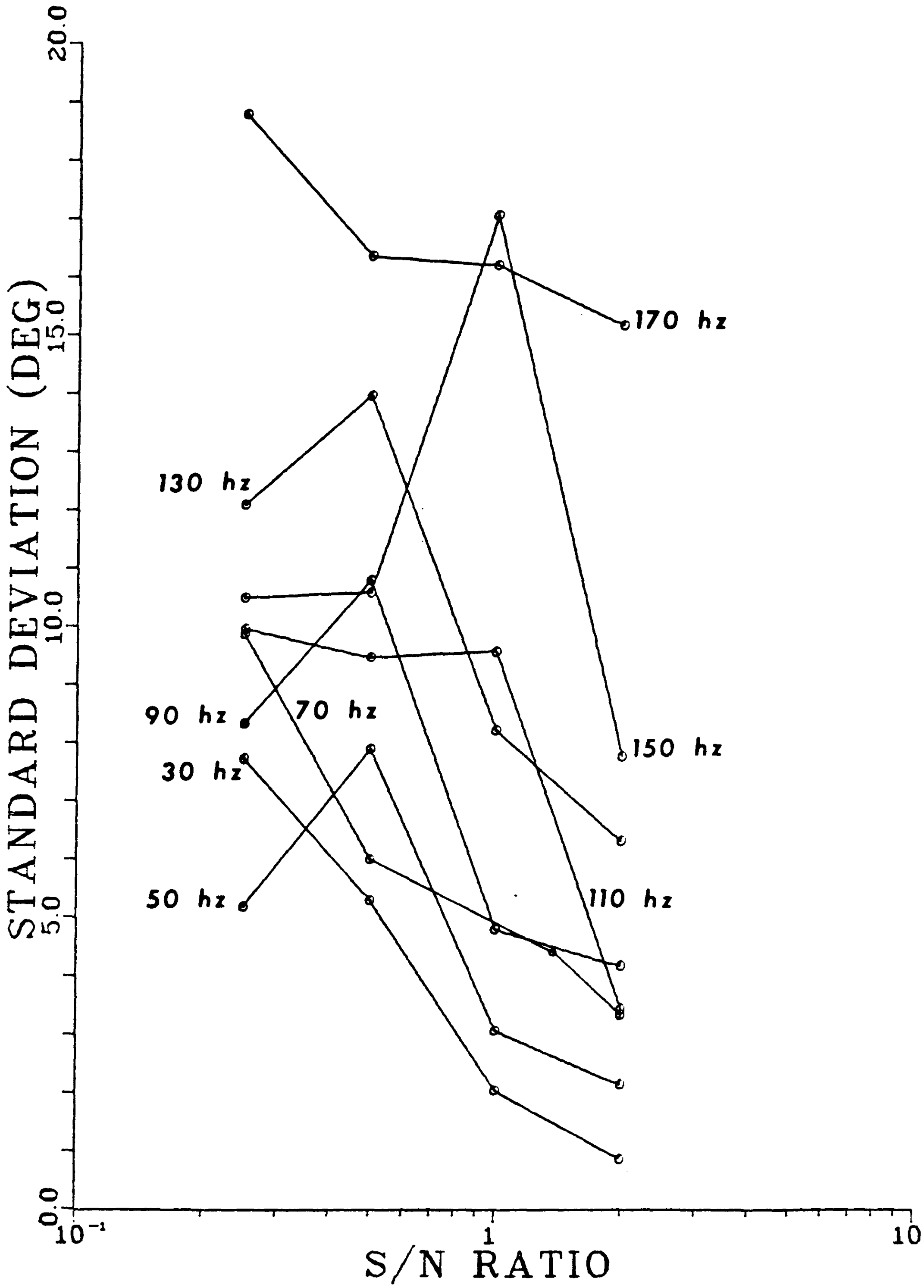


Figure 61

for the wavelet model, the standard deviation is minimal, being on the order of ten degrees or less.

This result implies once again, that for high frequencies and low signal-to-noise ratios, an inherent amplitude loss is to be expected as a consequence of CDP stacking.

WAVELET WITH GHOST MODEL

Marine ghosts can be described by the convolution of a wavelet with an operator consisting of a pair of approximately equal spikes of opposite polarity separated by Δt, the two-way travel time between the source towing depth and the ocean surface. Convolution of such a spike pair with a bandlimited wavelet produces an amplitude spectrum with notches at zero and positive integer multiples of $1/\Delta t$.

The third model examined consists of a 1.0, -0.99 spike pair, separated by twenty milliseconds, convolved with the original 5-10-125-188 bandpass zero phase wavelet. This operation produces the wavelet shown in Figure 62, with spectral characteristics shown in Figure 63. Clearly, the ghost operator has produced notches in the amplitude spectrum and 140 degree discontinuities in the phase spectrum every fifty Hertz.

Through application of a Wiener-Levinson filter, a minimum phase wavelet estimate (Figure 64) and its minimum phase inverse are designed. The amplitude and phase spectra of the minimum phase wavelet are depicted in Figure 65.

Consistent with the processing technique applied in the analysis of the first two models, noise corrupted minimum phase wavelets are designed and subsequently deconvolved by the ideal minimum phase inverse with a ghost. The deconvolved result is then bandpass filtered by the

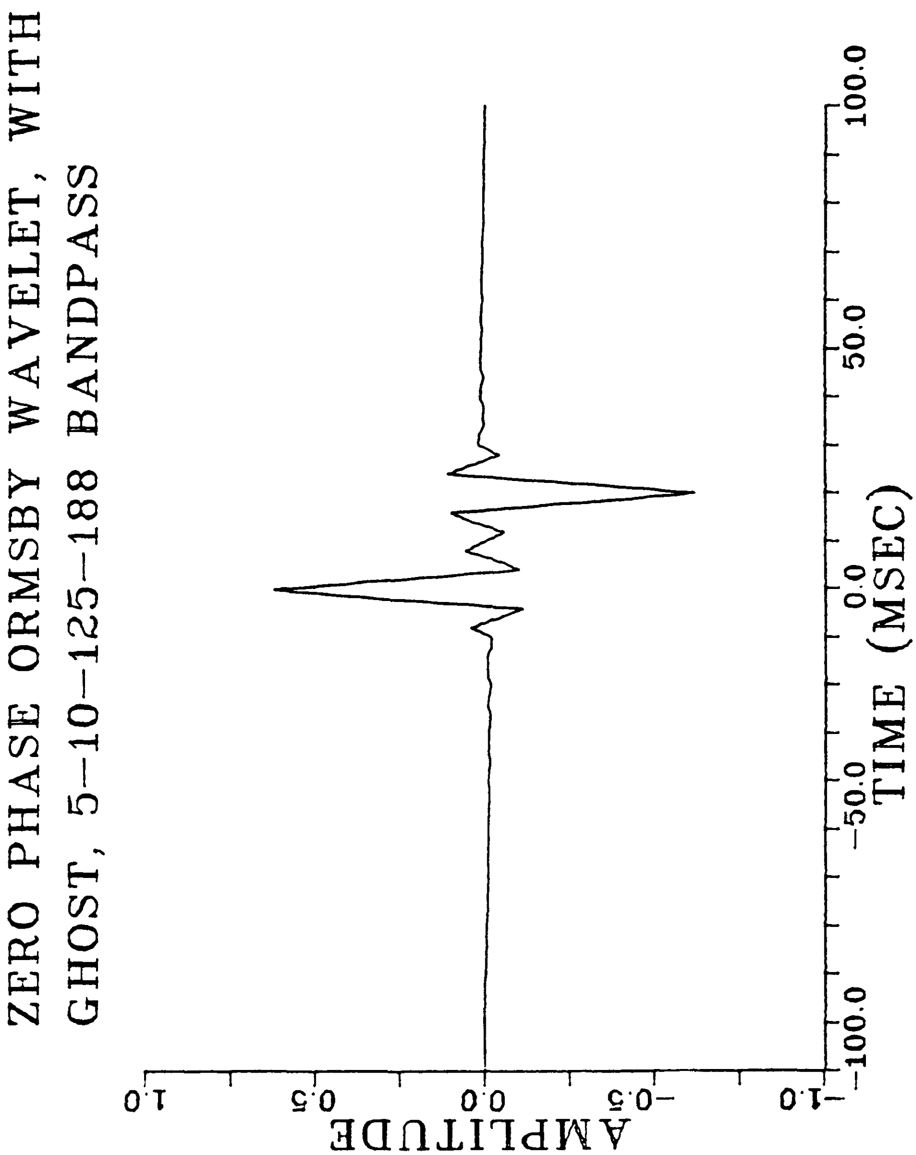
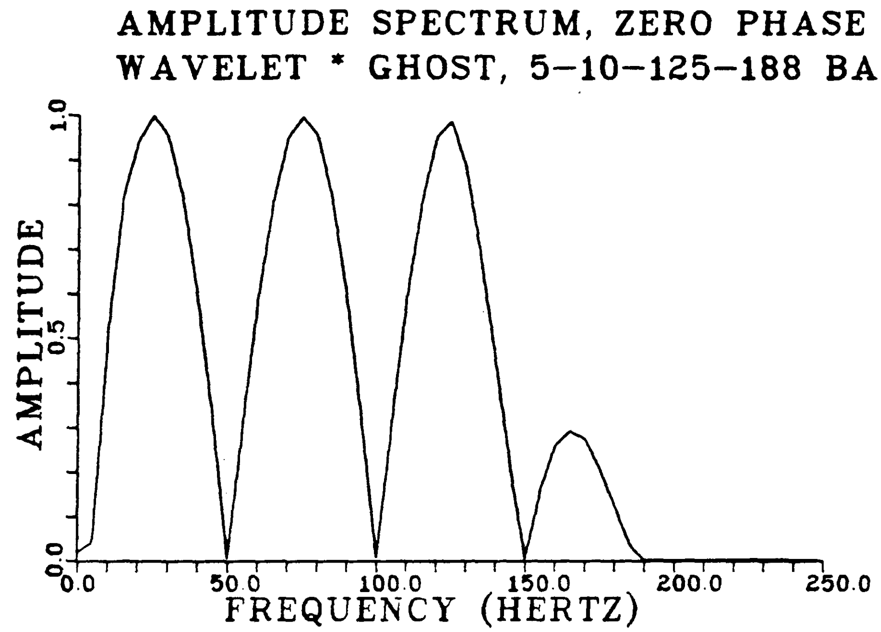
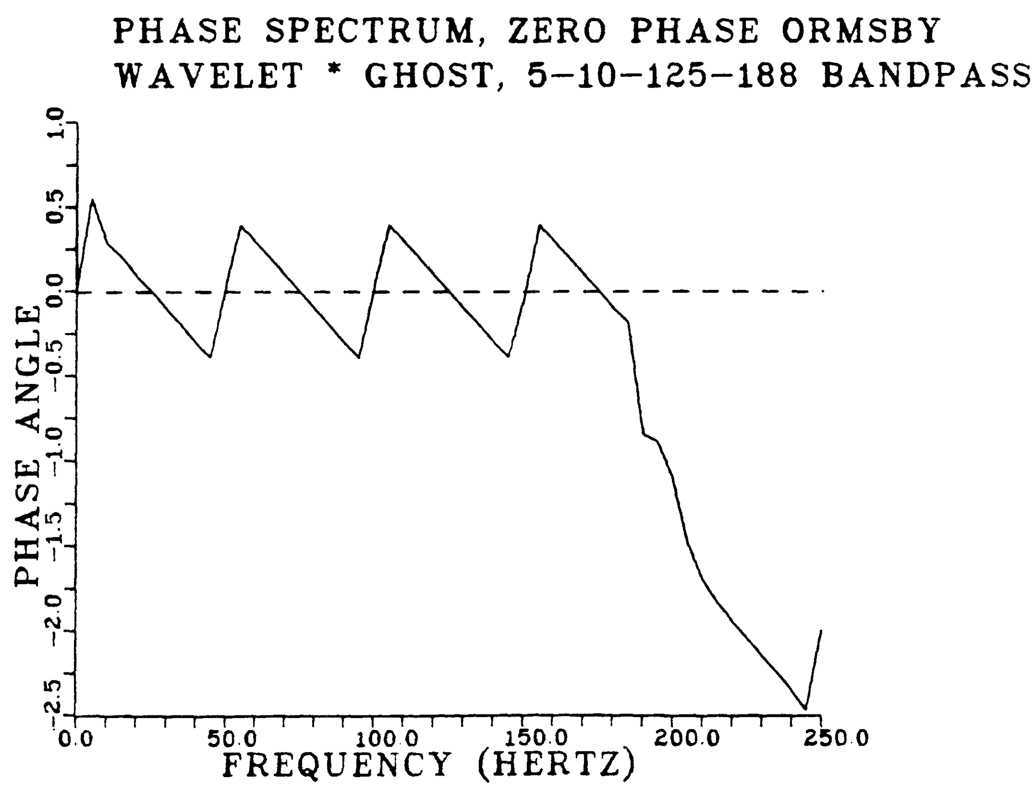

ZERO PHASE ORMSBY WAVELET, WITH
GHOST, 5-10-125-188 BANDPASS
AMPLITUDE
-1.0
-0.5
0.0
0.5
1.0
TIME (MSEC)
-100.0
-50.0
0.0
50.0
100.0

Figure 62


AMPLITUDE SPECTRUM, ZERO PHASE ORMSBY
WAVELET * GHOST, 5-10-125-188 BANDPASS
AMPLITUDE
0.0
0.5
1.0
0.0
50.0
100.0
150.0
200.0
250.0
FREQUENCY (HERTZ)

(a)


PHASE SPECTRUM, ZERO PHASE ORMSBY
WAVELET * GHOST, 5-10-125-188 BANDPASS
PHASE ANGLE
-2.5
-2.0
-1.5
-1.0
-0.5
0.0
0.5
1.0
0.0
50.0
100.0
150.0
200.0
250.0
FREQUENCY (HERTZ)

(b)

Figure 63

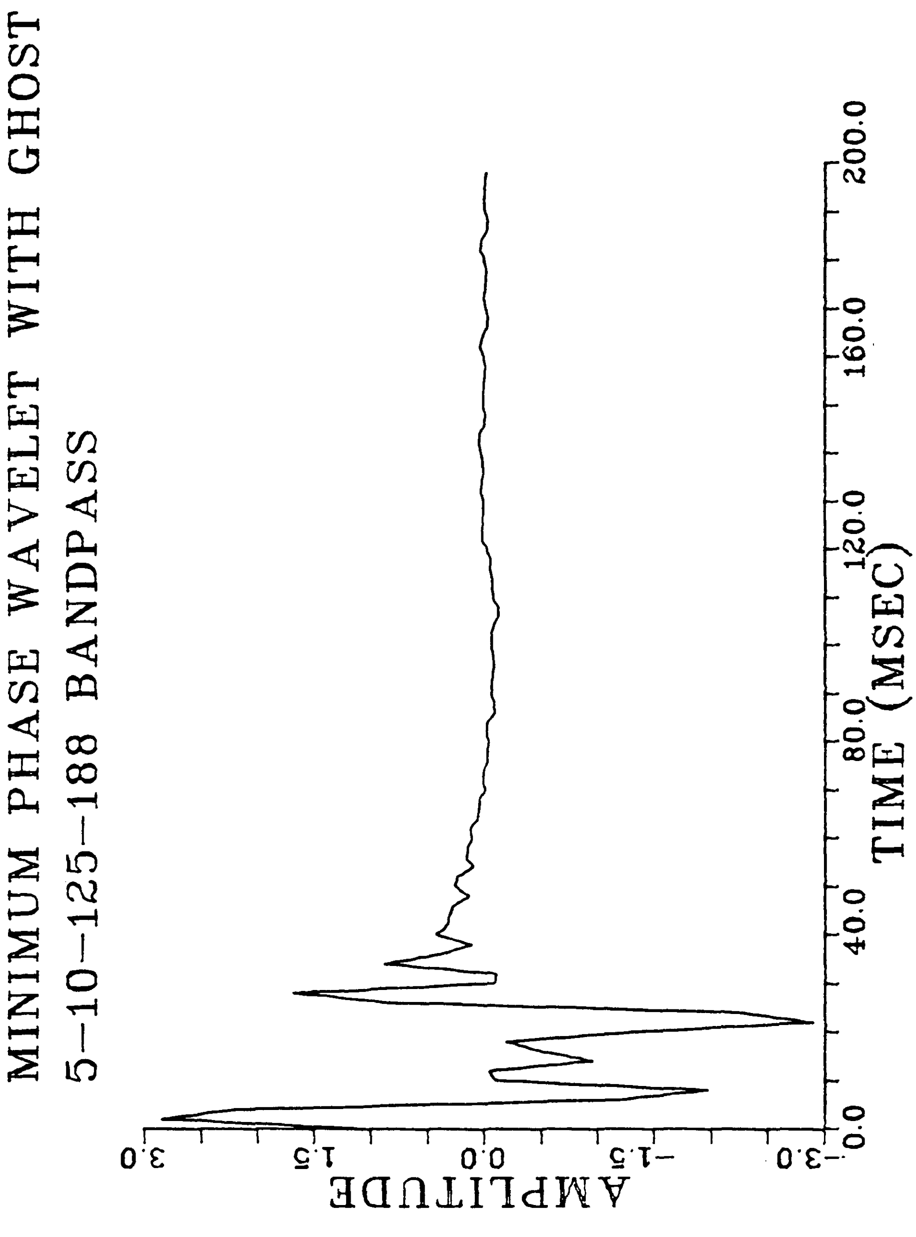
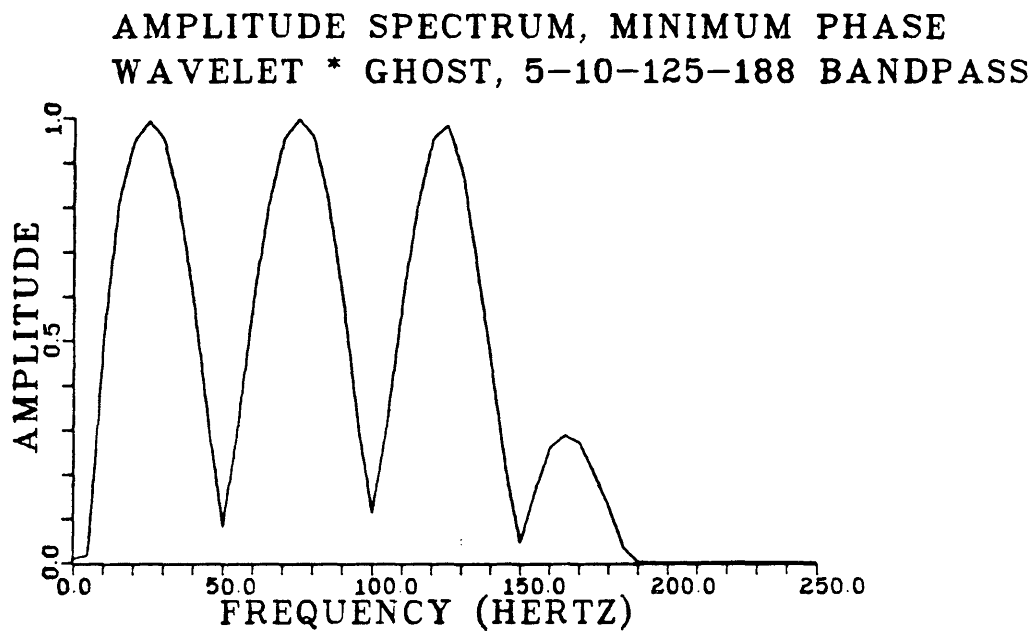
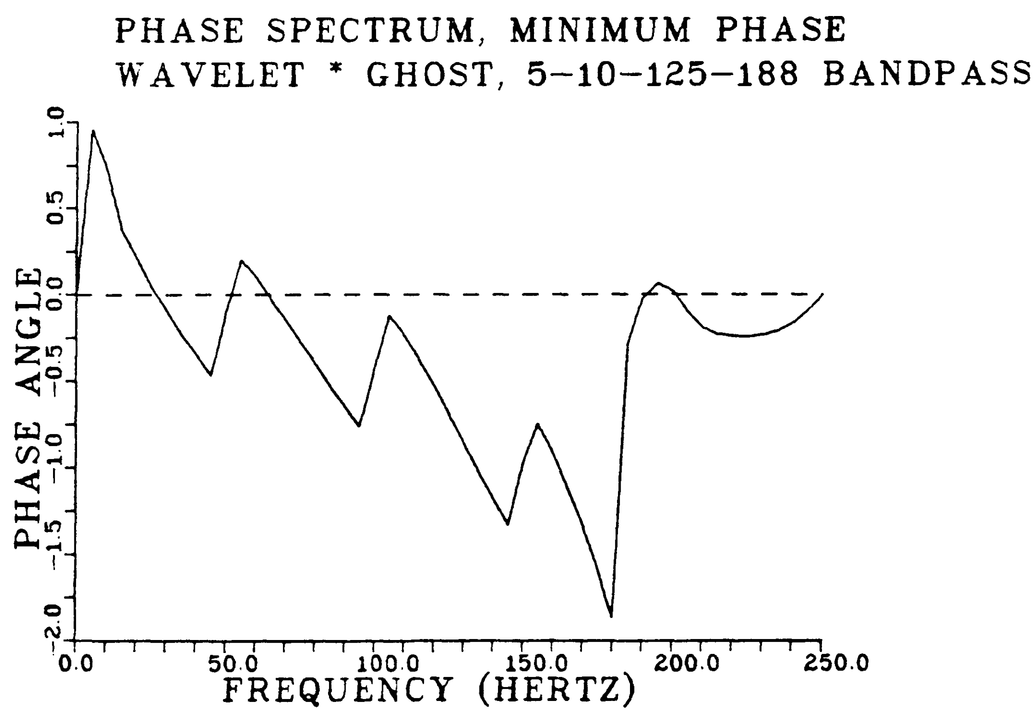

MINIMUM PHASE WAVELET WITH GHOST
5–10–125–188 BANDPASS
TIME (MSEC)
0.0
40.0
80.0
120.0
160.0
200.0
AMPLITUDE
3.0
1.5
0.0
-1.5
-3.0

Figure 64


AMPLITUDE SPECTRUM, MINIMUM PHASE
WAVELET * GHOST, 5-10-125-188 BANDPASS
AMPLITUDE
0.0
0.5
1.0
0.0
50.0
100.0
150.0
200.0
250.0
FREQUENCY (HERTZ)

(a)


PHASE SPECTRUM, MINIMUM PHASE
WAVELET * GHOST, 5-10-125-188 BANDPASS
PHASE ANGLE
-2.0
-1.5
-1.0
-0.5
0.0
0.5
1.0
0.0
50.0
100.0
150.0
200.0
250.0
FREQUENCY (HERTZ)

(b)

Figure 65

original wavelet. Applications of this process for signal-to-noise ratios of 2.0, 1.0, 0.5, and 0.25 yield the wavelets described by Figures 66 through 69. For a signal-to-noise ratio of two, the wavetrain is quite choppy and erratic, but deconvolution has removed the ghost quite well, as evidenced by its significantly reduced amplitude with repsect to the main pulse. When the signal-to-noise ratio decreases, the ghost is not effectively removed by deconvolution (Figures 67 and 68). In fact, when the signal-to-noise ratio reaches 0.25, the absolute amplitude of the ghost approximates that of the primary wavelet (Figure 69). Note however, that deconvolution has worked well on the main signal pulse in all the examples, despite the presence of the wavelet ghost.

Computations of filter performance parameters indicate a direct dependence upon signal-to-noise ratio, consistent with the results described in the first two models.

Examination of the amplitude and phase spectra of the four wavelets (Figures 70 through 73) reveals divergence from the desired result is a function of signal-to-noise ratio. The notches in the amplitude spectrum become wider and deeper at low signal-to-noise ratios. Also, as the signal-to-noise ratio decreases, the phase angle error increases. This observation, for signal and noise functions of different spectral shapes, is consistent with findings discussed for the attenuated wavelet model. Phase errors for the wavelet with a ghost model do not appear,

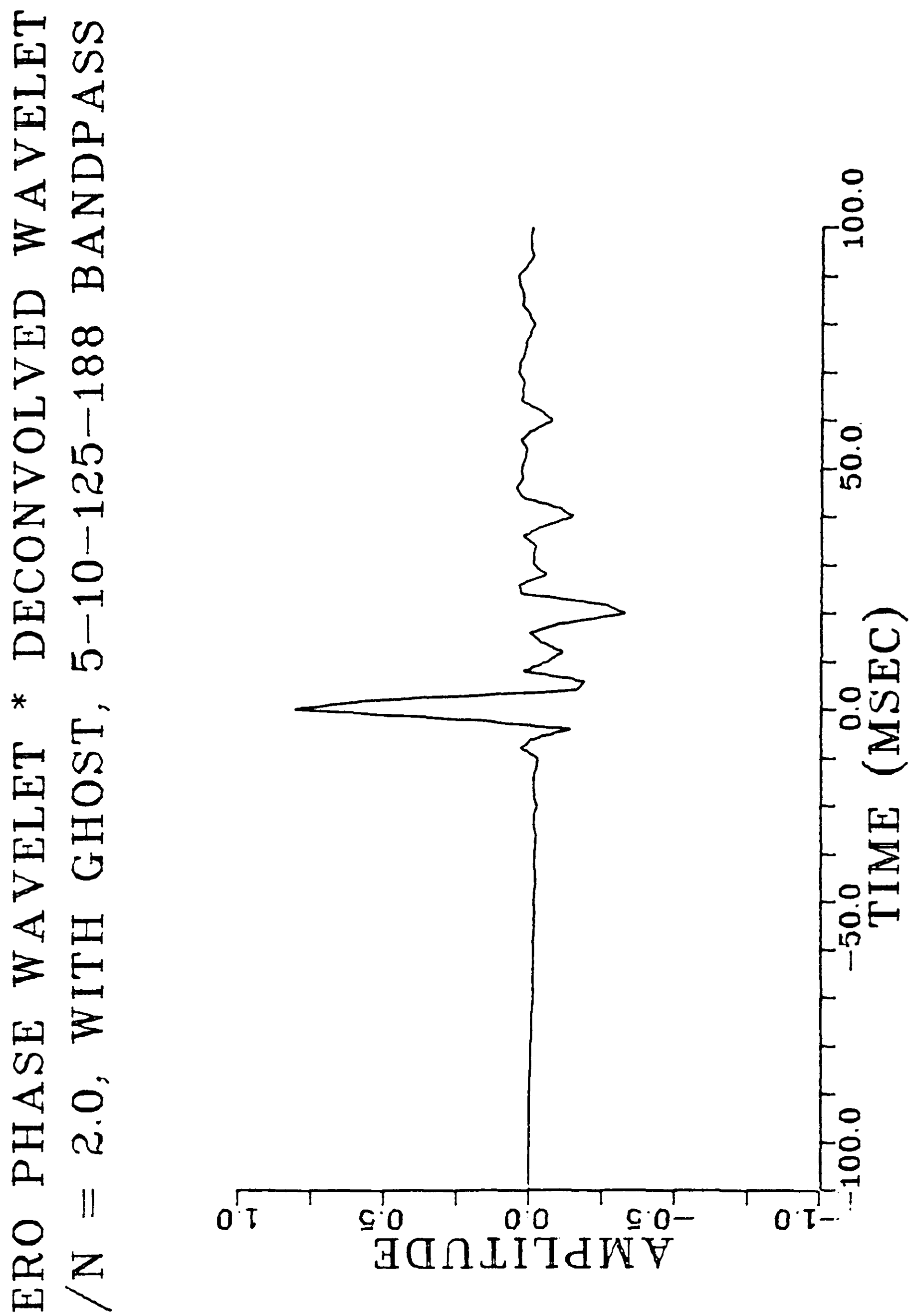

ZERO PHASE WAVELET * DECONVOLVED WAVELET
S/N = 2.0, WITH GHOST, 5-10-125-188 BANDPASS
AMPLITUDE
1.0
0.5
0.0
-0.5
-1.0
TIME (MSEC)
-100.0
-50.0
0.0
50.0
100.0

Figure 66

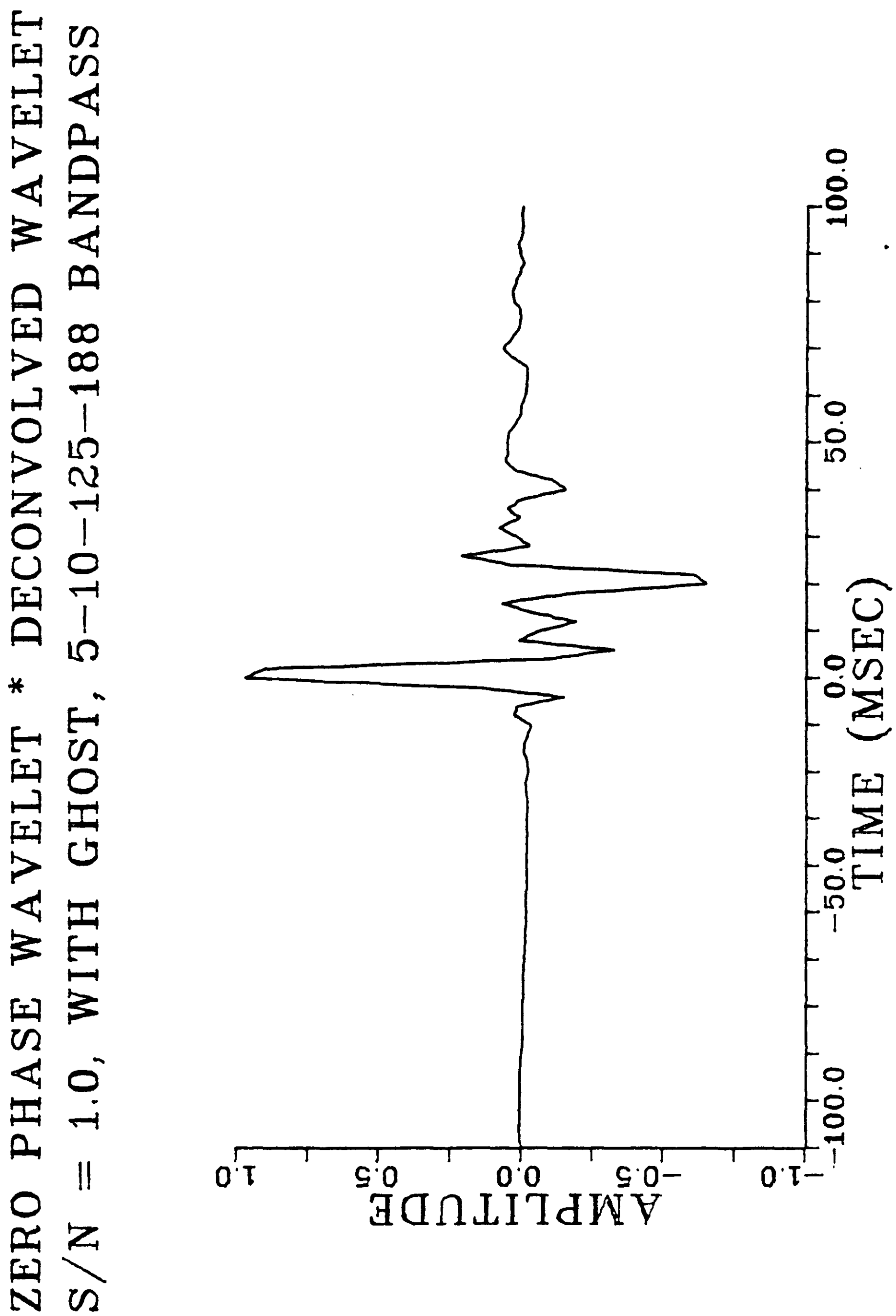

ZERO PHASE WAVELET * DECONVOLVED WAVELET
S/N = 1.0, WITH GHOST, 5–10–125–188 BANDPASS
AMPLITUDE
1.0
0.5
0.0
−0.5
−1.0
TIME (MSEC)
−100.0
−50.0
0.0
50.0
100.0

Figure 67

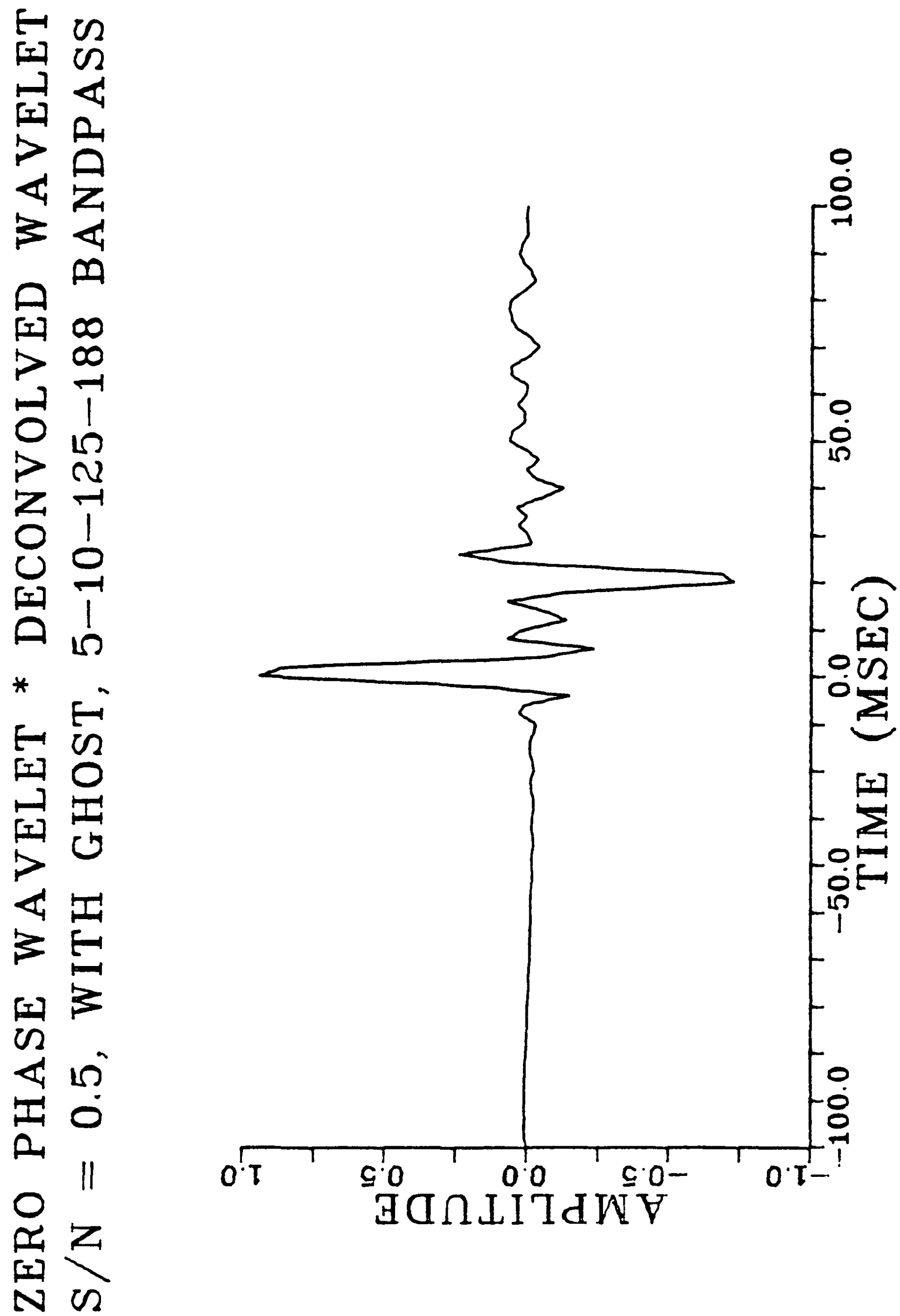

ZERO PHASE WAVELET * DECONVOLVED WAVELET
S/N = 0.5, WITH GHOST, 5-10-125-188 BANDPASS
AMPLITUDE
1.0
0.5
0.0
-0.5
-1.0
TIME (MSEC)
-100.0
-50.0
0.0
50.0
100.0

Figure 68

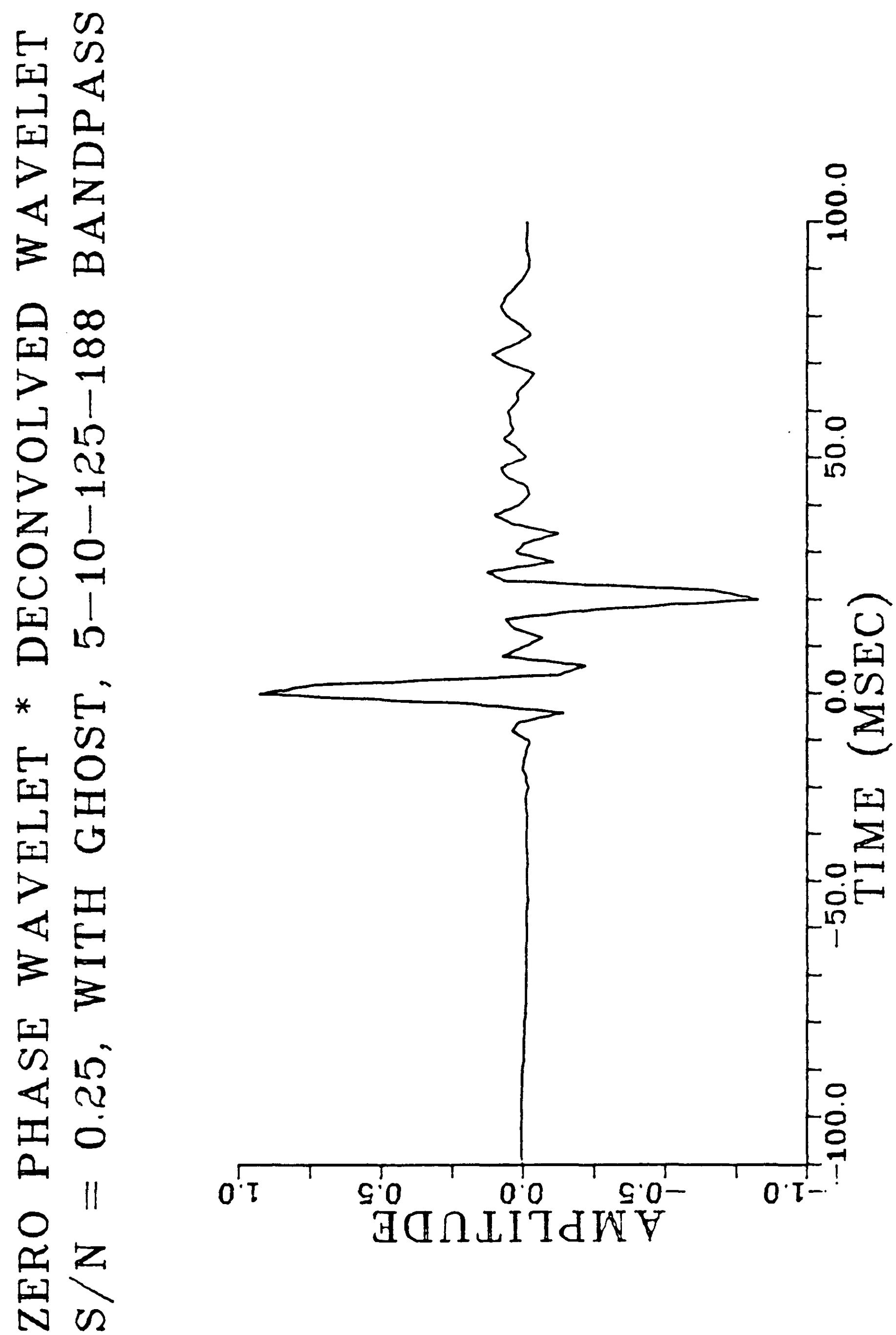
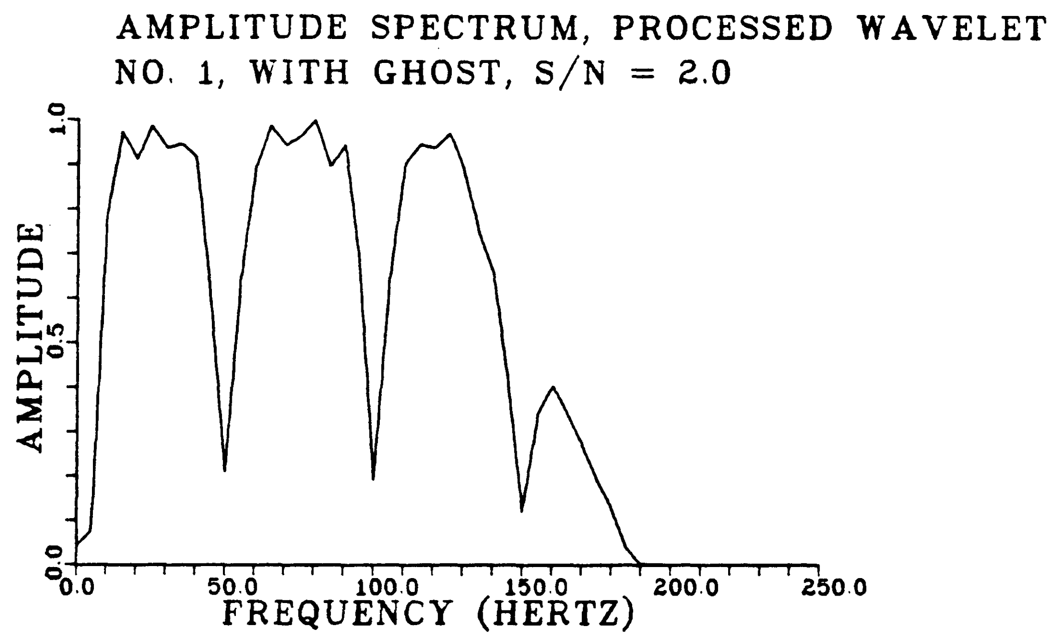
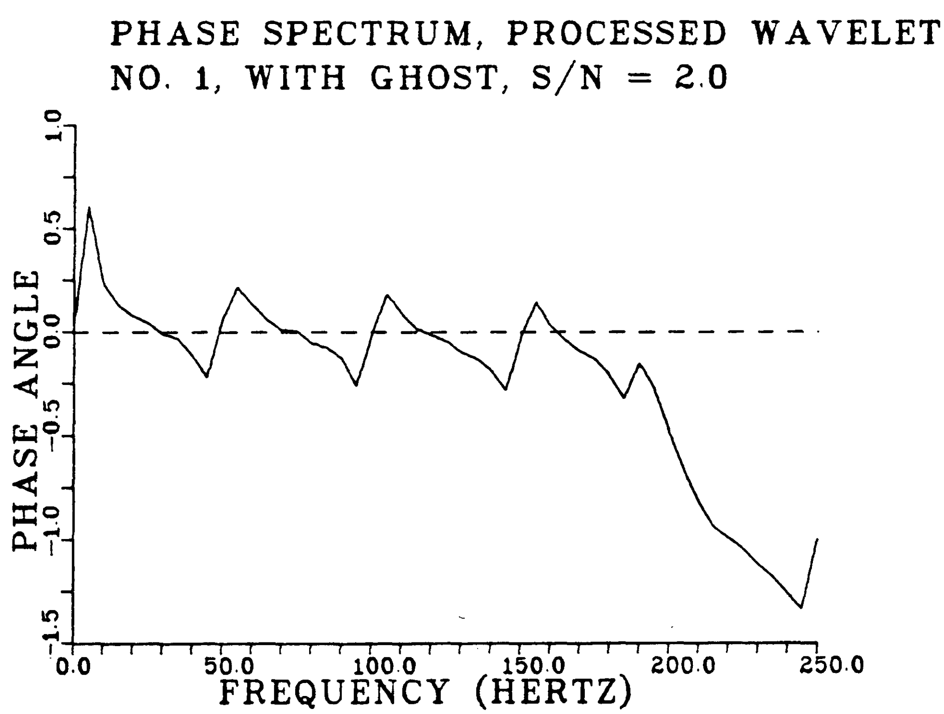

ZERO PHASE WAVELET * DECONVOLVED WAVELET
S/N = 0.25, WITH GHOST, 5-10-125-188 BANDPASS
AMPLITUDE
1.0
0.5
0.0
-0.5
-1.0
TIME (MSEC)
-100.0
-50.0
0.0
50.0
100.0

Figure 69

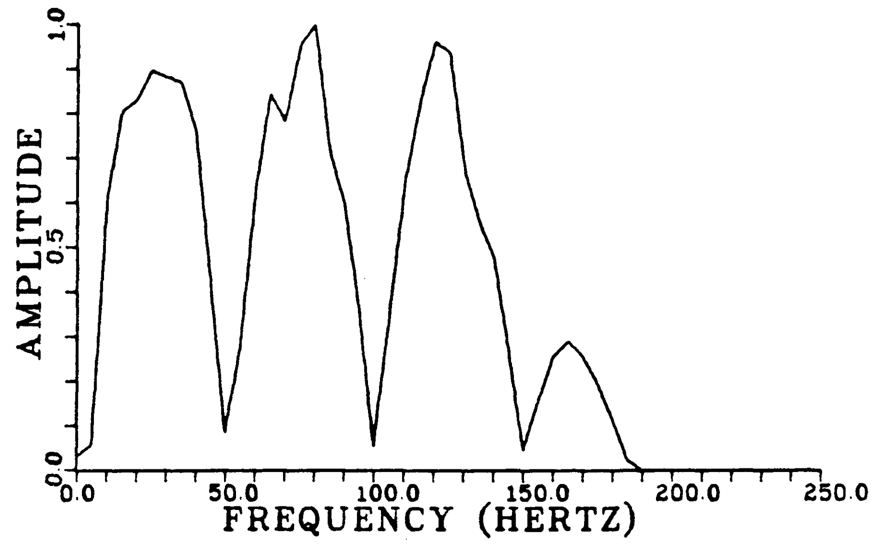

AMPLITUDE SPECTRUM, PROCESSED WAVELET
NO. 1, WITH GHOST, S/N = 2.0
AMPLITUDE
0.0
0.5
1.0
0.0
50.0
100.0
150.0
200.0
250.0
FREQUENCY (HERTZ)

(a)

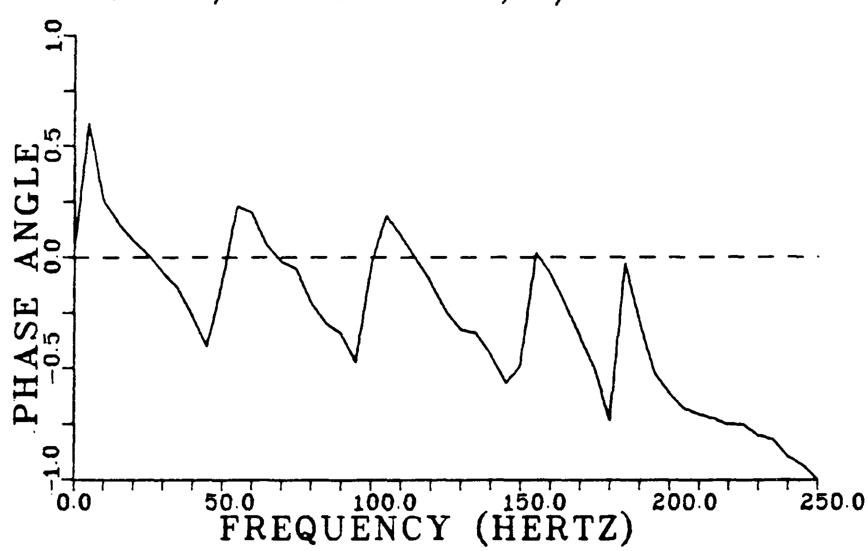

PHASE SPECTRUM, PROCESSED WAVELET
NO. 1, WITH GHOST, S/N = 2.0
PHASE ANGLE
-1.5
-1.0
-0.5
0.0
0.5
1.0
0.0
50.0
100.0
150.0
200.0
250.0
FREQUENCY (HERTZ)

(b)

Figure 70

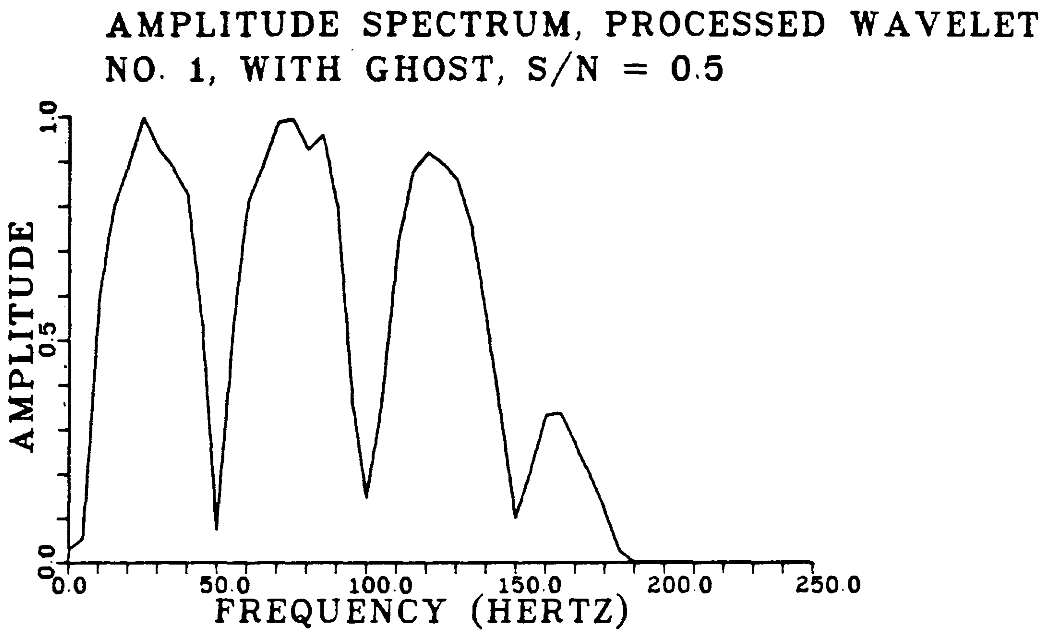

AMPLITUDE SPECTRUM, PROCESSED WAVELET
NO. 1, WITH GHOST, S/N = 1.0
AMPLITUDE
0.0
0.5
1.0
0.0
50.0
100.0
150.0
200.0
250.0
FREQUENCY (HERTZ)

(a)

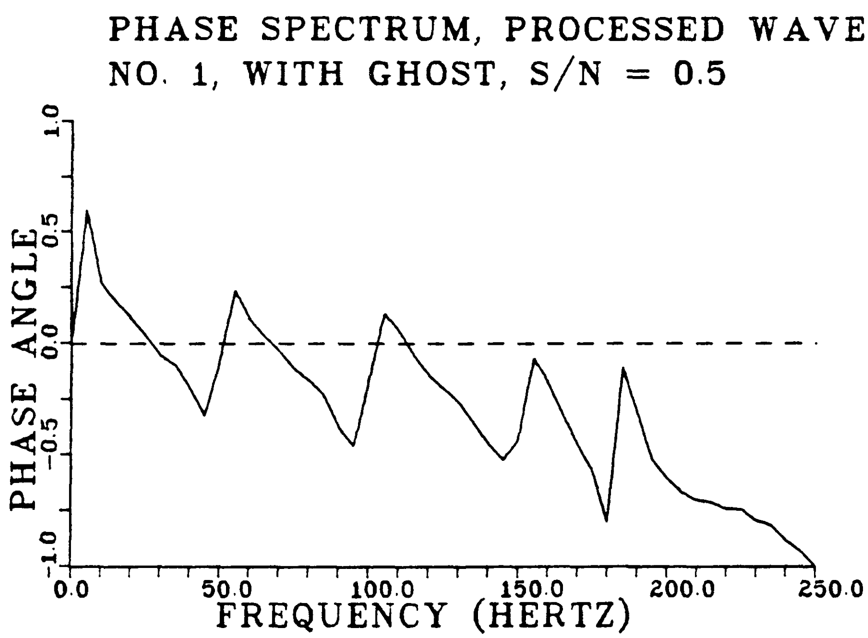

PHASE SPECTRUM, PROCESSED WAVELET
NO. 1, WITH GHOST, S/N = 1.0
PHASE ANGLE
-1.0
-0.5
0.0
0.5
1.0
0.0
50.0
100.0
150.0
200.0
250.0
FREQUENCY (HERTZ)

(b)

Figure 71

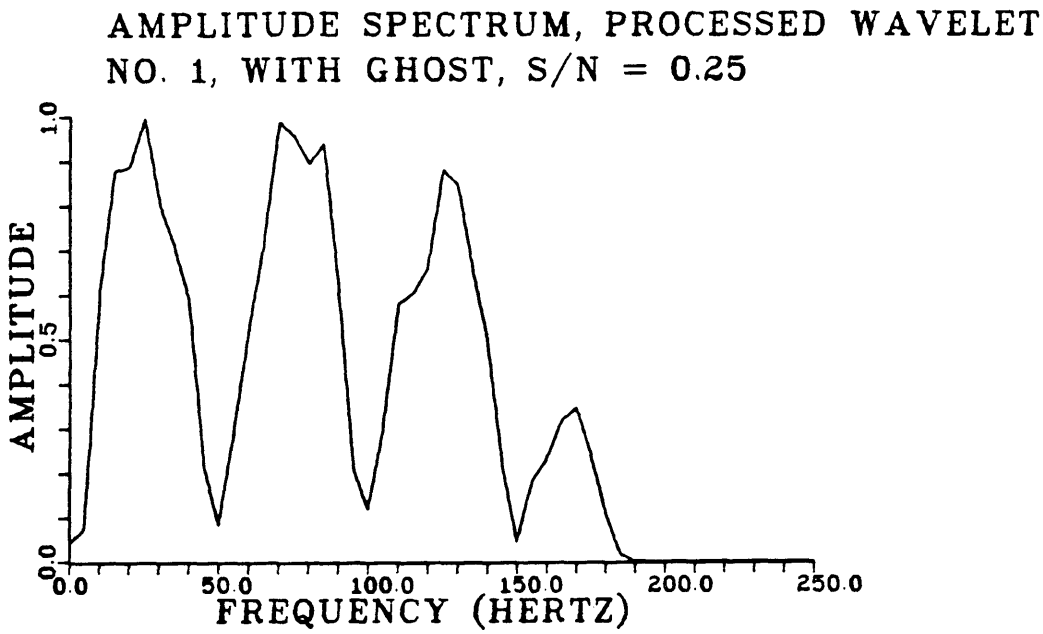

AMPLITUDE SPECTRUM, PROCESSED WAVELET
NO. 1, WITH GHOST, S/N = 0.5
AMPLITUDE
0.0
0.5
1.0
0.0
50.0
100.0
150.0
200.0
250.0
FREQUENCY (HERTZ)

(a)

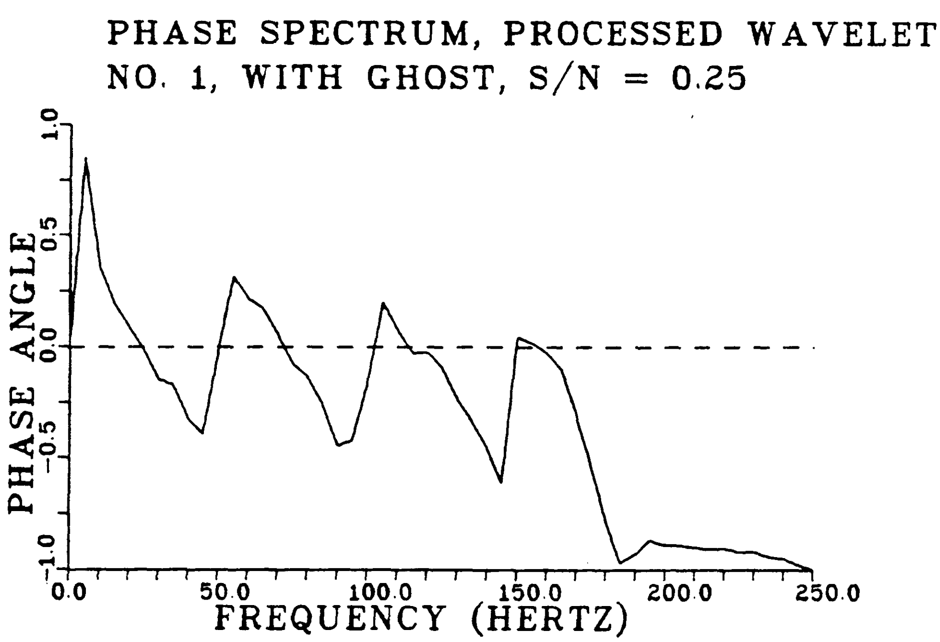

PHASE SPECTRUM, PROCESSED WAVELET
NO. 1, WITH GHOST, S/N = 0.5
PHASE ANGLE
-1.0
-0.5
0.0
0.5
1.0
0.0
50.0
100.0
150.0
200.0
250.0
FREQUENCY (HERTZ)

(b)

Figure 72


AMPLITUDE SPECTRUM, PROCESSED WAVELET
NO. 1, WITH GHOST, S/N = 0.25
AMPLITUDE
0.0
0.5
1.0
0.0
50.0
100.0
150.0
200.0
250.0
FREQUENCY (HERTZ)

(a)


PHASE SPECTRUM, PROCESSED WAVELET
NO. 1, WITH GHOST, S/N = 0.25
PHASE ANGLE
-1.0
-0.5
0.0
0.5
1.0
0.0
50.0
100.0
150.0
200.0
250.0
FREQUENCY (HERTZ)

(b)

Figure 73

however, to be as great as errors found upon analysis of the attenuated wavelet model.

For a signal-to-noise ratio of two, slope y-intercept analysis of Figure 70b indicates a seventy degree wavelet with a three millisecond delay. As the signal-to-noise ratio decreases to 0.25, the wavelet is shifted by ninety degrees and delayed by ten milliseconds. This result would be expected for an odd impulse pair with twenty millisecond separation.

Twelve individual wavelets processed with different random noise realizations are generated for the four signal-to-noise ratios. Each ensemble is averaged to simulate common depth point stacking. Figures 74, 75, 76 and 77 show the twelve component wavelets, the stacked wavelet, and the uncorrupted, desired result for respective signal-to-noise ratios of 2.0, 1.0, 0.5, and 0.25. Clearly, none of the component wavelets nor the stacked wavelets approximate the desired deconvolutional result very well. In fact, signal-to-noise ratios of up to 8.0 are required to successfully deconvolve the ghost out of the wavelet and approximate a zero phase result. Figure 78 demonstrates this. The ghost is successfully removed, and the wavelet is very nearly zero phase. Examination of the amplitude and phase spectra of the preceding wavelet, plotted in Figure 79, indicates the notches are nearly eliminated from the amplitude spectrum and the phase errors throughout the bandwidth are minimal. The wavelet is phase shifted

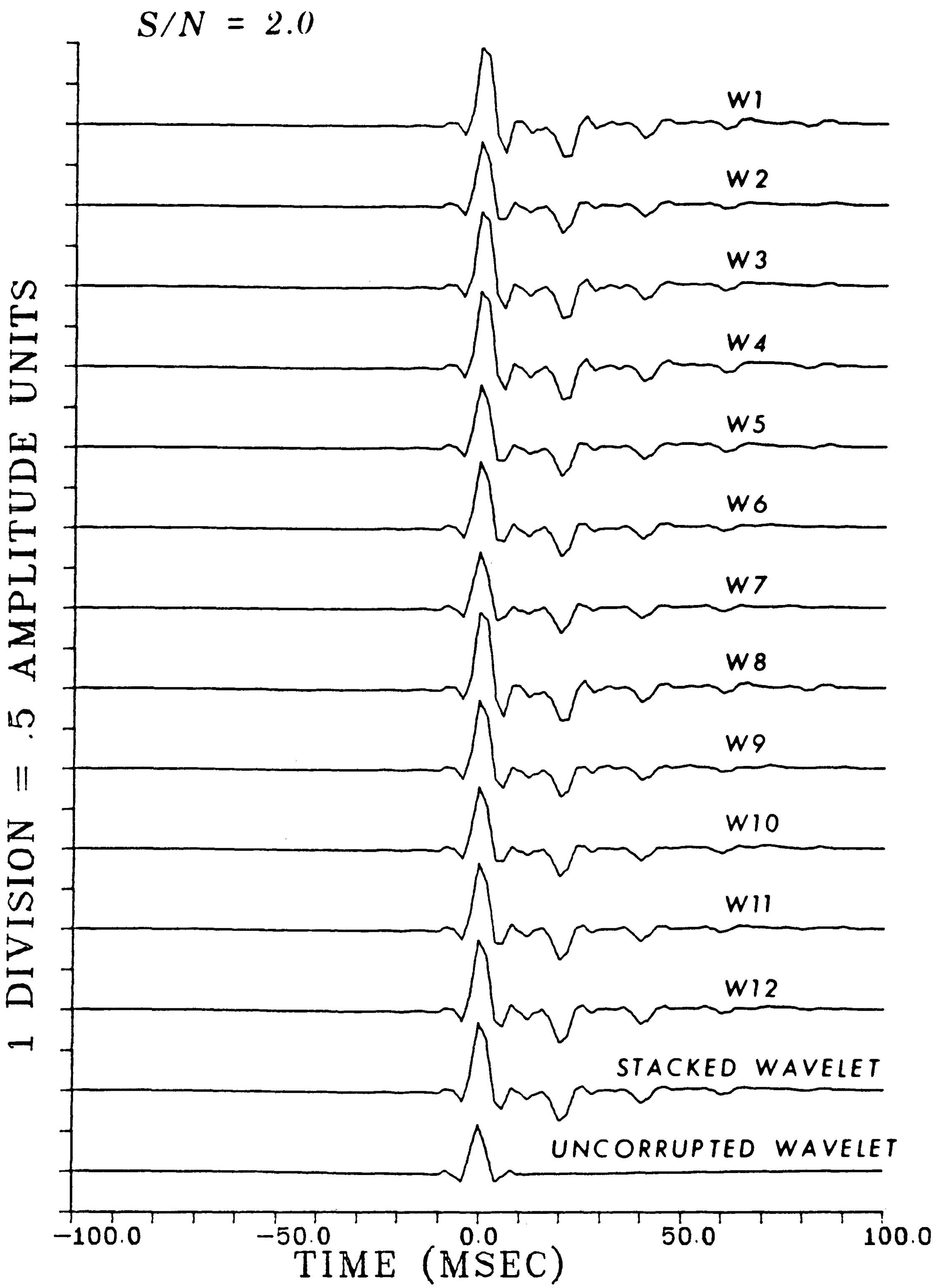


Figure 74 - Comparison of ideal vs. noise corrupted results, wavelet with ghost model

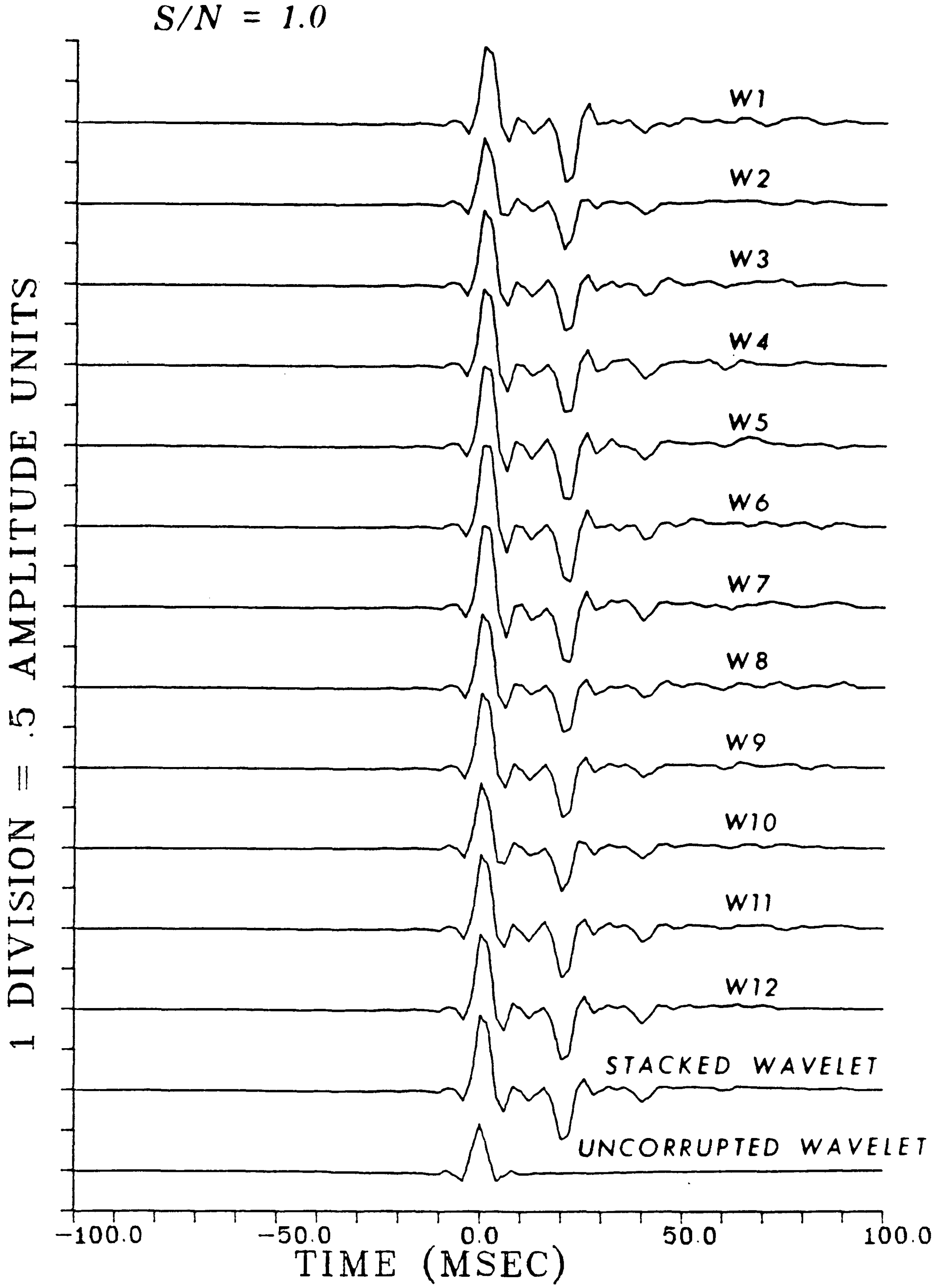


Figure 75 - Comparison of ideal vs. noise corrupted results, wavelet with ghost model

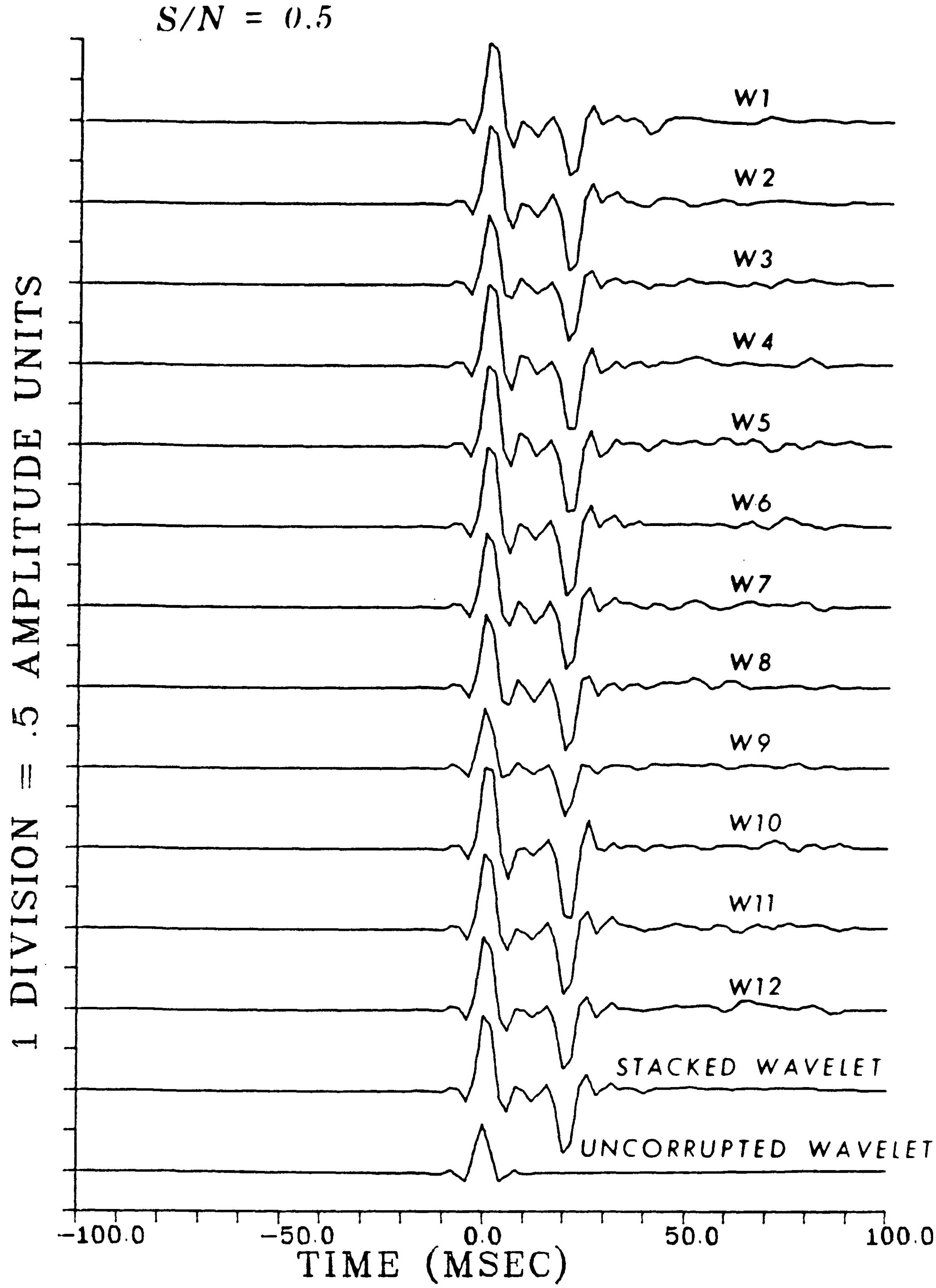


Figure 76 - Comparison of ideal vs. noise corrupted results, wavelet with ghost model

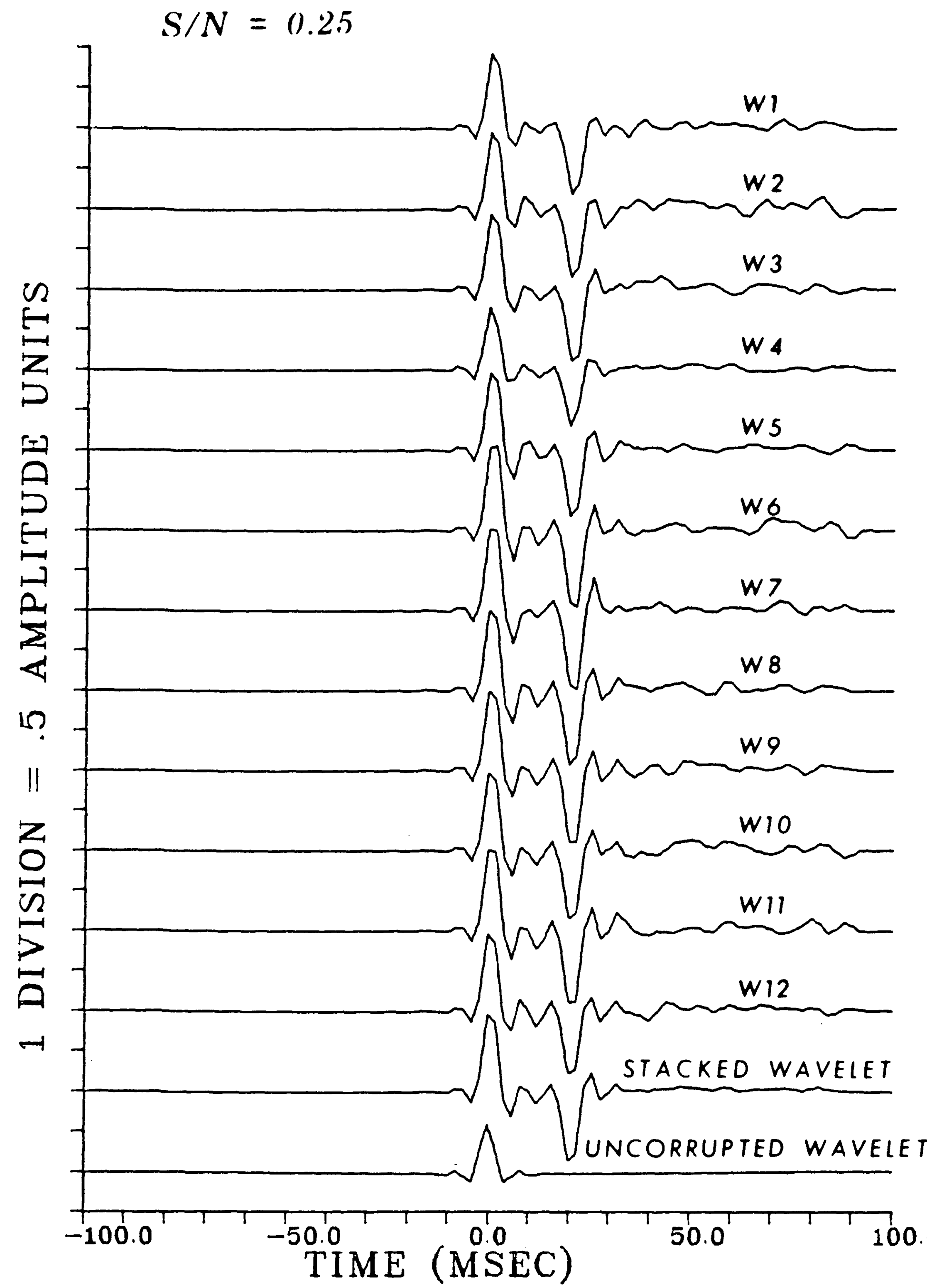


Figure 77 - Comparison of ideal vs. noise corrupted results, wavelet with ghost model

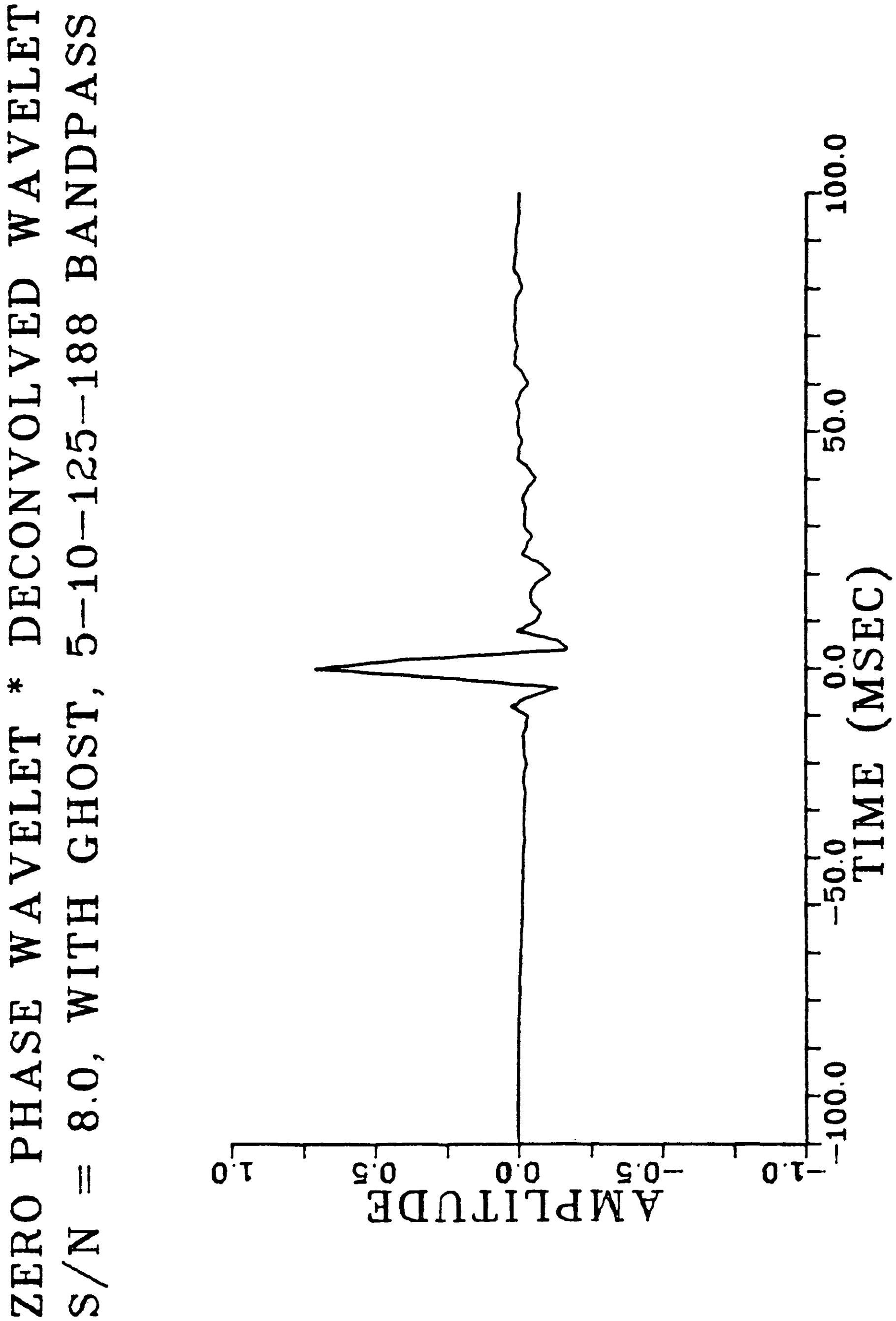
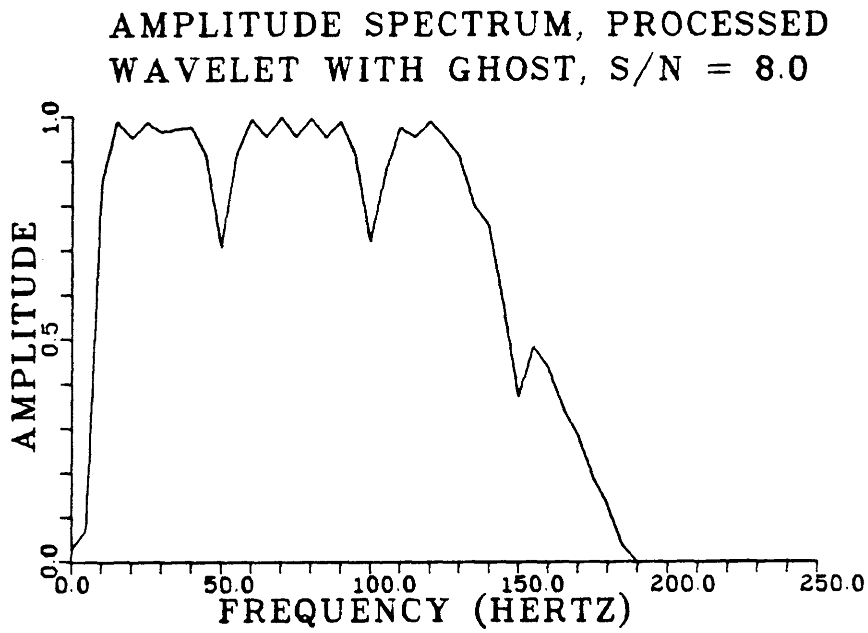
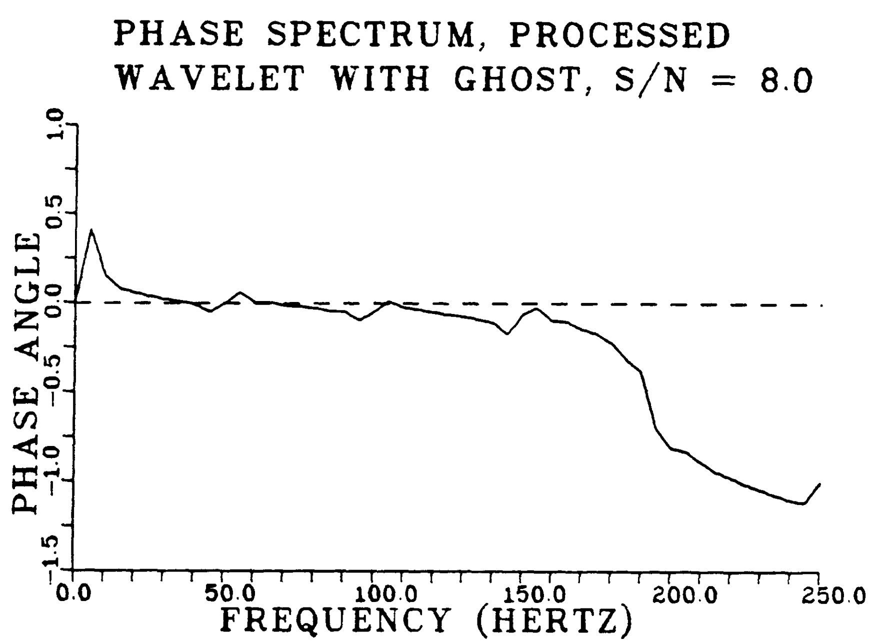

ZERO PHASE WAVELET * DECONVOLVED WAVELET
S/N = 8.0, WITH GHOST, 5-10-125-188 BANDPASS
-100.0
-50.0
0.0
50.0
100.0
TIME (MSEC)
-1.0
-0.5
0.0
0.5
1.0
AMPLITUDE

Figure 78


AMPLITUDE SPECTRUM, PROCESSED
WAVELET WITH GHOST, S/N = 8.0
AMPLITUDE
1.0
0.5
0.0
0.0
50.0
100.0
150.0
200.0
250.0
FREQUENCY (HERTZ)

(a)


PHASE SPECTRUM, PROCESSED
WAVELET WITH GHOST, S/N = 8.0
PHASE ANGLE
1.0
0.5
0.0
-0.5
-1.0
-1.5
0.0
50.0
100.0
150.0
200.0
250.0
FREQUENCY (HERTZ)

(b)

Figure 79

approximately 20 degrees, but the time lag is insignificant, perhaps one fifth of a millisecond.

Enlarged versions of the stacked wavelets shown in Figures 74 through 77, along with the wavelet amplitude and phase spectra, are plotted in Figures 80 through 87. The wavelets and spectra are much smoother than the unstacked component equivalents, but the overall curve shapes are much the same. As before, phase error is an inverse function of signal-to-noise ratio.

Figure 88 is a graph of phase error in degrees versus the signal-to-noise ratio for the wavelet with a ghost model. For the ninety Hertz curve, a phase error difference of forty degrees is observed. However, within the bandwidth, aboslute phase error differences are more along the order of twenty degrees or less. The differences observed here are just slightly less, overall, than the differences observed with the attenuated wavelet model, but are certainly greater than those observed with the original wavelet model.

The interpretation here, as with the attenuated wavelet model, is that for two functions of different spectral shapes the deconvolution algorithm is confused about which process to act upon, making the reliability of the output questionable. The desirability of not allowing notches introduced into the spectrum to fill up with noise prior to deconvolution is certainly in evidence.

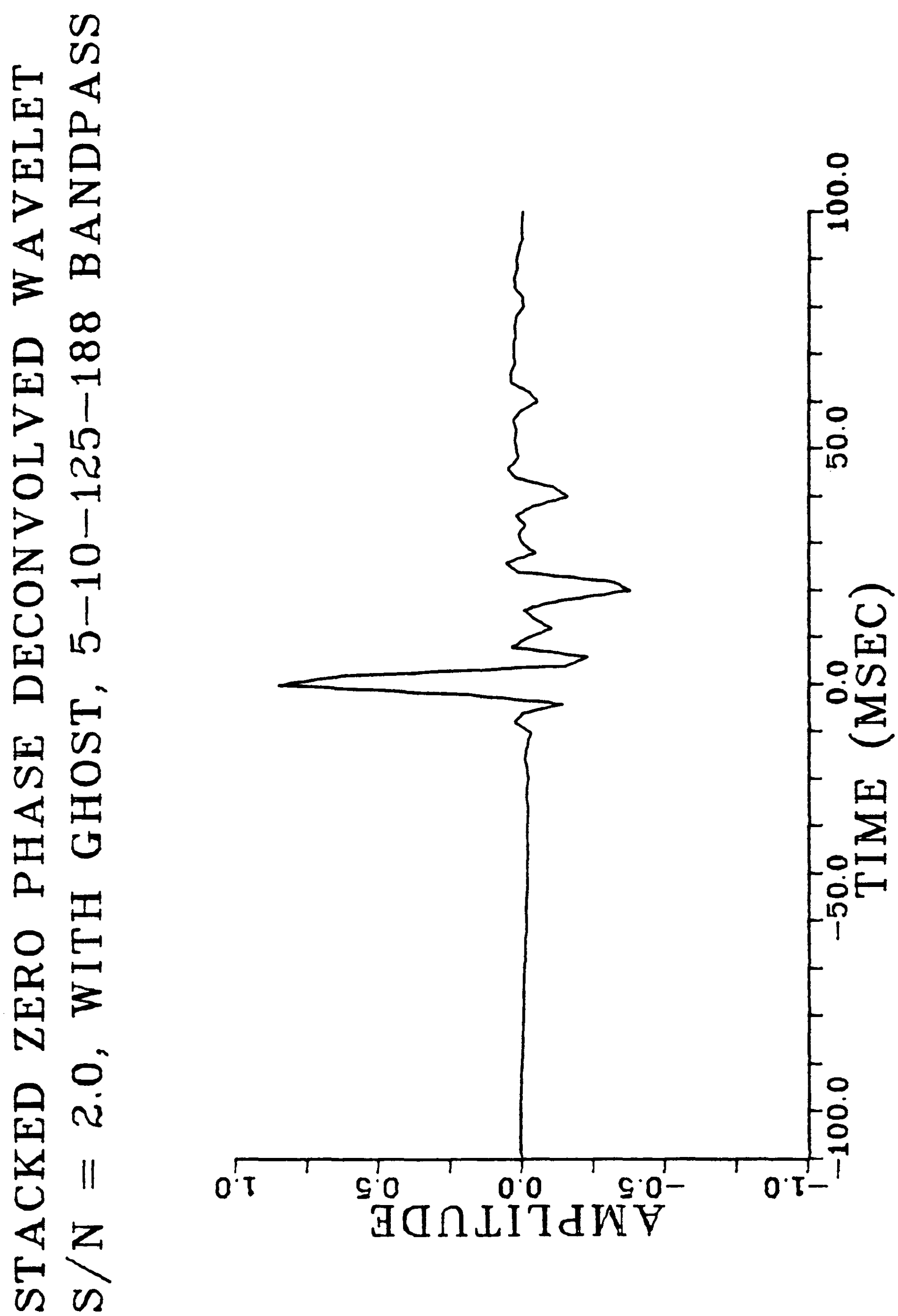
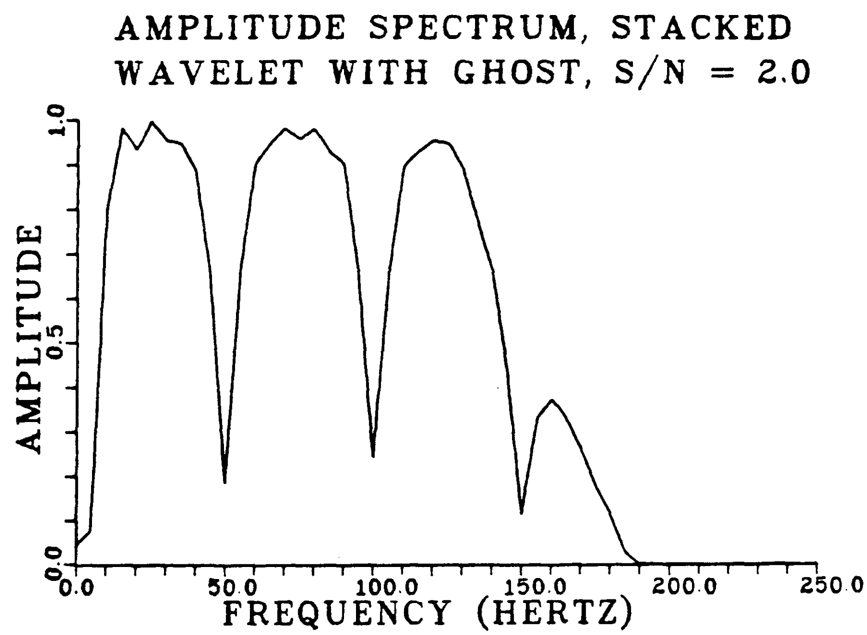
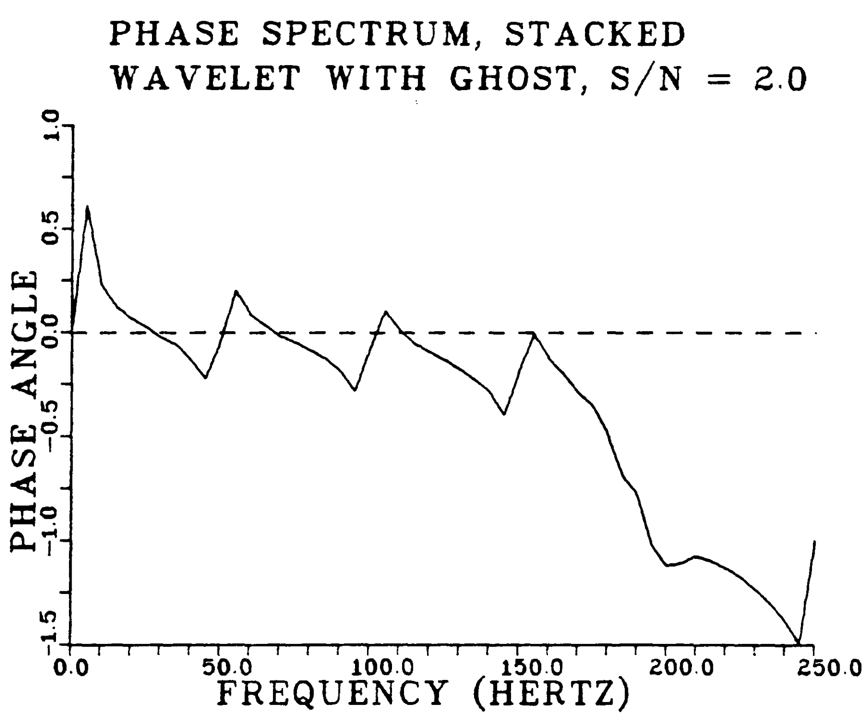

STACKED ZERO PHASE DECONVOLVED WAVELET
S/N = 2.0, WITH GHOST, 5–10–125–188 BANDPASS
AMPLITUDE
1.0
0.5
0.0
-0.5
-1.0
TIME (MSEC)
-100.0
-50.0
0.0
50.0
100.0

Figure 80


AMPLITUDE SPECTRUM, STACKED
WAVELET WITH GHOST, S/N = 2.0
AMPLITUDE
0.0
0.5
1.0
0.0
50.0
100.0
150.0
200.0
250.0
FREQUENCY (HERTZ)

(a)


PHASE SPECTRUM, STACKED
WAVELET WITH GHOST, S/N = 2.0
PHASE ANGLE
-1.5
-1.0
-0.5
0.0
0.5
1.0
0.0
50.0
100.0
150.0
200.0
250.0
FREQUENCY (HERTZ)

(b)

Figure 81

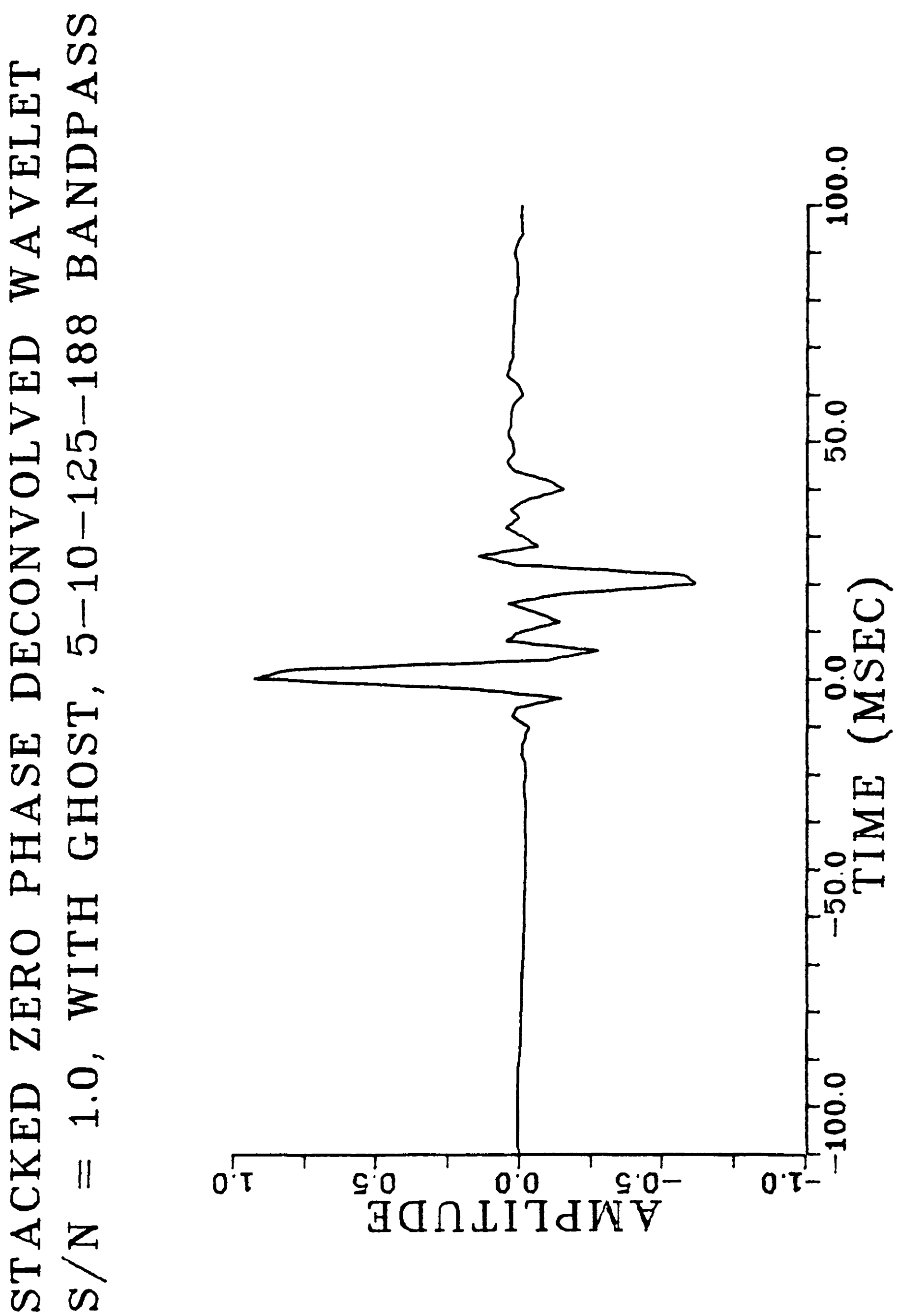
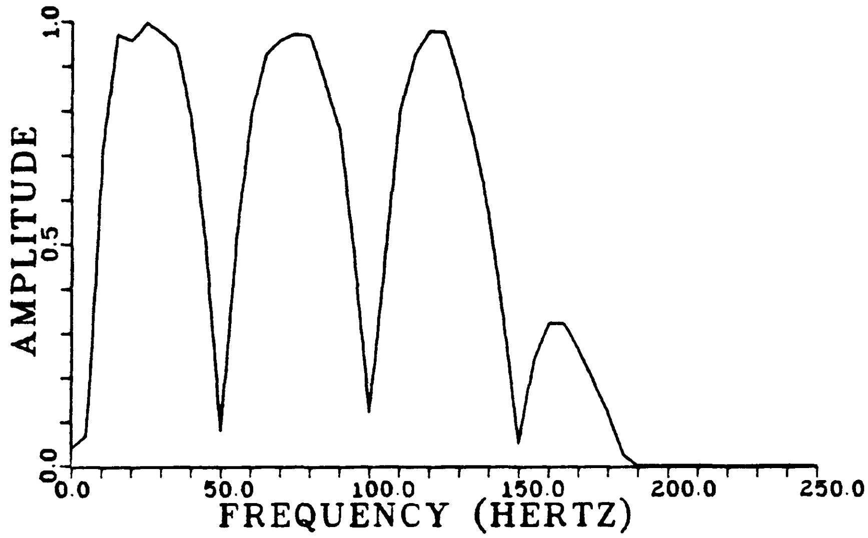

STACKED ZERO PHASE DECONVOLVED WAVELET
S/N = 1.0, WITH GHOST, 5–10–125–188 BANDPASS
AMPLITUDE
1.0
0.5
0.0
–0.5
–1.0
TIME (MSEC)
–100.0
–50.0
0.0
50.0
100.0

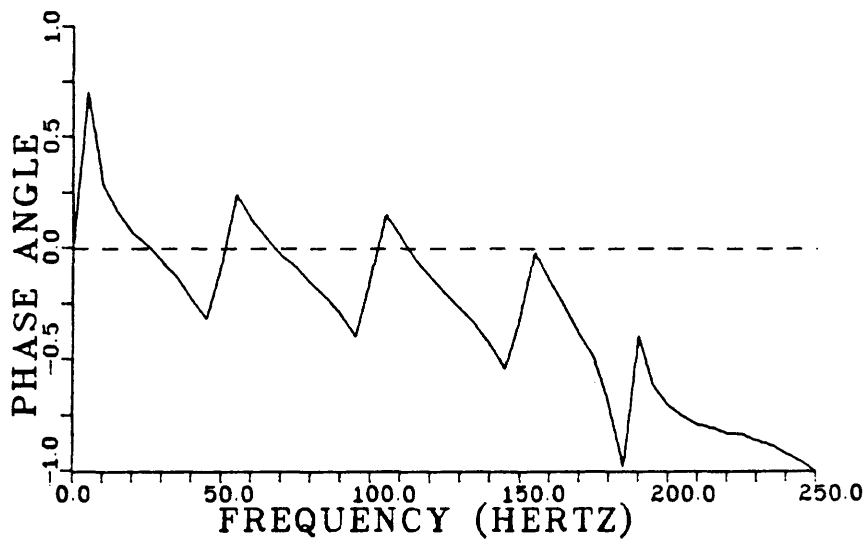
Figure 82

AMPLITUDE SPECTRUM, STACKED WAVELET WITH GHOST, S/N = 1.0



(a)

PHASE SPECTRUM, STACKED WAVELET WITH GHOST, S/N = 1.0



(b)

Figure 83

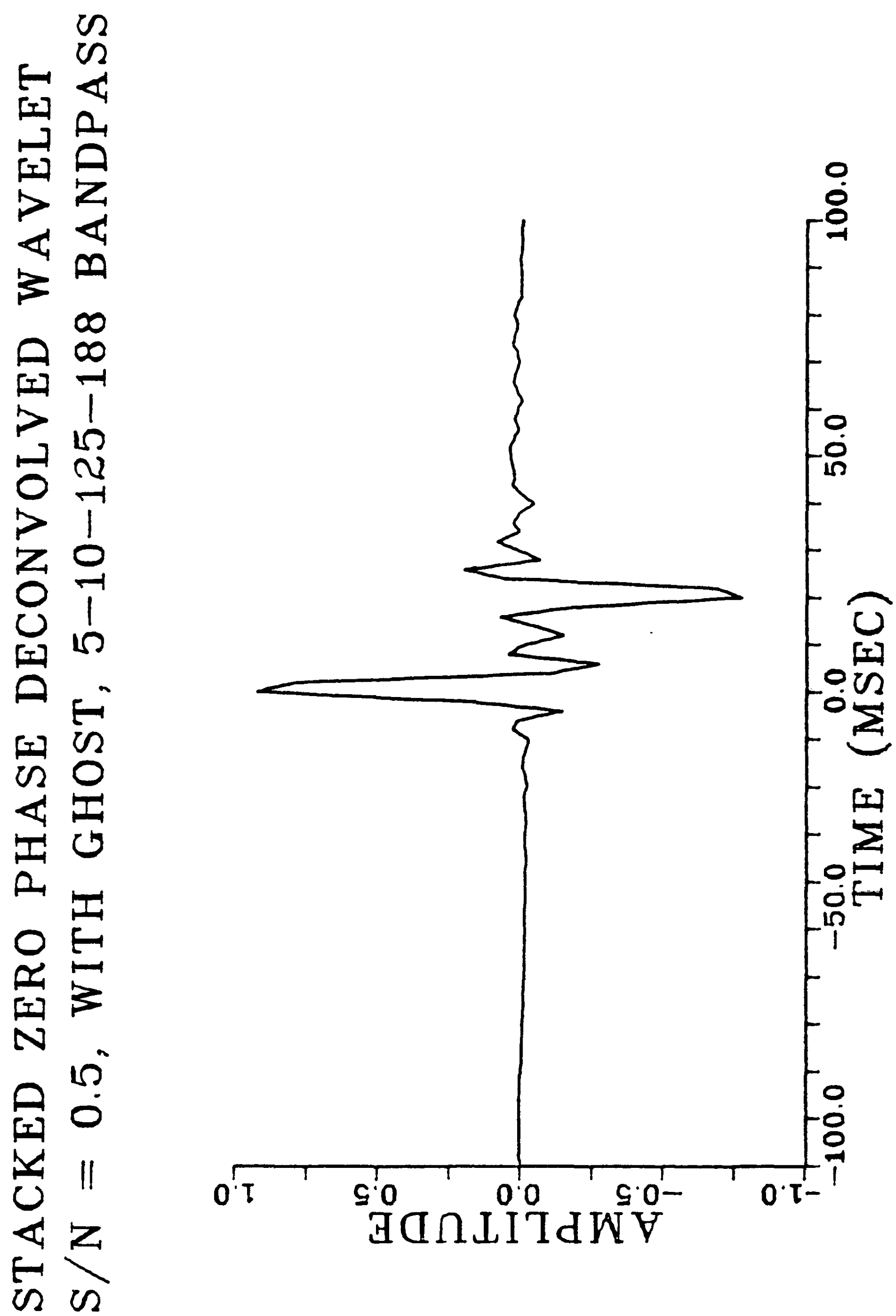
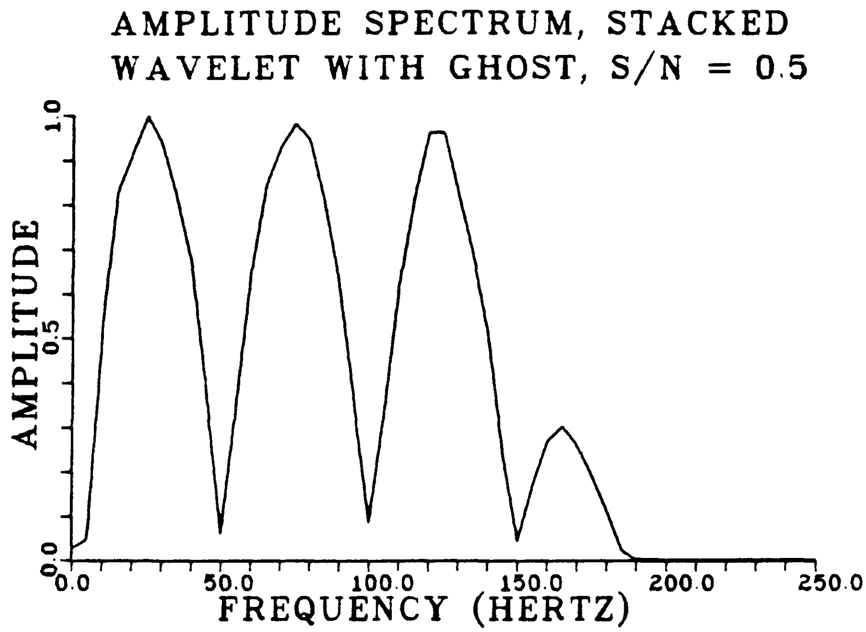
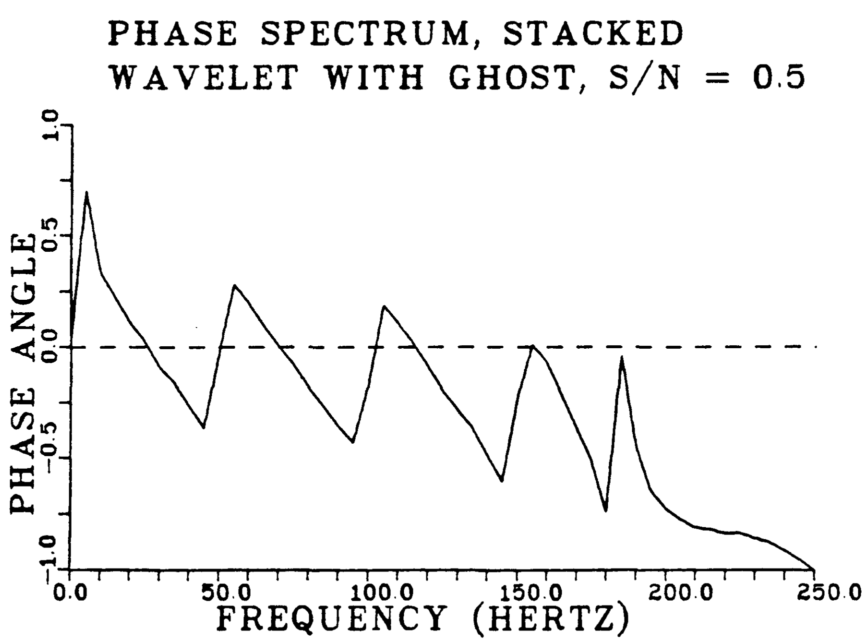

STACKED ZERO PHASE DECONVOLVED WAVELET
S/N = 0.5, WITH GHOST, 5-10-125-188 BANDPASS
AMPLITUDE
1.0
0.5
0.0
-0.5
-1.0
-100.0
-50.0
0.0
50.0
100.0
TIME (MSEC)

Figure 84


AMPLITUDE SPECTRUM, STACKED
WAVELET WITH GHOST, S/N = 0.5
AMPLITUDE
0.0
0.5
1.0
0.0
50.0
100.0
150.0
200.0
250.0
FREQUENCY (HERTZ)

(a)


PHASE SPECTRUM, STACKED
WAVELET WITH GHOST, S/N = 0.5
PHASE ANGLE
-1.0
-0.5
0.0
0.5
1.0
0.0
50.0
100.0
150.0
200.0
250.0
FREQUENCY (HERTZ)

(b)

Figure 85

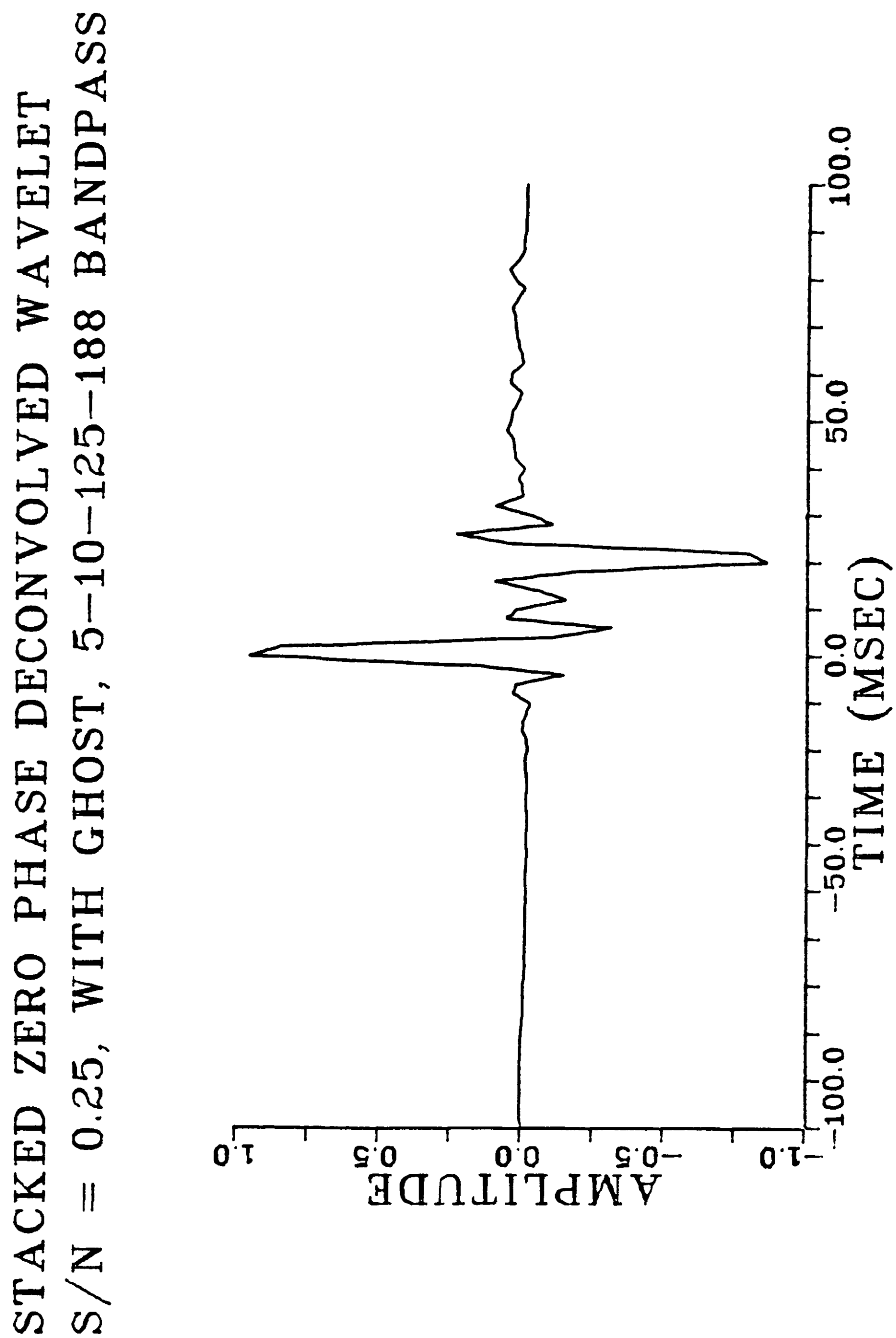
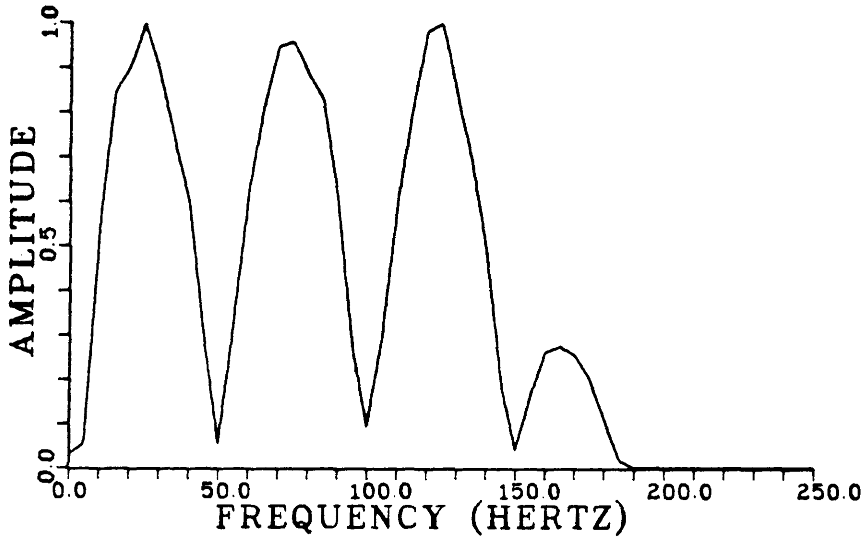

STACKED ZERO PHASE DECONVOLVED WAVELET
S/N = 0.25, WITH GHOST, 5-10-125-188 BANDPASS
AMPLITUDE
1.0
0.5
0.0
-0.5
-1.0
TIME (MSEC)
-100.0
-50.0
0.0
50.0
100.0

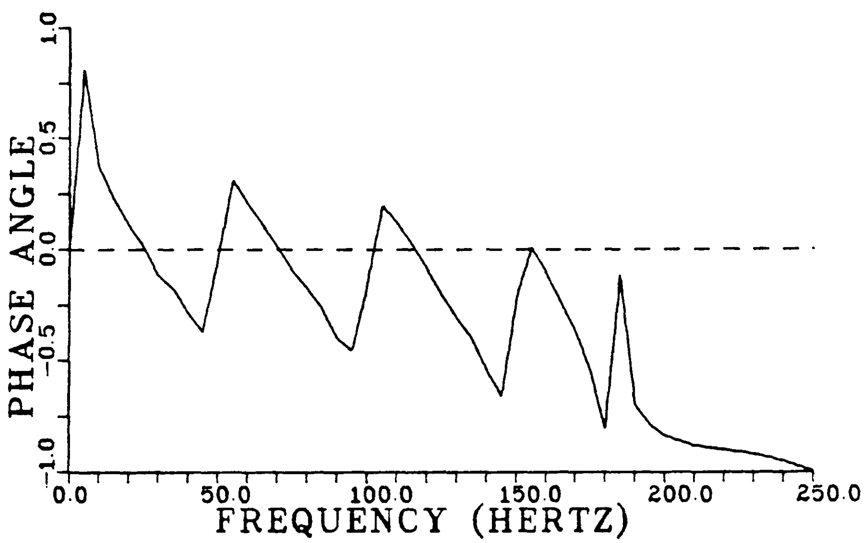
Figure 86

AMPLITUDE SPECTRUM, STACKED WAVELET WITH GHOST, S/N = 0.25



(a)

PHASE SPECTRUM, STACKED WAVELET WITH GHOST, S/N = 0.25



(b)

Figure 87

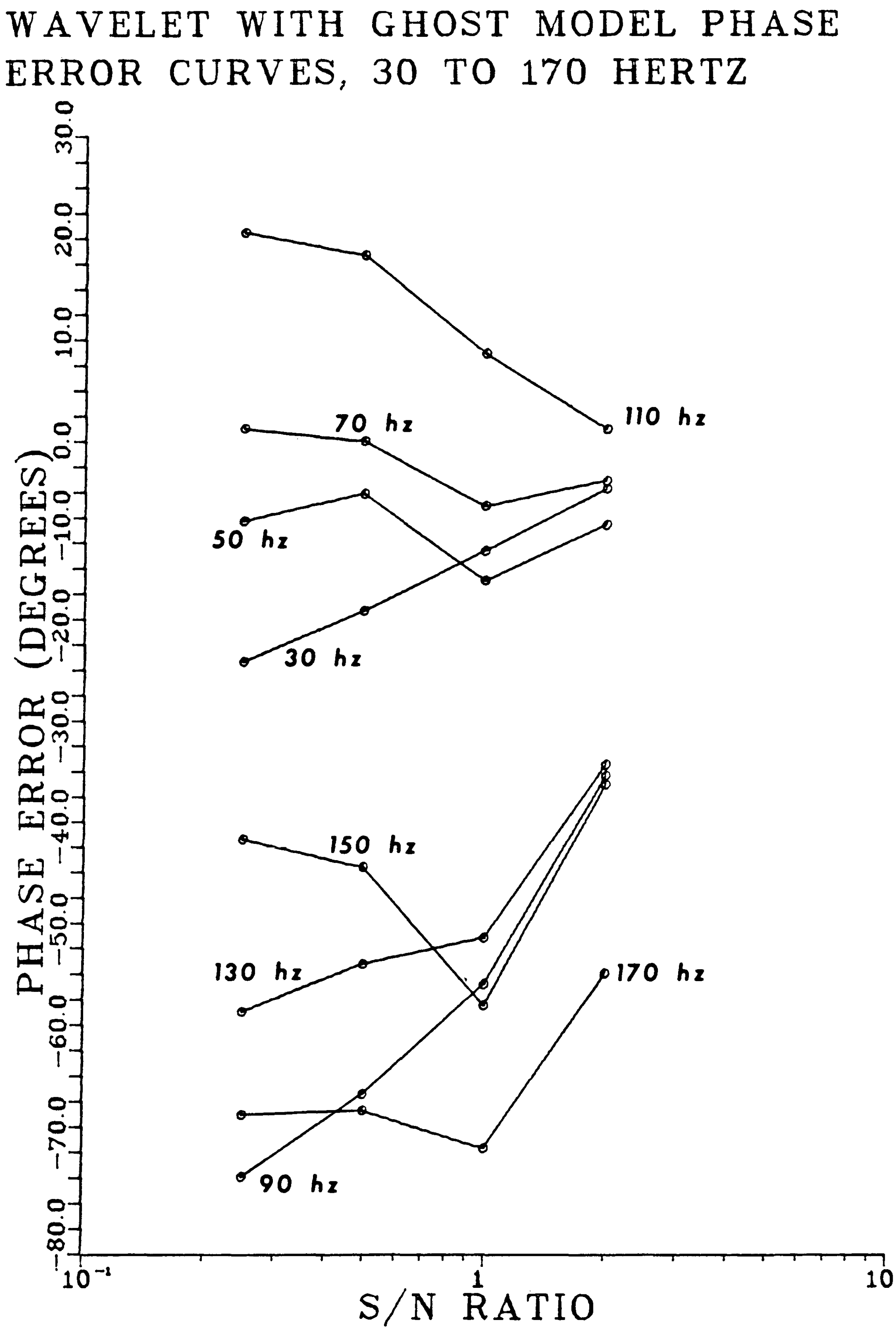

WAVELET WITH GHOST MODEL PHASE
ERROR CURVES, 30 TO 170 HERTZ
PHASE ERROR (DEGREES)
30.0
20.0
10.0
0.0
-10.0
-20.0
-30.0
-40.0
-50.0
-60.0
-70.0
-80.0
110 hz
70 hz
50 hz
30 hz
150 hz
130 hz
170 hz
90 hz
10^{-1}
1
10
S/N RATIO

Figure 88

The previously described phenomenon is indeed a problem because prior prediction of spectral notches is impossible or at best very difficult. However, instrument and geophone response rolloffs in dB per octave tend to yield a high order zero at DC. Consequently, this notch is typically predictable and should be forced into the spectrum by preconditioning.

The standard deviations for each twelve fold stack are plotted in Figures 89, 90, 91, and 92. A general increase in standard deviation with frequency is observed on all the curves. Dependence upon signal-to-noise ratio is only slight, as shown by relative comparison of the four figures.

Large, instantaneous standard deviations are observed every fifty Hertz, where the notches occur. This demonstrates the discontinuous nature of the ghost wavelet phase spectrum is potentially very troublesome. Marine sections are made up of many field traces, each from a different record. In common practice, the towing depth of the source will not remain constant, and the spike separations of the ghosts will vary from record to record. This implies the elements of a stacked trace will have phase discontinuities in different places, resulting in the stacked trace being of lesser amplitude. Superposition of phase errors resulting from deconvolution in the presence of noise can only amplify this problem.

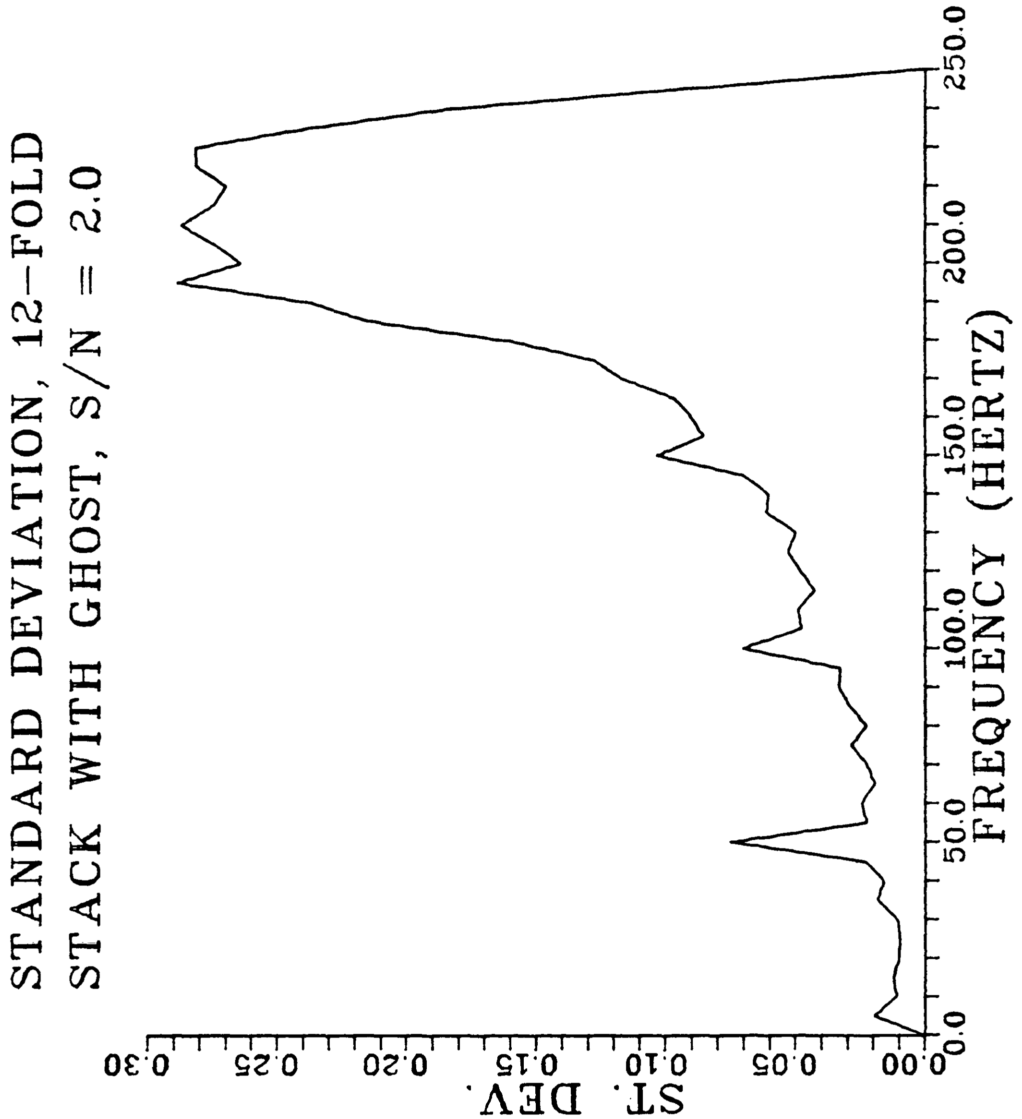

STANDARD DEVIATION, 12—FOLD
STACK WITH GHOST, S/N = 2.0
FREQUENCY (HERTZ)
ST. DEV.
0.0
50.0
100.0
150.0
200.0
250.0
0.00
0.05
0.10
0.15
0.20
0.25
0.30

Figure 89

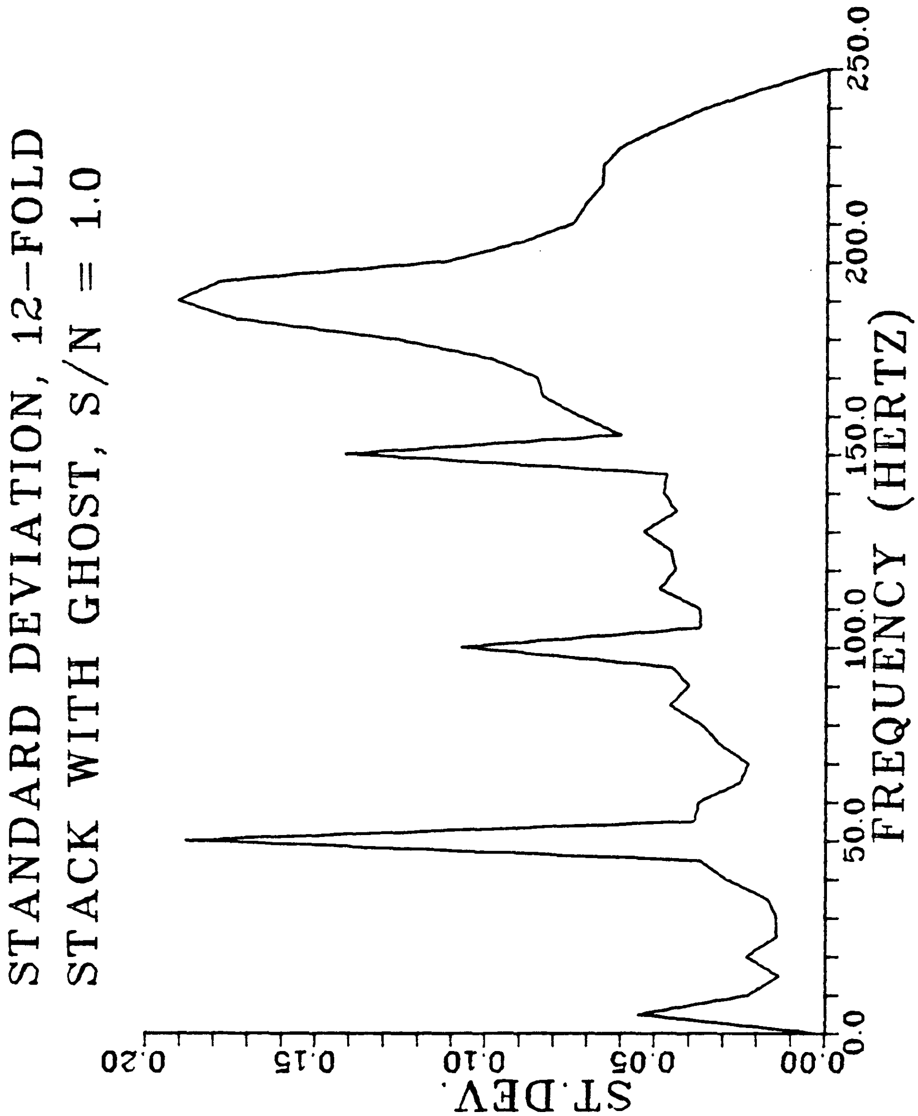

STANDARD DEVIATION, 12—FOLD
STACK WITH GHOST, S/N = 1.0
FREQUENCY (HERTZ)
0.0
50.0
100.0
150.0
200.0
250.0
ST.DEV.
0.00
0.05
0.10
0.15
0.20

Figure 90

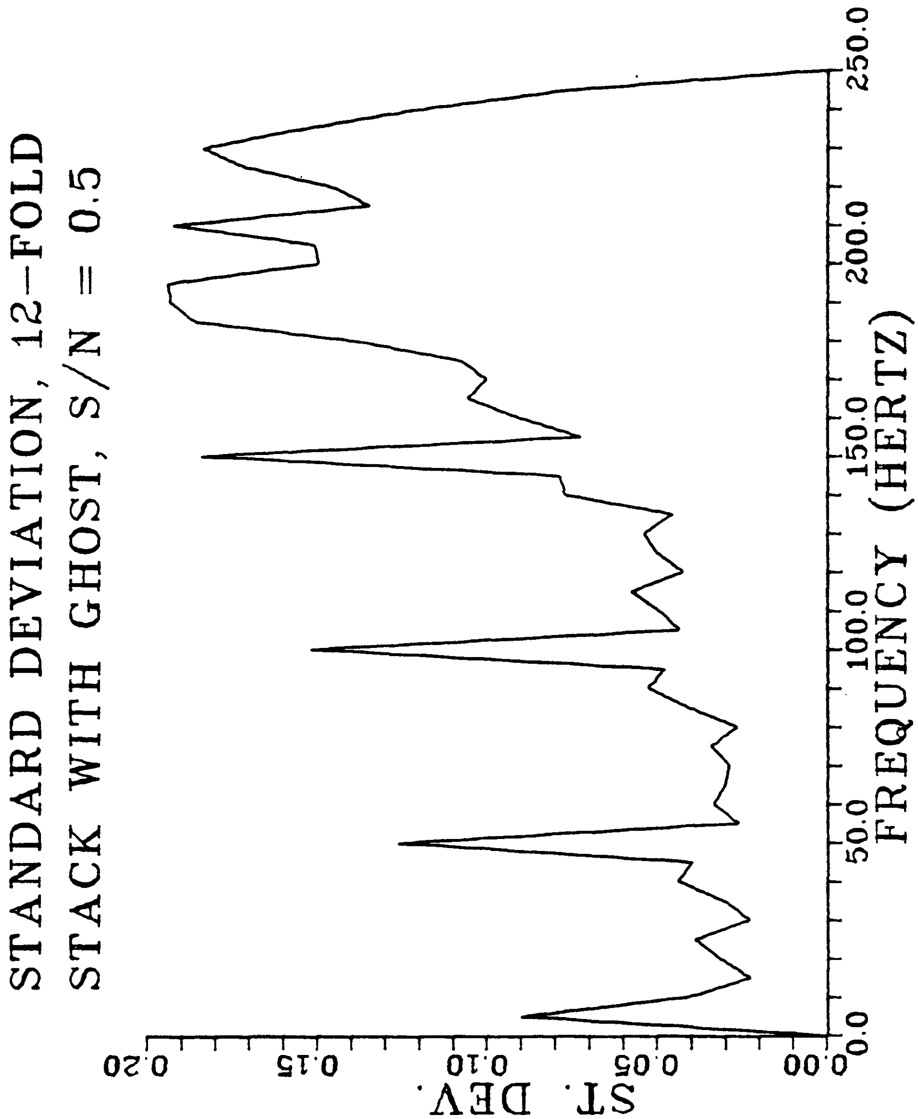

STANDARD DEVIATION, 12—FOLD
STACK WITH GHOST, S/N = 0.5
FREQUENCY (HERTZ)
ST. DEV.
0.0
50.0
100.0
150.0
200.0
250.0
0.00
0.05
0.10
0.15
0.20

Figure 91

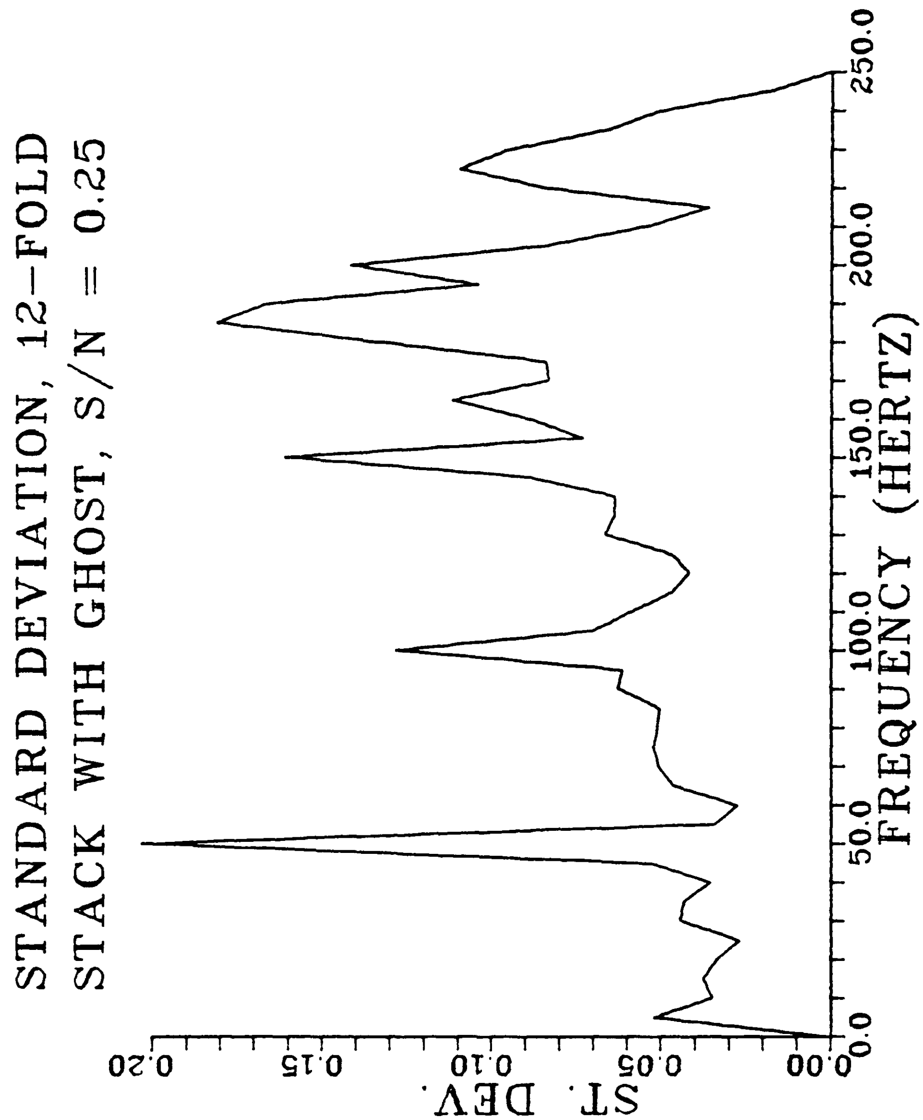

STANDARD DEVIATION, 12−FOLD
STACK WITH GHOST, S/N = 0.25
ST. DEV.
0.00
0.05
0.10
0.15
0.20
FREQUENCY (HERTZ)
0.0
50.0
100.0
150.0
200.0
250.0

Figure 92

The slight signal-to-noise ratio and frequency dependence of standard deviation is shown in Figure 93. Within the frequency bandwidth, the standard deviations are small, as are the standard deviation differences, even throughout a signal-to-noise ratio improvement of eighteen dB. The exception to this generality is the fifty Hertz curve, which shows highly erratic deviations of up to thirty-five degrees. Clearly, this results from a notch and associated phase discontinuity occurring at fifty Hertz.

WAVELET WITH GHOST MODEL STANDARD DEVIATION CURVES, 30 TO 170 HERTZ

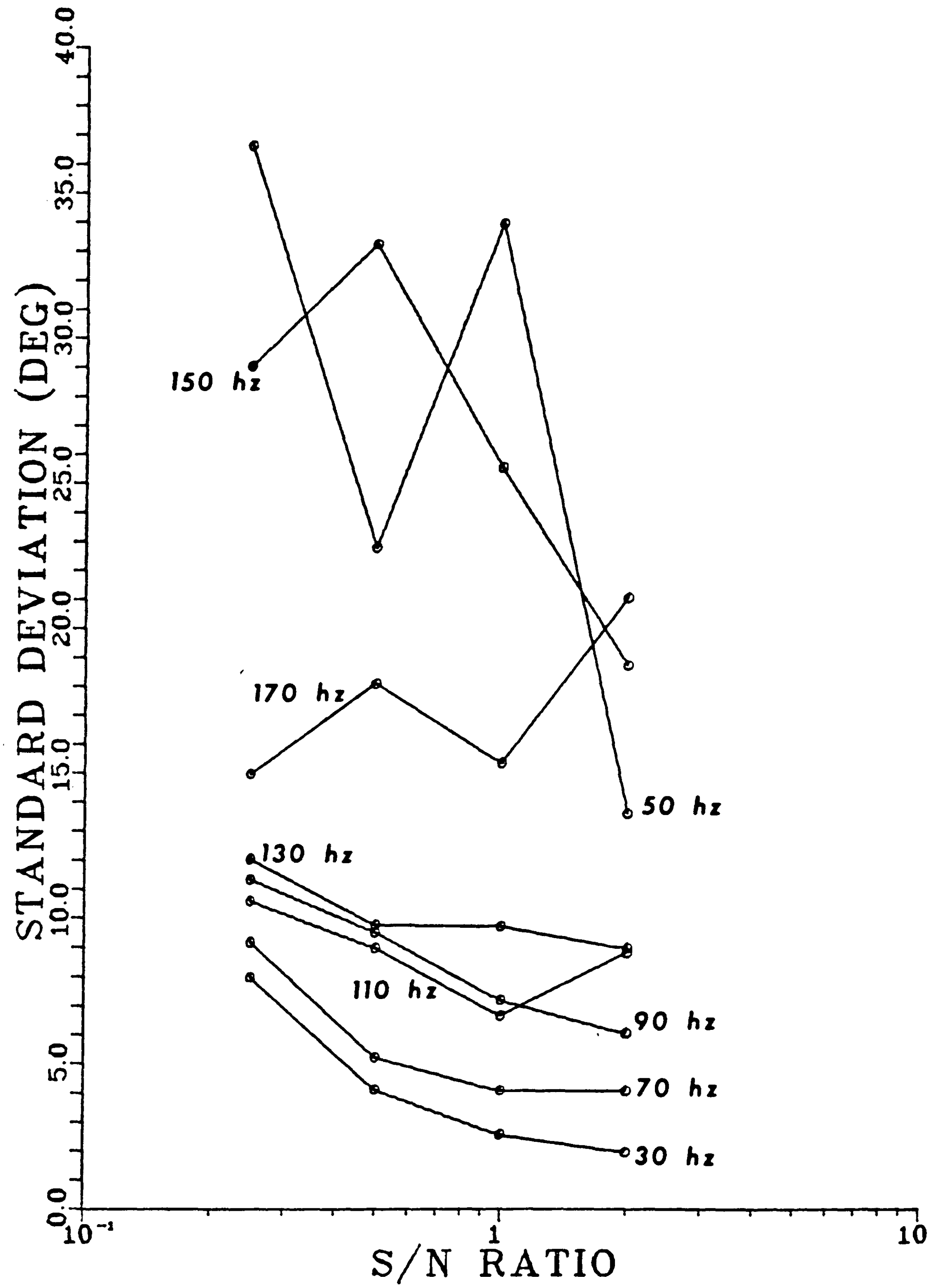


Figure 93

WHITE NOISE MODEL

The matrix Wiener-Levinson deconvolution algorithm contains a system of normal equations that are, in some instances, ill-conditioned. Westlake (1968) described a square matrix as being ill-conditioned when it is almost singular, that is, whenever a given row or column is nearly a linear combination of one or more of the remaining rows or columns. Given a singular matrix, the Wiener filter is very sensitive to the accuracy of the computer, and often yields unreliable solutions to the normal equations.

To resolve this problem, the zero lag value of the autocorrelation coefficient sequence r_o is replaced by a slightly larger term r_o', where

$$r_o' = r_o + \Delta r_o. \qquad (30)$$

this industry practice, called "prewhitening", is almost invariably applied during deconvolution, because it is more cost and time efficient for the computer to apply prewhitening rather than analytically determine if a system of normal equations are ill-conditioned. The quantity, P, where

$$P = 100 \frac{\Delta r_o}{r_o}, \qquad (31)$$

is used as a percent measure of the prewhitening level applied. Obviously, the replacement of r_o by r_o' results in a different system of normal equations, but the solution to this modified system produces a more reliable sequence of filter coefficients.

The wavelet, attenuated wavelet, and wavelet with ghost models are prewhitened, deconvolved, and analyzed with regard to phase errors. Two prewhitening levels, one percent and one-tenth of one percent, are used.

The two prewhitening levels are added to the zero-lag values of the autocorrelation matrices of the three different models, and whitened minimum phase wavelets are designed. The whitened minimum phase wavelets for the models are then deconvolved by the respective nonwhitened minimum phase inverses. The deconvolutional results are subsequently zero phase bandpass filtered.

The output of this process for the wavelet model with 0.1 percent prewhitening is shown in Figure 94. The phase is delayed in time and slightly assymetrical. Examination of the spectra of the wavelet (Figure 95) indicates the amplitude is remarkably white and uncorrupted, and the phase is minimum. An eight-degree wavelet delayed 0.7 milliseconds is calculated from the phase spectra. For the one percent prewhitened case, the wavelet (Figure 96), is again delayed in time, but slightly more assymetrical and noisy than the previous example. The phase lag spectrum (Figure 97b) is again minimum, but has more associated error. A twelve degree wavelet delayed 1.3 milliseconds is indicated.

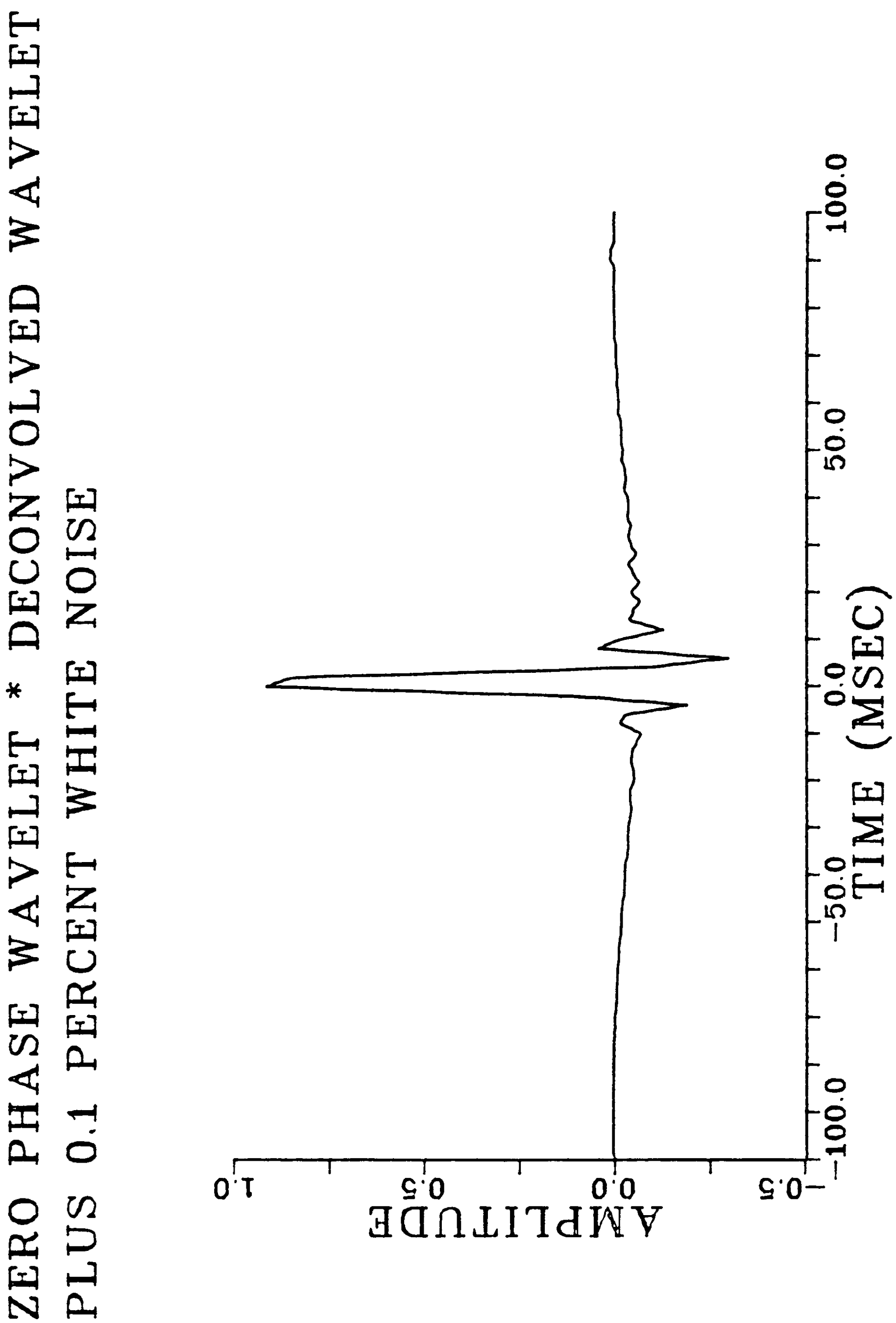
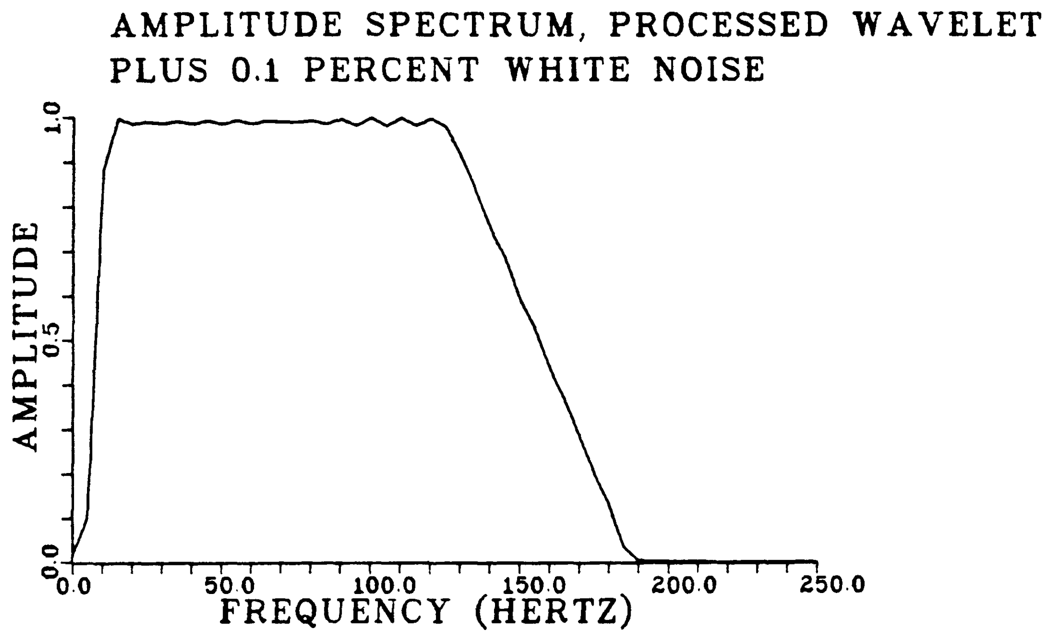
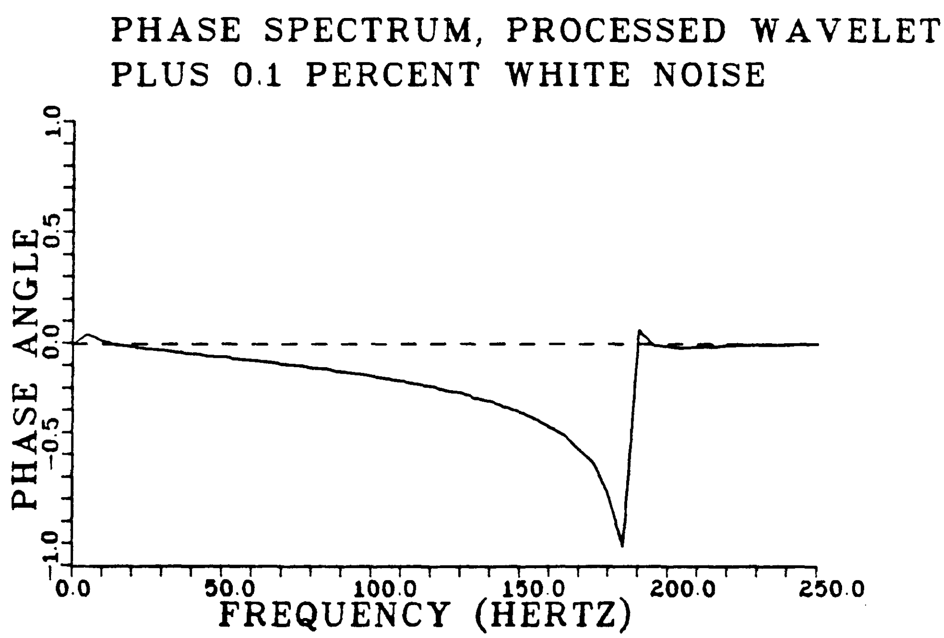

ZERO PHASE WAVELET * DECONVOLVED WAVELET
PLUS 0.1 PERCENT WHITE NOISE
AMPLITUDE
1.0
0.5
0.0
-0.5
TIME (MSEC)
-100.0
-50.0
0.0
50.0
100.0

Figure 94


AMPLITUDE SPECTRUM, PROCESSED WAVELET
PLUS 0.1 PERCENT WHITE NOISE
AMPLITUDE
1.0
0.5
0.0
0.0
50.0
100.0
150.0
200.0
250.0
FREQUENCY (HERTZ)

(a)


PHASE SPECTRUM, PROCESSED WAVELET
PLUS 0.1 PERCENT WHITE NOISE
PHASE ANGLE
1.0
0.5
0.0
-0.5
-1.0
0.0
50.0
100.0
150.0
200.0
250.0
FREQUENCY (HERTZ)

(b)

Figure 95

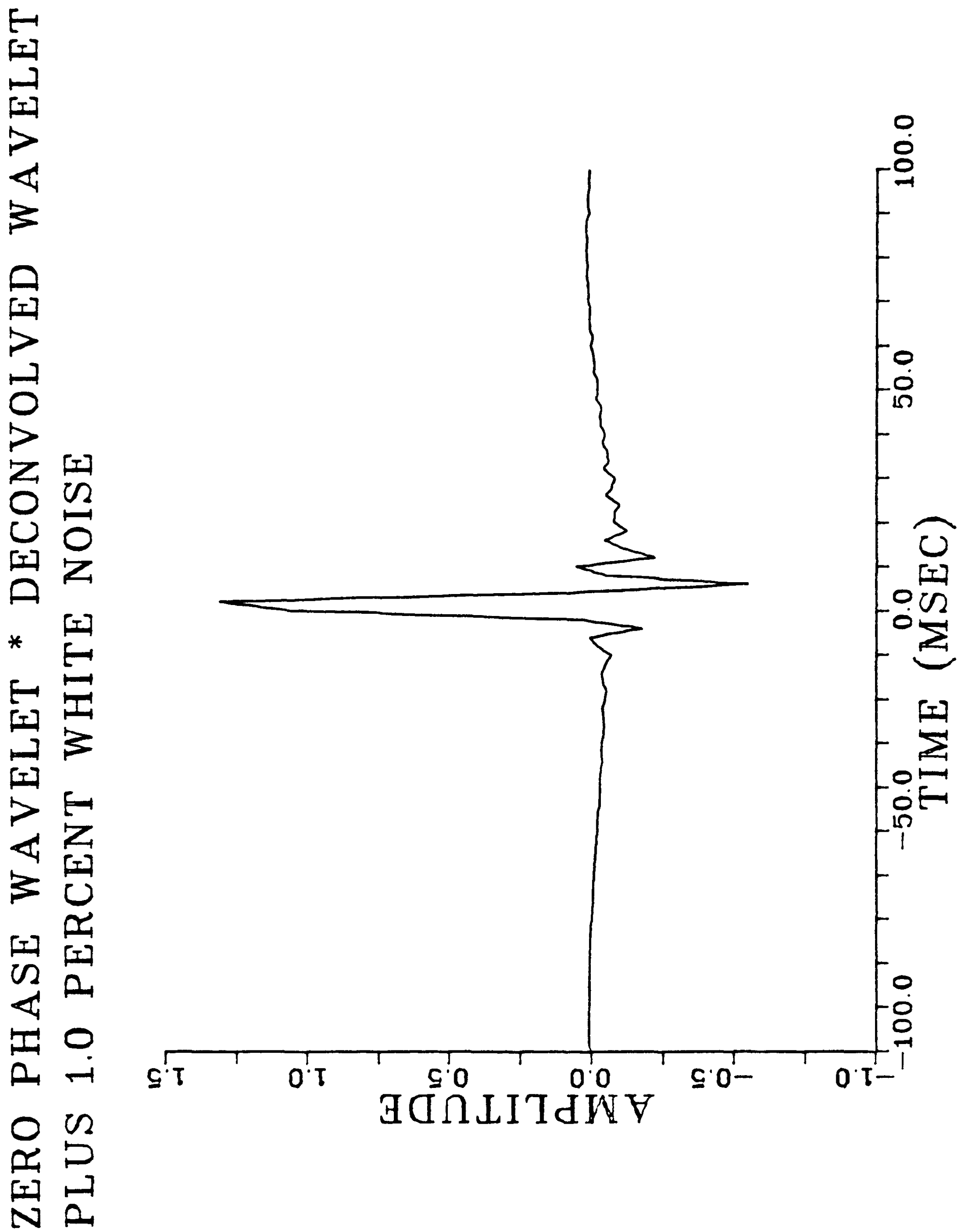
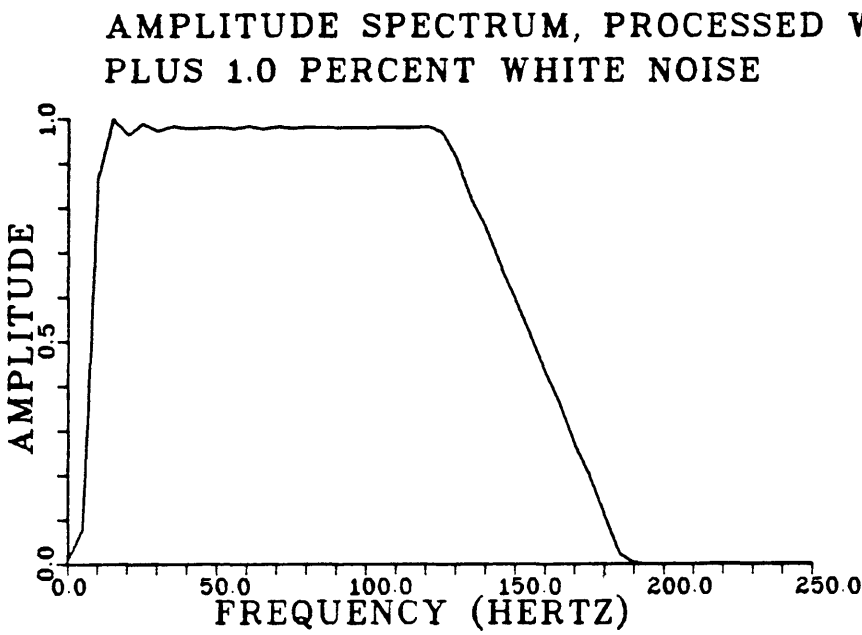
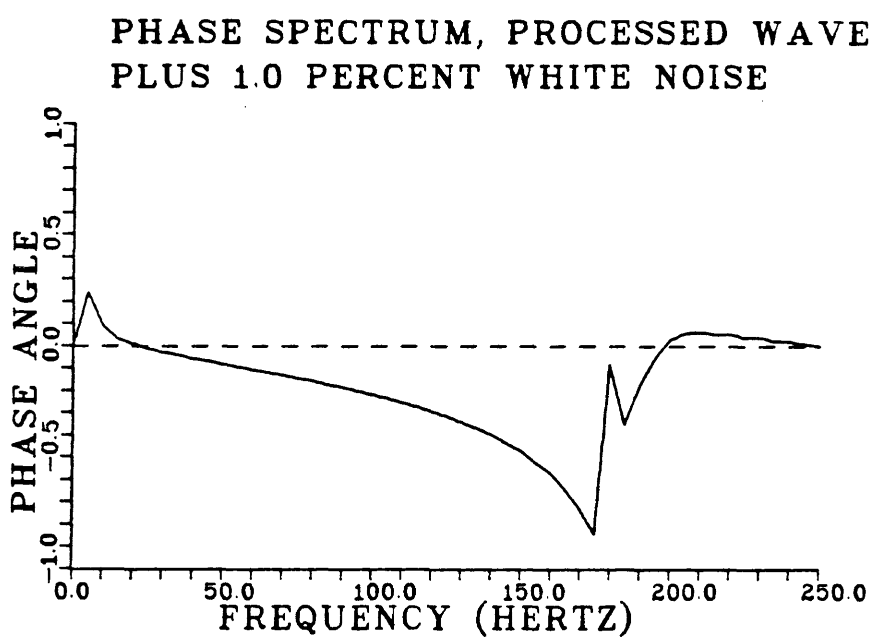

ZERO PHASE WAVELET * DECONVOLVED WAVELET
PLUS 1.0 PERCENT WHITE NOISE
AMPLITUDE
1.5
1.0
0.5
0.0
-0.5
-1.0
TIME (MSEC)
-100.0
-50.0
0.0
50.0
100.0

Figure 96


AMPLITUDE SPECTRUM, PROCESSED WAVELET
PLUS 1.0 PERCENT WHITE NOISE
AMPLITUDE
1.0
0.5
0.0
0.0
50.0
100.0
150.0
200.0
250.0
FREQUENCY (HERTZ)

(a)


PHASE SPECTRUM, PROCESSED WAVELET
PLUS 1.0 PERCENT WHITE NOISE
PHASE ANGLE
1.0
0.5
0.0
-0.5
-1.0
0.0
50.0
100.0
150.0
200.0
250.0
FREQUENCY (HERTZ)

(b)

Figure 97

Subsequent analysis of the attenuated wavelet and wavelet with ghost models yield results consistent with the conclusions reached for the wavelet model. These findings are depicted in Figures 98 through 105. Comparison of these figures reveals a general deterioration of the wavelet and an increase in phase error as the percentage of prewhitening is increased.

The results described here suggest only small percentage prewhitening levels be employed in deconvolution practice and such prewhitening techniques for normal equation stabilization should be applied with great care.

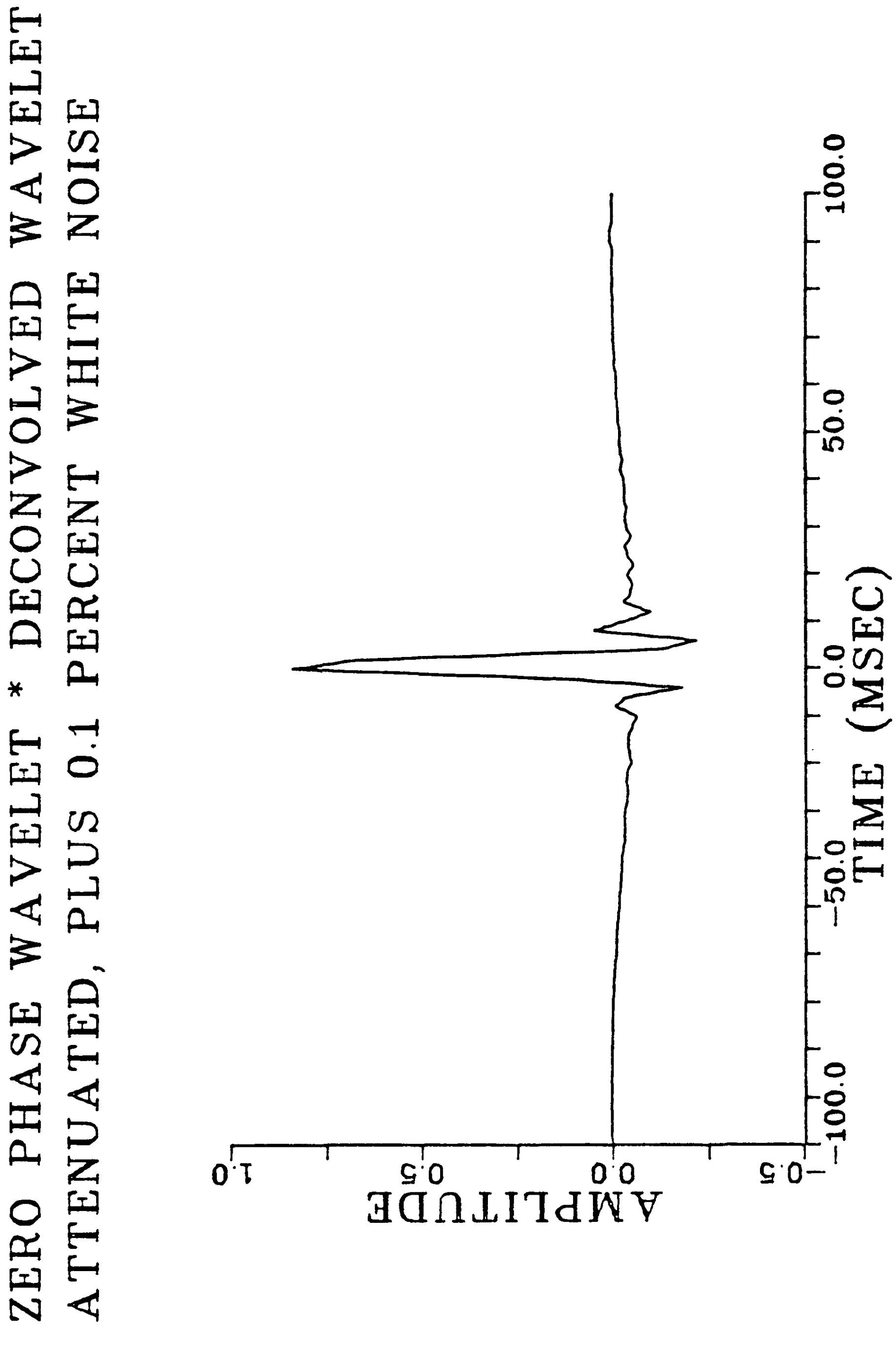
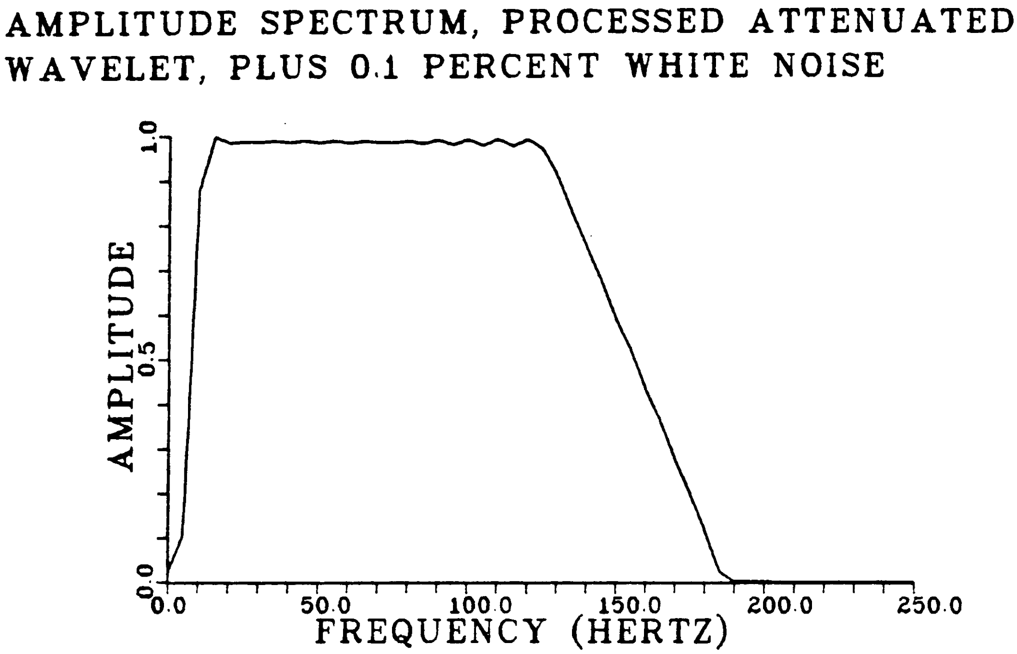
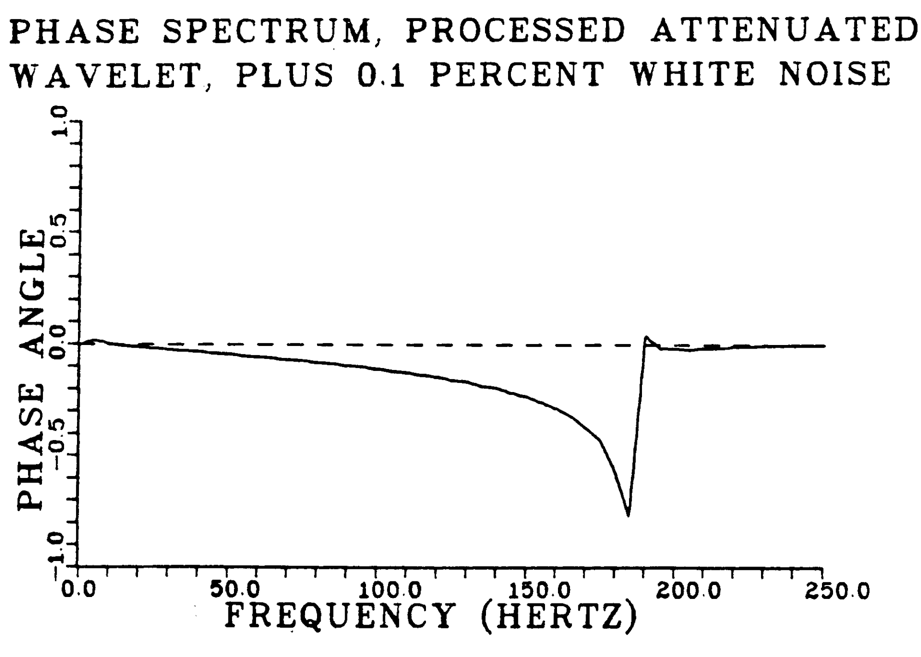

ZERO PHASE WAVELET * DECONVOLVED WAVELET
ATTENUATED, PLUS 0.1 PERCENT WHITE NOISE
AMPLITUDE
1.0
0.5
0.0
-0.5
TIME (MSEC)
-100.0
-50.0
0.0
50.0
100.0

Figure 98


AMPLITUDE SPECTRUM, PROCESSED ATTENUATED WAVELET, PLUS 0.1 PERCENT WHITE NOISE
AMPLITUDE
0.0
0.5
1.0
0.0
50.0
100.0
150.0
200.0
250.0
FREQUENCY (HERTZ)

(a)


PHASE SPECTRUM, PROCESSED ATTENUATED WAVELET, PLUS 0.1 PERCENT WHITE NOISE
PHASE ANGLE
-1.0
-0.5
0.0
0.5
1.0
0.0
50.0
100.0
150.0
200.0
250.0
FREQUENCY (HERTZ)

(b)

Figure 99

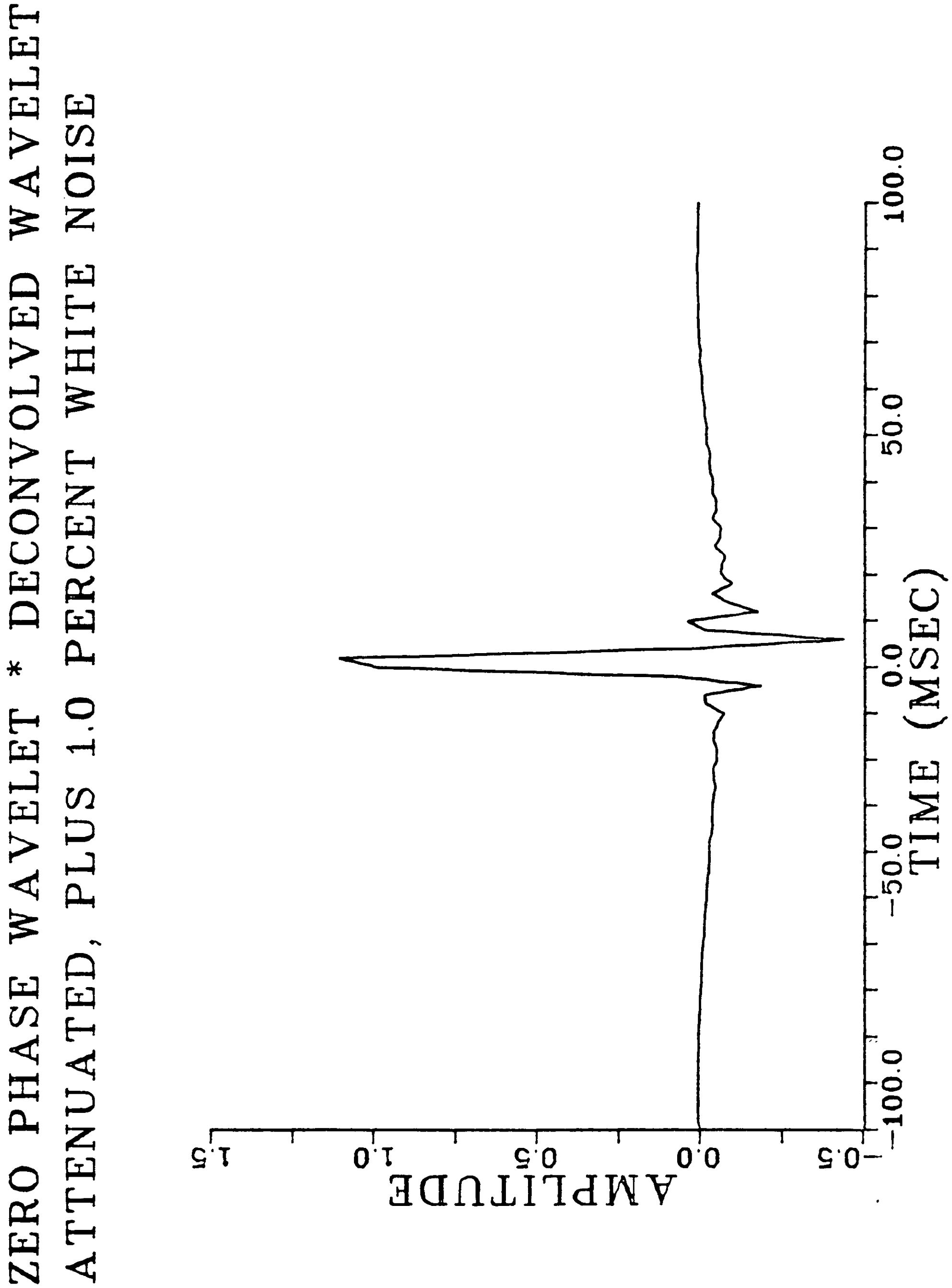
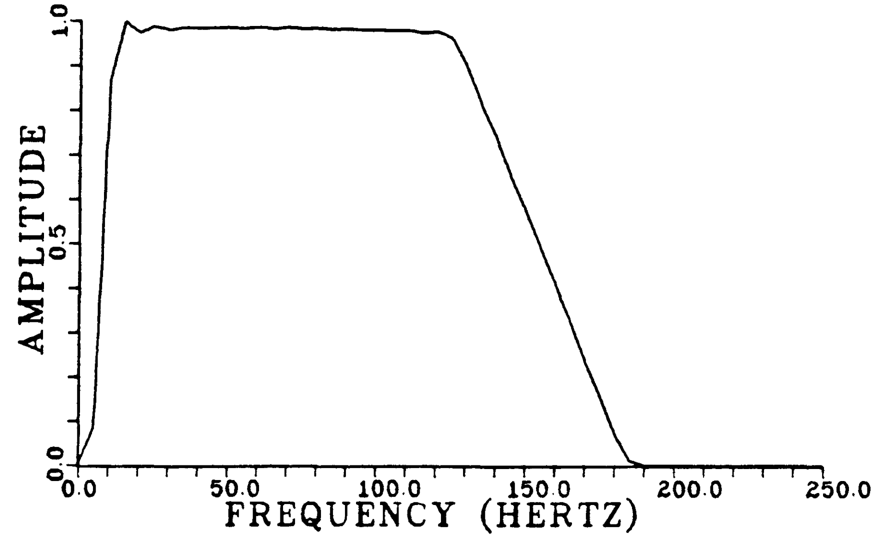

ZERO PHASE WAVELET * DECONVOLVED WAVELET
ATTENUATED, PLUS 1.0 PERCENT WHITE NOISE
AMPLITUDE
-0.5
0.0
0.5
1.0
1.5
TIME (MSEC)
-100.0
-50.0
0.0
50.0
100.0

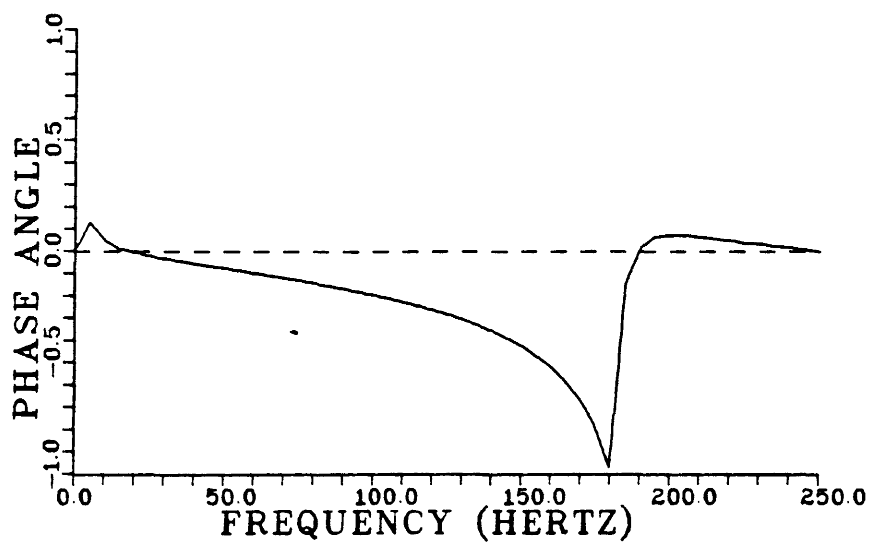
Figure 100

AMPLITUDE SPECTRUM, PROCESSED ATTENUATED WAVELET, PLUS 1.0 PERCENT WHITE NOISE



(a)

PHASE SPECTRUM, PROCESSED ATTENUATED WAVELET, PLUS 1.0 PERCENT WHITE NOISE



(b)

Figure 101

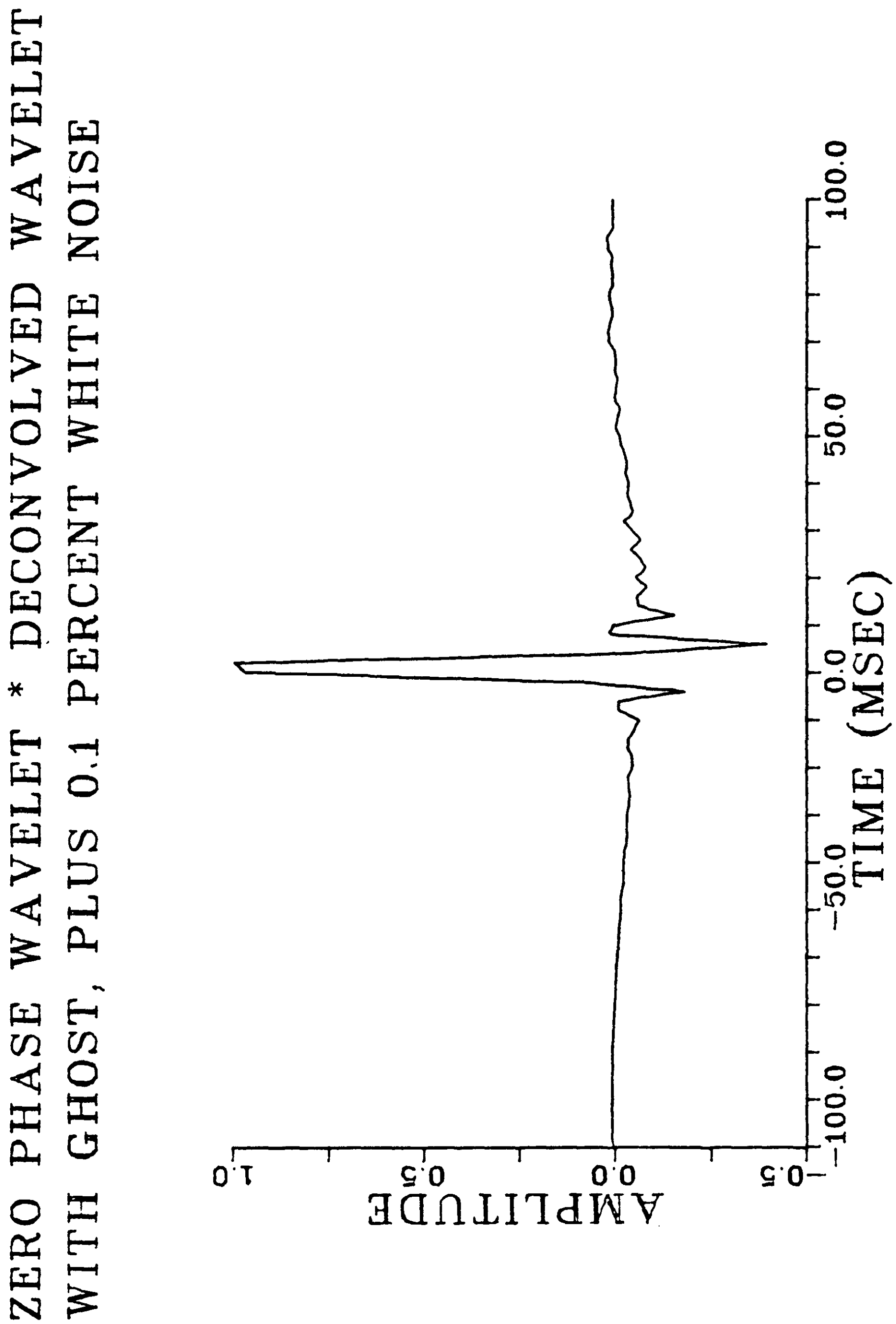
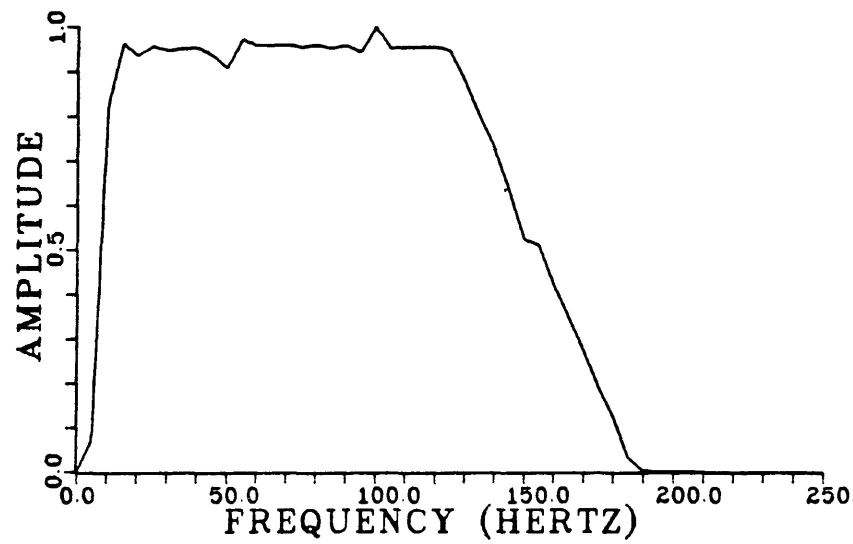

ZERO PHASE WAVELET * DECONVOLVED WAVELET
WITH GHOST, PLUS 0.1 PERCENT WHITE NOISE
AMPLITUDE
1.0
0.5
0.0
-0.5
TIME (MSEC)
-100.0
-50.0
0.0
50.0
100.0

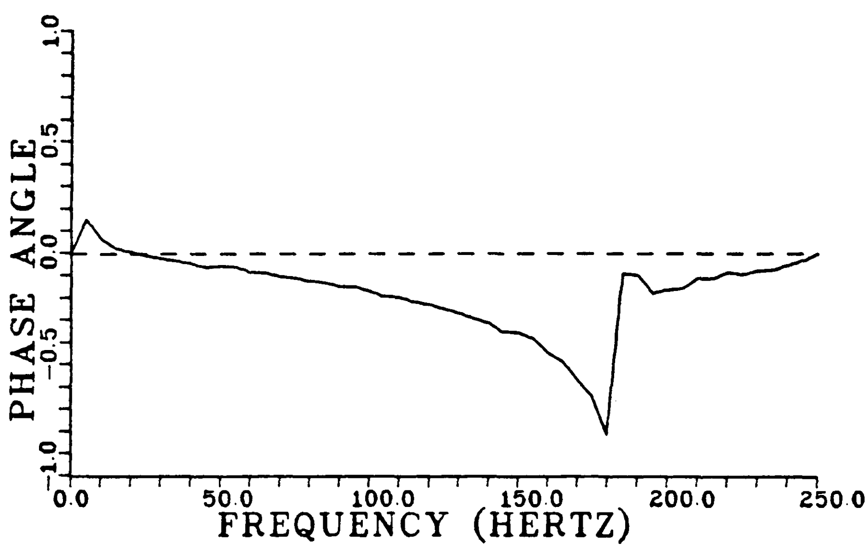
Figure 102

AMPLITUDE SPECTRUM, PROCESSED WAVELET WITH GHOST, PLUS 0.1 PERCENT WHITE NOISE



(a)

PHASE SPECTRUM, PROCESSED WAVELET WITH GHOST, PLUS 0.1 PERCENT WHITE NOISE



(b)

Figure 103

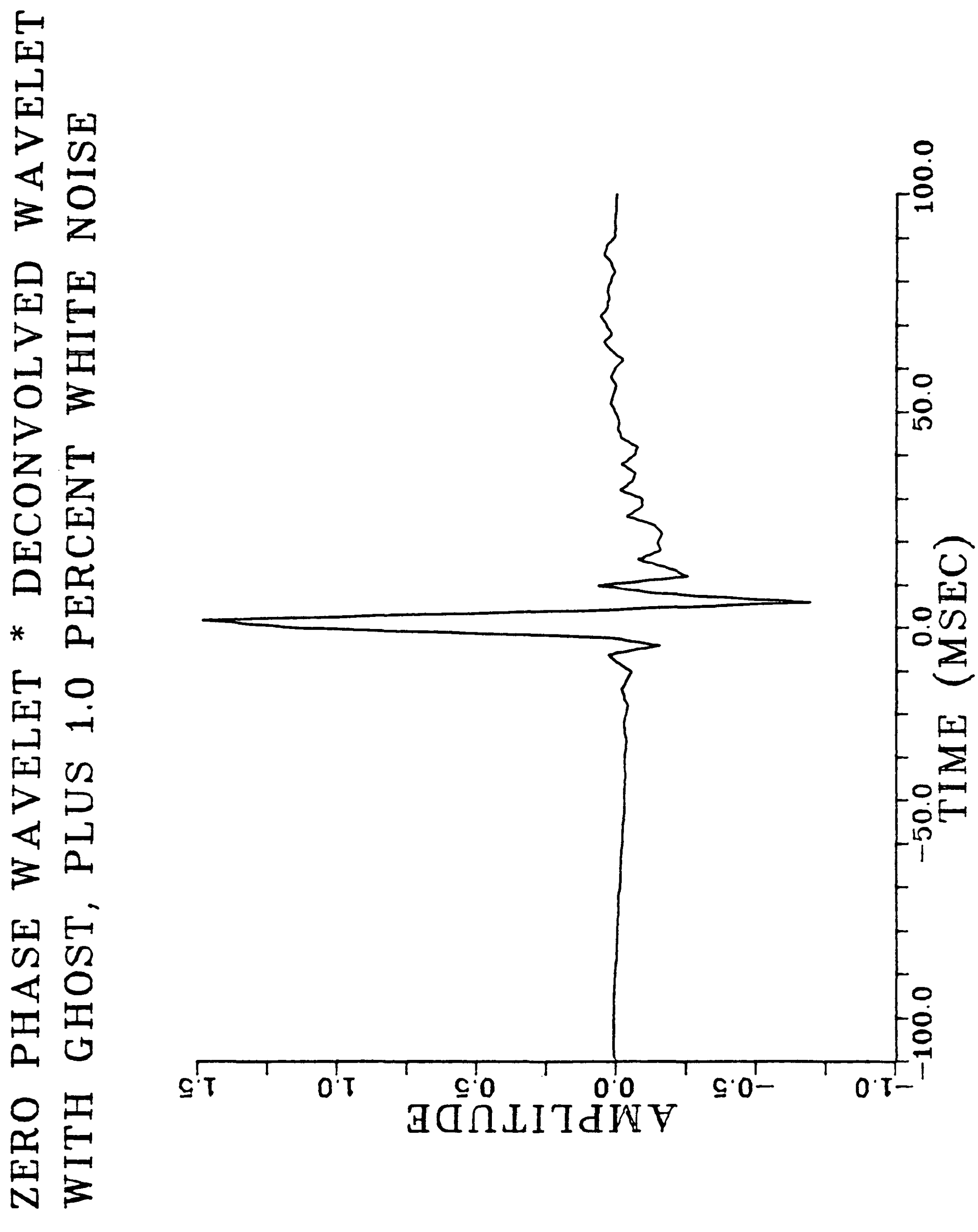
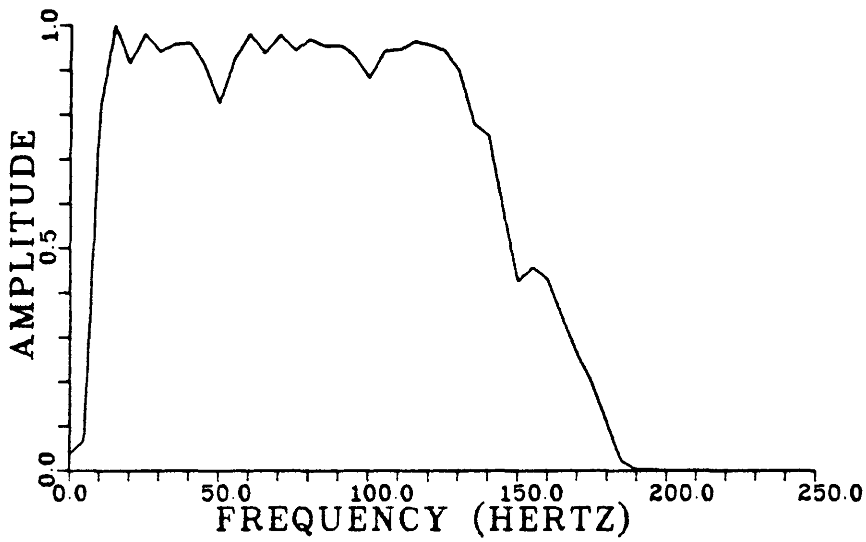

ZERO PHASE WAVELET * DECONVOLVED WAVELET
WITH GHOST, PLUS 1.0 PERCENT WHITE NOISE
AMPLITUDE
1.5
1.0
0.5
0.0
−0.5
−1.0
TIME (MSEC)
−100.0
−50.0
0.0
50.0
100.0

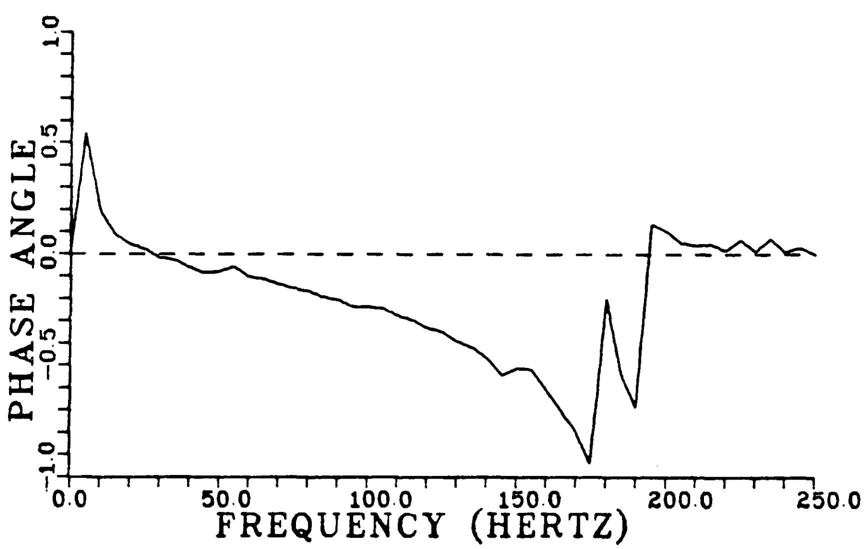
Figure 104

AMPLITUDE SPECTRUM, PROCESSED WAVELET WITH GHOST, PLUS 1.0 PERCENT WHITE NOISE



(a)

PHASE SPECTRUM, PROCESSED WAVELET WITH GHOST, PLUS 1.0 PERCENT WHITE NOISE



(b)

Figure 105

CONCLUSIONS

To be effective, inverse filtering or deconvolution must produce zero phase, broad-band wavelets. Knowledge of amplitude and phase is essential, particularly for data inversion schemes that produce acoustic impedence sections for stratigraphic interpretation.

It is shown here that spiking deconvolution in the presence of bandlimited seismic noise is not particularly suitable for producing zero phase, broad-band results.

Examples indicate one-sided Wiener filtering in the presence of noise will lead to minimum, not zero, phase wavelets that are delayed from zero time lag. The results also predict that for signal and noise of similar spectral characteristics, the phase angle is dependent upon frequency alone. However, for signal and noise of different spectral characteristics, the phase angle is a function of frequency and an inverse function of signal-to-noise ratio. Examination of standard deviation curves implies that phase error variations from trace to trace on a record will lead to final amplitude reduction upon application of a common depth point stack.

Interpretation of these results dictates that to improve the final section with regard to phase error, deconvolution should be applied after stack. Examination of ghost wavelet spectra reveals that an attempt to predict ghost or instrument response notches in the amplitude spectrum of a wavelet is desirable. Prior to deconvolution, an attempt to precondition the spectrum to force the existence of such notches, rather than let them fill up with noise, should be made. This will improve the filtered output.

Deconvolution in the presence of white noise leads to minimum phase results that are dependent upon the percentage of noise level used. Phase error increases with higher white noise percentages. Therefore, application of white noise to facilitate stabilization of the normal equations should be done with great care, and additive percentages of white noise should be kept minimal.

APPENDIX A

Program WAVLET.FOR

Program WAVLET.FOR computes a time domain zero phase wavelet, given an Ormsby filter design in the frequency domain. Four corner frequencies are used, defining a trapezoidal amplitude spectrum. The wavelet may also be attenuated by varying the amplitude of the third corner frequency between zero and one. The program was written to be used interactively, and the inputs are:

K = the sample rate in milliseconds

I = the starting value of the time window in milliseconds

N = the ending value of the time window in milliseconds

F1, F2, F3, and F4 = the four amplitude corner frequencies in hertz

AMP3 = the amplitude of the third corner frequency.

The output $w_{-i}, \ldots w_o, w_1, \ldots w_n$ is written to device number two. That is, the data is stored in a file called FOR02.DAT. Following is a listing of program WAVLET.FOR, including the output for a sample test case.

```
C   ****************************************************************
C
C   GIVEN A TRAPEZOIDAL AMPLITUDE SPECTRUM IN THE FREQUENCY DOMAIN,
C   PROGRAM WAVLET.FOR COMPUTES THE CORRESPONDING TIME-DOMAIN WAVE-
C   LET.  THE DESIRED TIME WINDOW, TIME SAMPLE RATE, AND THE FOUR
C   CORNER FREQUENCIES OF THE TRAPEZOIDAL SPECTRUM ARE INPUT TO THE
C   PROGRAM.  WAVLET.FOR ALSO ALLOWS THE USER TO ATTENUATE THE
C   WAVELET BY VARYING THE AMPLITUDE OF THE THIRD CORNER FREQUENCY.
C   WAVLET.FOR WAS WRITTEN BY NEAL E. FAUSSET IN FEBRUARY, 1979,
C   AND WAS DESIGNED FOR USE ON THE COLORADO SCHOOL OF MINES
C   DEC SYSTEM-10.
C
C   ****************************************************************
C
C
C   **********************************************
C
C   INPUT THE DESIRED SAMPLE RATE IN MILLISECONDS.
C   THIS NUMBER MUST BE AN INTEGER, AS THE
C   PROGRAM DOES NOT ACCOMODATE SAMPLE RATES
C   LESS THAN ONE MILLISECOND.
C
C   **********************************************
C
  800 WRITE (4,1)
    1 FORMAT(' ','WHAT IS THE SAMPLE RATE IN MSEC?')
      WRITE (4,801)
  801 FORMAT(' ','(THIS NUMBER MUST BE AN INTEGER.)')
      READ (4,2) K
    2 FORMAT (G)
      P=FLOAT(K)/1000.
      FNYQ = 1./(2.*P)
C
C   *****************************************************
C
C   INPUT THE BEGINNING VALUE OF THE DESIRED TIME WINDOW,
C   IN MILLISECONDS.
C
C   *****************************************************
C
      WRITE (4,3)
    3 FORMAT(' ','WHAT IS THE START TIME IN MSEC?')
      READ (4,2) I
C
C   *****************************************************************
C
C   INPUT THE END VALUE OF THE DESIRED TIME WINDOW, IN MILLISECONDS.
C
C   *****************************************************************
C
      WRITE (4,4)
```

```
    4 FORMAT(' ','WHAT IS THE END TIME IN MSEC?')
      READ (4,2) N
C
C  ************************************************
C
C  INPUT THE FOUR VALUES, IN HERTZ, THAT DEFINE THE
C  TRAPEZOIDAL SPECTRUM.
C
C  ************************************************
C
      WRITE (4,5)
    5 FORMAT(' ','INPUT THE FOUR CORNER FREQUENCIES IN HZ.')
      READ (4,6) F1,F2,F3,F4
    6 FORMAT (4G)
      F5 = F4-1.
      IF(F5.GT.FNYQ) GO TO 700
C
C  *************************************************
C
C  INPUT THE AMPLITUDE OF THE THIRD CORNER FREQUENCY.
C  (FOR AN UNATTENUATED WAVELET, F3 = 1.0.)
C
C  *************************************************
C
      WRITE (4,500)
  500 FORMAT(' ','WHAT IS THE AMPLITUDE OF F3?')
      WRITE (4,501)
  501 FORMAT(' ','(THIS NUMBER MUST BE REAL.)')
      READ (4,2) AMP3
      PI=3.141592654
C
C  ****************************************************************
C
C  COMPUTE THE ZERO-TIME VALUE OF THE WAVLET.  (EVALUATING THE
C  ALGORITHM AT TIME ZERO RESULTS IN A 0/0 INDETERMINATE FORM,
C  THEREFORE THE ORIGINAL EQUATION MUST BE CONSIDERED IN A LIMITING
C  SENSE, DEFINING A NEW ALGORITHM.)
C
C  ****************************************************************
C
      AA=0.0
      BB=P*(((AMP3*F4)+F3)-((AMP3*F2)+F1))
C
C  ************************************************
C
C  OUTPUT DATA HEADER INFORMATION IN FILE FOR02.DAT.
C
C  ************************************************
C
      WRITE (2,15) K
   15 FORMAT(' ','SAMPLE INTERVAL = ',I2,' MSEC',//)
```

```
      WRITE (2,8) F1,F2
    8 FORMAT(' ','LOWSIDE CUTOFF FREQUENCIES = ',F5.2,',',F5.2,//)
      WRITE (2,9) F3,F4
    9 FORMAT(' ','HIGHSIDE CUTOFF FREQUENCIES = ',F6.2,',',F6.2,//)
      WRITE (2,600) AMP3
  600 FORMAT(' ','THIRD CORNER FREQUENCY AMPLITUDE = ',F4.2,//)
      WRITE (2,10)
   10 FORMAT(' ',17X,'WINDOWED SINC OUTPUT WAVELET',//)
      WRITE (2,50)
   50 FORMAT(' ',17X,'TIME (MSEC)',10X,'WAVELET',/)
C
C   ****************************
C
C   COMPUTE THE WAVELET VALUES.
C
C   ****************************
C
      DO 100 J=I,N,K
      Q=FLOAT(J)/FLOAT(K)
      QQ=FLOAT(J)
C
C   ******************************************************************
C
C   INSURE THAT THE PROGRAM CORRECTLY COMPUTES THE ZERO-TIME VALUE.
C
C   ******************************************************************
C
      IF(ABS(Q).LT.0.00001) GO TO 90
      A=1./((PI**2)*(Q**2)*P)
      B=SIN(PI*(F4+F3)*Q*P)
      C=SIN(PI*(F4-F3)*Q*P)
      D=(B*C*AMP3)/(F4-F3)
      E=SIN(PI*(F2+F1)*Q*P)
      F=SIN(PI*(F2-F1)*Q*P)
      G=(E*F)/(F2-F1)
      X=SIN(PI*(F3+F2)*Q*P)
      Y=SIN(PI*(F3-F2)*Q*P)
      Z=((1.-AMP3)*X*Y)/(F3-F2)
      H=A*(D+Z-G)
C
C   ********************************************
C
C   OUTPUT THE WAVELET VALUES IN FILE FOR02.DAT.
C
C   ********************************************
C
      WRITE (2,20) QQ,H
      GO TO 100
   90 WRITE (2,20) AA,BB
  100 CONTINUE
   20 FORMAT(20X,F6.1,10X,F12.8)
```

```
      GO TO 750
  700 WRITE (4,105)
  105 FORMAT(' ','YOUR BANDPASS EXCEEDS THE NYQUIST FREQUENCY')
      WRITE (4,106)
  106 FORMAT(' ','FOR THE GIVEN SAMPLE RATE.')
      WRITE (4,107)
  107 FORMAT(' ','TRY IT AGAIN, IDIOT!',//)
      GO TO 800
  750 WRITE (4,101)
  101 FORMAT(' ','THE OUTPUT IS IN FILE FOR02.DAT.',//)
      STOP
      END
```

SAMPLE INTERVAL = 1 MSEC

LOWSIDE CUTOFF FREQUENCIES = 5.00,10.00

HIGHSIDE CUTOFF FREQUENCIES = 125.00,250.00

THIRD CORNER FREQUENCY AMPLITUDE = 1.00

WINDOWED SINC OUTPUT WAVELET

TIME (MSEC)	WAVELET
-10.0	-0.01033877
-9.0	-0.01096789
-8.0	-0.01460869
-7.0	-0.00885122
-6.0	-0.00352113
-5.0	-0.02630950
-4.0	-0.06556211
-3.0	-0.04678673
-2.0	0.08634584
-1.0	0.27158575
0.0	0.36000000
1.0	0.27158575
2.0	0.08634584
3.0	-0.04678673
4.0	-0.06556211
5.0	-0.02630950
6.0	-0.00352113
7.0	-0.00885122
8.0	-0.01460869
9.0	-0.01096789
10.0	-0.01033877

APPENDIX B

Program RNOISE.FOR

Using a fortran random number generator, program RNOISE.FOR creates a white noise series in the time domain. For a one millisecond sample rate, a two second long noise realization may be generated. The program was written to be used interactively, and its inputs are:

N = the desired length of the sequence, in milliseconds

K = the sample rate in milliseconds.

The output $n_o, n_1, \ldots n_n$ is written to device number one. That is, the data is stored in a file named FOR01.DAT. Following is a listing of program RNOISE.FOR, including the output for a sample test cast.

```
C     ******************************************************
C
C     PROGRAM RNOISE.FOR GENERATES A RANDOM NUMBER SEQUENCE
C     UP TO TWO SECONDS LONG.  THE PROGRAM WAS WRITTEN BY
C     NEAL E. FAUSSET IN NOVEMBER, 1978, AND WAS DESIGNED
C     FOR USE ON THE COLORADO SCHOOL OF MINES DEC SYSTEM-10.
C     THE PROGRAM INITIALLY GENERATES A VALUE EVERY MILLI-
C     SECOND, HOWEVER IT ALLOWS THE USER TO RESAMPLE THE
C     DATA AT A LESSER RATE, IF DESIRED.  THE DATA IS
C     OUTPUT UNDER FILENAME FOR01.DAT.
C
C     ******************************************************
C
          DIMENSION X(2000), D(1000)
C
C     **************************************
C
C     INPUT THE DESIRED LENGTH OF THE RANDOM
C     NOISE SEQUENCE, IN MILLISECONDS.
C
C     **************************************
C
          WRITE(4,2)
        2 FORMAT(' ','HOW MANY MSEC OF RANDOM NOISE DO YOU WANT?')
          READ(4,3) N
        3 FORMAT (G)
          M=N
          N=N+1
C
C     *********************************************
C
C     INPUT THE DESIRED SAMPLE RATE, IN MILLISECONDS.
C     THIS NUMBER MUST BE AN INTEGER.
C
C     *********************************************
C
          WRITE (4,4)
        4 FORMAT(' ','WHAT MSEC SAMPLE RATE DO YOU WANT?')
          WRITE (4,100)
      100 FORMAT(' ','(THIS NUMBER MUST BE AN INTEGER.)')
          READ (4,3) K
C
C     ******************************************
C
C     START THE FORTRAN RANDOM NUMBER GENERATOR,
C     USING A NEW INITIAL SEED FOR EACH EXECUTION.
C
C     ******************************************
C
          CALL TIME (A,B)
          CALL SETRAN (B)
```

```
      IF (K.NE.1.) GO TO 8
      WRITE (1,6)M
    6 FORMAT(' ',G,1X,'MSEC NOISE TRAIN, 1 MSEC SAMPLE RATE',/)
      WRITE(1,9)
    9 FORMAT(' ',17X,'TIME (MSEC)',9X,'NOISE TRAIN',/)
C
C  *******************************************
C
C  STORE THE RANDOM NUMBERS IN THE ARRAY X(I).
C
C  *******************************************
C
      Z=0.0
    8 DO 10 I=1,N
      X(I)=RAN(1.0)
      IF (K.NE.1) GO TO 10
C
C  *******************************************************
C
C  WRITE THE ONE MILLISECOND SAMPLED DATA IN FILE FOR01.DAT.
C
C  *******************************************************
C
      WRITE (1,11) Z,X(I)
      Z=Z+FLOAT(K)
   11 FORMAT (20X,F6.1,10X,F12.8)
   10 CONTINUE
      IF (K.EQ.1) GO TO 90
      WRITE(1,7)M,K
    7 FORMAT(' ',G,1X,'MSEC NOISE TRAIN,',I2,1X,'MSEC SAMPLE RATE',/)
      WRITE(1,9)
C
C  ***********************************************
C
C  IF DESIRED, RESAMPLE THE RANDOM SEQUENCE.  STORE
C  THE RESAMPLED VALUES IN THE ARRAY D(Y).
C
C  ***********************************************
C
      S=0.0
      DO 80 I=1,N
      Y=((FLOAT(K)-1.)+FLOAT(I))/FLOAT(K)
      C=Y - INT(Y)
      IF (C.NE.0.0) GO TO 80
      D(Y)=X(I)
C
C  *********************************************
C
C  WRITE THE RESAMPLED VALUES IN FILE FOR01.DAT.
C
C  *********************************************
```

```
C
      WRITE (1,11) S, D(Y)
      S=S+FLOAT(K)
   80 CONTINUE
   90 WRITE(4,50)
   50 FORMAT(' ','THE DATA IS IN FILE FOR01.DAT.',//)
      STOP
      END
```

30 MSEC NOISE TRAIN, 2 MSEC SAMPLE RATE

TIME (MSEC)	NOISE TRAIN
0.0	0.74347788
2.0	0.14330357
4.0	0.07806994
6.0	0.15971253
8.0	0.04936790
10.0	0.08415063
12.0	0.54092392
14.0	0.17274396
16.0	0.69744811
18.0	0.94218490
20.0	0.72524881
22.0	0.66202228
24.0	0.67237283
26.0	0.39038263
28.0	0.86047387
30.0	0.51938090

APPENDIX C

Program CONVLV.FOR

Program CONVLV.FOR was written to perform polynomial multiplication, or equivalently, the complete transient convolution of two signals. It calls upon subroutines FOLD and ZERO (Robinson, 1967). The program was written to be used interactively. Vector values are input and output through data files constrained by a 2A4 format. That is, the data is stored in a file whose filename and extension can each contain up to four alphanumeric characters, separated by a period. An example is DATA.DAT. The program inputs are:

LA = m+1 = the length of the first polynomial

X and Y = $t_o, t_1, .. t_{m,n}$ = the polynomial time indices

A = $a_o, a_1, .. a_m$ = the first polynomial values (X and A are input through FILE.EXT, the data filename and extension)

LB = n+1 = the length of the second polynomial

B = $b_o, b_1, .. b_n$ = the second polynomial values (Y and B are input through FILE.EXT).

The program outputs the convolved result $c_o, c_1, .. c_{m+n}$ of length m+n+1. Following is a listing of program CONVLV.FOR, including the output for a sample test case.

```
C   ****************************************************************
C
C   PROGRAM CONVLV.FOR PERFORMS POLYNOMIAL MULTIPLICATION, OR
C   EQUIVALENTLY, THE COMPLETE TRANSIENT CONVOLUTION OF TWO
C   SIGNALS.  THE PROGRAM CALLS UPON TWO SUBROUTINES, FOLD AND
C   ZERO, BOTH OBTAINED FROM THE BOOK "MULTICHANNEL TIME SERIES
C   ANALYSIS WITH DIGITAL COMPUTER PROGRAMS," BY ENDERS A.
C   ROBINSON.  CONVLV.FOR WAS WRITTEN BY NEAL E. FAUSSET IN
C   FEBRUARY, 1979, AND WAS DESIGNED FOR USE ON THE COLORADO
C   SCHOOL OF MINES DEC SYSTEM-10.
C
C   ****************************************************************
C
      DIMENSION X(1C05), A(1005), Y(1005), B(1005), C(1C05)
C
C   ****************************************
C
C   INPUT THE FILENAME OF THE FIRST VECTOR.
C
C   ****************************************
C
      WRITE (4,1)
    1 FORMAT(' ','ENTER THE FIRST SIGNAL FILENAME.')
      ACCEPT 2, INFIL1
    2 FORMAT (2A4)
      OPEN(UNIT=1,FILE=INFIL1)
C
C   **************************************
C
C   INPUT THE LENGTH OF THE FIRST VECTOR.
C
C   **************************************
C
      WRITE (4,3)
    3 FORMAT(' ','WHAT IS THE LENGTH OF THE SIGNAL?')
      READ (4,4) LA
    4 FORMAT (G)
C
C   *****************************************
C
C   INPUT THE FILENAME OF THE SECOND VECTOR.
C
C   *****************************************
C
      WRITE (4,5)
    5 FORMAT(' ','ENTER THE SECOND SIGNAL FILENAME.')
      ACCEPT 2, INFIL2
      OPEN(UNIT=2,FILE=INFIL2)
C
C   ***************************************
C
```

```
C   INPUT THE LENGTH OF THE SECOND VECTOR.
C
C   ***************************************
C
        WRITE (4,6)
      6 FORMAT(' ','WHAT IS THE LENGTH OF THE SIGNAL?')
        READ (4,4) LB
C
C   *****************************
C
C   SPECIFY THE OUTPUT FILENAME.
C
C   *****************************
C
        WRITE (4,7)
      7 FORMAT(' ','ENTER THE OUTPUT FILENAME.')
        ACCEPT 2, OUTFIL
        OPEN(UNIT=3,FILE=OUTFIL)
        WRITE (4,50)
     50 FORMAT(' ','ENTER THE SAMPLE RATE.')
        WRITE (4,51)
     51 FORMAT(' ','(THIS NUMBER MUST BE REAL.)')
        READ (4,4) R
C
C   ***************************************
C
C   COMPUTE THE LENGTH OF THE CONVOLUTION.
C
C   ***************************************
C
        LC=LA+LB-1.
        READ (1,8) (X(I),A(I),I=1,LA)
      8 FORMAT(20X,F6.1,10X,F12.8)
        READ (2,8) (Y(I),B(I),I=1,LB)
C
C   *************************
C
C   PERFORM THE CONVOLUTION.
C
C   *************************
C
        CALL FOLD (LA,A,LB,B,LC,C)
        WRITE (3,9)
      9 FORMAT(' ',23X,'INPUT VECTOR NUMBER ONE',/)
        WRITE (3,8) (X(I),A(I),I=1,LA)
        WRITE (3,10)
     10 FORMAT(' ',//,24X,'INPUT VECTOR NUMBER TWO',/)
        WRITE (3,8) (Y(I),B(I),I=1,LB)
C
C   ******************************
C
```

```
C  OUTPUT THE CONVOLVED RESULTS.
C
C  *****************************
C
      WRITE (3,11)
   11 FORMAT(' ',//,27X,'CONVOLVED OUTPUT',/)
      P = X(1)
      Z = Y(1)
      IF(Z.LT.P) P = Y(1)
      DO 100 K=1,LC
      WRITE (3,8) P, C(K)
      P=P+R
  100 CONTINUE
      STOP
      END
```

```
      SUBROUTINE FOLD(LA,A,LB,B,LC,C)
C
C   ************************************************************
C
C   SUBROUTINE FOLD PERFORMS THE CONVOLUTION.  IT REQUIRES
C   SUBROUTINE ZERO.
C
C   ************************************************************
C
      DIMENSION A(LA),B(LB),C(LC)
      LC=LA+LB-1
      CALL ZERO(LC,C)
      DO 1 I=1,LA
      DO 1 J=1,LB
      K=I+J-1
    1 C(K)=C(K)+A(I)*B(J)
      RETURN
      END
```

```
      SUBROUTINE ZERO(LX,X)
C
C    ************************************************************
C
C    SUBROUTINE ZERO'S PURPOSE IS TO STORE THE FLOATING POINT
C    NUMBER 0.0 IN EACH STORAGE LOCATION OF AN ARRAY.
C
C    ************************************************************
C
      DIMENSION X(LX)
      IF(LX.LE.0) RETURN
      DO 1 I=1,LX
    1 X(I)=0.0
      RETURN
      END
```

INPUT VECTOR NUMBER ONE

-6.0	-0.00352113
-5.0	-0.02630950
-4.0	-0.06556211
-3.0	-0.04678673
-2.0	0.08634584
-1.0	0.27158575
0.0	0.36000000
1.0	0.27158575
2.0	0.08634584
3.0	-0.04678673
4.0	-0.06556211
5.0	-0.02630950
6.0	-0.00352113

INPUT VECTOR NUMBER TWO

0.0	1.00000000
1.0	-0.21110000
2.0	-0.02600000
3.0	0.00000000

CONVOLVED OUTPUT

-6.0	-0.00352113
-5.0	-0.02556619
-4.0	-0.05991663
-3.0	-0.03226252
-2.0	0.09792713
-1.0	0.25457460
0.0	0.30042326
1.0	0.18852852
2.0	0.01965409
3.0	-0.07207557
4.0	-0.05793042
5.0	-0.01125288
6.0	0.00373742
7.0	0.00142736
8.0	0.00009155
9.0	0.00000000

APPENDIX D
Program AUTCOR.FOR

The purpose of program AUTCOR.FOR is to compute the autocorrelation of the signal x_o, x_1, . . . x_m. The program calls upon two subroutines, CROSS and DOT (Robinson, 1967). It was written to be used interactively. Actual vector values are input and output through data files constrained by a 2A4 format. That is, the data is stored in a file whose filename and extension can each contain up to four alphanumeric characters, separated by a period. An example is DATA.DAT. The program inputs are:

LX = m+1 = the number of values in the input vector

X = x_o, x_1, . . x_m = the input vector values

A = t_o, t_1, . . t_m = the input time indices (both X and A are input through FILE.EXT, the data filename and extension)

LR = k+1 = the desired number of output values

SR = the sample rate in milliseconds.

The output r_o, r_1, . . . r_k is the right-hand side autocorrelation function of the input vector. Following is a listing of program AUTCOR.FOR, including the output for a sample test case.

```
C   ***************************************************
C
C   PROGRAM AUTCOR.FOR COMPUTES THE RIGHT-HAND SIDE
C   OF THE AUTOCORRELATION FUNCTION OF A VECTOR.
C   THE PROGRAM WAS WRITTEN BY NEAL E. FAUSSET IN
C   DECEMBER, 1978.  IT CALLS UPON TWO SUBROUTINES,
C   CROSS AND DOT.  BOTH SUBROUTINES WERE OBTAINED FROM
C   THE BOOK "MULTICHANNEL TIME SERIES ANALYSIS WITH
C   DIGITAL COMPUTER PROGRAMS," BY ENDERS A. ROBINSON.
C   THE PROGRAM WAS DESIGNED FOR USE ON THE COLORADO
C   SCHOOL OF MINES DEC SYSTEM-10.
C
C   ***************************************************
C
      DIMENSION A(1000), X(1000), R(1000)
C
C   ****************************************************
C
C   INPUT THE LENGTH OF THE VECTOR TO BE AUTOCORRELATED.
C
C   ****************************************************
C
      WRITE (4,1)
    1 FORMAT(' ','HOW MANY POINTS ARE IN YOUR INPUT VECTOR?')
      READ (4,2) LX
    2 FORMAT (G)
C
C   ************************************************
C
C   INPUT THE DESIRED LENGTH OF THE AUTOCORRELATION.
C
C   ************************************************
C
      WRITE (4,3)
    3 FORMAT(' ','HOW MANY AUTOCORRELATION OUTPUTS DO YOU WANT?')
      READ (4,2) LR
C
C   ********************************************
C
C   INPUT THE FILENAME IN WHICH THE VECTOR TO BE
C   AUTOCORRELATED IS STORED.  THE FILENAME MUST
C   BE FOUR CHARACTERS LONG.
C
C   ********************************************
C
      WRITE (4,4)
    4 FORMAT(' ','ENTER THE INPUT VECTOR FILENAME.')
      ACCEPT 5, INFIL
    5 FORMAT(2A4)
      OPEN(UNIT=1,FILE=INFIL)
C
```

```
C  ****************************************
C
C  SPECIFY THE FILENAME IN WHICH THE OUTPUT
C  IS TO BE STORED.  THIS FILENAME MUST
C  ALSO BE FOUR CHARACTERS LONG.
C
C  ****************************************
C
      WRITE (4,6)
    6 FORMAT(' ','ENTER THE AUTOCORRELATION OUTPUT FILENAME.')
      ACCEPT 5, OUTFIL
      OPEN(UNIT=2,FILE=OUTFIL)
C
C  **********************
C
C  INPUT THE SAMPLE RATE.
C
C  **********************
C
      WRITE (4,100)
  100 FORMAT(' ','ENTER THE SAMPLE RATE.')
      WRITE (4,101)
  101 FORMAT(' ','(THIS NUMBER MUST BE REAL.)')
      READ (4,2) SR
C
C  ****************************************
C
C  READ IN THE VALUES TO BE AUTOCORRELATED.
C
C  ****************************************
C
      DO 7 I=1,LX
      READ (1,8) A(I), X(I)
      J=I
    7 CONTINUE
      WRITE (2,11)
   11 FORMAT(' ',17X,'INPUT VECTOR',/)
C
C  ******************************
C
C  WRITE THE INPUT VECTOR VALUES.
C
C  ******************************
C
      DO 12 I=1,J
      WRITE (2,8) A(I), X(I)
   12 CONTINUE
C
C  ****************************
C
C  PERFORM THE AUTOCORRELATION.
```

```
C     '
C   *****************************
C
      CALL CROSS(LX,X,LX,X,LR,R)
      WRITE (2,13)
   13 FORMAT(' ',/,17X,'AUTOCORRELATION FUNCTION',/)
C
C   ******************************************
C
C   WRITE THE AUTOCORRELATION OUTPUT VALUES.
C
C   ******************************************
C
      PP = 0.000
      DO 14 K=1,LR
      WRITE (2,8) PP, R(K)
      PP = PP + SR
   14 CONTINUE
    8 FORMAT (20X,F6.1,10X,F12.8)
      STOP
      END
```

```
      SUBROUTINE CROSS(LX,X,LY,Y,LG,G)
C
C   ***********************************************************
C
C   SUBROUTINE CROSS COMPUTES THE AUTOCORRELATION OF A VECTOR
C   OR THE CROSSCORRELATION OF TWO VECTORS.  SUBROUTINE CROSS
C   CALLS UPON SUBROUTINE DOT.
C
C   ***********************************************************
C
      DIMENSION X(LX), Y(LY), G(LG)
      DO 1  J=1,LG
    1 CALL DOT(MINO(LY,LX-J+1),X(J),Y,G(J))
      RETURN
      END
```

```
      SUBROUTINE DOT(L,X,Y,P)
C
C   **********************************************************
C
C   SUBROUTINE DOT COMPUTES THE DOT PRODUCT OF TWO VECTORS.
C
C   **********************************************************
C
      DIMENSION X(L), Y(L)
      P=0.0
      IF(L.LE.0) RETURN
      DO 1 I=1,L
    1 P=P+X(I)*Y(I)
      RETURN
      END
```

INPUT VECTOR

-10.0	-0.01033877
-9.0	-0.01096789
-8.0	-0.01460869
-7.0	-0.00885122
-6.0	-0.00352113
-5.0	-0.02630950
-4.0	-0.06556211
-3.0	-0.04678673
-2.0	0.08634584
-1.0	0.27158575
0.0	0.36000000
1.0	0.27158575
2.0	0.08634584
3.0	-0.04678673
4.0	-0.06556211
5.0	-0.02630950
6.0	-0.00352113
7.0	-0.00885122
8.0	-0.01460869
9.0	-0.01096789
10.0	-0.01033877

AUTOCORRELATION FUNCTION

0.0	0.30745069
1.0	0.24500081
2.0	0.10318100
3.0	-0.02442170
4.0	-0.07666725
5.0	-0.06272594
6.0	-0.03090738
7.0	-0.01555567
8.0	-0.01691666
9.0	-0.01919687

APPENDIX E
Program LSQINF.FOR

Program LSQINF.FOR computes the solution to the normal equations which arise in least squares filtering and prediction problems. In time series analysis, the autocorrelation coefficients r_t of a function are input as a matrix set of simultaneous equations where the matrix

$$\begin{bmatrix} r_o & r_1 & \cdots & r_m \\ r_{-1} & r_o & \cdots & r_{m-1} \\ & \cdots & & \\ r_{-m} & r_{-m+1} & \cdots & r_o \end{bmatrix}$$

is symmetric, positive definite, and Töplitz. The coefficients r_t for $t = 0,1,2,\ldots m$ are known, as are the right-hand side coefficients g_t for $t = 0,1,2,\ldots m$. The unknown filter coefficients f_s for $s = 0,1,2,\ldots m$ are solved for by a recursive algorithm described by Levinson (1947). An additional output is the prediction error operator a_s for $s = 0,1,2,\ldots m$. Ideally, the prediction error operator is equal to one minus the filter coefficients, or $a_s = 1,-f_1, \ldots -f_m$.

The program calls upon subroutine EUREKA (Robinson, 1967). The program was written to be used interactively. Vector values are input and output through data files constrained by a 2A4 format. That is, the data is stored in a file whose filename and extension can each contain up to four alphanumeric characters, separated by a period. An example is DATA.DAT. The program inputs are:

LR = m+1 = the length of the filter

B and Z = $t_o, t_1, .. t_m$ = the input time indices

R = $r_o, r_1, .. r_m$ = the autocorrelation coefficients (B and R are input through FILE.EXT, the data filename and extension)

G = $g_o, g_1, .. g_m$ = the right-hand side coefficients (Z and G are input through FILE.EXT).

The program outputs the filter coefficients $f_o, f_1, ... f_m$ and the prediction error operator $1, a_1, ... a_m$. Following is a listing of program LSQINF.FOR, including the output for a sample test case.

```
C   ************************************************************
C
C   PROGRAM LSQINF.FOR COMPUTES THE SOLUTION TO SINGLE CHANNEL
C   NORMAL EQUATIONS WHICH ARISE IN LEAST-SQUARES FILTERING AND
C   PREDICTION PROBLEMS.  THE PROGRAM WAS WRITTEN BY NEAL E.
C   FAUSSET IN DECEMBER, 1978.  IT CALLS UPON SUBROUTINE EUREKA.
C   SUBROUTINE EUREKA WAS OBTAINED FROM THE BOOK "MULTICHANNEL
C   TIME SERIES ANALYSIS WITH DIGITAL COMPUTER PROGRAMS," BY
C   ENDERS A. ROBINSON.  THE PROGRAM WAS DESIGNED FOR USE ON
C   THE COLORADO SCHOOL OF MINES DEC SYSTEM-10.
C
C   ************************************************************
C
      DIMENSION B(1050),R(1050),Z(1050),G(1050),F(1050),A(1050)
C
C   ************************************************
C
C   INPUT THE DESIRED NUMBER OF FILTER COEFFICIENTS.
C
C   ************************************************
C
      WRITE (4,1)
    1 FORMAT(' ','HOW MANY FILTER COEFFICIENTS DO YOU WANT?')
      READ (4,2) LR
    2 FORMAT(G)
C
C   *****************************************************************
C
C   INPUT THE FILENAME IN WHICH THE AUTOCORRELATED VECTOR IS STORED.
C
C   *****************************************************************
C
      WRITE (4,3)
    3 FORMAT(' ','ENTER THE AUTOCORRELATION COEFFICIENT FILENAME.')
      ACCEPT 4, INFIL1
    4 FORMAT(2A4)
      OPEN(UNIT=1,FILE=INFIL1)
C
C   ***********************************************
C
C   INPUT THE FILENAME IN WHICH THE RIGHT-HAND SIDE
C   COEFFICIENTS ARE STORED.
C
C   ***********************************************
C
      WRITE (4,5)
    5 FORMAT(' ','ENTER THE RIGHT-HAND SIDE COEFFICIENT FILENAME.')
      ACCEPT 4, INFIL2
      OPEN(UNIT=2,FILE=INFIL2)
C
C   ********************************************************
```

```
C
C   SPECIFY THE FILENAME IN WHICH THE OUTPUT IS TO BE STORED.
C
C   **********************************************************
C
      WRITE (4,6)
    6 FORMAT(' ','ENTER THE OUTPUT FILENAME.')
      ACCEPT 4, OUTFIL
      OPEN(UNIT=3,FILE=OUTFIL)
C
C   ******************************************************
C
C   READ THE AUTOCORRELATED VALUES INTO THE LEFT-HAND SIDE
C   TOPLITZ MATRIX.
C
C   ******************************************************
C
      DO 7 I=1,LR
      READ(1,8) B(I), R(I)
    7 CONTINUE
C
C   ****************************************
C
C   READ IN THE RIGHT-HAND SIDE COEFFICIENTS.
C
C   ****************************************
C
      DO 9 J=1,LR
      READ(2,8) Z(J), G(J)
    9 CONTINUE
C
C   **************************************
C
C   WRITE THE VALUES OF BOTH INPUT VECTORS.
C
C   **************************************
C
      WRITE (3,10)
   10 FORMAT(' ',20X,'AUTOCORRELATION COEFFICIENTS',/)
      DO 11 K=1,LR
      WRITE (3,8) B(K), R(K)
   11 CONTINUE
      WRITE (3,12)
   12 FORMAT(' ',//,21X,'RIGHT-HAND SIDE COEFFICIENTS',/)
      DO 13 L=1,LR
      WRITE (3,8) Z(L), G(L)
   13 CONTINUE
C
C   **************************************
C
C   COMPUTE THE FILTER COEFFICIENTS AND THE
```

```
C   PREDICTION ERROR OPERATOR.
C
C   ****************************************
C
       CALL EUREKA(LR,R,G,F,A)
C
C   *******************************
C
C   WRITE THE FILTER COEFFICIENTS.
C   F = (f1, f2, f3, .... fN).
C
C   *******************************
C
       WRITE (3,14)
    14 FORMAT(' ',//,26X,'FILTER COEFFICIENTS',/)
       DO 15 M=1,LR
       WRITE (3,8) B(M), F(M)
    15 CONTINUE
C
C   *************************************
C
C   WRITE THE PREDICTION ERROR OPERATOR.
C   A = (1, -f1, -f2, .... -fN-1), FOR
C   AN AUTOCORRELATED INPUT.
C
C   *************************************
C
       WRITE (3,16)
    16 FORMAT(' ',//,23X,'PREDICTION ERROR OPERATOR',/)
       DO 17 N=1,LR
       WRITE (3,8) B(N), A(N)
    17 CONTINUE
     8 FORMAT (20X,F6.1,10X,F12.8)
       STOP
       END
```

```
      SUBROUTINE EUREKA(LR,R,G,F,A)
C
C   ***********************************************************
C
C   SUBROUTINE EUREKA SOLVES SINGLE CHANNEL NORMAL EQUATIONS
C   BY THE LEVINSON RECURSION METHOD.  IT REQUIRES NO OTHER
C   SUBROUTINES.
C
C   ***********************************************************
C
      DIMENSION R(LR),G(LR),F(LR),A(LR)
      V=R(1)
      D=R(2)
      A(1)=1.
      F(1)=G(1)/V
      Q=F(1)*R(2)
      IF(LR.EQ.1) RETURN
      DO 4 L=2,LR
      A(L)=-D/V
      IF(L.EQ.2) GO TO 2
      L1=(L-2)/2
      L2=L1+1
      IF(L2.LT.2) GO TO 5
      DO 1 J=2,L2
      HOLD=A(J)
      K=L-J+1
      A(J)=A(J)+A(L)*A(K)
    1 A(K)=A(K)+A(L)*HOLD
    5 IF(2*L1.EQ.L-2) GO TO 2
      A(L2+1)=A(L2+1)+A(L)*A(L2+1)
    2 V=V+A(L)*D
      F(L)=(G(L)-Q)/V
      L3=L-1
      DO 3 J=1,L3
      K=L-J+1
    3 F(J)=F(J)+F(L)*A(K)
      IF(L.EQ.LR) RETURN
      D=0.0
      Q=0.0
      DO 4 I=1,L
      K=L-I+2
      D=D+A(I)*R(K)
    4 Q=Q+F(I)*R(K)
      END
```

AUTOCORRELATION COEFFICIENTS

0.0	0.38431336
1.0	0.30131213
2.0	0.16233441
3.0	0.02601536
4.0	-0.02829616
5.0	-0.02092389
6.0	0.00568244
7.0	0.02457978

RIGHT-HAND SIDE COEFFICIENTS

1.0	0.30131213
2.0	0.16233441
3.0	0.02601536
4.0	-0.02829616
5.0	-0.02092389
6.0	0.00568244
7.0	0.02457978
8.0	0.01672474

FILTER COEFFICIENTS

0.0	1.19066090
1.0	-0.21860014
2.0	-0.64510811
3.0	0.45384436
4.0	0.13832322
5.0	-0.32161464
6.0	0.19511567
7.0	-0.05759754

PREDICTION ERROR OPERATOR

0.0	1.00000000
1.0	-1.18334850
2.0	0.20074187
3.0	0.65524895
4.0	-0.42912764
5.0	-0.17606398
6.0	0.31005242
7.0	-0.12695771

APPENDIX F

Program DIVIDE.FOR

The function of program DIVIDE.FOR is to perform deconvolution, or alternately, to divide one polynomial by another. The program uses three subroutines, POLYDV, ZERO, and MOVE (Robinson, 1967). It was written to be used interactively. Vector values are input and output through data files constrained by a 2A4 format. That is, the data is stored in a file whose filename and extension can each contain up to four alphanumeric characters, separated by a period. An example is DATA.DAT. The program inputs are:

N = n+1 = the number of samples in the divisor

X and Y = $t_o, t_1, \ldots t_{n,m}$ = the time indices of the respective polynomials

DVS = $b_o, b_1, \ldots b_n$ = the divisor values (X and DVS are input through FILE.DAT, the data filename and extension)

M = m+1 = the number of samples in the dividend

DVD = $a_o, a_1, \ldots a_m$ = the dividend values (Y and DVD are input through FILE.DAT)

L = k+1 = the desired number of output values.

The quotient $q_o, q_1, \ldots q_k$ is output. Following is a listing of program DIVIDE.FOR, including the output for a sample test case.

```
C
C   *************************************************
C
C   PROGRAM DIVIDE.FOR DECONVOLVES ONE SIGNAL BY
C   ANOTHER, THAT IS, IT PERFORMS POLYNOMIAL
C   DIVISION.  THE PROGRAM WAS WRITTEN BY NEAL E.
C   FAUSSET IN MARCH, 1979.  IT CALLS UPON THREE
C   SUBROUTINES; POLYDV, ZERO, AND MOVE.  ALL THE
C   SUBROUTINES WERE OBTAINED FROM THE BOOK "MULTI-
C   CHANNEL TIME SERIES ANALYSIS WITH DIGITAL COM-
C   PUTER PROGRAMS," BY ENDERS A. ROBINSON.  THE
C   PROGRAM WAS DESIGNED FOR USE ON THE COLORADO
C   SCHOOL OF MINES DEC SYSTEM-10.
C
C   *************************************************
C
      DIMENSION X(1050),DVS(1050),Y(1050),DVD(1050),Q(1050)
C
C   ***********************************************
C
C   INPUT THE FILENAME OF THE DIVISOR POLYNOMIAL.
C
C   ***********************************************
C
      WRITE (4,1)
    1 FORMAT(' ','ENTER THE DIVISOR POLYNOMIAL FILENAME.')
      ACCEPT 2, INFIL1
    2 FORMAT (2A4)
      OPEN(UNIT=1,FILE=INFIL1)
C
C   ************************************************
C
C   INPUT THE FILENAME OF THE DIVIDEND POLYNOMIAL.
C
C   ************************************************
C
      WRITE (4,3)
    3 FORMAT(' ','ENTER THE DIVIDEND POLYNOMIAL FILENAME.')
      ACCEPT 2, INFIL2
      OPEN(UNIT=2,FILE=INFIL2)
C
C   *********************************************
C
C   INPUT THE DESIRED FILENAME OF THE QUOTIENT.
C
C   *********************************************
C
      WRITE (4,4)
    4 FORMAT(' ','ENTER THE OUTPUT FILENAME.')
      ACCEPT 2, OUTFIL
      OPEN(UNIT=3,FILE=OUTFIL)
```

```
C
C   *******************************************
C
C   INPUT THE NUMBER OF SAMPLES IN THE DIVISOR.
C
C   *******************************************
C
      WRITE (4,5)
    5 FORMAT(' ','ENTER THE LENGTH OF THE DIVISOR.')
      READ (4,6) N
    6 FORMAT (G)
C
C   ********************************************
C
C   INPUT THE NUMBER OF SAMPLES IN THE DIVIDEND.
C
C   ********************************************
C
      WRITE (4,7)
    7 FORMAT(' ','ENTER THE LENGTH OF THE DIVIDEND.')
      READ (4,6) M
C
C   ****************************************************
C
C   INPUT THE DESIRED NUMBER OF SAMPLES IN THE QUOTIENT.
C
C   ****************************************************
C
      WRITE (4,8)
    8 FORMAT(' ','ENTER THE LENGTH OF THE QUOTIENT.')
      READ (4,6) L
C
C   ***********************************************
C
C   READ THE COEFFICIENTS OF THE TWO INPUT SIGNALS.
C
C   ***********************************************
C
      DO 9 I=1,N
      READ (1,10) X(I),DVS(I)
    9 CONTINUE
   10 FORMAT(20X,F6.1,10X,F12.8)
      DO 11 J=1,M
      READ (2,10) Y(J),DVD(J)
   11 CONTINUE
C
C   ******************************************
C
C   WRITE THE COEFFICIENTS OF BOTH THE DIVISOR
C   AND THE DIVIDEND.
C
```

```
C   ********************************************
C
       WRITE (3,12)
    12 FORMAT(' ',24X,'DIVISOR COEFFICIENTS',/)
       DO 13 K=1,N
       WRITE (3,10) X(K),DVS(K)
    13 CONTINUE
       WRITE (3,14)
    14 FORMAT(' ',//,25X,'DIVIDEND COEFFICIENTS',/)
       DO 16 I=1,M
       WRITE (3,10) Y(I),DVD(I)
    16 CONTINUE
C
C   ***************************
C
C   PERFORM THE DECONVOLUTION.
C
C   ***************************
C
       CALL POLYDV(N,DVS,M,DVD,L,Q)
C
C   *******************************
C
C   WRITE THE DECONVOLVED RESULTS.
C
C   *******************************
C
       WRITE (3,17)
    17 FORMAT(' ',//,31X,'QUOTIENT',/)
       P = 0.00
       SR = X(2) - X(1)
       DO 18 J=1,L
       WRITE (3,10)P,C(J)
       P = P + SR
    18 CONTINUE
       STOP
       END
```

```
      SUBROUTINE POLYDV(N,DVS,M,DVD,L,Q)
C
C    ********************************************
C
C    SUBROUTINE POLYDV DIVIDES ONE POLYNOMIAL BY
C    ANOTHER, THAT IS, IT DECONVOLVES ONE SIGNAL
C    BY ANOTHER.  IT REQUIRES SUBROUTINES ZERO
C    AND MOVE
C
C    ********************************************
C
      DIMENSION DVS(N),DVD(M),Q(L)
      CALL ZERO (L,Q)
      CALL MOVE (MIN0(M,L),DVD,Q)
      DO 10 I=1,L
      Q(I)=Q(I)/DVS(1)
      IF (I.EQ.L) RETURN
      K=I
      ISUB=MIN0(N-1,L-1)
      DO 10 J=1,ISUB
      K=K+1
   10 Q(K)=Q(K)-Q(I)*DVS(J+1)
      RETURN
      END
```

```
      SUBROUTINE ZERO(LX,X)
C
C   *****************************************
C
C   SUBROUTINE ZERO'S PURPOSE IS TO STORE THE
C   FLOATING POINT NUMBER 0.0 IN EACH STORAGE
C   LOCATION OF AN ARRAY.
C
C   *****************************************
C
      DIMENSION X(LX)
      IF (LX.LE.0) RETURN
      DO 1 I=1,LX
    1 X(I)=0.0
      RETURN
      END
```

```
      SUBROUTINE MOVE(LX,X,Y)
C
C    ***********************************************
C
C    SUBROUTINE MOVE'S PURPOSE IS TO MOVE AN ARRAY
C    FROM ONE STORAGE LOCATION TO ANOTHER.
C
C    ***********************************************
C
      DIMENSION X(LX),Y(LX)
      DO 1 I=1,LX
    1 Y(I)=X(I)
      RETURN
      END
```

DIVISOR COEFFICIENTS

0.0	1.00000000
1.0	-1.18334850
2.0	0.20074187
3.0	0.65524895
4.0	-0.42912764
5.0	-0.17606398
6.0	0.31005242
7.0	-0.12695771

DIVIDEND COEFFICIENTS

0.0	5.46000000
1.0	3.27000000
2.0	2.74000000
3.0	1.08000000

QUOTIENT

0.0	5.46000000
1.0	9.73108280
2.0	13.15921200
3.0	11.12083800
4.0	6.48497780
5.0	1.95619730
6.0	-0.60647279
7.0	-0.59446859
8.0	0.03273769
9.0	0.75931535

APPENDIX G

Program SCALE.FOR

Program SCALE.FOR scales noise autocorrelation coefficients to wavelet autocorrelation coefficients according to a desired signal to noise ratio. It then sums the two vectors for input to a Wiener-Levinson filter. The program was written to be used interactively. Vector values are input and output through data files constrained by a 2A4 format. That is, the data is stored in a file whose filename and extension can each contain up to four alphanumeric characters, separated by a period. An example is DATA.DAT. The program inputs are:

C and D = $t_o, t_1, \ldots t_m$ = the time indices of the vectors

X = $x_o, x_1, \ldots x_m$ = the values of the wavelet vector (C and X are input through FILE.DAT, the filename and extension)

Y = $y_o, y_1, \ldots y_m$ = the values of the noise vector (D and Y are input through FILE.DAT)

LV = m+1 = the length of each autocorrelated vector

R = the desired signal to noise ratio.

The scaled and summed autocorrelation coefficients z_o, $z_1, \ldots z_m$ are output. Following is a listing of program SCALE.FOR, including the output for a sample test case.

```
C
C   *********************************************************
C
C   PROGRAM SCALE.FOR COMPUTES PROPERLY SCALED AND SUMMED
C   AUTOCORRELATION COEFFICIENTS FOR INPUT INTO THE TOPLITZ
C   MATRIX OF A LEVINSON RECURSION COMPUTER PROGRAM.  THE
C   AUTOCORRELATED WAVELET VECTOR, AUTOCORRELATED NOISE
C   VECTOR, AND DESIRED SIGNAL TO NOISE RATIO ARE INPUT TO
C   SCALE.FOR.  THE PROGRAM WAS WRITTEN BY NEAL E. FAUSSET
C   IN JANUARY, 1979, AND WAS DESIGNED FOR USE ON THE COLO-
C   RADO SCHOOL OF MINES DEC SYSTEM-10.
C
C   *********************************************************
C
      DIMENSION C(1C50), X(1050), D(1050), Y(1050), Z(1050)
C
C   *******************************************
C
C   INPUT THE AUTOCORRELATED WAVELET FILENAME.
C
C   *******************************************
C
      WRITE (4,1)
    1 FORMAT(' ','ENTER THE WAVELET AUTOCORRELATION FILENAME.')
      ACCEPT 2, INFIL1
    2 FORMAT (2A4)
      OPEN(UNIT=1,FILE=INFIL1)
C
C   ****************************************************
C
C   INPUT THE AUTOCORRELATED NOISE REALIZATION FILENAME.
C   THE FILENAME MUST BE FOUR CHARACTERS LONG.
C
C   ****************************************************
C
      WRITE (4,3)
    3 FORMAT(' ','ENTER THE NOISE AUTOCORRELATION FILENAME.')
      ACCEPT 2, INFIL2
      OPEN(UNIT=2,FILE=INFIL2)
C
C   *************************************
C
C   SPECIFY THE DESIRED OUTPUT FILENAME.
C   THIS FILENAME MUST ALSO BE FOUR
C   CHARACTERS LONG.
C
C   *************************************
C
      WRITE (4,4)
    4 FORMAT(' ','ENTER THE OUTPUT FILENAME.')
      ACCEPT 2, OUTFIL
```

```
      OPEN(UNIT=3,FILE=OUTFIL)
C
C    *************************************************
C
C    INPUT THE NUMBER OF AUTOCORRELATION COEFFICIENTS.
C
C    *************************************************
C
      WRITE (4,5)
    5 FORMAT(' ','ENTER THE LENGTH OF THE AUTOCORRELATIONS.')
      READ (4,6) LV
    6 FORMAT (G)
C
C    *****************************************
C
C    READ IN THE AUTOCORRELATION COEFFICIENTS.
C
C    *****************************************
C
      DO 7 I=1,LV
      READ (1,8) C(I), X(I)
      READ (2,8) D(I), Y(I)
    7 CONTINUE
C
C    ****************************************
C
C    INPUT THE DESIRED SIGNAL TO NOISE RATIO.
C    (THIS NUMBER MUST BE REAL.)
C
C    ****************************************
C
      WRITE (4,9)
    9 FORMAT(' ','ENTER THE DESIRED S/N RATIO.')
      WRITE (4,100)
  100 FORMAT(' ','(THIS NUMBER MUST BE REAL.)')
      READ (4,6) R
C
C    ************************************************************
C
C    IDENTIFY THE ZERO-LAG VALUES OF BOTH AUTOCORRELATION VECTORS.
C
C    ************************************************************
C
      A=X(1)
      B=Y(1)
      WRITE (3,10)
   10 FORMAT(' ',17X,'WAVELET AUTOCORRELATION COEFFICIENTS',/)
      DO 11 J=1,LV
      WRITE (3,8) C(J), X(J)
   11 CONTINUE
      WRITE (3,12)
```

```
   12 FORMAT(' ',//,18X,'NOISE AUTOCORRELATION COEFFICIENTS',/)
      DO 13 K=1,LV
      WRITE (3,8) D(K), Y(K)
   13 CONTINUE
      WRITE (3,14) R
   14 FORMAT(' ',//,18X,'SIGNAL TO NOISE RATIO =',1X,F4.2,/)
C
C  *************************************************************
C
C  COMPUTE THE SCALING FACTOR S.  S EQUALS THE ZERO-LAG VALUE
C  OF THE WAVELET AUTOCORRELATION DIVIDED BY THE PRODUCT OF
C  THE ZERO-LAG VALUE OF THE NOISE AUTOCORRELATION AND THE SQUARE
C  OF THE SIGNAL TO NOISE RATIO.
C
C  *************************************************************
C
      S=A/(B*R*R)
      WRITE (3,15)
   15 FORMAT(' ',//,18X,'SCALED AND SUMMED AUTOCORRELATIONS',/)
C
C  **************************************************************
C
C  MULTIPLY THE NOISE AUTOCORRELATION BY S.  ADD THE SCALED VECTOR
C  TO THE WAVELET AUTOCORRELATION AND OUTPUT THE NEW SCALED AND
C  SUMMED AUTOCORRELATION VECTOR.
C
C  **************************************************************
C
      DO 16 L=1,LV
      Y(L)=S*Y(L)
      Z(L)=X(L)+Y(L)
      WRITE (3,8) C(L), Z(L)
   16 CONTINUE
    8 FORMAT (20X,F6.1,10X,F12.8)
      STOP
      END
```

WAVELET AUTOCORRELATION COEFFICIENTS

0.0	0.30745069
1.0	0.24500081
2.0	0.10318100
3.0	-0.02442170
4.0	-0.07666725
5.0	-0.06272594
6.0	-0.03090738
7.0	-0.01555567
8.0	-0.01691666
9.0	-0.01919687

NOISE AUTOCORRELATION COEFFICIENTS

0.0	7.45561390
1.0	5.46215000
2.0	5.73783060
3.0	4.89235190
4.0	4.69195430
5.0	4.05476330
6.0	3.54918120
7.0	3.89310420
8.0	3.26318760
9.0	3.21520160

SIGNAL TO NOISE RATIO = 2.00

SCALED AND SUMMED AUTOCORRELATIONS

0.0	0.38431336
1.0	0.30131213
2.0	0.16233441
3.0	0.02601536
4.0	-0.02829616
5.0	-0.02092389
6.0	0.00568244
7.0	0.02457978
8.0	0.01672474
9.0	0.01394983

BIBLIOGRAPHY

Berkhout, A. J., 1977, Least-Squares Inverse Filtering and Wavelet Deconvolution: Geophysics, Vol. 42, No. 7, p. 1369-1383.

DeVoogd, N., 1978, The Phase Property of a Finite Realization of Random Noise and its Influence on Deconvolution: Geophysical Prospecting, Vol. 26, p. 509-524.

Levinson, N., 1947, The Wiener RMS (Root Mean Square) Error Criterion in Filter Design and Prediction: J. Math. Phys., Vol. 25, p. 261-278.

Lindseth, R. O., 1979, Synthetic Sonic Logs - a Process for Stratigraphic Interpretation: Geophysics, Vol. 44, No. 1, p. 3-26.

Peacock, K. L., and Treitel, S., 1969, Predictive Deconvolution: Theory and Practice: Geophysics, Vol. 34, No. 2, p. 155-169.

Robinson, E. A., 1967, Multichannel Time Series Analysis with Digital Computer Programs: Holden-Day Inc., San Francisco, CA.

Schneider, W. A., 1979, Oral Communication: Colorado School of Mines, Golden, CO.

Schoenberger, M., 1974, Resolution Comparison of Minimum-Phase and Zero-Phase Signals: Geophysics, Vol. 39, No. 6, p. 826-833.

Sheriff, R. E., 1973, Encyclopedic Dictionary of Exploration Geophysics: Society of Exploration Geophysicists, Tulsa, OK.

Treitel, S. and Robinson, E. A., 1966, The Design of High Resolution Digital Filters: IEEE Transactions on Goescience Electronics, Vol. GE-4, No. 1.

Westlake, J. R., 1968, A Handbook of Numerical Matrix Inversion and Solution of Linear Equations: Chapter 4, John Wiley and Sons, Inc., New York, NY.

Wiggins, R. A., and Robinson, E. A., 1965, Recursive Solution to the Multichannel Filtering Problem: J. Geophysical Research, Vol. 70, p. 1885-1891.